Technische Universität Dresden

Database-Inspired Reasoning Problems in Description Logics With Path Expressions

Dissertation

zur Erlangung des akademischen Grades

Doctor rerum naturalium (Dr. rer. nat.)

vorgelegt an der

Technischen Universität Dresden

Fakultät Informatik

eingereicht von

Bartosz Bednarczyk

geboren am 20. März 1995 in Strzelce Opolskie, Polen

Betreuer:

Prof. Dr. rer. nat. Sebastian Rudolph, Technische Universität Dresden (*Betreuer*)

Dr hab. Emanuel Kieroński, University of Wrocław (*co-Betreuer*)

Gutachter:

Prof. Dr. rer. nat. Sebastian Rudolph, Technische Universität Dresden

Prof. Dr.in techn. Magdalena Ortiz, Technischen Universität Wien

Fachreferent:

Prof. Dr. Markus Krötzsch, Technische Universität Dresden

Datum der Einreichung: 24.04.2024
Datum der Verteidigung: 25.06.2024

Dresden, 25.06.2024

Version 1.0 (without hyperlinks). Always check ArXiV for the newest version.

Declaration of Authorship

Author: Bartosz Bednarczyk
Immatriculation number: 4841230
Title: Database-Inspired Reasoning Problems in Description Logics With Path Expressions

I hereby declare that except where specific reference is made to the work of others, the contents of this thesis are original and have not been submitted in whole or in part to obtain a degree or any other qualification neither in this nor in any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text. No other resources apart from the references and auxiliary means indicated in the bibliography were used in the development of the presented work.

(Place and date) (Author's signature)

Plus ultra.

I always try to avoid making mistakes in my papers, but despite my best efforts they sometimes happen. You are strongly encouraged to send any corrections and comments to bartosz.bednarczyk@cs.uni.wroc.pl.

Last updated 27.06.2024.

Abstract

Formal ontologies are of significant importance in artificial intelligence, playing a central role in the Semantic Web, ontology-based information integration, or peer-to-peer data management. In such scenarios, an especially prominent role is played by description logics (DLs) – a robust family of logical formalisms used to describe ontologies and serving as the logical underpinning of contemporary standardised ontology languages. To put knowledge bases to full use as core part of intelligent information systems, much attention is being devoted to the area of ontology-based data-access, with conjunctive queries and their generalisations such as positive conjunctive two-way regular path queries being employed as a fundamental querying formalism. The most expressive exemplars of description logics feature advanced constructors for roles and path expressions. Among the most powerful knowledge representation formalisms on the verge of decidability, are the DLs from the $\mathcal{Z}$ family. For its most expressive proponent, $\mathcal{ZOIQ}$ (a.k.a. $\mathcal{ALCH}b^{\mathsf{Self}}_{\mathsf{reg}}\mathcal{OIQ}$), featuring nominals ($\mathcal{O}$), role inverses ($\mathcal{I}$), and number restrictions ($\mathcal{Q}$), querying is undecidable and even decidability of knowledge-base satisfiability is open, owing to the intricate interplay of the three mentioned features. Restricting the interaction of $\mathcal{O}$, $\mathcal{I}$, and $\mathcal{Q}$ however (or excluding one of the features altogether) leads to beneficial model-theoretic properties, which give rise to upper bounds of ExpTime for knowledge-base satisfiability and 2ExpTime for querying.

Aiming for better understanding of the "expressive power versus computational complexity" trade-off for the $\mathcal{Z}$ family of DLs, we provide a more fine-grained complexity analysis for the query entailment problem over ontologies. In the thesis we focus on *tame* fragments of $\mathcal{ZOIQ}$, namely the fragments in which either one the three features from $\{\mathcal{I}, \mathcal{O}, \mathcal{Q}\}$ is dropped or the class of models is restricted to the so-called quasi-forests. We employ the query languages ranging from (unions of) conjunctive queries ((U)CQs) to positive two-way regular path queries (P2RPQs). We mostly follow the classical semantics of entailment, but we also provide several results in the "finite-model" scenario. The most important results of the thesis are summarised below.

1. We provide a complete classification of the complexity of the query entailment problem (for various query languages discussed above) for tamed fragments of $\mathcal{ZOIQ}$ under the classical semantics. This involves several new ingredients such as: (i) a uniform exponential-time algorithm based on Lutz's spoiler technique for the entailment of unions of conjunctive queries for $\mathcal{ALCH}b_{\mathsf{reg}}$, (ii) new lower bounds for (rooted and unrooted) conjunctive query entailment over $\mathcal{ALC}\mathsf{Self}$ ontologies, and (iii) a novel reduction from the entailment of P2RPQs to the satisfiability problem for tamed $\mathcal{ZOIQ}$, yielding a uniform 2ExpTime upper bound for all the considered logics. As a preliminary step towards lifting the above results to the realm of data complexity, we establish that the satisfiability of tamed $\mathcal{ZOIQ}$ is NP-complete.
2. Under the finite model semantics, we focus on UCQs only. With the proviso that the finite satisfiability problem for $\mathcal{ZIQ}$ is ExpTime-complete, we also provide a complete picture of the complexity of the query entailment problems. The key insight is that $\mathcal{ZOI}$ and $\mathcal{ZOQ}$ are finitely controllable.
3. We conclude the thesis by investigating the decidability border of further extensions of the $\mathcal{Z}$ family of DLs. Our goal is to understand whether the class of regular languages present in path expressions in $\mathcal{Z}$ is maximal for guaranteeing decidability of the underlying logic. We provide a series of undecidability results involving simple, non-regular languages (a subclass of visibly pushdown languages).

Our proofs rely on well-established model- and graph-theoretic definitions. What is more, most of them generalise (in a uniform way) and solidify multiple results known by the DL community. Our proofs are also easily adjustable to freshly defined logics, without the need to reproduce nearly-identical proofs.

Contents

1

Introduction

Contents

1.1 A Very Brief Introduction

Formal ontologies play a crucial role in artificial intelligence, serving as the backbone of various applications such as the Semantic Web [BHS05], health and life sciences [HRG+96, McG99], natural language processing [GBFF91], ontology-based information integration [Noy04], and peer-to-peer data management [CDGLR04]. In reasoning about graph-structured data, a significant role is played by *description logics* (DLs) [BHLS17], a robust family of logical formalisms serving as the logical foundation of contemporary standardised ontology languages, including OWL 2 by the W3C [GHM+08, HKP+12]. To put ontologies to full use as core part of intelligent information systems, much attention is being devoted to the area of ontology-based data-access, with conjunctive queries and their generalisations such as positive conjunctive two-way regular path queries being employed as a fundamental querying formalism [Ov12, BOv15]. Among many features present in extensions of the basic description logic $\mathcal{ALC}$, an especially useful one is $\cdot_{\mathsf{reg}}$, supported by the popular $\mathcal{Z}$-family of description logics [CEO09]. With $\cdot_{\mathsf{reg}}$ one can specify regular path constraints, allowing the user to navigate graph-structured data. In recent years many extensions of $\mathcal{ALC}_{\mathsf{reg}}$ for ontology engineering were proposed [BCOv14, COv16, Ort10, Ort23].

Among the most powerful knowledge representation formalisms on the verge of decidability, are the members of the $\mathcal{Z}$ family of DLs. For its most expressive proponent, $\mathcal{ZOIQ}$ (a.k.a. $\mathcal{ALCH}b^{\mathsf{Self}}_{\mathsf{reg}}\mathcal{OIQ}$), featuring nominals ($\mathcal{O}$), role inverses ($\mathcal{I}$), number restrictions ($\mathcal{Q}$), role hierarchies ($\mathcal{H}$), safe boolean combinations of roles (b), regular path constraints (reg), and the Self operator querying is undecidable and even decidability of knowledge base satisfiability is open, owing to the intricate interplay of the three mentioned features $\mathcal{O}$, $\mathcal{I}$, and $\mathcal{Q}$. Restricting the interaction of $\mathcal{O}$, $\mathcal{I}$, and $\mathcal{Q}$ however (or excluding one of the features altogether) leads to beneficial model-theoretic properties, which give rise to upper bounds of ExpTime for knowledge base satisfiability and 2ExpTime for querying. There is a two-fold motivation behind studying fragments of $\mathcal{ZOIQ}$. First, the members of the $\mathcal{Z}$ family of DLs encode the members of the $\mathcal{SR}$ family [CEO09, Prop. 5.1], *i.e.* the family of logical frameworks that underpin the logical core of OWL 2 by the W3C [HKP+12]. Hence, effective procedures and model-theoretic properties for extensions of $\mathcal{Z}$ transfer to the extensions of $\mathcal{SR}$. Second, the core fragment of $\mathcal{Z}$, called $\mathcal{ALC}_{\mathsf{reg}}$, turns out to be a notational variant of Propositional Dynamic Logic [FL79] – a popular and well-studied program specification language. Thus, the results on the $\mathcal{Z}$ family can reach and influence the community working on formal methods in computer science.

1.2 Research Objectives and Methodology

This dissertation focuses on a fine-grained complexity analysis of the query entailment problem over ontologies, as defined below (consult Preliminaries if needed). The word "finite" in the problem description below determines that instead of testing if all models of a knowledge-base $\mathcal{K}$ (usually composed of an ABox $\mathcal{A}$ and a TBox $\mathcal{T}$) satisfy a query q, we restrict our attention to all *finite* models. In addition to the above, we consider two ways of measuring the complexity of the query entailment problem: the *combined complexity* where both $\mathcal{K}$ and q contribute equally to the size of the input, and the *data complexity* where the $\mathcal{T}$ and q are fixed beforehand and only the ABox $\mathcal{A}$ varies.

(Finite) entailment of $\mathscr{Q}$ queries over $\mathcal{DL}$-Knowledge-Bases (KBs)

Parameters:	Description logic $\mathcal{DL}$, and a class of queries $\mathscr{Q}$
Input:	A KB $\mathcal{K} \coloneqq (\mathcal{A}, \mathcal{T})$ ($\mathcal{A}$ is an ABox and $\mathcal{T}$ is a $\mathcal{DL}$-TBox), a query $q \in \mathscr{Q}$
Question:	Does $\mathcal{K} \models_{(\mathrm{fin})} q$ hold?

A *very ambitious* goal is to provide a complete characterisation of the complexity of the query entailment problem under both semantics and both complexity measures for all fragments of $\mathcal{ZOIQ}$ (obtained by dropping some features from $\mathcal{ALCH}b^{\mathsf{Self}}_{\mathsf{reg}}\mathcal{OIQ}$) that extend $\mathcal{ALC}$, and query languages that are (i) *local* queries (namely conjunctive queries and their positive boolean combinations) (ii) regular queries (conjunctive two-way regular path queries and their positive boolean combinations). Quick calculations reveal that there are at least $2 \cdot 2 \cdot 2^7 \cdot 4 = 2048$ possible theorems to establish. Fortunately, many of them are already present in the literature [Ov12], being established in the last twenty years. On the negative side, fresh results tend to be incremental and sketchy, and the provided proofs are not *robust* (in the sense that slight extensions of the underlying logic require fresh proofs). We prefer to avoid such a fiddle.

Our thesis aims to partially solve the above problem in an *elegant* way. More precisely:

- Our approach should be *efficient*, namely it should cover multiple logics in a single go. For this reason, we mostly focus on so-called *tamed* $\mathcal{ZOIQ}$, a logic that serves as a common umbrella for $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, and $\mathcal{ZOI}$, which are all the currently-known maximal decidable fragments of $\mathcal{ZOIQ}$.
- Our approach should be *robust*, *i.e.* it should be possible to employ the developed techniques as a black box in the future for freshly defined logics, without the need to produce nearly-identical proofs (or even worse, leaving such proofs as an exercise to the reader). Thus, our primary aim is to design *meta-algorithms* rather than algorithms. This remark of course applies only to the upper bounds; the lower bound proofs should be as general as possible, *i.e.* should hold for logics of the form $\mathcal{ALC}\Theta$ for a smallest possible set of features $\Theta \subseteq \{\mathcal{I}, \mathcal{O}, \mathcal{Q}, b, \mathcal{H}, \mathsf{Self}, \cdot_{\mathsf{reg}}\}$.
- Our approach should rely on variants of *well-established formalisms* from *model theory.* In the thesis we employ tailored notions of *forest models* (including so-called *quasi-forest models*) and *unravellings*, and explore broad classes of graphs that locally resemble trees.
- Our approach towards querying should be *logic-independent*, namely the presented method should rely on the syntax of the query, but not on the syntax of the underlying logic (with the exception of core logics such as $\mathcal{ALC}$). This thesis advocates the satisfiability problem as *the main* object of study, providing numerous reductions from query entailment problem to satisfiability.

Unfortunately the author of this thesis failed to *fully* solve such an ambitious task. However, we believe that we made *significant progress* towards the full classification of syntactic fragments of $\mathcal{ZOIQ}$. We describe our setting and our research methodology below.

1. Our ontology languages of choice are any syntactic fragments of $\mathcal{ZOIQ}$ that extend $\mathcal{ALC}$, in which we disallow at least one of the three dangerous features $\mathcal{O}$, $\mathcal{I}$, and $\mathcal{Q}$. To explain our decision, we stress that obtaining tight complexity results on the query entailment problem even for $\mathcal{ALCOIQ}$ is currently out of reach, and remains the most challenging open problem in the DL community.
2. Our query languages are: conjunctive queries (CQs), unions thereof (UCQs), positive existential queries (PEQs), conjunctive regular path queries (CRPQs), and positive two-way regular path queries (P2RPQs). The lower bounds are mostly provided for CQs, PEQs, or CRPQs, while the matching upper bounds are given either for UCQs or for P2RPQs.
3. We mostly focus on the classical semantics, but we also provide results for the class of local queries under the finite model semantics. Rather than designing novel algorithms, we prove meta-theorems such as *finite controllability* that allow us to transfer the previously established results from the unrestricted setting to the finite one. Note that the entailment of regular queries over ontologies in the finite is very difficult to reason about. A quite recent breakthrough was done by Gutowski et al. [GGIM22], namely they proved that the *finite* entailment of conjunctive regular path queries for $\mathcal{ALC}$ is 2ExpTime-complete. Unfortunately, the their approach breaks if nominals or inverses are allowed in the ontology language, and does not seem to apply to the DL $\mathcal{Z}$ and beyond.
4. We mostly focus on combined complexity of the query entailment problem, but whenever possible, we also offer results on the data complexity. Quite recently, we obtained an important result showing that the satisfiability problem for $\mathcal{ZOIQ}$ over quasi-forest models is NP-complete w.r.t. the data complexity. We are confident that these results can be lifted (by adapting approach from a recent work of Jung et al. [GIJM23]) to the proof of coNP-completeness of the entailment of P2RPQs over any logic between $\mathcal{ALC}$ and (tamed) $\mathcal{ZOIQ}$. We leave the details for future work.

The current state of the art is quite intricate, but many relevant results can be found in an excellent survey by Ortiz and Šimkus [Ov12]. We are currently working on a simple web application that will illustrate all the relevant results from the literature. For instance, the complete picture of the combined complexity of the query entailment under the classical semantics can be summarised as follows. The relevant references here are the work of the author with his supervisor [BR23, BR19], as well as the works of Rudolph and Glimm [Rud16, RG10], Lutz [Lut08a], and Ngo et al. [NOv16].

Logic $\mathcal{DL}$	UCQs	PEQs	CRPQs	P2RPQs
$\mathcal{ALC} \subseteq \mathcal{DL} \subseteq \mathcal{ALCH}b_{\mathsf{reg}}\mathcal{Q}$	Exp-c.	2Exp-c	2Exp-c	2Exp-c
$\mathcal{ALC}\mathsf{Self} \cup \mathcal{ALCI} \cup \mathcal{ALCO} \subseteq \mathcal{DL} \subseteq \mathcal{ZIQ} \cup \mathcal{ZOQ} \cup \mathcal{ZOI}$	2Exp-c.	2Exp-c	2Exp-c	2Exp-c
$\mathcal{ALCOIQ} \subseteq \mathcal{DL} \subseteq \mathcal{ALCH}b^{\mathsf{Self}}\mathcal{IOQ}$	dec.	dec.	?	?
$\mathcal{ALCOIQ}_{\mathsf{reg}} \subseteq \mathcal{DL} \subseteq \mathcal{ZOIQ}$	?	?	undec.	undec.

1.3 Organisation of the Thesis. Our Results and Their Significance

The thesis is naturally split into four parts (not counting the technical background given in Chapter 2).

1.3.1 Part I: Query Entailment in Forest-Friendly Description Logics

We first consider the query entailment problem for less expressive members of the $\mathcal{Z}$ family of DLs. Our main motivation here was to establish the precise complexity of the finite and unrestricted entailment of (unions of) CQs for plain $\mathcal{Z}$. While the 2ExpTime upper bound was known [CEO09, Thm. 4.3] for quite a long time, it was not *a priori* clear whether this upper bound is tight, especially in the light of $\mathcal{ALC}$ and $\mathcal{S}$ (without transitivity in queries) having lower complexity of the query entailment than the other members of their family (see Chapter 5 for a more detailed discussion). In Chapter 6 we show that this is indeed the case, presenting a novel lower bound for $\mathcal{ALC}$ with the Self operator.

Main Theorem 1

The finite and unrestricted conjunctive query entailment problems for $\mathcal{ALC}\mathsf{Self}$-TBoxes are 2ExpTime-hard. Thus, both finite and unrestricted CQ entailment problem over $\mathcal{Z}$-TBoxes is 2ExpTime-hard.

Our proof goes via encoding of computational trees of alternating Turing machines working in exponential space and follows a general hardness-proof-scheme by Lutz [Lut08a, Section 4]. However, to adjust the schema to $\mathcal{ALC}\mathsf{Self}$, novel ideas are required: the ability to speak about self-loops is exploited to produce a single query that traverses trees in a root-to-leaf manner and to simulate disjunction inside CQs, useful to express that certain paths are repeated inside the tree. We supplement the lower bound given above with a tight exponential-time meta-algorithm for fragments of the DL $\mathcal{Z}$ that do not involve the Self operator. We start by introducing the classes $\mathscr{C}_{\emptyset\mathsf{fr}}$ and $\mathscr{C}_{\emptyset\mathsf{fr}}^{\mathsf{fin}}$ of *locally-forward* and *finitely locally-forward* description logics, called in this thesis also (finitely) $\emptyset$-forest-friendly. These are classes of ontology languages extending[1] $\mathcal{ALC}^{\cap}$ in which any description logic $\mathcal{DL} \in \mathscr{C}_{\emptyset\mathsf{fr}}^{\mathsf{fin}}$ (resp. $\mathcal{DL} \in \mathscr{C}_{\emptyset\mathsf{fr}}$) enjoys a property that each (finitely) satisfiable $\mathcal{DL}$-knowledge-base $\mathcal{K}$ has a (finite) model that locally resembles (formalised by means of homomorphic equivalence) a forward tree. Afterwards, we revisit a classical algorithm for conjunctive querying $\mathcal{ALCHQ}$ designed by Lutz [Lut08b] based on the *spoiler technique*, and improve the technique in several ways: (i) our algorithm can be applied to unions of CQs rather than plain CQs, (ii) our algorithm works for any logic $\mathcal{DL} \in \mathscr{C}_{\emptyset\mathsf{fr}}$, (iii) our algorithm can be applied to the finite query entailment problem for logics from $\mathscr{C}_{\emptyset\mathsf{fr}}^{\mathsf{fin}}$. Despite the employment of Lutz's technique, most of our proofs are done from scratch in order to adjust them to the new, more abstract and more general, setting. In particular, in the case of finite model reasoning, or in reasoning about logics with global cardinality constraints, the intended models are no longer trees (and hence the proof sketches by Lutz that work for $\mathcal{SHQ}$ cannot be taken for granted in our setting). We establish the following theorem.

Main Theorem 2

Let $\mathcal{DL}$ be any (finitely) $\emptyset$-forest-friendly description logic with ExpTime-complete (finite) knowledge-base satisfiability problem. Then the (finite) entailment problem for unions of conjunctive queries over $\mathcal{DL}$-knowledge-bases is ExpTime-complete. In particular both finite and unrestricted entailment of unions of conjunctive queries in $\mathcal{ZQ}$ without the Self operator is ExpTime-complete.

In recent years, it has become apparent that various modelling features of DLs affect the complexity of conjunctive query entailment in a rather strong sense. It was first shown independently by Ortiz et al. [OvE08] and by Lutz [Lut08a] that CQ entailment is exponentially harder than the consistency problem for $\mathcal{ALC}$ extended with inverse roles ($\mathcal{I}$). Shortly after, a combination of transitivity and role hierarchies ($\mathcal{SH}$) was shown as another culprit of higher worst-case complexity of reasoning [ELOv09]. Finally, also nominals ($\mathcal{O}$) turned out to be problematic [NOv16]. Nevertheless, there are also more benign DL constructs regarding the complexity of CQ entailment. Examples are counting ($\mathcal{Q}$) [Lut08b] (the complexity stays the same even for expressive arithmetical constraints [BBR20]), role-hierarchies alone ($\mathcal{H}$) [EOv12] or a tamed use of higher-arity relations [Bed21a]. Our results, summarised below, provide a complete classification on when the complexity of the query entailment is the same as the complexity of the satisfiability of the underlying description logic.

	Feature θ with name	$\mathcal{ALC}\theta^{\cap} \in \mathscr{C}_{\emptyset\mathsf{fr}}$?	SAT=CQEnt?
good	functionality $\mathcal{F}$ and various counting: $\mathcal{N}/\mathcal{Q}/\mathcal{SCC}$	✓ [new!]	
	trans. closure $\cdot^*$, regular expr. $\cdot_{\mathsf{reg}}$, fixed-points μ		
	role hierar. $\mathcal{H}$, safe boolean comb. of roles b		
bad	inverses of roles $\mathcal{I}$	✗	
	nominals $\mathcal{O}$		
	transitivity $\mathcal{S}$, complex role inclusions $\mathcal{R}$		
bad	self-loops Self	✗ [new!]	

We stress that our technique can be used as a black box also for freshly defined "forward" logics. To do so, one may need to check if a given logic belongs to the class of (finite) $\emptyset$-forest-friendly DLs. This task

[1]The use of a role conjunction operator $^{\cap}$, allowing for specifying that two elements are connected via a conjunction of roles, is essential for our querying algorithm to work. See Preliminaries for the definition.

is however relatively difficult, due to the quite technical definition of (finitely) $\emptyset$-forest-friendliness. To simplify such a process, we propose several sufficient conditions based on novel model-theoretic notions of unravellings. Their definitions are given in Chapter 5. Hence, with a bit of luck, this relatively tedious task boils down to routine proofs involving structural induction. What is more, each of our constructions is supplemented with a handy list of properties preserved by our unravellings, that can simplify the reasoning even further. As an example, we provide tight complexity bounds for statistical extensions of $\mathcal{ALC}$.

Main Theorem 3

Finite entailment of unions of conjunctive queries for $\mathcal{ALCSCC}$ with ERCBoxes is ExpTime-complete.

Last but not least, our technique provides tight complexity bounds also for logics involving the Self operator and/or the inverse operator $\mathcal{I}$. Note however that in this case the complexity jumps exponentially. The appropriate model-theoretic definitions are slightly different, but we made a lot of effort to make our constructions understandable. For instance, by employing the revised version of Lutz's spoiler technique and a novel method of pumping models from Section 5.4, we derive an exponential-time reduction from the (finite) query entailment for $\mathcal{ZIQ}$ to its finite satisfiability problem.

Part I of this thesis revises and generalises material that was published in the following journals:

[BR23] B. Bednarczyk and S. Rudolph, "How to Tell Easy from Hard: Complexity of Conjunctive Query Entailment in Extensions of $\mathcal{ALC}$", in *Journal of Artificial Intelligence Research* (2023).

[Bed21c] B. Bednarczyk, "Statistical $\mathcal{EL}$ is ExpTime-complete", *Information Processing Letters* (2021).

The JAIR paper [BR23] is in turn based on the following conference papers and unpublished drafts:

[BBR20] F. Baader, B. Bednarczyk and S. Rudolph, "Satisfiability and Query Answering in Description Logics with Global and Local Cardinality Constraints", in *24th European Conference on Artificial Intelligence* (ECAI 2020).

[BR22] B. Bednarczyk and S. Rudolph, "The Price of Selfishness: Conjunctive Query Entailment for $\mathcal{ALC}^{\mathsf{Self}}$ Is 2ExpTime-Hard", in *36th AAAI Conference on Artificial Intelligence* (AAAI 2022).

[Bed21b] B. Bednarczyk, "Lutz's Spoiler Technique Revisited: A Unified Approach to Worst-Case Optimal Entailment of Unions of Conjunctive Queries in Locally-Forward Description Logics", in *ArXiV*.

1.3.2 Part II: Quasi-Forest Satisfiability of $\mathcal{ZOIQ}$

In Chapter 7 we study the data complexity of the satisfiability problem for decidable sublogics of $\mathcal{ZOIQ}$. In particular we establish NP-completeness of $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, and $\mathcal{ZOI}$. As all the mentioned DLs possess the *quasi-forest model property* (*i.e.* every satisfiable knowledge-base (KB) has a forest-like model), for the uniformity of our approach we focus on the satisfiability of $\mathcal{ZOIQ}$ over quasi-forests. Calvanese et al. [CEO09] proved that quasi-forest-satisfiability of $\mathcal{ZOIQ}$ is ExpTime-complete w.r.t. the combined complexity. Unfortunately, their approach is automata-based and relies on an internalisation of ABoxes inside $\mathcal{ZOIQ}$-concepts, and thus cannot be used to infer tight bounds w.r.t. the data complexity.

We employ the algorithm of Calvanese et al. as a black box and design a novel algorithm for quasi-forest-satisfiability of $\mathcal{ZOIQ}$-KBs. In our approach we construct a quasi-forest model step-by-step, *i.e.* we construct its root part (the *clearing*) separately from its subtrees. Our algorithm first pre-computes (an exponential w.r.t. the size of the TBox but of constant size if the TBox is fixed) set of quasi-forest-satisfiable $\mathcal{ZOIQ}$-concepts that indicate possible subtrees that can be "plugged in" to the clearing of the intended model. Then it guesses (in NP) the intended clearing and verifies its consistency in PTime based on the pre-computed concepts and roles. For the feasibility of our "modular construction" a lot of bookkeeping needs to be done. Most importantly, certain *decorations* are employed to decide the satisfaction of *automata concepts* and *number restrictions* by elements in an incomplete and fragmented forest. The first type of decorations, given an automaton $\mathcal{A}$, aggregate information about existing paths realising $\mathcal{A}$ and starting at one of the roots of the intended model. As a single such path may visit several subtrees, we cut such paths into relevant pieces and summarise them by means of “shortcut” roles and $\mathcal{ZOIQ}$-concepts describing paths fully contained inside a single subtree. The second type of decorations “localise” counting in the presence of nominals, as the nominals may have successors outside their own subtree and the clearing. These two “small tricks”, obfuscated by various technical difficulties, are the core ideas behind our quasi-forest-satisfiability algorithm.

Main Theorem 4

The satisfiability problem for (tamed fragment of) $\mathcal{ZOIQ}$ is NP-complete w.r.t. the data complexity. In particular, the satisfiability problems for $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, and $\mathcal{ZOI}$ are NP-complete.

We conclude with Chapter 8 by presenting how our algorithm can be adapted to the entailment problem of rooted queries over $\mathcal{ZIQ}$. The key idea here is to guess an “initial segment” of a quasi-forest with no query match, and check if it can be extended to a full model of the input KB. A novel coNExpTime lower bound for $\mathcal{ALC}\mathsf{Self}$ is also provided.

Main Theorem 5

The entailment problem for unions of rooted conjunctive queries over $\mathcal{ZIQ}$-KBs is coNExpTime-complete. The lower bound holds already for $\mathcal{ALC}$ extended with the Self operator.

The content of Part II will appear in:

[Bed24a] B. Bednarczyk, “Data Complexity in Expressive Description Logics With Path Expressions”, accepted to the *33rd International Joint Conference on Artificial Intelligence* (IJCAI 2024).

1.3.3 Part III: Entailment of Queries in Expressive Fragments of $\mathcal{ZOIQ}$

The next part of the thesis addresses the problem of entailment of regular queries (more precisely: positive two-way regular path queries, P2RPQs) in expressive members of the $\mathcal{Z}$ family of DLs. To unify our approach, we focus on tamed $\mathcal{ZOIQ}$, namely the fragment of $\mathcal{ZOIQ}$ that possesses the quasi-forest counter-model property (also discussed in the previous section). This allows us to handle $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, and $\mathcal{ZOI}$ in a single proof. In Chapter 9 we employ a certain “proof-like calculus” for deriving query matches in quasi-forest models. Its application allows us to exponentially reduce the P2RPQ entailment problem over tamed $\mathcal{ZOIQ}$ to its satisfiability problem, exponentially improving previous results [CEO09, Thm. 4.3] concerning $\mathcal{ZIQ}$ and $\mathcal{ZOQ}$ (which relied on the unary encoding of counters in number restrictions).

Main Theorem 6

Entailment of positive two-way regular path queries over (tamed) $\mathcal{ZOIQ}$-KBs (in particular, KBs written in $\mathcal{ZOQ}$, $\mathcal{ZIQ}$, $\mathcal{ZOI}$) is 2ExpTime-complete.

These results were already presented in:

[BR19] B. Bednarczyk and S. Rudolph, “Worst-Case Optimal Querying of Very Expressive Description Logics with Path Expressions and Succinct Counting”, in *Twenty-Eighth International Joint Conference on Artificial Intelligence* (IJCAI 2019).

In Chapter 10 we study whether we can lift the results concerning the entailment of local queries (positive existential queries, PEQs, namely positive boolean combinations of conjunctive queries) to the description logics $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, and $\mathcal{ZOI}$, *i.e.* the most expressive fragments of $\mathcal{ZOIQ}$. Our main result here is that $\mathcal{ZOQ}$ and $\mathcal{ZOI}$ are *finitely controllable*, meaning that a PEQ is entailed if and only if it is finitely entailed. Alternatively, the finite controllability property tells us that the existence of a countermodel for the entailment of the query is equivalent to the existence of a finite such countermodel. This model-theoretic property allows us to transfer 2ExpTime-completeness of the entailment problem for PEQs (that follows either from the original paper on $\mathcal{ZOIQ}$ [CEO14] or from the previous section of the thesis) for $\mathcal{ZOQ}$ and $\mathcal{ZOI}$ to the finite-world scenario. What is more, by applying the spoiler technique intensively studied in Part I, we can exponentially reduce the finite PEQ entailment problem for $\mathcal{ZIQ}$ to its finite knowledge-base satisfiability problem. This is optimal, modulo the conjecture that knowledge-base satisfiability problem for $\mathcal{ZIQ}$ in the finite is ExpTime-complete, which is quite probable. Unfortunately, we were not able to prove this conjecture and even show the decidability of the finite satisfiability problem for $\mathcal{ZIQ}$-TBoxes. This is probably due to the fact that the decidability results for logics with inverses and counting are usually obtained via integer programming, which does not seem to be suitable for dealing with regular path constraints.

Main Theorem 7

The logics $\mathcal{ZOQ}$ and $\mathcal{ZOI}$ are finitely controllable. This implies that the problem of finite entailment of positive existential queries for them is 2ExpTime-complete. Moreover, if the finite satisfiability for $\mathcal{ZIQ}$ is ExpTime-complete, then finite entailment of PEQs over $\mathcal{ZIQ}$-KBs is 2ExpTime-complete.

Roughly speaking, our proof method is as follows. We start from a finite-branching quasi-forest model $\mathcal{I}$ for a knowledge-base $\mathcal{K}$ (written either in $\mathcal{ZOI}$ or $\mathcal{ZOQ}$) that violates a query q (say, with K atoms). Without loss of generality, we can assume that whether an element requires a witness for the satisfaction of some existential restriction, then such a witness is always reachable by a path going downward the forest (and then possibly "jumping" to a nominal). The key ingredient in our construction is the identification of certain substructures of $\mathcal{I}$, that are based on *downward types*, *i.e.* the isomorphism types of subtrees of depth K rooted at some element from $\mathcal{I}$. For every downward type π we select one representative, then extend it in a minimal way by providing witnesses for its root for all existential restrictions of the form $\exists\mathcal{A}.\mathrm{C}$ appearing in $\mathcal{K}$, and finally make such a substructure sibling- and parent-closed. Note that a path witnessing the satisfaction of existential restrictions of the form $\exists\mathcal{A}.\mathrm{C}$ are finite by definition. We call such modified representatives *components*. The intended finite model of $\mathcal{K}$ that violates q is constructed by taking a sufficiently many isomorphic copies of components, and them linking them in a clever way.

The above results were already presented in the following paper:

[BK22] B. Bednarczyk and E. Kieroński, "Finite Entailment of Local Queries in the $\mathcal{Z}$ Family of Description Logics", in *Thirty-Sixth AAAI Conference on Artificial Intelligence* (AAAI 2022).

We stress that obtaining complexity results (or even decidability) for the finite entailment of (non-local) queries over extensions of $\mathcal{ALC}$ is currently beyond the author's reach. See the discussion in Section 1.2. We also apologise for the content of Part III not being as polished as the rest of the thesis. In the nearest future we aim to rewrite the material collected here and publish it in a journal.

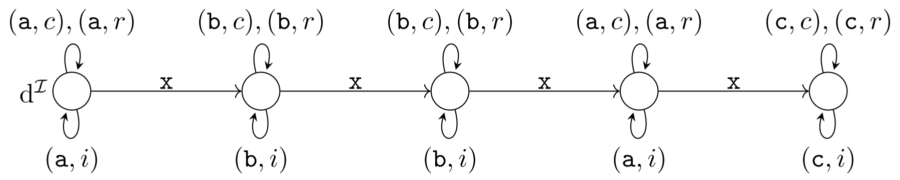
1.3.4 Part IV : A Step Beyond the $\mathcal{Z}$ Family of DLs

The last part of the thesis focuses on further extensions of $\mathcal{ALC}_{\mathsf{reg}}$, the core fragment of all the members of the $\mathcal{Z}$ family of DLs. In Chapter 11 we generalise path expressions present in $\mathcal{ALC}_{\mathsf{reg}}$ by allowing for non-regular languages. One prominent candidate is the class of visibly pushdown languages [AM09], a well-behaved subclass of deterministic context-free languages. A canonical visibly pushdown language is $r^{\#}s^{\#} := \{r^n s^n \mid n \in \mathbb{N}\}$. While the extension of $\mathcal{ALC}_{\mathsf{reg}}$ with path expressions involving visibly pushdown languages ($\mathcal{ALC}_{\mathsf{vpl}}$) was shown to be decidable by Löding and Serre [LLS07], we obtained several undecidability results for extensions of $\mathcal{ALC}_{\mathsf{vpl}}$ with popular features supported by W3C ontology languages.

Main Theorem 8

The concept satisfiability problem is undecidable for $\mathcal{ALC}_{\mathsf{vpl}}$ extended either with nominals or the Self operator. What is more, the entailment problem for conjunctive regular path queries, extended with atoms involving the non-regular language $r^{\#}s^{\#}$, is undecidable already for $\mathcal{ALC}$-TBoxes.

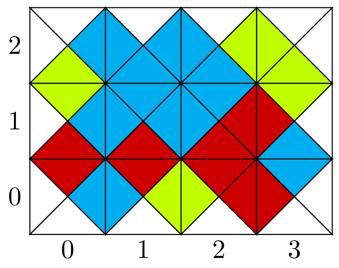
For the case of $\mathcal{ALC}_{\mathsf{vpl}}$ with the Self operator, we provide a reduction from the undecidable problem of non-emptiness of the intersection of deterministic one-counter automata (DOCA) [Val73, p. 75].

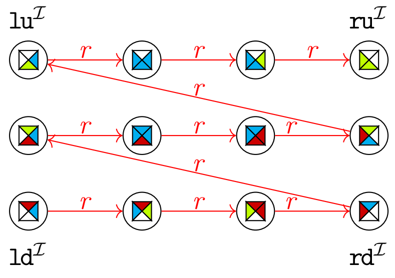


$(\mathtt{a}, c), (\mathtt{a}, r)$
$(\mathtt{b}, c), (\mathtt{b}, r)$
$(\mathtt{b}, c), (\mathtt{b}, r)$
$(\mathtt{a}, c), (\mathtt{a}, r)$
$(\mathtt{c}, c), (\mathtt{c}, r)$
$\mathrm{d}^{\mathcal{I}}$
x
x
x
x
$(\mathtt{a}, i)$
$(\mathtt{b}, i)$
$(\mathtt{b}, i)$
$(\mathtt{a}, i)$
$(\mathtt{c}, i)$

The key idea is that every language recognized by DOCA can be made visibly-pushdown in a projective sense, namely by enlarging the alphabet, for each letter $\mathtt{a}$, with a trio of letters $(\mathtt{a}, c)$, $(\mathtt{a}, i)$, and $(\mathtt{a}, r)$ representing the action of DOCA on the stack after reading such letters $\mathtt{a}$. In our reduction, we represent words by means of linear structures under the name of *metawords*. A metaword representing $\mathtt{abbac}$ is given above. We then label metawords with concepts, representing fragments of accepting runs of DOCAs.

In the case of $\mathcal{ALC}_{\mathsf{vpl}}$ with nominals and the query entailment problem, we provide a reduction from the tiling problem of either a finite rectangle or an octant. For instance, in the case of $\mathcal{ALC}_{\mathsf{vpl}}\mathcal{O}$, we represent the solution to the tiling problem as so-called snakes (depicted on the right hand side).





$\mathtt{lu}^{\mathcal{I}}$
$\mathtt{ru}^{\mathcal{I}}$
$\mathtt{ld}^{\mathcal{I}}$
$\mathtt{rd}^{\mathcal{I}}$
r

The key idea behind the proofs is to employ the non-regular language $r^{\#}s^{\#} := \{r^n s^n \mid n \in \mathbb{N}\}$ to *measure* distances between elements and ensure that they are *equal*. For more details consult Chapter 11.

This part of the thesis is based on the following journal paper:

[Bed24b] B. Bednarczyk, "Exploring Non-Regular Extensions of Propositional Dynamic Logic with Description-Logics Features", in *Logical Methods in Computer Science* (2024),

which is the full version of:

[Bed23] B. Bednarczyk, "Beyond $\mathcal{ALC}_{\mathsf{reg}}$: Exploring Non-Regular Extensions of PDL with Description Logics Features", in *Logics in Artificial Intelligence - 18th European Conference* (JELIA 2023).

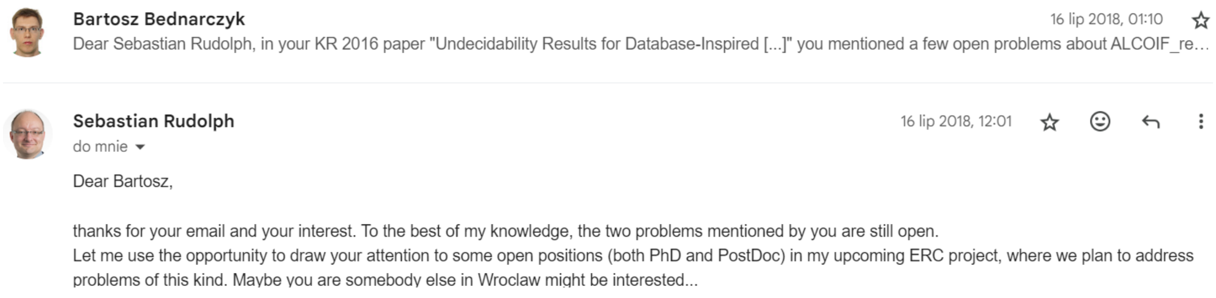
This work also received the **best student paper award** at JELIA 2023.

1.4 Acknowledgements

Due to the incredible luck I've had in life to meet the right people at the right time and in the right place, it would be impossible to thank all of them here. Therefore, the idea here is restrict myself to only a small subset of such people and keep these acknowledgments as short as possible as the written words cannot fully express my gratitude to all the people mentioned here. tl;dr

• Starting off, I would like to thank all the people related to my home town, Wrocław. Especially, I would like to thank Krzysztof Loryś and Witold Charatonik, who had an incredibly significant impact on my life since the early years of my undergraduate studies. I would like to thank Prof. Loryś for always finding time for me despite his many daily responsibilities, for providing me with guidance, and for allowing me to prove myself as a high-school teacher. It is mainly thanks to him that I did not succumb to the world of big corporations and decided to pursue work in academia. To Witek, my "academic parent", I owe a lot. He introduced me to the realm of computational logic, gave me the first open problem to solve and we published together my very first research paper. He always gave me advice whenever I needed it and helped me whenever he was able to do so (both in private life and the academic one). It was a lot of fun for me to work together with him over those many years, solving problems together, supervising students, and teaching my favourite undergraduate course in logic. This whole experience was extremely enjoyable and I will probably never be able to repay him for all his kindness. I thank the members of the Institute of Computer Science of the University of Wrocław for providing a friendly and creative atmosphere there. During all these ten years this made me feel like I was a part of this big family, and made the CS department my second home. In particular, I would like to thank all these with whom I had the pleasure of talking more often or collaborating (in random order): Darek and Gosia Biernaccy, Artur Jeż, Emanuel Kieroński, Witek Charatonik (already mentioned above), Kuba Michaliszyn, Janek Otop, Tomasz Wierzbicki, Piotrek (Nalewaja | Polesiuk | Wieczorek | Witkowski), Olek Łukasiewicz, Tomasz Judziński, Jerzy Marcinkowski, and Leszek Pacholski (the founder of this amazing place).

• I would like to move next to my main supervisor, Sebastian Rudolph. I believe it would be really challenging to find a supervisor more exemplary than Sebastian. Therefore, I am deeply grateful that he invited me to join his academic group in Dresden and began collaborating with me as early as the end of 2018. He is someone universally liked, and the atmosphere within his group is so friendly and conducive to productivity. Hence, despite receiving offers from prestigious institutions such as Oxford (my apologies to Michael Benedikt), I chose to pursue my doctorate under his guidance, which was definitely one of the best decisions in my life. Over the past five years, together with Sebastian we have managed to publish numerous papers together. Sebastian has always been willing to help, offering valuable insights, suggesting improvements to my methodologies, and frequently correcting my poor-quality writings. His patience with me has been remarkable, and he has granted me a considerable degree of academic freedom, allowing me to choose projects independently. Sebastian always made time for me, even when it was scarce, and supported my participation in scientific trips and conferences whenever I wished. I recognize that such generosity is not commonplace among supervisors. Sebastian has imparted a wealth of knowledge to me. He has consistently engaged with me in inspiring discussions about our proofs, provided feedback on sections of my work, offered guidance on my academic career, and provided invaluable advice. It has been common for Sebastian to dedicate late evenings and weekends to reviewing subsequent chapters of my work. Sometimes, we even worked together into the early hours of the morning to ensure the timely submission of articles before deadlines. Although his mentorship may not have taught me the importance of maintaining a work-life balance, I am eager to express my gratitude for his unwavering support and guidance. Despite my imperfections and occasional stubbornness, I sincerely hope that I have met at least some of his expectations as both a doctoral candidate and a collaborative colleague. Finally, for the sake of completeness, the following email is one of the coolest ways to hire academic staff.



Bartosz Bednarczyk 16 lip 2018, 01:10

Dear Sebastian Rudolph, in your KR 2016 paper "Undecidability Results for Database-Inspired [...]" you mentioned a few open problems about ALCOIF_re…

Sebastian Rudolph 16 lip 2018, 12:01

do mnie

Dear Bartosz,

thanks for your email and your interest. To the best of my knowledge, the two problems mentioned by you are still open.
Let me use the opportunity to draw your attention to some open positions (both PhD and PostDoc) in my upcoming ERC project, where we plan to address problems of this kind. Maybe you are somebody else in Wroclaw might be interested...

I thank people from Dresden, especially the ones from the CL group, who made my stays in Dresden more enjoyable and scientifically beneficial. In particular, I thank (in random order): Ramona Behling, Dörthe Arndt, Piotr Gorczyca, Luisa Herrmann, Jonas Karge, Pascal Kettmann, Tim Lyon, Hannes Straß, Shima Asaadi, Elisa Böhl, Martin Diller, Faiq Miftakhul Falakh, Thomas Feller, Sarah Gaggl, Dagmar Gromann, Lucía Gómez Álvarez , and Lukas Schweizer. I also thank Franz Baader and Markus Krötzsch for their support and scientific exchanges.

• I am deeply indebted to Stéphane Demri, my former mentor from ENS Paris-Saclay. It was under his guidance that I co-authored my very first LICS paper, and it is owing to his mentorship that my proofs exhibit such meticulous detail (let me refer to this as the "Cachan style"). I deeply regret not having had more time to spend with him, but perhaps someday I will be able to make up for it. At this point, I would acknowledge my former Cachan teachers: Sylvain Schmitz, Paul Gastin, Philippe Schnoebelen, David Baelde, and Paul-André Melliès with whom it was a pleasure to have classes with. Especially I owe Sylvain for all his help with enrolling me as a master's student in Cachan.

• Next, I would like to thank my second supervisor, Emanuel Kieroński for accepting me as his PhD student, supervising students together, fruitful collaboration since the end of my BSc studies, and valuable feedback on our past works. I also grateful to him for employing me in his research grant when I was a master student, which helped me to stay focused on doing research.

• I am immensely grateful to Magdalena Ortiz for agreeing to be the reviewer of this doctoral dissertation. As you can easily notice, the thesis is extremely long, and hence reading it will probably take ages and be very painful. I also thank her for offering me a job at TU Wien after the completion of my doctorate. Let's hope we will meet in Vienna from October 2024 onwards.

• I thank all my amazing co-authors: Sebastian Rudolph, Stéphane Demri, Emanuel Kieroński, Piotr Witkowski, Witold Charatonik, Alessio Mansutti, Raul Fervari, Franz Baader, Reijo Jaakkola, Jakub Michaliszyn, Tony Tan, Maja Orłowska, Ian Pratt-Hartmann, Daumantas Kojelis, Anna Pacanowska, Robert Ferens, Julien Grange, Mateusz Urbanczyk, Oskar Fiuk, and Piotr Ostropolski-Nalewaja.

Lastly, I express my gratitude to all my friends and family, as well as all the people who were not mentioned here due to space reasons. In particular, I would like to thank Kamil Matuszewski for his constant support and friendship.

Research published in this thesis was supported by the ERC Consolidator Grant No. 771779 (DeciGUT).

2

Preliminaries

Contents

Motivation

The role of this chapter is to introduce and fix the notations used throughout the thesis. Let us be frank about the expectations: this work *does not intend* to be a textbook (as there are far better books on the market) and is dedicated to researchers and experienced PhD students working in the area of computational logic. We assume familiarity with: naïve set theory [WJ96, Part 1], basics on syntax and semantics of first-order logic [Lib04, Sec. 2.1], standard description logics [BHLS17, Sec. 2.1–2.3], query entailment over description logic ontologies [Ov12, Sec 1–6, Sec. 8], regular and context-free languages [Sip13, Sec. 1–2], Turing machines [Sip13, Sec. 3], computability and (un)decidability [Sip13, Sec. 4–5], as well as the usual complexity classes [Lib04, Sec. 2.3] including the alternating complexity classes [Sip13, Sec. 10.3].

As usual, we employ $\mathbb{N}$ to denote the set of *non-negative* integers and $\mathbb{Z}_n$ to denote the set $\{0, 1, \ldots, n-1\}$. We use "square bracket" notation to denote images of functions, *e.g.* $f[X]$ denotes the image of X via f.

2.1 Basics on the Description Logic $\mathcal{ALC}$

Until the end of the thesis we fix countably infinite pairwise disjoint sets of *individual names* $\mathbf{N_I}$, *concept names* $\mathbf{N_C}$, and *role names* $\mathbf{N_R}$. We usually denote individual names with $\mathtt{a}, \mathtt{b}, \mathtt{c}$, concept names with A, B, C, and role names with r, s, t. Starting from $\mathbf{N_C}$ and $\mathbf{N_R}$, the set $\mathbf{C}_{\mathcal{ALC}}$ of $\mathcal{ALC}$-*concepts* is built using the following concept constructors: *negation* ($\neg$), *conjunction* ($\sqcap$), *existential restriction* ($\exists$), and the *bottom concept* ($\bot$), with the grammar:

$$\mathrm{C}, \mathrm{D} \ ::= \bot \ \mid \ \mathrm{A} \ \mid \ \neg\mathrm{C} \ \mid \ \mathrm{C} \sqcap \mathrm{D} \ \mid \ \exists r.\mathrm{C},$$

where $\mathrm{C}, \mathrm{D} \in \mathbf{C}_{\mathcal{ALC}}$, $\mathrm{A} \in \mathbf{N_C}$ and $r \in \mathbf{N_R}$. As convenient abbreviations, we often employ disjunction $\mathrm{C} \sqcup \mathrm{D} := \neg(\neg\mathrm{C} \sqcap \neg\mathrm{D})$, universal restriction $\forall r.\mathrm{C} := \neg\exists r.\neg\mathrm{C}$, the top concept $\top := \neg\bot$, and – not quite as widely used – "inline-implication" $\mathrm{C} \rightarrow \mathrm{D} := \neg\mathrm{C} \sqcup \mathrm{D}$ and "inline-equivalence" $\mathrm{C} \leftrightarrow \mathrm{D} := (\mathrm{C} \rightarrow \mathrm{D}) \sqcap (\mathrm{D} \rightarrow \mathrm{C})$. The set $\mathbf{C}_{\mathcal{EL}}$ of $\mathcal{EL}$-*concepts* is defined similarly, by dropping negation from the above grammar.

Assertions are expressions of the form $\mathrm{C}(\mathsf{a})$, $r(\mathsf{a}, \mathsf{b})$, or $\neg r(\mathsf{a}, \mathsf{b})$ for individual names $\mathsf{a}, \mathsf{b} \in \mathbf{N_I}$, concepts $\mathrm{C} \in \mathbf{C}_{\mathcal{ALC}}$, and role names $r \in \mathbf{N_R}$. An $\mathcal{ALC}$-*general-concept-inclusion* ($\mathcal{ALC}$-GCI) has the form $\mathrm{C} \sqsubseteq \mathrm{D}$ for $\mathcal{ALC}$-concepts $\mathrm{C}, \mathrm{D} \in \mathbf{C}_{\mathcal{ALC}}$. We use $\mathrm{C} \equiv \mathrm{D}$ as a shorthand for the joint occurrence of the two GCIs $\mathrm{C} \sqsubseteq \mathrm{D}$ and $\mathrm{D} \sqsubseteq \mathrm{C}$. An $\mathcal{ALC}$-*knowledge-base* ($\mathcal{ALC}$-KB) $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ is composed of a finite set $\mathcal{A}$ (called *ABox*) of assertions and a finite set $\mathcal{T}$ (called $\mathcal{ALC}$-*TBox*) of $\mathcal{ALC}$-GCIs. We call the elements of $\mathcal{A} \cup \mathcal{T}$ *axioms*. The set of all individual names appearing in $\mathcal{K}$ is denoted with $\mathsf{ind}(\mathcal{K})$.

The semantics of $\mathcal{ALC}$ is defined via *interpretations* $\mathcal{I} := (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ composed of a non-empty set $\Delta^{\mathcal{I}}$, called the *domain of* $\mathcal{I}$, and an *interpretation function* $\cdot^{\mathcal{I}}$, mapping individual names to elements of $\Delta^{\mathcal{I}}$, concept names to subsets of $\Delta^{\mathcal{I}}$, and role names to subsets of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. This mapping is inductively extended to concepts via

$$\begin{aligned} \bot^{\mathcal{I}} &:= \emptyset, \\ (\neg\mathrm{C})^{\mathcal{I}} &:= \Delta^{\mathcal{I}} \setminus \mathrm{C}^{\mathcal{I}}, \\ (\mathrm{C} \sqcap \mathrm{D})^{\mathcal{I}} &:= \mathrm{C}^{\mathcal{I}} \cap \mathrm{D}^{\mathcal{I}}, \\ (\exists r.\mathrm{C})^{\mathcal{I}} &:= \{\mathrm{d} \mid \exists \mathrm{e} \in \mathrm{C}^{\mathcal{I}}.\ (\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}\}, \end{aligned}$$

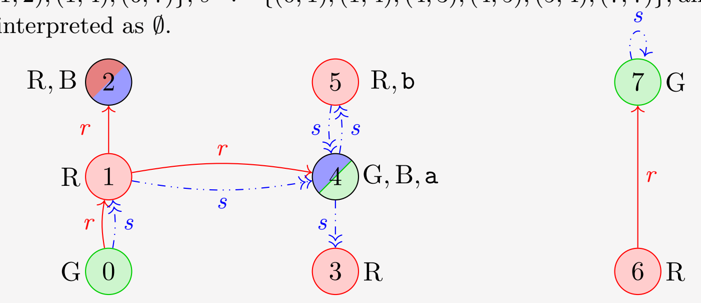
and finally used to define *satisfaction* of assertions and GCIs in an interpretation $\mathcal{I}$ by letting

$$\begin{array}{lll} \mathcal{I} \models \mathrm{C} \sqsubseteq \mathrm{D} & \textit{if and only if} & \mathrm{C}^{\mathcal{I}} \subseteq \mathrm{D}^{\mathcal{I}}, \\ \mathcal{I} \models \mathrm{C}(\mathsf{a}) & \textit{if and only if} & \mathsf{a}^{\mathcal{I}} \in \mathrm{C}^{\mathcal{I}}, \\ \mathcal{I} \models r(\mathsf{a}, \mathsf{b}) & \textit{if and only if} & (\mathsf{a}^{\mathcal{I}}, \mathsf{b}^{\mathcal{I}}) \in r^{\mathcal{I}}, \\ \mathcal{I} \models \neg r(\mathsf{a}, \mathsf{b}) & \textit{if and only if} & (\mathsf{a}^{\mathcal{I}}, \mathsf{b}^{\mathcal{I}}) \notin r^{\mathcal{I}}. \end{array}$$

Structures are like interpretations, with the only exception that the assignment of individual names may be partial, that is some individual names from $\mathbf{N_I}$ may not be "used". An interpretation $\mathcal{I}$ *satisfies* a knowledge base $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ (or that $\mathcal{I}$ is a *model* of $\mathcal{K}$, written: $\mathcal{I} \models \mathcal{K}$) if it satisfies all axioms of $\mathcal{A} \cup \mathcal{T}$. An interpretation $\mathcal{I}$ is *finite* (resp. *countable*) if its domain $\Delta^{\mathcal{I}}$ is finite (resp. *countable*). A knowledge base is (finitely) *consistent* (or (finitely) *satisfiable*) if it has a (finite) model, and (finitely) *inconsistent* (or (finitely) *unsatisfiable*) otherwise.

Example 2.1. Consider an interpretation $\mathcal{I}$ with $\Delta^{\mathcal{I}} := \mathbb{Z}_8$, defined as follows and depicted below:

- $\mathsf{a}^{\mathcal{I}} := 4, \mathsf{b}^{\mathcal{I}} := 5$ and all other individual names from $\mathbf{N_I}$ are interpreted as 1.
- $\mathrm{G}^{\mathcal{I}} := \{0, 4, 7\}$, $\mathrm{R}^{\mathcal{I}} := \{1, 2, 3, 5, 6\}$, $\mathrm{B}^{\mathcal{I}} := \{2, 4\}$, and all other concepts names are interpreted as $\emptyset$,
- $r^{\mathcal{I}} := \{(0, 1), (1, 2), (1, 4), (6, 7)\}$, $s^{\mathcal{I}} := \{(0, 1), (1, 4), (4, 3), (4, 5), (5, 4), (7, 7)\}$, and all other role names from $\mathbf{N_R}$ are interpreted as $\emptyset$.



Consider a knowledge base $\mathcal{K} := (\{s(\mathsf{a}, \mathsf{b}), \neg r(\mathsf{b}, \mathsf{a}), \mathrm{B}(\mathsf{a})\}, \{\top \sqsubseteq \mathrm{R} \sqcup \exists s.\top, \mathrm{R} \equiv \neg\mathrm{G}\})$. It is easy to verify that $\mathcal{I}$ satisfies $\mathcal{K}$, and thus we can write $\mathcal{I} \models \mathcal{K}$. Note that $\mathcal{I}$ is a *finite* model of $\mathcal{K}$.

Given a structure $\mathcal{I}$ and a pair of elements $\mathrm{d}, \mathrm{e} \in \Delta^{\mathcal{I}}$, we employ $\mathsf{Conc}_{\mathcal{I}}(\mathrm{d}) \coloneqq \{\mathrm{C} \in \mathbf{N_C} \mid \mathrm{d} \in \mathrm{C}^{\mathcal{I}}\}$ to denote the set of all atomic concepts satisfied by d in $\mathcal{I}$, and use $\mathsf{Rol}_{\mathcal{I}}(\mathrm{d}, \mathrm{e}) \coloneqq \{r \in \mathbf{N_R} \mid (\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}\}$ to denote the set of all atomic roles satisfied by the pair (d, e) in $\mathcal{I}$. For a given set of individual names $\mathsf{N} \subseteq \mathbf{N_I}$ we use $\mathsf{N}^{\mathcal{I}}$ to denote the set of N-*named domain elements* of $\mathcal{I}$, namely the set of all $\mathrm{d} \in \Delta^{\mathcal{I}}$ for which $\mathrm{d} = \mathsf{a}^{\mathcal{I}}$ holds for some name $\mathsf{a} \in \mathsf{N}$. The elements from $\Delta^{\mathcal{I}} \setminus \mathsf{N}^{\mathcal{I}}$ are called N-*anonymous*. With $\mathsf{ind}(\mathcal{I}) \coloneqq \{\mathsf{a} \in \mathbf{N_I} \mid \mathsf{a}^{\mathcal{I}} \in \Delta^{\mathcal{I}}\}$ we collect all individual names whose interpretation appears in a structure $\mathcal{I}$.

Example 2.2. Consider $\mathcal{I}$ from Example 2.1. We have $\mathsf{Conc}_{\mathcal{I}}(4) = \{\mathrm{G}, \mathrm{B}\}$, $\mathsf{Rol}_{\mathcal{I}}(0, 1) = \{r, s\}$, while $\mathsf{Rol}_{\mathcal{I}}(1, 0) = \emptyset$. The element 4 is $\{\mathsf{a}, \mathsf{b}\}$-named, while the elements from $\{0, 1, 2, 3, 5, 6, 7\}$ are $\{\mathsf{a}\}$-anonymous.

All notions from this section can be easily lifted to any logic $\mathcal{DL}$ semantically **extending** $\mathcal{ALC}$. We say that a logic[1] $\mathcal{DL}$ **extends** a logic $\mathcal{DL}'$ if every $\mathcal{DL}'$-concept C in one can compute in polynomial time (at most) polynomially larger, logically equivalent, $\mathcal{DL}$-concept. For instance, $\mathcal{ALC}$ extends the modal logic K, $\mathcal{ALC}b$ extends $\mathcal{ALCH}$, $\mathcal{ALC}_{\mathsf{reg}}$ extends Propositional Dynamic Logic, and $\mathcal{ALCSCC}$ extends $\mathcal{ALCHQ}$.

2.2 A Tiny Bit of Graph Theory

We revisit the classical notions of substructures, paths and connectivity. Let $\mathcal{I}$ be an interpretation. The *restriction* of $\mathcal{I}$ to a set $\mathrm{S} \subseteq \Delta^{\mathcal{I}}$, is the structure $\mathcal{I}\restriction_{\mathrm{S}}$ defined according to the following conditions:

$$\Delta^{\mathcal{I}\restriction_{\mathrm{S}}} \coloneqq \mathrm{S},\ r^{\mathcal{I}\restriction_{\mathrm{S}}} \coloneqq r^{\mathcal{I}} \cap (\mathrm{S} \times \mathrm{S}),\ \mathrm{A}^{\mathcal{I}\restriction_{\mathrm{S}}} \coloneqq \mathrm{A}^{\mathcal{I}} \cap \mathrm{S},\ \mathsf{a}^{\mathcal{I}\restriction_{\mathrm{S}}} \coloneqq \mathsf{a}^{\mathcal{I}} \text{ if } \mathsf{a}^{\mathcal{I}} \in \mathrm{S} \text{ and is undefined otherwise,}$$

for all $\mathrm{A} \in \mathbf{N_C}$, $r \in \mathbf{N_R}$ and $\mathsf{a} \in \mathbf{N_I}$. A *substructure* of $\mathcal{I}$ is any of its restrictions $\mathcal{I}\restriction_{\mathrm{S}}$ for any $\mathrm{S} \subseteq \Delta^{\mathcal{I}}$.

An *undirected path* (resp. a *directed path*) of length $k{-}1$ in an interpretation $\mathcal{I}$ is a finite word $\rho \coloneqq \rho_1\rho_2 \ldots \rho_k$ composed of elements from $\Delta^{\mathcal{I}}$, such that for any index $1 \leq i < k$ we have that $(\rho_i, \rho_{i+1}) \in r^{\mathcal{I}}$ or $(\rho_{i+1}, \rho_i) \in r^{\mathcal{I}}$ for some role name $r \in \mathbf{N_R}$ (or just $(\rho_i, \rho_{i+1}) \in r^{\mathcal{I}}$ in the case of directed paths). We use $\|\rho\|$ to denote the length of ρ (note that $\|\rho\| = |\rho|{-}1$). An element $\mathrm{e} \in \Delta^{\mathcal{I}}$ is *reachable* from $\mathrm{d} \in \Delta^{\mathcal{I}}$ via an (un)directed path if there exists an (un)directed path ρ in $\mathcal{I}$ for which $\rho_1 = \mathrm{d}$ and $\rho_{|\rho|} = \mathrm{e}$. In this case we also say that ρ *starts* from d and *ends* in e. A *cycle* is a path from an element to itself. We say that $\mathcal{I}$ is *connected* if any of its domain elements are reachable from any other via an undirected path. A structure $\mathcal{J}$ is a *connected component* of $\mathcal{I}$ if it is a $\subseteq$-maximal (in the sense of inclusion of domains) connected substructure of $\mathcal{I}$. The k-*neighbourhood* of d in $\mathcal{I}$, denoted with $\mathsf{Nbd}^k_{\mathcal{I}}(\mathrm{d})$, is the restriction of $\mathcal{I}$ to elements reachable from d in $\mathcal{I}$ by *undirected* paths of length at most k.

Example 2.3. Consider the interpretation $\mathcal{I}$ from Example 2.1. It has two connected components, namely $\mathcal{I}\restriction_{\{0,1,2,3,4,5\}}$ and $\mathcal{I}\restriction_{\{6,7\}}$ (visualised as the letters "H" and "I"). The element 5 is reachable from 0 via a (directed) path $\rho \coloneqq (0, 1, 4, 5)$. Note that $\|\rho\| = 3$. Moreover, 4 reaches 2 via an undirected path. Finally, we have $\mathsf{Nbd}^0_{\mathcal{I}}(0) = \mathcal{I}\restriction_{\{0\}}$, $\mathsf{Nbd}^1_{\mathcal{I}}(0) = \mathcal{I}\restriction_{\{0,1\}}$, $\mathsf{Nbd}^2_{\mathcal{I}}(0) = \mathcal{I}\restriction_{\{0,1,2,4\}}$, and $\mathsf{Nbd}^3_{\mathcal{I}}(0) = \mathcal{I}\restriction_{\{0,1,2,3,4,5\}}$ for all $k \geq 3$.

2.3 Morphisms

Let $\mathcal{I}$ and $\mathcal{J}$ be structures, and let N be a subset of $\mathbf{N_I}$. An N-*homomorphism* $\mathfrak{f}\colon \mathcal{I} \to \mathcal{J}$ if a function that:

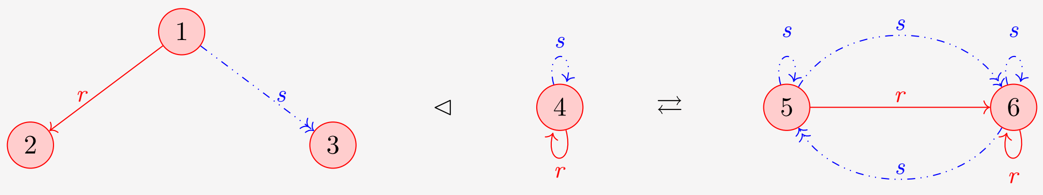
- maps elements from $\Delta^{\mathcal{I}}$ to $\Delta^{\mathcal{J}}$,
- preserves individual names from N, *i.e.* for all $\mathsf{a} \in \mathsf{N}$ if $\mathsf{a}^{\mathcal{I}}$ is defined then $\mathsf{a}^{\mathcal{J}} = \mathfrak{f}(\mathsf{a}^{\mathcal{I}})$,
- preserves atomic concepts, *i.e.* for all $\mathrm{d} \in \Delta^{\mathcal{I}}$ and all $\mathrm{A} \in \mathbf{N_C}$ we have that $\mathrm{d} \in \mathrm{A}^{\mathcal{I}}$ implies $\mathfrak{f}(\mathrm{d}) \in \mathrm{A}^{\mathcal{J}}$,
- preserves atomic roles, *i.e.* for all pairs $(\mathrm{d}, \mathrm{e}) \in (\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}})$ and all $r \in \mathbf{N_R}$ we have that $(\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}$ implies $(\mathfrak{f}(\mathrm{d}), \mathfrak{f}(\mathrm{e})) \in r^{\mathcal{J}}$.

An N-*isomorphism* $\mathfrak{f}\colon \mathcal{I} \to \mathcal{J}$ is a bijection such that $\mathfrak{f}$ and $\mathfrak{f}^{-1}$ are N-homomorphisms. We write $\mathcal{I} \lhd_{\mathsf{N}} \mathcal{J}$ to indicate the existence of an N-homomorphism from $\mathcal{I}$ to $\mathcal{J}$. In this case $\mathcal{I}$ is said to be N-homomorphically mapped to $\mathcal{J}$. Structures $\mathcal{I}$ and $\mathcal{J}$ are N-*homomorphically equivalent*, written: $\mathcal{I} \rightleftarrows_{\mathsf{N}} \mathcal{J}$,

[1]We have decided not to formally define what a *logic* is, suggesting that this notion should rather be understood naively. This can be made formal by means of abstract model theory, see the thesis of Piro [Pir12, Sec. 1.2].

if $\mathcal{I} \lhd_{\mathsf{N}} \mathcal{J}$ and $\mathcal{J} \lhd_{\mathsf{N}} \mathcal{I}$ hold. Finally, $\mathcal{I}$ and $\mathcal{J}$ are N-isomorphic, written: $\mathcal{I} \cong_{\mathsf{N}} \mathcal{J}$, if there exists an N-isomorphism between them. We often use the term homomorphism (resp. isomorphism) rather than $\emptyset$-homomorphism (resp. $\emptyset$-isomorphism). It is well-known that the composition of N-homomorphisms is also an N-homomorphism, and that the composition of N-isomorphisms is also an N-isomorphism.

Example 2.4. Consider structures $\mathcal{I}_1$, $\mathcal{I}_2$, $\mathcal{I}_3$ depicted below. Observe that there is a homomorphism from $\mathcal{I}_1$ to $\mathcal{I}_2$ but not vice versa, and that $\mathcal{I}_2$ and $\mathcal{I}_3$ are *homomorphically equivalent*, but not *isomorphic*.



We write $\mathfrak{f} \circ \mathfrak{g}$ to denote the composition of morphisms $\mathfrak{f} \colon \mathcal{I} \to \mathcal{I}'$ and $\mathfrak{g} \colon \mathcal{I}' \to \mathcal{I}''$ (*i.e.* we employ more "category-theory-friendly" syntax of composition, rather than the classical one).

2.4 Conjunctive Queries and Their Local Generalisations

Queries employ *variables* from a countably infinite set $\mathbf{N_V}$, which is disjoint from $\mathbf{N_I} \cup \mathbf{N_C} \cup \mathbf{N_R}$. A *conjunctive query* (CQ) is a conjunction of *atoms* of the form $r(x, y)$ or $\mathrm{A}(z)$, where r is a role name, A is a concept name and x, y, z are variables. More expressive query languages are also considered: a *union of conjunctive queries* (UCQ) is a disjunction of CQs and a *positive existential query* (PEQ) is a positive boolean combination of CQs (*i.e.* can be generated with the grammar: $q ::= \mathrm{A}(x) \mid r(x, y) \mid q \wedge q \mid q \vee q$). Any PEQ can be converted to a UCQ of (possibly) exponential size by turning it into disjunctive normal form. Let q be a PEQ and let $\mathcal{I}$ be a structure. The set of variables appearing in q is denoted with $\mathrm{Var}(q)$ and the number of atoms of q (*i.e.* the size of q) is denoted with $|q|$. The fact that $r(x, y)$ appears in q is indicated with $r(x, y) \in q$. Whenever some subset $\mathrm{V} \subseteq \mathrm{Var}(q)$ is given, let $q{\restriction}_{\mathrm{V}}$ denote the sub-query of q where all the atoms containing any variable outside V are removed (whenever it does not lead to ambiguity).

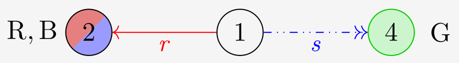
Example 2.5. Consider a PEQ $q := \mathrm{A}(x) \wedge (r(x, y) \vee (\mathrm{B}(x) \wedge s(x, z)))$. We have $|q| = 4$ and $\mathrm{Var}(q) = \{x, y, z\}$, and that $q{\restriction}_{\{x,y\}} = \mathrm{A}(x) \wedge (r(x, y) \vee \mathrm{B}(x))$ (*i.e.* the atom $s(x, z)$ got removed). Note that q is equivalent to a union of conjunctive queries $q_1 \vee q_2$, for CQs $q_1 := \mathrm{A}(x) \wedge \mathrm{B}(x) \wedge s(x, z)$, and $q_2 := \mathrm{A}(x) \wedge r(x, y)$.

Let $\pi \colon \mathrm{Var}(q) \to \Delta^{\mathcal{I}}$ be a *variable assignment.* We write $\mathcal{I} \models_{\pi} q$ whenever the boolean expression obtained from q by replacing each atom of the form $\mathrm{A}(x)$ (resp. $r(x, y)$) with the truth value of $\pi(x) \in \mathrm{A}^{\mathcal{I}}$ (resp. $(\pi(x), \pi(y)) \in r^{\mathcal{I}}$) evaluates to true. We say then that π is a *match* for q and $\mathcal{I}$. The interpretation $\mathcal{I}$ *satisfies* q (denoted with: $\mathcal{I} \models q$) whenever $\mathcal{I} \models_{\pi} q$ for some match π. The definitions are lifted to knowledge bases: q is *(finitely) entailed* by a knowledge base $\mathcal{K}$ (written: $\mathcal{K} \models_{(\mathrm{fin})} q$) if every (finite) model $\mathcal{I}$ of $\mathcal{K}$ satisfies q. The entailment relations $\models$ and $\models_{\mathrm{fin}}$ may not coincide, as witnessed by Example 2.6. When $\mathcal{I} \models \mathcal{K}$ but $\mathcal{I} \not\models q$, we call $\mathcal{I}$ a *countermodel* for $\mathcal{K}$ and q. Note that q is (finitely) entailed by $\mathcal{K}$ if there is no (finite) countermodel for $\mathcal{K}$ and q.

Example 2.6 (Example 3 from [GIJ18])**.** The description logic $\mathcal{S}$ extends $\mathcal{ALC}$ with axioms of the form $\mathsf{trans}(r)$ for role names $r \in \mathbf{N_R}$ that are satisfied by an interpretation $\mathcal{I}$ if and only if $r^{\mathcal{I}}$ is transitive. Let $\mathcal{K}$ be an $\mathcal{S}$-KB composed of the following axioms: (i) $\mathsf{trans}(r)$, and (ii) $\top \sqsubseteq \exists r.\top$. Note that axiom (ii) enforces the presence of an infinite r-path in every model of $\mathcal{K}$. Let $q := r(x, x)$ be a conjunctive query. Note that there are infinite models that satisfy $\mathcal{K}$ but violate q, for instance $\mathcal{I}$ with $\Delta^{\mathcal{I}} := \mathbb{N}$ that interprets r as the "less-or-equal relation $\leq$". However, by a combination of transitivity and finiteness, every finite model of $\mathcal{K}$ has an element decorated with an r-self-loop (and thus satisfies the query q). Hence $\mathcal{K} \not\models q$ but $\mathcal{K} \models_{\mathrm{fin}} q$.

Every conjunctive query q can be seen as a structure $\mathcal{I}_q := (\mathrm{Var}(q), \cdot^{\mathcal{I}_q})$ that interprets concept names $\mathrm{A} \in \mathbf{N_C}$ as $\mathrm{A}^{\mathcal{I}_q} := \{x \mid \mathrm{A}(x) \in q\}$, role names $r \in \mathbf{N_R}$ as $r^{\mathcal{I}_q} := \{(x, y) \mid r(x, y) \in q\}$, and does not interpret individual names. This implies that any match π for an interpretation $\mathcal{I}$ and a conjunctive query q can be seen as an $\mathbf{N_I}$-homomorphism (as well as $\emptyset$-homomorphism) from $\mathcal{I}_q$ to $\mathcal{I}$.

Example 2.7. Consider a CQ $q := \mathrm{R}(x_2) \land \mathrm{B}(x_2) \land r(x_1, x_2) \land s(x_1, x_4) \land \mathrm{G}(x_4)$ for which $\mathcal{I}_q$ can be visualised as:



Let $\mathcal{I}$ and $\mathcal{T}$ be the interpretation and the TBox from Example 2.1. Then $\mathcal{I} \models q$, which can be witnessed by the match $\pi\colon x_i \mapsto i$. However, we have $\mathcal{T} \not\models q$. This can be seen by taking an example model $\mathcal{J}$ of $\mathcal{T}$ composed of a single "red" element that interprets all role names as $\emptyset$. Clearly $\mathcal{J} \not\models q$, and thus $\mathcal{T} \not\models q$.

For a given CQ q, a structure $\mathcal{I}$, a match $\pi\colon \mathrm{Var}(q) \to \Delta^{\mathcal{I}}$, we define the equivalence relation $\approx_\pi$ as $\{(x, y) \in \mathrm{Var}(q) \times \mathrm{Var}(q) \mid \pi(x) = \pi(y)\}$. We use square brackets to denote equivalence classes, *e.g.* $[x]_{\approx_\pi}$ denotes the equivalence class of x w.r.t $\approx_\pi$. We define the *image of q via π* and denote it with $\pi[q]$, as a structure $\mathcal{J}$ with $\Delta^{\mathcal{J}} := \mathrm{Var}(q)/{\approx_\pi}$ that for all $\mathsf{a} \in \mathbf{N_I}, \mathrm{A} \in \mathbf{N_C}, r \in \mathbf{N_R}$ satisfies the conditions below:

- $\mathsf{a}^{\mathcal{J}}$ is defined only if there exists a variable x for which $\pi(x) = \mathsf{a}^{\mathcal{I}}$. In this case $\mathsf{a}^{\mathcal{J}} := [x]_{\approx_\pi}$.
- $\mathrm{d} \in \mathrm{A}^{\mathcal{J}}$ only if there exist a variable x such that $\mathrm{d} = [x]_{\approx_\pi}$, and $\mathrm{A}(x) \in q$.
- $(\mathrm{d}, \mathrm{e}) \in r^{\mathcal{J}}$ only if there exist variables x, y such that $(\mathrm{d}, \mathrm{e}) = ([x]_{\approx_\pi}, [y]_{\approx_\pi})$, and $r(x, y) \in q$.

Note that $\pi[q]$ usually differs from the substructure of $\mathcal{I}$ induced by $\pi[\mathrm{Var}(q)]$.

For the class of *path-shaped conjunctive queries*, namely the conjunctive queries whose query structure looks like a path, we often employ an alternative *path syntax* for conciseness. Thus, by a path-shaped conjunctive query we understand an expression of the form

$$(\mathrm{A}_0?; r_1; \mathrm{A}_1?; r_2; \mathrm{A}_2?; \ldots; \mathrm{A}_{n-1}?; r_n; \mathrm{A}_n?)(x_0, x_n)$$

with all $r_i \in \mathbf{N_R}$ and $\mathrm{A}_i \in \mathbf{N_C} \cup \{\top\}$, serving as a shorthand for

$$\bigwedge_{i=0}^{n} \mathrm{A}_i(x_i) \land \bigwedge_{i=1}^{n} r_i(x_{i-1}, x_i).$$

Whenever A_i happens to be $\top$, it will be removed from the expression; this does not create ambiguities. For instance, the query $r; \mathrm{A}?; r; s; \mathrm{B}?$ stands for $r(x_0, x_1) \land \mathrm{A}(x_1) \land r(x_1, x_2) \land s(x_2, x_3) \land \mathrm{A}(x_3)$. We stress that the alternative syntax for path-shaped CQs is just syntactic sugar and our queries should not be mistaken *e.g.* for regular path queries (RPQs).

We conclude by discussing the differences between our definitions of queries and the ones that are present in the literature. We first discuss the presence of answer variables and individual names in queries.

Remark 2.8. First, our queries are always assumed to be boolean, *i.e.* we do not allow for answer variables. This assumption is done [GLHS08, p. 164] w.l.o.g. as answer variables can be simulated with quantified variables and additional concept names. Second, individual names are not present in atoms in queries. This is again w.l.o.g. as one can proceed for any knowledge-base $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ and any PEQ q as follows. Take any individual name a appearing in query q and proceed as follows: (i) introduce a fresh variable x_{a} and fresh concept name A_{a}, (ii) replace each atom α in q involving a by $\alpha \land \mathrm{A}_{\mathsf{a}}(x_{\mathsf{a}})$, (iii) replace every occurrence of the individual name a in q by x_{a}, and (iv) append $\mathrm{A}_{\mathsf{a}}(\mathsf{a})$ to the ABox $\mathcal{A}$. Let q', $\mathcal{K}'$ be the resulting query and the resulting knowledge base. It is not too difficult to show that $\mathcal{K} \models_{(\mathrm{fin})} q$ if and only if $\mathcal{K}' \models_{(\mathrm{fin})} q'$.

The second remark is about the presence of "special" roles in queries.

Remark 2.9. We would like to stress that in the thesis we stick to the *usual definition* of conjunctive queries, meaning that we only allow for role names and concept names in query atoms. This is crucial for certain complexity results present in the literature. For instance, the query entailment problem (to be defined) for the description logic $\mathcal{S}$ is known to be coNExpTime-hard [ELOv09, Thm. 2], but it is decidable in ExpTime [Lut08b, Thm. 1] if role names that are stated to be transitive do not appear in queries.

2.5 Database-Inspired Decision Problems

We next recall important definitions of *database-inspired reasoning problems* considered in this thesis.

The first problem that we consider, called the (finite) *satisfiability problem* is parametrised by a logic $\mathcal{DL}$ and asks whether an input $\mathcal{DL}$-knowledge-base is (finitely) satisfiable. Here we mention a few results on $\mathcal{ALC}$ and related logics. It is well-known that $\mathcal{ALC}$ has the *finite model property* (FMP) [LAHS04, Cor. 4.3], *i.e.* every satisfiable $\mathcal{ALC}$-KB has a finite model. In other words, the satisfiability and the finite satisfiability problems coincide. This positive result on $\mathcal{ALC}$ can be further generalised to several very expressive description logics including $\mathcal{SHOQ}$ [LAHS04, Cor. 4.3], $\mathcal{ZOQ}$ and $\mathcal{ZOI}$ [BK22, Thm. 3.1], as well as to expressive decidable fragments of first-order logics, including the *guarded* [Grä99, Thm. 3.10], and *guarded-negation* [BtCS15, Thm. 3.4] fragments. Check Pratt-Hartmann's textbook [PH23] for more.

(Finite) $\mathcal{DL}$-KB satisfiability problem
Parameters: Description logic $\mathcal{DL}$
Input: A $\mathcal{DL}$-knowledge-base $\mathcal{K}$
Question: Does $\mathcal{K}$ have a (finite) model?

The second one, called the (finite) *query entailment problem*, is additionally parametrised by a query language $\mathscr{Q}$. It asks whether an input $\mathcal{DL}$-knowledge-base (finitely) entails an input query from $\mathscr{Q}$. The query entailment problem is at least as computationally difficult as the satisfiability problem (note that only an unsatisfiable knowledge base entails the $\bot$ query). The "query entailment analogue" of the finite model property is called *finite controllability.* The class of queries $\mathscr{Q}$ is finitely controllable for a description logic $\mathcal{DL}$ if for every $\mathcal{DL}$-KB $\mathcal{K}$ and every query $q \in \mathscr{Q}$ we have that $\mathcal{K} \models q$ if and only if $\mathcal{K} \models_{\text{fin}} q$. It is known that $\mathcal{ALC}$ (and its generalisation to the guarded fragment) is finitely controllable [BGO14, Thm. 1.2] for the class of positive existential queries. The same results hold also for the very expressive DLs $\mathcal{ZOI}$ and $\mathcal{ZOQ}$ [BK22, Thm. 3.1]. The notion of finite controllability for classes of queries beyond positive existential queries were recently studied by Figueira et. al [FFB20]. Regarding the complexity results, the satisfiability problem [DL96, Thm. 6] and the CQ-entailment problem for $\mathcal{ALC}$ [OvE08, Thm. 6] (and even $\mathcal{ALCHQ}$ [Lut08b, Thm. 1]) are ExpTime-complete, while the PEQ-entailment problem for $\mathcal{ALC}$ was shown to be 2ExpTime-hard [Ov14, Thm. 1]. The 2ExpTime upper bound can be obtained even for very expressive extensions of $\mathcal{ALC}$ and regular queries extending PEQs [CEO09, Thm. 5.23].

(Finite) entailment of $\mathscr{Q}$ queries over $\mathcal{DL}$-KB
Parameters: Description logic $\mathcal{DL}$, and a class of queries $\mathscr{Q}$
Input: A $\mathcal{DL}$-knowledge base $\mathcal{K}$, and a query $q \in \mathscr{Q}$
Question: Does $\mathcal{K} \models_{(\text{fin})} q$ hold?

In a typical setting of the query entailment problem, both a knowledge base and a query are treated equally when measuring the input. It turns out however, that in "practical applications" input ontology and input queries are relatively small, whilst an input database (ABox) can be huge. To take this "practical" view into account, we consider the *data complexity* [Var82] version of the query entailment problem, where both the TBox and the query are parameters of the problem (thus their sizes are treated as constants), and the input to the problem is just an ABox. In this setting the complexity of the query entailment problem drops drastically. For instance, the query entailment problem for $\mathcal{ALCI}$ is 2ExpTime-complete [Lut08a, Thm. 2], but just coNP-complete [GLHS08, Thm. 35] with respect to the data-complexity.

(Finite) entailment of $\mathscr{Q}$ queries over $\mathcal{DL}$-KB w.r.t Data Complexity
Parameters: Description logic $\mathcal{DL}$, a $\mathcal{DL}$-TBox $\mathcal{T}$, a class of queries $\mathscr{Q}$, and a query $q \in \mathscr{Q}$
Input: An ABox $\mathcal{A}$
Question: Does $(\mathcal{A}, \mathcal{T}) \models q$ hold?

The last problem that we consider is the classical *query containment problem* [CV93] tailored to the ontology-mediated setting [COv11, BLW12]. A relatively short definition is presented below.

(Finite) containment of $\mathcal{Q}$ queries over $\mathcal{DL}$-TBoxes

Parameters:	Description logic $\mathcal{DL}$, and a class of queries $\mathcal{Q}$
Input:	A $\mathcal{DL}$-TBox $\mathcal{T}$, and queries $q_1, q_2 \in \mathcal{Q}$
Question:	Does every (finite) model $\mathcal{I}$ of $\mathcal{T}$ that satisfies q_1 also satisfy q_2?

2.6 A Bit of Automata Theory

Let Σ be a finite alphabet (usually a finite subset of $\mathbf{N_R}$), and let ε denote the empty word. A *nondeterministic finite automaton* (NFA) $\mathcal{A}$ is tuple $(\Sigma, \mathtt{Q}, \mathtt{I}, \mathtt{F}, \mathtt{T})$, where $\mathtt{Q}$ is a finite set of *states*, $\mathtt{I} \subseteq \mathtt{Q}$ is a set of *initial states*, $\mathtt{F} \subseteq \mathtt{Q}$ is a set of *final states*, and $\mathtt{T}$ is a transition relation of type $\mathtt{T} \subseteq ((\mathtt{Q} \times \Sigma) \times \mathtt{Q})$. A deterministic finite automaton (DFA) is an NFA whose transition relation is a function. A *partial semiautomaton* (*PSA*) $\mathcal{A}$ [Gin68] is a nondeterministic finite automaton (NFA) that does not specify its initial and final states. For states $\mathtt{q}$ and $\mathtt{q}'$ of $\mathcal{A}$, $\mathcal{A}_{\mathtt{q},\mathtt{q}'}$ denotes the corresponding NFA with the initial (resp. final) state $\mathtt{q}$ (resp. $\mathtt{q}'$). Given a finite word $\mathtt{w} := \mathtt{a}_1 \ldots \mathtt{a}_n$ over Σ, a *run* of $\mathcal{A}$ on $\mathtt{w}$ is a sequence $\mathtt{q}_0 \rightarrow_{\mathtt{a}_1} \mathtt{q}_1 \rightarrow_{\mathtt{a}_2} \ldots \rightarrow_{\mathtt{a}_n} \mathtt{q}_n$ where $\mathtt{q}_0 \in \mathtt{I}$ and for all $i < n$ we have $(\mathtt{q}_i, \mathtt{a}_{i+1}, \mathtt{q}_{i+1}) \in \mathtt{T}$. A run is *accepting* if $\mathtt{q}_n \in \mathtt{F}$. We call a word *accepted* by $\mathcal{A}$ if there is an accepting run of $\mathcal{A}$ on it. A *language* is any set of finite words. The *language* $\mathcal{L}(\mathcal{A})$ of $\mathcal{A}$ is the set composed of all words accepted by $\mathcal{A}$. We say that a language $\mathcal{L} \subseteq \Sigma^*$ is *regular* if there exists an NFA *recognising it*, *i.e.* there exists an NFA $\mathcal{A}$ for which $\mathcal{L}(\mathcal{A}) = \mathcal{L}$. For more details on regular languages consult Sipser's textbook [Sip13, Sec. 1].

Regular expressions over an alphabet Σ are defined with the following grammar:

$$\mathcal{R}_1, \mathcal{R}_2 ::= \varepsilon \mid \mathtt{a} \mid \emptyset \mid (\mathcal{R}_1 + \mathcal{R}_2) \mid (\mathcal{R}_1 \circ \mathcal{R}_2) \mid (\mathcal{R}_1)^*,$$

for $\mathtt{a} \in \Sigma$. The *language* $\mathcal{L}(\mathcal{R})$ of a regular expression $\mathcal{R}$ is defined inductively: $\mathcal{L}(\mathtt{a}) := \{\mathtt{a}\}$, $\mathcal{L}(\varepsilon) := \{\varepsilon\}$, $\mathcal{L}(\emptyset) := \emptyset$, $\mathcal{L}(\mathcal{R}_1 + \mathcal{R}_2) := \mathcal{L}(\mathcal{R}_1) \cup \mathcal{L}(\mathcal{R}_2)$, $\mathcal{L}(\mathcal{R}_1 \circ \mathcal{R}_2) := \{\mathtt{wv} \mid \mathtt{w} \in \mathcal{L}(\mathcal{R}_1), \mathtt{v} \in \mathcal{L}(\mathcal{R}_2)\}$, $\mathcal{L}(\mathcal{R}_1^*) := \{\varepsilon\} \cup \mathcal{L}(\mathcal{R}_1^+)$, and $\mathcal{L}(\mathcal{R}_1^+) := \bigcup_{n=1}^{\infty} \mathcal{L}(\mathcal{R}_1^n)$. Here $\mathcal{R}^1 := \mathcal{R}$ and $\mathcal{R}^{n+1} := \mathcal{R}^n \circ \mathcal{R}$. The $\cdot^*$ and $\cdot^+$ operators are called, respectively, the *Kleene star* and *Kleene plus*. It is well-known that for every regular expression $\mathcal{R}$ there exists an NFA $\mathcal{A}$ of polynomial size (w.r.t. total number of symbols in $\mathcal{R}$) for which $\mathcal{L}(\mathcal{R}) = \mathcal{L}(\mathcal{A})$.

Example 2.10. Let $\mathcal{A} := (\{\mathtt{a}, \mathtt{b}\}, \{\mathtt{q}_0, \mathtt{q}_1\}, \{\mathtt{q}_0\}, \{\mathtt{q}_0\}, \{(\mathtt{q}_0, \mathtt{a}, \mathtt{q}_1), (\mathtt{q}_1, \mathtt{a}, \mathtt{q}_0), (\mathtt{q}_0, \mathtt{b}, \mathtt{q}_0), (\mathtt{q}_1, \mathtt{b}, \mathtt{q}_1)\})$ be a deterministic finite automaton. It is easy to verify that the language of $\mathcal{A}$ is composed of all words from $\{\mathtt{a}, \mathtt{b}\}^*$ that have an even number of "a". Moreover, $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathtt{b}^*(\mathtt{a}\mathtt{b}^*\mathtt{a}\mathtt{b}^*)^*)$.

Let $\mathcal{A} := (\Sigma, \mathtt{Q}, \mathtt{I}, \mathtt{F}, \mathtt{T})$ be an NFA. For $\mathtt{q}, \mathtt{q}' \in \mathtt{Q}$ we use $\mathcal{A}_{\mathtt{q},\mathtt{q}'}$ to denote the automaton $(\Sigma, \mathtt{Q}, \{\mathtt{q}\}, \{\mathtt{q}'\}, \mathtt{T})$, *i.e.* the automaton obtained from $\mathcal{A}$ by setting the initial state to $\mathtt{q}$ and the final state to $\mathtt{q}'$. Analogously, $\mathcal{A}_{\mathtt{q}}$ denotes the automaton $(\Sigma, \mathtt{Q}, \{\mathtt{q}\}, \mathtt{F}, \mathtt{T})$, *i.e.* $\mathcal{A}$ with the initial state switched to $\mathtt{q}$. For future purposes we also introduce the automaton $\mathcal{A}^- := (\Sigma^-, \mathtt{Q}, \mathtt{F}, \mathtt{I}, \mathtt{T}^-)$, where Σ^- is a fresh alphabet composed of "inverted" symbols $\mathtt{a}^-$ for $\mathtt{a}$ in Σ, with initial and final states swapped and the state transitions flipped: the transition $(\mathtt{q}', \mathtt{a}^-, \mathtt{q})$ belongs to $\mathtt{T}^-$ if and only if $(\mathtt{q}, \mathtt{a}, \mathtt{q}') \in \mathtt{T}$. We call $\mathcal{A}^-$ the *reverse automaton of* $\mathcal{A}$. Note that the language of the reverse automaton $\mathcal{A}^-$ is the reverse of the language of $\mathcal{A}$ in the following sense: a word $(\mathtt{a}_1\mathtt{a}_2 \ldots \mathtt{a}_n)$ is in $\mathcal{L}(\mathcal{A})$ if and only if the word $(\mathtt{a}_n^-\mathtt{a}_{n-1}^- \ldots \mathtt{a}_1^-)$ belongs to $\mathcal{L}(\mathcal{A}^-)$.

2.7 Description Logics with Path Expressions

We treat the set $\Sigma_{\mathsf{all}} := \mathbf{N_R} \cup \{C? \mid C \in \mathbf{N_C}\}$ as an infinite alphabet. Let $\mathbb{ALL}$ and $\mathbb{REG}$ be the classes of all Turing-recognizable (resp. all regular) languages of finite words over any finite subset of Σ_{all}. When reasoning in the $\mathcal{Z}$ family of DLs, we also employ *simple roles* from in place of role names (see Section 2.9).

Given a language $\mathcal{L} \in \mathbb{ALL}$ we say that a path ρ *realises* $\mathcal{L}$ or that ρ *is an* $\mathcal{L}$*-path* (both denoted with $\rho \models \mathcal{L}$) if $\mathcal{L}$ contains a word $w_1 r_1 \ldots w_{|\rho|-1} r_{|\rho|-1} w_{|\rho|}$, where all r_i is a (simple) roles and all w_i are (possibly empty) sequences of tests, satisfying $(\rho_i, \rho_{i+1}) \in r_i^{\mathcal{I}}$ and $\rho_i \in \mathrm{C}^{\mathcal{I}}$ for all $i \leq |\rho|$ and tests C? in w_i. The above-defined notions are lifted (in an obvious way) to other objects "recognizing" languages, such as various notions of automata, regular expressions, Turing machines, and so on. For instance, if $\mathcal{L}$ is recognizable by an automaton $\mathcal{A}$, we may speak about $\mathcal{A}$-paths. For convenience, we will also say that $\mathrm{e} \in \Delta^{\mathcal{I}}$ is *$\mathcal{L}$-reachable from* $\mathrm{d} \in \Delta^{\mathcal{I}}$ (or alternatively that the element d *$\mathcal{L}$-reaches* e) whenever there exists an $\mathcal{L}$-path ρ that starts from d and ends in e.

The logic $\mathcal{ALC}_{\mathsf{all}}$ extends $\mathcal{ALC}$ with concept constructors of the form $\exists \mathcal{M}_{\mathcal{L}}.\mathrm{C}$, where $\mathcal{L} \in \mathbb{ALL}$, C is an $\mathcal{ALC}_{\mathsf{all}}$-concept, and $\mathcal{M}_{\mathcal{L}}$ is any Turing machine [Sip13, Sec. 3.1] recognizing $\mathcal{L}$. Their semantics is as follows: $(\exists \mathcal{M}_{\mathcal{L}}.\mathrm{C})^{\mathcal{I}}$ is the set of all $\mathrm{d} \in \Delta^{\mathcal{I}}$ that can $\mathcal{L}$-reach some $\mathrm{e} \in \mathrm{C}^{\mathcal{I}}$, and $\forall \mathcal{M}_{\mathcal{L}}.\mathrm{C}$ stands for $\neg \exists \mathcal{M}_{\mathcal{L}}.\neg \mathrm{C}$. The logic $\mathcal{ALC}_{\mathsf{reg}}$ is a restriction of $\mathcal{ALC}_{\mathsf{all}}$ in which languages existential restrictions are given by NFA (or by regular expressions for convenience). The logic $\mathcal{ALC}_{\mathsf{reg}}$ is a notational variant of the well-known Propositional Dynamic Logic [FL79], popular in the community of formal verification.

2.8 Conjunctive Regular Path Queries and Beyond

Given a class $\mathbb{C}$ of languages that is a subclass of $\mathbb{ALL}$, the class of *$\mathbb{C}$-enriched Positive Existential Queries* (abbreviated as $\mathbb{C}$-PEQs) is defined with the following syntax:

$$q, q' ::= \bot \mid \mathrm{A}(x) \mid r(x,y) \mid \mathcal{L}(x,y) \mid q \vee q' \mid q \wedge q',$$

where $\mathrm{A} \in \mathbf{N_C}$, $r \in \mathbf{N_R}$, $\mathcal{L} \in \mathbb{C}$, and x, y are variables from a countably-infinite set $\mathbf{N_V}$. Their semantics is defined as an expected generalisation of the semantics for PEQs: $\mathcal{L}(x,y)$ evaluates to true under a variable assignment $\pi \colon \mathbf{N_V} \to \Delta^{\mathcal{I}}$ if and only if $\pi(x)$ can $\mathcal{L}$-reach $\pi(y)$ in $\mathcal{I}$. In total analogy to Section 2.4, we define $\mathbb{C}$-*CQs* as disjunction-free $\mathbb{C}$-PEQs, and $\mathbb{C}$-*UCQs* as disjunctions of $\mathbb{C}$-CQs (or, in other words, $\mathbb{C}$-PEQs in which disjunction is allowed only at the outermost level). We identify $\emptyset$-PEQs with PEQs.

We are particularly interested in the class of $\mathbb{REG}$-PEQs and $\mathbb{REG}$-CQs, more commonly known as (Positive) Conjunctive Regular Path Queries [FLS98] (short: (P)CRPQ). In most of the cases we assume that the regular languages appearing in PRPQs are presented as regular expressions. We also consider a (even more expressive) class of two-way CRPQs (denoted C2RPQ) that is defined in total analogy to CRPQ but with the notion of regular languages over finite subsets of $\mathbf{N_R}$ replaced by regular languages over extended alphabets being finite subsets of $\Sigma_{\mathsf{all}} \cup \{r^- \mid r \in \mathbf{N_R}\}$. The semantics of the key components, namely $\mathcal{L}$-paths from Section 2.7, is defined in almost the same way: the only difference is that we interpret the "inverted" role names r^- as (relational) inverses of roles (see Section 2.9.2 for a related feature).

Example 2.11. Suppose that a genealogical tree employs a *hasParent* role. The query $q \coloneqq \mathit{hasParent}(z,x) \wedge \mathit{hasParent}(z,y) \wedge (\mathit{hasParent}^* \circ (\mathit{hasParent}^-)^*)(x,y)$ is then satisfied by any variable assignment π mapping x,y,z to triples for which $\pi(x)$ and $\pi(y)$ are (possibly distant) relatives having a common child $\pi(z)$.

For more examples of CRPQs employed in the description-logic-based setting, consult the paper by Bienvenu et. al [BOv15, Sec. 3], or the survey by Bienvenu and Ortiz [BO15, Sec. 6.1]. Conjunctive regular path queries have also real-life applications as they serve as building blocks for querying graphs-structured databases [Bar13, AAB+17], including SPARQL [HS13], the W3C standard for querying RDF data, as well as G-Core [AAB+18], Cypher [FGG+18], and the ongoing standardisation project of GQL [FGG+23].

2.9 Primitive Extensions of Description Logics

We recall various DL modelling features present in the literature with their definitions.

2.9.1 Role Inclusions and Safe Boolean Combinations of Roles

Role hierarchies, denoted with $(\mathcal{H})$, allow for specifying inclusions between *atomic* roles by means of an extra axiom type of the shape $r \subseteq s$ for *role names* $r, s \in \mathbf{N_R}$. Formally $\mathcal{I} \models (r \subseteq s)$ if and only if $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$. *Safe boolean role combinations* [RKH08] (b) introduce a notion of a *simple role*, that is defined inductively as follows: (i) every role name is simple, (ii) inverted role is simple if the logic admits $(\mathcal{I})$, and (iii) if r, s

are simple then so are $r \cap s$, $r \cup s$, and $r \setminus s$. The semantics of simple roles follows by the usual set-theoretic semantics of operations $\cup, \cap, \setminus$. Simple roles can then be employed in existential and universal restrictions (and in number restrictions if the logic admits them), replacing the usual notion of roles.

Example 2.12. Role hierarchies can express that a *hasMother* role is a special case of *hasParent* role via *hasMother* $\subseteq$ *hasParent*. With (b) available, one can also use the role union *hasMother*$\cup$*hasFather* in place of *hasParent* role (*e.g.* to avoid redundancy in data). Note that a concept $(\exists(s \setminus r).\top) \sqsubseteq \bot$ is equivalent to a role inclusion $(r \subseteq s)$. Thus (b) subsumes $(\mathcal{H})$. Moreover, disjointness of roles can be expressed via $\top \sqsubseteq \forall(r \cap s).\bot$.

2.9.2 Self-loops, Nominals, and Inverses

The feature $(\mathcal{I})$ (*inverses of roles*) introduces, per each role name $r \in \mathbf{N_R}$, a fresh role name r^- interpreted by $\mathcal{I}$ via $(r^-)^\mathcal{I} := \{(\mathrm{e}, \mathrm{d}) \mid (\mathrm{d}, \mathrm{e}) \in r^\mathcal{I}\}$. The feature $(\mathcal{O})$ (*nominals*) introduces, per each individual name $\mathsf{a} \in \mathbf{N_I}$, a fresh concept $\{\mathsf{a}\}$, interpreted by $\mathcal{I}$ via $\{\mathsf{a}\}^\mathcal{I} := \{\mathsf{a}^\mathcal{I}\}$. The feature (Self) introduces concepts of the form $\exists r.\mathsf{Self}$ for role names $r \in \mathbf{N_R}$, that are interpreted by $\mathcal{I}$ via $(\exists r.\mathsf{Self})^\mathcal{I} := \{\mathrm{d} \in \Delta^\mathcal{I} \mid (\mathrm{d}, \mathrm{d}) \in r^\mathcal{I}\}$.

Example 2.13. A combination of self-loops and nominals can express that Gilderoy Lockhart is a narcissist via a GCI $\{\mathtt{GilderoyLockhart}\} \sqsubseteq \exists \mathit{loves}.\mathsf{Self}$. In the presence of $(\mathcal{I})$ and the role *isParent*, the role *isChild* can be eliminated and replaced with *isParent*$^-$.

2.9.3 Counting

Qualified number restrictions $(\mathcal{Q})$ extend the definition of concepts with constructs of the form $(\geq n\ r).\mathrm{C}$, where $n \in \mathbb{N}$ is a number, $r \in \mathbf{N_R}$ is a role name, and C is a concept. They are interpreted by $\mathcal{I}$ via

$$\mathrm{d} \in ((\geq n\ r).\mathrm{C})^\mathcal{I} \text{ \textit{if and only if} } |\{\mathrm{e} \in \Delta^\mathcal{I} : (\mathrm{d}, \mathrm{e}) \in r^\mathcal{I} \text{ and } \mathrm{e} \in \mathrm{C}^\mathcal{I}\}| \geq n.$$

Constructs for other comparison symbols, namely $<$, $\leq$, $>$, and $=$ are defined analogously. In unqualified number restrictions $(\mathcal{N})$, the concept C in the expression $(\geq n\ r).\mathrm{C}$ must be $\top$ (and is usually omitted). *Functionality* $(\mathcal{F})$ allows us for specifying, by means of a new axiom type of the form $\mathsf{func}(r)$, that a given role name r must be interpreted as a functional relation, *i.e.* whenever $(\mathrm{d}, \mathrm{e}) \in r^\mathcal{I}$ and $(\mathrm{d}, \mathrm{e}') \in r^\mathcal{I}$ holds in an interpretation $\mathcal{I}$ satisfying $\mathsf{func}(r)$, then $\mathrm{e} = \mathrm{e}'$ must hold. Note that $(\mathcal{Q})$ can express $(\mathcal{N})$, and that $(\mathcal{N})$ can express $(\mathcal{F})$. If inverses of roles $(\mathcal{I})$ are allowed in the logic, the inverted roles are allowed to appear in number restrictions and functionality statements. We always assume that number values appearing in number restrictions are encoded in *binary*, implying that storing such a number requires only logarithmically many bits rather than linearly many. This may influence the complexity of the logic: a good example is modal logic with global counting operators that is ExpTime-complete under unary encoding of numbers [AHD10, Thm. 4.2], but NExpTime-complete otherwise [ZST13, Thm. 5].

Example 2.14. Using number restrictions we can specify that He-Who-Must-Not-Be-Named created 7 horcruxes, *i.e.* in interpretation $\mathcal{I}$ the element $\mathtt{Voldemort}^\mathcal{I}$ should belong to the concept $((=7\ \mathit{created}).\mathrm{Horcrux})^\mathcal{I}$.

2.9.4 Presburger Counting $(\mathcal{SCC})$

Counting features from the previous section can be generalised even further by employing the quantifier-free fragment of Boolean Algebra with Presburger Arithmetic (abbreviated as QFBAPA). For presentation, we closely follow Baader et al. [BBR20, Sec. 2]. See also the recording of our ECAI'20 talk [Bed20].

We start with an introduction of the logic QFBAPA. The basic building blocks of QFBAPA are *set terms*. They are defined inductively, starting from *set variables*, and the set constants $\emptyset$ and $\mathcal{U}$. More complex set terms are obtained by the application of boolean operators (intersection $\cap$, union $\cup$, and complement $\cdot^c$) on other set terms. Set terms s, t can be used to state *equality* and *inclusion constraints*, having the form $\mathrm{s} = \mathrm{t}$ and $\mathrm{s} \subseteq \mathrm{t}$. *Presburger Arithmetic (PA) expressions* are built from integer constants and set cardinalities $|\mathrm{s}|$ using addition as well as multiplication with an integer constant. *Cardinality constraints* are of the form $k = \ell, k < \ell, \mathrm{N}\ \mathit{dvd}\ \ell$, where k, ℓ are PA expressions, N is an integer constant, and *dvd* stands for division. A *QFBAPA formula* is a boolean combination of set and cardinality constraints.

Example 2.15. Let S, R, and T be set variables. Then $\emptyset \cup (S^c \cap (T \cup R)^c) \subseteq R^c \cap (S \cup \mathcal{U})$ is an example set constraint, while $5 \, dvd \, 2 \cdot |T^c \cup S| + (-3) \cdot |\mathcal{U} \cap S|$ is an example PA expression.

A *substitution* σ assigns a finite set $\sigma(\mathcal{U})$ to $\mathcal{U}$, the empty set to $\emptyset$, and subsets of $\sigma(\mathcal{U})$ to set variables. It is extended to set terms by interpreting the boolean operations $\cap$, $\cup$, and $\cdot^c$ as set intersection, set union, and set complement w.r.t. $\sigma(\mathcal{U})$, respectively. The substitution σ satisfies the set constraint $\mathrm{s} = \mathrm{t}$ (resp. $\mathrm{s} \subseteq \mathrm{t}$) if $\sigma(\mathrm{s}) = \sigma(\mathrm{t})$ (resp. $\sigma(\mathrm{s}) \subseteq \sigma(\mathrm{t})$). It is further extended to a mapping from PA expressions to integers by interpreting $|\mathrm{s}|$ as the cardinality of the finite set $\sigma(\mathrm{s})$, and interpreting addition and multiplication with an integer constant in the usual way. The substitution σ satisfies the cardinality constraint $k = \ell$ if $\sigma(k) = \sigma(\ell)$, $k < \ell$ if $\sigma(k) < \sigma(\ell)$, and $\mathrm{N} \, dvd \, \ell$ if the integer constant N is a divisor of $\sigma(\ell)$. The notion of satisfaction of a boolean combination of set and cardinality constraints is now defined in an obvious way by interpreting $\wedge, \vee, \neg$ as in propositional logic. The substitution σ is a *solution* of the QFBAPA formula φ if it satisfies φ in this sense. A QFBAPA formula φ is *satisfiable* if it has a solution. Kunčak and Rinard [KR07, p. 222] proved NP-completeness of the satisfiability problem for QFBAPA.

We are finally ready to introduce the DL feature called $(\mathcal{SCC})$. It enlarges the set of concept constructors of the underlying logic with concepts of the form $succ(\alpha)$, where α is either a set constraint or a cardinality constraint that employ role names and already defined concepts in place of set variables. Their formal semantics is presented next with the proviso that we consider only *finitely branching interpretations*, *i.e.* interpretations $\mathcal{I}$ such that for every domain element $\mathrm{d} \in \Delta^{\mathcal{I}}$ the set $\bigcup_{r \in \mathbf{N_R}} \{\mathrm{e} \mid (\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}\}$ is finite. This assumption may be questionable; consider the paper by Baader and De Bortoli [BD19, p. 206] for a discussion of other semantics of QFBAPA. For a given element $\mathrm{d} \in \Delta^{\mathcal{I}}$ the substitution $\sigma^{\mathcal{I}}_{\mathrm{d}}$ assigns the *finite* set $\bigcup_{r \in \mathbf{N_R}} \{\mathrm{e} \mid (\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}\}$ to $\mathcal{U}$, the empty set to $\emptyset$, and the sets $\{\mathrm{e} \mid (\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}\}$ to r and $\mathrm{A}^{\mathcal{I}} \cap \bigcup_{r \in \mathbf{N_R}} \{\mathrm{e} \mid (\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}\}$ to A, where $r \in \mathbf{N_R}$ and $\mathrm{A} \in \mathbf{N_C}$ are viewed as set variables. The interpretation function $\cdot^{\mathcal{I}}$ and the substitutions $\sigma^{\mathcal{I}}_{\mathrm{d}}$ for $\mathrm{d} \in \Delta^{\mathcal{I}}$ are inductively extended to concepts by interpreting the boolean operators $\sqcap, \sqcup, \neg$ in the usual way and the successor expressions *succ* as follows:

$$\begin{aligned} succ(\alpha)^{\mathcal{I}} &:= \{\mathrm{d} \in \Delta^{\mathcal{I}} \mid \text{the substitution } \sigma^{\mathcal{I}}_{\mathrm{d}} \text{ satisfies } \alpha\}, \\ \sigma^{\mathcal{I}}_{\mathrm{d}}(succ(\alpha)) &:= succ(\alpha)^{\mathcal{I}} \cap \textstyle\bigcup_{r \in \mathbf{N_R}} \{\mathrm{e} \mid (\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}\}. \end{aligned}$$

Example 2.16. Baader [Baa17, Lem. 1] proved the equality $((\geq n \; r).\mathrm{C})^{\mathcal{I}} = succ(|\mathrm{C} \cap r| \geq n)^{\mathcal{I}}$ for any finitely-branching interpretation $\mathcal{I}$, concept C, and a role name r. Thus $(\mathcal{SCC})$ subsumes $(\mathcal{Q})$. Baader [Baa17, p. 49] provides an example that $\mathcal{ALCSCC}$ can define employers that employ more no-relatives than relatives with $\mathrm{Employer} \sqcap succ\,(2 \cdot |related \cap employs| < employs)$, or that creatures have even number of legs via $\mathrm{Creature} \sqcap succ\,(2 \, dvd \, |hasLimb \cap \mathrm{Leg}|)$, without the need of specifying how many legs a respective creature actually has.

2.9.5 Transitivity and Composition

Transitivity $(\mathcal{S})$ allows us for specifying, by means of a new axiom type of the form $\mathsf{trans}(r)$, that a given role name r is interpreted as a transitive relation. The feature $(\mathcal{R})$ allows us for specifying *chains of complex role inclusions* [HKS06, p. 2–3] that generalise transitivity. We omit the definition as we will never use it in the thesis. We usually assume that if a role name r occurs in a transitivity statement $\mathsf{trans}(r)$ or on the right hand side of a complex role inclusion, it is disallowed from number restrictions. This is often required to guarantee decidability. There are exceptions however [GGBIG+19, BKW21, GIJM23]. The feature $(\cdot_{\mathsf{reg}})$ allowing for regular expressions in existential restrictions was given in Section 2.7.

2.9.6 Fixed-Points

The next feature (μ) is quite technical. It extends the underlying description logic $\mathcal{DL}$ with fixed-points. For its definition we closely follow the description of $\mu\mathcal{ALCQ}$ by De Giacomo and Lenzerini [DL97, Sec. 4].

Let $\mathbf{N_F}$ be a countably-infinite set of *fixed-point variables*, that is pairwise disjoint from $\mathbf{N_C}$, $\mathbf{N_R}$, and $\mathbf{N_V}$. The logic $\mu\mathcal{DL}$ extends the set of concepts constructors of a logic $\mathcal{DL}$ with the use of variables X from $\mathbf{N_F}$ (treated as atomic concepts), and two new "quantified expressions" (called *fixed-point operators*)

μX.C and νX.C, where C is a concept, with the restriction that only a variable X occurring positively in C can be bound by a fixpoint μ/ν in μX.C and νX.C. By *positive* we mean that every free occurrence of a variable X is under an even number of negations. A valuation η on an interpretation $\mathcal{I}$ is a mapping that assigns variables from $\mathbf{N_F}$ to subsets of $\Delta^{\mathcal{I}}$. For a given valuation η, we use $\eta[\mathrm{X}/\mathrm{E}]$ to denote the valuation identical to η with the exception of $\eta[\mathrm{X}/\mathrm{E}](\mathrm{X}) := \mathrm{E}$.

Take an interpretation $\mathcal{I}$ and a valuation η. We define the semantics of concepts by associating to $\mathcal{I}$ and η an extension function $\cdot^{\mathcal{I}}_{\eta}$ mapping concepts to subsets of $\Delta^{\mathcal{I}}$ as follows:

$$\begin{aligned} \mathrm{X}^{\mathcal{I}}_{\eta} &:= \eta(\mathrm{X}) \text{ for all variables X}, \\ (\mu\mathrm{X.C})^{\mathcal{I}}_{\eta} &:= \bigcap \left\{ \mathrm{E} \subseteq \Delta^{\mathcal{I}} \mid \mathrm{C}^{\mathcal{I}}_{\eta[\mathrm{X}/\mathrm{E}]} \subseteq \mathrm{E} \right\}, \\ (\nu\mathrm{X.C})^{\mathcal{I}}_{\eta} &:= \bigcup \left\{ \mathrm{E} \subseteq \Delta^{\mathcal{I}} \mid \mathrm{E} \subseteq \mathrm{C}^{\mathcal{I}}_{\eta[\mathrm{X}/\mathrm{E}]} \right\}, \end{aligned}$$

and with all concepts without variables and fixed-point operators interpreted as usual. The notion of GCIs $\mathrm{C} \sqsubseteq \mathrm{D}$ (where both C and D do not contain free fixed-point variables) is lifted to the case with valuation function in a natural way, by requiring that $\mathcal{I} \models \mathrm{C} \sqsubseteq \mathrm{D}$ if and only if for all valuation functions η we have $(\mathcal{I}, \eta) \models \mathrm{C}^{\mathcal{I}}_{\eta} \subseteq \mathrm{D}^{\mathcal{I}}_{\eta}$. The notion of knowledge bases and their satisfaction is defined analogously. For more intuitions and results on fixed-points and (extensions of) $\mu\mathcal{ALC}$ (more commonly known as *μ-calculus*) we highly recommend reading the excellent survey by Bradfield and Walukiewicz [BW18].

Example 2.17. In functional programming we usually define *lists* as inductive datatypes such that: (i) the *empty list* is a list, (ii) a *node* that has exactly one successor that is a list is also a list, and (iii) nothing else is a list. Let EmptyList and Node be concept names that are interpreted as disjoint concepts, and let *succ* be a role name. Then, the following $\mu\mathcal{ALCQ}$-GCI defines a list [DL97, p. 93–95]:

$$\mathrm{List} \equiv \mu\mathrm{X}.\big(\mathrm{EmptyList} \sqcup [\mathrm{Node} \sqcap ({=}1\ succ).\top \sqcap \exists succ.\mathrm{X}]\big).$$

2.10 The Very Expressive Description Logic $\mathcal{ZOIQ}$

The description logic $\mathcal{ALCH}b^{\mathsf{Self}}_{\mathsf{reg}}\mathcal{IOQ}$ (abbreviated as $\mathcal{ZOIQ}$) is a very expressive logic whose decidability status up to this day remains open[2] [Rud16, p. 255]. Its three maximal subfragments, namely $\mathcal{ZIQ}$, $\mathcal{ZOI}$, and $\mathcal{ZOQ}$ are decidable though, and only ExpTime-complete [CEO09, Thm. 3.11]. Some progress towards solving the satisfiability problem of $\mathcal{ZOIQ}$ was done by Jung et al. [JLZ20, Thm. 14&Thm. 18].

We start by presenting a grammar that defines *atomic concepts* B, *concepts* C, *atomic roles* r, *simple roles* s and *roles* t for $\mathcal{ZOIQ}$, where o is an individual name, A is a concept name, and p is a role name:

$$\begin{aligned} \mathrm{B} &::= \mathrm{A} \mid \{\mathsf{o}\} \mid \top \mid \bot \\ \mathrm{C} &::= \mathrm{B} \mid \neg\mathrm{C} \mid \mathrm{C} \sqcap \mathrm{C} \mid \mathrm{C} \sqcup \mathrm{C} \mid \forall t.\mathrm{C} \mid \exists t.\mathrm{C} \mid \forall\overline{\top}.\mathrm{C} \mid \exists\overline{\top}.\mathrm{C} \mid ({\geqslant}n\ s).\mathrm{C} \mid ({\leqslant}n\ s).\mathrm{C} \mid \exists s.\mathsf{Self} \\ r &::= p \mid p^{-} \\ s &::= r \mid s \cap s \mid s \cup s \mid s \setminus s \\ t &::= s \mid t + t \mid t \circ t \mid t^{*} \mid \mathrm{C}? \end{aligned}$$

The usual definitions regarding interpretations, satisfaction, knowledge bases and so on are lifted from the case of $\mathcal{ALC}$ in a straightforward way (consult Section 2.1 if needed). For completeness and convenience, semantics of concepts and roles for $\mathcal{ZOIQ}$ is summarised by Table 2.1. We define $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, and $\mathcal{ZOI}$ by dropping from the syntax of $\mathcal{ZOIQ}$, respectively, nominals, role inverses, and number restrictions. We stress that in the case of $\mathcal{ZOQ}$, role inverses are also forbidden in the corresponding notion of simple roles.

[2] $\mathcal{ZOIQ}$ is wrongly claimed to be undecidable by Jung, Lutz, and Zeume [JLZ20] in the last 7 lines of their KR 2020 paper. Unfortunately as confirmed by our personal communication with Jean Jung, their proof does not work.

Table 2.1: Concepts and roles in $\mathcal{ZOIQ}$.

Name	Syntax	Semantics
bottom / top	$\bot$ / $\top$	$\emptyset$ / $\Delta^{\mathcal{I}}$
nominal	$\{\mathsf{o}\}$	$\{\mathsf{o}^{\mathcal{I}}\}$
concept negation	$\neg\mathrm{C}$	$\Delta^{\mathcal{I}} \setminus \mathrm{C}^{\mathcal{I}}$
concept intersection	$\mathrm{C}_1 \sqcap \mathrm{C}_2$	$\mathrm{C}_1^{\mathcal{I}} \cap \mathrm{C}_2^{\mathcal{I}}$
concept union	$\mathrm{C}_1 \sqcup \mathrm{C}_2$	$\mathrm{C}_1^{\mathcal{I}} \cup \mathrm{C}_2^{\mathcal{I}}$
existential restriction	$\exists t.\mathrm{C}$	$\{\mathrm{d} \mid \exists \mathrm{e} \in \mathrm{C}^{\mathcal{I}}.\ (\mathrm{d},\mathrm{e}) \in t^{\mathcal{I}}\}$
universal restriction	$\forall t.\mathrm{C}$	$\{\mathrm{d} \in \Delta^{\mathcal{I}}\ :\ \forall \mathrm{e} \in \Delta^{\mathcal{I}}.\ (\mathrm{d},\mathrm{e}) \in t^{\mathcal{I}} \rightarrow \mathrm{e} \in \mathrm{C}^{\mathcal{I}}\}$
qualified number restriction	$(\leqslant n\ s).\mathrm{C}$	$\{\mathrm{d} \in \Delta^{\mathcal{I}}\ :\ \lvert\{\mathrm{e} \in \mathrm{C}^{\mathcal{I}} : (\mathrm{d},\mathrm{e}) \in s^{\mathcal{I}}\}\rvert \leq n\}$
qualified number restriction	$(\geqslant n\ s).\mathrm{C}$	$\{\mathrm{d} \in \Delta^{\mathcal{I}}\ :\ \lvert\{\mathrm{e} \in \mathrm{C}^{\mathcal{I}} : (\mathrm{d},\mathrm{e}) \in s^{\mathcal{I}}\}\rvert \geq n\}$
Self concept	$\exists s.\mathsf{Self}$	$\{\mathrm{d} \mid\ (\mathrm{d},\mathrm{d}) \in s^{\mathcal{I}}\}$
universal role	$\overline{\top}$	$\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
inverse role	p^{-}	$\{(\mathrm{e},\mathrm{d}) \mid (\mathrm{d},\mathrm{e}) \in p^{\mathcal{I}}\}$
role intersection	$s_1 \cap s_2$	$s_1^{\mathcal{I}} \cap s_2^{\mathcal{I}}$
role union	$s_1 \cup s_2,\ t_1 + t_2$	$s_1^{\mathcal{I}} \cup s_2^{\mathcal{I}},\ t_1^{\mathcal{I}} \cup t_2^{\mathcal{I}}$
role difference	$s_1 \setminus s_2$	$s_1^{\mathcal{I}} \setminus s_2^{\mathcal{I}}$
role concatenation	$t_1 \circ t_2$	$\{(\mathrm{c},\mathrm{e}) \mid \exists \mathrm{d} \in \Delta^{\mathcal{I}}.(\mathrm{c},\mathrm{d}) \in t_1^{\mathcal{I}} \wedge (\mathrm{d},\mathrm{e}) \in t_2^{\mathcal{I}}\}$
Kleene star	t^*	$\bigcup_{i=0}^{\infty}(t^{\mathcal{I}})^i$
concept test	$\mathrm{C}?$	$\{(\mathrm{d},\mathrm{d}) \mid \mathrm{d} \in \mathrm{C}^{\mathcal{I}}\}$

We next make precise the content of ABoxes and TBoxes in $\mathcal{ZOIQ}$. This is illustrated by Table 2.2 below, assuming that $\mathsf{a},\mathsf{b} \in \mathbf{N_I}$, $\mathrm{A} \in \mathbf{N_C}$, $r \in \mathbf{N_R}$, s, t are simple roles, and C,D are $\mathcal{ZOIQ}$-concepts. We stress that our definition of ABoxes do not involve complex concepts, *i.e.* concepts that are not just concept names. This decision is crucial for the forthcoming section on the data complexity of $\mathcal{ZOIQ}$ and its fragments. Otherwise, as TBoxes can be internalised with $\mathcal{ALC}_{\mathsf{reg}}$-concepts [BCM+03, p. 186], already the ABox satisfiability problem would be ExpTime-hard for $\mathcal{ALC}_{\mathsf{reg}}$ (and thus also for $\mathcal{ZOIQ}$).

Table 2.2: ABoxes and TBoxes in $\mathcal{ZOIQ}$.

Axiom α	$\mathcal{I} \models \alpha$, if	
$\mathrm{C} \sqsubseteq \mathrm{D}$	$\mathrm{C}^{\mathcal{I}} \subseteq \mathrm{D}^{\mathcal{I}}$	TBox $\mathcal{T}$
$s \subseteq t$	$s^{\mathcal{I}} \subseteq t^{\mathcal{I}}$	
$\mathrm{A}(\mathsf{a})$	$\mathsf{a}^{\mathcal{I}} \in \mathrm{A}^{\mathcal{I}}$	ABox $\mathcal{A}$
$r(\mathsf{a},\mathsf{b})$	$(\mathsf{a}^{\mathcal{I}},\mathsf{b}^{\mathcal{I}}) \in r^{\mathcal{I}}$	
$\mathsf{a} \approx \mathsf{b}$	$\mathsf{a}^{\mathcal{I}} = \mathsf{b}^{\mathcal{I}}$	
$\neg\alpha$	$\mathcal{I} \not\models \alpha$	

As a common umbrella for reasoning about decidable fragments of $\mathcal{ZOIQ}$, we focus on *tamed* $\mathcal{ZOIQ}$. It comprises all $\mathcal{ZOIQ}$-KBs possessing a certain model-theoretic property related to forest-like models dubbed *quasi-forest* (defined in Chapter 7). Tamed $\mathcal{ZOIQ}$ strictly subsumes $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, and $\mathcal{ZOI}$.

Definition 2.18 A $\mathcal{ZOIQ}$-KB $\mathcal{K}$ has the **quasi-forest hom-cover property (QFHC)** if for every model $\mathcal{I}$ of $\mathcal{K}$ there is an $\mathsf{ind}(\mathcal{K})$-homomorphism from $\mathcal{I}$ into some **quasi-forest model** $\mathcal{I}'$ of $\mathcal{K}$. **Tamed** $\mathcal{ZOIQ}$ consists of all $\mathcal{ZOIQ}$-KB exhibiting the QFHC.

Normal Form for $\mathcal{ZOIQ}$.

To simplify reasoning about $\mathcal{ZOIQ}$-knowledge-bases we transform them into a suitable polynomial-time computable Scott's normal form [BK22, Sec. 2]. Such a normal form involves *automaton roles* [Ngu20, Sec. 2.2] and existential restrictions of the form $\exists\mathcal{A}.\mathrm{C}$ for each automaton $\mathcal{A}$ over finite subsets of Σ_{all} extended with simple roles, defined and interpreted as in Section 2.7.

Definition 2.19 A $\mathcal{ZOIQ}$-TBox is in **Scott's normal form** if all its GCIs are in the following forms:

$$\mathrm{A} \equiv \{\mathsf{o}\}, \quad \mathrm{A} \equiv \mathrm{B}, \quad \mathrm{A} \equiv \neg\mathrm{B}, \quad \mathrm{A} \equiv \mathrm{B}\sqcap\mathrm{B}', \quad \mathrm{A} \equiv (\geqslant n\ r).\top, \quad \mathrm{A} \equiv \exists\mathcal{A}_{\mathsf{q},\mathsf{q}'}.\top, \quad \mathrm{A} \equiv \exists r.\mathsf{Self}, \quad s = s'$$

where $\mathrm{A}, \mathrm{B}, \mathrm{B}' \in \mathbf{N_C} \cup \{\top, \bot\}$, r is a role name, s and s' are simple roles, $\mathcal{A}$ is an NFA, and o is an individual name. A $\mathcal{ZOIQ}$-KB is in **Scott's normal form** if so is its TBox.

The main goal of this section is to sketch the proof that every $\mathcal{ZOIQ}$-KB can be turned into Scott's normal form in PTime. We first get rid of the universal role from concepts of the form $\forall\overline{\top}.\mathrm{C}$ and $\exists\overline{\top}.\mathrm{C}$.

Lemma 2.20 (follows from the PhD thesis of M. Ortiz) For any $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ there exists a $\mathcal{ZOIQ}$-KB $\mathcal{K}' := (\mathcal{A}_\top, \mathcal{T}')$ of size polynomial in $|\mathcal{K}|$ and computable in polynomial time w.r.t. $|\mathcal{K}|$ such that the universal role $\overline{\top}$ does not occur in $\mathcal{K}'$, and all the conditions below hold.

- $\mathcal{K}'$ is tamed $\mathcal{ZOIQ}$ if and only if $\mathcal{K}$ is.
- For all logics $\mathcal{DL} \in \{\mathcal{ZOQ}, \mathcal{ZOI}, \mathcal{ZIQ}\}$ we have that if $\mathcal{K}$ is in $\mathcal{DL}$ then so is $\mathcal{K}'$.
- $\mathcal{A}_\top := \mathcal{A} \cup \{s_\top(\mathsf{a}_1, \mathsf{a}_2), \ldots, s_\top(\mathsf{a}_{n-1}, \mathsf{a}_n), s_\top(\mathsf{a}_n, \mathsf{a}_1)\}$, where $s_\top$ is a fresh role name, and $\mathsf{a}_1, \ldots, \mathsf{a}_n$ is some enumeration of individual names from $\mathcal{A}$.
- $\mathcal{K}'$ is satisfiable if and only if $\mathcal{K}$ is.
- For every P2RPQ q we have $\mathcal{K}' \not\models q$ if and only if $\mathcal{K} \not\models q$.

Proof sketch. The desired rewriting is given by Ortiz [Ort10, p. 39, l. 7–17] in her PhD Thesis, together with its correctness proof [Ort10, Proof of Prop. 3.1.5: l .27 p. 40 – l. 7 p. 41]. By analising her proof, it is not hard to see that all the conditions from Lemma 2.20 indeed hold. □

Before moving forward, we introduce a handy notion of (model) conservative extensions.

Definition 2.21 We say that a $\mathcal{ZOIQ}$-KB $\mathcal{K}'$ is a **conservative extension** of a $\mathcal{ZOIQ}$-KB $\mathcal{K}$ if (i) every model of $\mathcal{K}'$ is also a model of $\mathcal{K}$, and (ii) every model of $\mathcal{K}$ can be extended (by reinterpreting symbols that appear in $\mathcal{K}'$ but not in $\mathcal{K}$) to a model of $\mathcal{K}'$.

Note that the transformation by Ortiz employed in the proof of Lemma 2.20 does not yield a conservative extension of the input KB. We also point out that if a KB $\mathcal{K}'$ is a conservative extension of $\mathcal{K}$ then for all P2RPQs that use only concept and role names that are present in $\mathcal{K}$, then $\mathcal{K}' \models q$ if and only if $\mathcal{K} \models q$.

For the next lemma, we first introduce a measure of the complexity of $\mathcal{ZOIQ}$-concepts.

Definition 2.22 The **complexity** of a $\mathcal{ZOIQ}$-concept C, denoted $\mathrm{cmp}(\mathrm{C})$, is defined inductively as:

- $\mathrm{cmp}(\mathrm{C}) = 0$ if $\mathrm{C} \in \mathbf{N_C} \cup \{\top, \bot\}$.
- $\mathrm{cmp}(\mathrm{C}) = 1$ if $\mathrm{C} = \{\mathsf{o}\}$ for some individual name $\mathsf{o} \in \mathbf{N_I}$.
- $\mathrm{cmp}(\mathrm{C}) = 1$ if C has the form $\exists s.\mathsf{Self}$.
- $\mathrm{cmp}(\mathrm{C}) = 1 + \mathrm{cmp}(\mathrm{D})$ if $\mathrm{C} = \neg\mathrm{D}$ for some $\mathcal{ZOIQ}$-concept D.
- $\mathrm{cmp}(\mathrm{C}) = 1 + \mathrm{cmp}(\mathrm{C}_1) + \mathrm{cmp}(\mathrm{C}_2)$ if $\mathrm{C} = \mathrm{C}_1 \sqcap \mathrm{C}_2$ or $\mathrm{C} = \mathrm{C}_1 \sqcup \mathrm{C}_2$ for $\mathcal{ZOIQ}$-concepts C_1, C_2.
- $\mathrm{cmp}(\mathrm{C}) = 1 + \mathrm{cmp}(t) + \mathrm{cmp}(\mathrm{D})$ if C has the form either $\exists t.\mathrm{D}$ or $\forall t.\mathrm{D}$, and $\mathrm{cmp}(t)$ denotes the sum of the complexities of all concepts E for which the test E? appears in t.

- $\mathrm{cmp}(\mathrm{C}) = 1 + \mathrm{cmp}(\mathrm{D})$ if C has the form either $(\geqslant n\ s).\mathrm{D}$ or $(\leqslant n\ s).\mathrm{D}$.

By the **complexity** of GCIs $\mathrm{C} \sqsubseteq \mathrm{D}$ and $\mathrm{C} \equiv \mathrm{D}$ we mean the number $\mathrm{cmp}(\mathrm{C}) + \mathrm{cmp}(\mathrm{D})$. The **complexity** of a TBox is the sum of the complexities of its GCIs. A GCI is **flat** if it has the form $\mathrm{A} \equiv \mathrm{B}$, where $\mathrm{cmp}(\mathrm{A}) = 0$ and $\mathrm{cmp}(\mathrm{B}) \leq 1$. A TBox is **flat** if all its GCIs are flat.

By employing a rewriting algorithm based on a routine renaming technique, we show that every $\mathcal{ZOIQ}$-KB can be made flat. This is illustrated by the pseudocode below. Given a TBox $\mathcal{T}$ and concepts D and B, by $\mathcal{T}[\mathrm{D}/\mathrm{B}]$ we mean the TBox obtained by replacing all occurrences of D in $\mathcal{T}$ with B.

Procedure 1: Making $\mathcal{ZOIQ}$-TBoxes flat.

```
Input: A $\mathcal{ZOIQ}$-TBox $\mathcal{T}$ that does not contain the universal role $\overline{\top}$ and $\equiv$.
Output: A flat $\mathcal{ZOIQ}$-TBox $\mathcal{T}'$, which is a conservative extension of $\mathcal{T}$.
1  Let $\mathcal{T}' := \mathcal{T}$.                    // We start from $\mathcal{T}$ and recursively rewrite it.
2  while $\mathcal{T}'$ contains a GCI $\mathrm{C} \sqsubseteq \mathrm{D}$ do
3      Replace $\mathrm{C} \sqsubseteq \mathrm{D}$ with $\top \equiv \neg\mathrm{C} \sqcup \mathrm{D}$.     // Correctness: by the semantics of implication.
   // Observation: from now on all GCIs in $\mathcal{T}'$ have the form $\mathrm{A} \equiv \mathrm{C}$, where the complexity of A is 0.
4  while $\mathcal{T}'$ is not flat do
5      Let $\mathrm{A} \equiv \mathrm{C}$ be any GCI from $\mathcal{T}'$ of maximal complexity.  // By design: $\mathrm{cmp}(\mathrm{A}) = 0$, $\mathrm{cmp}(\mathrm{C}) \geq 2$.
           // Below we consider all possible shapes of C. As $\mathrm{cmp}(\mathrm{C}) \geq 2$, no other option is possible.
6      If $\mathrm{C} = \neg\mathrm{D}$ then $\mathcal{T}'' := (\mathcal{T}'[\mathrm{D}/\mathrm{B}] \cup \{\mathrm{B} \equiv \mathrm{D}\})$ for a fresh concept name B.
7      If $\mathrm{C} = \mathrm{D} \otimes \mathrm{D}'$ or $\mathrm{C} = \mathrm{D}' \otimes \mathrm{D}$ for $\otimes \in \{\sqcap, \sqcup\}$ and $\mathrm{cmp}(\mathrm{D}) > 0$ then
         $\mathcal{T}'' := (\mathcal{T}'[\mathrm{D}/\mathrm{B}] \cup \{\mathrm{B} \equiv \mathrm{D}\})$ for a fresh concept name B.
8      If $\mathrm{C} = \mathcal{Q}t.\mathrm{D}$ for $\mathcal{Q} \in \{\exists, \forall, \leqslant n, \geqslant n\}$ and $\mathrm{cmp}(\mathrm{D}) > 0$ then let $\mathcal{T}'' := (\mathcal{T}'[\mathrm{D}/\mathrm{B}] \cup \{\mathrm{B} \equiv \mathrm{D}\})$ for
         a fresh concept name B.
9      If $\mathrm{C} = \mathcal{Q}t.\mathrm{E}$ for $\mathcal{Q} \in \{\exists, \forall\}$ and $\mathrm{cmp}(\mathrm{E}) = 0$ then take any concept D with $\mathrm{cmp}(\mathrm{D}) > 0$ for
         which D? appears in $t$ and let $\mathcal{T}'' := (\mathcal{T}'[\mathrm{D}?/\mathrm{B}?] \cup \{\mathrm{B} \equiv \mathrm{D}\})$ for a fresh concept name B.
10     $\mathcal{T}' := \mathcal{T}''$.
11 Return $\mathcal{T}'$.
```

We now provide several useful observations concerning Procedure 1, needed to establish its correctness. They follow immediately by analysing the above algorithm.

Observation 2.23. Let $\mathcal{T}$ be an input for Procedure 1. Then the following properties hold.

1. The TBox $\mathcal{T}'$ constructed in Steps 1–3 of Procedure 1 is equivalent to $\mathcal{T}$, has the size linear in $|\mathcal{T}|$, and its complexity is at most twice the complexity of $\mathcal{T}$.
2. In each iteration of the while loop in Step 4 of Procedure 1 applied to the TBox $\mathcal{T}'$ and a GCI $\mathrm{A} \equiv \mathrm{C}$ we see that:
   (I) the GCI $\mathrm{B} \equiv \mathrm{D}$ appended in Steps 5–9 have strictly smaller complexity than the GCI $\mathrm{A} \equiv \mathrm{C}$,
   (II) the GCI $(\mathrm{A} \equiv \mathrm{C})[\mathrm{D}/\mathrm{B}]$ is either flat or its complexity is smaller than the complexity of $\mathrm{A} \equiv \mathrm{C}$,
   (III) hence, the total number of GCIs in $\mathcal{T}'$ having the maximal complexity decreases and the complexity of the resulting TBox does not increase,
   (IV) the resulting TBox $\mathcal{T}''$ is a conservative extension of $\mathcal{T}'$ (it suffices to interpret B equally to D),
   (V) the difference between the size of $\mathcal{T}''$ and $|\mathcal{T}'|$ can be bounded by a constant,
   (VI) and if $\mathcal{T}'$ is in $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, or $\mathcal{ZOI}$, then so is $\mathcal{T}''$.
3. The total number of iterations of the while loop in Step 4 of Procedure 1 is bounded by the total number of subconcepts of $\mathcal{T}'$ constructed in Steps 1–3 of Procedure 1 (and hence, is linear in $|\mathcal{T}|$).

Based on Observation 2.23 and analysis of the above pseudocode, we conclude:

Lemma 2.24 For every $\mathcal{ZOIQ}$-TBox $\mathcal{T}$ that does not employ the universal role $\overline{\top}$, one can compute with Procedure 1 in polynomial time a polynomially larger *flat* conservative extension $\mathcal{T}'$ of $\mathcal{T}$ such that for all logics $\mathcal{DL} \in \{\mathcal{ZOQ}, \mathcal{ZOI}, \mathcal{ZIQ}\}$ we have that if $\mathcal{T}$ is in $\mathcal{DL}$ then so is $\mathcal{T}'$.

Proof sketch. Let $\mathcal{T}'$ be the result of Procedure 1 applied to a $\mathcal{ZOIQ}$-TBox $\mathcal{T}$. Its termination in polynomial time w.r.t. $|\mathcal{T}|$ is due to Items 1 and 3 from Observation 2.23. From the 4th Step of Procedure 1 and its termination, we see that $\mathcal{T}'$ is flat. The fact that $\mathcal{T}'$ is a conservative extension of $\mathcal{T}$ is due to Item 2(IV) of Observation 2.23. The fact that $\mathcal{T}'$ is bounded polynomially w.r.t. $|\mathcal{T}|$ is due to Item 2(V) of Observation 2.23. Finally, the fact that $\mathcal{T}'$ belong to the same sublogic of $\mathcal{ZOIQ}$ as $\mathcal{T}$ does, follows from Item 2(VI) of Observation 2.23. □

As the next and final step, we rewrite flat $\mathcal{ZOIQ}$-TBoxes to Scott's normal form.

Procedure 2: From flat $\mathcal{ZOIQ}$-TBoxes to Scott's normal form.

```
Input: A flat $\mathcal{ZOIQ}$-TBox $\mathcal{T}$ that does not contain the universal role $\overline{\top}$.
Output: A $\mathcal{ZOIQ}$-TBox $\mathcal{T}'$ in Scott's normal form, which is a conservative extension of $\mathcal{T}$.
1  Let $\mathcal{T}' := \mathcal{T}$.                    // We start from $\mathcal{T}$ and recursively rewrite it.
2  while $\mathcal{T}'$ is not in Scott's normal form we take any axiom $\alpha \in \mathcal{T}'$ violating it and do
       // Below we consider all possible shapes of the axiom $\alpha$. No other option is possible.
3      If $\alpha = s \subseteq s'$ then $\mathcal{T}' := (\mathcal{T}' \setminus \{\alpha\}) \cup \{s \cup r = s'\}$ for a fresh role name $r$.
                                              // Correctness follows from the semantics of $\cup$.
4      If $\alpha = (\mathrm{A} \equiv \mathrm{B} \sqcup \mathrm{B}')$ then $\mathcal{T}' := (\mathcal{T}' \setminus \{\alpha\}) \cup \{\mathrm{A} \equiv \neg\mathrm{C}, \mathrm{C} \equiv \mathrm{D} \sqcap \mathrm{D}', \mathrm{D} \equiv \neg\mathrm{B}, \mathrm{D}' \equiv \neg\mathrm{D}'\}$ for
        fresh concept names $\mathrm{C}, \mathrm{D}, \mathrm{D}' \in \mathbf{N_C}$.  // Here we use the equivalence of $\mathrm{B} \sqcup \mathrm{B}'$ and $\neg(\neg\mathrm{B} \sqcap \neg\mathrm{B}')$.
5      If $\alpha = (\mathrm{A} \equiv \exists s.\mathsf{Self})$ for $s \notin \mathbf{N_R}$ then $\mathcal{T}' := (\mathcal{T}' \setminus \{\alpha\}) \cup \{\mathrm{A} \equiv \exists r.\mathsf{Self}, r = s\}$ for a fresh role
        name $r \in \mathbf{N_R}$.                                     // Obvious renaming.
6      If $\alpha = (\mathrm{A} \equiv \exists t.\mathrm{B})$ then we construct an NFA $\mathcal{A}$ equivalent to the regular expression $t \circ \mathrm{B}?$
        with a starting state $\mathsf{q}$ and a single final state $\mathsf{q}'$, and let $\mathcal{T}' := (\mathcal{T}' \setminus \{\alpha\}) \cup \{\mathrm{A} \equiv \exists \mathcal{A}_{\mathsf{q},\mathsf{q}'}.\top\}$.
        // Clearly the concepts $\exists t.\mathrm{B}$ and $\exists(t \circ \mathrm{B}?).\top$ are equivalent. The construction of an NFA from a
        regular expression is a classical result. This can be done with Thompson's construction [Tho68]
        and results in a polynomial-size NFA [GH15, Thm. 6] w.r.t. a given regular expression.
7      If $\alpha = (\mathrm{A} \equiv \forall t.\mathrm{B})$ then $\mathcal{T}' := (\mathcal{T}' \setminus \{\alpha\}) \cup \{\mathrm{A} \equiv \neg\mathrm{C}, \mathrm{C} \equiv \exists t.\mathrm{D}, \mathrm{D} \equiv \neg\mathrm{B}\}$ for fresh $\mathrm{C}, \mathrm{D} \in \mathbf{N_C}$.
        // Concepts $\forall t.\mathrm{B}$ and $\neg(\exists r.\neg\mathrm{B})$ are equivalent. The GCI $\mathrm{C} \equiv \exists t.\mathrm{D}$ will be simplified recursively.
8      If $\alpha = (\mathrm{A} \equiv (\geqslant n\ s).\mathrm{B})$ then $\mathcal{T}' := (\mathcal{T}' \setminus \{\alpha\}) \cup \{\mathrm{A} \equiv (\geqslant n\ r).\top, r \cup r' = s, \top \equiv \forall r.\mathrm{B},$
        $\bot \equiv \exists r'.\mathrm{B}\}$ for a fresh role names $r$ and $r'$.  // Here we employ two fresh role names $r$ and $r'$ to
        divide $s$-successors (with the axiom $r \cup r' = s$) into disjoint sets of $r$-successors (precisely the
        ones satisfying $\mathrm{B}$, guaranteed by the GCI $\top \equiv \forall r.\mathrm{B}$) and $r'$-successors (the ones not satisfying
        $\mathrm{B}$, as guaranteed by the GCI $\bot \equiv \exists r'.\mathrm{B}$). In the presence of such axioms, the concepts $(\geqslant n\ s).\mathrm{B}$
        and $(\geqslant n\ r).\top$ are clearly equivalent. We then simplify $\top \equiv \forall r.\mathrm{B}$ recursively.
9      If $\alpha = (\mathrm{A} \equiv (\leqslant n\ s).\mathrm{B})$ then $\mathcal{T}' := (\mathcal{T}' \setminus \{\alpha\}) \cup \{\mathrm{A} \equiv \neg\mathrm{D}, \mathrm{C} \equiv \neg\mathrm{B}, \mathrm{D} \equiv (\geqslant n{+}1\ s).\mathrm{C}\}$ for fresh
        concept names $\mathrm{C}, \mathrm{D} \in \mathbf{N_C}$.  // We use the equivalence of concepts $(\leqslant n\ s).\mathrm{B}$ and $\neg(\geqslant n{+}1\ s).\neg\mathrm{B}$.
        The GCI $\mathrm{D} \equiv (\geqslant n{+}1\ s).\mathrm{C}$ will be simplified recursively.
10 Return $\mathcal{T}'$.
```

Lemma 2.25 For every flat $\mathcal{ZOIQ}$-TBox $\mathcal{T}$ that does not employ the universal role $\overline{\top}$, one can compute with Procedure 2 in polynomial time a polynomially larger *flat* conservative extension $\mathcal{T}'$ of $\mathcal{T}$ such that for all logics $\mathcal{DL} \in \{\mathcal{ZOQ}, \mathcal{ZOI}, \mathcal{ZIQ}\}$ we have that if $\mathcal{T}$ is in $\mathcal{DL}$ then so is $\mathcal{T}'$.

Proof sketch. By analysing Procedure 2 we see that for each axiom α we (recursively) produce at most 8 axioms in Scott's normal form, each of size linear w.r.t. the input TBox. Moreover, the number of recursive calls needed to rewrite an axiom α is bounded by 3 (the worst case happens for GCIs of the form $\mathrm{A} \equiv (\leqslant n\ s).\mathrm{B}$). Hence, for an input $\mathcal{T}$, Procedure 2 terminates in polynomial time w.r.t. $|\mathcal{T}|$ and produces a TBox $\mathcal{T}'$ of polynomial size w.r.t. $|\mathcal{T}|$. The resulting TBox is in Scott's normal form, which follows by analysing the GCIs produced by Steps 3–9 and the condition in the while loop from Step 2. Note that our transformation does not introduce nominals, inverses, or number restrictions in case they were not present in the input. Hence, the resulting TBox is in the same fragment of $\mathcal{ZOIQ}$ as the input TBox. Finally, it can be readily checked that the TBoxes produced in Steps 3–9 of Procedure 2 from a given

TBox $\mathcal{T}'$, are indeed conservative extensions of $\mathcal{T}'$. The correctness of our transformations is provided in the comments appearing in our pseudocode. □

We can now combine Lemma 2.25, Lemma 2.24, and Lemma 2.20 to provide a proof that $\mathcal{ZOIQ}$-KBs can be turned into Scott's normal form in PTime. The statement of the forthcoming lemma is written in a slightly unexpected way, to make it applicable also in sections concerning the data complexity.

Lemma 2.26 For every $\mathcal{ZOIQ}$-TBox $\mathcal{T}$ we can compute in PTime a $\mathcal{ZOIQ}$-TBox $\mathcal{T}'$ in Scott's normal form that possibly employs a fresh role name $s_{\top}$, such that for every ABox $\mathcal{A}$ we have:

- if $\mathcal{T}$ is in $\mathcal{DL} \in \{\mathcal{ZIQ}, \mathcal{ZOQ}, \mathcal{ZOI}\}$ then $\mathcal{T}'$ is also in $\mathcal{DL}$,
- $(\mathcal{A}, \mathcal{T})$ is satisfiable if and only if $(\mathcal{A}_{\top}, \mathcal{T}')$ is satisfiable, and
- for every P2RPQ q that uses only concepts and roles present in $\mathcal{T}$ we have that $(\mathcal{A}, \mathcal{T}) \models q$ if and only if $(\mathcal{A}_{\top}, \mathcal{T}') \models q$,

where $\mathcal{A}_{\top} := \mathcal{A} \cup \{s_{\top}(\mathsf{a}_1, \mathsf{a}_2), \ldots, s_{\top}(\mathsf{a}_{n-1}, \mathsf{a}_n), s_{\top}(\mathsf{a}_n, \mathsf{a}_1)\}$ for some enumeration $\mathsf{a}_1, \ldots, \mathsf{a}_n$ of individual names appearing in $\mathcal{A}$.

Proof sketch. Take a $\mathcal{ZOIQ}$-TBox $\mathcal{T}$ and an ABox $\mathcal{A}$. Let $\mathcal{A}_{\top}$ be as defined above. Call a $\mathcal{ZOIQ}$-TBox $\mathcal{T}'$ *neat* if it satisfies the three conditions from Lemma 2.26. We first apply Lemma 2.20 to compute (in polynomial time) a neat $\mathcal{ZOIQ}$-TBox $\mathcal{T}_0$ that does not use the universal role. Second, we invoke Lemma 2.24 to compute (in PTime) a flat $\mathcal{ZOIQ}$-TBox $\mathcal{T}_1$ from $\mathcal{T}_0$. Observe that the TBox $\mathcal{T}_1$ is neat. Indeed, the first condition of being neat follows from the statement of Lemma 2.24. The remaining two conditions follow from the fact that $\mathcal{T}_1$ is a conservative extension of $\mathcal{T}_0$. Third, we apply Lemma 2.25 to compute (in PTime) a $\mathcal{ZOIQ}$-TBox $\mathcal{T}'$ in Scott's normal form from $\mathcal{T}_1$. Once more, $\mathcal{T}_1$ is neat. The first condition is explicitly mentioned in Lemma 2.25. The other two conditions follow by the fact that $\mathcal{T}'$ is a conservative extension of $\mathcal{T}_1$. Hence, the TBox $\mathcal{T}'$ is as desired. □

Part I

Query Entailment in Forest-Friendly Description Logics

3

Forest-Friendly Description Logics

Contents

Motivation and Our Contribution

In the quest to find computational logics, Moshe Vardi [Var96] asked his famous question *"Why Is Modal Logic So Robustly Decidable?"*. As it turned out, one of the prominent answers is the so-called *tree model property* of modal logics [GO07, Cor. 24] and their generalisations to the guarded fragment [Grä99, Sec. 3.3] and beyond [BBV16, Sec. 4][FLOR23, Thm. 7]. In ontology-based querying, due to the presence of individual names in ABoxes, an analogous role is played by *forests* [Ov12, Sec. 8.1].

This chapter is intended to serve as an introduction to different notions of forests and trees that will play crucial roles in abstract generalisations of the description logics $\mathcal{ALC}$, $\mathcal{ALCI}$, $\mathcal{ALC}^{\mathsf{Self}}$, and $\mathcal{ALCI}^{\mathsf{Self}}$. Our ambition is to cover multiple description logics in one go, without the need of reproducing nearly-identical proofs for freshly defined logics. Hence, nearly all of the forthcoming notions will be parametrised by a set of features $\Theta \subseteq \{\mathsf{I}, \mathsf{Self}\}$, indicating what kind of DL-like features we have in mind when producing a definition, a proof, or an algorithm. We hope that such a parametrisation will not obfuscate the content of the forthcoming chapters too much, and in general we found it better than copy-pasting proofs.

Overview of the Chapter and Prerequisites

We assume that the reader is familiar with the definition of $\mathcal{ALCI}^{\mathsf{Self}}$, and Sections 2.1–2.5 from Preliminaries. We start by defining a suitable notion of "forest-like" structures in Section 3.1 as well as setting up the terminology employed in later parts of the thesis. Next, in Section 3.2 we will see the correspondence between tree-shaped queries and concepts in the spirit of the well-known rolling-up technique [HT00, Sec. 4]. We conclude with Section 3.3 which relaxes the notion of "forest-likeness" to (local) neighbourhoods, and introduces broad families of forest-friendly description logics that we will study later.

3.1 Treelike structures

Throughout the chapter we employ the standard set-theoretic reconstruction of the notion of a **tree** as a prefix-closed subset of $\mathbb{D}^*$ for some non-empty $\mathbb{D}$ (dubbed the **data domain**). We use $\preceq$ to denote the *prefix-ordering* on $\mathbb{D}^*$ ($\mathrm{d} \preceq \mathrm{e}$ holds if d is a prefix of e), and $\cdot$ to denote concatenation. We assume that $\mathbb{D}$ is equipped with a linear order $<_{\mathbb{D}}$. If $\mathbb{D}$ equals $\mathbb{N}$, the symbol $<_{\mathbb{D}}$ denotes the usual ordering of integers.

Definition 3.1 Let Θ be a subset of $\{\mathsf{I}, \mathsf{Self}\}$. We say that an interpretation $\mathcal{I}$ is a **Θ-forest** if its domain is a prefix-closed subset of $\mathbb{D}^+$ (for some non-empty set $\mathbb{D}$) without ε, and for all role names r and all pairs $(\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}$ at least one of the following cases hold.

(a) Both d and e belong to $\mathbb{D}$.

(b) There exists $\mathrm{c} \in \mathbb{D}$ such that $\mathrm{e} = \mathrm{d} \cdot \mathrm{c}$.

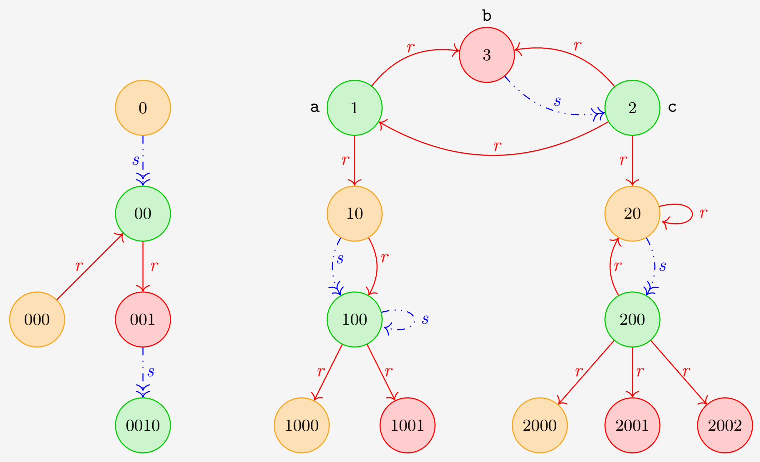
(c) $\mathsf{I} \in \Theta$ and there exists $\mathrm{c} \in \mathbb{D}$ such that $\mathrm{d} = \mathrm{e} \cdot \mathrm{c}$.

(d) $\mathsf{Self} \in \Theta$ and $\mathrm{d} = \mathrm{e}$.

We omit the parameter Θ whenever it can be deduced from the context or it is not important. The elements d from $\mathbb{D} \cap \Delta^{\mathcal{I}}$ are called **roots**. A **Θ-tree** is a connected Θ-forest with a single root.

For a $\Theta \subseteq \{\mathsf{I}, \mathsf{Self}\}$ we write $\mathcal{ALC}\Theta$ to denote the extensions of $\mathcal{ALC}$ with features from Θ. As an example, we identify $\mathcal{ALC}\emptyset$ with $\mathcal{ALC}$, and $\mathcal{ALC}\{\mathsf{I}, \mathsf{Self}\}$ with $\mathcal{ALCI}^{\mathsf{Self}}$. The presented notion of Θ-forests serves as a common umbrella for different notions of "tree-like" structures, naturally appearing in the context of reasoning in $\mathcal{ALC}\Theta$ with conjunction of roles. For instance, the presence of the letter "I" in Θ reflects the "two-wayness" of underlying forest models [Lut08b, p. 8], while the presence of Self in Θ indicates a possible appearance of self-loops. We illustrate these notions with an example below.

Example 3.2. Consider the interpretation $\mathcal{I}$ depicted below. It is easy to see that $\mathcal{I}$ is an $\{\mathsf{I}, \mathsf{Self}\}$-forest. After removing all self-loops, the resulting structure is an $\{\mathsf{I}\}$-forest. After removing all arrows going from child to a parent, the resulting structure is a $\{\mathsf{Self}\}$-forest. Finally, after removing both self-loops and reversed edges, the resulting structure is a $\emptyset$-forest. The restriction of $\mathcal{I}$ to the set $\{0, 00, 000, 001, 0010\}$ is a tree.



When working with forests we employ tailored terminology from graph theory [Die17, Sec. 1.5]. If two different domains elements satisfy $\mathrm{d} \preceq \mathrm{e}$ we say that d is an *ancestor* of e (or, alternatively, that e is a *descendant* of d). The set of all descendants of an element d in a forest $\mathcal{I}$ is denoted $\mathsf{Desc}_{\mathcal{I}}(\mathrm{d})$. If e is of the form $\mathrm{d} \cdot \mathrm{c}$ for some $\mathrm{c} \in \mathbb{D}$, we say that d is a *parent* of e (or, alternatively, e is a *child* of d). The set of all children of an element d in a forest $\mathcal{I}$ is denoted $\mathsf{Chlds}_{\mathcal{I}}(\mathrm{d})$. An element with no child is called a *leaf*, and an element without a parent is called a *root*. Whenever two elements have the same parent, we call them *siblings*. The *subtree* rooted at d in $\mathcal{I}$, denoted with $\mathcal{I}^{[\mathrm{d}\preceq]}$, is the restriction of $\mathcal{I}$ to $\{\mathrm{d}\} \cup \mathsf{Desc}_{\mathcal{I}}(\mathrm{d})$. Finally, a *branch* in $\mathcal{I}$ is a sequence of elements $\mathrm{d}_1, \mathrm{d}_2, \ldots$ from $\Delta^{\mathcal{I}}$ such that for any index $i \geq 0$ if d_{i+1} exists, then d_{i+1} is a child of d_i. For brevity, we also use the notion of Θ-*reachability*, which is understood as reachability via undirected paths if $\mathsf{I} \in \Theta$ and as reachability via directed paths otherwise. Similarly, to define the notion of Θ-*connectivity*, we replace the notion of reachability in its definition by Θ-reachability.

Example 3.3. Let $\mathcal{I}$ be the $\{\mathsf{I}, \mathsf{Self}\}$-forest from Example 3.2. Consider the element 20. It has a unique child, namely 200. Hence, $\mathsf{Chlds}_{\mathcal{I}}(20) = \{200\}$. The set of descendants of 20, denoted $\mathsf{Desc}_{\mathcal{I}}(20)$, is $\{200, 2000, 2001, 2002\}$. The roots of $\mathcal{I}$ are 0, 1, 2, and 3, while the leaves of $\mathcal{I}$ are 000, 0010, 1000, 1001, 2000, 2001, and 2002. An example branch of $\mathcal{I}$ is the sequence $2, 20, 200$.

We conclude the section by lifting the notion of "being a forest" to models of knowledge bases.

Definition 3.4 Let N be a *non-empty* set of individual names. We say that a forest $\mathcal{I}$ is **N-rooted** if:

- for all names $\mathsf{a} \in \mathsf{N}$ we have that $\mathsf{a}^{\mathcal{I}}$ is defined and it is a root of $\mathcal{I}$, and
- for each root $\mathrm{d} \in \Delta^{\mathcal{I}}$ there exists a name $\mathsf{a} \in \mathsf{N}$ for which $\mathrm{d} = \mathsf{a}^{\mathcal{I}}$.

A **forest model** of a knowledge base $\mathcal{K}$ is an $\mathsf{ind}(\mathcal{K})$-rooted forest that satisfies $\mathcal{K}$.

For convenience we also define **$\emptyset$-rooted forests** to be **trees**. This design decision may seem awkward, but it significantly reduces the total number of different cases that we need to consider in the forthcoming proofs. Moreover, it fits nicely with the fact that ABox-free knowledge bases written in the extensions of $\mathcal{ALC}$ often have tree models rather than forest models [BHLS17, Sec. 3.5].

3.2 Correspondence between Θ-trees and $\mathcal{ALC}\Theta^{\cap}$

We next revisit the well-known rolling-up technique [HT00, Sec. 4] of transforming tree-shaped conjunctive queries into concepts, and provide a link between Θ-trees and $\mathcal{ALC}\Theta^{\cap}$-definable properties. Let q be a conjunctive query that is **Θ-tree-shaped**, *i.e.* the query q viewed as an interpretation $\mathcal{I}_q$ is a Θ-tree. Our goal is to construct, for every variable $x \in \mathrm{Var}(q)$, an $\mathcal{ALC}\Theta^{\cap}$-concept Subt_q^x stating that $\mathrm{d} \in (\mathrm{Subt}_q^x)^{\mathcal{I}}$ holds whenever the subtree of $\mathcal{I}_q$ rooted at the variable x can be mapped below d in $\mathcal{I}$ (made precise in Lemma 3.6). A formal, inductive definition is given next. The main idea behind the definition is to traverse the input tree in a bottom-up manner, describing its shape with $\mathcal{ALC}\Theta^{\cap}$ concepts, and gradually "rolling-up" the input Θ-tree into smaller chunks until the root is reached.

Definition 3.5 For a Θ-tree-shaped conjunctive query q and any of its variables $v \in \mathrm{Var}(q)$ we define an $\mathcal{ALC}\Theta^{\cap}$-concept Subt_q^v as:

$$\mathrm{Subt}_q^v := \bigsqcap_{\mathrm{A}(v)\in q} \mathrm{A} \sqcap \bigsqcap_{r(v,v)\in q \text{ only if } \mathsf{Self}\in\Theta} \exists r.\mathsf{Self} \sqcap \bigsqcap_{u\in\mathsf{Chlds}(v)} \exists\left(\bigcap_{r(v,u)\in q} r \cap \bigcap_{r(u,v)\in q \text{ only if } \mathsf{I}\in\Theta} r^{-}\right).\mathrm{Subt}_q^u,$$

where the empty conjunctions and intersections are omitted in the definition. We employ Match_q as an abbreviation of $\mathrm{Subt}_q^{v_r}$ with v_r being the root of $\mathcal{I}_q$.

As every query atom contributes to exactly one subconcept of Match_q, we infer that the size of Match_q is linear in $|q|$. The following lemma is folklore in the description logic community.

Lemma 3.6 For any interpretation $\mathcal{I}$, any Θ-tree-shaped CQ q and any of its variables $v \in \mathrm{Var}(q)$, the following equivalence holds: $\mathrm{d} \in (\mathrm{Subt}_q^v)^{\mathcal{I}}$ if and only if there exists a homomorphism $\mathfrak{h}\colon \mathcal{I}_q^{[v\preceq]} \to \mathcal{I}$ satisfying $\mathfrak{h}(v) = \mathrm{d}$, where $\mathcal{I}_q^{[v\preceq]}$ denotes the subtree of $\mathcal{I}_q$ rooted at the variable v.

Proof. We prove the statement inductively on the ordering $\prec$ (*i.e.* the strict $\preceq$). We start from the base case, *i.e.* when x is $\prec$-maximal (x is a leaf). The domain of $\mathcal{I}_q^{[x\preceq]}$ is then precisely $\{x\}$.

- From $\mathrm{d} \in (\mathrm{Subt}_q^x)^{\mathcal{I}}$ to the existence of a homomorphism $\mathfrak{h}\colon \mathcal{I}_q^{[x\preceq]} \to \mathcal{I}$ satisfying $\mathfrak{h}(x) = \mathrm{d}$. It suffices to show that $\mathfrak{h}$ preserves concepts and self-loops (in the case $\mathsf{Self} \in \Theta$). Consider a concept name $\mathrm{A} \in \mathbf{N_C}$ for which $x \in \mathrm{A}^{\mathcal{I}_q^{[x\preceq]}}$. By the definition of $\mathcal{I}_q^x$ we infer $\mathrm{A}(x) \in q$. Hence, by Definition 3.5, the concept name A appears in the first "big conjunct" of Subt_q^x. As $\mathrm{d} \in (\mathrm{Subt}_q^x)^{\mathcal{I}}$, we conclude that $\mathfrak{h}(x)$ (which is equal to d) belongs to $\mathrm{A}^{\mathcal{I}}$. To deal with self-loops, we consider a role name $r \in \mathbf{N_R}$ for which $(x, x) \in r^{\mathcal{I}_q^{[x\preceq]}}$. Then, we proceed analogously, relying on the semantics of Self and the second "big conjunct" of Subt_q^x.
- From the existence of a homomorphism $\mathfrak{h}\colon \mathcal{I}_q^x \to \mathcal{I}$, defined as $\mathfrak{h}(x) := \mathrm{d}$, to $\mathrm{d} \in (\mathrm{Subt}_q^x)^{\mathcal{I}}$. We simply consider all conjuncts from Subt_q^x and prove the membership of d in them. Take any of them. There are only two possible options: either the selected conjunct is A

for some concept name $\mathrm{A} \in \mathbf{N_C}$ or is $\exists r.\mathsf{Self}$ for some role name $r \in \mathbf{N_R}$ (when $\mathsf{Self} \in \Theta$). We can apply Definition 3.5 to infer that $\mathrm{A}(x) \in q$ (and $r(x,x) \in q$ in the second case) By the definition of $\mathcal{I}_q^x$, we get $x \in \mathrm{A}^{\mathcal{I}_q^{[x\preceq]}}$ (and $(x,x) \in r^{\mathcal{I}_q^{[x\preceq]}}$ in the second case). By the fact that $\mathfrak{h}$ is a homomorphism, we infer that $\mathfrak{h}(x)$ (which is equal to d) belongs to $\mathrm{A}^{\mathcal{I}}$, and that $(\mathfrak{h}(x), \mathfrak{h}(x))$ (which is equal to (d, d)) belongs to $r^{\mathcal{I}}$. Hence, d belongs to each conjunct of Subt_q^x, resulting in $\mathrm{d} \in (\mathrm{Subt}_q^x)^{\mathcal{I}}$, as desired.

Now, as we are done with the base case, let us assume that x is not $\prec$-maximal, and that for all variables y with $x \prec y$ the statement of the lemma holds. There are two cases to consider.

- From $\mathrm{d} \in (\mathrm{Subt}_q^x)^{\mathcal{I}}$ to the existence of a homomorphism $\mathfrak{h}\colon \mathcal{I}_q^{[x\preceq]} \to \mathcal{I}$ satisfying $\mathfrak{h}(x) = \mathrm{d}$. From the fact that $\mathrm{d} \in (\mathrm{Subt}_q^x)^{\mathcal{I}}$ and from the last "big conjunct" from Definition 3.5, we obtain that for each variable $y \in \mathsf{Chlds}(x)$ there is a domain element $\mathrm{d}_y \in \Delta^{\mathcal{I}}$ satisfying:

$$(\mathrm{d}, \mathrm{d}_y) \in \left(\bigcap_{r(x,y)\in q} r^{\mathcal{I}} \cap \bigcap_{r(y,x)\in q} (r^-)^{\mathcal{I}} \right) \text{ and } \mathrm{d}_y \in (\mathrm{Subt}_q^y)^{\mathcal{I}}. \qquad (\clubsuit)$$

  Moreover, by the induction hypothesis, for every $y \in \mathsf{Chlds}(x)$ there is a homomorphism $\mathfrak{h}_y\colon (\mathrm{Subt}_q^y)^{\mathcal{I}} \to \mathcal{I}$ with $\mathfrak{h}_y(y) = \mathrm{d}_y$. Let us define a function $\mathfrak{h}\colon \mathcal{I}_q^{[x\preceq]} \to \mathcal{I}$ as $\mathfrak{h}(x) := \mathrm{d}$ and for all $z \in \mathrm{Var}(q)$ we set $\mathfrak{h}(z) := \mathfrak{h}_y(z)$, where $y \in \mathsf{Chlds}(x)$ such that $y \preceq z$. We first explain why the definition of $\mathfrak{h}$ is correct, and then explain why $\mathfrak{h}$ is indeed a homomorphism, finishing the proof. The correctness of the definition comes directly from the fact that $\mathcal{I}_q^{[x\preceq]}$ is a Θ-tree: any variable is then either the root of $\mathcal{I}_q^{[x\preceq]}$ or has the unique ancestor being a child of the root. Now, to argue that $\mathfrak{h}$ is a homomorphism, we establish the preservation of concepts and roles by $\mathfrak{h}$. To see that $\mathfrak{h}$ preserves concepts and self-loops, we either (a) use the same reasoning as in the base case for the root variable, or (b) invoke an inductive hypothesis that $\mathfrak{h}_y$ are homomorphisms for other variables. For the preservation of roles, let us consider any role name $r \in \mathbf{N_R}$ with $r^{\mathcal{I}_q^{[x\preceq]}}$ non-empty. Due to Θ-tree-shapedness of $\mathcal{I}_q^{[x\preceq]}$, it suffices to consider to following four cases.

  (i) The set $r^{\mathcal{I}_q^{[x\preceq]}}$ contains a pair (x,x). This case is already resolved.

  (ii) The set $r^{\mathcal{I}_q^{[x\preceq]}}$ contains a pair of the form (x,y) for some $y \in \mathsf{Chlds}(x)$. Observe that if $(x,y) \in r^{\mathcal{I}_q^x}$ holds for some variable $y \in \mathsf{Chlds}(x)$ then $r(x,y) \in q$, and by Equation ($\clubsuit$) we conclude that the pair $(\mathfrak{h}(x), \mathfrak{h}(y))$, equal to $(\mathrm{d}, \mathrm{d}_y)$, belongs to $r^{\mathcal{I}}$.

  (iii) The set $r^{\mathcal{I}_q^{[x\preceq]}}$ contains a pair of the form (y,x) for some $y \in \mathsf{Chlds}(x)$. Once more, observe that if $(y,x) \in r^{\mathcal{I}_q^x}$ holds for some variable $y \in \mathsf{Chlds}(x)$ then $r(y,x) \in q$. By Equation ($\clubsuit$) we again conclude that the pair $(\mathfrak{h}(x), \mathfrak{h}(y))$, which is equal to $(\mathrm{d}, \mathrm{d}_y)$, belongs to $(r^-)^{\mathcal{I}}$. Thus the pair $(\mathfrak{h}(y), \mathfrak{h}(x))$ belongs to $r^{\mathcal{I}}$.

  (iv) The set $r^{\mathcal{I}_q^{[x\preceq]}}$ contains a pair of the form (z,v) where none of z and v is equal to x. As $\mathcal{I}_q^x$ is a Θ-tree, we deduce that there is a unique variable y from $\mathsf{Chlds}(x)$ satisfying $y \preceq z$. Hence, from the fact that $\mathfrak{h}_y$ is a homomorphism, we conclude $(\mathfrak{h}(z), \mathfrak{h}(v)) \in r^{\mathcal{I}}$.

  This completes the proof that $\mathfrak{h}$ is a desired homomorphism.

- From the existence of a homomorphism $\mathfrak{h}\colon \mathcal{I}_q^x \to \mathcal{I}$, satisfying $\mathfrak{h}(x) = \mathrm{d}$, to $\mathrm{d} \in (\mathrm{Subt}_q^x)^{\mathcal{I}}$. It suffices to show that d satisfies all the conjuncts from Subt_q^x. Showing that d belongs to the first two conjuncts (*i.e.* the ones responsible for atoms of the form $\mathrm{A}(x)$ and self-loops $r(x,x)$) can be done in precisely the same way as in the base case. Thus, we focus on the last "big conjunct". Let $\exists \left(\cap_{r(x,y)\in q} r \cap \cap_{r(y,x)\in q} r^-\right).\mathrm{Subt}_q^y$ be any of its components, and let $\mathrm{d}_y := \mathfrak{h}(y)$. Let $\mathfrak{h}_y$ be the restriction of $\mathfrak{h}$ to $\{z \mid y \preceq z\}$. Note that $\mathfrak{h}_y\colon \mathcal{I}_q^{[y\preceq]} \to \mathcal{I}$ is a homomorphism satisfying $\mathfrak{h}_y(y) = \mathrm{d}_y$. Thus, we may invoke the inductive assumption, to deduce $\mathrm{d}_y \in (\mathrm{Subt}_q^y)^{\mathcal{I}}$. It remains to show that $(\mathrm{d}, \mathrm{d}_y) \in \cap_{r(x,y)\in q} r^{\mathcal{I}} \cap \cap_{r(y,x)\in q} (r^-)^{\mathcal{I}}$. Take any atom $r(x,y) \in q$. By the definition of $\mathcal{I}_q^x$, we conclude that $(x,y) \in r^{\mathcal{I}_q^{[x\preceq]}}$. Since $\mathfrak{h}$ is a homomorphism, we get that $(\mathfrak{h}(x), \mathfrak{h}(y))$ (which is equal to $(\mathrm{d}, \mathrm{d}_y)$) belongs to $r^{\mathcal{I}}$, as required. The case of inverted roles (when $\mathsf{I} \in \Theta$) is analogous. Thus, $\mathrm{d} \in (\mathrm{Subt}_q^x)^{\mathcal{I}}$.

We established soundness of both implications, which by induction concludes the proof. □

By unravelling the definition of Match_q and by applying Lemma 3.6 for the root variable of q, as an immediate consequence we conclude Corollary 3.7. Informally, it states that any element belonging to the interpretation of Match_q "detects" a match of the tree-shaped conjunctive query q.

Corollary 3.7

For any Θ-tree-shaped conjunctive query q there exists an $\mathcal{ALC}\Theta^{\cap}$-concept Match_q of size linear in $|q|$ such that $(\mathrm{Match}_q)^{\mathcal{I}}$ is non-empty if and only if there exists a homomorphism $\mathfrak{h}\colon \mathcal{I}_q \to \mathcal{I}$.

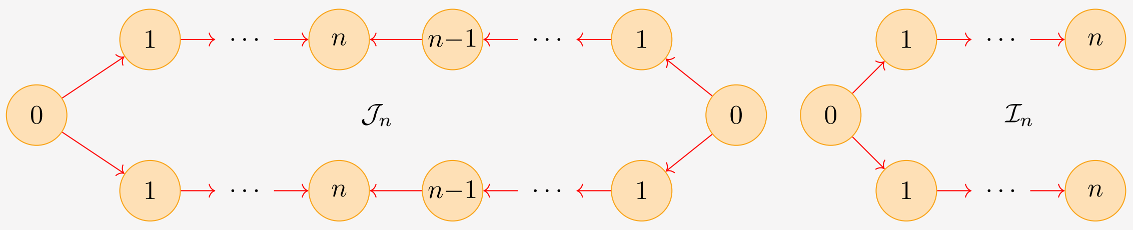
3.3 Locally Treelike Structures and Forest-Friendly Logics

As our next step, we are going to define suitable classes of structures that from the local point of view are indistinguishable from Θ-forests. The intuition is that in (n, N, Θ)-forests any sufficiently small (*i.e.* of radius n) neighbourhood resembles either a Θ-tree or a (suitably rooted) Θ-forest.

Definition 3.8 Let $n \in \mathbb{N}$ be a positive integer, and let $\mathsf{N} \subseteq \mathbf{N_I}$ be a set of names. An interpretation $\mathcal{I}$ is **(n, N)-locally Θ-forest** (short: **(n, N, Θ)-forest**) if and only if every n-neighbourhood $\mathcal{J}$ in $\mathcal{I}$ is $(\mathsf{ind}(\mathcal{J}) \cap \mathsf{N})$-homomorphically equivalent to some $(\mathsf{ind}(\mathcal{J}) \cap \mathsf{N})$-rooted Θ-forest.

For convenience, we refer to (n, N, Θ)-forests for unspecified yet parameters as *locally-forest-like* interpretations. Note that locally-forest-like structures may contain cycles composed of anonymous elements.

Remark 3.9. Consider a $\emptyset$-tree $\mathcal{I}_n$ (right), and a structure $\mathcal{J}_n$ (left) that is composed of $\mathcal{I}_n$ and its mirrored image glued together, both depicted below. Clearly $\mathcal{I}_n$ and $\mathcal{J}_n$ are homomorphically-equivalent, which implies that $\mathcal{J}_n$ is a $(2n, \emptyset, \emptyset)$-forest. On the other hand, $\mathcal{I}_n$ is acyclic and $\mathcal{J}_n$ contains an undirected cycle of size greater than n. Thus locally-forest-like structures may contain arbitrarily large undirected cycles.



We next employ (n, N)-locally Θ-forests as "coverings" of other interpretations. The property below is inspired by the quasi-forest homomorphism covers by Bourhis, Krötzsch, and Rudolph [BKR14, Prop. 1].

Definition 3.10 A knowledge base $\mathcal{K}$ is **(finitely) Θ-coverable** if for any (finite) model $\mathcal{I}$ of $\mathcal{K}$ and for every positive integer $n \in \mathbb{N}$, there exists a (finite) $(n, \mathsf{ind}(\mathcal{K}))$-locally Θ-forest model $\mathcal{J}$ of $\mathcal{K}$ that *covers* $\mathcal{I}$, *i.e.* any n-neighbourhood from $\mathcal{J}$ can be $\mathsf{ind}(\mathcal{K})$-homomorphically-mapped to $\mathcal{I}$.

We conclude the section by employing Definition 3.10 to define various classes of description logics. We give their definition first, and supplement it afterwards with a bunch of instructive examples.

Definition 3.11 A description logic $\mathcal{DL}$ is said to be **(finitely) Θ-forest-friendly** if all $\mathcal{DL}$-KBs are (finitely) Θ-coverable. We use $\mathscr{C}_{\Theta\mathsf{fr}}$ and $\mathscr{C}_{\Theta\mathsf{fr}}^{\mathsf{fin}}$ to denote, respectively, the classes of Θ-forest-friendly and finitely Θ-forest-friendly description logics.

The above definition may seem to be artificial as it relies on a somehow ad-hoc notion of coverability. What actually matters when reasoning about logics that are forest-friendly, is the following key property.

Lemma 3.12 For any description logic $\mathcal{DL}$ from $\mathscr{C}^{(\mathsf{fin})}_{\Theta\mathsf{fr}}$, any $\mathcal{DL}$-KB $\mathcal{K}$ and any UCQ $q := \bigvee_{i=1}^{m} q_i$, $\mathcal{K} \not\models_{(\text{fin})} q$ implies the existence of a (finite) $(|q|, \mathsf{ind}(\mathcal{K}))$-locally Θ-forest countermodel for $\mathcal{K}$ and q.

Proof. Let $\mathcal{I}$ be a (finite) countermodel for $\mathcal{K}$ and q. By the fact that $\mathcal{K}$ is (finitely) Θ-coverable we infer the existence of a (finite) $(|q|, \mathsf{ind}(\mathcal{K}), \Theta)$-forest model for $\mathcal{K}$ that covers $\mathcal{I}$, which will be the desired $(|q|, \mathsf{ind}(\mathcal{K}), \Theta)$-forest countermodel $\mathcal{J}$ for $\mathcal{K}$ and q. By contraposition, suppose that $\mathcal{J}$ satisfies q. Then $\mathcal{J} \models_{\pi} q_i$ holds for some $1 \leq i \leq m$ and some match π. Observe that the connected components of $\mathcal{J}\!\upharpoonright_{\{\pi(x) \mid x \in \mathrm{Var}(q_i)\}}$ are of size at most $|q|$. Hence, they can be homomorphically mapped to $\mathcal{I}$ by assumption. This implies $\mathcal{I} \models q_i$, which results in $\mathcal{I} \models q$. $\square$

We now turn our attention to examples of logics that are forest-friendly. We are also going to dedicate Chapter 5 to sufficient conditions for description logics to be (finitely) forest-friendly, which can serve in the future as a tool for generating examples of forest-friendly logics. We say that a description logic $\mathcal{DL}$ has the Θ-*forest countermodel property* if for any $\mathcal{DL}$-KB $\mathcal{K}$ and any PEQ q whenever $\mathcal{K} \not\models q$ holds there exists a Θ-forest model of $\mathcal{K}$ that violates q. By adapting existing proofs of Lutz [Lut08b, p. 8], we can show that the logics $\mathcal{ALC}\Theta$ have the aforementioned property and thus they belong to $\mathscr{C}_{\Theta\mathsf{fr}}$. Interestingly, they also belong to the class $\mathscr{C}^{\mathsf{fin}}_{\Theta\mathsf{fr}}$. Another prominent example is the logic $\mathcal{ALCIQ}$, an extension of $\mathcal{ALCI}$ with counting. One can show that $\mathcal{ALCIQ}$ has the $\{\mathsf{I}\}$-forest countermodel property, for instance by employing a suitable notion of unravelling [BHLS17, p. 64]. However, as soon as we stick to finite models, the $\{\mathsf{I}\}$-forest (counter)model property is lost (*e.g.* consider an ontology defining an infinite path composed of functional and backwards-functional relations). In the later part of the thesis we establish that $\mathcal{ALCIQ}$ is finitely $\{\mathcal{I}\}$-forest-friendly, so some form of forest-likeness can be regained in the finite. The role of "locality" is crucial here. There are also description logics for which only the finite model semantics makes sense. This includes the logic $\mathcal{ALCSCC}$ extended with ERCBoxes [BBR20], *i.e.* positive boolean combinations of linear inequalities over the domain (such constraints trivialise outside the realm of finite models). We will see that this logic is finitely $\emptyset$-forest-friendly. Finally, let us mention that extending $\mathcal{ALC}$ with nominals, transitivity, self-loops, or complex role inclusions spoil its forest-friendliness.

4

Revisiting Lutz's Spoiler Technique

Contents

Motivation and Our Contribution

The *spoiler technique* by Carsten Lutz [Lut08b] is a classical technique that given a $\mathcal{DL}$-knowledge-base $\mathcal{K}$ and a conjunctive query q, reduces the entailment problem (does $\mathcal{K} \models q$ hold?) to the satisfiability problem of a sequence of carefully crafted $\mathcal{DL}^{\cap}$-knowledge-bases (where $\mathcal{DL}^{\cap}$ extends $\mathcal{DL}$ with conjunctions of roles). The spoiler technique was originally used by Lutz [Lut08b, Thm. 1] to establish ExpTime-completeness of conjunctive query entailment for $\mathcal{SHQ}$ (for queries forbidding transitive roles), but some of its variations were also used for the query entailment problem for $\mathcal{SHIQ}$ [GLHS08, Thm. 32] and $\mathcal{SHOQ}$ [GHS08, Thm. 9], for the query entailment over temporal and probabilistic knowledge bases [Koo19, BBL15], as well as for the query non-emptiness problem [BBLW16, Thm. 16]. In this chapter we revisit the spoiler technique and propose a meta-algorithm solving the query entailment problem for certain classes of description logics. To do so, we closely follow the original work of Lutz [Lut08b, p. 3—8], aiming to reuse as much material from his work as possible. While from the bird's eye our proof scheme and Lutz's one are roughly the same, we adjust and improve his technique in several ways stated below.

(i) We propose a meta-algorithm (instead of a single algorithm) that is additionally parametrized by a set of features $\Theta \subseteq \{\mathsf{I}, \mathsf{Self}\}$, and thus we cover a plethora of extensions of $\mathcal{ALC}$ with a *single* proof.

(ii) Our proofs and a meta-algorithm are logic-independent (in contrast to all previous works) and rely on natural model-theoretic notions from Section 3.3. In particular, this means that our meta-algorithm can be applied to any (finitely) Θ-forest-friendly description logic, making it useful as a black box for future use for freshly defined logics.

(iii) In contrast to the original work of Lutz [Lut08b], our meta-algorithm is applicable to the entailment problem of unions of conjunctive queries (not just conjunctive queries), and can be used to infer complexity bounds in terms of data complexity (not only the usual combined complexity).

(iv) Our meta-algorithm can be employed for reasoning in the finite model semantics. To the best of our knowledge — excluding our works and logics that are finitely controllable — there are *no* work that employs the spoiler technique and which can be applied to the finite model setting.

As a prominent application of our meta-algorithm, the next chapter will provide a complete picture of the complexity of the query entailment problem – for (unions of) conjunctive queries – over various description logics extending $\mathcal{ALC}$ and contained in $\mathcal{ZOIQ}$ that do not simultaneously employ the features $\mathcal{I}$, $\mathcal{O}$, and $\mathcal{Q}$. Be cautious however. Despite being worst-case optimal, our algorithm is just complexity-theoretic, and thus completely useless in practice. For a practical algorithm a good idea would be to investigate further the so-called "knots technique" of Eiter, Ortiz, and Šimkus [EOv12] (see also works of Eiter et. al for a gentle introduction to the topic [EOv08, ELOv09]).

Overview of the Chapter and Prerequisites

We assume familiarity with Chapter 3 and its prerequisites. Starting from an informal explanation of Lutz's spoiler technique (Section 4.1), the following sections describe the required components: fork rewritings (Section 4.2), splittings (Section 4.3), spoilers (Section 4.4), and finally the meta-algorithm announced in the "motivations" (Section 4.6). The reader may find the forthcoming content quite technical.

4.1 An Informal Explanation of Lutz's Spoiler Technique

We start by giving a rather informal explanation of *Lutz's spoiler technique*, dedicated to the readers who are not familiar with the original work of Lutz on querying $\mathcal{ALCHQ}$ [Lut08b, Sec. 3].[1] Most of the forthcoming notions are very similar to those from the work of Lutz [Lut08b, Sec. 3] and actually we aimed at reusing as much material from his work as possible. However, many of our statements require separate proofs in order to make them logic-independent and adjustable to Θ-forest-friendly DLs.

Our goal is to decide, given a (finitely) Θ-coverable $\mathcal{DL}$-KB $\mathcal{K}$ and a conjunctive query q, whether $\mathcal{K} \models_{(\text{fin})} q$ holds, which boils down to checking if there is a (finite or arbitrary, depending on the problem) countermodel for $\mathcal{K}$ and q. Due to Lemma 3.12 we can restrict our attention to $(|q|, \mathsf{ind}(\mathcal{K}), \Theta)$-forests. An important observation is that a match π of q over an $(|q|, \mathsf{ind}(\mathcal{K}), \Theta)$-forest $\mathcal{I}$ induces a very specific partition of $\mathrm{Var}(q)$, namely π divides the variables of q into three disjoint categories: (i) the variables mapped to the N-named elements of $\mathcal{I}$, (ii) the variables forming Θ-subtrees "dangling" from some of the N-named elements of $\mathcal{I}$ and (iii) the variables forming Θ-trees that lie "far" from N-named elements. The notion of a *splitting* abstractly describes such a partition, independently of the choice of π and $\mathcal{I}$. The existence of a splitting *compatible* with a $(|q|, \mathsf{ind}(\mathcal{K}), \Theta)$-forest $\mathcal{I}$ implies that $\mathcal{I} \models q$ holds and vice versa. Hence, to establish $\mathcal{K} \not\models_{(\text{fin})} q$, it suffices to find a (finite) $(|q|, \mathsf{ind}(\mathcal{K}), \Theta)$-forest model $\mathcal{I}^{\lightning}$ of $\mathcal{K}$ such that no splitting is compatible with it, or, in other words, $\mathcal{I}^{\lightning}$ that *spoils* all the splittings. To do so, for a splitting Π_q of the query q we design a $\mathcal{DL}$-KB $\mathcal{K}^{\lightning}_{\Pi_q}$, called a *spoiler* for Π_q, with the intended meaning that every $(|q|, \mathsf{ind}(\mathcal{K}))$-locally Θ-forest-like model of $\mathcal{K} \cup \mathcal{K}^{\lightning}_{\Pi_q}$ *spoils* its compatibility with Π_q. The construction of spoilers employs, among other ingredients, the well-known *rolling-up technique* [HT00, Sec. 4], already introduced in Section 3.2, useful to detect Θ-tree-shaped query matches from points (ii)–(iii) above. For feasibility of the "rolling-up" technique, we additionally require that $\mathcal{DL}$ extends $\mathcal{ALC}\Theta^{\cap}$, *i.e.* for every $\mathcal{ALC}\Theta^{\cap}$-concept one can compute in polynomial time (at most polynomially larger) equivalent concept in $\mathcal{DL}$. Having the splittings defined, we prove that (finite) $(|q|, \mathsf{ind}(\mathcal{K}), \Theta)$-forest models of $\mathcal{K} \cup \bigcup_{\Pi_q} \mathcal{K}^{\lightning}_{\Pi_q}$ are also (finite) countermodels for $\mathcal{K}$ and q. This establishes a Turing-reduction from the query entailment problem to satisfiability: the query entailment problem can be solved by doubly-exponentially many (w.r.t the sizes of $\mathcal{K}$ and q) satisfiability checks of exponentially larger (w.r.t the sizes of $\mathcal{K}$ and q) $\mathcal{DL}$-KBs.

The presented algorithm has optimal (doubly-exponential) worst-case time complexity, for description logics with ExpTime-complete satisfiability problem that involve inverses [Lut08a, Thm. 2] or self-loops [BR22, Thm. 8.2] (consult Chapter 6 for our 2ExpTime-hardness proof for querying $\mathcal{ALC}^{\mathsf{Self}}$). However, in the case of $\emptyset$-forest-friendly description logics the above methods yield a non-optimal complexity, *e.g.* entailment of conjunctive queries over $\mathcal{ALC}$-KBs is "only" ExpTime-complete [EOv12, Cor. 3]. To get the optimal (exponential) upper bound in such a case, we parallelise the construction of $\bigcup_{\Pi_q} \mathcal{K}^{\lightning}_{\Pi_q}$.

[1]Lutz works with $\mathcal{SHQ}$, an extension of $\mathcal{ALCHQ}$ with transitive roles, but he does not allow for transitive roles in queries. This is crucial since their presence makes the CQ entailment problem exponentially harder [ELOv09, Thm. 1]. Hence, Lutz's work is more about querying $\mathcal{ALCHQ}$ than $\mathcal{SHQ}$.

This means, intuitively, that the KB $\bigcup_{\Pi_q} \mathcal{K}^{\text{ϟ}}_{\Pi_q}$ is going to be divided into exponentially many chunks called *super-spoilers* $\mathcal{K}^{\text{ϟ}\star}_q$ with the meaning that $\mathcal{K} \not\models_{(\text{fin})} q$ if and only if $\mathcal{K} \cup \mathcal{K}^{\text{ϟ}\star}_q$ has a (finite) $(|q|, \mathsf{ind}(\mathcal{K}), \emptyset)$-forest model for some super-spoiler $\mathcal{K}^{\text{ϟ}\star}_q$. We then show that each super-spoiler is only of polynomial size and that the set of super-spoilers can be enumerated in a single exponential time. This provides a Turing reduction from the (finite) query entailment problem to exponentially many (finite) satisfiability checks of polynomial-size $\mathcal{DL}$-KBs, which yields an optimal complexity in the case of $\emptyset$-forest-friendly DLs.

4.2 Step I: Rolling-Up Concepts and Fork Rewritings

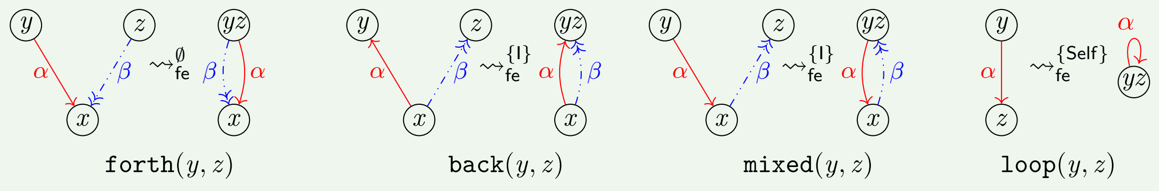
Let us recall that in Section 3.2 we established a tight correspondence between Θ-tree-shaped conjunctive queries and $\mathcal{ALC}\Theta^{\cap}$-concepts. Informally, for a given Θ-tree-shaped conjunctive query q, we can employ Corollary 3.7 to produce an $\mathcal{ALC}\Theta^{\cap}$-concept Match_q for which any element in the interpretation of Match_q "detects" a possible match of q. Hence, relying on Corollary 3.7 we can design an algorithm for the entailment problem for Θ-tree-shaped queries: for a given input query q and an input knowledge base $\mathcal{K}$, it suffices to check whether $\mathcal{K} \cup \{\top \sqsubseteq \neg\text{Match}_q\}$ is satisfiable. Unfortunately, the presented method of detecting query matches works only for Θ-tree-shaped queries. To detect matches of arbitrary CQs, we require a stronger match-detection mechanism, namely the notions of fork rewritings and splittings.

A conjunctive query can induce many different query matches, possibly of arbitrary shapes. In our case, however, query matches are of a very specific form due to the local-forest-likeness of target structures. For instance, suppose that a query q contains a *fork*, that is a subquery of the form $r(z, x) \wedge s(y, x)$, and that the query q is satisfied in a $\emptyset$-tree $\mathcal{I}$ (as witnessed by some match π). As every non-root element in a tree has a unique parent, this clearly implies that the variables z and y are mapped via π to the same element, and hence, can be "considered equal". A similar situation occurs for $\{\mathsf{I}\}$-trees and forks involving inverted roles. Finally, in the case when self-loops are allowed in trees, it is possible for a query atom $r(z, x)$ to collapse into a self-loop, *i.e.* z and x are identified by a query match. We formalise this intuition with the forthcoming notion of fork rewritings [Lut08b, p. 4], intended to remove this kind of "redundancy" from queries.

Definition 4.1 Given CQs q and q', we say that the q' is obtained from q by **Θ-fork elimination** (notation: $q \rightsquigarrow^{\Theta}_{\mathsf{fe}} q'$) if q' is obtained by identifying some pair of variables y and z in q for which:

- $[\texttt{forth}(y, z)]$: there exist some atoms $\alpha(y, x)$ and $\beta(z, x)$ in q,
- $[\texttt{back}(y, z)]$: $\mathsf{I} \in \Theta$ and there exist some atoms $\alpha(x, y)$ and $\beta(x, z)$ in q,
- $[\texttt{mixed}(y, z)]$: $\mathsf{I} \in \Theta$ and there exist some atoms $\alpha(y, x)$ and $\beta(x, z)$ in q, or
- $[\texttt{loop}(y, z)]$: $\mathsf{Self} \in \Theta$ and there exists some atom $\alpha(y, z)$ in q.

The pairs (y, z) satisfying $[\texttt{forth}(y, z)]$ are called *$\emptyset$-forks*, the ones satisfying $[\texttt{back}(y, z)]$ or $[\texttt{mixed}(y, z)]$ are called *$\{\mathsf{I}\}$-forks*, and the pairs satisfying $[\texttt{loop}(y, z)]$ are called *$\{\mathsf{Self}\}$-forks*. We refer to them jointly as **Θ-forks** for a suitable $\Theta \subseteq \{\mathsf{I}, \mathsf{Self}\}$, meaning that θ-forks are also Θ-forks whenever $\theta \subseteq \Theta$.



A CQ q' is called a **Θ-fork rewriting** of q if q' can be obtained from q by applying Θ-fork-elimination of q, possibly multiple times. When the fork elimination process is applied exhaustively on q we say that the resulting query, denoted $\mathsf{maxfr}_{\Theta}(q)$, is a **maximal Θ-fork rewriting** of q.

To gain more intuitions on how fork elimination works, consult Example 4.2. To guide the reader even further, we discuss the notion of maximal fork rewritings in more detail, and present several auxiliary lemmas relating matches of queries and their fork rewritings (see Lemma 4.4 and Corollary 4.8).

Example 4.2. Consider a conjunctive query $q \coloneqq r(x,y) \land r(x,z) \land s(y,v) \land r(v,z) \land \mathrm{A}(x) \land \mathrm{B}(y) \land \mathrm{C}(z) \land \mathrm{D}(v)$. By applying either $[\texttt{forth}(x,v)]$ (for the subquery $r(x,z) \land r(v,z)$) or $[\texttt{mixed}(x,v)]$ (for the subquery $r(x,y) \land s(y,v)$) we obtain the following fork rewriting of q: $r(xv,y) \land s(xv,y) \land r(xv,z) \land \mathrm{B}(y) \land \mathrm{A}(xv) \land \mathrm{D}(xv) \land \mathrm{C}(z)$, where the variable xv is fresh. Consult Figure 4.1. One can further apply $[\texttt{loop}(xv,y)]$ to obtain the query $r(xyv,z) \land r(xyv,xyv) \land s(xyv,xyv) \land \mathrm{B}(xyv) \land \mathrm{A}(xyv) \land \mathrm{D}(xyv) \land \mathrm{C}(z)$ for a fresh variable xyv. Further application of $[\texttt{loop}(xyv,z)]$ yields the maximal $\{\mathsf{I},\mathsf{Self}\}$-fork rewriting of q, namely $\mathsf{maxfr}(q) \coloneqq r(xyzv,xyzv) \land s(xyzv,xyzv) \land \mathrm{B}(xyzv) \land \mathrm{A}(xyzv) \land \mathrm{D}(xyzv) \land \mathrm{C}(xyzv)$ for a fresh variable $xyzv$.

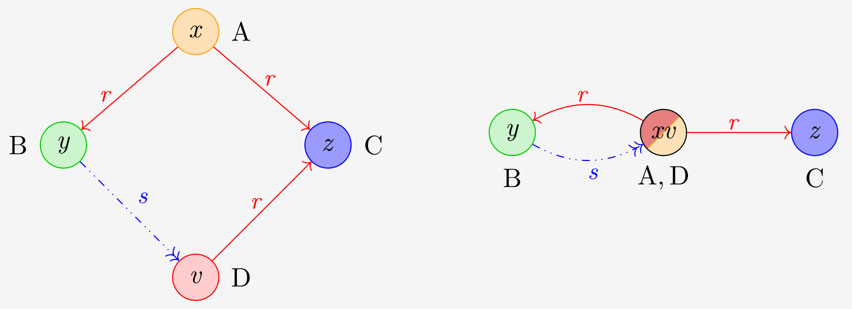


Figure 4.1: An example conjunctive query (left) and one of its fork rewritings (right).

The rest of the section is dedicated to proving useful properties of fork rewritings. First, note that maximal fork rewritings are not really interesting when the Self operator is present in Θ, *i.e.* any connected query maximally rewrites into a "self-loop-shaped" query, as presented in Example 4.2. In stark contrast, maximal $\emptyset$-fork rewritings will play an instrumental role when reasoning about $\emptyset$-forest-friendly logics, allowing us for obtaining tighter complexity bounds. Not spoiling the fun yet, we present an important result of Lutz. In his proof [Lut08b, Appendix A], Lutz employs a handy convention that variables in queries are sets (initially treating variables as singleton sets), and the "identification of variables" x and y in $\emptyset$-fork rewritings is implemented by replacing x and y in the query with their union $x \cup y$.

Lemma 4.3 (Lemma 1 by Lutz [Lut08b]) Every conjunctive query has a unique (up to a variable renaming) maximal $\emptyset$-fork rewriting.

A rather immediate application of Definition 4.1 yields that the entailment of a fork rewriting of a query implies the entailment of the input query itself. The proof goes via a routine induction.

Lemma 4.4 Let q and q' be conjunctive queries, such that q' is obtained from q by Θ-fork rewriting. Moreover, let $\mathcal{I}$ be a structure and π' be a match for q' and $\mathcal{I}$. Then $\mathcal{I} \models q$, and one can compute a match π for q and $\mathcal{I}$ for which the images $\pi[\mathrm{Var}(q)]$ and $\pi'[\mathrm{Var}(q')]$ are equal.

Proof. Suppose $\mathcal{I} \models q'$. Since q' is a Θ-fork rewriting of q, we can find a derivation $q_n \rightsquigarrow^{\Theta}_{\mathsf{fe}} q_{n-1} \rightsquigarrow^{\Theta}_{\mathsf{fe}} \ldots \rightsquigarrow^{\Theta}_{\mathsf{fe}} q_0$ with $q = q_n$ and $q_0 = q'$. Reasoning inductively, it suffices to show that for all indices $0 \leq i < n$ we have that $\mathcal{I} \models q_i$ implies $\mathcal{I} \models q_{i+1}$. Then we conclude the lemma by taking $i \coloneqq (n-1)$. Assume $\mathcal{I} \models q_i$. Then there is a homomorphism $\mathfrak{h}_i \colon \mathcal{I}_{q_i} \to \mathcal{I}$. Since $q_{i+1} \rightsquigarrow^{\Theta}_{\mathsf{fe}} q_i$ holds, we can find the variables x, y, z such that (i) $\mathrm{Var}(q_i) \setminus \{x,y,z\} = \mathrm{Var}(q_{i+1}) \setminus \{x,y,z\}$, and (ii) q_i was obtained from q_{i+1} by replacing each occurrence of x or y in any atoms with z. Hence, let $\mathfrak{f} \colon \mathcal{I}_{q_{i+1}} \to \mathcal{I}_{q_i}$ be a function satisfying $\mathfrak{f}(x) = \mathfrak{f}(y) = z$ and $\mathfrak{f}(v) = v$ for all other variables. From (i) and (ii) we immediately infer that $\mathfrak{f}$ is a homomorphism. Thus $(\mathfrak{f} \circ \mathfrak{h}_i) \colon \mathcal{I}_{q_{i+1}} \to \mathcal{I}$ is a homomorphism, establishing $\mathcal{I} \models q_{i+1}$. The rest of the lemma follows by our construction. □

Note that the reverse direction of Lemma 4.4 does not hold, as witnessed by the example below.

Example 4.5. Consider the query $q \coloneqq r(z,x) \land s(y,x)$ and its $\emptyset$-fork-rewriting $q' \coloneqq r(yz,x) \land s(yz,x)$. Let $\mathcal{I}$ be any structure with $\Delta^{\mathcal{I}} \coloneqq \{x,y,z\}$, $r^{\mathcal{I}} \coloneqq \{(z,x)\}$, and $s^{\mathcal{I}} \coloneqq \{(y,x)\}$. Clearly $\mathcal{I} \models q$ but $\mathcal{I} \not\models q'$.

We next link Θ-trees, Θ-tree-shaped queries, and Θ-fork-rewritings in the following two lemmas.

Lemma 4.6 Let $\mathsf{I} \notin \Theta$, q be a *connected* conjunctive query, $\mathcal{I}$ be a structure, and π be a match for q and $\mathcal{I}$ such that (i) $\pi[q]$ is a Θ-tree, and (ii) there are no Θ-forks (x, y) in q satisfying $\pi(x) = \pi(y)$. Then q is Θ-tree-shaped.

Proof. We[2] proceed inductively w.r.t. $(2^{\mathrm{Var}(q)} \setminus \{\emptyset\}, \subseteq)$, where the inductive assumption states that for all non-empty sets $\mathrm{V} \in 2^{\mathrm{Var}(q)}$ for which $q{\restriction_{\mathrm{V}}}$ is connected, we have that $q{\restriction_{\mathrm{V}}}$ is Θ-tree-shaped. Then the statement of the lemma follows. For the base case, consider any singleton V for which $q{\restriction_{\mathrm{V}}}$ contains at least one atom. Then $\pi[q{\restriction_{\mathrm{V}}}]$ is isomorphic to $q{\restriction_{\mathrm{V}}}$, and thus is a Θ-tree by the assumption. Hence, $q{\restriction_{\mathrm{V}}}$ is Θ-tree-shaped. For the inductive step, suppose V has the form $\mathrm{V}' \cup \{v\}$ for some v. For brevity, we say that a variable x *points to* a variable y if some atom of the form $\alpha(x, y)$ belongs to $q{\restriction_{\mathrm{V}}}$. Suppose that $q{\restriction_{\mathrm{V}}}$ is connected. This implies the existence of a partition $\mathrm{V}_1, \mathrm{V}_2, \ldots, \mathrm{V}_n$ of variables from V', such that (i) for all $i \leq n$ the query $q{\restriction_{\mathrm{V}_i}}$ is connected (thus Θ-tree-shaped by the inductive assumption), (ii) for all $i \neq j$ there are no variable from V_i that points to a variable from V_j (*i.e.* these sets induce the “connected components” of $q{\restriction_{\mathrm{V}'}}$), and (iii) for every $i \leq n$ there is $u \in \mathrm{V}_i$ such that u points to v or vice versa. The key property required to establish Θ-tree-shapedness of $q{\restriction_{\mathrm{V}}}$ is that for every index $i \leq n$ there exists *precisely one* variable $u_i \in \mathrm{V}_i$ that points to v or vice versa (and even more specifically, there is at most one index i for which u_i points to v). Observe that:

(A) There are no pair of distinct variables u and w that point to v.
Indeed, as $\pi[q{\restriction_{\mathrm{V}}}]$ is a Θ-tree, this implies that $\pi(w) = \pi(u)$, contradicting the assumption that all such $\emptyset$-forks were eliminated from q.

(B) If v points to $u_i \in \mathrm{V}_i$, then u_i is the root variable of $q{\restriction_{\mathrm{V}_i}}$.
Indeed, otherwise we would have a $\emptyset$-fork involving u_i, v and the parent w of u_i. This, together with the fact $\pi[q{\restriction_{\mathrm{V}}}]$ is a Θ-tree, yields $\pi(w) = \pi(v)$. Similarly to the previous case, this contradicts the fact that all such $\emptyset$-forks were eliminated from q.

(C) For all $i \leq n$ there are no (possibly equal) $w, u \in \mathrm{V}_i$ such that w points to v and v points to u.
Suppose such variables w and u exist. Then, by the previous observation, u would be the root variable of $q{\restriction_{\mathrm{V}_i}}$. Hence, there exists a directed cycle ρ in $q{\restriction_{\mathrm{V}_i \cup \{v\}}}$ that contains the variables v, u, and w. If Θ is empty, we get a contradiction with the acyclicity of $\pi[q]$. Otherwise, whenever $\Theta = \{\mathsf{Self}\}$, the image of ρ via π collapses into a self-loop. This implies $\pi(u) = \pi(v)$ and contradicts the fact that $\{\mathsf{Self}\}$-forks are not present in q.

We next sketch the proof that $q{\restriction_{\mathrm{V}}}$ is Θ-tree-shaped. First, let u_i, for each $i \leq n$, be the *unique* variable from V_i that either points to v or v points to it (note that both cases cannot happen simultaneously by the third observation, and that uniqueness of u_i is guaranteed by the combination of the three above observations). Second, let S be the set of all indices i for which v points to u_i. Then the query q restricted to $\{v\} \cup \bigcup_{s \in \mathrm{S}} \mathrm{V}_s$ is Θ-tree-shaped: its root is v and it has all of u_s “assigned” as children (here we rely on the fact that the sets V_s are pairwise-disjoint and there are no atoms employing variables simultaneously from V_s and $\mathrm{V}_{s'}$ for $s \neq s'$). If S contains all the indices from $\{1, 2, \ldots, n\}$ then we are done. Otherwise there is a *unique* (by the first observation) index $1 \leq j \leq n$ that is not in S and for which u_j points to v. It is not too difficult to see that the query $q{\restriction_{\mathrm{V}}}$ is also Θ-tree-shaped: the query $q{\restriction_{\mathrm{V}_j}}$ is a Θ-tree by the inductive assumption and the tree rooted to v simply becomes a subtree of u_j. ☐

We next establish the proof of the analogous statement for the case of $\mathsf{I} \in \Theta$.

Lemma 4.7 Let $\mathsf{I} \in \Theta$, q be a *connected* conjunctive query, $\mathcal{I}$ be a structure, and π be a match for q and $\mathcal{I}$ such that (i) $\pi[q]$ is a Θ-tree, and (ii) there are no Θ-forks (x, y) in q satisfying $\pi(x) = \pi(y)$. Then q is Θ-tree-shaped.

[2] I thank Jan Otop for suggesting this simple, inductive proof-approach.

Proof. We argue along the lines of the proof of Lemma 4.7, employing the same inductive assumption and the same naming scheme (thus consult the previous proof when in doubt). The base case is exactly the same as in Lemma 4.6, so we focus on the inductive step only. Recall that our main goal is to show that there is no index i for which there are different $u, w \in \mathrm{V}_i$ pointing to v (or vice versa). Towards a contradiction, assume that such index i and $u, w \in \mathrm{V}_i$ exist. Observe that $\pi(v) \neq \pi(u)$ and $\pi(w) \neq \pi(u)$. Indeed, this follows from the fact that we either eliminated such forks in the case of $\mathsf{Self} \in \Theta$, or $\pi(v)$ would carry a self-loop, contradicting the fact that it is a $\{\mathsf{I}\}$-tree. Next, we show that ($\heartsuit$): if ρ is a path in $q{\restriction}_{\mathrm{V}_i}$ composed of pairwise-different elements, then the elements $\pi(\rho_1), \ldots, \pi(\rho_{|\rho|})$ are also pairwise-different. The proof is via *reductio ad absurdum.* Suppose that there are $i < j$ for which $\pi(\rho_i) = \pi(\rho_j)$. If $j = i{+}1$ we get a contradiction employing the same reasoning as for the proof of $\pi(v) \neq \pi(u)$. If $j = i{+}2$ we get a contradiction from the fact that such $\{\mathsf{I}\}$-forks are not present in q. Finally, if $j > i{+}2$ then we have that $\pi[q]$ contains a cycle of length greater than 2, contradicting its Θ-tree-shapedness. We now employ observation ($\heartsuit$) as follows. Let ρ be any path from u to w in $q{\restriction}_{\mathrm{V}_i}$ that is composed of pairwise-different elements (guaranteed by connectedness of $q{\restriction}_{\mathrm{V}_i}$). Then (by the choice of u and w) $v\rho v$ is a cycle in $q{\restriction}_{\mathrm{V}\cup\{v\}}$. We claim that $\pi(v)$ is pairwise-different from the images of the elements from ρ. Recall that we have already established that $\pi(v) \neq \pi(u)$ and $\pi(v) \neq \pi(w)$. Hence, suppose that there is an index $1 < i < |\rho|$ for which $\pi(v) = \pi(\rho_i)$. This however yields the existence of a cycle of length more than 2 in $\pi[q]$, namely $\pi(\rho_i) \ldots \pi(\rho_{|\rho|})\pi(v)$. A contradiction with the Θ-tree-shapedness of $\pi[q]$. Hence, $\pi(v)$ is pairwise-different from all the images of elements of ρ. Then the image of $v\rho v$ via π is again a long cycle, contradicting Θ-tree-shapedness of $\pi[q]$.

To establish Θ-tree-shapedness of $q{\restriction}_{\mathrm{V}}$ it suffices to show that it does not contain cycles of length greater than 2 (self-loops are handled separately in case $\mathsf{Self} \notin \Theta$ by employing the fact about tree-shapedness of the image of q via π). Recall that we proved that such cycles are present in no subquery of the form $q{\restriction}_{\mathrm{V}_i\cup\{v\}}$. This in turn can be lifted to the general case by employing the fact that there are no atoms involving variables from V_i and V_j for $i \neq j$. Thus $q{\restriction}_{\mathrm{V}}$ is indeed Θ-tree-shaped, which concludes the proof. □

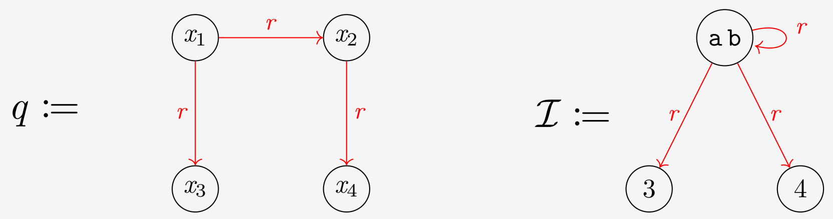
By summarising Lemma 4.6 and Lemma 4.7 we can now conclude Corollary 4.8.

Corollary 4.8

If $\Theta \subseteq \{\mathsf{I}, \mathsf{Self}\}$, q is a *connected* conjunctive query, $\mathcal{I}$ is a structure, and π is a match for q and $\mathcal{I}$ such that the image of q via π is a Θ-tree then the Θ-fork-rewriting q', obtained by eliminating all Θ-forks (x, y) from q satisfying $\pi(x) = \pi(y)$, is Θ-tree-shaped.

It would be tempting to generalise Lemma 4.6 and Lemma 4.7 to the case of N-rooted Θ-forests. Such a generalisation would be, however, *false* as witnessed by the following example.

Remark 4.9. Consider the query q and the $\{\mathtt{a}, \mathtt{b}\}$-rooted $\emptyset$-forest $\mathcal{I}$ depicted below. Clearly $\mathcal{I} \models q$ holds, as witnessed by the match $\pi := \{x_1 \mapsto \mathtt{a}^{\mathcal{I}}, x_2 \mapsto \mathtt{b}^{\mathcal{I}}, x_3 \mapsto 3, x_4 \mapsto 4\}$. In particular, $\mathcal{I}_q$ is not an $\{\mathtt{a}, \mathtt{b}\}$-rooted $\emptyset$-forest (after interpreting $\mathtt{a}$ as x_1 and $\mathtt{b}$ as x_2) while its image via π is an $\{\mathtt{a}, \mathtt{b}\}$-rooted $\emptyset$-forest. However, the variables x_1 and x_2 cannot be identified by a $\emptyset$-fork, even though $\pi(x_1) = \pi(x_2)$.



We conclude by providing an upper bound on the total number of queries produces by Θ-fork rewritings.

Lemma 4.10 Let q be a conjunctive query with m variables, and let $\mathsf{ForkRewrs}_{\Theta}(q)$ denote the set of all Θ-fork rewritings of q. Then $\mathsf{ForkRewrs}_{\Theta}(q)$ contains at most m^m queries, and the elements of $\mathsf{ForkRewrs}_{\Theta}(q)$ can be enumerated in time exponential w.r.t $|q|$.

Proof sketch. Under the naming convention of Lutz, mentioned shortly before Lemma 4.3, it becomes apparent that every Θ-fork rewriting of q can be identified with a partition of variables from $\mathrm{Var}(q)$ (but not necessarily vice versa). Thus, the size of $\mathsf{ForkRewrs}_{\Theta}(q)$ is clearly bounded by the m-th Bell number, and hence by m^m. For the algorithm enumerating Θ-fork rewritings, we proceed as follows. We first fix an auxiliary enumeration for variables $x_0, x_1, \ldots, x_{m-1}$ from q. Note that the fork elimination can be applied to q at most $(m-1)$ times. Thus any Θ-fork rewriting q_k of q that is obtained with a derivation $q \rightsquigarrow^{\Theta}_{\mathsf{fe}} q_2 \rightsquigarrow^{\Theta}_{\mathsf{fe}} \ldots \rightsquigarrow^{\Theta}_{\mathsf{fe}} q_k$ can be represented as a sequence $(i_1, j_1), (i_2, j_2), \ldots, (i_{k-1}, j_{k-1})$ where each pair (i_ℓ, j_ℓ) represents the indices of variables (integers between 0 and $(m-1)$) of the Θ-fork (x_{i_ℓ}, x_{j_ℓ}) that were identified in the ℓ-th Θ-fork elimination. Hence, it suffices to first iterate through the possible values of k (from 0 to $m-1$), then iterate through all possible sequences of length k of pairs of indices $(i_1, j_1), (i_2, j_2), \ldots, (i_k, j_k)$, and then eliminate (if possible) the corresponding Θ-forks. $\square$

4.3 Step II: Splittings

The following notion of splittings [Lut08b, p. 4] provides an abstraction of how a conjunctive query q matches a $(|q|, \mathsf{N}, \Theta)$-forest, while referring neither to a concrete interpretation nor to a concrete match. Intuitively, the role of splittings is to partition the variables of some fork rewriting q of the input query, depending on the three possible scenarios: (i) either a variable v is expected to be mapped to one of the N-named elements, or (ii) v, together with some other variables, are expected to be mapped such that they form a Θ-subtree dangling from one of the N-named elements, or (iii) v is expected to be mapped somewhere "further down" inside the structure, being disconnected from the N-named elements.

Definition 4.11 Fix $\mathsf{N} \subseteq \mathbf{N_I}$, $\Theta \subseteq \{\mathcal{I}, \mathsf{Self}\}$, and a CQ q. An **(N, Θ)-splitting** Π^{N}_q of q is a tuple

$$\Pi^{\mathsf{N}}_q \coloneqq (\mathtt{Roots}, \mathtt{name}, \mathtt{SubTree}_1, \mathtt{SubTree}_2, \ldots, \mathtt{SubTree}_n, \mathtt{root\text{-}of}, \mathtt{Trees}),$$

where the sets $\mathtt{Roots}, \mathtt{SubTree}_1, \ldots, \mathtt{SubTree}_n, \mathtt{Trees}$ induce a partition of $\mathrm{Var}(q)$, $\mathtt{name} \colon \mathtt{Roots} \to \mathsf{N}$ is a function naming the roots, and $\mathtt{root\text{-}of} \colon \{1, 2, \ldots, n\} \to \mathtt{Roots}$ assigns to each $\mathtt{SubTree}_i$ an element from $\mathtt{Roots}$. Moreover, to be an (N, Θ)-splitting, Π^{N}_q has to satisfy all the conditions below:

(a) The query $q{\restriction}_{\mathtt{Trees}}$ is a conjunction of variable-disjoint Θ-tree-shaped queries.

(b) The queries $q{\restriction}_{\mathtt{SubTree}_i}$ are Θ-tree-shaped for all indices $i \in \{1, 2, \ldots, n\}$.

(c) For any atom $\alpha(x, y) \in q$ we have that the variables x, y belong to the same set or there exists an index $i \in \{1, 2, \ldots, n\}$ for which either:

  (i) $\mathtt{root\text{-}of}(i) = x$, $x \in \mathtt{Roots}$, $y \in \mathtt{SubTree}_i$, and y is the root of $q{\restriction}_{\mathtt{SubTree}_i}$,

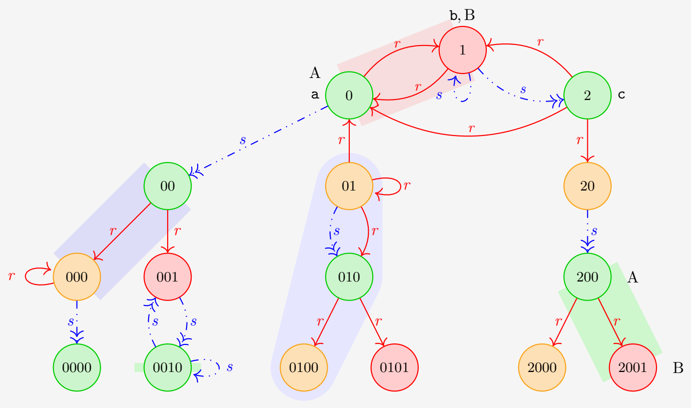
  (ii) $\mathtt{root\text{-}of}(i) = y$, $y \in \mathtt{Roots}$, $x \in \mathtt{SubTree}_i$, and x is the root of $q{\restriction}_{\mathtt{SubTree}_i}$.

(d) For every $1 \leq i \leq n$, taking x_i as the root of $q{\restriction}_{\mathtt{SubTree}_i}$, we have that either (i) there is an atom $r(\mathtt{root\text{-}of}(i), x_i)$ in q, or (ii) $\mathsf{I} \in \Theta$ and there is an atom $r(x_i, \mathtt{root\text{-}of}(i))$ in q.

It may help to think that a splitting represents an abstraction of an image of the query in a target structure, consisting of named roots, corresponding to the ABox part of the model, together with some of their subtrees and of some auxiliary Θ-trees lying somewhere detached from the roots. Consult Example 4.12.

Example 4.12. Consider an $\{\mathsf{a},\mathsf{b},\mathsf{c}\}$-rooted $\{\mathsf{I},\mathsf{Self}\}$-forest $\mathcal{I}$ depicted below and a (non-tree-shaped) CQ q:

$$q \coloneqq (\mathrm{A}(x_0) \wedge r(x_0, x_1) \wedge r(x_1, x_0) \wedge \mathrm{B}(x_1)) \wedge (s(x_0, x_{00}) \wedge r(x_{00}, x_{000}) \wedge r(x_{000}, x_{000})) \\ \wedge (r(x_{01}, x_0) \wedge s(x_{01}, x_{010}) \wedge r(x_{010}, x_{0100})) \wedge (\mathrm{A}(x_{200}) \wedge r(x_{200}, x_{2001}) \wedge \mathrm{B}(x_{2001}) \wedge s(x_{0010}, x_{0010}))\,.$$



$\texttt{Roots} \coloneqq \{x_0, x_1\}$

$\texttt{SubTree}_1 \coloneqq \{x_{00}, x_{000}\}$

$\texttt{SubTree}_2 \coloneqq \{x_{01}, x_{010}, x_{0100}\}$

$\texttt{Trees} \coloneqq \{x_{200}, x_{2001}, x_{0010}\}$

$\texttt{name}(x_0) \coloneqq \mathsf{a}, \texttt{name}(x_1) \coloneqq \mathsf{b}$

$\texttt{root-of}(1) \coloneqq x_0, \texttt{root-of}(2) \coloneqq x_0$

Note that $\mathcal{I}$ (depicted on the left) satisfies q, which is witnessed by a match $\pi \colon x_i \mapsto i$ (this match is depicted by the highlighted areas of the picture). Now consider the $(\{\mathsf{a},\mathsf{b},\mathsf{c}\}, \{\mathsf{I},\mathsf{Self}\})$-splitting $\Pi_q^{\{\mathsf{a},\mathsf{b},\mathsf{c}\}} \coloneqq \big(\texttt{Roots}, \texttt{name}, \texttt{SubTree}_1, \texttt{SubTree}_2, \texttt{root-of}, \texttt{Trees}\big)$ defined on the right-hand side of the picture. We encourage the reader to consult Definition 4.11 and check that this is indeed a splitting (the highlighted areas correspond to its components). Moreover, such a splitting is compatible with $\mathcal{I}$ (see: Definition 4.14).

Similarly to Lemma 4.10, we would like to bound the total number of (N, Θ)-splittings of a given conjunctive query and sketch a worst-case optimal algorithm for their enumeration.

Lemma 4.13 Let q be an m-variable CQ and N be a finite set of names. Then $\mathsf{Splts}_{\mathsf{N},\Theta}(q)$, the set of all (N, Θ)-splittings of q, has size at most $m^{2m} \cdot |\mathsf{N}|^m$. Moreover, its members can be enumerated in time $\mathfrak{e}(|q|, |\mathsf{N}|)$ for some exponential function $\mathfrak{e}$ that becomes polynomial for a *fixed* query q.

Proof sketch. Recall that splittings are tuples of the form $(\texttt{Roots}, \texttt{name}, \texttt{SubTree}_1, \ldots, \texttt{SubTree}_n, \texttt{root-of}, \texttt{Trees})$ for some integer $0 \leq n \leq |\mathrm{Var}(q)|$. As an over-approximation, we view the function $\texttt{name}$ as a function of type $\mathrm{Var}(q) \to \mathsf{N}$, and the function $\texttt{root-of}$ as a function of type $\mathrm{Var}(q) \to \mathrm{Var}(q)$. Clearly, there are $|\mathsf{N}|^m$ and m^m of such functions. The remaining components for a splitting induce a partition of variables of q, and hence their number can be bounded by the m-th Bell number, and consequently by m^m. The desired bound from the statement of the lemma follows now by the multiplication principle.

The enumeration of candidates for splittings from $\mathsf{Splts}_{\mathsf{N},\Theta}(q)$ can be done by: (A) enumerating all possible subsets of $\mathrm{Var}(q)$ to obtain $\texttt{Roots}$, then (B) enumerating[3] all functions from N to $\texttt{Roots}$, (C) enumerating possible values n from 0 to $|\mathrm{Var}(q) \setminus \texttt{Roots}|$, (D) enumerating all n-partitions of $(\mathrm{Var}(q) \setminus \texttt{Roots})$, (E) enumerating all assignments $\{1, 2, \ldots, n\} \to \texttt{Roots}$ and all assignments $\{1, 2, \ldots, n\} \to (\mathrm{Var}(q) \setminus \texttt{Roots})$ to find the "root variable" of each component. The steps (A), (D), and (E) can be implemented to work in time exponential w.r.t $|q|$, the step (C) works in time polynomial w.r.t $|q|$, while the step (B) works in time exponential w.r.t $\log_2(|\mathsf{N}|) \cdot |\texttt{Roots}|$, and thus exponential w.r.t $|q| + |\mathsf{N}|$ (but polynomial w.r.t $|\mathsf{N}|$ if q is fixed). Given a candidate $\Pi_q^{\mathsf{N}} \coloneqq (\texttt{Roots}, \texttt{name}, \texttt{SubTree}_1, \ldots, \texttt{SubTree}_n, \texttt{root-of}, \texttt{Trees})$ for a splitting, testing conditions (a) and (b) of Definition 4.11 can be done in time polynomial w.r.t $|\Pi_q^{\mathsf{N}}|$ with a classical use of the DFS algorithm. Verification of the two other conditions of Definition 4.11 can be done by analysing the shape of the query q (again, implementable to work in time polynomial w.r.t $|\Pi_q^{\mathsf{N}}|$). This concludes the design of our algorithm. Its running time is simply the product of the running times of each enumeration and verification step. □

[3]This can be achieved, *e.g.* by enumerating all bitstrings of length $\log_2(|\mathsf{N}|) \cdot |\texttt{Roots}|$.

We finish the section by showing that splittings indeed fulfil their purposes. In order to do it, we first introduce an intermediate definition of *compatibility* of a splitting with a given $(|q|, \mathsf{N}, \Theta)$-forest.

Definition 4.14 Let $\mathsf{N} \subseteq \mathbf{N_I}$ be a set of names, q be a conjunctive query, and $\mathcal{I}$ be a $(|q|, \mathsf{N}, \Theta)$-forest. We say that an (N, Θ)-splitting Π_q^{N} of q is **compatible with** $\mathcal{I}$ if all the conditions below are satisfied.

(A) For each component $\hat{q}$ of $q\!\restriction_{\mathtt{Trees}}$ there exists a d in $(\mathrm{Match}_{\hat{q}})^{\mathcal{I}}$. (see: Section 3.2.)

(B) For all atoms $\mathrm{A}(x) \in q$ with $x \in \mathtt{Roots}$ we have $(\mathtt{name}(x))^{\mathcal{I}} \in \mathrm{A}^{\mathcal{I}}$.

(C) For all atoms $\alpha(x, y) \in q$ with $x, y \in \mathtt{Roots}$ we have $\big(\mathtt{name}(x)^{\mathcal{I}}, \mathtt{name}(y)^{\mathcal{I}}\big) \in \alpha^{\mathcal{I}}$.

(D) For all indices $i \in \{1, 2, \ldots, n\}$ and x_i being the root of $q\!\restriction_{\mathtt{SubTree}_i}$, the following equation holds:

$$\mathtt{name}(\mathtt{root\text{-}of}(i))^{\mathcal{I}} \in \left(\exists \left(\bigcap_{r(\mathtt{root\text{-}of}(i),x_i)\in q} r \quad \cap \bigcap_{r(x_i,\mathtt{root\text{-}of}(i))\in q \text{ only if } \mathcal{I}\in\Theta} r^-\right).\mathrm{Match}_{q\restriction_{\mathtt{SubTree}_i}}\right)^{\mathcal{I}}.$$

We next present lemmas that link together all the notions presented so far. The main goal here is to establish the equivalence between the satisfaction of a query and the existence of a compatible splitting. The forthcoming proofs are quite delicate and rely on exhaustive case analysis. Most of the time this becomes relatively tedious, so we welcome the reader to skip the forthcoming proofs for the first-time reading.

Lemma 4.15 Let q, N, Θ, and $\mathcal{I}$ be as in Definition 4.14. Suppose that there is a Θ-fork rewriting q' of q and an (N, Θ)-splitting $\Pi_{q'}^{\mathsf{N}}$ of q', such that $\Pi_{q'}^{\mathsf{N}}$ is compatible with $\mathcal{I}$. Then $\mathcal{I} \models q$.

Proof. By Lemma 4.4 it suffices to prove $\mathcal{I} \models q'$. We construct $\mathfrak{h}\colon \mathrm{Var}(q') \to \Delta^{\mathcal{I}}$ as follows:

- For every root variable $x \in \mathtt{Roots}$ we put $\mathfrak{h}(x) := (\mathtt{name}(x))^{\mathcal{I}}$.
- Fix an index $1 \le i \le n$. By Item (b) of Definition 4.11 we know that the query $q'\!\restriction_{\mathtt{SubTree}_i}$ is Θ-tree-shaped. Let x_i be its root. Moreover, by Item (D) of Definition 4.14 there exists an element $\mathrm{d}_i \in \Delta^{\mathcal{I}}$ that belongs to the interpretation of $\mathrm{Match}_{q'\restriction_{\mathtt{SubTree}_i}}$ and satisfies:
$$\big(\mathtt{name}(\mathtt{root\text{-}of}(i))^{\mathcal{I}}, \mathrm{d}_i\big) \in \left(\bigcap_{r(\mathtt{root\text{-}of}(i),x_i)\in q'} r^{\mathcal{I}} \quad \cap \bigcap_{r(x_i,\mathtt{root\text{-}of}(i))\in q' \text{ only if } \mathcal{I}\in\Theta} (r^-)^{\mathcal{I}}\right) \quad (\spadesuit)$$
By Θ-tree-shapedness of $q'\!\restriction_{\mathtt{SubTree}_i}$ and Lemma 3.6, there exists a homomorphism $\mathfrak{h}_i$ from $\mathcal{I}_{q'\restriction_{\mathtt{SubTree}_i}}$ to $\mathcal{I}$ with $\mathfrak{h}_i(x_i) = \mathrm{d}_i$. Thus we can simply put $\mathfrak{h}(x) := \mathfrak{h}_i(x)$ for all $x \in \mathtt{SubTree}_i$.
- Take any component $\hat{q}$ of $q'\!\restriction_{\mathtt{Trees}}$, which by Item (a) of Definition 4.11 is Θ-tree-shaped. By compatibility of $\Pi_{q'}^{\mathsf{N}}$ with $\mathcal{I}$ and Item (A) of Definition 4.14 we infer the existence of an element $\mathrm{d} \in \Delta^{\mathcal{I}}$ in the interpretation of $\mathrm{Match}_{\hat{q}}$. By invoking Corollary 3.7, we obtain a homomorphism $\mathfrak{h}_{\hat{q}}\colon \mathcal{I}_{\hat{q}} \to \mathcal{I}$. Finally, we put $\mathfrak{h}(x) := \mathfrak{h}_{\hat{q}}(x)$ for all $x \in \mathrm{Var}(\hat{q})$.

We claim that $\mathfrak{h}$ is indeed a function. This follows by the fact that (i) the sets $\mathtt{Roots}$, $\mathtt{SubTree}_1$, …, $\mathtt{SubTree}_n$, $\mathtt{Trees}$ induce a partition of $\mathrm{Var}(q')$, (ii) all Θ-tree-shaped queries from $\mathtt{Trees}$ are variable-disjoint, and (iii) the employed homomorphism are functions themselves. Hence, it remains to show that $\mathfrak{h}$ is a homomorphism from $\mathcal{I}_{q'}$ to $\mathcal{I}$. Proving the preservation of atomic concepts and self-loops (in case $\mathsf{Self} \in \Theta$) by $\mathfrak{h}$ is immediate: for root variables we employ Item (B) of Definition 4.14, while for the other variables we rely on the fact that the value of $\mathfrak{h}$ is then defined via another homomorphism. To see that $\mathfrak{h}$ preserves roles, we take any atom $\alpha(x, y) \in q'$. We establish $(\mathfrak{h}(x), \mathfrak{h}(y)) \in \alpha^{\mathcal{I}}$ with a case analysis relying on Item (c) of Definition 4.11 . There are four possible cases to consider, based on the location of x and y.

- Both x and y belong to $\mathtt{Roots}$. Then $(\mathfrak{h}(x), \mathfrak{h}(y)) = \big(\mathtt{name}(x)^{\mathcal{I}}, \mathtt{name}(y)^{\mathcal{I}}\big)$ holds, and the inclusion of $(\mathfrak{h}(x), \mathfrak{h}(y))$ in $\alpha^{\mathcal{I}}$ follows from Item (C) of Definition 4.14.
- There exists an index $1 \le i \le n$ such that $x, y \in \mathtt{SubTree}_i$. Then $(x, y) \in \alpha^{\mathcal{I}_{q'\restriction_{\mathtt{SubTree}_i}}}$ holds. By equality $(\mathfrak{h}(x), \mathfrak{h}(y)) = (\mathfrak{h}_i(x), \mathfrak{h}_i(y))$ and the fact that $\mathfrak{h}_i$ is a homomorphism, we conclude the inclusion of $(\mathfrak{h}(x), \mathfrak{h}(y))$ in $\alpha^{\mathcal{I}}$.

- Both x and y belong to `Trees`. From the fact that $(x, y) \in \alpha^{\mathcal{I}_{q'}}$ we know that x and y are in the same Θ-subtree of $\hat{q}$ of `Trees`. Thus $(x, y) \in \alpha^{\mathcal{I}_{\hat{q}}}$. Now it suffices to apply the fact that $\mathfrak{h}_{\hat{q}}$ is a homomorphism and the equality $(\mathfrak{h}(x), \mathfrak{h}(y)) = (\mathfrak{h}_{\hat{q}}(x), \mathfrak{h}_{\hat{q}}(y))$ to conclude the desired inclusion of $(\mathfrak{h}(x), \mathfrak{h}(y))$ in $\alpha^{\mathcal{I}}$.
- The variables x and y belong to different sets. From Item (c) of Definition 4.11, we know that there are two cases to consider. In the first case we know that $x \in$ `Roots` and there is an index i for which `root-of`$(i) = x$ and $y = x_i \in$ `SubTree`$_i$ is the root of $q\restriction_{\texttt{SubTree}_i}$. The other case is when the inverse operator is allowed in Θ and the roles of x and y are swapped. As this case is analogous to the previous one, we omit it. Note that by Equation (♠) we know that $(\mathfrak{h}(x), \mathfrak{h}(y))$ is actually equal to $\big(\texttt{name}(\texttt{root-of}(i))^{\mathcal{I}}, \mathrm{d}_i\big)$ for some already-fixed $\mathrm{d}_i \in \Delta^{\mathcal{I}}$. Finally, by applying the first part of Equation (♠), we get that $(\mathfrak{h}(x), \mathfrak{h}(y)) = \big(\texttt{name}(\texttt{root-of}(i))^{\mathcal{I}}, \mathrm{d}_i\big)$ belongs to $\alpha^{\mathcal{I}}$, as required.

This establishes that $\mathfrak{h}$ is indeed a homomorphism, finishing the proof. □

As a preliminary step towards the reverse direction of Lemma 4.15, we introduce a handy notion of (N, Θ)-*simple matches*. They reflect the intuition that our queries match interpretations resembling Θ-forests, and hence such matches should themselves look like Θ-forests and they should not contain the "redundancy" in variables that may occur in the presence of Θ-forks. Consult Definition 4.1 if needed.

Definition 4.16 Let q, N, Θ, and $\mathcal{I}$ be as in Definition 4.14. A match $\pi\colon \mathrm{Var}(q) \to \Delta^{\mathcal{I}}$ is an **(N, Θ)-simple match** if (i) the image $\mathcal{J}_\pi := \pi[\mathcal{J}]$ of each connected component $\mathcal{J}$ of $\mathcal{I}_q$ via π is an $(\mathsf{ind}(\mathcal{J}_\pi) \cap \mathsf{N})$-rooted Θ-forest, and (ii) there are no Θ-fork (x, y) in q satisfying $\pi(x) = \pi(y)$.

We stress that despite the fact that π is an (N, Θ)-simple, there still can be variables x and y satisfying $\pi(x) = \pi(y)$. The following example illustrates a difference between simple and non-simple matches.

Example 4.17. Consider the $(4, \emptyset, \emptyset)$-forest $\mathcal{I}$ depicted below, and the conjunctive query $q := r(x_1, x_2) \wedge r(x_1, x_3) \wedge r(x_2, x_4) \wedge r(x_3, x_5) \wedge r(x_{2'}, x_4) \wedge r(x_{3'}, x_5) \wedge r(x_{1'}, x_{2'}) \wedge r(x_{1'}, x_{3'})$ (having the same shape as $\mathcal{I}$).

$$\mathcal{I} := \begin{tikzcd} & 2 \arrow[r, "r"] & 4 & 2' \arrow[l, "r"'] & \\ 1 \arrow[ur, "r"] \arrow[dr, "r"'] & & & & 1' \arrow[ul, "r"'] \arrow[dl, "r"] \\ & 3 \arrow[r, "r"] & 5 & 3' \arrow[l, "r"] & \end{tikzcd} \qquad q := \begin{tikzcd} & x_2 \arrow[r, "r"] & x_4 & x_{2'} \arrow[l, "r"'] & \\ x_1 \arrow[ur, "r"] \arrow[dr, "r"'] & & & & x_{1'} \arrow[ul, "r"'] \arrow[dl, "r"] \\ & x_3 \arrow[r, "r"] & x_5 & x_{3'} \arrow[l, "r"] & \end{tikzcd}$$

The mapping $\pi\colon x_i \mapsto i, x_{i'} \mapsto i'$ is clearly a match for q and $\mathcal{I}$, but it is not a simple. Yet another match for q and $\mathcal{I}$ is $\pi'\colon x_i \mapsto i, x_{i'} \mapsto i$. By eliminating forks involving elements having equal image via π, q gives raise to a fork rewriting $q' := r(x_1, x_2) \wedge r(x_1, x_3) \wedge r(x_2, x_4) \wedge r(x_3, x_5)$ of q, obtained by identifying the "primed" variables with their "non-primed" counterparts. Note that the match $\pi''\colon x_i \mapsto i$ for q' and $\mathcal{I}$ is $(\emptyset, \emptyset)$-simple.

We next see that every query has a Θ-fork rewriting that enjoys an (N, Θ)-simple match.

Lemma 4.18 Fix q, N, Θ, and $\mathcal{I}$ as in Definition 4.14 and suppose that $\mathcal{I} \models q$. Then there exists a Θ-fork rewriting q' of q, and a match $\pi'\colon \mathrm{Var}(q') \to \Delta^{\mathcal{I}}$ that is (N, Θ)-simple.

Proof. Let π be any match witnessing $\mathcal{I} \models_\pi q$. We first modify π to make its image look like a Θ-forest. Take any maximal connected substructure $\mathcal{J}$ of $\mathcal{I}_q$, that is induced by some $\mathrm{V} \subseteq \mathrm{Var}(q')$. Moreover, let $\mathcal{J}_\pi := \pi[\mathcal{J}]$ be the image of $\mathcal{J}$ via π. Note that $|\mathcal{J}| \leq |q|$. By the fact that $\mathcal{I}$ is a $(|q|, \mathsf{N}, \Theta)$-forest, we infer the existence of an $(\mathsf{ind}(\mathcal{J}_\pi) \cap \mathsf{N})$-homomorphisms $\mathfrak{f}$ from $\mathcal{J}_\pi$ to some $(\mathsf{ind}(\mathcal{J}_\pi) \cap \mathsf{N})$-rooted Θ-forest $\mathcal{J}'$ and $\mathfrak{g}$ from $\mathcal{J}'$ to $\mathcal{I}$. We next redefine π so that it maps any variable v from V to $\mathfrak{g}(\mathfrak{f}(\pi(v)))$. As composition of homomorphisms is also a homomorphism, the resulting mapping is a match for q and $\mathcal{I}$. What is more, this guarantees that the image of V by π constitutes an $(\mathsf{ind}(\mathcal{J}_\pi) \cap \mathsf{N})$-rooted Θ-forest. We repeat the process for all maximal connected substructures $\mathcal{J}$ of $\mathcal{I}_q$. Call the resulting match π'. Finally, we let q' be the query obtained by exhaustively eliminating Θ-forks (x, y) from q for which $\pi'(x) = \pi'(y)$ holds. Note that π', modulo a trivial renaming required after identifying variables during fork elimination, is the desired (N, Θ)-simple match for q' and $\mathcal{I}$. This concludes the proof. ☐

Knowing Lemma 4.18, we will see how the satisfaction of conjunctive queries implies the existence of splittings. We define a suitable splitting first, and in consecutive lemmas we establish its correctness.

Definition 4.19 Having q, N, Θ, and $\mathcal{I}$ as before, suppose that π is an (N, Θ)-simple match witnessing $\mathcal{I} \models q$. We define the **canonical (N, Θ)-splitting** of q w.r.t. π as

$$\Pi^{\mathsf{N}}_q := (\mathtt{Roots}, \mathtt{name}, \mathtt{SubTree}_1, \mathtt{SubTree}_2, \ldots, \mathtt{SubTree}_n, \mathtt{root\text{-}of}, \mathtt{Trees}),$$

where the consecutive definitions of each of the components are provided below.

- The set $\mathtt{Roots}$ is composed of all variables $x \in \mathrm{Var}(q)$ for which $\pi(x)$ is an N-named element of $\mathcal{I}$. For all such variables x we set $\mathtt{name}(x) := \mathsf{a}$ for *any* corresponding $\mathsf{a} \in \mathsf{N}$.
- We say that a variable x is Θ-*dangling from a root* if there exists a variable $x_r \in \mathtt{Roots}$ and either (i) some atom $r(x_r, x)$ belongs to q, or (ii) $\mathsf{I} \in \Theta$ and some atom $r(x, x_r)$ belongs to q. Let V be the $\subseteq$-maximal set of variables from $(\mathrm{Var}(q) \setminus \mathtt{Roots})$ that are Θ-dangling from the roots. We put $n := |\mathrm{V}|$ and fix an ordering $x_1, x_2, \ldots, x_n$ on the members of V. We use $x_r^1, x_r^2, \ldots, x_r^n$ to denote any roots from which x_i are dangling. For each index $1 \leq i \leq n$ we put $\mathtt{root\text{-}of}(i) := x_r^i$, and define $\mathtt{SubTree}_i$ as the set composed of x_i and of all the variables that are Θ-reachable from the variable x_i in the query structure of q restricted to the set $\mathrm{Var}(q) \setminus \mathtt{Roots}$.
- The set $\mathtt{Trees}$ is composed of all remaining variables.

The following three lemmas establish correctness of the above definition of a canonical splitting.

Lemma 4.20 The components of the canonical splitting from Definition 4.19 are well-defined.

Proof. We employ the naming scheme from Definition 4.16. The function $\mathtt{name}$ is well-defined from the fact that every N-named element is $\{\mathsf{a}\}$-named for at least one name a from N. The function $\mathtt{root\text{-}of}$ is well-defined as every variable from V has a root from which it is dangling (by design). Hence, it suffices to show that the sets $\mathtt{Roots}$, $\mathtt{SubTree}_1$, …, $\mathtt{SubTree}_n$, $\mathtt{Trees}$ form a partition of $\mathrm{Var}(q)$. While the fact that the union of these sets is equal to $\mathrm{Var}(q)$ follows from their construction and the definition $\mathtt{Trees}$, we need to show that these sets are pairwise disjoint. By construction, the sets $\mathtt{Roots}$ and $\mathtt{Trees}$ are disjoint from the other sets. Thus, it suffices to consider any pair $i \neq j$ and establish the disjointness of $\mathtt{SubTree}_i$ and $\mathtt{SubTree}_j$. Towards a contradiction suppose that $\mathtt{SubTree}_i \cap \mathtt{SubTree}_j \neq \emptyset$, *i.e.* there exists a variable v that is Θ-reachable from both x_i and x_j (the elements selected in the construction of V, different by definition) in the query structure of q restricted to, respectively, $\mathtt{SubTree}_i$ and $\mathtt{SubTree}_j$. Consider the following cases:

1. $\mathtt{SubTree}_i = \{x_i\}$ and $\mathtt{SubTree}_j = \{x_j\}$. This implies that $v = x_j$ or $v = x_i$, and contradicts the fact that each of the above sets is a singleton.
2. $\mathsf{I} \notin \Theta$ and $v = x_j$ (the case of $v = x_i$ is analogous). Then there exists a variable u in $\mathtt{SubTree}_i$ as well as some atoms $r(u, x_j)$ and $s(x_r^j, x_j)$ are present in the query q. Note that it suffices to show that the images of u, x_j, and x_r^j via π are pairwise-different. Indeed, by the (N, Θ)-simplicity of π and by N-anonymity of $\pi(u)$ and $\pi(x_j)$, this contradicts the fact that $\mathcal{I}$ restricted to $\{\pi(u), \pi(x_r^j), \pi(x_j)\}$ is an N'-rooted Θ-forest for some non-empty subset N' of N containing the name assigned to x_r^j. First, the inequality $\pi(x_j) \neq \pi(x_r^j)$ follows from the fact that $\pi(x_r^j)$ is N-named while $\pi(x_j)$ is not by its inclusion in $\mathtt{SubTree}_i$. Second, the inequality $\pi(u) \neq \pi(x_r^j)$ follows from the fact that we eliminated all such

$\emptyset$-forks from q (see the definition of (N, Θ)-simple matches). Third, we establish the inequality $\pi(x_j) \neq \pi(u)$ via an indirect proof. Suppose that $\pi(x_j) = \pi(u)$ holds, which implies that some N-anonymous element $\mathrm{d} \coloneqq \pi(u)$ carries an r-self-loop (*i.e.* $(\mathrm{d}, \mathrm{d}) \in r^{\mathcal{I}}$). If $\mathsf{Self} \notin \Theta$, we have a contradiction with the fact that $\mathcal{I}$ restricted to $\{\mathrm{d}\}$ was supposed to be an $\emptyset$-tree. Otherwise, we have a contradiction with the fact that we exhaustively applied $\{\mathsf{Self}\}$-fork elimination on q for all pairs having equal image via π.

3. $\mathsf{I} \in \Theta$ and $v = x_j$ (the case of $v = x_i$ is analogous). Let $\mathcal{J}$ be the connected component of $\mathcal{I}_q$ containing all of x_i, x_j, x_r^j, and x_r^j (guaranteed by the assumption on $\mathtt{SubTree}_i \cap \mathtt{SubTree}_j$, and the fact that x_i and x_j are dangling from their roots x_r^j, and x_r^j). Moreover, let $\mathcal{J}_\pi \coloneqq \pi[\mathcal{J}]$ be the image of $\mathcal{J}$ via π. By Lemma 4.18 we have that $\mathcal{J}_\pi$ is an $(\mathsf{N} \cap \mathsf{ind}(\mathcal{J}_\pi))$-rooted Θ-forest. Let $\rho \coloneqq \rho_1 \ldots \rho_m$ be a shortest (undirected) path from x_i to x_j in $\mathcal{I}_q$ restricted to $\mathtt{SubTree}_i$ (it exists by the assumption). We are going to prove inductively that for every $\ell < m$ we have that $\pi(\rho_{\ell+1})$ in $\mathcal{J}_\pi$ is a child of $\pi(\rho_\ell)$ in $\mathcal{J}_\pi$. This would yield a contradiction with the fact ρ_m $(= x_j)$ is mapped to a child of a root (the distance is too big). For the base of our induction, observe that $\pi(\rho_2)$ is a child of $\pi(\rho_1)$ due to the fact that $\pi(\rho_2)$ is N-anonymous (otherwise ρ_2 would be a member of $\mathtt{Roots}$). Take $\ell > 2$ and suppose that the inductive assumption holds for all indices lower than ℓ. From the facts that all members of ρ are N-anonymous and $\mathcal{J}_\pi$ is an $(\mathsf{N} \cap \mathsf{ind}(\mathcal{J}_\pi))$-rooted Θ-forest, there are only three possible options: (i) $\pi(x_\ell)$ is a son of $\pi(x_{\ell-1})$, (ii) $\pi(x_\ell) = \pi(x_{\ell-2})$ (informally: the path ρ goes down and then returns back), (iii) $\mathsf{Self} \in \Theta$ and $\pi(x_{\ell-1}) = \pi(x_\ell)$ (informally: the element $\rho_{\ell-1}$ enters a self-loop). The second and the third case clearly cannot happen as the match π is (N, Θ)-simple (such forks were eliminated!). Hence, $\pi(x_j)$ is indeed a son of $\pi(x_{j-1})$, concluding the proof of this sub-case.
4. The variable v is different from x_i and x_j. The case of $\mathsf{I} \in \Theta$ is already solved by the previous case (*i.e.* the union of paths from x_i to v and from x_j to v leads from x_i to v_j), hence assume $\mathsf{I} \notin \Theta$. The proof is will be roughly the same as the proof of the second case, so we prefer to keep its description short. From the assumption we infer the existence of variables u and w in $\mathtt{SubTree}_i$ as well as some atoms $r(w, v)$ and $s(u, v)$. Once more, we aim at showing that the images of u, v, and w are pairwise different, which will lead us to a contradiction with the (N, Θ)-simplicity of the match π. All three inequalities are obtained in precisely the same way as before. The first one, namely $\pi(u) \neq \pi(w)$, follows from the fact that we eliminated all such $\emptyset$-forks from q. The two remaining inequalities, namely $\pi(v) \neq \pi(w)$ and $\pi(u) \neq \pi(v)$, can be established by copy-pasting (modulo variable renaming) the proof of the third inequality of the second case.

The above case analysis implies the sets $\mathtt{Roots}$, $\mathtt{SubTree}_1$, …, $\mathtt{SubTree}_n$, $\mathtt{Trees}$ indeed are a partition of $\mathrm{Var}(q)$. This concludes the proof that components of Π_q^{N} are well-defined. ☐

Lemma 4.21 The canonical splitting defined in Definition 4.19 is an (N, Θ)-splitting.

Proof sketch. We follow Definition 4.11 and establish the correctness of all its items. As the reader should now be familiar with all the "tricks" required to complete the proof, we provide a proof sketch only. Let us start with Item (a). Given a variable $v \in \mathtt{Trees}$, let T_v denote the set of all variables w that are Θ-reachable in $q{\restriction}_{\mathtt{Trees}}$. What is more, let T to be a $\subseteq$-minimal set of variables v whose union of T_v equals to $\mathtt{Trees}$. We need to show that for any $v \neq u$ the sets T_v and T_u are disjoint, and the queries $q{\restriction}_{\mathrm{T}_v}$ are Θ-tree-shaped. The first part can be shown in total analogy to the case analysis done in the proof of Lemma 4.20, and hence we omit the proof. The second part can be shown as follows. Take any $v \in \mathrm{T}$, and note that (i) $q{\restriction}_{\mathrm{T}_v}$ is connected by design, and (ii) all elements of T_v are mapped via π to N-anonymous elements (by design of $\mathtt{Trees}$). Hence, by the fact that π is (N, Θ)-simple, the image of $q{\restriction}_{\mathrm{T}_v}$ via π is a Θ-tree. Thus, by invoking Corollary 4.8, we infer that $q{\restriction}_{\mathrm{T}_v}$ is Θ-tree-shaped. Thus $q{\restriction}_{\mathtt{Trees}}$ is indeed a variable-disjoint conjunction of Θ-tree-shaped CQs. The same reasoning also proves Item (b) of Definition 4.11. Establishing Item (c) of Definition 4.11 can be done with an exhaustive case analysis that concerns atoms of the form $\alpha(x, y)$ in q for which x and y are in different components of a splitting and that violate Item (c). The contradiction arises either (i) by the presence of a fork (u, v) with $\pi(u) = \pi(v)$, or (ii) by the construction of the sets $\mathtt{SubTree}_i$ (the definition of "dangling") and their disjointness. Finally, Item (d) of Definition 4.11 is a direct consequence of our definition of elements "dangling from the roots". □

Lemma 4.22 The canonical splitting defined for $\mathcal{I}$ in Definition 4.19 is compatible with $\mathcal{I}$.

Proof. Item (A) and Item (D) of Definition 4.14 follows from the fact that all components $\hat{q}$ of $q{\restriction}_{\mathtt{Trees}}$ as well as all queries $q{\restriction}_{\mathtt{SubTree}_i}$ are Θ-tree-shaped, and thus the images of their roots belong to the interpretation of the "query concepts" by Corollary 3.7. Items (B) and (C) of Definition 4.14 are immediate consequence of the fact that π is a match for q and $\mathcal{I}$. □

Summarising all the previous lemmas, we are finally able to establish the reverse direction of Lemma 4.15.

Lemma 4.23 Let q, N, Θ, and $\mathcal{I}$ be as in Definition 4.14. If $\mathcal{I} \models q$, then there exists a Θ-fork rewriting q' of q and an (N, Θ)-splitting $\Pi^{\mathsf{N}}_{q'}$ of q', such that $\Pi^{\mathsf{N}}_{q'}$ is compatible with $\mathcal{I}$.

Proof. Suppose $\mathcal{I} \models q$. By Lemma 4.18 there exists a Θ-fork rewriting q' of q and a variable assignment $\pi' \colon \mathrm{Var}(q') \to \Delta^{\mathcal{I}}$ that is an (N, Θ)-simple match. Hence, we can apply Definition 4.19 we infer a canonical (N, Θ)-splitting $\Pi^{\mathsf{N}}_{q'}$, which is a desired (N, Θ)-splitting splitting of q' compatible with $\mathcal{I}$ (as follows from Lemma 4.20, Lemma 4.21, and Lemma 4.22). □

We conclude the section by compiling Lemma 4.15 and Lemma 4.23 into a single corollary.

Corollary 4.24
Let $\mathsf{N} \subseteq \mathbf{N_I}$, $\Theta \subseteq \{\mathsf{I}, \mathsf{Self}\}$, q be a CQ, and $\mathcal{I}$ be a $(|q|, \mathsf{N}, \Theta)$-forest. Then $\mathcal{I} \models q$ if and only if there is a Θ-fork rewriting q' of q and an (N, Θ)-splitting $\Pi^{\mathsf{N}}_{q'}$ of q' compatible with $\mathcal{I}$.

4.4 Step III: Spoilers

Spoilers [Lut08b, p. 6] are $\mathcal{ALCO}^{\cap}$-knowledge-bases dedicated for preventing query matches.

Definition 4.25 Fix $\mathsf{N} \subseteq \mathbf{N_I}$, $\Theta \subseteq \{\mathcal{I}, \mathsf{Self}\}$, a CQ q, and an (N, Θ)-splitting Π^{N}_q with its usual components $(\mathtt{Roots}, \mathtt{name}, \mathtt{SubTree}_1, \ldots, \mathtt{SubTree}_n, \mathtt{root\text{-}of}, \mathtt{Trees})$. An **$(\mathsf{N}, \Theta)$-spoiler $\mathcal{K}^{\lightning}_{\Pi^{\mathsf{N}}_q}$** for Π^{N}_q is an $\mathcal{ALCO}^{\cap}$-KB containing at least one of the following axioms:

(A) $\top \sqsubseteq \neg\mathrm{Match}_{\hat{q}}$ for some Θ-tree-shaped query $\hat{q}$ being a Θ-connected component of $q{\restriction}_{\mathtt{Trees}}$.

(B) $\neg\mathrm{A}(\mathsf{a})$ for some atom $\mathrm{A}(x) \in q$ with $x \in \mathtt{Roots}$ and $\mathsf{a} \coloneqq \mathtt{name}(x)$.

(C) $\neg r(\mathsf{a}, \mathsf{b})$ from some atom $r(x, y)$ of q with $x, y \in \mathtt{Roots}$ and $(\mathsf{a}, \mathsf{b}) \coloneqq (\mathtt{name}(x), \mathtt{name}(y))$.

(D) $\neg\mathrm{C}_i(\mathsf{a})$, where $\mathsf{a} \coloneqq \mathtt{name}(\mathtt{root\text{-}of}(i))$ and an $\mathcal{ALCO}^{\cap}$-concept C_i, for some index $1 \leq i \leq n$ and for a variable x_i denoting the root variable of $q{\restriction}_{\mathtt{SubTree}_i}$, is defined as follows

$$\mathrm{C}_i \coloneqq \exists \left(\bigcap_{r(\mathtt{root\text{-}of}(i), x_i) \in q} r \quad \cap \bigcap_{r(x_i, \mathtt{root\text{-}of}(i)) \in q, \text{ only if } \mathsf{I} \in \Theta} r^- \right) .\mathrm{Match}_{q{\restriction}_{\mathtt{SubTree}_i}}.$$

Observe the correspondence between Items (A)–(D) of the above definition and Items (A)–(D) from Definition 4.11. These cases can be seen as potential ways of "blocking" the compatibility of a splitting.

Following Lutz [Lut08b, p. 7] we say that a role conjunction $r_1 \cap \ldots \cap r_n \cap s_1^- \cap \ldots \cap s_m^-$ **occurs** in a conjunctive query q if we can find two variables v and v from $\mathrm{Var}(q)$ for which equations

$$\{r \in \mathbf{N_R} \mid r(v, v') \in q\} = \{r_1, r_2, \ldots, r_n\} \text{ and } \{s \in \mathbf{N_R} \mid s(v', v) \in q\} = \{s_1, s_2, \ldots, s_m\}$$

hold. Similarly, we speak about concept and role names that **occur** in the query q. Note that the role conjunctions, concept and role names that are used in Definition 4.14 occur in q. We can first show:

Lemma 4.26 Let N, Θ, q, and Π^{N}_q be as in Definition 4.25, and let $m := |\mathrm{Var}(q)|$. Then there are at most $(|q|^2 \cdot m^2 \cdot 2^{2m}) \cdot |\mathsf{N}|^4$ $\subseteq$-minimal (N, Θ)-spoilers $\mathcal{K}^{\lightning}_{\Pi^{\mathsf{N}}_q}$ for Π^{N}_q, and each of them is of size at most linear in $|q|$. Moreover, the set of all $\subseteq$-minimal (N, Θ)-spoilers can be enumerated in time polynomial w.r.t. $(|q|^2 \cdot m^2 \cdot 2^{2m}) \cdot |\mathsf{N}|^4$.

Proof sketch. The number of queries in axioms from Item (A) of Definition 4.25 is bounded by the total number of subsets of $\mathrm{Var}(q)$ (thus 2^m), and every such axiom is of size linear in $|q|$ (Corollary 3.7). The number of axioms from Item (B) of Definition 4.25 can clearly be bounded by $|q| \cdot |\mathsf{N}|$, and every such axiom is of constant size. Similarly, the number of axioms from Item (C) of Definition 4.25 can clearly be bounded by $|q| \cdot |\mathsf{N}|^2$, and every such axiom is of constant size. Finally, we see that the choice of a, x_i and its root variable, and a subset of variables from $\mathrm{Var}(q)$ completely determines the axiom Item (D) of Definition 4.25 (note that the role conjunction present there *occurs* in q). Reasoning analogously to the first case, we can bound the total number of such axioms by $|\mathsf{N}| \cdot m^2 \cdot 2^m$ and bound their sizes by some function linear in $|q|$. We are done by simplifying $(2^m) \cdot (|q| \cdot |\mathsf{N}|) \cdot (|q| \cdot |\mathsf{N}|^2) \cdot (|\mathsf{N}| \cdot m^2 \cdot 2^m)$.
The enumeration of $\subseteq$-minimal spoilers can be done by enumerating axioms from each of the cases from Definition 4.25 relying on Corollary 3.7 to construct "rolling-up" concepts. □

By collecting spoilers for all possible splittings we obtain *super spoilers*. The first part of their definitions ensures that super spoilers "spoil" compatibility of all possible splittings, while the second part of the definition keeps sizes of super spoilers under control (and suggests a way of their construction).

Definition 4.27 Let (N, Θ, q) be as in Definition 4.25. We say that an $\mathcal{ALCO}^{\cap}$-KB $\mathcal{K}^{\lightning\star}_q$ is an **(N, Θ)-super-spoiler** for q if for all Θ-fork-rewritings q' of q and for all (N, Θ)-splittings $\Pi^{\mathsf{N}}_{q'}$ of q' we have that $\mathcal{K}^{\lightning\star}_q$ is an (N, Θ)-spoiler for $\Pi^{\mathsf{N}}_{q'}$. Moreover, we require that $\mathcal{K}^{\lightning\star}_q$ is **functional**, *i.e.* for every axiom α present in $\mathcal{K}^{\lightning\star}_q$ there exists a Θ-fork-rewriting q' of q and an (N, Θ)-splitting $\Pi^{\mathsf{N}}_{q'}$ such that $\{\alpha\}$ is an (N, Θ)-spoiler for $\Pi^{\mathsf{N}}_{q'}$ of the form mentioned by Items (A)–(D) of Definition 4.25.

"Functionality" in the above definition is intended to play the role analogous to "$\subseteq$-minimality". Checking whether a super-spoiler does not contain a super-spoiler as a strict subset is however, costly.

The forthcoming lemmas show that the existence of an (N, Θ)-super-spoiler indeed "spoils" the (finite) entailment of an input conjunctive query over (finite) $(|q|, \mathsf{N}, \Theta)$-forests. Consult the following lemmas.

Lemma 4.28 Let (N, Θ, q) be as in Definition 4.25, and let $\mathcal{I}$ be a $(|q|, \mathsf{N}, \Theta)$-forest. Then $\mathcal{I} \not\models q$ implies the existence of a (N, Θ)-super-spoiler $\mathcal{K}^{\lightning\star}_q$ for q for which $\mathcal{I} \models \mathcal{K}^{\lightning\star}_q$.

Proof. We inductively construct a sequence $\mathcal{K}_0 := \emptyset, \mathcal{K}_1, \ldots$ of $\mathcal{ALCO}^{\cap}$-KBs converging to an (N, Θ)-super-spoiler for q. To do so, we first fix some ordering on pairs $\mathrm{P}_i := (q', \Pi^{\mathsf{N}}_{q'})$ of Θ-fork rewritings q' and (N, Θ)-splittings of q', and then proceed inductively with the i-th such pair. Note that $\Pi^{\mathsf{N}}_{q'}$ is not compatible with $\mathcal{I}$. Indeed, Lemma 4.15 would then imply $\mathcal{I} \models q$. Thus, there is at least one item of Definition 4.14 that is violated and there exists the corresponding axiom α for which $\mathcal{I} \not\models \alpha$ holds. Let β be the corresponding "negated" axiom from Definition 4.25. We put $\mathcal{K}_i := \mathcal{K}_{i-1} \cup \{\beta\}$. From the definition of a spoiler and by construction, we see that $\mathcal{K}_i$ is an (N, Θ)-spoiler for P_i as well as for all previous pairs. Moreover, $\mathcal{I} \models \beta$.

There are only finitely many pairs P_i to consider, hence we can take $\mathcal{K}_q^{\bigstar}$ to be the last knowledge base on the list. Clearly $\mathcal{I} \models \mathcal{K}_q^{\bigstar}$, and thus $\mathcal{K}_q^{\bigstar}$ is the desired (N, Θ)-super-spoiler for q. $\square$

Lemma 4.29 Let (N, Θ, q) be as in Definition 4.25, and let $\mathcal{I}$ be a $(|q|, \mathsf{N}, \Theta)$-forest. Then the existence of a (N, Θ)-super-spoiler $\mathcal{K}_q^{\bigstar}$ for q for which $\mathcal{I} \models \mathcal{K}_q^{\bigstar}$ holds implies that $\mathcal{I} \not\models q$.

Proof. Take an (N, Θ)-super-spoiler $\mathcal{K}_q^{\bigstar}$ for q guaranteed by the statement of the lemma. Towards a contradiction, suppose that $\mathcal{I} \models q$. Hence, by Lemma 4.23 we infer the existence of a Θ-fork rewriting q' of q and an (N, Θ)-splitting $\Pi^{\mathsf{N}}_{q'}$ of q' that is compatible with $\mathcal{I}$. By the fact that $\mathcal{K}_q^{\bigstar}$ is an (N, Θ)-super-spoiler for q we also know (by Definition 4.27) that it is a spoiler for $\Pi^{\mathsf{N}}_{q'}$. This means that at least one of the conditions (A)–(D) from Definition 4.25 applied to $\Pi^{\mathsf{N}}_{q'}$ hold, contradicting the compatibility of $\Pi^{\mathsf{N}}_{q'}$ with $\mathcal{I}$. $\square$

Relying on the presented lemmas we can provide a reduction from the (U)CQ entailment problem to the problem of checking the existence of an $(\mathsf{ind}(\mathcal{K}), \Theta)$-super-spoiler spoiling the (finite) satisfiability of $\mathcal{K}$.

Lemma 4.30 Let $\mathcal{DL}$ be a (finitely) Θ-forest-friendly description logic extending $\mathcal{ALC}\Theta^{\cap}$, $\mathcal{K}$ be a $\mathcal{DL}$-knowledge-base and $q \coloneqq \bigvee_{i=1}^{m} q_i$ be a union of conjunctive queries. Then $\mathcal{K} \not\models_{(\mathrm{fin})} q$ if and only if for each query q_i there exists an $(\mathsf{ind}(\mathcal{K}), \Theta)$-super-spoiler $\mathcal{K}_{q_i}^{\bigstar}$, and $\mathcal{K} \cup \bigcup_{i=1}^{m} \mathcal{K}_{q_i}^{\bigstar}$ is (finitely) satisfiable.

Proof. Suppose $\mathcal{K} \not\models_{(\mathrm{fin})} q$, and take any (finite) $(|q|, \mathsf{ind}(\mathcal{K}), \Theta)$-forest model $\mathcal{I}$ of $\mathcal{K}$ that violates q (guaranteed by Lemma 3.12). This in particular means that for each $1 \leq i \leq n$ we have that $\mathcal{I} \not\models q_i$. By Lemma 4.28 we obtain an $(\mathsf{ind}(\mathcal{K}), \Theta)$-super-spoilers $\mathcal{K}_{q_i}^{\bigstar}$ for each q_i for which $\mathcal{I} \models \mathcal{K}_{q_i}^{\bigstar}$. Thus $\mathcal{I} \models \mathcal{K} \cup \bigcup_i \mathcal{K}_{q_i}^{\bigstar}$, implying (finite) satisfiability of $\mathcal{K} \cup \bigcup_i \mathcal{K}_{q_i}^{\bigstar}$. For the other direction, suppose that $\mathcal{K}' \coloneqq \mathcal{K} \cup \bigcup_{i=1}^{m} \mathcal{K}_{q_i}^{\bigstar}$ is (finitely) satisfiable. Then by Definition 3.11 we know that $\mathcal{K}'$ is (finitely) Θ-coverable. Thus, by Definition 3.10 there exists a (finite) $(|q|, \mathsf{ind}(\mathcal{K}))$-locally Θ-forest model $\mathcal{J}$ of $\mathcal{K}'$ that covers $\mathcal{I}$. After multiple applications of Lemma 4.28 we know that $\mathcal{J} \not\models q_i$ for all $1 \leq i \leq m$. Hence, $\mathcal{J} \not\models q$, concluding $\mathcal{K} \not\models_{(\mathrm{fin})} q$. $\square$

4.5 Step IV: Enumerating Super-Spoilers

In this section we obtain upper bounds on the total number of super-spoilers and their sizes. We also provide a worst-case optimal algorithm for their enumeration. We conclude by showing how these bounds and the presented algorithm can be exponentially improved in the case of $(\mathsf{N}, \emptyset)$-super-spoilers. Indeed:

Lemma 4.31 Let N, Θ, q be as in Definition 4.25, let $m \coloneqq |\mathrm{Var}(q)|$, and let $\mathsf{SupSpl}_{\mathsf{N},\Theta}(q)$ denote the set of all (N, Θ)-super-spoilers for q. We have that:

(i) $|\mathsf{SupSpl}_{\mathsf{N},\Theta}(q)| \leq (m^{3m} \cdot |\mathsf{N}|^{m})^{(|q|^2 \cdot m^2 \cdot 2^{2m}) \cdot |\mathsf{N}|^4}$, and

(ii) every $\mathcal{K}_q^{\bigstar}$ from $\mathsf{SupSpl}_{\mathsf{N},\Theta}(q)$ has size linear in $(m^{3m} \cdot |q|) \cdot |\mathsf{N}|^{m}$,

(iii) the set $\mathsf{SupSpl}_{\mathsf{N},\Theta}(q)$ can be enumerated in time doubly-exponential w.r.t $|q|+|\mathsf{N}|$.

Moreover, given an $\mathcal{ALC}\Theta^{\cap}$-KB $\mathcal{K}$ we can verify in time exponential w.r.t $|q|+|\mathsf{N}|+|\mathcal{K}|$ (but polynomial w.r.t $|\mathsf{N}|+|\mathcal{K}|$ whenever the query q is fixed) whether $\mathcal{K}$ is an (N, Θ)-super-spoiler for q.

Proof sketch. By "functionality" mentioned in Definition 4.27 we know that every (N, Θ)-super-spoiler can be "over-approximated" with a function that takes a pair $(q', \Pi^{\mathsf{N}}_{q'})$ of a Θ-fork-rewriting q' of q and an (N, Θ)-splitting $\Pi^{\mathsf{N}}_{q'}$ of q', and returns a corresponding (N, Θ)-spoiler. Applying Lemma 4.10, Lemma 4.13, and Lemma 4.26 we conclude that there are at most $(m^{m} \cdot m^{2m} \cdot |\mathsf{N}|^{m})^{(|q|^2 \cdot m^2 \cdot 2^{2m}) \cdot |\mathsf{N}|^4}$ different super-spoilers. For the size of a single super-spoiler, we proceed analogously. By the multiplication principle, a single super-spoiler has size bounded by the total number of fork rewritings multiplied by the total number of splittings, and multiplied by the maximal size of a single spoiler. The aforementioned lemmas together with the size of a single spoiler from Lemma 4.26 shows that a single super-spoiler has size at most $(m^m) \cdot (m^{2m} \cdot |\mathsf{N}|^m) \cdot (\mathrm{c} \cdot |q|)$ for some constant c.

The enumeration of super-spoilers can be done by enumerating Θ-fork-rewritings (Lemma 4.10 yields an exponential function w.r.t. $|q|$), then enumerating the corresponding splittings (Lemma 4.13 yields an exponential function $\mathfrak{e}(|q|, |\mathsf{N}|)$ that becomes polynomial when q is fixed), and then selecting a single corresponding spoiler (Lemma 4.26 gives a polynomial algorithm w.r.t. $(|q|^2 \cdot m^2 \cdot 2^{2m}) \cdot |\mathsf{N}|^4)$. A possible implementation would be simply to enumerate all bit-strings of length $[(m^m \cdot m^{2m} \cdot |\mathsf{N}|^m)] \cdot \log_2((|q|^2 \cdot m^2 \cdot 2^{2m}) \cdot |\mathsf{N}|^4)$ that encode, per each fork-rewriting and splitting, an index of the selected spoiler. Hence, the enumeration process can be done in time exponential w.r.t $[(m^m \cdot m^{2m} \cdot |\mathsf{N}|^m)] \cdot \log_2((|q|^2 \cdot m^2 \cdot 2^{2m}) \cdot |\mathsf{N}|^4)$, and thus doubly-exponential w.r.t. $|q| + |\mathsf{N}|$. Finally, to check whether a given $\mathcal{ALC}\Theta^{\sqcap}$-KB $\mathcal{K}$ is a (N, Θ)-super-spoiler, we can enumerate all Θ-fork-rewritings, all corresponding (N, Θ)-splittings, and all corresponding (N, Θ)-spoilers and check if some o of them appear in $\mathcal{K}$ (this step can be done in time linear in $|\mathcal{K}|$, the running times of the other steps are specified above). The running time of this algorithm is bounded by the product of the running times of each of its components. □

This concludes the case of (N, Θ)-super-spoilers for $\Theta \neq \emptyset$. Relying on the notion of maximal $\emptyset$-fork rewriting and work by Lutz [Lut08b, Lemma 4–5], we will exponentially improve the bounds from Lemma 4.31.

Definition 4.32 Let q be a CQ, and $q^\star$ be the maximal $\emptyset$-fork-rewriting of q, and let $\mathsf{Reach}(v)$ denote the set all variables from $q^\star$ that are $\emptyset$-reachable from v in $\mathcal{I}_{q^\star}$. We define $\mathsf{QTree}(q)$ as the set of all $\emptyset$-tree-shaped queries obtained by selecting any variable v of $q^\star$ and restricting $q^\star$ to $\mathsf{Reach}(v)$. In symbols:

$$\mathsf{QTree}(q) := \left\{ q^\star{\restriction}_{\mathsf{Reach}(v)} \mid v \in \mathrm{Var}(q^\star), q^\star{\restriction}_{\mathsf{Reach}(v)} \text{ is } \emptyset\text{-tree-shaped} \right\}.$$

Clearly $|\mathsf{QTree}(q)|$ is polynomial w.r.t $|\mathsf{maxfr}(q)|$, and thus also in $|q|$. As it turns out, only trees from $\mathsf{QTree}(q)$ are relevant for $(\mathsf{N}, \emptyset)$-super-spoilers, as illustrated by the following Lemma 4.33.

Lemma 4.33 Let N and q be as before, q' be a $\emptyset$-fork-rewriting of q, $\Pi^{\mathsf{N}}_{q'} := (\mathtt{Roots}, \mathtt{name}, \mathtt{SubTree}_1, \ldots, \mathtt{SubTree}_n, \mathtt{root\text{-}of}, \mathtt{Trees})$ be an $(\mathsf{N}, \emptyset)$-splitting of q'. Moreover, let $q'_1, q'_2, \ldots, q'_k$ be the disconnected components of $q'{\restriction}_{\mathtt{Trees}}$, and $x^1_r, x^2_r, \ldots, x^n_r$ be the root variables of the corresponding $\emptyset$-tree-shaped queries $q'{\restriction}_{\mathtt{SubTree}_i}$. The following hold true:

(i) q'_i belong $\mathsf{QTree}(q)$ for all $1 \le i \le k$,

(ii) $q'{\restriction}_{\mathtt{SubTree}_i}$ belong $\mathsf{QTree}(q)$ for all $1 \le i \le n$, and

(iii) the role conjunctions $\bigcap\{r \mid r(\mathtt{root\text{-}of}(i), x^i_r) \in q'\}$ occur in $\mathsf{maxfr}(q)$.

Proof sketch. A notational variant of a proof by Lutz [Lut08b, Lemma 4] following his naming convention [Lut08b, Appendix A] Watch out! There is a glitch in Lutz's proof. In the inductive assumption no. 4, there should be $\{v, v'\} \neq \{v, v'\} \cap S_j \neq \emptyset$ rather than $\{v, v'\} \cap S_j \neq \emptyset$. □

We can now invoke Lemma 4.33 to describe the shapes of $(\mathsf{N}, \emptyset)$-super-spoilers more carefully.

Lemma 4.34 Let $\mathsf{N} \subseteq \mathbf{N_I}$ be finite, q be a CQ, and let $\mathcal{K}^{\star}_q$ be an $(\mathsf{N}, \emptyset)$-super-spoiler for q. Then all the axioms contained in $\mathcal{K}^{\star}_q$ are of one of the following forms:

(A′) $\top \sqsubseteq \neg\mathrm{Match}_{\hat{q}}$ for some $\emptyset$-tree-shaped query $\hat{q} \in \mathsf{QTree}(q)$,

(B′) $\neg A(\mathsf{a})$ for some name $\mathsf{a} \in \mathsf{N}$ and a concept name A occurring in $\mathsf{maxfr}(q)$,

(C′) $\neg r(\mathsf{a}, \mathsf{b})$ for some names $\mathsf{a}, \mathsf{b} \in \mathsf{N}$ and a role name r occurring in $\mathsf{maxfr}(q)$,

(D′) $(\neg\exists\,(s_1 \cap s_2 \cap \ldots \cap s_k)\,.\mathrm{Match}_{\hat{q}})\,(\mathsf{a})$ for some $\emptyset$-tree-shaped $\hat{q} \in \mathsf{QTree}(q)$, name $\mathsf{a} \in \mathsf{N}$, and a role conjunction $s_1 \cap s_2 \cap \ldots \cap s_k$ that *occurs* in $\mathsf{maxfr}(q)$.

Proof. Take any $(\mathsf{N}, \emptyset)$-super-spoiler for q, and let α be any of its axioms. We show that α is of the shape mentioned above. If α is of the form of Item (B) or Item (C) of Definition 4.25 we are done by the fact that (1) if r or A occurs in q then it also occurs in the maximal $\emptyset$-fork-rewriting, and (2) `name` assigns values from N. If α is of the form of Item (A) of Definition 4.25 we simply invoke Lemma 4.33. Applying all mentioned arguments we are also done with the case when α has the form from Item (D) of Definition 4.25. □

Finally, we employ Lemma 4.34 to provide the analogue of Lemma 4.31 and conclude the section.

Lemma 4.35 Let N, q, and m be as in Lemma 4.31. We have that:

(i) $|\mathsf{SupSpl}_{\mathsf{N},\emptyset}(q)|$ is exponential w.r.t $|q|+|\mathsf{N}|$,

(ii) every $\mathcal{K}_q^{\lightning\star}$ from $\mathsf{SupSpl}_{\mathsf{N},\emptyset}(q)$ has size polynomial w.r.t $|q| + |\mathsf{N}|$, and

(iii) the set $\mathsf{SupSpl}_{\mathsf{N},\emptyset}(q)$ can be enumerated in time exponential w.r.t $|q|+|\mathsf{N}|$.

Moreover, given an $\mathcal{ALC}^{\cap}$-KB $\mathcal{K}$ we can verify in time exponential w.r.t $|q|+|\mathsf{N}|+|\mathcal{K}|$ whether $\mathcal{K}$ is an $(\mathsf{N}, \emptyset)$-super-spoiler for q (and in time polynomial w.r.t $|\mathsf{N}| + |\mathcal{K}|$ if the query q is fixed).

Proof sketch. The set $|\mathsf{QTree}(q)|$ is bounded polynomially in $|q|$, and clearly can be computed in time polynomial w.r.t. $|q|$. The same holds for the maximal $\emptyset$-fork-rewriting $q^\star$ of q. To bound the size of $(\mathsf{N}, \emptyset)$-super-spoilers and their total number, we invoke Lemma 4.34 and see that the total number of such axioms can be bounded, respectively, by $|\mathsf{QTree}(q)|$, $|q| \cdot |\mathsf{N}|$, $|q| \cdot |\mathsf{N}|^2$ and $|q| \cdot |\mathsf{QTree}(q)| \cdot |\mathsf{N}|$. For their sizes, we simply invoke the fact that all "rolling-up" concepts are of polynomial size are w.r.t $|q|$ (see also Corollary 3.7).

We enumerate $(\mathsf{N}, \emptyset)$-super-spoilers by enumerating all $\mathcal{ALC}^{\cap}$-KBs containing only the axioms stated in Lemma 4.34 (requires time exponential in $|\mathsf{N}| + |q|$ by selecting appropriate atoms from the query $q^\star$ or "rolling-up" concepts obtained from $\emptyset$-tree-shaped queries from $\mathsf{QTree}(q)$). To check if a knowledge-base $\mathcal{K}$ is indeed an $(\mathsf{N}, \emptyset)$-super-spoiler, we iterate through all $\emptyset$-fork-rewritings with Lemma 4.10 (there are only m^m of them), and all $(\mathsf{N}, \emptyset)$-splittings for them (in time guaranteed by Lemma 4.13). Then we apply the definition of $(\mathsf{N}, \emptyset)$-spoilers to check if the considered knowledge-base indeed blocks the splitting, which can be performed, after fixing an N-spoiler and an N-splitting, in polynomial time w.r.t $|q|+|\mathsf{N}|+|\mathcal{K}|$. The execution times are multiplied, so the overall algorithm works in time exponential in $|\mathsf{N}|+|q|$. □

4.6 Step V: The Algorithm

We are now ready to present a meta-algorithm for deciding the (finite or unrestricted) entailment problem for unions of conjunctive queries over (finitely) Θ-forest-friendly description logics extending $\mathcal{ALC}\Theta^{\cap}$. This yields optimal algorithms for (finitely) Θ-forest-friendly DLs with ExpTime-complete (finite) satisfiability problem. We present a pseudocode below, and then establish its correctness.

Procedure 3: (Finite) UCQ entailment over (finitely) Θ-forest-friendly DLs extending $\mathcal{ALC}\Theta^{\cap}$.

```
  Input: UCQ $q := \bigvee_{i=1}^{m} q_i$ and $\mathcal{DL}$-KB $\mathcal{K}$, where $\mathcal{DL} \in \mathscr{C}_{\Theta\mathsf{fr}}$ (and $\mathcal{DL} \in \mathscr{C}^{\mathsf{fin}}_{\Theta\mathsf{fr}}$ in the finite case).
1 If $\mathcal{K}$ is not (finitely) satisfiable return True.          // Checkable in $\mathsf{SAT}_{\mathcal{DL}}(|\mathcal{K}|)$.
2 foreach selection of $(\mathsf{ind}(\mathcal{K}), \Theta)$-super spoilers $\mathcal{K}_{q_1}^{\lightning\star}, \ldots, \mathcal{K}_{q_m}^{\lightning\star}$ for $q_1, \ldots, q_m$
          // In $\mathsf{expexp}(|\mathsf{ind}(\mathcal{K})|+|q|)$ by Lemma 4.31 but in $\mathsf{exp}(|\mathsf{ind}(\mathcal{K})|+|q|)$ for empty $\Theta$ by Lemma 4.35.
3 do
4     If $\mathcal{K} \cup \bigcup_i \mathcal{K}_{q_i}^{\lightning\star}$ is (finitely) satisfiable return False.
      // In $\mathsf{SAT}_{\mathcal{DL}}(\mathsf{exp}(|\mathcal{K}| + |q|))$ by Lemma 4.31 but in $\mathsf{SAT}_{\mathcal{DL}}(\mathsf{poly}(|\mathcal{K}| + |q|))$ by Lemma 4.35 for empty $\Theta$.
5 return True.
```

In the following lemma we establish correctness of the presented algorithm.

Lemma 4.36 Procedure 4 returns `True` if and only if $\mathcal{K} \models_{(\mathrm{fin})} q$. Moreover, there exists a polynomial function poly, an exponential function exp, and a doubly-exponential function expexp, for which Procedure 4 can be implemented to work in time $\mathsf{expexp}(|\mathcal{K}| + |q|) \cdot \mathsf{SAT}_{\mathcal{DL}}(\mathsf{exp}(|\mathcal{K}|+|q|))$ for non-empty Θ and in time $\mathsf{exp}(|\mathcal{K}| + |q|) \cdot \mathsf{SAT}_{\mathcal{DL}}(\mathsf{poly}(|\mathcal{K}|+|q|))$ for empty Θ, where $\mathsf{SAT}_{\mathcal{DL}}$ denotes the worst-case optimal running time of the (finite) satisfiability problem for $\mathcal{DL}$-KBs.

Proof. For the first statement of the lemma we consider the following cases. If $\mathcal{K}$ is not (finitely) satisfiable then it entails every query. Our procedure returns `True` in this case. If $\mathcal{K}$ is (finitely) satisfiable but does not (finitely) entail q, then by Lemma 4.30 there are $(\mathsf{ind}(\mathcal{K}), \Theta)$-super-spoilers $\mathcal{K}_{q_i}^{\lightning\star}$ for q_i such that $\mathcal{K} \cup \bigcup_i \mathcal{K}_{q_i}^{\lightning\star}$ is (finitely) satisfiable and hence, the fourth line of the algorithm returns `False`. Otherwise, $\mathcal{K}$ is (finitely) satisfiable and (finitely) entails q. Thus again, by lemma 4.30, there are no such $(\mathsf{ind}(\mathcal{K}), \Theta)$-super-spoilers and so the (finite) satisfiability test in the 4th line of Procedure 4 will never succeed. Hence, the 5th line will be executed, returning `True`. The second part of the lemma follows by multiplying running times of each component of the algorithm, stated in Lemma 4.31 and Lemma 4.35. □

Relying on the above lemma, we conclude our main theorem of this chapter.

Theorem 4.37

Let $\mathcal{DL}$ be any (finitely) Θ-forest-friendly DL $\mathcal{DL}$, extending $\mathcal{ALCO}\Theta^{\cap}$, that has the (finite) $\mathcal{DL}$-KB-satisfiability problem decidable in time $\mathsf{SAT}_{\mathcal{DL}}(\cdot)$. Then there exists a polynomial, an exponential, and a doubly-exponential function poly, exp, and expexp, such that the (finite) UCQ-entailment problem over $\mathcal{DL}$-KB is decidable, for an input $\mathcal{DL}$-KB $\mathcal{K}$, and a UCQ q, in time bounded by

- $\mathsf{expexp}(|\mathcal{K}| + |q|) \cdot \mathsf{SAT}_{\mathcal{DL}}(\mathsf{exp}(|\mathcal{K}|+|q|))$ if Θ is non-empty, and
- $\mathsf{exp}(|\mathcal{K}| + |q|) \cdot \mathsf{SAT}_{\mathcal{DL}}(\mathsf{poly}(|\mathcal{K}|+|q|))$ if Θ is empty.

The most important application of our work is when the (finite) knowledge base satisfiability problem for $\mathcal{DL}$ is ExpTime-complete. Then the function $\mathsf{exp}(|\mathcal{K}|+|q|) \cdot \mathsf{SAT}_{\mathcal{DL}}(\mathsf{poly}(|\mathcal{K}|+|q|))$ reduces to a single exponential, and hence, we have the following corollary (where the matching lower bounds follow either from $\mathcal{ALC}$ [Sch91, Prop. 3], or from querying $\mathcal{ALCI}$ [Lut08a, Thm. 2] and $\mathcal{ALC}^{\mathsf{Self}}$ [BR22, Thm. 8.2]).

Corollary 4.38

Let $\mathcal{DL}$ be as in Theorem 4.37 and assume that it has ExpTime-complete (finite) knowledge-base satisfiability problem. Then the (finite) UCQ entailment problem over $\mathcal{DL}$-KBs is 2ExpTime-complete for non-empty Θ and ExpTime-complete for empty Θ.

As the last remark: any positive existential query q can be transformed into a UCQ $\bigvee_i q_i$, in which the total number of queries is exponential w.r.t $|q|$, but the size of each q_i is only polynomial w.r.t $|q|$. Thus the 2nd step of Procedure 4 can be implemented to work in time doubly-exponential w.r.t $|q|+|\mathsf{N}|$. This yields us 2ExpTime upper bound for (finite) PEQ-entailment over any logics mentioned in Corollary 4.38. The matching lower bound holds already for $\mathcal{ALC}$ [Ov14, Thm. 1].

Corollary 4.39

For any description logic $\mathcal{DL}$ satisfying conditions from Corollary 4.38 we have that the (finite) PEQ entailment problem over $\mathcal{DL}$-KBs is 2ExpTime-complete.

By reasoning in a similar way, one can obtain similar results for the query entailment problem for the case when the size of the problem is measured in terms of *data complexity*. We present the algorithm first.

Procedure 4: (Finite) UCQ non-entailment over (finitely) Θ-forest-friendly DLs ext. $\mathcal{ALC}\Theta^{\cap}$.

```
Input: ABox $\mathcal{A}$.
Fixed: A UCQ $q \coloneqq \bigvee_{i=1}^{m} q_i$ and an $\mathcal{DL}$-TBox $\mathcal{T}$, where $\mathcal{DL} \in \mathscr{C}_{\Theta\mathsf{fr}}$ ($\mathcal{DL} \in \mathscr{C}^{\mathsf{fin}}_{\Theta\mathsf{fr}}$ in the finite case).
1 Guess $(\mathsf{ind}(\mathcal{A}), \Theta)$-super spoilers the queries $\mathcal{K}^{\lightning\star}_{q_1}, \ldots, \mathcal{K}^{\lightning\star}_{q_m}$ for $q_1, \ldots, q_m$. // In NP by Lemma 4.31.
2 If $\mathcal{K} \cup \bigcup_i \mathcal{K}^{\lightning\star}_{q_i}$ is (finitely) satisfiable return False and return True otherwise.
      // Data-complexity the same as for checking the (finite) satisfiability of $\mathcal{K}$.
```

After reading the above algorithm, the following theorem follows quite easily.

Theorem 4.40

Suppose that $\mathcal{DL}$ is a (finitely) Θ-forest-friendly description logic that extends $\mathcal{ALC}\Theta^{\cap}$ and has NP-complete (in terms of data-complexity) satisfiability problem. Then the (finite) PEQ-entailment problem over $\mathcal{DL}$-KB (in terms of data-complexity) is coNP-complete.

Proof sketch. Correctness of Procedure 4 can be shown as in the proof of Lemma 4.36, while the bounds on the running time of the algorithm follow from the NP-completeness of the satisfiability problem for $\mathcal{DL}$, and coNP-hardness result for $\mathcal{ALC}$ by Schaerf [Sch93, Thm. 3.2]. □

5

Sufficient Conditions for Forest-Friendliness and Their Applications

Contents

Motivation and Our Contribution

In recent years, it has become apparent that various modelling features of DLs affect the complexity of conjunctive query entailment in a rather strong sense. Let us focus on the most popular DL, namely $\mathcal{ALC}$. It was first shown independently by Ortiz et al. [OvE08, Thm. 6] and by Lutz [Lut08a, Thm. 2] that CQ entailment is exponentially harder than the consistency problem for $\mathcal{ALC}$ extended with inverse roles ($\mathcal{I}$). Shortly after, a combination of transitivity and role hierarchies ($\mathcal{SH}$) was shown as another culprit of higher worst-case complexity of reasoning [ELOv09, Thm. 1]. Finally, also nominals ($\mathcal{O}$) turned out to be problematic [NOv16, Thm. 9]. Nevertheless, there are also more benign DL constructs regarding the complexity of CQ entailment. Examples are counting ($\mathcal{Q}$) [Lut08b, Thm. 1] (the complexity stays the same even for expressive arithmetical constraints [BBR20, Thm. 21]), role-hierarchies alone ($\mathcal{H}$) [EOv12, Cor. 3] or a tamed use of higher-arity relations [Bed21a, Thm. 20]. Despite the considerable effort in establishing the complexity of the query entailment problem over $\mathcal{ALC}$ extended with various primitive features, for many of the extensions the precise computational complexity is still unknown. To be more precise, the complexity of CQ entailment problem for $\mathcal{ALC}$ extended with any of safe-boolean role combinations (b), transitive closure of roles ($\cdot^+$), regular role expressions ($\cdot_{\mathsf{reg}}$), fixed points (μ), or the self operator (Self) was *not known*. Doubly-exponential upper bounds for querying any of the mentioned logics follow from existing results, *e.g.* from the results on the $\mathcal{Z}$ family of DLs [CEO09] or from the results on the guarded-negation fixpoint logic [BtCS15]. However, it is not at all clear whether such complexity bounds are tight. And even more intriguingly, we do not know whether the previously established upper bounds for CQ entailment can be adapted to a slightly more general class of queries, namely to the case of entailment of *unions of* conjunctive queries (UCQs). While it is known that for the case of $\mathcal{ALCH}$, the complexities of querying with CQs and UCQs coincide [Ort10, Thm. 6.5.1], the case of $\mathcal{ALCQ}$ is not yet resolved. There is no reason to believe that for some DL the complexity of UCQ entailment differs from the complexity of

UCQ entailment, but jumping to the class of even more expressive queries, like positive existential queries (PEQs) or conjunctive regular path queries (CRPQs) result in an increase of complexity [Ov14, Thm. 1].

The aim of this section is to provide *robust* answers to the questions above, and more generally, to clarify why the query entailment problem for certain extensions of $\mathcal{ALC}$ is computationally harder than their knowledge base satisfiability problem while for others the complexities coincide. Recall that in Section 3.3 we introduced the broad class of (finitely) Θ-forest friendly description logics, and in Section 4.6 we provided an efficient algorithm for query entailment over such logics. Unfortunately, the definition of Θ-forest friendly DLs (Definition 3.11) is quite involved and it is not a priori clear how to check whether a freshly defined logic is or is not (finitely) Θ-forest friendly. To simplify checking forest-friendliness, we propose several sufficient conditions based on novel model-theoretic notions of unravellings or "pumping". With such conditions in hand and a bit of luck, the relatively tedious task of deciding forest-friendliness boils down to routine proofs involving structural induction. What is more, each of our constructions is supplemented with a handy list of properties preserved by our unravellings or pumping, that can simplify such reasoning even further. Together with a novel lower bound for $\mathcal{ALC}\mathsf{Self}$ tackled in the next section, our results yield a complete classification of the "querying-satisfiability trade-off" for $\mathcal{ALC}$ extended with popular primitive[1] features. This is summarised by the table below; references for logics having harder query entailment than satisfiability were given above.

	Feature θ with name	$\mathcal{ALC}\theta^{\cap} \in \mathscr{C}_{\emptyset\mathsf{fr}}$?	SAT=CQEnt?
good	functionality $\mathcal{F}$ and various counting: $\mathcal{N}/\mathcal{Q}/\mathcal{SCC}$	✓ [new!]	
	trans. closure $\cdot^*$, regular expr. $\cdot_{\mathsf{reg}}$, fixed-points μ		
	role hierar. $\mathcal{H}$, safe boolean comb. of roles b		
bad	inverses of roles $\mathcal{I}$	✗	
	nominals $\mathcal{O}$		
	transitivity $\mathcal{S}$, complex role inclusions $\mathcal{R}$		
bad	self-loops Self	✗ [new!]	

The features $\mathcal{F}$, $\mathcal{N}$, $\mathcal{Q}$, $\mathcal{H}$ and b are subsumed by $\mathcal{SCC}$, while $\cdot^*$ and $\cdot_{\mathsf{reg}}$ are subsumed by μ [BCM+03, p. 204]. Hence, $\mu\mathcal{ALCSCC}$ is a super-logic that comprises all the positive features together.

Overview of the Chapter and Prerequisites

We assume familiarity with Chapter 3 and its prerequisites. Material from Chapter 4 is not required.

Our agenda is as follows. In Sections 5.1–5.3 we focus on $\emptyset$-forest-friendly logics, and in Section 5.4 we focus on Θ-forest-friendly logics for Θ containing the "inverse-role" feature I. We first introduce a concept of *forward unravellings* that turn interpretations into (usually infinite) $\emptyset$-forests. If the satisfaction of $\mathcal{DL}$-KBs is preserved under such unravellings, then $\mathcal{DL}$ is $\emptyset$-forest-friendly. To obtain suitable conditions in the finite model reasoning scenario, we develop the notion of *scattered forward unravellings*. Their main advantage is that the scattered forward unravellings of finite interpretations are indeed finite whilst from the queries' view point they are indistinguishable from infinite $\emptyset$-forests. We conclude that $\mathcal{DL}$ is finitely $\emptyset$-forest-friendly whenever the satisfaction of any $\mathcal{DL}$-KB is preserved under scattered forward unravellings. Then we discuss the case of logics expressing statistical constraints over the domain, and see how the notion of scattered forward unravellings can be adjusted to them. We conclude with an appropriate model transformation[2] that transforms (finite) interpretations into ones that are locally-Θ-forest-like (for Θ containing I). This generalises the model-pumping construction by Baader et al. [BBR19, Sec. 5.2] and significantly simplifies developments of Ibáñez-García et al. [ILS14, Sec. 3] and Otto [Ott12, Sec. 3.2].

[1]Note that we do not consider role axioms like role disjointness or reflexivity from $\mathcal{SROIQ}$ [HKS06] as *primitives*, as they can be expressed with other features.

[2]Developed after a series of email exchanges with Ian Pratt-Hartmann, and is also included in his recent book [PH23].

5.1 Forward Unravellings

Forward unravellings make interpretations $\emptyset$-forest-shaped. Such a notion differs only slightly from classical unravellings [BHLS17, Def. 3.21]. The crucial difference is that forward unravellings preserve substructures induced by the selected named individuals. More precisely, the sequences starting with two named individuals are excluded from the domain and roles linking named individuals are assigned "manually"(*cf.* the last item from Definition 5.1). Before reading the definition we suggest consulting Figure 5.1.

Definition 5.1 Let $\mathsf{N} \subseteq \mathbf{N_I}$ be a non-empty set of names. An **N-rooted forward unravelling** of an interpretation $\mathcal{I}$ is an interpretation $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}} := (\Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}, \cdot^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}})$ satisfying all four conditions below:

1. The domain of $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ consists of all non-empty words of elements from $\Delta^{\mathcal{I}}$, except those, where the first two elements are named or where two consecutive elements are "disconnected". In symbols: $\Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}} := (\Delta^{\mathcal{I}})^{+} \setminus$

$$\Big(\mathsf{N}^{\mathcal{I}} \cdot \mathsf{N}^{\mathcal{I}} \cdot (\Delta^{\mathcal{I}})^{*} \cup (\Delta^{\mathcal{I}})^{*} \cdot \{\mathrm{de} \mid \mathrm{d}, \mathrm{e} \in \Delta^{\mathcal{I}}, \mathsf{Rol}_{\mathcal{I}}(\mathrm{d}, \mathrm{e}) = \emptyset\} \cdot (\Delta^{\mathcal{I}})^{*} \Big).$$

   For convenience, we do not syntactically distinguish elements from $\Delta^{\mathcal{I}}$ and single-letter words from $\Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$. In particular, this means that $\Delta^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$. We often use last, *i.e.* the function that maps a sequence to its last element.
2. For all individual names $\mathsf{a} \in \mathsf{N}$ we have $\mathsf{a}^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}} = \mathsf{a}^{\mathcal{I}}$ and for all other names $\mathsf{a} \in (\mathbf{N_I} \setminus \mathsf{N})$ there is some $\mathsf{b} \in \mathsf{N}$ fulfilling $\mathsf{a}^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}} = \mathsf{b}^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$.
3. For any concept name $\mathrm{A} \in \mathbf{N_C}$ the equality $\mathrm{A}^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}} = \{w \mid \mathsf{last}(w) \in \mathrm{A}^{\mathcal{I}}\}$ holds.
4. For all role names $r \in \mathbf{N_R}$ we define $r^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$ as the intersection of $\Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}} \times \Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$ and the set

$$\Big(r^{\mathcal{I}} \cap (\mathsf{N}^{\mathcal{I}} \times \mathsf{N}^{\mathcal{I}})\Big) \cup \Big\{(w, w \cdot \mathrm{d}) \mid (\mathsf{last}(w), \mathrm{d}) \in r^{\mathcal{I}}\Big\}.$$

   Intuitively, the above set is composed of two parts, where the first component preserves N-named parts, while the other one preserves roles, mimicking the classical unravelling.

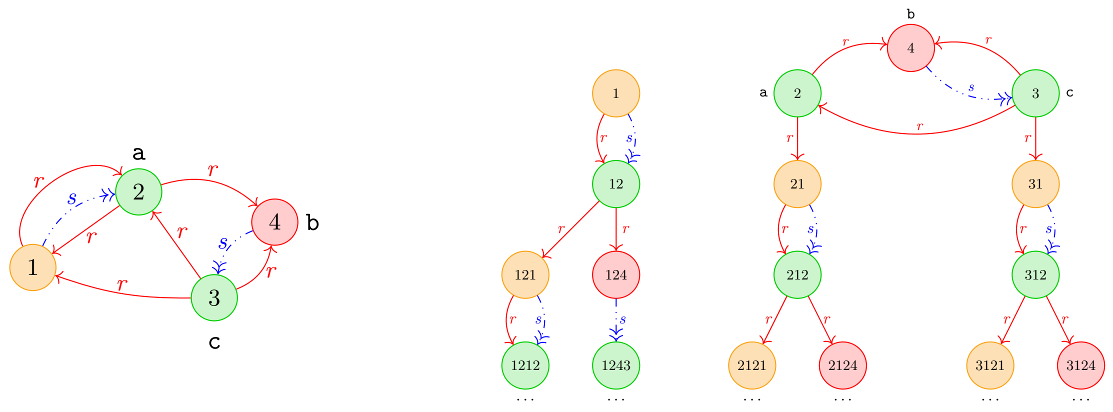


Figure 5.1: An interpretation $\mathcal{I}$ (left) with a fragment of its $\{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$*-rooted forward unravelling* $\mathcal{I}_{\{\mathsf{a},\mathsf{b},\mathsf{c}\}}^{\vec{\omega}}$ (right). Note that $\mathcal{I}_{\{\mathsf{a},\mathsf{b},\mathsf{c}\}}^{\vec{\omega}}$ is a forest with two connected components. Nodes of the same colour satisfy the same atomic concepts.

It is not difficult to see that forward unravellings produce forest-shaped interpretations that can be homomorphically mapped to the original structures.

Lemma 5.2 Let $\mathcal{I}$ be an interpretation, $\mathsf{N} \subseteq \mathbf{N_I}$ be a set of names and let $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ be any N-rooted forward unravelling of $\mathcal{I}$. Then $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ is N-rooted $\emptyset$-forest-shaped and the function $\mathsf{last}\colon \Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}} \to \Delta^{\mathcal{I}}$ is an N-homomorphism from $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ to $\mathcal{I}$.

Proof. $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ is an N-rooted $\Delta^{\mathcal{I}}$-$\emptyset$-forest. To see that $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ is a $\emptyset$-forest we observe that its domain is prefix-closed (follows from Item 1 of Definition 5.1) and that it satisfies all forests' criteria on roles (by Item 4 of Definition 5.1). The N-rootedness follows from Item 2 of Definition 5.1. For the second claim, it suffices to show that last satisfies every item from the definition of a homomorphism. The preservation of individual names from N by last follows immediately from Item 2 of Definition 5.1. Similarly, the preservation of concepts follows from Item 3 of Definition 5.1. Lastly, to show that last preserves roles, take any two elements $u, v \in \Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$ satisfying $(u, v) \in r^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$. We aim to prove that $(\mathsf{last}(u), \mathsf{last}(v)) \in r^{\mathcal{I}}$ holds. There are two cases to consider: either both u, v are N-named or at least one of $u \neq v$ is not named. For the first case, note that u, v are single elements. Thus, we infer that $u = \mathsf{last}(u)$ and $v = \mathsf{last}(v)$ hold and by the first part of the equation in Item 4 of Definition 5.1 we conclude $(\mathsf{last}(u), \mathsf{last}(v)) \in r^{\mathcal{I}}$. Finally, if one of u, v is not named, we have that $v = u \cdot \mathrm{d}$ holds for some $\mathrm{d} \in \Delta^{\mathcal{I}}$. By applying the second part of the equation in Item 4 of Definition 5.1, we infer $(\mathsf{last}(u), \mathsf{last}(v)) \in r^{\mathcal{I}}$. □

This leads us to a sufficient condition for a description logic to be $\emptyset$-forest-friendly.

Definition 5.3 A description logic $\mathcal{DL}$ is **preserved under forward unravellings** if for all $\mathcal{DL}$-knowledge-bases $\mathcal{K}$ we have that $\mathcal{I} \models \mathcal{K}$ implies $\mathcal{I}_{\mathsf{ind}(\mathcal{K})}^{\vec{\omega}} \models \mathcal{K}$.

Membership of such logics to the class of $\emptyset$-forest-friendly logics follows rather immediately.

Theorem 5.4

If a description logic $\mathcal{DL}$ is preserved under forward unravellings then $\mathcal{DL}$ is $\emptyset$-forest-friendly.

Proof. By unfolding Definition 3.11 and Definition 3.10, it suffices to take any satisfiable $\mathcal{DL}$-KB $\mathcal{K}$, its model $\mathcal{I}$ and any $n \in \mathbb{N}$ and show that there exists an $(n, \mathsf{ind}(\mathcal{K}), \emptyset)$-forest model $\mathcal{J} \models \mathcal{K}$, whose all n-neighbourhoods can be $\mathsf{ind}(\mathcal{K})$-homomorphically-mapped to $\mathcal{I}$. Take $\mathcal{J}$ to be $\mathcal{I}_{\mathsf{ind}(\mathcal{K})}^{\vec{\omega}}$, and take any of its n-neighbourhoods $\mathcal{J}'$. Since last is an $\mathsf{ind}(\mathcal{K})$-homomorphism from $\mathcal{J}$ to $\mathcal{I}$, it is also an $\mathsf{ind}(\mathcal{K})$-homomorphism from $\mathcal{J}'$ to $\mathcal{I}$. This concludes the proof. □

We conclude by presenting several useful properties of forward unravellings that helps for a quick test to see whether the modelhood of some knowledge base is preserved.

Property 5.5. Let $\mathcal{I}$ be an interpretation, let $\mathsf{N} \subseteq \mathbf{N_I}$ be a set of names and let $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ be any N-rooted forward unravelling of $\mathcal{I}$. Then the following conditions are satisfied:

(A) The interpretations $\mathcal{I}$ and $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ restricted to all N-named elements are isomorphic.

(B) For any concept name $\mathrm{A} \in \mathbf{N_C}$ we have that $\mathrm{A}^{\mathcal{I}}$ is non-empty if and only if $\mathrm{A}^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$ is. Similarly, for any role name $r \in \mathbf{N_R}$ we have that $r^{\mathcal{I}}$ is non-empty if and only if $r^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$ is.

(C) For any $w \in \Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$ we have $\mathsf{Conc}_{\mathcal{I}}(\mathsf{last}(w)) = \mathsf{Conc}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}(w)$. Moreover, for all v satisfying $(w, v) \in r^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$ for some $r \in \mathbf{N_R}$, we have $\mathsf{Rol}_{\mathcal{I}}(\mathsf{last}(w), \mathsf{last}(v)) = \mathsf{Rol}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}(w, v)$.

(D) For any $w \in \Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$ we have that w in $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ and $\mathsf{last}(w)$ in $\mathcal{I}$ have the same number of successors satisfying the same roles and concepts. Formally, for any *non-empty* set of role names $\mathsf{R} \subseteq \mathbf{N_R}$ and any set of concept names $\mathsf{C} \subseteq \mathbf{N_C}$ we have that the cardinalities of the two sets below coincide:

$$\left\{ v \in \Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}} \mid \mathsf{Conc}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}(v) = \mathsf{C} \text{ and } \mathsf{Rol}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}(w, v) = \mathsf{R} \right\},$$

$$\left\{ \mathrm{e} \in \Delta^{\mathcal{I}} \mid \mathsf{Conc}_{\mathcal{I}}(\mathrm{e}) = \mathsf{C} \text{ and } \mathsf{Rol}_{\mathcal{I}}(\mathsf{last}(w), \mathrm{e}) = \mathsf{R} \right\}.$$

(E) For any $w \in \Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$ we have that w and $\mathsf{last}(w)$ are *directed-path-equivalent*, meaning that:

- If ρ with $\rho_1 := w$ is a (possibly infinite) directed path in $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ then ρ', defined as $\rho'_i := \mathsf{last}(\rho_i)$ for all i, is a directed path in $\mathcal{I}$ such that for all i we have $\mathsf{Conc}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}(\rho_i) = \mathsf{Conc}_{\mathcal{I}}(\rho'_i)$, and for all $i > 1$ we have $\mathsf{Rol}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}(\rho_{i-1}, \rho_i) = \mathsf{Rol}_{\mathcal{I}}(\rho'_{i-1}, \rho'_i)$.
- If ρ with $\rho_1 := \mathsf{last}(w)$ is a (possibly infinite) directed path in $\mathcal{I}$ then ρ' defined as:
  (i) $\rho'_1 := w$, and
  (ii) $\rho'_i := \rho_i$ if both ρ'_{i-1} and ρ_i are N-named, and $\rho'_i := \rho'_{i-1}\rho_i$ otherwise for all remaining i,
  is a directed path in $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ such that for all i we have $\mathsf{Conc}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}(\rho'_i) = \mathsf{Conc}_{\mathcal{I}}(\rho_i)$, and for all $i > 1$ we have $\mathsf{Rol}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}(\rho'_{i-1}, \rho'_i) = \mathsf{Rol}_{\mathcal{I}}(\rho_{i-1}, \rho_i)$.

Proof. We will proceed with each of the items separately.

(A) The last function restricted to $\mathsf{N}^{\mathcal{I}}$, is actually the identity function, so it suffices to show that it is also a homomorphism. This already follows from Lemma 5.2.

(B) For the first part, take any element $\mathrm{d} \in \mathrm{C}^{\mathcal{I}}$. Then $\mathrm{d} \in \mathrm{C}^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$. Similarly, if $w \in \mathrm{C}^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$ then $\mathsf{last}(w) \in \mathrm{C}^{\mathcal{I}}$. Hence $\mathrm{C}^{\mathcal{I}}$ is non-empty if and only if $\mathrm{C}^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$ is. For the second part, let us take $(\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}$. Then if both d, e are N-named, then by construction $(\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$. Otherwise we have that $(\mathrm{d}, \mathrm{de}) \in r^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$. For the other direction assume that $(w, v) \in r^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$ holds. Then we again distinguish two cases: if both w, v are N-named then $w, v \in \Delta^{\mathcal{I}}$ and we infer $(w, v) \in r^{\mathcal{I}}$ by the manual assignment of roles between named elements. Otherwise by definition we have $(\mathsf{last}(w), \mathsf{last}(v)) \in r^{\mathcal{I}}$, yielding the desired equivalence.

(C) Depending on whether both w and v are N-named or not, the result follows either from the first or from the second part of the equation in Item 4 of Definition 5.1. The equality between $\mathsf{Conc}_{\mathcal{I}}(\mathsf{last}(w))$ and $\mathsf{Conc}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}(w)$ follows from Item 3 of Definition 5.1.

(D) Fix C, R, w and $\mathrm{d} := \mathsf{last}(w)$ as in the statement of Property 5.5. We first show that last is a bijection between the sets $\mathsf{last}[A] := \{\mathsf{last}(v) \mid v \in A\}$ and A defined below:

$$\left\{ v \in \Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}} \mid \mathsf{C} = \mathsf{Conc}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}(v) \text{ and } \mathsf{R} = \mathsf{Rol}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}(w, v) \right\},$$

Surjectivity is obvious, so we focus on injectivity only. Take any $u, v \in A$ that are mapped to the same element by last. This implies that $u = u_0\mathrm{e}$ and $v = v_0\mathrm{e}$ for some (possibly empty) words $u_0, v_0 \in (\Delta^{\mathcal{I}})^*$ and $\mathrm{e} \in \Delta^{\mathcal{I}}$. There are three cases to consider:

- Both u_0, v_0 are non-empty. Then u_0, v_0 are equal to w as $(w, v) \in r^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$ for some $r \in \mathsf{R}$ and the second part of the equation in Item 4 of Definition 5.1 holds. Thus $u = v$.
- Both u_0, v_0 are empty. Then obviously $v = u$ holds.
- One of u_0, v_0 is empty and the other one is not. W.l.o.g assume $u_0 = \varepsilon$ and $v \neq \varepsilon$. Then by construction of $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ (more precisely the first part of the equation in Item 4 of Definition 5.1) we infer that w is N-named and that u is N-named, and that both w and u are single-element sequences. Since v has length at least two, it is not named. Thus by the second part of the equation in Item 4 of Definition 5.1, we infer $w = v_0$. But this means that v is a two-element sequence composed of N-named elements, which were excluded from the domain of $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$, *cf.* Item 1 of Definition 5.1. A contradiction. So such a case is not possible.

Thus last is indeed a bijection between A and $\mathsf{last}[A]$.

Our next claim is that the identity function is the bijection between $\mathsf{last}[A]$ and B given below. From that we conclude $|A| = |B|$ (and thus the whole proof) by transitivity of $=$.

$$B := \left\{ \mathrm{e} \in \Delta^{\mathcal{I}} \mid \mathsf{C} = \mathsf{Conc}_{\mathcal{I}}(\mathrm{e}) \text{ and } \mathsf{R} = \mathsf{Rol}_{\mathcal{I}}(\mathsf{last}(w), \mathrm{e}) \right\}.$$

We show that $\mathsf{last}[A] \subseteq B$ and $B \subseteq \mathsf{last}[A]$. The first inclusion follows from the way we defined roles in the unravelling, *cf.* Item 4 of Definition 5.1. For the other inclusion we distinguish the cases depending on whether w is N-named or not.

- w is not N-named.
  Take any $\mathrm{e} \in B$. Then we have $(\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}$ for some $r \in \mathsf{R}$, and hence we have that $(w, w\mathrm{e}) \in r^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$ (by the second item of Item 4 of Definition 5.1). Applying Item (C) of Property 5.5 we infer that $\mathrm{e} \in A$, and thus $\mathrm{e} \in \mathsf{last}[A]$.

- If w is N-named, then $w = \mathrm{d}$.
  Again, take any $\mathrm{e} \in B$. Then we have $(\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}$ for some $r \in \mathsf{R}$. If e is N-named then e is also N-named in $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ (by Item 2 of Definition 5.1). Thus, by the first part of the equation in Item 4 of Definition 5.1, we know that $(\mathrm{d}, \mathrm{e}) = (\mathsf{last}(w), \mathsf{last}(\mathrm{e}))$ belongs to $r^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$. By Item (C) of Property 5.5 we infer that $\mathrm{e} \in A$, and hence $\mathrm{e} \in \mathsf{last}[A]$. Otherwise, if e is not N-named, we apply the same reasoning as in the case of an unnamed w. This concludes the proof.

(E) Immediate from Item 1 of Definition 5.1 and Item (C) of Property 5.5.

□

Let us explain how Property 5.5 can actually be used for freshly defined DLs.

Remark 5.6. One can employ Property 5.5 to show that forward unravellings, among other properties, preserve: satisfaction of ABoxes (via Item (A)), existence of "unary-types" and "role-types" (via Item (B)), cardinality constraints on the total number of successors (via Item (D)), role hierarchies and safe boolean role combinations (via Item (C)), regular role expressions and fixed points (via Item (E)). The later can be shown by induction on the complexity of the formula and fixed-point approximations [BW18, p. 5]. It is an exercise to see that our unravellings usually do not preserve inverse roles, nominals, transitivity or self-loops.

Hence, as a corollary we can infer:

Corollary 5.7

Let Θ be any set of features from $\Theta \subseteq \{\mathcal{H}, b, \mathcal{F}, \mathcal{N}, \mathcal{Q}, \mathcal{SCC}, \mathsf{reg}, \mu\}$. Then $\mathcal{ALC}\Theta$ has an ExpTime-complete satisfiability problem, and is preserved under forward unraveling. Hence, the entailment problem for UCQs over $\mathcal{ALC}\Theta$-knowledge-bases is also ExpTime-complete.

Proof. It suffices to establish the result for $\mu\mathcal{ALCSCC}$ as it combines all the features mentioned above. First, $\mu\mathcal{ALCSCC}$ can be quite easily encoded into the graded μ-calculus with polynomial inequalities [HS22, Ex. 2.1(5)] (that subsumes Presburger μ calculus, a notational variant of $\mu\mathcal{ALCSCC}$), for which the knowledge base problem is ExpTime-complete: it follows by a combination the result by Kupke et al. [KPS22, Thm. 8.7] and the ExpTime-completeness of the so-called strict one-step satisfiability problem shown by Hausmann et. al [HS22, Remark 4.5]. Second, the fact that $\mu\mathcal{ALCSCC}$ is preserved under forward unraveling follows by Remark 5.6. Thus, by Theorem 5.4 we know that $\mu\mathcal{ALCSCC}$ is $\emptyset$-forest-friendly. Hence, by Corollary 4.38 we conclude ExpTime-completeness of its query entailment problem. □

5.2 Scattered Forward Unravellings

For a certain class of logics, namely for DLs preserved under forward unravellings, forward unravellings produce $\emptyset$-forest (counter)models out of (possibly) non-forest ones. The presented construction is sufficient to employ Lutz's spoiler method (described in Chapter 4) over the class of *arbitrary* structures, but it is useless once we want to achieve results in the finite-model scenario. The reason is trivial: forward unravellings of finite interpretations are nearly always infinite. To find a suitable counterpart of forward unravellings in the finite realm, we design the notion of *scattered forward unravellings.* Its aim is, given a threshold n, a set of names N, and a finite interpretation $\mathcal{I}$, to turn $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ into a finite $(n, \mathsf{N}, \emptyset)$-forest (consult again Definition 3.8 if needed). The construction is a little bit involved and relies on cutting out finitely many tree-like components from $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ and then glueing them together mimicking "parent-to-leaf connections", so that any neighbourhood of size n from the desired structure is homomorphically-equivalent to some n-neighbourhood from the forward unravelling. The novel model construction presented here took some inspiration from similar constructions, namely the ones by Bednarczyk and Kieroński [BK22, Sec. 3], by Otto [Ott04, Sec. 4.2], and by Emerson and Halpern [EH85, Sec. 3.5].

We start the construction by defining basic building blocks, called here the *components.*

Definition 5.8 Fix a number $n \in \mathbb{N}$, a finite set of names $\mathsf{N} \subseteq \mathbf{N_I}$ and a finite interpretation $\mathcal{I}$ whose domain is linearly ordered. We are going to select certain auxiliary substructures of $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$.

- The **(N, n)-king component $\mathcal{I}_{\text{♚}}$** of $\mathcal{I}$ is obtained from $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ by a restriction to all words from $\Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$ of length at most $2n{+}1$.
- Call $\mathrm{d} \in \Delta^{\mathcal{I}}$ *deep* whenever there is a word $w := u \cdot \mathrm{d}$ in $\Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}$ such that $|u| > n$. The lexicographically smallest such word w is called the *deep realisation of* d. The **(N, n)-pawn component $\mathcal{I}_{\text{♟},\mathrm{d}}$ of a deep element** d is obtained by restricting $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ to all words of the from $w \cdot v$, where w is the deep realisation of d and $|v| \leq 2n$.

Consult Figure 5.2 for a visualisation. We use $\mathscr{I}(n, \mathsf{N}, \mathcal{I})$ to denote the set of all components of $\mathcal{I}$, and $L(n, \mathsf{N}, \mathcal{I})$ to denote the maximal number of leaves among all components in $\mathscr{I}(n, \mathsf{N}, \mathcal{I})$. We assign a number $1 \leq \ell \leq L(n, \mathsf{N}, \mathcal{I})$ to any leaf in any component of $\mathscr{I}(n, \mathsf{N}, \mathcal{I})$ and refer to it as this component's *ℓ-th leaf*. For future purposes, we define a function $\mathsf{orig}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}} \colon (\bigcup \mathscr{I}(n, \mathsf{N}, \mathcal{I})) \to \mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ that maps an element from any component to the element from whom it originated. We also employ a mapping $\mathsf{orig}_{\mathcal{I}}$ that maps elements d to $\mathsf{last}(\mathsf{orig}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}(\mathrm{d}))$, *i.e.* $\mathsf{orig}_{\mathcal{I}} := \mathsf{orig}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}} \circ \mathsf{last}$.

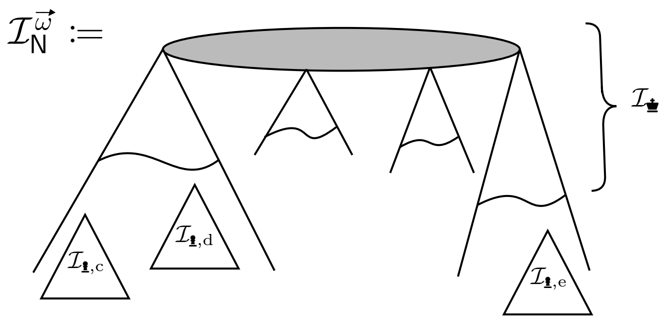

$\mathcal{I}_{\mathsf{N}}^{\vec{\omega}} :=$
$\mathcal{I}_{\text{♚}}$
$\mathcal{I}_{\text{♟},\mathrm{c}}$
$\mathcal{I}_{\text{♟},\mathrm{d}}$
$\mathcal{I}_{\text{♟},\mathrm{e}}$

Figure 5.2: Visualization of the selection of components out of the forward unravelling of $\mathcal{I}$.

The king component $\mathcal{I}_{\text{♚}}$ is simply the forward unravelling cut off after $2n{+}1$ steps, while each pawn component $\mathcal{I}_{\text{♟},\mathrm{d}}$ is a subtree of depth $2n$ rooted at some deep enough copy of some element of $\Delta^{\mathcal{I}}$. We stress that $\mathcal{I}_{\text{♚}}$ is a finite N-rooted $\emptyset$-forest, while all $\mathcal{I}_{\text{♟},\mathrm{d}}$ are $\emptyset$-trees.

Definition 5.9 Take $(n, \mathsf{N}, \mathcal{I})$ as in Definition 5.8. The set $\mathscr{J}(n, \mathsf{N}, \mathcal{I})$ of **copies of components** is composed of the king component $\mathcal{I}_{\text{♚}}$ from $\mathscr{I}(n, \mathsf{N}, \mathcal{I})$ and *isomorphic copies* $\mathcal{I}_{\text{♟},\mathrm{d},h}^{(\ell,\mathrm{s})}$ *of pawn components* $\mathcal{I}_{\text{♟},\mathrm{d}}$ from $\mathscr{I}(n, \mathsf{N}, \mathcal{I})$ that are indexed by: a deep element $\mathrm{d} \in \Delta^{\mathcal{I}}$, $h \in \{0, 1\}$, $1 \leq \ell \leq L(n, \mathsf{N}, \mathcal{I})$, and $\mathrm{s} \in (\Delta^{\mathcal{I}} \cup \{\text{♚}\})$, that are called, respectively, the *target*, the *hue*, the *leaf number*, and the *source* of a component. We often employ $*$ that serves as a wildcard which can be used in place of various parameters, whenever such parameters are of no importance. For convenience, we will also speak about the "hue" of a domain element, meaning the "hue" of the unique copy of the pawn component to which such an element belongs.

The main idea behind the quite loaded notation of $\mathcal{I}_{\text{♟},\mathrm{d},h}^{(\ell,\mathrm{s})}$ is that the root of $\mathcal{I}_{\text{♟},\mathrm{d},h}^{(\ell,\mathrm{s})}$ "can serve as a d-witness for the ℓ-th leaf of any copy of $\mathcal{I}_{\text{♟},\mathrm{s}}$ of hue $1{-}h$". Note that it follows immediately from finiteness of $\Delta^{\mathcal{I}}$ and n, that the set $\mathscr{J}(n, \mathsf{N}, \mathcal{I})$ is finite.

We are finally ready to present the definition of n-scattered forward unravellings.

Definition 5.10 Let $(n, \mathsf{N}, \mathcal{I})$ are as in Definition 5.8. The **(n, N)-scattered-forward-unravelling $\mathcal{I}_{\mathsf{N}}^{\vec{n}}$** is obtained by taking the disjoint sum of structures from $\mathscr{J}(n, \mathsf{N}, \mathcal{I})$, and then extending the interpretation of each role name r, with a (u, v) of $\mathcal{I}_{\mathsf{N}}^{\vec{n}}$, whenever $(\mathsf{orig}_{\mathcal{I}}(u), \mathsf{orig}_{\mathcal{I}}(v)) \in r^{\mathcal{I}}$ and either

(i) u is the ℓ-th leaf of $\mathcal{I}_{\text{♚}}$, and v is the root of $\mathcal{I}_{\text{♟},\mathsf{orig}_{\mathcal{I}}(v),0}^{(\ell,\text{♚})}$, or

(ii) u is the ℓ-th leaf of $\mathcal{I}_{\text{♟},\mathrm{e},h}^{(*,*)}$ for some $\mathrm{e} \in \Delta^{\mathcal{I}}$ and $h \in \{0, 1\}$, and v is the root of $\mathcal{I}_{\text{♟},\mathsf{orig}_{\mathcal{I}}(v),1-h}^{(\ell,\mathrm{e})}$.

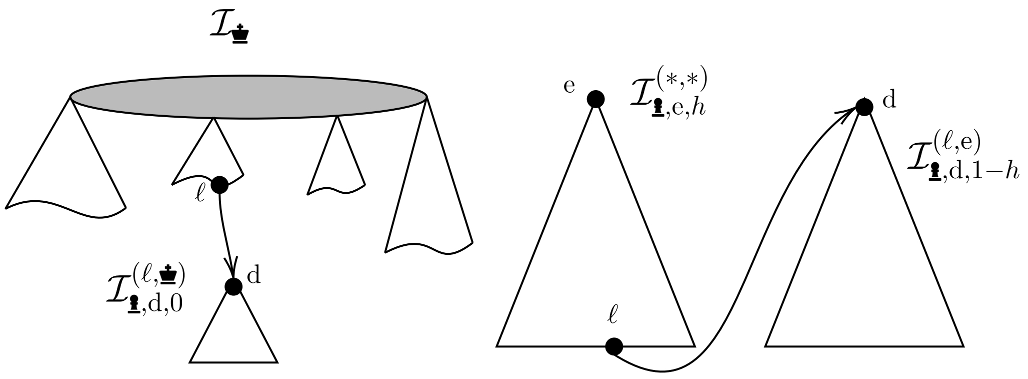

$\mathcal{I}_{♚}$
ℓ
$\mathcal{I}^{(\ell,♚)}_{♟,\mathrm{d},0}$
d
e
$\mathcal{I}^{(*,*)}_{♟,\mathrm{e},h}$
ℓ
d
$\mathcal{I}^{(\ell,\mathrm{e})}_{♟,\mathrm{d},1-h}$

Figure 5.3: Linking components in the construction of (n, N)-scattered unravelling of $\mathcal{I}$, *i.e.* visualisation of cases (i) and (ii) from Definition 5.10. The left picture illustrates connections between the king component and pawn components, while the right picture depicts connections between two pawn components.

Our next goal is to show that the aforementioned construction fulfils its purposes. *i.e.* that (N, n)-scattered forward unravellings of *finite* structures are indeed finite (which we already discussed) and that they are $(n, \mathsf{N}, \emptyset)$-forests. As a preliminary step we establish the following lemma.

Lemma 5.11 Assume $(n, \mathsf{N}, \mathcal{I})$ are as in Definition 5.8. Then every undirected path in $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$ leading from a root of some pawn component to any of its leaves has length at least $2n$.

Proof. Suppose that there exists an undirected path $\rho := \rho_1 \ldots \rho_m$ in $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$ of length less than $2n$ for which ρ_1 and ρ_m are, respectively, a leaf and the root of the same pawn component. Note that ρ contains elements from at least two components, as the root-to-leaf paths in every pawn component are of length $2n$ by design. Thus by the construction of $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$ (especially by our way of "linking" components in Definition 5.10) we know that one of the following cases holds for any two consecutive elements ρ_{i-1} and ρ_i of the path ρ:

- ρ_{i-1} and ρ_i are in the same component,
- ρ_{i-1} is a leaf of some pawn component, say $\mathcal{I}^{(*,*)}_{♟,*,h}$, and ρ_i is is the root of $\mathcal{I}^{(*,*)}_{♟,*,1-h}$, or
- ρ_i is a leaf of some pawn component, say $\mathcal{I}^{(*,*)}_{♟,*,h}$, and ρ_{i-1} is is the root of $\mathcal{I}^{(*,*)}_{♟,*,1-h}$.

In particular, the above observation implies that any two consecutive elements of ρ taken from different components are of different "hue". Thus ρ has the shape $\rho := \rho'_0 \mathrm{d}_1 \ldots \rho_{2k} \mathrm{d}_{2k+1}$, where the even-numbered paths ρ'_{2i} are (possibly single-element) leaf-to-leaf paths traversing a single component, and the odd-numbered elements d_{2i+1} are roots of components. But this implies that ρ_1 and $\rho_m = \mathrm{d}_{2k+1}$ belong to components with different "hue", contradicting the fact that they belong to the same component. Hence such a path ρ does not exist. □

We call an n-neighbourhood *safe* if it is fully contained in some (copy of a) component, and *unsafe* otherwise. A joint application of Lemma 5.11 and the definition of the king component, yield a general characterisation on how n-neighbourhoods interact with components in scattered unravellings.

Lemma 5.12 With $(n, \mathsf{N}, \mathcal{I})$ as in Definition 5.8, let $\mathcal{N}$ be any unsafe n-neighbourhood of $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$. Then $\mathcal{N}$ does not contain any N-named elements, and for all components $\mathcal{J}$ of $\mathscr{J}(n, \mathsf{N}, \mathcal{I})$ sharing an element with $\mathcal{N}$, we have the following dichotomy: either $\mathcal{N}$ contains some leaf of $\mathcal{J}$ (we call such component **$\mathcal{N}$-upper**) or $\mathcal{N}$ contains the root of $\mathcal{J}$ (we call such component **$\mathcal{N}$-lower**).

Proof. Let $\mathcal{N}$ be any unsafe n-neighbourhood of $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$. For the first property, note that if $\mathcal{N}$ would contain an N-named, then by the definition of the king component (the first item of Definition 5.8) it would be fully contained in the king component, contradicting the unsafety of $\mathcal{I}$. For the second property, suppose that $\mathcal{J}$ is a component that shares an element with $\mathcal{N}$. As $\mathcal{N}$ is unsafe, it is not fully contained in $\mathcal{J}$. Thus, by the construction of cross-component connections (Items (i) and (ii) from Definition 5.10) we know that either $\mathcal{N}$ contains a leaf or the root of $\mathcal{J}$, hence $\mathcal{J}$ is either $\mathcal{N}$-lower or $\mathcal{N}$-upper. Suppose now that $\mathcal{J}$ is both $\mathcal{N}$-lower and $\mathcal{N}$-upper. This implies that there is a path between the root of $\mathcal{J}$ and some of its leaves (present in $\mathcal{N}$). As $\mathcal{N}$ is an n-neighbourhood, such a path is of length less than $2n$. But this is clearly not possible by Lemma 5.11. Hence, $\mathcal{J}$ cannot be both $\mathcal{N}$-lower and $\mathcal{N}$-upper. $\square$

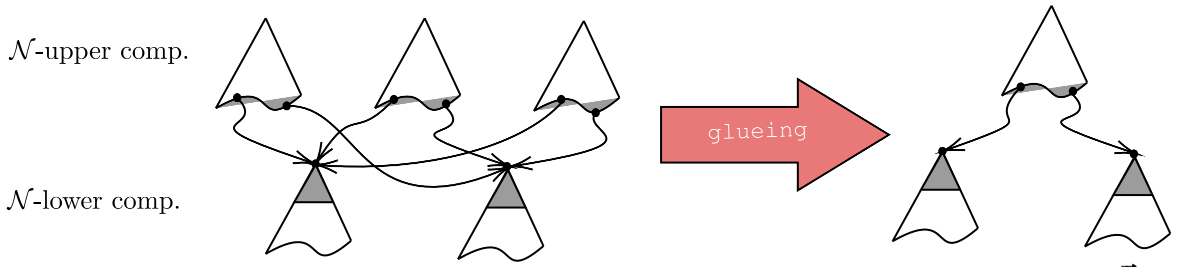


Figure 5.4: Visualisation of $\mathcal{N}$-lower and $\mathcal{N}$-upper components of some n-neighbourhood $\mathcal{N}$ of $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$. The right part of the picture presents the notion of glueing, to be defined later.

We next establish that all $\mathcal{N}$-upper components look alike as formalised by Lemma 5.13.

Lemma 5.13 Let $(n, \mathsf{N}, \mathcal{I})$ be as in Definition 5.8, and take any unsafe n-neighbourhood $\mathcal{N}$ of $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$. Then either the only $\mathcal{N}$-upper component is the king component, or there exists a unique element $\mathrm{d} \in \Delta^{\mathcal{I}}$ such that every $\mathcal{N}$-upper component is of the from $\mathcal{I}^{(*,*)}_{\mathbf{1},\mathrm{d},*}$. In particular, this means that all $\mathcal{N}$-upper components are isomorphic.

Proof. Let c_1 and c_2 be leaves from $\mathcal{N}$-upper components, and let $\mathcal{I}^{(*,*)}_{\mathbf{1},\mathrm{e},*}$ be the component to which c_1 belongs (the proof for the case when c_1 is included in the king component is completely analogous). We closely follow the proof of Lemma 5.11. Among other things, we proved there that there is a path ρ between c_1 and c_2 of length at most n, having the form $\rho \coloneqq \rho'_0 \mathrm{d}_1 \ldots \rho_{2k} \mathrm{d}_{2k+1} \rho_{2k+2}$, where the even-numbered paths ρ'_{2i} are (possibly single-element) leaf-to-leaf paths contained in a single component, and the odd-numbered elements d_{2i+1} are roots of components. By assumption, all the elements from ρ'_0 are in $\mathcal{I}^{(*,*)}_{\mathbf{1},\mathrm{e},*}$. Suppose that we have already shown that the elements of ρ'_i belong to $\mathcal{I}^{(*,*)}_{\mathbf{1},\mathrm{e},*}$ (for parameters $*$ possibly different from the initial ones), and let us establish the same for the elements from ρ'_{i+2}. Since the elements from ρ'_i belong to $\mathcal{I}^{(*,*)}_{\mathbf{1},\mathrm{e},*}$, the last element of ρ'_i belongs to $\mathcal{I}^{(*,*)}_{\mathbf{1},\mathrm{e},*}$. By the linking process (*i.e.* Item (ii) of Definition 5.10), we conclude that d_{i+1} is the root of some $\mathcal{I}^{(*,\mathrm{e})}_{\mathbf{1},*,*}$. Analogously, this implies that the first element of ρ'_{i+2} belongs to $\mathcal{I}^{(*,*)}_{\mathbf{1},\mathrm{e},*}$, yielding that all the elements of ρ'_{i+2} are in $\mathcal{I}^{(*,*)}_{\mathbf{1},\mathrm{e},*}$. Hence, by induction, c_1 and c_2 belong to copies of the same component. $\square$

We would like to establish that unsafe n-neighbourhoods are homomorphically-equivalent to $\emptyset$-trees. In the process of doing so, we define a handy operation of glueing, that, intuitively, given an unsafe component $\mathcal{N}$ will collapse isomorphic components that have common elements with $\mathcal{N}$ into a single component.

Definition 5.14 Let $(n, \mathsf{N}, \mathcal{I})$ be as in Definition 5.8. Given an unsafe n-neighbourhood $\mathcal{N}$ of $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$, a **glueing** $\mathsf{glue}(\mathcal{I})$ of $\mathcal{N}$ is defined as the restriction of $\mathcal{N}$ to all elements from all $\mathcal{N}$-lower components and all elements from one *single* $\mathcal{N}$-upper component.

Note that by Lemma 5.13 all $\mathcal{N}$-upper components are isomorphic copies of the same component, thus *the* glueing of $\mathcal{N}$ is unique up to isomorphism. It follows from the construction of $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$ that the glueing of a neighbourhood $\mathcal{N}$ is a $\emptyset$-tree. Moreover, the identity function clearly serves a homomorphism from $\mathsf{glue}(\mathcal{N})$ to $\mathcal{N}$. A homomorphism in the other direction is established next.

Lemma 5.15 Let $(n, \mathsf{N}, \mathcal{I})$ be as in Definition 5.8, and take any unsafe n-neighbourhood $\mathcal{N}$ of $\mathcal{I}_{\mathsf{N}}^{\vec{n}}$. Then the glueing $\mathsf{glue}(\mathcal{N})$ of $\mathcal{N}$ is a $\emptyset$-tree, homomorphically equivalent to $\mathcal{N}$.

Proof. As we already discussed shortly before the proof, the fact that $\mathsf{glue}(\mathcal{N})$ is a $\emptyset$-tree follows from the linking process of components (Definition 5.10), and it is immediate to see that the identity function is a homomorphism from $\mathsf{glue}(\mathcal{N})$ to $\mathcal{N}$. To craft a homomorphism from $\mathcal{N}$ to $\mathsf{glue}(\mathcal{N})$, let $\mathfrak{h}$ be a mapping that serves as the identity function of $\mathcal{N}$-lower components and a function that maps elements from $\mathcal{N}$-upper components to their corresponding ones in the $\mathsf{glue}(\mathcal{N})$-upper component. Note that $\mathfrak{h}$ is well-defined as all $\mathcal{N}$-upper components are isomorphic (as provided by Lemma 5.13). Moreover, $\mathfrak{h}$ maps ℓ-th leaves of upper components to the ℓ-th leaf of their isomorphic copy. We would like to point out that when the only $\mathcal{N}$-component is the king component, then $\mathfrak{h}$ is the identity function, and hence a homomorphism. Thus for the rest of the proof we assume that all $\mathcal{N}$-lower components are pawn components. To see that $\mathfrak{h}$ is a homomorphism, we take any pair $(\mathrm{d}, \mathrm{e}) \in r^{\mathcal{N}}$ and show that $(\mathfrak{h}(\mathrm{d}), \mathfrak{h}(\mathrm{e})) \in r^{\mathsf{glue}(\mathcal{N})}$ holds. Note that by construction of $\mathcal{I}_{\mathsf{N}}^{\vec{n}}$ only the following cases can occur:

- Both d, e are in the same component.
  Then we either conclude by the fact that $\mathfrak{h}$ is the identity function (the case of $\mathcal{N}$-lower components) or by the fact that any two $\mathcal{N}$-upper components are isomorphic.
- The element d is a leaf of some component and e is the root of some component. Hence, by invoking Item (i) of Definition 5.10, we know that there exists parameters ℓ and h for which the element d is the ℓ-th leaf of $\mathcal{I}_{\text{♟},\mathrm{c},h}^{(*,*)}$, and c is the root of $\mathcal{I}_{\text{♟},\mathsf{orig}_{\mathcal{I}}(\mathrm{e}),1-h}^{(\ell,\mathrm{c})}$. We actually know more: *every* ℓ-th leaf d' of every $\mathcal{I}_{\text{♟},\mathrm{c},h}^{(*,*)}$ component satisfies $(\mathrm{d}', \mathrm{e}) \in r^{\mathcal{N}}$. In particular, this implies that between the ℓ-th leaf d' of the unique $\mathcal{N}$-upper component of $\mathsf{glue}(\mathcal{N})$ and e there is an $r^{\mathsf{glue}(\mathcal{N})}$-role connection. Thus $(\mathfrak{h}(\mathrm{d}), \mathfrak{h}(\mathrm{e})) \in r^{\mathsf{glue}(\mathcal{N})}$ holds.

This concludes the proof that $\mathfrak{h}$ is indeed a homomorphism. □

As an immediate corollary of the previous lemma we conclude the following.

Corollary 5.16

For any finite $n \in \mathbb{N}$, any set of names $\mathsf{N} \subseteq \mathbf{N_I}$, and any finite interpretation $\mathcal{I}$, every (n, N)-scattered-forward-unravelling $\mathcal{I}_{\mathsf{N}}^{\vec{n}}$ of $\mathcal{I}$ is $(n, \mathsf{N}, \emptyset)$-forest.

Analogously to Definition 5.3, we can define description logics preserved under our new unravellings.

Definition 5.17 A description logic $\mathcal{DL}$ is **preserved under scattered forward unravellings** if for all finitely satisfiable $\mathcal{DL}$-KBs $\mathcal{K}$, their finite models $\mathcal{I}$, and for all but finitely-many positive integers $n \in \mathbb{N}$, the $(n, \mathsf{ind}(\mathcal{K}))$-scattered forward unravelling $\mathcal{I}_{\mathsf{ind}(\mathcal{K})}^{\vec{n}}$ of $\mathcal{I}$ is a finite model of $\mathcal{K}$.

Relying on Corollary 5.16 one can finally show that any DL preserved under scattered forward unravellings is also finitely $\emptyset$-forest-friendly. The proof is nearly the same as the proof of Theorem 5.4.

Theorem 5.18

If a DL $\mathcal{DL}$ is preserved under scattered forward unravellings then $\mathcal{DL}$ is finitely $\emptyset$-forest-friendly.

Proof. Take any finitely satisfiable $\mathcal{DL}$-KB $\mathcal{K}$, any of its finite models $\mathcal{I}$, and any positive integer $m \in \mathbb{N}$. It suffices to show that there exists an $(m, \mathsf{ind}(\mathcal{K}), \emptyset)$-forest model $\mathcal{J} \models \mathcal{K}$. Take $\mathcal{J} := \mathcal{I}_{\mathsf{ind}(\mathcal{K})}^{\vec{n}}$ for a sufficiently large n greater or equal to m. By preservation under scattered forward unravellings, we infer $\mathcal{J} \models \mathcal{K}$. Together with Corollary 5.16 we derive that $\mathcal{J}$ is a finite $(n, \mathsf{ind}(\mathcal{K}), \emptyset)$-forest (thus also $(m, \mathsf{ind}(\mathcal{K}), \emptyset)$-forest) that covers $\mathcal{I}$, as requested. □

Similarly to the end of the previous section, we conclude with a list of useful properties of scattered forward unravellings. The idea is that for future applications one can employ Property 5.19 in order to show preservation of: satisfaction of ABoxes (via Item (A)), existence of "unary-types" and "role-types" (via Item (B)), cardinality constraints on the total number of successors (via Item (D)), role hierarchies and safe boolean role combinations (via Item (C)), regular role expressions and fixed points (via Item (E)). Consult the following lemma:

Property 5.19. Assume that the parameters $(n, \mathsf{N}, \mathcal{I})$ are as in Definition 5.8 and let $\mathcal{I}_{\mathsf{N}}^{\vec{n}}$ be any (n, N)-scattered-forward-unravelling of $\mathcal{I}$. Then the following conditions are satisfied:

(A) The interpretations $\mathcal{I}$ and $\mathcal{I}_{\mathsf{N}}^{\vec{n}}$ restricted to all N-named elements are isomorphic.

(B) For any concept name C we have that $\mathrm{C}^{\mathcal{I}}$ is non-empty iff $\mathrm{C}^{\mathcal{I}_{\mathsf{N}}^{\vec{n}}}$ is non-empty. Similarly, for any role name r we have that $r^{\mathcal{I}}$ is non-empty iff $r^{\mathcal{I}_{\mathsf{N}}^{\vec{n}}}$ is non-empty.

(C) For any $w \in \Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{n}}}$ we have $\mathsf{Conc}_{\mathcal{I}}(\mathsf{orig}_{\mathcal{I}}(w)) = \mathsf{Conc}_{\mathcal{I}_{\mathsf{N}}^{\vec{n}}}(w)$. Moreover, for all v satisfying $(w, v) \in r^{\mathcal{I}_{\mathsf{N}}^{\vec{n}}}$ for some $r \in \mathbf{N_R}$, we have $\mathsf{Rol}_{\mathcal{I}}(\mathsf{orig}_{\mathcal{I}}(w), \mathsf{orig}_{\mathcal{I}}(v)) = \mathsf{Rol}_{\mathcal{I}_{\mathsf{N}}^{\vec{n}}}(w, v)$.

(D) For all $w \in \Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{n}}}$ we have that w in $\mathcal{I}_{\mathsf{N}}^{\vec{n}}$ and $\mathsf{orig}_{\mathcal{I}}(w)$ in $\mathcal{I}$ have the same number of successors satisfying the same roles and concepts, *i.e.* an analogue of Item (D) of Property 5.5 holds.

(E) For any $w \in \Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{n}}}$ we have that w and $\mathsf{orig}_{\mathcal{I}}(w)$ are directed-path-equivalent, *i.e.* an analogue of Item (E) of Property 5.5 holds.

Proof. We proceed with all items one-by-one, showing their satisfaction.

- Proof of Item (A) and Item (B) of Property 5.19.
  By the construction of the king component in Definition 5.8, we see that $\mathcal{I}_{\mathsf{N}}^{\vec{n}}$ contains an isomorphic copy of $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$ restricted to all words of length at most two. Thus by applying Item (A) and Item (B) of Property 5.5, we are done.
- Proof of Item (C) of Property 5.19.
  Take any w, v and suppose that $(w, v) \in r^{\mathcal{I}_{\mathsf{N}}^{\vec{n}}}$ holds for some $r \in \mathbf{N_R}$. By Item (C) of Property 5.5 it suffices to show that:

  $$\mathsf{Conc}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}(\mathsf{orig}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}(w)) = \mathsf{Conc}_{\mathcal{I}_{\mathsf{N}}^{\vec{n}}}(w) \text{ and } \mathsf{Rol}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}(\mathsf{orig}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}(w), \mathsf{orig}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}(v)) = \mathsf{Rol}_{\mathcal{I}_{\mathsf{N}}^{\vec{n}}}(w, v).$$

  For the equality between sets of concepts, this follows from the fact that w is just an isomorphic copy of $\mathsf{orig}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}(w)$, according to Definition 5.9. For the equality between sets of roles, we consider two cases:

  (a) w and v are in the same component. Then we are again done by the presence of an isomorphism between copies and selected fragments of $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$.

  (b) w and v are in different components, implying that w is a leaf of some component, while v is the root of some components. We then conclude by our linking process described in Definition 5.10.
- Proof of Item (D) of Property 5.19.
  Take any $w \in \Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{n}}}$, non-empty set of role names $\mathsf{R} \subseteq \mathbf{N_R}$, and any set of concept names $\mathsf{C} \subseteq \mathbf{N_C}$. We consider two cases. In the first one, we assume that w is a leaf of a component. Then by Item (D) of Property 5.5 it suffices to establish a bijection between the sets

  $$\left\{v \in \Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{n}}} \mid \mathsf{Conc}_{\mathcal{I}_{\mathsf{N}}^{\vec{n}}}(v) = \mathsf{C} \text{ and } \mathsf{Rol}_{\mathcal{I}_{\mathsf{N}}^{\vec{n}}}(w, v) = \mathsf{R}\right\}, \text{and}$$

  $$\left\{v \in \Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}} \mid \mathsf{Conc}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}(v) = \mathsf{C} \text{ and } \mathsf{Rol}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}(\mathsf{orig}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}(w), v) = \mathsf{R}\right\},$$

  which exists by the fact that components are isomorphic to neighbourhoods of $\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}$. From now on assume that w is not a leaf of a component. It suffices to prove that

  $$A := \left\{v \in \Delta^{\mathcal{I}_{\mathsf{N}}^{\vec{n}}} \mid \mathsf{Conc}_{\mathcal{I}_{\mathsf{N}}^{\vec{n}}}(v) = \mathsf{C} \text{ and } \mathsf{Rol}_{\mathcal{I}_{\mathsf{N}}^{\vec{n}}}(w, v) = \mathsf{R}\right\}, \text{and}$$

  $$B := \left\{\mathrm{d} \in \Delta^{\mathcal{I}} \mid \mathsf{Conc}_{\mathcal{I}}(\mathrm{d}) = \mathsf{C} \text{ and } \mathsf{Rol}_{\mathcal{I}}(\mathsf{orig}_{\mathcal{I}_{\mathsf{N}}^{\vec{\omega}}}(w), \mathrm{d}) = \mathsf{R}\right\}.$$

are equicardinal. Without loss of generality let us assume that w is a member of a pawn component (the proof for the king component is analogous), which implies that w is the ℓ-th leaf of $\mathcal{I}^{(*,*)}_{\text{♟},\mathrm{e},h}$ for some $\mathrm{e} \in \Delta^{\mathcal{I}}$ and $h \in \{0,1\}$. To show equicardinality of A and B, we employ the theorem of Cantor-Bernstein. It suffices to establish that:

– $\mathsf{orig}_{\mathcal{I}^{\vec{\varpi}}_{\mathsf{N}}}$ is an injection from A to B.
  The fact that $\mathsf{orig}_{\mathcal{I}^{\vec{\varpi}}_{\mathsf{N}}}$ is a function follows by the linking process of Definition 5.10. For injectivity, it suffices to see that each $v \in A$ is a root of some pawn component $\mathcal{I}^{(\ell,\mathrm{e})}_{\text{♟},*,1-h}$ and that per each $\mathrm{d} \in \Delta^{\mathcal{I}}$ such a component $\mathcal{I}^{(\ell,\mathrm{e})}_{\text{♟},\mathrm{d},1-h}$ is unique.
– There is an injection from B to A. The mapping f that assigns to each $\mathrm{d} \in B$ the root of $\mathcal{I}^{(\ell,\mathrm{e})}_{\text{♟},\mathrm{d},1-h}$ is the desired injection. The fact that f is a function follows again from the linking process, while injectivity of f is due to the uniqueness of $\mathcal{I}^{(\ell,\mathrm{e})}_{\text{♟},\mathrm{d},1-h}$.

• Proof of Item (E) of Property 5.19.
  For any $w \in \Delta^{\mathcal{I}^{\vec{n}}_{\mathsf{N}}}$ we have that w and $\mathsf{orig}_{\mathcal{I}}(w)$ are directed-path-equivalent, that is:
  – If ρ with $\rho_1 := w$ is a (possibly infinite) directed path in $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$ then ρ', defined as $\rho'_i := \mathsf{orig}_{\mathcal{I}}(\rho_i)$ for all i, is a directed path in $\mathcal{I}$ such that for all i we have $\mathsf{Conc}_{\mathcal{I}^{\vec{n}}_{\mathsf{N}}}(\rho_i) = \mathsf{Conc}_{\mathcal{I}}(\rho'_i)$, and for all $i > 1$ we have $\mathsf{Rol}_{\mathcal{I}^{\vec{n}}_{\mathsf{N}}}(\rho_{i-1}, \rho_i) = \mathsf{Rol}_{\mathcal{I}}(\rho'_{i-1}, \rho'_i)$.
  – If ρ with $\rho_1 := \mathsf{orig}_{\mathcal{I}}(w)$ is a (possibly infinite) directed path in $\mathcal{I}$ then ρ' defined as:
    (i) $\rho'_1 := w$, and
    (ii) $\rho'_i := \rho_i$ if both ρ'_{i-1} and ρ_i are N-named,
    (iii) $\rho'_i := \rho'_{i-1}\rho_i$ if ρ'_{i-1} is not a leaf of the component,
    (iv) ρ'_i is the root of $\mathcal{I}^{(\ell,\mathrm{e})}_{\text{♟},\rho_i,1-h}$ if ρ'_{i-1} is the ℓ-th leaf of $\mathcal{I}^{(*,*)}_{\text{♟},\mathrm{e},h}$ for some $\mathrm{e} \in \Delta^{\mathcal{I}}$ and $h \in \{0,1\}$,
    (v) ρ'_i is the root of $\mathcal{I}^{(\ell,\text{♚})}_{\text{♟},\rho_i,0}$ if ρ'_{i-1} is the ℓ-th leaf of $\mathcal{I}_{\text{♚}}$,
    is a directed path in $\mathcal{I}^{\vec{n}}_{\mathsf{N}}$ such that for all i we have $\mathsf{Conc}_{\mathcal{I}^{\vec{n}}_{\mathsf{N}}}(\rho'_i) = \mathsf{Conc}_{\mathcal{I}}(\rho_i)$, and for all $i > 1$ we have $\mathsf{Rol}_{\mathcal{I}^{\vec{n}}_{\mathsf{N}}}(\rho'_{i-1}, \rho'_i) = \mathsf{Rol}_{\mathcal{I}}(\rho_{i-1}, \rho_i)$.
  Once the construction of ρ' from ρ (and vice versa) is given, its correctness follows from Item (C) of Property 5.19.

□

As a potential application of scattered forward unravellings we can provide results for logics that are not or are not known to be finitely controllable. A toy example is presented below. This exponentially improves the doubly-exponential upper bound that follows from the finite satisfiability problem for the guarded fixed-point negation fragment [BtCS15, Thm. 4.4].

Corollary 5.20
The finite entailment problem for UCQs over $\mu\mathcal{ALC}$-KBs is ExpTime-complete.

Proof. The fact that $\mu\mathcal{ALC}$ is preserved under scattered forward unravellings follow from Property 5.19, with a routine induction over the complexity of a formula using approximation semantics of fixed-points [BW18, p. 5] (as also done in the previous section). The satisfiability of the knowledge-based satisfiability problem for $\mu\mathcal{ALC}$ follows from [SV01, Thm. 2], which is also applicable to the *finite* satisfiability problem due to the finite model property of $\mu\mathcal{ALC}$ with nominals [Tam15, Thm. 5.17]. Hence, by Theorem 5.18 and Theorem 4.37 we are done. □

Note that all logics contained in $\mathcal{ZQ}$ are finitely controllable [BK22, Thm. 3.1], and hence the results concerning their query entailment from Corollary 5.7 can be transferred to the finite-world scenario. On the other hand, employing scattered forward unravellings one can provide an independent proof of this fact.

Corollary 5.21
The finite entailment problem for UCQs over $\mathcal{ZQ}$-KBs is ExpTime-complete.

5.3 Scattered Unravellings with Rebalancing

Some description logics can express both local cardinality constraints (*i.e.* constraints concerning the role successors of specific individuals) and global cardinality constraints (*i.e.* constraints on the overall cardinality of concepts). Prominent examples of such DLs are, *e.g.* Statistical $\mathcal{ALC}$ [PP17], and $\mathcal{ALCSCC}$ with Restricted Cardinality Boxes [Baa17]. In this section we will see how the notion of scattered forward unravellings can be adjusted so that the construction from the previous section additionally preserves *ERCBoxes*, a broad class of linear constraints over the domain.

Definition 5.22 A **semi-restricted cardinality constraint** is an expression δ of the form[a]

$$\delta := \mathrm{N}_1 \cdot x_{\mathrm{C}_1} + \ldots + \mathrm{N}_k \cdot x_{\mathrm{C}_k} + M \leq \mathrm{N}_{k+1} \cdot x_{\mathrm{C}_{k+1}} + \ldots + \mathrm{N}_{k+\ell} \cdot x_{\mathrm{C}_{k+\ell}},$$

where all C_i are concept names, x_{C_i} are non-negative integer variables, and all N_i as well as M are non-negative integer constants, for all $1 \leq i \leq k{+}\ell$. A solution $\mathfrak{s}$ for δ is a mapping of variables to non-negative integers under which δ evaluates to true. An interpretation $\mathcal{I}$ satisfies δ (written $\mathcal{I} \models \delta$) whenever the mapping $x_{\mathrm{C}} \mapsto |\mathrm{C}^{\mathcal{I}}|$ is a solution for δ. An **extended restricted cardinality box (ERCBox)** [BBR20] is a positive boolean combination of semi-restricted cardinality constraints. The notion of solutions and satisfaction by an interpretation is lifted to ERCBoxes in the obvious way.

[a]Note the asymmetry in the definition: we do not allow for "spare" integer constants on the RHS. This is because we do not want to give cardinality constraints the power to express nominals.

An important property of ERCBoxes is that they enjoy arbitrarily large solutions.

Lemma 5.23 If an ERCBox $\mathcal{E}$ has a solution $\mathfrak{s}$, then for any positive integer n, the mapping $\mathfrak{s}' : x \mapsto n \cdot \mathfrak{s}(x)$ is also a solution for $\mathcal{E}$. Thus solvable ERCBoxes have arbitrarily large solutions.

Proof. Let δ be a semi-restricted cardinality constraint of the form:

$$\delta := \mathrm{N}_1 \cdot x_{\mathrm{C}_1} + \ldots + \mathrm{N}_k \cdot x_{\mathrm{C}_k} + M \leq \mathrm{N}_{k+1} \cdot x_{\mathrm{C}_{k+1}} + \ldots + \mathrm{N}_{k+\ell} \cdot x_{\mathrm{C}_{k+\ell}},$$

and let $\mathfrak{s}$ be a solution for δ, *i.e.* we have that the inequality

$$(\heartsuit)\colon\ \mathrm{N}_1 \cdot \mathfrak{s}(x_{\mathrm{C}_1}) + \ldots + \mathrm{N}_k \cdot \mathfrak{s}(x_{\mathrm{C}_k}) + M \leq \mathrm{N}_{k+1} \cdot \mathfrak{s}(x_{\mathrm{C}_{k+1}}) + \ldots + \mathrm{N}_{k+\ell} \cdot \mathfrak{s}(x_{\mathrm{C}_{k+\ell}})$$

as well as the inequality (obtained by weakening $(\heartsuit)$)

$$(\clubsuit)\colon\ \mathrm{N}_1 \cdot \mathfrak{s}(x_{\mathrm{C}_1}) + \ldots + \mathrm{N}_k \cdot \mathfrak{s}(x_{\mathrm{C}_k}) \leq \mathrm{N}_{k+1} \cdot \mathfrak{s}(x_{\mathrm{C}_{k+1}}) + \ldots + \mathrm{N}_{k+\ell} \cdot \mathfrak{s}(x_{\mathrm{C}_{k+\ell}})$$

evaluate to true. Take a positive integer $n > 1$, and let $\mathfrak{s}_n$ be a mapping that maps every variable x to $n \cdot \mathfrak{s}(x)$. We claim that $\mathfrak{s}_n$ is also a solution for δ. By multiplying both sides of inequality $(\clubsuit)$ by $(n{-}1)$, adding the inequality $(\heartsuit)$, and simplifying the terms we get:

$$\mathrm{N}_1 \cdot n \cdot \mathfrak{s}(x_{\mathrm{C}_1}) + \ldots + \mathrm{N}_k \cdot n \cdot \mathfrak{s}(x_{\mathrm{C}_k}) + M \leq \mathrm{N}_{k+1} \cdot n \cdot \mathfrak{s}(x_{\mathrm{C}_{k+1}}) + \ldots + \mathrm{N}_{k+\ell} \cdot n \cdot \mathfrak{s}(x_{\mathrm{C}_{k+\ell}}),$$

which by the definition of $\mathfrak{s}_n$ is clearly equal to

$$\mathrm{N}_1 \cdot \mathfrak{s}_n(x_{\mathrm{C}_1}) + \ldots + \mathrm{N}_k \cdot \mathfrak{s}_n(x_{\mathrm{C}_k}) + M \leq \mathrm{N}_{k+1} \cdot \mathfrak{s}_n(x_{\mathrm{C}_{k+1}}) + \ldots + \mathrm{N}_{k+\ell} \cdot \mathfrak{s}_n(x_{\mathrm{C}_{k+\ell}}).$$

This justifies our claim. Next, let $\mathfrak{s}$ be a solution for an ERCBox $\mathcal{E}$. Then $\mathfrak{s}_n$ (defined as above) is a solution for at least the same semi-restricted cardinality constraints from $\mathcal{E}$ as $\mathfrak{s}$. As $\mathcal{E}$ is negation-free, we conclude that $\mathfrak{s}_n$ is also a solution for $\mathcal{E}$, finishing the proof. □

Having a finite model $\mathcal{I}$ of an ERCBox $\mathcal{E}$, it can happen that none of the scattered forward unravellings of $\mathcal{I}$ is a model of $\mathcal{E}$, as we did not care about statistical quantities of elements in the constructed models. In order to produce finite models of ERCBoxes, we design a way of "repairing" scattered forward unravellings, in order to restore the satisfaction of ERCBoxes. Before we move to the main result, we introduce a handy notion of *forward duplication* of domain elements, which is completely independent from the presented unravellings (and thus can be potentially useful for future applications). The key idea is to make a copy of an input element and connect it to all neighbours of the original.

Definition 5.24 Let $\mathcal{I}$ be an interpretation and let $\mathrm{d} \in \Delta^{\mathcal{I}}$ be a domain element. The **d-forward-duplication** of $\mathcal{I}$ is an interpretation $\mathcal{I}_{+\mathrm{d}}$ defined as follows:

- $\Delta^{\mathcal{I}_{+\mathrm{d}}} := \Delta^{\mathcal{I}} \cup \{\mathrm{d}'\}$ for a fresh element d'.
- $\mathcal{I}_{+\mathrm{d}}$ restricted to $\Delta^{\mathcal{I}}$ is isomorphic to $\mathcal{I}$.
- For each concept name $\mathrm{C} \in \mathbf{N_C}$ we have $\mathrm{d}' \in \mathrm{C}^{\mathcal{I}_{+\mathrm{d}}}$ if and only if $\mathrm{d} \in \mathrm{C}^{\mathcal{I}}$.
- For each role name $r \in \mathbf{N_R}$ and all $\mathrm{e} \in \Delta^{\mathcal{I}}$ we have $(\mathrm{d}', \mathrm{e}) \in r^{\mathcal{I}_{+\mathrm{d}}}$ if and only if $(\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}$.

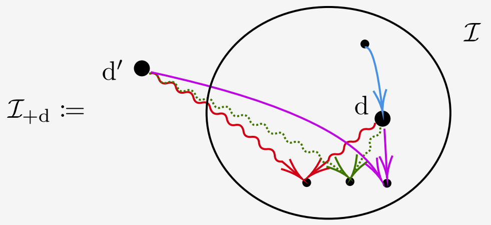
We will call d' a **copy of** d. We stress that d' has no "incoming edges" even if d has.

The notion of d-forward-duplication can be exemplified as follows.

Example 5.25. Let $\mathcal{I}$ be an interpretation with $\Delta^{\mathcal{I}} := \{\mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d}, \mathrm{e}\}$ defined as follows:

- All individual names are interpreted as a (they are irrelevant for this example).
- $\mathrm{A}^{\mathcal{I}} := \{\mathrm{a}, \mathrm{c}, \mathrm{e}\}$, $\mathrm{B}^{\mathcal{I}} := \{\mathrm{b}, \mathrm{d}\}$, $\mathrm{C}^{\mathcal{I}} := \{\mathrm{d}, \mathrm{e}\}$, and all other concept names are interpreted as $\emptyset$.
- $r^{\mathcal{I}} := \{(\mathrm{e}, \mathrm{d})\}$, $s_1^{\mathcal{I}} := \{(\mathrm{d}, \mathrm{a})\}$, $s_2^{\mathcal{I}} := \{(\mathrm{d}, \mathrm{b})\}$, $s_3^{\mathcal{I}} := \{(\mathrm{d}, \mathrm{c})\}$, and other role names are interpreted as $\emptyset$.

The interpretation $\mathcal{I}_{+\mathrm{d}}$, depicted below, has the domain $\{\mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d}, \mathrm{d}', \mathrm{e}\}$. It interprets all individual names as $\mathcal{I}$ do, all concept names are interpreted as in $\mathcal{I}$ with an exception of $\mathrm{B}^{\mathcal{I}_{+\mathrm{d}}} = \{\mathrm{b}, \mathrm{d}, \mathrm{d}'\}$, and $\mathrm{C}^{\mathcal{I}_{+\mathrm{d}}} = \{\mathrm{d}, \mathrm{d}', \mathrm{e}\}$, and role names are interpreted as in $\mathcal{I}$ with an exception of $s_1^{\mathcal{I}_{+\mathrm{d}}} := \{(\mathrm{d}, \mathrm{a}), (\mathrm{d}', \mathrm{a})\}$, $s_2^{\mathcal{I}_{+\mathrm{d}}} := \{(\mathrm{d}, \mathrm{b}), (\mathrm{d}', \mathrm{b})\}$, and $s_3^{\mathcal{I}_{+\mathrm{d}}} := \{(\mathrm{d}, \mathrm{c}), (\mathrm{d}', \mathrm{c})\}$.



The process of duplication can be repeated multiple times, which we formalise next. Given an interpretation $\mathcal{I}$ and a finite set $\mathrm{S} := \{(\mathrm{d}_1, n_1), (\mathrm{d}_2, n_2), \ldots, (\mathrm{d}_k, n_k)\} \subseteq 2^{(\Delta^{\mathcal{I}} \times \mathbb{N}_+)}$, the S-forward-duplication of $\mathcal{I}$ is the interpretation obtained by iterative application of d_i-forward-duplication n_i times for each $1 \leq i \leq k$. For readers eager to see the definition:

Definition 5.26 Let $\mathcal{I}$ be an interpretation and take $\mathrm{S} := \{(\mathrm{d}_1, n_1), (\mathrm{d}_2, n_2), \ldots, (\mathrm{d}_k, n_k)\} \subseteq 2^{(\Delta^{\mathcal{I}} \times \mathbb{N}_+)}$. The **S-forward-duplication** of $\mathcal{I}$ is an interpretation $\mathcal{I}_{+\mathrm{S}}$ defined as follows:

- $\Delta^{\mathcal{I}_{+\mathrm{S}}} := \Delta^{\mathcal{I}} \cup \{\mathrm{d}_i^{(j_i)} \mid 1 \leq i \leq k, 1 \leq j_i \leq n_i\}$, where all the elements $\mathrm{d}_i^{(j)}$ are fresh.
- $\mathcal{I}_{+\mathrm{S}}$ restricted to $\Delta^{\mathcal{I}}$ and $\mathcal{I}$ are isomorphic.
- For each concept name $\mathrm{A} \in \mathbf{N_C}$ we have $\mathrm{d}_i^{(j)} \in \mathrm{A}^{\mathcal{I}_{+\mathrm{S}}}$ if and only if $\mathrm{d}_i \in \mathrm{A}^{\mathcal{I}}$.
- For each role name $r \in \mathbf{N_R}$ and all $\mathrm{e} \in \Delta^{\mathcal{I}}$ we have $(\mathrm{d}_i^{(j)}, \mathrm{e}) \in r^{\mathcal{I}_{+\mathrm{S}}}$ if and only if $(\mathrm{d}_i, \mathrm{e}) \in r^{\mathcal{I}}$.

The key property of duplication, which follows by unfolding Definition 5.26 is as follows:

Fact 5.27. Let $\mathcal{I}$ be a *finite* interpretation. Then for any *finite* $\mathrm{S} := \{(\mathrm{d}_1, n_1), \ldots, (\mathrm{d}_k, n_k)\} \subseteq 2^{(\Delta^{\mathcal{I}} \times \mathbb{N}_+)}$, and any concept name $\mathrm{A} \in \mathbf{N_C}$, the following equation holds:

$$|\mathrm{A}^{\mathcal{I}_{+\mathrm{S}}}| = |\mathrm{A}^{\mathcal{I}}| + \sum_{i \in \{1,2,\ldots,k\},\ \mathrm{d}_i \in \mathrm{A}^{\mathcal{I}}} n_i.$$

We will first see that our notion of duplication can be used to restore satisfaction of ERCBoxes by scattered unravellings, and second, that all the good properties of scattered forward unravellings are preserved by it. More precisely we will show:

Lemma 5.28 Let $\mathcal{I}$ be a finite model of an ERCBox $\mathcal{E}$. Then for every positive $n \in \mathbb{N}$ and every finite set of names N, there exists a finite set $\mathrm{S} \subseteq 2^{(\Delta^{\mathcal{I}} \times \mathbb{N}_{+})}$ for which $(\mathcal{I}_{\mathsf{N}}^{\vec{n}})_{+S} \models \mathcal{E}$.

Proof. Let $\mathcal{I}$ be a finite model of an ERCBox $\mathcal{E}$, and let $\mathbf{C}_{\mathcal{I}} \coloneqq \{\mathrm{C}_{\mathrm{d}} \mid \mathrm{d} \in \Delta^{\mathcal{I}}\}$ be a set of fresh concept names per each domain element $\mathrm{d} \in \mathcal{I}$. Without loss of generality we assume that $\mathcal{I}$ interprets concepts from $\mathbf{C}_{\mathcal{I}}$ as empty sets. Take $\mathcal{J}$ to be the unique interpretation whose $(\mathbf{N_C} \setminus \mathbf{C}_{\mathcal{I}})$-reduct is $\mathcal{I}$, and that interprets C_{d} as singletons $\{\mathrm{d}\}$. This implies that the interpretations of fresh symbols from $\mathbf{C}_{\mathcal{I}}$ in $\mathcal{J}$ are pairwise-disjoint, which we will exploit later. Moreover, observe that $\mathcal{I}_{\mathsf{N}}^{\vec{n}}$ is the $(\mathbf{N_C} \setminus \mathbf{C}_{\mathcal{I}})$-reduct of $\mathcal{J}_{\mathsf{N}}^{\vec{n}}$.

We rewrite $\mathcal{E}$ to make it aware of fresh concept symbols. We proceed with each $\delta \in \mathcal{E}$:

$$\delta \coloneqq \mathrm{N}_1 \cdot x_{\mathrm{C}_1} + \ldots + \mathrm{N}_k \cdot x_{\mathrm{C}_k} + M \leq \mathrm{N}_{k+1} \cdot x_{\mathrm{C}_{k+1}} + \ldots + \mathrm{N}_{k+\ell} \cdot x_{\mathrm{C}_{k+\ell}}$$

and replace it by the inequality δ' given below:

$$\mathrm{N}_1 \cdot \left(\sum_{\mathrm{d} \in \mathrm{C}_1^{\mathcal{I}}} x_{\mathrm{C}_{\mathrm{d}}} \right) + \ldots + \mathrm{N}_k \cdot \left(\sum_{\mathrm{d} \in \mathrm{C}_k^{\mathcal{I}}} x_{\mathrm{C}_{\mathrm{d}}} \right) + M \leq \mathrm{N}_{k+1} \cdot \left(\sum_{\mathrm{d} \in \mathrm{C}_{k+1}^{\mathcal{I}}} x_{\mathrm{C}_{\mathrm{d}}} \right) \ldots + \ldots + \mathrm{N}_{k+\ell} \cdot \left(\sum_{\mathrm{d} \in \mathrm{C}_{k+\ell}^{\mathcal{I}}} x_{\mathrm{C}_{\mathrm{d}}} \right).$$

Moreover, per each concept A mentioned in $\mathcal{E}$ we append the inequalities:

$$\sum_{\mathrm{d} \in \mathrm{A}^{\mathcal{I}}} x_{\mathrm{C}_{\mathrm{d}}} \leq x_{\mathrm{A}} \qquad x_{\mathrm{A}} \leq \sum_{\mathrm{d} \in \mathrm{A}^{\mathcal{I}}} x_{\mathrm{C}_{\mathrm{d}}},$$

expressing the obvious relationship between the total number of elements in concepts and their sizes. Call the resulting ERCBox $\mathcal{E}'$. Since concept names from $\mathbf{C}_{\mathcal{I}}$ are interpreted in $\mathcal{J}$ as pairwise-disjoint singleton sets whose union is $\Delta^{\mathcal{J}}$, and because $\mathcal{E}'$ is just a finer-grained description of $\mathcal{E}$, it should be clear that $\mathcal{J} \models \mathcal{E}'$ if and only if $\mathcal{I} \models \mathcal{E}$. Note that by construction, the variables $x_{\mathrm{C}_{\mathrm{d}}}$ and $x_{\mathrm{C}_{\mathrm{e}}}$ for different $\mathrm{d} \neq \mathrm{e}$ are linearly independent, in the sense that the duplicating an element from concept the C_{d} does not influence the total number of elements in C_{e} and vice versa (but can, of course, influence sizes of other concepts).

Consider a solution $\mathfrak{s} \colon x_{\mathrm{C}} \mapsto |\mathrm{C}^{\mathcal{J}}|$ for $\mathcal{E}'$. By Lemma 5.23, we infer that the mapping $\mathfrak{s}' \colon x_{\mathrm{C}} \mapsto \left(|\Delta^{\mathcal{J}_{\mathsf{N}}^{\vec{n}}}| + 1\right) \cdot \mathfrak{s}(x_{\mathrm{C}})$, assigning "sufficiently large" values, is also a solution for $\mathcal{E}'$. By the fact that scattered unravellings preserve concepts (*cf.* Item (B) of Property 5.19) we know that for each variable $x_{\mathrm{C}_{\mathrm{d}}}$ mapped by $\mathfrak{s}$ to a positive value (resp. 0), the concepts C_{d} are non-empty (resp. empty) in the unravelling. Thus our proof plan is simple: we are going to duplicate elements d to make the cardinality of the concept C_{d} in $\mathcal{J}_{\mathsf{N}}^{\vec{n}}$ equal to $\mathfrak{s}'(x_{\mathrm{C}_{\mathrm{d}}})$. Formally, we take

$$\mathrm{S} \coloneqq \left\{ \left(\mathrm{d}, \mathfrak{s}'(x_{\mathrm{C}_{\mathrm{d}}}) - |\mathrm{C}_{\mathrm{d}}^{\mathcal{J}_{\mathsf{N}}^{\vec{n}}}|\right) \mid \mathrm{d} \in \Delta^{\mathcal{I}}, \left(\mathfrak{s}'(x_{\mathrm{C}_{\mathrm{d}}}) - |\mathrm{C}_{\mathrm{d}}^{\mathcal{J}_{\mathsf{N}}^{\vec{n}}}|\right) > 0 \right\}.$$

We stress that the value $\left(\mathfrak{s}'(x_{\mathrm{C}_{\mathrm{d}}}) - |\mathrm{C}_{\mathrm{d}}^{\mathcal{J}_{\mathsf{N}}^{\vec{n}}}|\right)$ is always non-negative, as the values of $\mathfrak{s}'$ were taken to be "sufficiently large". With a bit of routine calculations and Fact 5.27 we conclude that $(\mathcal{J}_{\mathsf{N}}^{\vec{n}})_{+\mathrm{S}} \models \mathcal{E}'$. Finally, by the presence of additional inequalities relating concepts from $\mathbf{C}_{\mathcal{I}}$ with concepts appearing in $\mathcal{E}$, we conclude that $(\mathcal{I}_{\mathsf{N}}^{\vec{n}})_{+\mathrm{S}} \models \mathcal{E}$ holds. ☐

To see that our notion of duplication preserves all the good properties of scattered unravellings, we are going to show that the analogue of Corollary 5.16 and Property 5.19 hold.

Lemma 5.29 Suppose that $(n, \mathsf{N}, \mathcal{I})$ are as in Definition 5.8, and take any finite $\mathrm{S} \subseteq 2^{(\Delta^{\mathcal{I}} \times \mathbb{N}_+)}$. Then $(\mathcal{I}_{\mathsf{N}}^{\vec{n}})_{+\mathrm{S}}$ is a finite $(n, \mathsf{N}, \emptyset)$-forest, whose n-neighbourhoods can be N-homomorphically mapped to $\mathcal{I}$.

Proof. Finiteness of $(\mathcal{I}_{\mathsf{N}}^{\vec{n}})_{+S}$ follows from the following three facts: (i) finiteness of $\mathcal{I}_{\mathsf{N}}^{\vec{n}}$, (ii) finiteness of S, and (iii) Fact 5.27. For the rest of the proof, it suffices to employ Lemma 5.15 after observing that the function mapping duplicated elements to their originals (and behaving as the identity function on other elements) is an N-homomorphism from $(\mathcal{I}_{\mathsf{N}}^{\vec{n}})_{+S}$ to $\mathcal{I}_{\mathsf{N}}^{\vec{n}}$. □

We are ready to proceed with our usual list of properties.

Property 5.30. Assume the parameters $(n, \mathsf{N}, \mathcal{I}, \mathrm{S})$ as in Lemma 5.29. Then the analogue of Property 5.19 holds, namely:

(A) The interpretations $\mathcal{I}$ and $(\mathcal{I}_{\mathsf{N}}^{\vec{n}})_{+\mathrm{S}}$ restricted to all N-named elements are isomorphic.

(B) For any concept name C we have that $\mathrm{C}^{\mathcal{I}}$ is non-empty if and only if $\mathrm{C}^{(\mathcal{I}_{\mathsf{N}}^{\vec{n}})_{+\mathrm{S}}}$ is non-empty. Similarly, for any role name r we have that $r^{\mathcal{I}}$ is non-empty if and only if $r^{(\mathcal{I}_{\mathsf{N}}^{\vec{n}})_{+\mathrm{S}}}$ is non-empty.

(C) For any $w \in \Delta^{(\mathcal{I}_{\mathsf{N}}^{\vec{n}})_{+\mathrm{S}}}$ we have $\mathsf{Conc}_{\mathcal{I}}(\mathsf{orig}^{\star}_{\mathcal{I}}(w)) = \mathsf{Conc}_{(\mathcal{I}_{\mathsf{N}}^{\vec{n}})_{+\mathrm{S}}}(w)$. Moreover, for all v with $(w, v) \in r^{(\mathcal{I}_{\mathsf{N}}^{\vec{n}})_{+\mathrm{S}}}$ for some $r \in \mathbf{N_R}$, we have $\mathsf{Rol}_{\mathcal{I}}(\mathsf{orig}^{\star}_{\mathcal{I}}(w), \mathsf{orig}^{\star}_{\mathcal{I}}(v)) = \mathsf{Rol}_{(\mathcal{I}_{\mathsf{N}}^{\vec{n}})_{+\mathrm{S}}}(w, v)$.

(D) For all w from $(\mathcal{I}_{\mathsf{N}}^{\vec{n}})_{+\mathrm{S}}$ we have that w in $(\mathcal{I}_{\mathsf{N}}^{\vec{n}})_{+\mathrm{S}}$ and $\mathsf{orig}^{\star}_{\mathcal{I}}(w)$ in $\mathcal{I}$ have the same number of successors satisfying the same roles and concepts, *i.e.* an analogue of Item (D) of Property 5.5 holds.

(E) For any w from $(\mathcal{I}_{\mathsf{N}}^{\vec{n}})_{+\mathrm{S}}$ we have that w and $\mathsf{orig}^{\star}_{\mathcal{I}}(w)$ are directed-path-equivalent, *i.e.* an analogue of Item (E) of Property 5.5 holds.

The function $\mathsf{orig}^{\star}_{\mathcal{I}}$ mentioned above is the mapping that first maps all duplicates to their originals (and behaves like identity on other elements) and then employs $\mathsf{orig}_{\mathcal{I}}$.

Proof sketch. For brevity, let $\mathcal{J} := (\mathcal{I}_{\mathsf{N}}^{\vec{n}})_{+\mathrm{S}}$. Note that, by construction of S-duplication, the interpretation $\mathcal{J}$ restricted to the elements from $\mathcal{I}_{\mathsf{N}}^{\vec{n}}$ and the interpretation $\mathcal{I}_{\mathsf{N}}^{\vec{n}}$ are isomorphic. Moreover, the process of S-duplication does not affect the named elements of $\mathcal{I}_{\mathsf{N}}^{\vec{n}}$. Hence, by transitivity and Property 5.19, we establish Item (A) of Property 5.30. For the remaining properties we proceed as follows. Take any element w and its copy w'. By construction (see: Definition 5.24) we know that $\mathsf{Conc}_{\mathcal{J}}(w) = \mathsf{Conc}_{\mathcal{J}}(w')$ as well as for any $v \in \Delta^{\mathcal{J}}$ we have $\mathsf{Rol}_{\mathcal{J}}(w, v) = \mathsf{Rol}_{\mathcal{J}}(w', v)$. This also implies that for any directed path ρ in $\mathcal{J}$ we have that $w\rho$ is a directed path in $\mathcal{J}$ if and only if $w'\rho$ is. The above properties, by transitivity and Property 5.19 imply the satisfaction of Items (B)–(E) of Property 5.30. □

Similarly to the two previous sections we define a new class of description logics.

Definition 5.31 A description logic $\mathcal{DL}$ is **preserved under scattered forward unravellings with rebalancing** if for all finitely satisfiable $\mathcal{DL}$-KBs $\mathcal{K}$, their finite models $\mathcal{I}$, and for all but finitely many positive integers $n \in \mathbb{N}$, there is a finite set $\mathrm{S} \subseteq 2^{(\Delta^{\mathcal{I}} \times \mathbb{N}_+)}$ for which $(\mathcal{I}_{\mathsf{N}}^{\vec{n}})_{+\mathrm{S}} \models \mathcal{K}$.

Once more we establish that preservation under unravellings guarantees being forest-friendly.

Theorem 5.32

If a description logic $\mathcal{DL}$ is preserved under scattered forward unravellings with rebalancing then $\mathcal{DL}$ is finitely $\emptyset$-forest-friendly.

Proof. Take any finitely satisfiable $\mathcal{DL}$-KB $\mathcal{K}$, any of its finite models $\mathcal{I}$, and any positive integer $m \in \mathbb{N}$. It suffices to show that there exists an $(m, \mathsf{ind}(\mathcal{K}), \emptyset)$-forest model $\mathcal{J} \models \mathcal{K}$. Take $\mathcal{J} := \mathcal{I}_{\mathsf{ind}(\mathcal{K})}^{\vec{n}}$ for a sufficiently large n greater or equal to m. By the assumption about preservation under scattered unravelling by rebalancing, we infer the existence of a set S, so that $\mathcal{J}_{+\mathrm{S}} \models \mathcal{K}$. Thus, what remains to be done is to show that $\mathcal{J}_{+\mathrm{S}}$ is a finite $(n, \mathsf{N}, \emptyset)$-forest that covers $\mathcal{I}$, but this follows from Lemma 5.29. □

As an application of Theorem 5.32, one can establish the exact complexity for the UCQ entailment problem over $\mathcal{ALCSCC}$ ontologies extended with ERCBoxes [BBR20, Sec. 2&Def. 6].

Corollary 5.33

The finite UCQ entailment problem for $\mathcal{ALCSCC}$ with ERCBoxes is ExpTime-complete.

Proof sketch. Let a knowledge base $\mathcal{K}$ be composed of an ABox $\mathcal{A}$, $\mathcal{ALCSCC}$-TBox $\mathcal{T}$, and an ERCBox $\mathcal{E}$. W.l.o.g. we can assume that (by routine renaming à la Scott) that concept names appear in $\mathcal{K}$ at "quantifier depth" at most one, and that the ERCBox $\mathcal{E}$ is of the form presented in Definition 5.22 (the original definition is slightly broader). Take a finite model $\mathcal{I} \models \mathcal{K}$ and any positive integer n, and let $\mathcal{J}$ be the $(n, \mathsf{ind}(\mathcal{K}))$-scattered-forward-unravelling of $\mathcal{I}$. Applying Lemma 5.28 we infer the existence of a set S for which $\mathcal{J}_{+\mathrm{S}} \models \mathcal{E}$. By Item (A) of Property 5.30 we deduce $\mathcal{J}_{+\mathrm{S}} \models \mathcal{A}$, and by Item (D) of Property 5.30 we deduce $\mathcal{J}_{+\mathrm{S}} \models \mathcal{T}$. Thus $\mathcal{J}_{+\mathrm{S}} \models \mathcal{K}$. Since $\mathcal{J}_{+\mathrm{S}}$ is an $(n, \mathsf{ind}(\mathcal{K}), \emptyset)$-forest that covers $\mathcal{I}$ (consult Lemma 5.29), thus by Theorem 5.32 we can conclude that $\mathcal{ALCSCC}$+ERCBoxes is $\emptyset$-forest-friendly. The desired ExpTime upper bound follows now from Corollary 4.38 and the ExpTime-completeness of the finite satisfiability problem for $\mathcal{ALCSCC}$+ERCBoxes by Baader et. al [BBR20, Thm. 7]. □

5.3.1 Statistical $\mathcal{EL}$ is ExpTime-hard

Interestingly enough, one can show that a very simple form of ERCBoxes, called Probabilistic Conditionals, causes a blow-up in the complexity of satisfiability for lightweight description logics like $\mathcal{EL}$, leading to intractability. This closes the gap in complexity results by Peñaloza and Potyka [PP17]. The goal of this section is to provide a suitable lower bound. We start with auxiliary definitions.

Definition 5.34 The set $\mathbf{C}_{\mathcal{EL}^{\neg}}$ of **$\mathcal{EL}^{\neg}$-concepts** is defined by a slight extension of the grammar for $\mathcal{EL}$:

$$\mathrm{C}, \mathrm{D} \ ::= \ \top \mid \mathrm{A} \mid \bar{\mathrm{A}} \mid \mathrm{C} \sqcap \mathrm{D} \mid \exists r.\mathrm{C},$$

where $\mathrm{C}, \mathrm{D} \in \mathbf{C}_{\mathcal{EL}^{\neg}}$, $\mathrm{A} \in \mathbf{N_C}$ and $r \in \mathbf{N_R}$. The semantics of $\mathcal{EL}^{\neg}$-concepts is defined as in the case of $\mathcal{EL}$ and $\mathcal{ALC}$ with the exception that the concepts of the form $\bar{\mathrm{A}}$ have the semantics $\bar{\mathrm{A}}^{\mathcal{I}} := \Delta^{\mathcal{I}} \setminus \mathrm{A}^{\mathcal{I}}$.

In contrast to plain $\mathcal{EL}$, the (finite) TBox-satisfiability problem for $\mathcal{EL}^{\neg}$ is no longer trivial and is actually ExpTime-complete [BBL05, Theorem 6]. A straightforward reduction from $\mathcal{ALC}$ follows by turning $\mathcal{ALC}$-TBoxes into NNF, getting rid of disjunction, and replacing negated atomic concepts $\neg\mathrm{A}$ by $\bar{\mathrm{A}}$.

Statistical $\mathcal{EL}$, abbreviated as $\mathcal{SEL}$, is a probabilistic description logic introduced a couple of years ago by Peñaloza and Potyka [PP17, Sec. 4] to reason about statistical properties over finite domains.

Definition 5.35 **$\mathcal{SEL}$-ontologies** are composed of **probabilistic conditionals** of the form $(\mathrm{C} \mid \mathrm{D})\,[k, l]$, where C, D are $\mathcal{EL}$-concepts from $\mathbf{C}_{\mathcal{EL}}$ and $k, l \in \mathbb{Q}$ are rational numbers satisfying $0 \leq k \leq l \leq 1$. An interpretation $\mathcal{I}$ satisfies a probabilistic conditional $(\mathrm{C} \mid \mathrm{D})\,[k, l]$ if

$$\text{either } \mathrm{D}^{\mathcal{I}} = \emptyset \text{ or } k \leq \frac{|(\mathrm{C} \sqcap \mathrm{D})^{\mathcal{I}}|}{|\mathrm{D}^{\mathcal{I}}|} \leq l.$$

The *size* of $\mathcal{SEL}$-TBoxes is defined as in $\mathcal{ALC}$ except that the numbers in probabilistic conditionals also contribute to the size and are measured in binary. In the satisfiability problem for $\mathcal{SEL}$-ontologies we ask if there is a *finite* model $\mathcal{I}$ of all probabilistic conditionals from an input $\mathcal{SEL}$-ontology $\mathcal{O}$.

It is an exercise to show that [PP17, Prop. 4] the usual $\mathcal{EL}$-GCIs $\mathrm{D} \sqsubseteq \mathrm{C}$ are equivalent to $(\mathrm{C} \mid \mathrm{D})\,[1,1]$. Hence, each $\mathcal{EL}$-TBox can be seen as an $\mathcal{SEL}$-ontology, and we are free to employ GCIs in place of probabilistic conditionals. We are now ready to prove the main result of this section.

Theorem 5.36

The satisfiability problem for $\mathcal{SEL}$ is ExpTime-complete, even if the only numbers allowed in probabilistic conditionals are $0, 0.5$, and 1.

The ExpTime upper bound is due to Lutz and Schröder [LS10, Thm. 9]. We focus on the lower bound. Let $\mathcal{O}$ be an arbitrary $\mathcal{EL}^{\neg}$-TBox, and let $\mathbf{C}_{\mathcal{O}}$ denote the set of all concept names that appear (possibly under negation) in $\mathcal{O}$. In what follows we are going to design a polynomially-larger $\mathcal{SEL}$-ontology $\mathcal{O}_{red}$ such that $\mathcal{O}_{red}$ is satisfiable if and only if the input $\mathcal{O}$ is satisfiable. It will be composed of two $\mathcal{SEL}$ ontologies, namely $\mathcal{O}_{tr}$ and $\mathcal{O}_{corr}$, responsible, respectively, for "translating" $\mathcal{O}$ into $\mathcal{SEL}$ and for guaranteeing the correctness of the translation. This results in ExpTime-hardness of the satisfiability problem for $\mathcal{SEL}$.

The main idea of the encoding is as follows. We first produce for each concept name A from $\mathbf{C}_{\mathcal{O}}$, two fresh concepts $\mathrm{A}_+, \mathrm{A}_- \not\in \mathbf{C}_{\mathcal{O}}$ intuitively intended to contain all members of A and, respectively, all members of its complement. Due to the lack of negation, we clearly are not able to fully formalise the above intuition, but the best we can do is to enforce, with the ontology $\mathcal{O}_{corr}$, that these concepts are interpreted as disjoint sets and each of them contains exactly half of the domain. This is sufficient for our purposes, since with fresh, pairwise different, concepts $\mathrm{Real}, \mathrm{Real}_+, \mathrm{Real}_- \not\in \mathbf{C}_{\mathcal{O}}$ we can separate the "real" model of $\mathcal{O}$ from the auxiliary parts required for the encoding. Finally, in the "translation" ontology $\mathcal{O}_{tr}$ we state that the restriction of a model of $\mathcal{O}_{red}$ to Real_+ satisfies $\mathcal{O}$. The translation simply changes all occurrences of A (resp. $\bar{\mathrm{A}}$) into A_+ (resp. A_-) and employs Real_+ to relativise concepts.

We start with the definition of $\mathcal{O}_{corr}$.

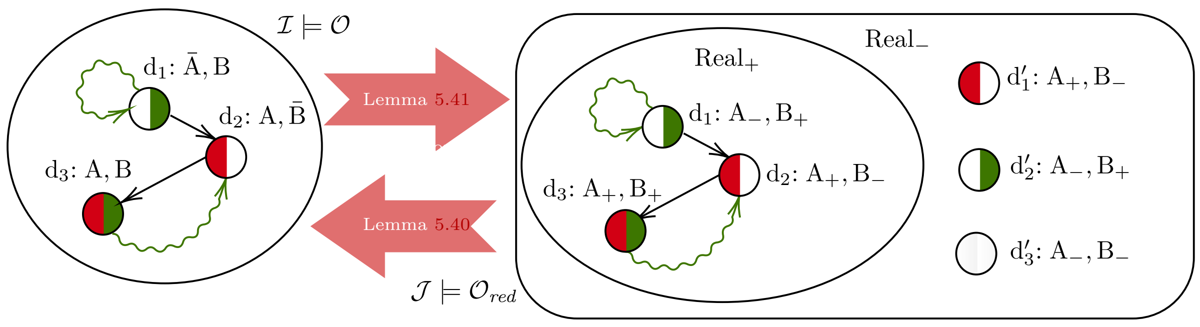
$$\mathcal{O}_{corr} \coloneqq \Big\{ (\mathrm{A}_+ \mid \top)\,[0.5, 0.5]\,, (\mathrm{A}_- \mid \top)\,[0.5, 0.5]\,, (\mathrm{A}_+ \mid \mathrm{A}_-)\,[0, 0] \;\mid\; \mathrm{A} \in \{\mathrm{Real}\} \cup \mathbf{C}_{\mathcal{O}} \Big\}$$

By unfolding the definition of probabilistic conditionals we immediately conclude the following facts.

Fact 5.37. For any $\mathrm{A} \in \mathbf{N_C}$ we have that $\mathcal{I} \models (\mathrm{A} \mid \top)\,[0.5, 0.5]$ if and only if $|\Delta^{\mathcal{I}}|$ is even and $|\mathrm{A}^{\mathcal{I}}| = \frac{1}{2}|\Delta^{\mathcal{I}}|$.

Fact 5.38. For any different concept names A, B we have that $\mathcal{I} \models (\mathrm{A} \mid \mathrm{B})\,[0, 0]$ if and only if $\mathrm{A}^{\mathcal{I}} \cap \mathrm{B}^{\mathcal{I}} = \emptyset$.

Now we focus on the "translating" ontology $\mathcal{O}_{tr}$. Let $\mathfrak{t}$ be a translation function defined by $\mathfrak{t}(\top) \coloneqq \mathrm{Real}_+$, $\mathfrak{t}(\mathrm{A}) \coloneqq \mathrm{A}_+ \sqcap \mathrm{Real}_+$ and $\mathfrak{t}(\bar{\mathrm{A}}) \coloneqq \mathrm{A}_- \sqcap \mathrm{Real}_+$ for all concept names $\mathrm{A} \in \mathbf{N_C}$ as well as $\mathfrak{t}(\mathrm{C} \sqcap \mathrm{D}) \coloneqq \mathfrak{t}(\mathrm{C}) \sqcap \mathfrak{t}(\mathrm{D})$ and $\mathfrak{t}(\exists r.\mathrm{C}) \coloneqq \mathrm{Real}_+ \sqcap \exists r.(\mathfrak{t}(\mathrm{C}) \sqcap \mathrm{Real}_+)$ for complex concepts. The ontology $\mathcal{O}_{tr}$ is obtained by replacing each GCI $\mathrm{C} \sqsubseteq \mathrm{D}$ from $\mathcal{O}$ with $\mathfrak{t}(\mathrm{C}) \sqsubseteq \mathfrak{t}(\mathrm{D})$. Finally, we put $\mathcal{O}_{red} \coloneqq \mathcal{O}_{corr} \cup \mathcal{O}_{tr}$. Note that the size of $\mathcal{O}_{red}$ is polynomial in $|\mathcal{O}|$. For more intuitions, consult the picture below.


$\mathcal{I} \models \mathcal{O}$
d_1: $\bar{\mathrm{A}}$, B
d_2: A, $\bar{\mathrm{B}}$
d_3: A, B
Lemma 5.41
Lemma 5.40
$\mathcal{J} \models \mathcal{O}_{red}$
Real_+
Real_-
d_1: $\mathrm{A}_-, \mathrm{B}_+$
d_2: $\mathrm{A}_+, \mathrm{B}_-$
d_3: $\mathrm{A}_+, \mathrm{B}_+$
d'_1: $\mathrm{A}_+, \mathrm{B}_-$
d'_2: $\mathrm{A}_-, \mathrm{B}_+$
d'_3: $\mathrm{A}_-, \mathrm{B}_-$

Let us start with an auxiliary notion of interpretations that are *good-for-encoding*. We say that $\mathcal{J}$ is **good-for-encoding** if for all concept names $\mathrm{A} \in \mathbf{C}_{\mathcal{O}}$ it satisfies $\mathrm{A}^{\mathcal{J}} = \mathrm{A}_+^{\mathcal{J}}$ and $\mathrm{A}_-^{\mathcal{J}} = \Delta^{\mathcal{J}} \setminus \mathrm{A}^{\mathcal{J}}$. The following lemma relates the translation function $\mathfrak{t}$, good-for-encoding interpretations and their submodels.

Lemma 5.39 (Agreement lemma) Let $\mathcal{J}$ be good-for-encoding, and let $\mathcal{I}$ be $\mathcal{J}$ restricted to $\mathrm{Real}_+^{\mathcal{J}}$. Then all $\mathcal{EL}^\neg$-concepts C employing *only* concept names from $\mathbf{C}_\mathcal{O}$ satisfy $C^\mathcal{I} = \mathfrak{t}(C)^\mathcal{J}$. Moreover for such concepts C, D we have: $\mathcal{J} \models \mathfrak{t}(C) \sqsubseteq \mathfrak{t}(D)$ if and only if $\mathcal{I} \models C \sqsubseteq D$.

Proof. Induction on the shape of concepts C. The cases for $C = \top, A$ or $\bar{A}$ for $A \in \mathbf{N_C}$ follow immediately from the definition of $\mathfrak{t}$ and the assumptions $A^\mathcal{J} = A_+^\mathcal{J}$ and $A_-^\mathcal{J} = \Delta^\mathcal{J} \setminus A^\mathcal{J}$. The case of $C = D \sqcap E$ follows from the fact that $\mathfrak{t}$ is homomorphic for $\sqcap$. Hence, the only interesting case is when $C = \exists r.D$. Assuming $D^\mathcal{I} = \mathfrak{t}(D)^\mathcal{J}$ we show two inclusions.

- For the first inclusion, take $d \in (\exists r.D)^\mathcal{I}$. Thus $d \in \Delta^\mathcal{I} (= \mathrm{Real}_+^\mathcal{J})$ and there is an $e \in \Delta^\mathcal{I}$ satisfying both $(d,e) \in r^\mathcal{I}$ and $e \in D^\mathcal{I}$. Note that $e \in \Delta^\mathcal{I}$ implies $e \in \mathrm{Real}_+^\mathcal{J}$. Moreover, by the equality $D^\mathcal{I} = \mathfrak{t}(D)^\mathcal{J}$ we have $e \in (\mathrm{Real}_+ \sqcap \mathfrak{t}(D))^\mathcal{J}$. Since $r^\mathcal{I} \subseteq r^\mathcal{J}$ holds we infer that $d \in (\exists r.(\mathrm{Real}_+ \sqcap \mathfrak{t}(D)))^\mathcal{J}$, but because d belongs to $\mathrm{Real}_+^\mathcal{J}$ we can conclude that $d \in (\mathrm{Real}_+ \sqcap \exists r.(\mathrm{Real}_+ \sqcap \mathfrak{t}(D)))^\mathcal{J} = \mathfrak{t}(\exists r.D)^\mathcal{J}$.
- For the opposite direction take $d \in \mathfrak{t}(\exists r.D)^\mathcal{J} = (\mathrm{Real}_+ \sqcap \exists r.(\mathrm{Real}_+ \sqcap \mathfrak{t}(D)))^\mathcal{J}$. This implies that $d \in \mathrm{Real}_+^\mathcal{J} (= \Delta^\mathcal{I})$ as well as that there is an $e \in \mathrm{Real}_+^\mathcal{J} (= \Delta^\mathcal{I})$ witnessing $(d,e) \in r^\mathcal{J}$ and $e \in \mathfrak{t}(D)^\mathcal{J} (= D^\mathcal{I})$. Since both d, e belong to $\Delta^\mathcal{I}$ we infer that $(d,e) \in r^\mathcal{I}$, and hence $d \in (\exists r.D)^\mathcal{I}$.

For the last statement of the lemma: to show that $\mathcal{J} \models \mathfrak{t}(C) \sqsubseteq \mathfrak{t}(D)$ if and only if $\mathcal{I} \models C \sqsubseteq D$ hold, it suffices to invoke the equalities $C^\mathcal{I} = \mathfrak{t}(C)^\mathcal{J}$ and $D^\mathcal{I} = \mathfrak{t}(D)^\mathcal{J}$. ☐

We employ the agreement lemma to show that the satisfiability of $\mathcal{O}_{red}$ implies the satisfiability of $\mathcal{O}$.

Lemma 5.40 If $\mathcal{O}_{red}$ is satisfiable then so is $\mathcal{O}$.

Proof. Let $\mathcal{J}$ be a model of $\mathcal{O}_{red}$ with $A^\mathcal{J} := A_+^\mathcal{J}$ (since A does not appear in $\mathcal{O}_{red}$ this can be assumed w.l.o.g.) for all concept names A from $\mathbf{C}_\mathcal{O}$. By the satisfaction of $\mathcal{O}_{corr}$ we know that $A_+^\mathcal{J}$ and $A_-^\mathcal{J}$ are disjoint and thus $\mathcal{J}$ is good-for-encoding. Hence, take $\mathcal{I}$ to be its induced subinterpretation with the domain $\mathrm{Real}_+^\mathcal{J}$. Applying Lemma 5.39 we know that for each GCI $C \sqsubseteq D$ from $\mathcal{O}$ the satisfaction of $\mathcal{J} \models \mathfrak{t}(C) \sqsubseteq \mathfrak{t}(D)$ implies $\mathcal{I} \models C \sqsubseteq D$. Thus we get $\mathcal{I} \models \mathcal{O}$, which implies that $\mathcal{O}$ is satisfiable. ☐

We next show that the satisfiability of $\mathcal{O}$ implies the satisfiability of $\mathcal{O}_{red}$. In the proof we basically take a model of $\mathcal{O}$, duplicate each domain element and define the memberships of fresh concepts introduced by $\mathcal{O}_{red}$. Such concepts are defined in such a way that if an element from a model $\mathcal{I}$ of $\mathcal{O}$ is a member of $A^\mathcal{I}$ then the corresponding element in a constructed model $\mathcal{J}$ of $\mathcal{O}_{red}$ is a member of $A_+^\mathcal{J}$ while its copy belongs to $A_-^\mathcal{J}$. This way the total number of elements in every concept is equal to half of the domain.

Lemma 5.41 If $\mathcal{O}$ is satisfiable then so is $\mathcal{O}_{red}$.

Proof. Let $\mathcal{I} \models \mathcal{O}$, and let $\Delta^\mathcal{I} := \{d_1, d_2, \ldots, d_n\}$. We define an interpretation $\mathcal{J}$ as follows:

1. $\Delta^\mathcal{J} := \{d_1, d_1', d_2, d_2', \ldots, d_n, d_n'\}$.
2. For all concept names $A \in \mathbf{C}_\mathcal{O}$ we put
   - $A_+^\mathcal{J} := A^\mathcal{J} := \{d_i \mid d_i \in A^\mathcal{I}\} \cup \{d_i' \mid d_i \not\in A^\mathcal{I}\}$,
   - $A_-^\mathcal{J} := \{d_i \mid d_i \not\in A^\mathcal{I}\} \cup \{d_i' \mid d_i \in A^\mathcal{I}\}$,
   - $\mathrm{Real}_+^\mathcal{J} := \Delta^\mathcal{I}$ and $\mathrm{Real}_-^\mathcal{J} := \Delta^\mathcal{J} \setminus \Delta^\mathcal{I}$,

   and for all other concept names B we put $B^\mathcal{I} := \Delta^\mathcal{I}$.
3. For each role name r we put $r^\mathcal{J} := r^\mathcal{I}$.
4. All individual names are interpreted as some auxiliary fixed domain element, say d_1.

We first show $\mathcal{J} \models \mathcal{O}_{corr}$. To this end, take any concept name $\mathrm{A} \in \{\mathrm{Real}\} \cup \mathbf{C}_{\mathcal{O}}$. We prove $\mathcal{J} \models (\mathrm{A}_+ \mid \mathrm{A}_-)\,[0,0]$, which by Fact 5.38 is equivalent to showing disjointness of $\mathrm{A}_+^{\mathcal{J}}$ and $\mathrm{A}_-^{\mathcal{J}}$. Assume the contrary, *i.e.* that there is a domain element $\mathrm{d} \in \mathrm{A}_+^{\mathcal{J}} \cap \mathrm{A}_-^{\mathcal{J}}$. If $\mathrm{d} = \mathrm{d}_i$ for some index i then, by Item 2, it means that $\mathrm{d}_i \in \mathrm{A}^{\mathcal{I}}$ and $\mathrm{d}_i \not\in \mathrm{A}^{\mathcal{I}}$ at the same time, which is clearly not possible. The case when $\mathrm{d} = \mathrm{d}'_i$ for some index i is treated similarly. Next, by invoking Item 2 of the definition of $\mathcal{J}$ we can perform some basic calculations:

$$|\mathrm{A}_+^{\mathcal{J}}| = |\{\mathrm{d}_i \mid \mathrm{d}_i \in \mathrm{A}^{\mathcal{I}}\}| + |\{\mathrm{d}'_i \mid \mathrm{d}_i \not\in \mathrm{A}^{\mathcal{I}}\}| =$$

$$|\{\mathrm{d}_i \mid \mathrm{d}_i \in \mathrm{A}^{\mathcal{I}}\}| + |\{\mathrm{d}_i \mid \mathrm{d}_i \not\in \mathrm{A}^{\mathcal{I}}\}| = |\Delta^{\mathcal{I}}| = \frac{1}{2} \cdot |\Delta^{\mathcal{J}}|.$$

$$|\mathrm{A}_-^{\mathcal{J}}| = |\{\mathrm{d}_i \mid \mathrm{d}_i \not\in \mathrm{A}^{\mathcal{I}}\}| + |\{\mathrm{d}'_i \mid \mathrm{d}_i \in \mathrm{A}^{\mathcal{I}}\}| =$$

$$|\{\mathrm{d}'_i \mid \mathrm{d}_i \not\in \mathrm{A}^{\mathcal{I}}\}| + |\{\mathrm{d}'_i \mid \mathrm{d}_i \in \mathrm{A}^{\mathcal{I}}\}| = |\{\mathrm{d}'_1, \ldots, \mathrm{d}'_n\}| = \frac{1}{2} \cdot |\Delta^{\mathcal{J}}|.$$

Hence by Fact 5.37 we get $\mathcal{J} \models (\mathrm{A}_+ \mid \top)\,[0.5, 0.5]$ and $\mathcal{J} \models (\mathrm{A}_- \mid \top)\,[0.5, 0.5]$, finishing the proof of $\mathcal{J} \models \mathcal{O}_{corr}$. To prove $\mathcal{J} \models \mathcal{O}_{tr}$ we take any GCI $\mathfrak{t}(\mathrm{C}) \sqsubseteq \mathfrak{t}(\mathrm{D})$ from $\mathcal{O}_{tr}$. From $\mathcal{I} \models \mathcal{O}$, we know that $\mathcal{I} \models \mathrm{C} \sqsubseteq \mathrm{D}$. Note that $\mathcal{J}$ is good-for-encoding, hence by Lemma 5.39 it follows that $\mathcal{J} \models \mathfrak{t}(\mathrm{C}) \sqsubseteq \mathfrak{t}(\mathrm{D})$. Thus $\mathcal{J} \models \mathcal{O}_{tr}$. □

Lemma 5.40 and Lemma 5.41 show that the presented reduction is correct. Since our reduction is polynomial for every $\mathcal{EL}^{\neg}$-ontology $\mathcal{O}$, by ExpTime-hardness of $\mathcal{EL}^{\neg}$ we can finally conclude Theorem 5.36.

An interesting direction for future work is to study the query entailment problem in the presence of probabilistic conditionals. We already obtained results for querying $\mathcal{ALCSCC}$+ERCBoxes (that generalise $\mathcal{SEL}$) in Corollary 5.33, but the presented technique of rebalancing is not applicable to generalisation of $\mathcal{ALC}$ that employ inverses or nominals. It would be also interesting to study fragments of existential rules [MT14] under the "probabilistic setting".

5.4 Pumping

In the last three sections we discussed sufficient conditions for testing whether a given description logic is Θ-forest-friendly, neglecting the case when the set of features Θ is non-empty. In this section, we design a novel model-theoretic technique, dubbed "pumping", that will help us in establishing $\{\mathsf{I}, \mathsf{Self}\}$-forest-friendliness of various logics, including the DL $\mathcal{ZIQ}$. An auxiliary definition first.

Definition 5.42 Let $\mathcal{I}$ be an interpretation and $\rho := \rho_1 \ldots \rho_n \in (\Delta^{\mathcal{I}})^*$ be an undirected cycle in $\mathcal{I}$, and $\mathsf{N} \subseteq \mathbf{N_I}$ be a set of names. We say that ρ an **N-cycle** if it contains an N-anonymous element and there is no index i satisfying $\rho_i = \rho_{i+1}$ or $\rho_i = \rho_{i+2}$. The **N-girth** of $\mathcal{I}$, denoted $\mathsf{girth}_{\mathsf{N}}(\mathcal{I})$, is the minimal length among all N-cycles in $\mathcal{I}$ (and ∞ if $\mathcal{I}$ does not contain any such cycles).

The notion of N-girth can be used as an alternative way to see $\{\mathsf{I}, \mathsf{Self}\}$-forest-likeness. Indeed:

Fact 5.43. Let $\mathcal{I}$ be an interpretation, n be a positive integer, and N be a finite set of names from $\mathbf{N_I}$. Then from the fact that the N-girth of $\mathcal{I}$ is at least n we can conclude that $\mathcal{I}$ is an $(n, \mathsf{N}, \{\mathsf{I}, \mathsf{Self}\})$-forest.

As an auxiliary step towards defining our "pumping" method, we introduce *twists*. The key idea is that, given an interpretation $\mathcal{I}$ and its elements c, d, c', and d', we replace all role connections between c and d with these between c' and d', and vice versa. Formally:

Definition 5.44 Let $\mathcal{I}$ be an interpretation, and $\mathrm{c}, \mathrm{d}, \mathrm{c}', \mathrm{d}'$ be its domain elements. The **$(\mathrm{c}, \mathrm{d}, \mathrm{c}', \mathrm{d}')$-twist** of $\mathcal{I}$ is the interpretation $\mathcal{J}$ of the same domain of $\mathcal{I}$, the same interpretation of individual names and concept names, but for every role name $r \in \mathbf{N_R}$, we put $r^{\mathcal{J}} := r^{\mathcal{I}} \setminus \{(\mathrm{c},\mathrm{d}), (\mathrm{c}',\mathrm{d}'), (\mathrm{d}',\mathrm{c}'), (\mathrm{d},\mathrm{c})\} \cup$

$$\Big(\{(\mathrm{c},\mathrm{d}) \mid (\mathrm{c}',\mathrm{d}') \in r^{\mathcal{I}}\} \cup \{(\mathrm{d},\mathrm{c}) \mid (\mathrm{d}',\mathrm{c}') \in r^{\mathcal{I}}\}\Big) \cup \Big(\{(\mathrm{c}',\mathrm{d}') \mid (\mathrm{c},\mathrm{d}) \in r^{\mathcal{I}}\} \cup \{(\mathrm{d}',\mathrm{c}') \mid (\mathrm{d},\mathrm{c}) \in r^{\mathcal{I}}\}\Big).$$

We can now define the "pumping" of finite interpretations, as formally stated below.

Definition 5.45 Let $\mathcal{I}$ be a finite interpretation with $\Delta^{\mathcal{I}} = \mathbb{Z}_{\mathrm{K}}$ for some positive integer K, let $\mathsf{N} \subseteq \mathbf{N_I}$ be a finite set of names, and G be a positive integer. Moreover, let $\mathrm{M} := \mathrm{K} \cdot (1 + \mathrm{K} + \ldots + \mathrm{K}^{\mathrm{G}+1}) + 1$. An (N, G)**-pumping** of $\mathcal{I}$ is the last interpretation in a sequence of interpretations $\mathcal{I}_0, \mathcal{I}_1, \ldots$ (called an (N, G)**-twisting-sequence**) for $\mathcal{I}$, defined inductively below.

- The interpretation $\mathcal{I}_0$ is defined as the disjoint sum of M isomorphic copies of $\mathcal{I}$. More formally, (i) we put $\Delta^{\mathcal{I}_0} := \Delta^{\mathcal{I}} \times \mathbb{Z}_{\mathrm{M}}$, (ii) for all individual names $\mathsf{a} \in \mathbf{N_I}$ we assign $\mathsf{a}^{\mathcal{I}_0} := \mathsf{a}^{\mathcal{I}}$, (iii) for all concept names $\mathrm{A} \in \mathbf{N_C}$ we set $\mathrm{A}^{\mathcal{I}_0} := \mathrm{A}^{\mathcal{I}} \times \mathbb{Z}_{\mathrm{M}}$, and (iv) for all role names $r \in \mathbf{N_R}$ we put $r^{\mathcal{I}_0} = \{((\mathrm{d}, m), (\mathrm{e}, m)) \mid m \in \mathbb{Z}_{\mathrm{M}}, (\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}\}$.
- Suppose that $\mathcal{I}_{i-1}$ is already defined. If the N-girth of $\mathcal{I}$ is at least G we stop the construction. Otherwise, $\mathsf{girth}_{\mathsf{N}}(\mathcal{I}_{i-1}) < \mathrm{G}$. Hence, fix any shortest N-cycle ρ in $\mathcal{I}_i$ witnessing that $\mathsf{girth}_{\mathsf{N}}(\mathcal{I}_{i-1})$ is less than G. Clearly, $|\rho| < \mathrm{G}$. Fix any pair (ρ_j, ρ_{j+1}) of two consecutive elements of ρ, so that ρ is N-anonymous. Such a pair exists by design. Note that there exists $k \in \mathbb{Z}_{\mathrm{M}}$ for which $\rho_j = (\mathrm{d}, k)$ and $\rho_{j+1} = (\mathrm{e}, k)$. By the choice of M, the following condition $(\heartsuit)$ is satisfied:

$$
\begin{gathered}
(\heartsuit)\colon \text{ “there exist elements } \varrho_j := (\mathrm{d}, \ell) \text{ and } \varrho_{j+1} := (\mathrm{e}, \ell) \text{ in } \mathcal{I}_{i-1} \\
\text{such that } \rho_j \text{ and } \rho_{j+1} \text{ are not reachable from} \\
\text{neither } \rho_j,\ \rho_{j+1}, \text{ nor any N-named element of } \mathcal{I}_{i-1} \text{ with a path of length smaller than G.”}
\end{gathered}
$$

  We take any such ϱ_j and ϱ_{j+1}, and define $\mathcal{I}_i$ as the $(\rho_j, \rho_{j+1}, \varrho_j, \varrho_{j+1})$-twist of $\mathcal{I}_{i-1}$.

We next elaborate on several useful properties of pumping, starting from its well-definedness.

Lemma 5.46 Let $\mathcal{I}$, K, M, G, $\mathcal{I}_0, \mathcal{I}_1 \ldots$ be as in Definition 5.45, and suppose that there is a positive index $i \leq \mathrm{M}$ for which $\mathsf{girth}_{\mathsf{N}}(\mathcal{I}_{i-1}) < \mathrm{G}$. Then the interpretation $\mathcal{I}_i$ is well-defined, namely for every shortest N-cycle ρ in $\mathcal{I}_{i-1}$ and every pair (ρ_j, ρ_{j+1}) of consecutive elements of ρ with ρ_{j+1} being N-anonymous, the condition $(\heartsuit)$ from Definition 5.45 holds.

Proof. Take any such ρ and a pair of elements (ρ_j, ρ_{j+1}) from $\mathcal{I}_{i-1}$. Before we start, a routine induction reveals that the size of every element in $\mathcal{I}_{i-1}$ is bounded by K, and for pair of elements (c, d) from $\mathcal{I}$ there are exactly M pairs of elements $(\mathrm{c}', \mathrm{d}')$ in $\mathcal{I}_{i-1}$ having their first coordinates equal to (c, d). By the size of 1-neighbourhoods, we know that the total number of elements reachable by paths of length at most G is bounded by the expression $(1 + \mathrm{K} + \ldots + \mathrm{K}^{\mathrm{G}+1})$. Hence, the total number of elements reachable from ρ_j, ρ_{j+1} or the N-named elements of $\mathcal{I}_{i-1}$ is bounded by $\mathrm{K} \cdot (1 + \mathrm{K} + \ldots + \mathrm{K}^{\mathrm{G}+1})$, *i.e.* $\mathrm{M}-1$. Thus, by the pigeonhole principle, we conclude the existence of an appropriate pair ϱ_j and ϱ_{j+1} satisfying the condition $(\heartsuit)$. □

Lemma 5.47 Let $\mathcal{I}$, K, M, G be as in Definition 5.45, and consider any (N, G)-twisting-sequence $\mathcal{I}_0, \mathcal{I}_1, \ldots$. Then such a sequence is finite, and the N-girth of its last element is at least G.

Proof. It suffices to show, for all indices i, that if the N-girth of $\mathcal{I}$ is equal to some g smaller than G, then (i) $\mathcal{I}_{i+1}$ does not contain N-cycles of length less than g, and that (ii) the total number of N-cycles of length g in $\mathcal{I}_{i+1}$ is strictly smaller than the total number of such cycles in $\mathcal{I}_i$. By construction, note that the N-cycle ρ selected from $\mathcal{I}_i$ in the construction of $\mathcal{I}_{i+1}$ no longer appears in $\mathcal{I}_{i+1}$. Moreover, let $(\rho_j, \rho_{j+1}, \varrho_j, \varrho_{j+1})$ be the 4-tuple for which $\mathcal{I}_{i+1}$ is the $(\rho_j, \rho_{j+1}, \varrho_j, \varrho_{j+1})$-twist of $\mathcal{I}_i$. Suppose now that there is a new N-cycle $\hat{\varrho}$ in $\mathcal{I}_{i+1}$ that does not appear in $\mathcal{I}_i$. This implies, that $\hat{\varrho}$ contains as a subword either (i) $\rho_j \cdot \varrho_{j+1}$, (ii) $\varrho_j \cdot \rho_{j+1}$, or their inverses. By the choice of ϱ_j, ϱ_{j+1} such elements are not reachable neither from ρ_j, ρ_{j+1} nor any N-named elements in $\mathcal{I}_i$ in less than G steps, and thus the length of the freshly created N-cycle is at G. The last statement of the lemma follows now by construction. □

We next proceed with the desired list of useful properties of "pumping".

Property 5.48. Let $\mathcal{I}$ be a finite interpretation, G be a positive integer, N be a finite set of names, and $\mathcal{J}$ be any (N, G)-pumping of $\mathcal{I}$. Then the analogue of Property 5.19 holds, namely:

(A) The interpretations $\mathcal{I}$ and $\mathcal{J}$ restricted to all N-named elements are isomorphic.

(B) For any concept name C we have that $\mathrm{C}^{\mathcal{I}}$ is non-empty if and only if $\mathrm{C}^{\mathcal{J}}$ is non-empty. Similarly, for any role name r we have that $r^{\mathcal{I}}$ is non-empty if and only if $r^{\mathcal{J}}$ is non-empty.

(C) For any $(\mathrm{d}, \ell) \in \Delta^{\mathcal{J}}$ we have $\mathsf{Conc}_{\mathcal{I}}(\mathrm{d}) = \mathsf{Conc}_{\mathcal{J}}((\mathrm{d}, \ell))$. Moreover, for all (e, ℓ') with $((\mathrm{d}, \ell), (\mathrm{e}, \ell')) \in r^{\mathcal{J}} \cup (r^{-})^{\mathcal{J}}$ for some $r \in \mathbf{N_R}$, we have $\mathsf{Rol}_{\mathcal{I}}(\mathrm{d}, \mathrm{e}) = \mathsf{Rol}_{\mathcal{J}}((\mathrm{d}, \ell), (\mathrm{e}, \ell'))$ and $\mathsf{Rol}_{\mathcal{I}}(\mathrm{e}, \mathrm{d}) = \mathsf{Rol}_{\mathcal{J}}((\mathrm{e}, \ell'), (\mathrm{d}, \ell))$.

(D) For all (d, ℓ) from $\mathcal{J}$ we have that (d, ℓ) in $\mathcal{J}$ and d in $\mathcal{I}$ have the same number of successors satisfying the same roles and concepts, *i.e.* an analogue of Item (D) of Property 5.5 holds.

(E) For any (d, ℓ) from $\mathcal{J}$ we have that (d, ℓ) and d are path-equivalent, *i.e.* an analogue of Item (E) of Property 5.5 holds.

(F) $\mathcal{I}$ and $\mathcal{J}$ are homomorphically equivalent.

Proof sketch. The proof is routine and similar to the previous proofs of this chapter, and hence we only provide its key ingredients. We proceed inductively, over the (N, G)-twisting-sequence $\mathcal{I}_0, \mathcal{I}_1, \ldots, \mathcal{J}$ used to construct $\mathcal{J}$. The inductive claim states, for all indices i, that:

- For all $\ell \leq \mathrm{M}$ we have that the 1-neighbourhood of (d, ℓ) in $\mathcal{I}_i$ and the 1-neighbourhood of d in $\mathcal{I}$ are isomorphic, for all $\mathrm{d} \in \Delta^{\mathcal{I}}$ and $\ell \leq \mathrm{M}$.
- The projection onto the first coordinate is a homomorphism from $\mathcal{I}_i$ to $\mathcal{I}$, and the mapping $\mathrm{d} \mapsto (\mathrm{d}, 0)$ is a homomorphism in a reverse direction.

With the above ingredients, establishing the properties (A)–(F) is rather immediate. □

To employ the above properties to derive results on freshly defined logics, we propose a broad class of DLs that are preserved under pumping. The definition is given below.

Definition 5.49 A description logic $\mathcal{DL}$ is **preserved under pumping** if for all finitely-satisfiable $\mathcal{DL}$-knowledge-bases $\mathcal{K}$, all finite models $\mathcal{I}$ of $\mathcal{K}$, all but finitely many positive integers G, and all $(\mathsf{ind}(\mathcal{K}), \mathrm{G})$-pumpings $\mathcal{J}$ of $\mathcal{I}$ we have that $\mathcal{I} \models \mathcal{K}$ implies $\mathcal{J} \models \mathcal{K}$.

Here comes our standard theorem that relates the logics with forest-friendliness. Indeed:

Theorem 5.50

If a description logic $\mathcal{DL}$ is preserved under pumpings then $\mathcal{DL}$ is finitely $\{\mathsf{I}, \mathsf{Self}\}$-forest-friendly.

Proof. Take any finitely satisfiable $\mathcal{DL}$-KB $\mathcal{K}$, any of its finite models $\mathcal{I}$, and any positive integer $m \in \mathbb{N}$. Following the definition, it suffices to show that there exists a finite $(m, \mathsf{ind}(\mathcal{K}), \{\mathsf{I}, \mathsf{Self}\})$-forest model $\mathcal{J} \models \mathcal{K}$ that covers $\mathcal{I}$. Take $\mathcal{J}$ to be any $(\mathsf{ind}(\mathcal{K}), m)$-pumping of $\mathcal{I}$. Note that $\mathcal{J}$ is $(m, \mathsf{ind}(\mathcal{K}), \{\mathsf{I}, \mathsf{Self}\})$-forest by Fact 5.43. By preservation under pumpings, we infer $\mathcal{J} \models \mathcal{K}$. By Item (F) of Property 5.48 we know that $\mathcal{J}$ covers $\mathcal{I}$. This concludes the proof. □

Recall that $\mathcal{ZIQ}$ is the sublogic of $\mathcal{ZOIQ}$ that features role hierarchies ($\mathcal{H}$), (safe) boolean role combinations (b), counting ($\mathcal{Q}$), inverses of roles ($\mathcal{I}$), the Self operator, and regular path expressions ($\cdot_{\mathsf{reg}}$). It is not difficult to establish that our pumping method preserves the modelhood of TBoxes that employ such features. Indeed, the preservation of GCIs involving ($\mathcal{H}$), (b), ($\mathcal{I}$), and Self is due to Item (C) of Property 5.48, the preservation of GCIs involving ($\mathcal{Q}$) follows from Item (D) of Property 5.48, and the preservation of GCIs involving ($\cdot_{\mathsf{reg}}$) is due to Item (E) of Property 5.48. Preservation of the satisfaction of ABoxes is immediate by Item (A) of Property 5.48. Thus:

Corollary 5.51

The description logic $\mathcal{ZIQ}$ is preserved under pumping. Thus, $\mathcal{ZIQ}$ is finitely $\{\mathsf{I}, \mathsf{Self}\}$-forest-friendly.

We can now employ Theorem 4.37 to infer the following conditional result. The matching lower bound follows from the CQ-entailment over $\mathcal{ALCI}$-TBoxes by the result of Lutz [Lut08a, Thm. 2].

Theorem 5.52

If the finite knowledge-base satisfiability problem for $\mathcal{ZIQ}$ is ExpTime-complete, the PEQ-entailment problem over $\mathcal{ZIQ}$-KBs is 2ExpTime-complete.

It is reasonable to conjecture that the finite satisfiability problem for $\mathcal{ZIQ}$-KBs is ExpTime-complete. Unfortunately, we were not even able to establish decidability of the logic, which remains till now a challenging open problem. The main difficulty arising when attempting to solve the finite satisfiability problem for $\mathcal{ZIQ}$ is that it seems to require fundamentally different techniques from the one used for $\mathcal{ALCIQ}$ and its generalisations. More precisely, the only currently known technique for dealing with the finite satisfiability of $\mathcal{ALCIQ}$-KBs relies on a mosaic method and a reduction to integer programming [Pra07, Thm. 1], which does not seem to be suitable for dealing with regular path expressions.

We provide two further applications of our techniques. It was recently shown by Jung et al. [JLZ20, Thm. 14, Thm. 18] that the finite knowledge-base satisfiability problems for certain fragments of $\mathcal{ZIQ}$, called $\mathcal{ALCIF}^1_{\mathsf{reg}}$ and $\mathcal{ALCIF}^2_{\mathsf{reg}}$, are decidable in NExpTime and 2NExpTime, respectively. The algorithm of Jung et al. [JLZ20, Lemma 17, Lemma 20] is automata-based, and hence it is relatively easy to accommodate conjunctions of roles and self-loops easily.

Corollary 5.53

Assuming that the algorithms of Jung et al. [JLZ20, L. 17 and L. 20] can be adjusted to accommodate conjunction of roles $\cdot^{\cap}$ and the Self operator, the finite entailment of PEQs over knowledge-bases written in $\mathcal{ALCIF}^1_{\mathsf{reg}}$ or $\mathcal{ALCIF}^2_{\mathsf{reg}}$ is decidable, respectively, in 2NExpTime and 3NExpTime.

Finally, we employ Theorem 4.37 to lift the existing results on CQ entailment for fragments of $\mathcal{ZIQ}$ contained in $\mathcal{ALCIH}b^{\mathsf{Self}}_{\mathsf{reg}}\mathcal{Q}$ to the setting of UCQs and PEQs. The matching lower bound for PEQs holds already for $\mathcal{ALC}$ [Ov14, Thm. 1], and the ExpTime-completeness of the knowledge-base satisfiability problem follows by work of Pratt-Hartmann [PH23, Thm. 8.22].

Corollary 5.54

Let $\mathcal{DL}$ be a description logic that subsumes $\mathcal{ALC}$ and is contained in $\mathcal{ALCIH}b^{\mathsf{Self}}_{\mathsf{reg}}\mathcal{Q}$. Then the finite entailment problem for PEQs over $\mathcal{DL}$-KBs is 2ExpTime-complete.

In particular, the above corollary applies to the two-variable guarded fragment of first-order logic with counting quantifiers, and confirms the conjecture by Ibáñez-García et al. [ILS14, l. 34–39, col.1, p. 9]

6

Lower Bounds On Querying $\mathcal{ALC}$Self

Contents

Motivation and Our Contribution

Our results from previous chapters provide a complete classification of the conjunctive query entailment problem for extensions of $\mathcal{ALC}$ with various features supported by the OWL 2 Web Ontology Language and the DL $\mathcal{SROIQ}$ [HKS06]. However, the mentioned classification does not settle the case of the Self operator. The Self operator allows us to specify the situation when an element is related to it*self* by a binary relationship. Among other things, this allows us to formalise the concept of a "narcissist" with Narcissist $\sqsubseteq \exists loves$.Self. Due to the simplicity of the Self operator (it only refers to one element), it is easy to accommodate for automata techniques [CEO09] or consequence-based methods [ORv10] and thus, so far, there has been no real indication that the added expressivity provided by Self may change anything, complexity-wise. Arguably, this impression is further corroborated by the observation that Self features in two profiles of OWL 2 (the EL and the RL profile), again without harming tractability [KRH08]. We show however, a rather surprising result, namely that conjunctive query entailment for $\mathcal{ALC}^{\mathsf{Self}}$ is exponentially harder than for $\mathcal{ALC}$. Our proof goes via encoding of computation trees of alternating Turing machines working in exponential space and follows the general hardness-proof-scheme by Lutz [Lut08a] However, to adjust the schema to $\mathcal{ALC}$Self, novel ideas are required: the ability to speak about self-loops is exploited to produce a single query that traverses trees in a root-to-leaf manner and to simulate disjunction inside conjunctive queries, useful to express that certain paths are repeated inside the tree. Our hardness result can be additionally employed to establish 2ExpTime-hardness of query entailment for the $\mathcal{Z}$ family (a.k.a. $\mathcal{ALCH}b^{\mathsf{Self}}_{\mathsf{reg}}$) of DLs [CEO09], which until now has remained open[1] as well as the 2ExpTime-hardness of querying ontologies formulated in the fluted guarded fragment [Bed21a] with equality.[2]

[1]We stress that 2ExpTime hardness of the conjunctive query entailment problem over $\mathcal{Z}$ ontologies *does not* follow from 2ExpTime-hardness of the same problem for $\mathcal{SH}$ since we cannot define in $\mathcal{Z}$ that a given role is transitive nor that it is a transitive closure of another role (to simulate transitivity).

[2]Self can be expressed with $\forall x_1 \mathrm{self}_{\mathrm{r}}(x_1) \to \exists x_2\, (r(x_1, x_2) \land x_1{=}x_2) \;\land\; \forall x_1 \forall x_2 r(x_1, x_2) \to (x_1{=}x_2 \to \mathrm{self}_{\mathrm{r}}(x_2))$.

Overview of the Chapter and Prerequisites

We assume that the reader is familiar with basic definitions concerning $\mathcal{ALC}$ and $\mathcal{ALC}^{\mathsf{Self}}$, and the definitions concerning the query entailment problem from Chapter 2. Section 6.1 fixes required definitions concerning Alternating Turing Machines (ATMs). Then, Section 6.2 presents a high-level overview of our construction. The goal of Section 6.3, Section 6.4, Section 6.5, and Section 6.6 is to gradually define interpretations encoding possibly faulty-runs of ATMs. Section 6.7 presents a conjunctive query detecting errors in encodings of consecutive configurations, and concludes the main theorem of this chapter (Theorem 6.18).

6.1 Extra Preliminaries: Alternating Turing Machines

We first fix the notation of *alternating Turing machines* over a binary alphabet $\{\mathtt{0},\mathtt{1}\}$ working in exponential space (simply: *ATMs*). As a convention, when speaking about ATMs, their configurations and tape contents, we employ the `typewriter` font.

Definition 6.1 An **ATM** is a tuple $\mathcal{M} := (\mathtt{N}, \mathtt{Q}, \mathtt{Q}_{\exists}, \mathtt{s}_I, \mathtt{s}_A, \mathtt{s}_R, \mathtt{T})$, where $\mathtt{Q}$ is a finite set of *states* (usually denoted with $\mathtt{s}$); $\mathtt{Q}_{\exists} \subseteq \mathtt{Q}$ is a set of *existential* states; $\mathtt{s}_I, \mathtt{s}_A, \mathtt{s}_R \in \mathtt{Q}$ are, respectively, pairwise different *initial*, *accepting*, and *rejecting* states; we assume that $\mathtt{s}_I \in (\mathtt{Q} \setminus \mathtt{Q}_{\exists})$; $\mathtt{T} \subseteq (\mathtt{Q} \times \{\mathtt{0},\mathtt{1}\}) \times (\{\mathtt{0},\mathtt{1}\} \times \mathtt{Q} \times \{-1,+1\})$ is the *transition relation*; and the natural number $\mathtt{N}$ (encoded in unary) is a parameter governing the size of the working tape. We call the states from $\mathtt{Q}_{\forall} := \mathtt{Q} \setminus \mathtt{Q}_{\exists}$ *universal*. The *size* of $\mathcal{M}$, denoted with $|\mathcal{M}|$, is defined as $\mathtt{N} + |\mathtt{Q}| + |\mathtt{Q}_{\exists}| + 3 + |\mathtt{T}|$.

We next discuss the notion of configurations. A *configuration* of $\mathcal{M}$ is a word $\mathtt{wsw}' \in \{\mathtt{0},\mathtt{1}\}^*\mathtt{Q}\{\mathtt{0},\mathtt{1}\}^*$ with $|\mathtt{ww}'| = 2^{\mathtt{N}}$. We call the configuration $\mathtt{wsw}'$ (i) *existential* (resp. *universal*) if $\mathtt{s}$ is existential (resp. universal), (ii) *final* if $\mathtt{s}$ is either $\mathtt{s}_A$ or $\mathtt{s}_R$ (iii) *non-final* if it is not final (iv) *accepting* if $\mathtt{s} = \mathtt{s}_A$. *Successor configurations* are defined in terms of the transition relation $\mathtt{T}$. For $\mathtt{a},\mathtt{b},\mathtt{c},\mathtt{d} \in \{\mathtt{0},\mathtt{1}\}$ and $\mathtt{v},\mathtt{v}',\mathtt{w},\mathtt{w}' \in \{\mathtt{0},\mathtt{1}\}^*$ with $|\mathtt{v}| = |\mathtt{w}|$, we let $\mathtt{wbs}'\mathtt{w}'$ be a *quasi-successor* configuration of $\mathtt{vsav}'$ whenever $(\mathtt{s},\mathtt{a},\mathtt{b},\mathtt{s}',+1) \in \mathtt{T}$, and we let $\mathtt{ws}'\mathtt{dbw}'$ be a quasi-successor configuration of $\mathtt{vcsav}'$ whenever $(\mathtt{s},\mathtt{a},\mathtt{b},\mathtt{s}',-1) \in \mathtt{T}$. If additionally $\mathtt{w} = \mathtt{v}$, $\mathtt{w}' = \mathtt{v}'$, and $\mathtt{c} = \mathtt{d}$ hold we speak of *successor* configurations.[3]

Remark 6.2. W.l.o.g we make the following additional assumptions about $\mathcal{M}$. First, for each non-final (*i.e.* non-accepting and non-rejecting) state $\mathtt{s}$ and every letter $\mathtt{a} \in \{\mathtt{0},\mathtt{1}\}$ the set $\mathtt{T}(\mathtt{s},\mathtt{a}) := \{(\mathtt{s},\mathtt{a},\mathtt{b},\mathtt{s}',d) \in \mathtt{T}\}$ contains exactly two elements, denoted $\mathtt{T}_1(\mathtt{s},\mathtt{a})$ and $\mathtt{T}_2(\mathtt{s},\mathtt{a})$. Hence, every configuration has exactly two successor configurations. Second, for any $(\mathtt{s},\mathtt{a},\mathtt{b},\mathtt{s}',d) \in \mathtt{T}$, if $\mathtt{s}$ is existential then $\mathtt{s}'$ is universal and vice versa. Third, the machine reaches a final state no later than after $2^{2^{\mathtt{N}}}$ steps (for configuration sequences). Fourth and last, $\mathcal{M}$ never attempts to move left (resp. right) on the left-most (resp. right-most) tape cell.

We next define runs of ATMs. A *run* of $\mathcal{M}$ is a finite tree, with nodes labelled by configurations of $\mathcal{M}$, that satisfies all the conditions below:

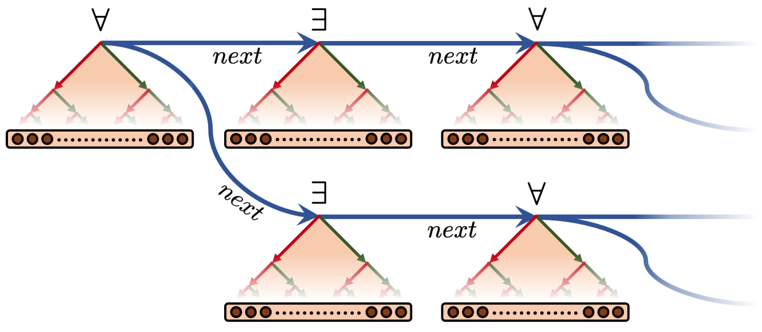
- the root is labelled with the initial configuration $\mathtt{s}_I\mathtt{0}^{2^{\mathtt{N}}}$,
- each node labelled with a non-final existential configuration $\mathtt{wsw}'$ has a single child node which is labelled with one of the successor configurations of $\mathtt{wsw}'$,
- each node labelled with a non-final universal configuration $\mathtt{wsw}'$ has two child nodes which are labelled with the two successor configurations (wrt. $\mathtt{T}_1$ and $\mathtt{T}_2$) of $\mathtt{wsw}'$,
- no node labelled with a final configuration has successors.

Quasi-runs of $\mathcal{M}$ are defined analogously by replacing the notions of successors with quasi-successors. Note that every run is also a quasi-run but not vice versa. An ATM $\mathcal{M}$ is (quasi-)*accepting* if it has an *accepting* (quasi)-run, *i.e.* one whose all leaves are labelled by accepting configurations. By results of Chandra et al. [CKS81, Corollary 3.6] the problem of checking if a given ATM is accepting is 2ExpTime-hard.

[3]In words, this corresponds to the common definition of successor configurations, while for quasi-successor configurations, untouched tape cells may change arbitrarily during the transition.

6.2 A High-Level Overview of the Encoding

Let $\mathcal{M}$ be an ATM. The core contribution of this section is to present a polynomial-time reduction that, given $\mathcal{M}$, constructs a pair $(\mathcal{K}_{\mathcal{M}}, q_{\mathcal{M}})$ — composed of an $\mathcal{ALC}^{\mathsf{Self}}$ knowledge base and a conjunctive query — such that $\mathcal{K}_{\mathcal{M}} \not\models q_{\mathcal{M}}$ if and only if $\mathcal{M}$ is accepting. Intuitively, the models of $\mathcal{K}$ will encode accepting quasi-runs of $\mathcal{M}$, *i.e.* trees in which every node is a meaningful configuration of $\mathcal{M}$, but the tape contents of consecutive configurations might not be in sync as they should. The query $q_{\mathcal{M}}$ will be responsible for detecting such errors. Hence, the existence of a countermodel for $\mathcal{K}_{\mathcal{M}}$ and $q_{\mathcal{M}}$ will coincide with the existence of an accepting run of $\mathcal{M}$. The intended models of $\mathcal{K}_{\mathcal{M}}$ look as follows:



The depicted triangles are called the *configuration trees* and encode configurations of $\mathcal{M}$. The information contained in these configuration trees is "superimposed" on identical *configuration units*: full binary trees of height $\mathtt{N}+1$ decorated with many self-loops[4] that will provide the "navigational infrastructure" for the query $q_{\mathcal{M}}$ to detect "tape mismatches". Every such tree has $2^{\mathtt{N}}$ nodes at its $\mathtt{N}$-th level and each of these nodes represents a single tape cell of a machine. However, somehow unexpectedly, we do not just label them directly with concepts representing a letter from the alphabet. Instead, every node at the $\mathtt{N}$-th level also has two children labelled (from left to right) respectively with either $\mathbb{0}$ and $\mathbb{1}$, or with $\mathbb{1}$ and $\mathbb{0}$. Whenever the left child is in $\mathbb{0}$ and the right child is in $\mathbb{1}$, we think that their parent represents a cell filled with the letter $\mathtt{0}$, while the converse situation encodes a cell filled with $\mathtt{1}$.

encoding of tape cell carrying symbol $\mathtt{0}$ encoding of tape cell carrying symbol $\mathtt{1}$

This encoding will be useful to avoid a seemingly required disjunction in the construction of $q_{\mathcal{M}}$. Lastly, the roots of configuration units store all remaining necessary information required for encoding: the current state of $\mathcal{M}$, the previous and the current head position as well as the transition used to arrive at this node from the previous configuration. Finally, the roots of configuration trees are interconnected by the role *next* indicating that $(\mathrm{r}, \mathrm{r}') \in next^{\mathcal{I}}$ holds if and only if the configuration represented by r' is a quasi-successor of the configuration of r.

6.3 Configuration Units

In our encoding, a vital role is played by *n-configuration units*, which will later form the backbone of configuration trees. Roughly speaking, each n-configuration unit is a full binary tree of depth n, decorated with certain concepts, roles, and self-loops. We introduce configuration units by providing the formal definition, followed by a graphical depiction and an intuitive description. In order to represent configuration

[4]The concrete purpose of the abundant presence of self-loops will only become clear later, starting from Corollary 6.7.

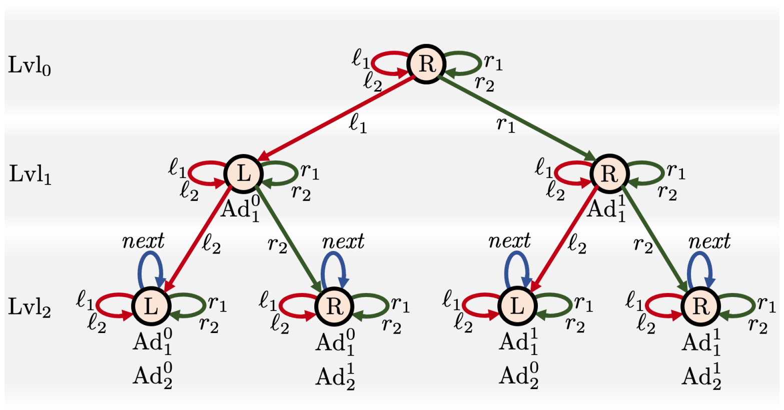
units inside interpretations, we employ role names from $\mathbf{R}_{unit}$ as well as concept names from $\mathbf{C}_{unit}$:

$$\mathbf{R}_{unit} \coloneqq \{\ell_i, r_i, next \mid 1 \le i \le n\}, \qquad \mathbf{C}_{unit} \coloneqq \{\mathrm{Lvl}_0, \mathrm{Lvl}_i, \mathrm{L}, \mathrm{R}, \mathrm{Ad}_i^0, \mathrm{Ad}_i^1 \mid 1 \le i \le n\}.$$

Definition 6.3 (configuration unit) Given a positive integer n, an **n-configuration unit $\mathcal{U}$** is an interpretation $(\Delta^{\mathcal{U}}, \cdot^{\mathcal{U}})$ fulfilling all the conditions below:

- $\Delta^{\mathcal{U}} = \{0,1\}^{\le n} \coloneqq \{w \in \{0,1\}^* \mid |w| \le n\}$,
- $\mathrm{L}^{\mathcal{U}} \setminus \{\varepsilon\} = \{w0 \in \Delta^{\mathcal{U}}\}$, $\mathrm{R}^{\mathcal{U}} = \Delta^{\mathcal{U}} \setminus \mathrm{L}^{\mathcal{U}}$,
- $\mathrm{Lvl}_i^{\mathcal{U}} = \{w \in \Delta^{\mathcal{U}} \mid |w| = i\}$, $next^{\mathcal{U}} = \{(w,w) \mid |w| = n\}$,
- $\ell_i^{\mathcal{U}} = \{(w, w0) \mid |w| = i{-}1\} \cup \{(w,w) \mid w \in \Delta^{\mathcal{U}}\}$,
- $r_i^{\mathcal{U}} = \{(w, w1) \mid |w| = i{-}1\} \cup \{(w,w) \mid w \in \Delta^{\mathcal{U}}\}$,
- $(\mathrm{Ad}_i^b)^{\mathcal{U}} = \{w \in \Delta^{\mathcal{U}} \mid |w| \ge i$ and its i-th letter is $b\}$.

The following drawing depicts a 2-configuration unit.



As one can see, the nodes in the tree are layered into levels according to their distance from the root. Nodes at the i-th level are members of the Lvl_i concept and their distance from the root is equal to i. Next, each non-leaf node at the i-th level has two children, the left one and the right one (satisfying, respectively, the concepts L and R) and is connected to them via the role ℓ_i and r_i, respectively. All nodes are equipped with ℓ_i- and r_i-self-loops and all leaves are additionally endowed with *next*-loops. With all nodes inside the tree, we naturally associate their addresses, *i.e.* their "numbers" when nodes from the i-th level are enumerated from left to right. In order to encode the address of a given node at the i-th level, we employ concepts $\mathrm{Ad}_1^b, \mathrm{Ad}_2^b, \ldots, \mathrm{Ad}_i^b$ with "values" b either 0 or 1, meaning that a node is in Ad_j^b if and only if the j-th bit of its address is equal to b. The most significant bit is Ad_1^b.

We next proceed with an axiomatisation of n-configuration units in $\mathcal{ALC}^{\mathsf{Self}}$, obtained with the forthcoming GCIs. As usual in such encodings, we cannot formalise such structures up to isomorphism, but the axiomatisation provided is sufficient in a sense made formally precise in the subsequent lemmas.

1. Each node is at exactly one level.

   (LvlCov) $\top \sqsubseteq \bigsqcup_{i=0}^{n} \mathrm{Lvl}_i$

   (LvlDisj[i,j]) $\mathrm{Lvl}_i \sqcap \mathrm{Lvl}_j \sqsubseteq \bot$ (with $0 \le i < j \le n$)

2. All nodes carry self-loops for all role names from $\mathbf{R}_{unit}$ except *next* and all leaf nodes (and only they) carry a *next*-loop.

   (all-loops-but-next) $\top \sqsubseteq \bigsqcap_{s \in \mathbf{R}_{unit} \setminus \{next\}} \exists s.\mathsf{Self}$

   (leaves-next-loop) $\mathrm{Lvl}_n \equiv \exists next.\mathsf{Self}$

3. Every node is either a "left" node or a "right" node.

(LRCov) $\top \sqsubseteq \mathrm{L} \sqcup \mathrm{R}$ **(LRDisj)** $\mathrm{L} \sqcap \mathrm{R} \sqsubseteq \bot$

4. Each node at any level $0 \leq i < n$ has two successors (one left and one right).

   (LsuccLvl[i]) $\mathrm{Lvl}_i \sqsubseteq \exists \ell_{i+1}.(\mathrm{Lvl}_{i+1}) \sqcap \forall \ell_{i+1}.(\mathrm{Lvl}_{i+1} \to \mathrm{L})$
   (RsuccLvl[i]) $\mathrm{Lvl}_i \sqsubseteq \exists r_{i+1}.(\mathrm{Lvl}_{i+1}) \sqcap \forall r_{i+1}.(\mathrm{Lvl}_{i+1} \to \mathrm{R})$

5. Address information for the nodes is created bit-wise and propagated down the tree. That is, once we are in the left (resp. right) node on the i-th level, this node and all nodes further below will have the i-th bit of their address set to 0 (resp. 1). Below we have $1 \leq i \leq n$, $b \in \{0,1\}$ and $0 \leq j < i$.

   (LBitZero[i]) $\mathrm{Lvl}_i \sqcap \mathrm{L} \sqsubseteq \mathrm{Ad}_i^0$ **(RBitOne[i])** $\mathrm{Lvl}_i \sqcap \mathrm{R} \sqsubseteq \mathrm{Ad}_i^1$
   (AdDisj[i]) $\mathrm{Ad}_i^0 \sqcap \mathrm{Ad}_i^1 \sqsubseteq \bot$ **(AdLvlDisj[i,j])** $\mathrm{Ad}_i^b \sqcap \mathrm{Lvl}_j \sqsubseteq \bot$

   (PropBit[i]) $\mathrm{Ad}_i^b \sqsubseteq \bigsqcap_{j=1}^{n} \forall \ell_j.\mathrm{Ad}_i^b \sqcap \forall r_j.\mathrm{Ad}_i^b$

This finishes the axiomatisation of n-configuration units. Let $\mathcal{K}_{unit}^n$ denote the KB composed of all GCIs presented so far. What remains to be done is to show that our axiomatisation is correct, in the sense of the following two lemmas. Their proofs are routine, hence the reader may skip them at first reading.

Lemma 6.4 Each n-configuration unit is a model of $\mathcal{K}_{unit}^n$.

Proof. Let $\mathcal{U}$ be an n-configuration unit. We proceed with axioms α of $\mathcal{K}_{unit}^n$, showing $\mathcal{U} \models \alpha$.

1. Note that $\Delta^{\mathcal{U}}$ is a set of all binary words of length $\leq n$ and by definition we have that $w \in \mathrm{Lvl}_i^{\mathcal{U}}$ if and only if $|w| = i$. Since every word w has a *unique* length, it follows that $\mathcal{U} \models$ (LvlCov) and that $\mathcal{U} \models$ (LvlDisj[i,j]) for any indices $0 \leq j < i \leq n$.
2. By definitions of $\mathrm{Lvl}_n^{\mathcal{U}}$ and $next^{\mathcal{U}}$, we immediately conclude $\mathcal{U} \models$ (leaves-next-loop). Moreover, for any role name s from $\mathbf{R}_{unit} \setminus \{next\}$, we conclude $\mathcal{U} \models \exists s.\mathsf{Self}$ from the fact that the set $\{(w,w) \mid \in \Delta^{\mathcal{U}}\}$ is explicitly stated as a part of $s^{\mathcal{U}}$ in its definition. Thus, we infer $\mathcal{U} \models$ (all-loops-but-next).
3. The satisfaction of (LRCov) and (LRDisj) by $\mathcal{U}$ is due to the equality $\mathrm{R}^{\mathcal{U}} = \Delta^{\mathcal{U}} \setminus \mathrm{L}^{\mathcal{U}}$.
4. Suppose $i < n$. We will show that $\mathcal{U} \models$ (LsuccLvl[i]) (the satisfaction of (RsuccLvl[i]) is analogous). Hence, take any $w \in \mathrm{Lvl}_i^{\mathcal{U}}$. Then w (by definition of ℓ_{i+1}) has exactly two ℓ_{i+1}-successors: w and $w0$. Moreover, by definition of $\mathrm{Lvl}_{i+1}^{\mathcal{U}}$ and $\mathrm{L}^{\mathcal{U}}$ we conclude that $w0 \in \mathrm{Lvl}_{i+1}^{\mathcal{U}} \cap \mathrm{L}^{\mathcal{U}}$ and $w \notin \mathrm{Lvl}_{i+1}$. Hence, $\mathcal{U} \models$ (LsuccLvl[i]) holds, since $w0$ is the required (only) witness for the $\exists$- ($\forall$-) restriction.
5. To see $\mathcal{U} \models$ (LBitZero[i]), it suffices to take any element $w \in \mathrm{Lvl}_i^{\mathcal{U}} \cap \mathrm{L}^{\mathcal{U}}$. By the first inclusion we infer that $|w| = i$ and by the second that the last letter of w is 0. Hence, we are done by the definition of $(\mathrm{Ad}_i^b)^{\mathcal{U}}$. Similarly, we get $\mathcal{U} \models$ (RBitOne[i]). The property $\mathcal{U} \models$ (AdDisj[i]) is due to the fact that words from $\mathcal{U}$ carry only one letter per position. Next, to show $\mathcal{U} \models$ (AdLvlDisj[i,j]) for any $1 \leq j < i \leq n$ it suffices to see that, by definition, $\mathrm{Lvl}_j^{\mathcal{U}}$ contains words of length $= j$ and $(\mathrm{Ad}_i^b)^{\mathcal{U}}$ contains words of length $\geq i$. Thus their intersection is empty, implying the satisfaction of (AdLvlDisj[i,j]). Finally, we need to prove $\mathcal{U} \models$ (PropBit[i]) for $1 \leq i \leq n$. To this end, note that $w \in (\mathrm{Ad}_i^b)^{\mathcal{U}}$ is equivalent to saying that $|w| \geq i$ and that the i-th letter of w is b. Now observe that, by definition, that for every $s \in \mathbf{R}_{unit} \setminus \{next\}$ we have that any s-successor of w can only be w, $w0$, or $w1$ (if $i \neq 0$). In any case, such a successor has length $\geq i$ and has its i-th letter equal to b. Thus its membership in $(\mathrm{Ad}_i^b)^{\mathcal{U}}$ follows, finishing the proof.

□

The proof of the next lemma is established by constructing an n-configuration unit. It starts from an element d and recursively traverse ℓ_i and r_i roles.

Lemma 6.5 For any model $\mathcal{I}$ of $\mathcal{K}^n_{unit}$ and any $\mathrm{d} \in \mathrm{Lvl}_0^{\mathcal{I}}$ there is an n-configuration unit $\mathcal{U}$ and a homomorphism $\mathfrak{h}$ from $\mathcal{U}$ into $\mathcal{I}$ with $\mathfrak{h}(\varepsilon) = \mathrm{d}$.

Proof. Let $\mathcal{U}$ be an n-configuration unit with $\varepsilon \in \mathrm{L}^{\mathcal{U}}$ if and only if $\mathrm{d} \in \mathrm{L}^{\mathcal{I}}$, and that interprets of all role and concept names outside $\mathbf{R}_{unit} \cup \mathbf{C}_{unit}$ as empty sets. It is obvious that exactly one such unit exists. In what follows, we are going to define a function $\mathfrak{h}\colon \Delta^{\mathcal{U}} \to \Delta^{\mathcal{I}}$ inductively. Denoting the restriction of $\mathfrak{h}$ to $\{0,1\}^{\leq k}$ by $\mathfrak{h}_{\leq k}$, our inductive assumption states, for a given $k \leq n$, that $\mathfrak{h}_{\leq i}$ is defined for all $i < k$ and $\mathfrak{h}_{\leq i}$ is a homomorphism from $\mathcal{U}\restriction_{\{0,1\}^{\leq k}}$ to $\mathcal{I}$.

We first set $\mathfrak{h}(\varepsilon) := \mathrm{d}$. It is immediate to check that $\mathfrak{h}_{\leq 0}$ is indeed a homomorphism (by $\mathcal{I} \models$ (all-loops-but-next) and our assumptions on $\varepsilon \in \mathrm{L}^{\mathcal{U}}$, and on empty interpretations of symbols outside $\mathbf{R}_{unit} \cup \mathbf{C}_{unit}$). For the inductive step, suppose that the assumption holds for some $1 \leq k \leq n$ and take a word $w \in \{0,1\}^{k-1}$. We are going to define $\mathfrak{h}(w0)$ as follows (the case of $\mathfrak{h}(w1)$ is symmetric). Note that since $\mathfrak{h}(w) \in \mathrm{Lvl}_{k-1}^{\mathcal{I}}$ (by the fact that $\mathfrak{h}_{\leq k-1}$ is a homomorphism) and since $\mathcal{I} \models$ (LsuccLvl[i]) (for i equal to $k-1$) we conclude the existence of $\mathrm{d}' \in \mathrm{Lvl}_k^{\mathcal{I}}$ satisfying $(\mathfrak{h}(w), \mathrm{d}') \in \ell_k^{\mathcal{I}}$. Note that also $\mathrm{d}' \in \mathrm{L}^{\mathcal{I}} \cap (\mathrm{Ad}_k^0)^{\mathcal{I}}$ holds (by $\mathcal{I} \models$ (LsuccLvl[i]) and $\mathcal{I} \models$ (LBitZero[i]) with $i = k$). Thus, we simply let $\mathfrak{h}(w0) := \mathrm{d}'$.

What remains to be shown is the fact that $\mathfrak{h}_{\leq k}$ is a homomorphism from $\mathcal{U}\restriction_{\{0,1\}^{\leq k}}$ to $\mathcal{I}$. We already know that $\mathfrak{h}_{\leq k-1}$ preserves concepts and roles, thus we can focus on concepts and roles involving words of length k. Hence, take any w of length k and proceed with concepts first. Let A be any concept name and assume that $w \in \mathrm{A}^{\mathcal{U}}$. Our goal is to show that $\mathfrak{h}(w) \in \mathrm{A}^{\mathcal{I}}$. The cases of $\mathrm{A} \in \{\mathrm{Lvl}_k, \mathrm{L}, \mathrm{R}, \mathrm{Ad}_k^b\}$ follow immediately from the construction (see the discussion while defining them). The cases of $\mathrm{A} = \mathrm{Lvl}_j$ with $j \neq k$ and $\mathrm{A} = \mathrm{Ad}_i^b$ with $i \leq k$ cannot happen by the definition of n-configuration unit. Thus the only cases left are these with $\mathrm{A} = \mathrm{Ad}_i^b$ with $i < k$. But this is easy: let $w = uv$ with $|v| = 1$. By definition of $\mathcal{U}$ we have that $u \in (\mathrm{Ad}_i^b)^{\mathcal{U}}$. Since $\mathfrak{h}_{\leq k-1}$ is a homomorphism, we infer $\mathfrak{h}(u) \in (\mathrm{Ad}_i^b)^{\mathcal{I}}$ and, by $\mathcal{I} \models$ (PropBit[i]), we conclude $\mathfrak{h}(w) \in (\mathrm{Ad}_i^b)^{\mathcal{I}}$. Now we proceed with the case of role preservation by $\mathfrak{h}$. Reasoning analogously, we may focus on roles s from $\mathbf{R}_{unit}$ and involving w only. Thus, by definition of $\mathcal{U}$, the only cases are self-loops (that follows by $\mathcal{U} \models$ (all-loops-but-next), (leaves-next-loop)) and the roles $\ell_k^{\mathcal{U}}$ and $r_k^{\mathcal{U}}$ between the parent of w and w, that follow from the construction.

This finishes the induction, implying that $\mathfrak{h}$ is indeed a homomorphism from $\mathcal{U}$ to $\mathcal{I}$ satisfying $\mathfrak{h}(\varepsilon) = \mathrm{d}$. We conclude by showing that $\mathfrak{h}$ is injective. Ad absurdum, suppose that there are $u \neq v \in \Delta^{\mathcal{U}}$ such that $\mathfrak{h}(u) = \mathfrak{h}(v)$. If $|u| \neq |v|$ we have that $\mathfrak{h}(u) \in \mathrm{Lvl}_{|u|}^{\mathcal{I}} \cap \mathrm{Lvl}_{|v|}^{\mathcal{I}}$ (by the definition of $\mathcal{U}$ and by preservation of concepts by $\mathfrak{h}$). This contradicts $\mathcal{I} \models$ (LvlDisj[i,j]). Otherwise, $|u| = |v|$ but their i-th letters differ. Again, since $\mathfrak{h}$ is a homomorphism, we conclude $\mathfrak{h}(v) \in (\mathrm{Ad}_i^0)^{\mathcal{I}} \cap (\mathrm{Ad}_i^1)^{\mathcal{I}}$, which violates $\mathcal{I} \models$ (AdDisj[i]). Hence, $\mathfrak{h}$ is injective. $\square$

At this point, we would like to give the reader some intuitions on why units are decorated with different self-loops. First, we show that their presence can be exploited to navigate top-down through a given unit.

Lemma 6.6 Let $\mathcal{U}$ be an n-configuration unit. Then for all $w \in \Delta^{\mathcal{U}}$ we have $(\varepsilon, w) \in {\ell_1}^{\mathcal{U}} \circ {r_1}^{\mathcal{U}} \circ \ldots \circ {\ell_n}^{\mathcal{U}} \circ {r_n}^{\mathcal{U}}$ with "$\circ$" denoting the composition of relations.

Proof. For simplicity we use $s_i^{\mathcal{U}}$ as an abbreviation of ${\ell_1}^{\mathcal{U}} \circ {r_1}^{\mathcal{U}} \circ \ldots \circ {\ell_i}^{\mathcal{U}} \circ {r_i}^{\mathcal{U}}$. The proof is by induction, where the assumption is that for all $1 \leq i \leq n$ we have that all words w of length at most i satisfy $(\varepsilon, w) \in s_i^{\mathcal{U}}$. The base case (for $w \in \{\varepsilon, 0, 1\}$) is immediate to verify. Now take any word w of length at most $i+1$ and consider the following two cases:

1. $|w| \leq i$. Hence, by the inductive assumption we have $(\varepsilon, w) \in s_i^{\mathcal{U}}$. Since $(w, w) \in \ell_{i+1}^{\mathcal{U}}$ and $(w, w) \in r_{i+1}^{\mathcal{U}}$, by the definition of composition we conclude $(\varepsilon, w) \in s_{i+1}^{\mathcal{U}}$.
2. $|w| = i+1$. Hence, $w = u0$ or $u1$ for some $|u| = i$. We focus on the first case, the second one is symmetric. By the inductive assumption we infer that $(\varepsilon, u) \in s_i^{\mathcal{U}}$. Since $(u, u0) \in \ell_{i+1}^{\mathcal{U}}$ and $(w, w) \in r_{i+1}^{\mathcal{U}}$ we conclude $(\varepsilon, w) \in s_{i+1}^{\mathcal{U}}$, which finishes the proof.

□

Employing Lemma 6.6 can now conclude that there is a *single* CQ detecting root-leaf pairs in units.

Corollary 6.7

Let $\mathcal{U}$ be an n-configuration unit. There is a single conjunctive query $q_{rl}(x_0, x_{2n})$ with $x_0, x_{2n} \in \mathrm{Var}(q_{rl})$

$$q_{rl} \coloneqq (\mathrm{Lvl}_0?; \ell_1; r_1; \ldots; \ell_n; r_n; \mathrm{Lvl}_n?)(x_0, x_{2n})$$

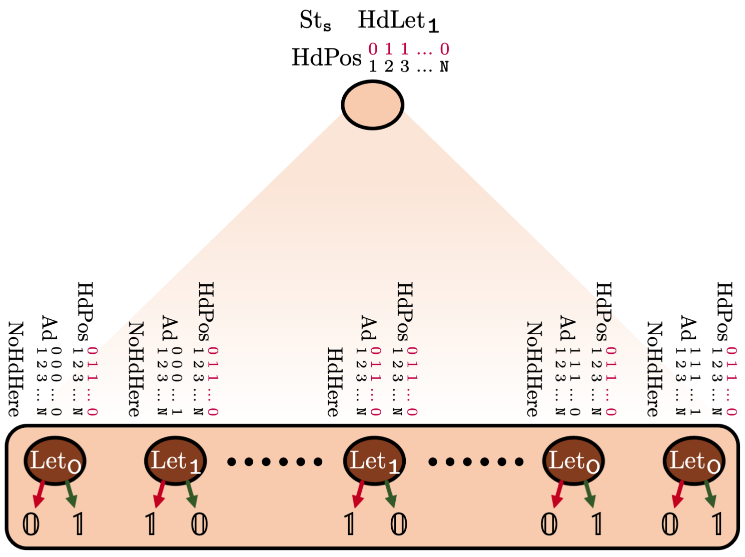
which is of size polynomial in n, such that the set $M = \{(\pi(x_0), \pi(x_{2n})) \mid \mathcal{U} \models_\pi q_{rl}\}$ is equal to the set of root-leaf pairs from $\mathcal{U}$, *i.e.* $\mathrm{Lvl}_0^{\mathcal{U}} \times \mathrm{Lvl}_n^{\mathcal{U}}$.

We encourage the reader to play with the query $q \coloneqq (\mathrm{Lvl}_0?; \ell_1; r_1; \mathrm{Lvl}_1?; \ell_2; r_2; \mathrm{Lvl}_2?)(x_0, x_4)$ and the example 2-configuration unit $\mathcal{U}$ depicted after Definition 6.3. This will make the reader more familiar with the path-syntax of CQs, provide more intuition on the key role played by self-loops in our construction, and justify that indeed any root-leaf pair can be assigned to (x_0, x_4) in an example match π witnessing $\mathcal{U} \models_\pi q$.

6.4 From Units to Configuration Trees

In the next step, we enrich $(\mathtt{N}{+}1)$-configuration units with additional concepts, allowing the units to represent a meaningful configuration of our ATM $\mathcal{M} = (\mathtt{N}, \mathtt{Q}, \mathtt{Q}_\exists, \mathtt{s}_I, \mathtt{s}_A, \mathtt{s}_R, \mathtt{T})$. To this end, we employ a variety of new concept names from $\mathbf{C}_{conf}$ consisting of

$$\mathbf{C}_{conf} \coloneqq \big\{\mathrm{HdHere}, \mathrm{NoHdHere}, \mathrm{St}_{\mathtt{s}}, \mathrm{HdPos}_i^b, \mathrm{HdLet}_{\mathtt{a}}, \mathrm{Let}_{\mathtt{a}}, \mathbb{0}, \mathbb{1} \mid \mathtt{s} \in \mathtt{Q}, b \in \{0,1\}, i \in \{1, \ldots, \mathtt{N}\}, \mathtt{a} \in \{\mathtt{0}, \mathtt{1}\}\big\}.$$



Before turning to a formal definition we first describe how configurations are structurally represented in models. Recall that a configuration of $\mathcal{M}$ is a word $\mathtt{wsw}'$ with $|\mathtt{ww}'| = 2^{\mathtt{N}}$ (called *tape*) and $\mathtt{s} \in \mathtt{Q}$. In our encoding, this configuration will be represented by an $(\mathtt{N}{+}1)$-configuration unit $\mathcal{C}$ decorated by concepts from $\mathbf{C}_{conf}$. The interpretation $\mathcal{C}$ stores the state $\mathtt{s}$, by associating the state concept $\mathrm{St}_{\mathtt{s}}$ to its root. The tape content $\mathtt{ww}'$ is represented by the internal nodes of $\mathcal{C}$: the i-th letter of $\mathtt{ww}'$ (*i.e.* the content of the ATM's i-th tape cell) is represented by the i-th node (according to their binary addresses) at the $\mathtt{N}$-th level. In case this letter is $\mathtt{0}$, the corresponding node will be assigned the concept $\mathrm{Let}_{\mathtt{0}}$, while $\mathtt{1}$ is represented by $\mathrm{Let}_{\mathtt{1}}$. Yet, for reasons that will become clear only later, the tape cells' content is additionally represented in another way: if it is $\mathtt{0}$, then we label the i-th node's left child with $\mathbb{0}$ and its right child with $\mathbb{1}$. The reverse situation is implemented when the node represents the letter $\mathtt{1}$. Finally, there is a unique tape cell that is visited by the head of $\mathcal{M}$ and the node corresponding to this cell is explicitly marked by the concept HdHere while all other "tape cell nodes" are marked by NoHdHere. In order to implement this marking correctly, the head's position's address needs to be explicitly recorded. Consequently, $\mathcal{C}$'s root node stores this address (binarily encoded using the HdPos_i^b concepts) and from there, these concept assignments are broadcast to and stored in all tape cell nodes on the $\mathtt{N}$-th level. Similarly, we decorate $\mathcal{C}$'s root with the concept $\mathrm{HdLet}_{\mathtt{a}}$ meaning that the current letter scanned by the head is $\mathtt{a}$. The structure just described can be visualised on the next page. After this informal description and depiction, the formal definition of configuration trees should be plausible.

Definition 6.8 (configuration tree) A **configuration tree** $\mathcal{C}$ of $\mathcal{M}$ is an interpretation $\mathcal{C} := (\Delta^{\mathcal{C}}, \cdot^{\mathcal{C}})$ such that $\mathcal{C}$ is an $(\mathtt{N}{+}1)$-configuration unit additionally satisfying:

- There exists a unique state $\mathtt{s} \in \mathtt{Q}$ such that $(\mathrm{St}_{\mathtt{s}})^{\mathcal{C}} = \{\varepsilon\}$ and $(\mathrm{St}_{\mathtt{s}'})^{\mathcal{C}} = \emptyset$ for all $\mathtt{s}' \neq \mathtt{s}$.
- $(\mathrm{Lvl}_{\mathtt{N}+1})^{\mathcal{C}} = \mathbb{0}^{\mathcal{C}} \cup \mathbb{1}^{\mathcal{C}}$ and $\mathbb{0}^{\mathcal{C}} \cap \mathbb{1}^{\mathcal{C}} = \emptyset$.
- $(\mathrm{Let}_{\mathtt{0}})^{\mathcal{C}} = \{w \mid w0 \in \mathbb{0}^{\mathcal{C}}, w1 \in \mathbb{1}^{\mathcal{C}}\}$, $(\mathrm{Let}_{\mathtt{1}})^{\mathcal{C}} = \{w \mid w0 \in \mathbb{1}^{\mathcal{C}}, w1 \in \mathbb{0}^{\mathcal{C}}\}$, and $(\mathrm{Let}_{\mathtt{0}})^{\mathcal{C}} \cup (\mathrm{Let}_{\mathtt{1}})^{\mathcal{C}} = \mathrm{Lvl}_{\mathtt{N}}^{\mathcal{C}}$.
- There is a *unique* word w_{head} of length $\mathtt{N}$ witnessing $\mathrm{HdHere}^{\mathcal{C}}{=}\{w_{head}\}$ and $\mathrm{NoHdHere}^{\mathcal{C}}{=}\mathrm{Lvl}_{\mathtt{N}}^{\mathcal{C}} \setminus \{w_{head}\}$.
- For $1 \leq i \leq \mathtt{N}$ and $b \in \{0,1\}$ satisfying $w_{head} \in (\mathrm{Ad}_i^b)^{\mathcal{C}}$ we put[a] $(\mathrm{HdPos}_i^b)^{\mathcal{C}} = \mathrm{Lvl}_0^{\mathcal{C}} \cup \mathrm{Lvl}_{\mathtt{N}}^{\mathcal{C}}$ and $(\mathrm{HdPos}_i^{1-b})^{\mathcal{C}} = \emptyset$.
- $\mathrm{HdLet}_{\mathtt{a}}^{\mathcal{C}} = \{\varepsilon\}$, $\mathrm{HdLet}_{\mathtt{1-a}}^{\mathcal{C}} = \emptyset$, where $\mathtt{a} \in \{\mathtt{0}, \mathtt{1}\}$ is the unique letter with $w_{head} \in \mathrm{Let}_{\mathtt{a}}^{\mathcal{C}}$.

[a]This is well-defined since for any i, we have that w_{head} belongs to exactly one of $(\mathrm{Ad}_i^0)^{\mathcal{C}}, (\mathrm{Ad}_i^1)^{\mathcal{C}}$ by the definition of a unit.

We next proceed with the corresponding axiomatisation.

1. To express that $\mathcal{C}$ is an $(\mathtt{N}{+}1)$-configuration unit we integrate all the GCIs from $\mathcal{K}_{unit}^{\mathtt{N}+1}$.
2. The root of $\mathcal{C}$ is labelled with a unique state concept.

   (StCov) $\mathrm{Lvl}_0 \equiv \bigsqcup_{\mathtt{s} \in \mathtt{Q}} \mathrm{St}_{\mathtt{s}}$

   (StDisj[s,s′]) $\mathrm{St}_{\mathtt{s}} \sqcap \mathrm{St}_{\mathtt{s}'} \sqsubseteq \bot$ (for all $\mathtt{s} \neq \mathtt{s}'$)

3. To axiomatise the coherent representation of the tape's content we employ:

   (LetDisj) $\mathbb{0} \sqcap \mathbb{1} \sqsubseteq \bot$ **(LetConDisj)** $\mathrm{Let}_{\mathtt{0}} \sqcap \mathrm{Let}_{\mathtt{1}} \sqsubseteq \bot$

   (LetCov) $\mathrm{Lvl}_{\mathtt{N}+1} \equiv \mathbb{0} \sqcup \mathbb{1}$ **(LetConCov)** $\mathrm{Let}_{\mathtt{0}} \sqcup \mathrm{Let}_{\mathtt{1}} \equiv \mathrm{Lvl}_{\mathtt{N}}$

   (EncLetZero) $\mathrm{Let}_{\mathtt{0}} \sqsubseteq \forall \ell_{\mathtt{N}+1}(\mathrm{Lvl}_{\mathtt{N}+1} \to \mathbb{0}) \sqcap \forall r_{\mathtt{N}+1}(\mathrm{Lvl}_{\mathtt{N}+1} \to \mathbb{1})$

   (EncLetOne) $\mathrm{Let}_{\mathtt{1}} \sqsubseteq \forall \ell_{\mathtt{N}+1}(\mathrm{Lvl}_{\mathtt{N}+1} \to \mathbb{1}) \sqcap \forall r_{\mathtt{N}+1}(\mathrm{Lvl}_{\mathtt{N}+1} \to \mathbb{0})$

4. Next, for the concepts $\mathrm{HdPos}_1^b, \ldots, \mathrm{HdPos}_{\mathtt{N}}^b$ we make sure they encode exactly one proper binary address (meant to encode the head's current position) in the root of $\mathcal{C}$. Below we assume $1 \leq i \leq \mathtt{N}$.

   (HdPosCov[i]) $\mathrm{Lvl}_0 \sqcup \mathrm{Lvl}_{\mathtt{N}} \equiv \mathrm{HdPos}_i^0 \sqcup \mathrm{HdPos}_i^1$

   (HdPosDisj[i]) $\mathrm{HdPos}_i^0 \sqcap \mathrm{HdPos}_i^1 \sqsubseteq \bot$

5. Another step is to propagate the head address stored in the root to all nodes on the $\mathtt{N}$-th level of $\mathcal{C}$. Here we exploit the presence of self-loops and Lemma 6.6, and use the following GCIs (for $1 \leq i \leq \mathtt{N}$ and $b \in \{0,1\}$):[5]

   (PropHdPos[i,b]) $\mathrm{Lvl}_0 \sqcap \mathrm{HdPos}_i^b \sqsubseteq \forall \ell_1 \forall r_1 \ldots \forall \ell_{\mathtt{N}} \forall r_{\mathtt{N}}\, (\mathrm{Lvl}_{\mathtt{N}} \to \mathrm{HdPos}_i^b)$

6. We distinguish between the node representing the cell visited by the head (assigning HdHere) and the other cell nodes (assigning NoHdHere) by having every cell node compare their address (stored in the Ad_i^b concepts) with the head address received through the broadcast from the root.

[5]The same can be achieved without exploitation of self-loops by iteratively propagating the HdPos_i^b through the tree, but the author believes that the presented formulation is elegant and makes the reader get used to the presence of self-loops.

(HdHereCov) $\mathrm{HdHere} \sqcup \mathrm{NoHdHere} \equiv \mathrm{Lvl}_{\mathtt{N}}$

(HdHereEqualAdr) $\mathrm{Lvl}_{\mathtt{N}} \sqcap \bigsqcap_{i=1}^{\mathtt{N}} \bigsqcup_{b\in\{0,1\}} \left(\mathrm{Ad}_i^b \sqcap \mathrm{HdPos}_i^b\right) \sqsubseteq \mathrm{HdHere}$

(NoHdHereDiffrAdr) $\mathrm{Lvl}_{\mathtt{N}} \sqcap \bigsqcup_{i=1}^{\mathtt{N}} \bigsqcup_{b\in\{0,1\}} \left(\mathrm{Ad}_i^b \sqcap \mathrm{HdPos}_i^{1-b}\right) \sqsubseteq \mathrm{NoHdHere}$

7. We synchronise the letter scanned by the head with its "recording" in the root (below $\mathtt{a} \in \{\mathtt{0}, \mathtt{1}\}$).

(HdLetCov) $\mathrm{HdLet}_{\mathtt{0}} \sqcup \mathrm{HdLet}_{\mathtt{1}} \equiv \mathrm{Lvl}_0$

(RetrHdLet[a]) $\mathrm{Lvl}_0 \sqcap \exists\ell_1\exists r_1 \ldots \exists\ell_{\mathtt{N}}\exists r_{\mathtt{N}}(\mathrm{HdHere} \sqcap \mathrm{Let}_{\mathtt{a}}) \sqsubseteq \mathrm{HdLet}_{\mathtt{a}}$

(HdLetUnique[a]) $\mathrm{Lvl}_0 \sqcap \mathrm{HdLet}_{\mathtt{a}} \sqsubseteq \forall\ell_1\forall r_1 \ldots \forall\ell_{\mathtt{N}}\forall r_{\mathtt{N}}(\mathrm{HdHere} \to \mathrm{Let}_{\mathtt{a}})$

This finishes the axiomatisation of configuration trees. For the knowledge base $\mathcal{K}_{conf}$, composed of all presented GCIs, we present its correctness in the following lemmas. Similarly to the previous section, both of them are routine and the reader may omit them at first reading.

Lemma 6.9 Any configuration tree $\mathcal{C}$ is a model of $\mathcal{K}_{conf}$.

Proof. Let $\mathcal{C}$ be a configuration tree. We proceed with all axioms α of $\mathcal{K}_{conf}$ showing $\mathcal{C} \models \alpha$.

1. Since $\mathcal{C}$ is an $(\mathtt{N}{+}1)$-configuration unit by definition, by Lemma 6.4 we infer $\mathcal{C} \models \mathcal{K}_{unit}^{\mathtt{N}+1}$.
2. The satisfactions $\mathcal{C} \models$ (StCov) and $\mathcal{C} \models$ (StDisj[s,s′]) follow immediately from the first item of Definition 6.8.
3. We have $\mathcal{C} \models$ (LetDisj) and $\mathcal{C} \models$ (LetCov) by the second item of Definition 6.8. Next, we have $\mathcal{C} \models$ (LetConDisj) by the fact that words in $\mathrm{Let}_{\mathtt{0}}^{\mathcal{C}}$ and $\mathrm{Let}_{\mathtt{1}}^{\mathcal{C}}$ have different last letters. The satisfaction of (LetConCov) by $\mathcal{C}$ is due to the 3rd property in the 3rd item of Definition 6.8. What remains to show is the satisfaction of (EncLetZero) (the proof of (EncLetOne) is symmetric), but this is due to the fact that the only $\ell_{\mathtt{N}+1}$- (resp. $r_{\mathtt{N}+1}$-) successor of any $w \in \mathrm{Let}_{\mathtt{0}}^{\mathcal{C}}$ (thus also from $\mathrm{Lvl}_{\mathtt{N}}^{\mathcal{C}}$ by the previous statement) is $w0$ (resp. $w1$) that belong to $\mathbb{0}^{\mathcal{C}}$ (resp. $\mathbb{1}^{\mathcal{C}}$) by definition.
4. Before we proceed with the next part of the axiomatisation, we establish a few properties about concept memberships of w_{head}. First, we have $w_{head} \in \mathrm{Lvl}_{\mathtt{N}}^{\mathcal{C}}$ by definition (4th item). Moreover, for every i we have that w_{head} belongs to exactly one of $(\mathrm{Ad}_i^0)^{\mathcal{C}}, (\mathrm{Ad}_i^1)^{\mathcal{C}}$ by the definition of a unit. Hence, the choice of w_{head} fixes interpretations of $\mathrm{Ad}_i^0, \mathrm{Ad}_i^1$ and, as we will see, also HdPos_i^0 and HdPos_i^1. Indeed, by the 5th item of Definition 6.8, whenever $w_{head} \in (\mathrm{Ad}_i^b)^{\mathcal{C}}$ then $(\mathrm{HdPos}_i^b)^{\mathcal{C}} = \mathrm{Lvl}_0^{\mathcal{C}} \cup \mathrm{Lvl}_{\mathtt{N}}^{\mathcal{C}}$ and $(\mathrm{HdPos}_i^{1-b})^{\mathcal{C}} = \emptyset$, thus also $(\mathrm{HdPos}_i^b)^{\mathcal{C}} \cap (\mathrm{HdPos}_i^{1-b})^{\mathcal{C}} = \emptyset$ and $(\mathrm{HdPos}_i^b)^{\mathcal{C}} \cup (\mathrm{HdPos}_i^{1-b})^{\mathcal{C}} = \mathrm{Lvl}_0^{\mathcal{C}} \cup \mathrm{Lvl}_{\mathtt{N}}^{\mathcal{C}}$ hold. This establishes $\mathcal{C} \models$ (HdPosDisj[i]) and $\mathcal{C} \models$ (HdPosCov[i]) for all $1 \leq i \leq \mathtt{N}$.
5. If $\varepsilon \in (\mathrm{Lvl}_0 \sqcap \mathrm{HdPos}_i^b)^{\mathcal{C}}$, then by definition $w_{head} \in (\mathrm{Ad}_i^b)^{\mathcal{C}}$. This implies $(\mathrm{HdPos}_i^b)^{\mathcal{C}} = \mathrm{Lvl}_0^{\mathcal{C}} \cup \mathrm{Lvl}_{\mathtt{N}}^{\mathcal{C}}$ and thus $\mathrm{Lvl}_{\mathtt{N}}^{\mathcal{C}} \subseteq (\mathrm{HdPos}_i^b)^{\mathcal{C}}$, which is even stronger than the meaning of the GCI (PropHdPos[i,b]). Hence, $\mathcal{C} \models$ (PropHdPos[i,b]) for all $1 \leq i \leq \mathtt{N}$ and $0 \leq b \leq 1$.
6. We next focus on proving $\mathcal{C} \models$ (HdHereEqualAdr) (the proof of (NoHdHereDiffrAdr) is symmetric) and $\mathcal{C} \models$ (HdHereCov). The second GCI follows by definition, hence we focus on the first one. Ad absurdum, assume that there is $w \in (\mathrm{Lvl}_{\mathtt{N}} \sqcap \bigsqcap_{i=1}^{\mathtt{N}} \bigsqcup_{b\in\{0,1\}} \left(\mathrm{Ad}_i^b \sqcap \mathrm{HdPos}_i^b\right))^{\mathcal{C}}$ but $w \not\in (\mathrm{HdHere})^{\mathcal{C}}$. Since $w \in (\mathrm{Lvl}_{\mathtt{N}})^{\mathcal{C}}$ we infer $|w| = \mathtt{N}$ and, by $w \not\in (\mathrm{HdHere})^{\mathcal{C}}$ and the 4th item of Definition 6.8, we infer $w \neq w_{head}$. Thus there is a position $1 \leq k \leq \mathtt{N}$ such the k-th letter of w differs from the k-th letter of w_{head} (called it b). So we have $w \not\in (\mathrm{Ad}_k^b)^{\mathcal{C}}$ and $w \in (\mathrm{Ad}_k^{1-b})^{\mathcal{C}}$ (by definition of Ad_i^b in units). At the same time the 5th item of Definition 6.8 informs us that $(\mathrm{HdPos}_k^b)^{\mathcal{C}} = \mathrm{Lvl}_0^{\mathcal{C}} \cup \mathrm{Lvl}_{\mathtt{N}}^{\mathcal{C}}$ and $(\mathrm{HdPos}_k^{1-b})^{\mathcal{C}} = \emptyset$, which implies $w \in (\mathrm{HdPos}_k^b)^{\mathcal{C}}$ and $w \not\in (\mathrm{HdPos}_k^{1-b})^{\mathcal{C}}$. This contradicts the fact that $w \in (\bigsqcup_{b\in\{0,1\}} \left(\mathrm{Ad}_i^b \sqcap \mathrm{HdPos}_i^b\right))^{\mathcal{C}}$.
7. We proceed with the last three GCIs. Satisfaction of (HdLetCov) by $\mathcal{C}$ follows by definition. For (RetrHdLet[a]), assume that its antecedent is non-empty (which means that it is equal to $\{\varepsilon\}$). This implies, by the 4th and the last item of Definition 6.8, that $w_{head} \in \mathrm{Let}_{\mathtt{a}}^{\mathcal{C}}$. Thus, by definition, $\mathrm{HdLet}_{\mathtt{a}}^{\mathcal{C}}$ is equal to $\{\varepsilon\}$, which obviously contains ε. Hence, $\mathcal{C} \models$ (RetrHdLet[a]). Finally, $\mathcal{C} \models$ (HdLetUnique[a]) is shown as follows. If the antecedent of (HdLetUnique[a]) is non-empty then it is equal to $\{\varepsilon\}$. Thus by the list

item of Definition 6.8, we have $\mathrm{Let}^{\mathcal{C}}_{\mathtt{a}} = \{w_{head}\}$. Since $\mathrm{HdHere}^{\mathcal{C}} = \{w_{head}\}$ (by the 4th item of Definition 6.8), we conclude $\mathrm{HdHere}^{\mathcal{C}} \subseteq \mathrm{Let}^{\mathcal{C}}_{\mathtt{a}}$. Therefore $\mathcal{C} \models$ (HdLetUnique[a]). □

Lemma 6.10 For any model $\mathcal{I}$ of $\mathcal{K}_{conf}$ and any $\mathrm{d} \in \mathrm{Lvl}^{\mathcal{I}}_{0}$ there is a configuration tree $\mathcal{C}$ and a homomorphism $\mathfrak{h}$ from $\mathcal{C}$ into $\mathcal{I}$ with $\mathfrak{h}(\varepsilon) = \mathrm{d}$.

Proof. By Lemma 6.5 there is an $(\mathtt{N}{+}1)$-configuration unit $\mathcal{U}$ and a homomorphism $\mathfrak{h} : \mathcal{U} \to \mathcal{I}$ with $\mathfrak{h}(\varepsilon) = \mathrm{d}$. Moreover, as the symbols outside $\mathbf{R}_{unit} \cup \mathbf{C}_{unit}$ do not appear in Definition 6.3 we can assume that $\mathcal{U}$ interprets them as empty sets. Let $\mathcal{C} = (\Delta^{\mathcal{U}}, \cdot^{\mathcal{C}})$ be an interpretation that is obtained from changing the meaning of concepts from $\mathbf{C}_{conf}$ as follows: for any $\mathrm{C} \in \mathbf{C}_{conf}$ we let $\mathrm{C}^{\mathcal{C}} := \{w \mid \mathfrak{h}(w) \in \mathrm{C}^{\mathcal{U}}\}$. All other symbols are interpreted as in $\mathcal{U}$. Clearly $\mathfrak{h}$ is a homomorphism from $\mathcal{C}$ into $\mathcal{I}$ with $\mathfrak{h}(\varepsilon) = \mathrm{d}$. It suffices to prove that $\mathcal{C}$ is a configuration tree. This is done by routine investigation of items from Definition 6.8 and the presented GCIs.

- Let $\mathtt{s}$ be the unique state satisfying $\mathfrak{h}(\varepsilon) \in \mathrm{St}^{\mathcal{I}}_{\mathtt{s}}$: it exists by $\mathcal{I} \models$ (StCov) and is unique by $\mathcal{I} \models$ (StDisj[s,s′]). Hence, $(\mathrm{St}_{\mathtt{s}})^{\mathcal{C}} = \{\varepsilon\}$ holds. Moreover, $(\mathrm{St}_{\mathtt{s}'})^{\mathcal{C}} = \emptyset$ for all $\mathtt{s}' \neq \mathtt{Q}$. Note that $\varepsilon \notin (\mathrm{St}_{\mathtt{s}'})^{\mathcal{C}}$ by (StDisj[s,s′]) and for other elements (so from $\mathrm{Lvl}^{\mathcal{C}}_{i}$ for some $i > 0$) their membership in $(\mathrm{St}_{\mathtt{s}'})^{\mathcal{C}}$ would violate $\mathcal{I} \models \bigsqcup_{\mathtt{s} \in \mathtt{Q}} \mathrm{St}_{\mathtt{s}}$.
- Similarly, the equalities $(\mathrm{Lvl}_{\mathtt{N}+1})^{\mathcal{C}} = \mathbb{0}^{\mathcal{C}} \cup \mathbb{1}^{\mathcal{C}}$ and $\mathbb{0}^{\mathcal{C}} \cap \mathbb{1}^{\mathcal{C}} = \emptyset$ follow by $\mathcal{I} \models$ (LetDisj) and $\mathcal{I} \models$ (LetCov).
- Analogously, we have $(\mathrm{Let}_{\mathtt{0}})^{\mathcal{C}} \cup (\mathrm{Let}_{\mathtt{1}})^{\mathcal{C}} = \mathrm{Lvl}^{\mathcal{C}}_{\mathtt{N}}$ by $\mathcal{I} \models$ (LetConDisj) and $\mathcal{I} \models$ (LetCon-Cov). Note that this implies $(\mathrm{Let}_{\mathtt{0}})^{\mathcal{C}} \subseteq \mathrm{Lvl}^{\mathcal{C}}_{\mathtt{N}}$. We next show the equality $(\heartsuit) : (\mathrm{Let}_{\mathtt{0}})^{\mathcal{C}} = \{w \in \Delta^{\mathcal{U}} \mid w0 \in \mathbb{0}^{\mathcal{C}}, w1 \in \mathbb{1}^{\mathcal{C}}\}$ (the related equality for $\mathrm{Let}_{\mathtt{1}}$ is symmetric). Take any $w \in (\mathrm{Let}_{\mathtt{0}})^{\mathcal{C}}$ (thus also $\in \mathrm{Lvl}^{\mathcal{C}}_{\mathtt{N}}$). By the fact that $\mathcal{C}$ is a unit, we infer that $w0 \in \mathrm{Lvl}^{\mathcal{C}}_{\mathtt{N}+1}$ and $w1 \in \mathrm{Lvl}^{\mathcal{C}}_{\mathtt{N}+1}$ exist, and moreover w is linked to them, respectively, by the roles $\ell^{\mathcal{C}}_{\mathtt{N}+1}$ and $r^{\mathcal{C}}_{\mathtt{N}+1}$. Hence, by the homomorphic assignment of concepts from $\mathbf{C}_{conf}$ and the satisfaction $\mathcal{I} \models$ (EncLetZero) we have that $w0 \in \mathbb{0}^{\mathcal{C}}$ and $w1 \in \mathbb{1}^{\mathcal{C}}$. Hence, the $\subseteq$-relationship of $(\heartsuit)$ follows. For the $\supseteq$-relationship take any $w \in \Delta^{\mathcal{U}}$ s.t. $w0 \in \mathbb{0}^{\mathcal{C}}, w1 \in \mathbb{1}^{\mathcal{C}}$ and note that $w \in \mathrm{Lvl}^{\mathcal{C}}_{\mathtt{N}}$ and $wb \in \mathrm{Lvl}^{\mathcal{C}}_{\mathtt{N}+1}$ hold. Otherwise, by the fact that $\mathcal{C}$ is an $(\mathtt{N}{+}1)$-configuration unit, the element $w0$ does not exist or it belongs to $\mathrm{Lvl}^{\mathcal{C}}_{i}$ for $i \neq \mathtt{N}{+}1$, which violates $\mathcal{I} \models$ (LetCov). By $\mathcal{I} \models$ (LetConCov) we know that $w \in (\mathrm{Let}_{\mathtt{0}})^{\mathcal{C}} \cup (\mathrm{Let}_{\mathtt{1}})^{\mathcal{C}}$. If $w \in (\mathrm{Let}_{\mathtt{0}})^{\mathcal{C}}$ holds then we are done. Thus, assume towards a contradiction that $w \in (\mathrm{Let}_{\mathtt{1}})^{\mathcal{C}}$. But then the first conjunct of the consequent of (EncLetOne) is violated, contradicting its satisfaction by $\mathcal{I}$. Hence $(\heartsuit)$ holds.
- Let w_{head} be the unique $\mathtt{N}$-digit binary word whose i-th letter is equal to b iff $\varepsilon \in (\mathrm{HdPos}^{b}_{i})^{\mathcal{C}}$ holds. This is well defined due to $\mathcal{I} \models$ (HdPosDisj[i]) and $\mathcal{I} \models$ (HdPosCov[i]) for all $1 \leq i \leq \mathtt{N}$. It remains to show that this w_{head} indeed plays the role of w_{head} in the sense of Definition 6.8. Take any $1 \leq i \leq \mathtt{N}$ and let b be the i-th letter of w_{head}. We will show that $(\mathrm{HdPos}^{1-b}_{i})^{\mathcal{C}} = \emptyset$ holds. Ad absurdum, assume that it is non-empty and contains w. By $\mathcal{I} \models$ (HdPosCov[i]) we have that either $w \in \mathrm{Lvl}^{\mathcal{C}}_{0}$ or $w \in \mathrm{Lvl}^{\mathcal{C}}_{\mathtt{N}}$. The first case is not possible due to $\mathcal{I} \models$ (HdPosDisj[i]) (we already have that $\varepsilon \in (\mathrm{HdPos}^{b}_{i})^{\mathcal{C}}$). Thus $w \in \mathrm{Lvl}^{\mathcal{C}}_{\mathtt{N}}$. But then it violates $\mathcal{I} \models$ (PropHdPos[i,b]), which by Lemma 6.6 enforces that $w \in (\mathrm{HdPos}^{b}_{i})^{\mathcal{C}}$. Hence, ($\clubsuit$:) $(\mathrm{HdPos}^{1-b}_{i})^{\mathcal{C}} = \emptyset$, which by (HdPosCov[i]) implies ($\clubsuit'$:) $\mathrm{Lvl}^{\mathcal{C}}_{0} \cup \mathrm{Lvl}^{\mathcal{C}}_{\mathtt{N}} = (\mathrm{HdPos}^{b}_{i})^{\mathcal{C}}$. Thus it follows, by definition of w_{head}, that $w_{head} \in (\mathrm{Ad}^{b}_{i})^{\mathcal{C}}$ iff $w_{head} \in (\mathrm{HdPos}^{b}_{i})^{\mathcal{C}}$, concluding that $\mathcal{C}$ satisfies the 5th item of Definition 6.8. Next, by exploiting ($\clubsuit$) and ($\clubsuit'$) and applying the satisfaction of (HdHereEqualAdr), (NoHdHereDiffrAdr) and (HdHereCov) by $\mathcal{C}$, we conclude the satisfaction of the 4th item of Definition 6.8. This, among the other properties, results in $\mathrm{HdHere}^{\mathcal{C}} = \{w_{head}\}$. Finally, let $\mathtt{a}$ be the unique letter satisfying $w_{head} \in \mathrm{Let}^{\mathcal{C}}_{\mathtt{a}}$ (it exists and is unique by $\mathcal{I} \models$ (LetConDisj), (LetConCov)). We claim that $\mathrm{HdLet}^{\mathcal{C}}_{\mathtt{a}} = \{\varepsilon\}$ and $\mathrm{HdLet}^{\mathcal{C}}_{\mathtt{1}-\mathtt{a}} = \emptyset$. Note that both $\mathrm{HdLet}^{\mathcal{C}}_{\mathtt{a}}$ and $\mathrm{HdLet}^{\mathcal{C}}_{\mathtt{1}-\mathtt{a}}$ are subsets of $\{\varepsilon\}$ by $\mathcal{I} \models$ (HdLetCov), thus it suffices to show

that $\varepsilon \in \mathrm{HdLet}^{\mathcal{C}}_{\mathtt{a}}$ and $\varepsilon \notin \mathrm{HdLet}^{\mathcal{C}}_{\mathtt{1-a}}$. The first property holds due to $\mathcal{I} \models$ (RetrHdLet[a]). For the second property, towards a contradiction assume that $\varepsilon \in \mathrm{HdLet}^{\mathcal{C}}_{\mathtt{1-a}}$. Hence, by $\mathcal{I} \models$ (HdLetUnique[a]) we conclude that $\mathrm{HdHere}^{\mathcal{C}} = \{w_{head}\} \subseteq \mathrm{Let}^{\mathcal{C}}_{\mathtt{1-a}}$. But $\mathrm{Let}^{\mathcal{C}}_{\mathtt{1-a}}$ is empty by $\mathcal{I} \models$ (LetConDisj) and the definition of $\mathtt{a}$. A contradiction.

Hence the interpretation $\mathcal{C}$ is indeed a configuration tree, concluding the proof. □

6.5 Enriching Configuration Trees

Recall that the purpose of configuration trees is to place them inside a model that describes the run of the Turing machine $\mathcal{M}$. In particular, this will require to describe the progression of one configuration to another. In order to prepare for that, we next introduce an extended version of configuration trees that are enriched by additional information pertaining to their predecessor configuration in a run. To this end, we use new concept names from

$$\mathbf{C}_{enr} := \{\mathrm{PTrns}_{\mathtt{t}}, \mathrm{Ini}, \mathrm{PHdHere}, \mathrm{NoPHdHere}, \mathrm{PHdAbv}, \mathrm{NoPHdAbv}, \mathrm{PHdPos}^{b}_{i}, \mathrm{PHdLet}_{\mathtt{a}}\},$$

with $\mathtt{t} \in \mathtt{T}$, $1 \leq i \leq \mathtt{N}$, $b \in \{0,1\}$, and $\mathtt{a} \in \{\mathtt{0},\mathtt{1}\}$. We use $\mathbf{C}_{ptr}$ to denote $\{\mathrm{Ini}, \mathrm{PTrns}_{\mathtt{t}} \mid \mathtt{t} \in \mathtt{T}\}$.

The concept $\mathrm{PTrns}_{\mathtt{t}}$, assigned to the root, indicates the transition, through which the configuration has been reached from the previous configuration, while Ini is used as its replacement for the initial configuration. In addition, concepts PHdPos^{b}_{i} and $\mathrm{PHdLet}_{\mathtt{a}}$ are attached to the root in order to — in a way very similar to HdPos^{b}_{i} and $\mathrm{HdLet}_{\mathtt{a}}$ — indicate the previous configuration's head position as well as the letter stored in that position *on the current configuration's tape*. For the sake of our encoding we also employ the concepts PHdHere, NoPHdHere that play the role analogous to the HdHere and NoHdHere concepts from configuration-trees.[6] For technical reasons, we also introduce the concepts PHdAbv and NoPHdAbv that will label nodes on the ($\mathtt{N}$+1)-th level if and only if their parent is labelled with the corresponding concept from $\{\mathrm{PHdHere}, \mathrm{NoPHdHere}\}$.

We proceed with the formal definition of the structures just described.

Definition 6.11 (enriched configuration tree) An **enriched configuration tree** $\mathcal{E}$ of $\mathcal{M}$ is an interpretation $\mathcal{E} := (\Delta^{\mathcal{E}}, \cdot^{\mathcal{E}})$ such that $\mathcal{E}$ is a configuration tree additionally satisfying the following conditions on concepts from $\mathbf{C}_{enr}$:

- There is exactly one concept $\mathrm{C} \in \mathbf{C}_{ptr}$ for which $\mathrm{C}^{\mathcal{E}} = \{\varepsilon\}$ and for all $\mathrm{C}' \in \mathbf{C}_{ptr} \setminus \{\mathrm{C}\}$ we have $(\mathrm{C}')^{\mathcal{E}} = \emptyset$.
- $\mathrm{PTrns}^{\mathcal{E}}_{(\mathtt{s},\mathtt{a},\mathtt{b},\mathtt{s}',d)} = \{\varepsilon\}$ implies $(\mathrm{St}_{\mathtt{s}'})^{\mathcal{E}} = \{\varepsilon\}$ for all transitions $(\mathtt{s},\mathtt{a},\mathtt{b},\mathtt{s}',d) \in \mathtt{T}$.
- $\mathrm{PHdHere}^{\mathcal{E}} = \{w_{phd}\}$ and $\mathrm{NoPHdHere}^{\mathcal{E}} = \mathrm{Lvl}^{\mathcal{E}}_{\mathtt{N}} \setminus \{w_{phd}\}$ for the $\mathtt{N}$-digit binary word w_{phd} encoding[a]
  - the number obtained as $w_{head} - d$ (see: Def. 6.8) whenever $\mathrm{PTrns}^{\mathcal{E}}_{(\mathtt{s},\mathtt{a},\mathtt{b},\mathtt{s}',d)} = \{\varepsilon\}$, or
  - the number 0 in case $\mathrm{Ini}^{\mathcal{E}} = \{\varepsilon\}$.
- $\mathrm{PHdAbv}^{\mathcal{E}} = \{w0, w1 \mid w \in \mathrm{PHdHere}^{\mathcal{E}}\}$ and $\mathrm{NoPHdAbv}^{\mathcal{E}} = \mathrm{Lvl}^{\mathcal{E}}_{\mathtt{N}+1} \setminus \mathrm{PHdAbv}^{\mathcal{E}}$.
- $(\mathrm{PHdPos}^{b}_{i})^{\mathcal{E}} = \mathrm{Lvl}^{\mathcal{E}}_{0} \cup \mathrm{Lvl}^{\mathcal{E}}_{\mathtt{N}}$ and $(\mathrm{PHdPos}^{1-b}_{i})^{\mathcal{E}} = \emptyset$ for all $1 \leq i \leq \mathtt{N}$ and $0 \leq b \leq 1$ with $w_{phd} \in (\mathrm{Ad}^{b}_{i})^{\mathcal{E}}$.
- $\mathrm{PHdLet}^{\mathcal{E}}_{\mathtt{a}} = \{\varepsilon\}$ and $\mathrm{PHdLet}^{\mathcal{E}}_{\mathtt{1-a}} = \emptyset$, where $\mathtt{a}$ is the unique letter from $\{\mathtt{0},\mathtt{1}\}$ such that $w_{phd} \in \mathrm{Let}^{\mathcal{E}}_{\mathtt{a}}$.
- $\mathrm{Ini}^{\mathcal{E}} = \{\varepsilon\}$ implies $\varepsilon \in \mathrm{L}^{\mathcal{E}}$, $\mathrm{St}^{\mathcal{E}}_{\mathtt{s}_I} = \{\varepsilon\}$, $\mathrm{Let}^{\mathcal{E}}_{\mathtt{0}} = \mathrm{Lvl}^{\mathcal{E}}_{\mathtt{N}}$, and $\mathrm{HdPos}^{0}_{i} = \mathrm{PHdPos}^{0}_{i} = \mathrm{Lvl}^{\mathcal{E}}_{0} \cup \mathrm{Lvl}^{\mathcal{E}}_{\mathtt{N}}$ for all $1 \leq i \leq \mathtt{N}$.

[a] Here we exploit that $\mathcal{M}$ never attempts to move left (resp. right) on the left-most (resp. right-most) tape cell.

As usual, we supplement the above definition with the corresponding axiomatisation.

[6] For simplicity, the initial configuration will also carry previous head information, but it will be irrelevant.

1. We ensure that the root unambiguously indicates the previous transition (or initiality). Below $\mathtt{t} \neq \mathtt{t}' \in \mathtt{T}$.
$$\textbf{(TrCov)}\ \mathrm{Lvl}_0 \equiv \mathrm{Ini} \sqcup \textstyle\bigsqcup_{\mathtt{t}\in\mathtt{T}} \mathrm{PTrns}_{\mathtt{t}},$$
$\textbf{(TrInitDisj}[\mathtt{t}]\textbf{)}$ $\mathrm{Ini} \sqcap \mathrm{PTrns}_{\mathtt{t}} \sqsubseteq \bot$, $\textbf{(TrDisj}[\mathtt{t},\mathtt{t}']\textbf{)}$ $\mathrm{PTrns}_{\mathtt{t}} \sqcap \mathrm{PTrns}_{\mathtt{t}'} \sqsubseteq \bot$.
2. We provide the encoding of the previous head position and the previous letter scanned by the head. This is done by means of the PHdPos_i^b, $\mathrm{PHdLet}_{\mathtt{a}}$, PHdHere, and NoPHdHere concepts in analogy to how it was done for the current head position (see the last four points of the axiomatisation from the previous section). Below we assume $1 \leq i \leq \mathtt{N}$, $b \in \{0,1\}$, and $\mathtt{a} \in \{\mathtt{0},\mathtt{1}\}$.
$\textbf{(PHdPosCov[i])}$ $\mathrm{Lvl}_0 \sqcup \mathrm{Lvl}_{\mathtt{N}} \equiv \mathrm{PHdPos}_i^0 \sqcup \mathrm{PHdPos}_i^1$,
$\textbf{(PHdPosDisj[i])}$ $\mathrm{PHdPos}_i^0 \sqcap \mathrm{PHdPos}_i^1 \sqsubseteq \bot$,
$\textbf{(PropPHdPos[i,b])}$ $\mathrm{Lvl}_0 \sqcap \mathrm{PHdPos}_i^b \sqsubseteq \forall \ell_1 \forall r_1 \ldots \forall \ell_{\mathtt{N}} \forall r_{\mathtt{N}}\,(\mathrm{Lvl}_{\mathtt{N}} \to \mathrm{PHdPos}_i^b)$,
$\textbf{(PHdHereCov)}$ $\mathrm{PHdHere} \sqcup \mathrm{NoPHdHere} \equiv \mathrm{Lvl}_{\mathtt{N}}$
$\textbf{(PHdHereEqualAdr)}$ $\mathrm{Lvl}_{\mathtt{N}} \sqcap \bigsqcap_{i=1}^{\mathtt{N}} \bigsqcup_{b\in\{0,1\}} \left(\mathrm{Ad}_i^b \sqcap \mathrm{PHdPos}_i^b\right) \sqsubseteq \mathrm{PHdHere}$,
$\textbf{(NoPHdHereDiffAdr)}$ $\mathrm{Lvl}_{\mathtt{N}} \sqcap \bigsqcup_{i=1}^{\mathtt{N}} \bigsqcup_{b\in\{0,1\}} \left(\mathrm{Ad}_i^b \sqcap \mathrm{PHdPos}_i^{1-b}\right) \sqsubseteq \mathrm{NoPHdHere}$,
$\textbf{(PHdLetCov)}$ $\mathrm{PHdLet}_{\mathtt{0}} \sqcup \mathrm{PHdLet}_{\mathtt{1}} \equiv \mathrm{Lvl}_0$,
$\textbf{(RetrPHdLet}[\mathtt{a}]\textbf{)}$ $\mathrm{Lvl}_0 \sqcap \exists \ell_1 \exists r_1 \ldots \exists \ell_{\mathtt{N}} \exists r_{\mathtt{N}} (\mathrm{PHdHere} \sqcap \mathrm{Let}_{\mathtt{a}}) \sqsubseteq \mathrm{PHdLet}_{\mathtt{a}}$,
$\textbf{(PHdLetUnique}[\mathtt{a}]\textbf{)}$ $\mathrm{Lvl}_0 \sqcap \mathrm{PHdLet}_{\mathtt{a}} \sqsubseteq \forall \ell_1 \forall r_1 \ldots \forall \ell_{\mathtt{N}} \forall r_{\mathtt{N}} (\mathrm{PHdHere} \to \mathrm{Let}_{\mathtt{a}})$.
3. Next, the concepts PHdAbv and NoPHdAbv are assigned via
$\textbf{(PHdAbvCov)}$ $\mathrm{PHdAbv} \sqcup \mathrm{NoPHdAbv} \equiv \mathrm{Lvl}_{\mathtt{N}+1}$,
$\textbf{(PHdAbvDisj)}$ $\mathrm{PHdAbv} \sqcap \mathrm{NoPHdAbv} \sqsubseteq \bot$,
$\textbf{(PropPHdAbv)}$ $\mathrm{PHdHere} \sqsubseteq \forall \ell_{\mathtt{N}+1} \forall r_{\mathtt{N}+1} \mathrm{Lvl}_{\mathtt{N}+1} \to \mathrm{PHdAbv}$,
$\textbf{(PropNoPHdAbv)}$ $\mathrm{NoPHdHere} \sqsubseteq \forall \ell_{\mathtt{N}+1} \forall r_{\mathtt{N}+1} \mathrm{Lvl}_{\mathtt{N}+1} \to \mathrm{NoPHdAbv}$.
4. We ensure consistency of the current configuration with the previous transition. Below we assume that $(\mathtt{s},\mathtt{a},\mathtt{b},\mathtt{s}',d) \in \mathtt{T}$.
$$\textbf{(TransiCons)}\ \mathrm{PTrns}_{(\mathtt{s},\mathtt{a},\mathtt{b},\mathtt{s}',d)} \sqsubseteq \mathrm{PHdLet}_{\mathtt{b}} \sqcap \mathrm{St}_{\mathtt{s}'} \sqcap \text{“PHdPos} + d = \text{HdPos”},$$
where the last right-hand-side expression, specifying decrements or increments of binary encodings of numbers, is implemented in a usual way [BHLS17, p. 127] via:
$$\bigsqcup_{i=1}^{\mathtt{N}} \Big(\mathrm{A}_i^0 \sqcap \mathrm{B}_i^1 \sqcap \bigsqcap_{j=1}^{i-1} (\mathrm{A}_j^1 \sqcap \mathrm{B}_j^0) \sqcap \bigsqcap_{j=i+1}^{\mathtt{N}} \big((\mathrm{A}_j^1 \sqcap \mathrm{B}_j^1) \sqcup (\mathrm{A}_j^0 \sqcap \mathrm{B}_j^0)\big)\Big)$$
with $\mathrm{A} := \mathrm{PHdPos}$ and $\mathrm{B} := \mathrm{HdPos}$ if $d = +1$, and with A and B swapped if $d = -1$.
5. We encode the initial configuration as follows.
$$\textbf{(IC)}\ \mathrm{Ini} \sqsubseteq \mathrm{Lvl}_0 \sqcap \mathrm{L} \sqcap \mathrm{St}_{\mathtt{s}_I} \sqcap \textstyle\bigsqcap_{i=1}^{\mathtt{N}} (\mathrm{HdPos}_i^0 \sqcap \mathrm{PHdPos}_i^0) \sqcap \forall \ell_1 \forall r_1 \ldots \forall \ell_{\mathtt{N}} \forall r_{\mathtt{N}} (\mathrm{Lvl}_{\mathtt{N}} \to \mathrm{Let}_{\mathtt{0}}),$$

For the KB $\mathcal{K}_{enr}$, composed of all the GCIs presented so far, we show correctness in the following lemmas. The proofs are routine and similar to proofs from the previous section.

Lemma 6.12 Any enriched configuration tree of $\mathcal{E}$ is a model of $\mathcal{K}_{enr}$.

Proof. Since $\mathcal{E}$ is a configuration tree by definition, by Lemma 6.9 we infer $\mathcal{E} \models \mathcal{K}_{conf}$. Hence, we may focus on the GCIs presented in this section only. For the GCIs from the 2nd group, we essentially use the same proof that we used for their “non-previous” counterparts the proof of Lemma 6.9 and thus we do not repeat it here. Satisfaction of (PHdAbvCov) and (PHd-AbvDisj) follows by the 4th item of Definition 6.11. Next, to prove that also (PropPHdAbv) is satisfied by $\mathcal{E}$ (the proof for (PropNoPHdAbv) is analogous) we take any $w \in \mathrm{PHdHere}^{\mathcal{E}}$, which by definition is equal to w_{phd} and see that the antecedent of the implication on the right hand of (PropPHdAbv) is satisfied only by $w_{phd}0$ and $w_{phd}1$, which are in $\mathrm{PHdAbv}^{\mathcal{E}}$ by definition. Next, to show satisfaction of (TransiCons) we assume $\varepsilon \in \mathrm{PTrns}^{\mathcal{E}}_{(\mathtt{s},\mathtt{a},\mathtt{b},\mathtt{s}',d)}$. Then we have $\varepsilon \in \mathrm{PHdLet}_{\mathtt{b}}^{\mathcal{E}}$ (by the second to last items of Definition 6.11), $\varepsilon \in \mathrm{St}_{\mathtt{s}'}^{\mathcal{E}}$ (by the 2nd item of Definition 6.11) and that $\varepsilon \in$ "PHdPos $+ d =$ HdPos"$^{\mathcal{E}}$ (by correctness of incrementation/decrementation of binary encodings and by the 1st subitem of the 3rd item of Definition 6.11). Thus $\mathcal{E} \models$ (TransiCons). Finally, $\mathcal{E} \models$ (IC) follows directly from the last item of Definition 6.11. □

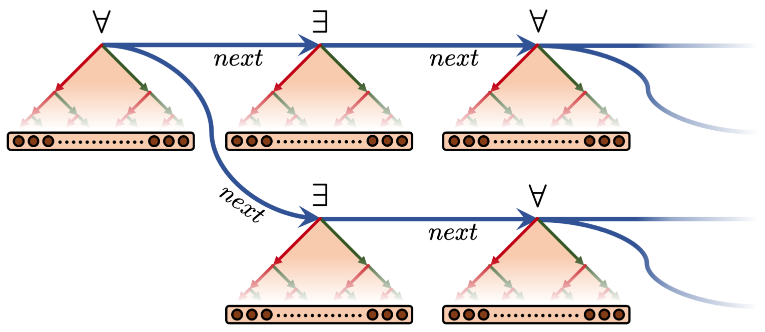
Lemma 6.13 For any model $\mathcal{I}$ of $\mathcal{K}_{enr}$ and any $\mathrm{d} \in \mathrm{Lvl}_0^{\mathcal{I}}$, there is an enriched configuration tree $\mathcal{E}$ and a homomorphism $\mathfrak{h}$ from $\mathcal{E}$ into $\mathcal{I}$ with $\mathfrak{h}(\varepsilon) = \mathrm{d}$.

Proof. We follow the proof scheme of Lemma 6.10. By Lemma 6.10, there is a homomorphism $\mathfrak{h}$ from a configuration tree $\mathcal{C}$ to $\mathcal{I}$ with $\mathfrak{h}(\varepsilon) = \mathrm{d}$. Moreover, as the symbols outside $\mathbf{R}_{unit} \cup \mathbf{C}_{unit} \cup \mathbf{C}_{conf}$ do not appear in Definition 6.8 we can assume that $\mathcal{C}$ interprets them as empty sets. Let $\mathcal{E} = (\Delta^{\mathcal{C}}, \cdot^{\mathcal{U}})$ be an interpretation that is obtained from changing the meaning of concepts from $\mathbf{C}_{enr}$ as follows: for any $\mathrm{C} \in \mathbf{C}_{enr}$ we let $\mathrm{C}^{\mathcal{E}} := \{w \mid \mathfrak{h}(w) \in \mathrm{C}^{\mathcal{C}}\}$. All other symbols are interpreted as in $\mathcal{C}$. Clearly $\mathfrak{h}$ is a homomorphism from $\mathcal{E}$ into $\mathcal{I}$ with $\mathfrak{h}(\varepsilon) = \mathrm{d}$ and it suffices to show that $\mathcal{E}$ is an enriched configuration tree (we already know that it is a configuration tree), which is done by routine investigation of Definition 6.11 and the presented GCIs.

The existence of the unique concept $\mathrm{C} \in \mathbf{C}_{ptr}$ claimed in the 1st item of Definition 6.11 is provided by the first three GCIs, namely the existence is due to (TrCov) and uniqueness due to (TrInitDisj[$\mathtt{t}$]) and (TrDisj[$\mathtt{t}, \mathtt{t}'$]). The 2nd item of Definition 6.11 is due to $\mathcal{I} \models$ (TransiCons) (more precisely, the first conjunct of the rhs). Similarly to the proof of Lemma 6.10, we establish the existence of a unique w_{phd} and desired properties of concepts $\mathrm{PHdHere}^{\mathcal{E}}$, $\mathrm{NoPHdHere}^{\mathcal{E}}$, $(\mathrm{PHdPos}_i^b)^{\mathcal{E}}$ and $\mathrm{PHdLet}_{\mathtt{a}}^{\mathcal{E}}$. Since such a proof is nearly identical (modulo adding the "P" letter in front of some concept names) to the one from the previous section, we do not repeat the details here. Then, the fact that w_{phd} satisfies $w_{phd} = w_{head} + d$ or is equal to 0 in case $\mathrm{Ini}^{\mathcal{E}} = \{\varepsilon\}$ is by, respectively, membership of ε in "PHdPos $+ d =$ HdPos"$^{\mathcal{E}}$ and $(\forall \ell_1 \forall r_1 \ldots \forall \ell_{\mathtt{N}} \forall r_{\mathtt{N}}\ (\mathrm{Lvl}_{\mathtt{N}} \rightarrow \mathrm{PHdPos}_i^b))^{\mathcal{E}}$, guaranteed by $\mathcal{I} \models$ (TransiCons) and $\mathcal{I} \models$ (IC). Finally, the satisfaction of the last item of Definition 6.11 is immediate by $\mathcal{I} \models$ (IC). □

6.6 Describing Accepting Quasi-Runs

Recall that a quasi-run $\mathfrak{R}$ of $\mathcal{M}$ is simply a tree labelled with configurations of $\mathcal{M}$ where the root is labelled with the initial configuration $\mathtt{s}_I \mathtt{0}^{2^{\mathtt{N}}}$. Each node representing an existential configuration has one child labelled with a quasi-successor configuration, while each node representing a universal configuration has two children labelled by quasi-successor configurations obtained via different transitions.



In order to represent an accepting quasi-run by a model, we employ the notion of a *quasi-computation tree* $\mathcal{Q}$, a structure intuitively defined from some $\mathfrak{R}$ as follows: replace every node of $\mathfrak{R}$ by its corresponding configuration tree, adequately enriched with information about its generating transition and the predecessor configuration. The roots of these enriched configuration trees are linked via the *next* role to express the quasi-succession relation of $\mathfrak{R}$. The roots of enriched configuration trees representing universal configurations are chosen to be labelled with L, their left *next*-child with L and their right *next*-child with R (both corresponding to existential configurations). As expected, the Ini concept decorates the root of the distinguished enriched configuration tree that represents $\mathfrak{R}$'s initial configuration. As our attention is restricted to *accepting* quasi-runs $\mathfrak{R}$, we require that no enriched configuration tree occurring in $\mathcal{Q}$ carries a rejecting state. We now give a formal definition of such a structure $\mathcal{Q}$.

Definition 6.14 (quasi-computation tree) A **quasi-computation tree** $\mathcal{Q}$ of $\mathcal{M}$ is an interpretation $\mathcal{Q} := (\Delta^{\mathcal{Q}}, \cdot^{\mathcal{Q}})$ satisfying the following properties:

- $\Delta^{\mathcal{Q}} := \mathfrak{T} \times \{0,1\}^{\leq \mathtt{N}+1}$, where $\mathfrak{T}$ is[a] a prefix-closed subset of $\{\mathfrak{1}\mathfrak{0}, \mathfrak{0}\mathfrak{0}\}^* \cdot \{\varepsilon, \mathfrak{0}, \mathfrak{1}\}$ with $\mathfrak{w}\mathfrak{1} \in \mathfrak{T}$ implying $\mathfrak{w}\mathfrak{0} \in \mathfrak{T}$.
- For every $\mathfrak{w} \in \mathfrak{T}$, the substructure of $\mathcal{Q}$ induced by $\{\mathfrak{w}\} \times \{0,1\}^{\leq \mathtt{N}+1}$ is isomorphic to an enriched configuration tree of $\mathcal{M}$ via the isomorphism $(\mathfrak{w}, w) \mapsto w$.
- $(\varepsilon, \mathfrak{w}) \in \mathrm{R}^{\mathcal{Q}}$ if $\mathfrak{w}$ ends with $\mathfrak{1}$, otherwise $(\varepsilon, \mathfrak{w}) \in \mathrm{L}^{\mathcal{Q}}$.
- For any $\mathfrak{w} \neq \mathfrak{w}'$ and arbitrary $w, w' \in \{0,1\}^{\leq \mathtt{N}+1}$ we have that $((\mathfrak{w}, w), (\mathfrak{w}', w')) \notin s^{\mathcal{Q}}$ holds for any $s \in \mathbf{R}_{unit} \setminus \{next\}$.
- $next^{\mathcal{Q}} \setminus \{(\mathrm{d},\mathrm{d}) \mid \Delta^{\mathcal{Q}} \times \Delta^{\mathcal{Q}}\} = \{((\mathfrak{w}, \varepsilon), (\mathfrak{w}\mathfrak{b}, \varepsilon)) \mid \mathfrak{w}\mathfrak{b}, \mathfrak{w} \in \mathfrak{T}, \mathfrak{b} \in \{\mathfrak{0}, \mathfrak{1}\}\}$.
- $\mathrm{Ini}^{\mathcal{Q}} = \{(\varepsilon, \varepsilon)\}$.
- For any $\mathfrak{w}\mathfrak{0} \in \mathfrak{T}$ with $(\mathfrak{w}, \varepsilon) \in \mathrm{St}_{\mathtt{s}}^{\mathcal{Q}}$ and $(\mathfrak{w}, \varepsilon) \in \mathrm{Let}_{\mathtt{a}}^{\mathcal{Q}}$
  - if $\mathfrak{w}\mathfrak{1} \in \mathfrak{T}$ then $(\mathfrak{w}\mathfrak{0}, \varepsilon) \in \mathrm{PTrns}^{\mathcal{Q}}_{\mathtt{T}_1(\mathtt{s},\mathtt{a})}$ and $(\mathfrak{w}\mathfrak{1}, \varepsilon) \in \mathrm{PTrns}^{\mathcal{Q}}_{\mathtt{T}_2(\mathtt{s},\mathtt{a})}$,
  - if $\mathfrak{w}\mathfrak{1} \notin \mathfrak{T}$ then $(\mathfrak{w}\mathfrak{0}, \varepsilon) \in \mathrm{PTrns}^{\mathcal{Q}}_{\mathtt{T}_1(\mathtt{s},\mathtt{a})}$ or $(\mathfrak{w}\mathfrak{0}, \varepsilon) \in \mathrm{PTrns}^{\mathcal{Q}}_{\mathtt{T}_2(\mathtt{s},\mathtt{a})}$.
- If $(\mathfrak{w}, w) \in \mathrm{HdHere}^{\mathcal{Q}}$ and $\mathfrak{w}\mathfrak{b} \in \mathfrak{T}$ then $(\mathfrak{w}\mathfrak{b}, w) \in \mathrm{PHdHere}^{\mathcal{Q}}$.
- $\mathrm{St}^{\mathcal{Q}}_{\mathtt{s}_R} = \emptyset$ as well as $(\mathfrak{w}, \varepsilon) \in \mathrm{St}^{\mathcal{Q}}_{\mathtt{s}_A}$ if and only if $\mathfrak{w} \in \mathfrak{T}$ and $\mathfrak{w}\mathfrak{0} \notin \mathfrak{T}$.

[a]This is just a scary-looking definition of a binary tree in which nodes at the i-th level have exactly 2 children if i is even and exactly one child otherwise. We use 𝔣𝔯𝔞𝔨𝔱𝔲𝔯 letters for quasi-computations.

We move on to provide an appropriate axiomatisation.

1. We incorporate all axioms from $\mathcal{K}_{enr}$ to ensure the indicated substructures correspond to enriched computation trees.
2. Every non-final existential configuration has one successor configuration while every non-final universal configuration has two. Final configurations do not have any successors. Below $\mathtt{s}_e \in \mathtt{Q}_\forall \setminus \{\mathtt{s}_A, \mathtt{s}_R\}$, $\mathtt{s}_e \in \mathtt{Q}_\exists \setminus \{\mathtt{s}_A, \mathtt{s}_R\}$, and $\mathtt{s}_f \in \{\mathtt{s}_A, \mathtt{s}_R\}$.

   (EConfSucc[$\mathtt{s}_e$]) $\mathrm{St}_{\mathtt{s}_e} \sqsubseteq \exists next.\mathrm{L} \sqcap \exists next.\mathrm{R}$
   (AConfSucc[$\mathtt{s}_a$]) $\mathrm{St}_{\mathtt{s}_a} \sqsubseteq \exists next.\top \sqcap \forall next.\mathrm{L}$
   (FinConfSucc[$\mathtt{s}_f$]) $\mathrm{St}_{\mathtt{s}_f} \sqsubseteq \forall next.\bot$
3. To transfer the previous head position to the successor configurations we use ($1 \leq i \leq \mathtt{N}, b \in \{0,1\}$):

   $$\textbf{(TransHeadPos}[i,b]\textbf{)}\ \ \mathrm{Lvl}_0 \sqcap \mathrm{HdPos}_i^b \sqsubseteq \forall next.\mathrm{PHdPos}_i^b$$
4. For any $\mathtt{s}_\exists \in \mathtt{Q}_\exists$ we specify that the corresponding configuration tree linked via *next*-role is a successor configuration of the current one.

   $$\textbf{(TransiExistState)}\ \ \mathrm{St}_{\mathtt{s}_\exists} \sqcap \mathrm{HdLet}_{\mathtt{a}} \sqsubseteq \bigsqcup_{\mathtt{t} \in \mathtt{T}(\mathtt{s}_\exists, \mathtt{a})} \forall next.\mathrm{PTrns}_{\mathtt{t}}$$
5. For every universal state $\mathtt{s}_\forall \in \mathtt{Q}_\forall$ and a letter $\mathtt{a}$ currently scanned by the head there are only two possible choices of transitions.

   $$\textbf{(TransiUnivStateL)}\ \ \mathrm{St}_{\mathtt{s}_\forall} \sqcap \mathrm{HdLet}_{\mathtt{a}} \sqsubseteq \forall next.(\mathrm{L} \to \mathrm{PTrns}_{\mathtt{T}_1(\mathtt{s}_\forall,\mathtt{a})})$$
   $$\textbf{(TransiUnivStateR)}\ \ \mathrm{St}_{\mathtt{s}_\forall} \sqcap \mathrm{HdLet}_{\mathtt{a}} \sqsubseteq \forall next.(\mathrm{R} \to \mathrm{PTrns}_{\mathtt{T}_2(\mathtt{s}_\forall,\mathtt{a})})$$
6. Since we want to have accepting quasi-runs of $\mathcal{M}$ only, we state that we never encounter the rejecting state.

   $$\textbf{(NoRejectState)}\ \ \mathrm{St}_{\mathtt{s}_R} \sqsubseteq \bot$$

Let $\mathcal{T}_{\mathcal{M}}$ be the set of all GCIs presented so far and let $\mathcal{A}_{\mathcal{M}}$ be an ABox composed of a single axiom $\mathrm{Ini}(\mathsf{a})$ for a fresh individual name a. Put $\mathcal{K}_{\mathcal{M}} := (\mathcal{A}_{\mathcal{M}}, \mathcal{T}_{\mathcal{M}})$. We claim that:

Lemma 6.15 The knowledge base $\mathcal{K}_{\mathcal{M}}$ is of size polynomial in $|\mathcal{M}|$. Any accepting quasi-computation tree $\mathcal{Q}$ of $\mathcal{M}$ is a model of $\mathcal{K}_{\mathcal{M}}$.

Proof. To see $\mathcal{Q} \models \mathcal{K}_{enr}$ it suffices to observe that (1) by the 2nd item of Definition 6.14 all the substructures of $\mathcal{Q}$ induced by $\{\mathfrak{w}\} \times \{0,1\}^{\leq \mathtt{N}+1}$ are isomorphic to some computation tree and hence, by Lemma 6.12 they satisfy $\mathcal{K}_{enr}$, (2) the use of roles from $\mathbf{R}_{unit} \setminus \{\mathit{next}\}$ is restricted to enriched configuration trees and hence $\mathcal{Q}$ satisfies all the GCIs not involving *next* and (3) the only GCI involving *next* from $\mathcal{K}_{enr}$ is (leaves-next-loop) and it is satisfied in $\mathcal{Q}$ due to the mentioned isomorphism property. Next, the satisfaction of (AConfSucc[s_a]), (TransiUnivStateL), and (TransiUnivStateR) by $\mathcal{Q}$ is due to the 7th item (1st subitem) of Definition 6.14. Similarly, we infer that $\mathcal{Q} \models$ (EConfSucc[s_e]) and $\mathcal{Q} \models$ (TransiExistState) by the 7th item (2nd subitem) of Definition 6.14. By the last item of Definition 6.14 we immediately conclude $\mathcal{Q} \models$ (FinConfSucc[s_f]) and $\mathcal{Q} \models$ (NoRejectState). Hence, it remains to prove satisfaction of (TransHeadPos[i, b]), which is immediate by the second to last item of Definition 6.14. □

Lemma 6.16 For any model $\mathcal{I}$ of $\mathcal{K}_{\mathcal{M}}$ there exists an accepting quasi-computation tree $\mathcal{Q}$ and a homomorphism $\mathfrak{h}\colon \mathcal{Q} \to \mathcal{I}$ with $\mathfrak{h}(\varepsilon, \varepsilon) = \mathsf{a}^{\mathcal{I}}$.

Proof. We construct a tree $\mathfrak{T}$ and its origin function $\mathfrak{f}\colon \mathfrak{T} \to \mathcal{I}$ as follows. First, let $\varepsilon \in \mathfrak{T}$ and $\mathfrak{f}(\varepsilon) := \mathsf{a}^{\mathcal{I}}$. We next proceed as follows. Take any word $\mathfrak{w} \in \mathfrak{T}$ and consider three cases:

- $\mathfrak{f}(\mathfrak{w})$ is labelled with a non-final universal state. Hence, by the first axiom provided, we know that $\mathfrak{f}(\mathfrak{w})$ has at least two *next*-successors, one of which is in $\mathrm{L}^{\mathcal{I}}$ and the other in $\mathrm{R}^{\mathcal{I}}$. Call them, respectively, $\mathrm{e}_l, \mathrm{e}_r$. Hence, we extend $\mathfrak{T}$ with the words $\mathfrak{w}\mathtt{0}, \mathfrak{w}\mathtt{1}$ and extend the function $\mathfrak{f}$ with $\mathfrak{f}(\mathfrak{w}\mathtt{0}) := \mathrm{e}_l$ and $\mathfrak{f}(\mathfrak{w}\mathtt{1}) := \mathrm{e}_r$. Repeat the process from $\mathfrak{w}\mathtt{0}$ and $\mathfrak{w}\mathtt{1}$.
- $\mathfrak{f}(\mathfrak{w})$ is labelled with a non-final existential state. Then we take its *next*-successor e and extend $\mathfrak{T}$ with $\mathfrak{w}\mathtt{0}$ and $\mathfrak{f}$ with $\mathfrak{f}(\mathfrak{w}\mathtt{0}) := \mathrm{e}$. Repeat the process from $\mathfrak{w}\mathtt{0}$.
- $\mathfrak{f}(\mathfrak{w})$ is labelled with a final state. No action required.

We associate a word $\mathfrak{w} \in \mathfrak{T}$ with an enriched configuration tree $\mathcal{E}_{\mathfrak{w}}$ such that there is a homomorphism $\mathfrak{g}_{\mathfrak{w}}$ from $\mathcal{E}_{\mathfrak{w}}$ to $\mathcal{I}$ with $\mathfrak{f}(\mathfrak{w}) = \mathfrak{g}_{\mathfrak{w}}(\varepsilon)$. The existence of $\mathcal{E}_{\mathfrak{w}}$ and $\mathfrak{g}_{\mathfrak{w}}$ is provided by Lemma 6.13. Finally, we decorate each node of $\mathcal{E}_{\mathfrak{w}}$ with “Pr” concepts as suggested by the homomorphism $\mathfrak{g}_{\mathfrak{w}}$. A $\mathfrak{T}$-quasi-computation tree $\mathcal{Q}$ is then defined by stipulating that, for every $\mathfrak{w} \in \mathfrak{T}$, the substructure of $\mathcal{Q}$ induced by $\{\mathfrak{w}\} \times \{0,1\}^{\leq \mathtt{N}+1}$ be isomorphic to the decorated $\mathcal{E}_{\mathfrak{w}}$. The homomorphism $\mathfrak{h}\colon \Delta^{\mathcal{Q}} \to \mathcal{I}$ is then defined componentwise by $(\mathfrak{w}, w) \mapsto \mathfrak{g}_{\mathfrak{w}}(w)$, essentially taking the disjoint unions of the homomorphisms for all enriched configuration trees. Since all the roles except *next* are restricted to the components and we made sure that the roots of $\mathcal{Q}$ were created from the elements linked via *next*-roles, we conclude that $\mathfrak{h}$ is the claimed homomorphism. □

6.7 Detecting Faulty Runs with a Single Conjunctive Query

We finally have reached the point where querying comes into play. Our last goal is to design *one single* conjunctive query that detects “faulty configuration progressions” in quasi-computation trees, meaning that it matches a pair of two positions in consecutive configuration trees representing the same cell and being untouched by the head of $\mathcal{M}$ yet storing different letters. Note that the lack of such cells in a quasi-computation tree means that any two consecutive configuration trees represent not only quasi-successor configuration but actually proper successors and hence the structure as such even represents a “proper” run. We start by formalising our requirements for such a query:

Lemma 6.17 There exists a conjunctive query $q_{\mathcal{M}}$ of size polynomial in $\mathtt{N}$ with two distinguished variables x, y such that for all quasi-computation trees $\mathcal{Q}$ we have $\mathcal{Q} \models_{\pi} q_{\mathcal{M}}$ if and only if there exists a word $\mathfrak{w}$, a letter $\mathfrak{b}$ and a word w of length $\mathtt{N}+1$ such that:

- $\pi(x) = (\mathfrak{w}, w)$, $\pi(y) = (\mathfrak{w}\mathfrak{b}, w)$,
- $\pi(y) \in \text{NoPHdAbv}^{\mathcal{Q}}$,
- $\pi(x) \in \mathbb{0}^{\mathcal{Q}}$ and $\pi(y) \in \mathbb{1}^{\mathcal{Q}}$.

Note the asymmetry in the 3rd bullet point above – we ignore the reverse constellation. Yet, due to our encoding if the reverse situation occurs then so does the original one. Hence, every mismatch in a sense causes two inconsistencies from the point of $\mathtt{N}+1$-level nodes. This solves the mystery of introducing level $\mathtt{N}+1$ in our configuration trees and the particular encoding of tape symbols: it is crucial for catching faulty progressions by using one single CQ. Before proving Lemma 6.17 we show how it implies the main theorem here, namely:

Theorem 6.18

Conjunctive query entailment over $\mathcal{ALC}^{\mathsf{Self}}$-KBs is 2ExpTime-hard.

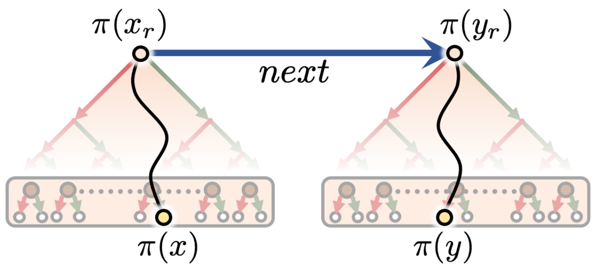
Proof. Since co2ExpTime=2ExpTime, it is sufficient to show that the CQ non-entailment problem over $\mathcal{ALC}^{\mathsf{Self}}$-KBs is 2ExpTime-hard. Take $\mathcal{K}_{\mathcal{M}}$ as defined in Section 6.6, and the query $q_{\mathcal{M}}$ as given by Lemma 6.17. Since both $\mathcal{K}_{\mathcal{M}}$ and $q_{\mathcal{M}}$ are of size polynomial w.r.t $|\mathcal{M}|$, it remains to show that $\mathcal{K}_{\mathcal{M}} \not\models q_{\mathcal{M}}$ if and only if $\mathcal{M}$ is accepting. The "if" direction is easy: we take an accepting run of $\mathcal{M}$ and turn it into a quasi-computation tree $\mathcal{Q}$. By Lemma 6.15 we conclude $\mathcal{Q} \models \mathcal{K}_{\mathcal{M}}$. We also have that $\mathcal{Q} \not\models q$ due to the fact that any two consecutive configuration trees represent proper successor configurations. For the second direction it suffices to show that if $\mathcal{M}$ is not accepting then $\mathcal{K}_{\mathcal{M}} \models q_{\mathcal{M}}$. Indeed, assume that $\mathcal{M}$ is not accepting and consider an arbitrary model $\mathcal{I}$ of $\mathcal{K}_{\mathcal{M}}$ (in case $\mathcal{K}_{\mathcal{M}}$ is unsatisfiable then trivially $\mathcal{K}_{\mathcal{M}} \models q_{\mathcal{M}}$). By Lemma 6.16 there is a quasi-computation tree $\mathcal{Q}$ and a homomorphism $\mathfrak{h} : \mathcal{Q} \to \mathcal{I}$ with $\mathfrak{h}(\varepsilon, \varepsilon) = \mathsf{a}^{\mathcal{I}}$. But this quasi-computation tree must represent a "faulty" run – in the opposite case it would correspond to an accepting run of $\mathcal{M}$, which does not exist by assumption. Hence there must be a match of $q_{\mathcal{M}}$ to $\mathcal{Q}$. As query matches are preserved under homomorphisms, we conclude $\mathcal{I} \models q_{\mathcal{M}}$. Thus all models $\mathcal{I}$ of $\mathcal{K}_{\mathcal{M}}$ have matches of $q_{\mathcal{M}}$, which implies $\mathcal{K}_{\mathcal{M}} \models q_{\mathcal{M}}$. □

In the forthcoming query definitions, we employ a convenient naming scheme. By writing $q[x, y]$ we indicate that the variables $x, y \in \text{Var}(q)$ are *global* (*i.e.* the same across (sub)queries that we might join together) while its other variables are *local* (*i.e.* pairwise different from any variables occurring in other queries — this can always be enforced by renaming). Going back to the query, we proceed as follows. We first prepare a query $q_{main}[x, y]$ with two global distinguished variables x, y that relates any two domain elements whenever they are leaf nodes of consecutive computation trees. Then $q_{main}[x, y]$ is combined with queries $q^{i}_{adr}[x, y]$ for all $1 \leq i \leq \mathtt{N}+1$ with the intended meaning that x and y have the same i-th bit of their addresses. Additionally, our final query will require that x be mapped to a node satisfying $\mathbb{0}$ and y to a node satisfying $\mathbb{1}$ and NoPHdHere.

To construct $q_{main}[x, y]$ we essentially employ Lemma 6.6.

Lemma 6.19 There exists a conjunctive query $q_{main}[x, y]$ of size polynomial in $|\mathcal{M}|$ such that for any quasi-computation tree $\mathcal{Q}$ the set $M_{q_{main}} := \{(\pi(x), \pi(y)) \mid \mathcal{Q} \models_{\pi} q_{main}\}$ is composed precisely of any pair of leaves of two consecutive configuration trees of $\mathcal{Q}$. Formally:

$$M_{q_{main}} = \big\{((\mathfrak{w}, w),(\mathfrak{w}\mathfrak{b}, v)) \in \Delta^{\mathcal{Q}} \mid |w| = |v| = \mathtt{N}+1, \mathfrak{b} \in \{0, 1\}\big\}.$$



Proof. It suffices to take $q_{main} \coloneqq q_{rl}[x_r, x] \wedge \mathit{next}(x_r, y_r) \wedge q_{rl}[y_r, y]$. Let $\mathcal{Q} \models_\pi q_{main}$. That $M_{q_{main}}$ is a superset of the set above follows from the fact that quasi-computation trees are computation units and hence, containment follows by Corollary 6.7. We now focus on the other direction. Note that by the 5th item of Definition 6.14 we know that $\pi(x_r)$ and $\pi(y_r)$ must be two distinct roots of enriched configuration trees $\mathcal{E}_{x_r}, \mathcal{E}_{y_r}$. By the 4th item of Definition 6.14 we know that the interpretation of the rs and ℓs is restricted to pairs of domain elements located inside the same enriched configuration tree (and by their definition to configuration trees and by their definition to configuration units). Since q_{rl} only employs the roles ℓ_i, r_i and the concepts $\mathrm{Lvl}_0, \mathrm{Lvl}_{\mathtt{N}+1}$ we conclude that q_{rl} has exactly the same set of matches in $\mathcal{E}_{x_r}$ as in its underlying unit. Hence, by Corollary 6.7 we know that x (resp. y) is indeed mapped to a leaf of $\mathcal{E}_{x_r}$ (resp. to a leaf of $\mathcal{E}_{y_r}$), which finishes the proof. □

The next part of our query construction focuses on sub-queries $q^i_{adr}[x, y]$ that are meant to relate leaves having equal i-th bits of addresses. In order to construct it we combine together several smaller queries, written in path syntax below.

- We let $q_\downarrow[x, y] \coloneqq (\ell_1; r_1; \ldots; \ell_{\mathtt{N}+1}; r_{\mathtt{N}+1})(x, y)$ define the *top-down query*. It intuitively traverses an enriched configuration tree in a top-down manner. Note that $q_\downarrow[x, y]$ is actually the major sub-query of $q_{rl}[x, y]$.
- The ℓ_i*-top-down query* $q_{\ell i\downarrow}[x, y]$ is similar to $q_\downarrow[x, y]$, but with the $\ell_i; r_i$ part replaced by just ℓ_i. The intended behaviour is that again a tree is traversed from root to leaves, but this time, an ℓ_i edge must be crossed when going from the $(i-1)$-th to the i-th level. The r_i*-top-down query* $q_{ri\downarrow}[x, y]$ is defined alike, by replacing $\ell_i; r_i$ in $q_\downarrow[x, y]$ with r_i.

An important ingredient in the construction is the query $q^{i\text{-th bit}}_{=0}[x, y]$ defined as follows:

$$\mathrm{Lvl}_{\mathtt{N}+1}(x) \wedge q_{\ell i\downarrow}[x',x] \wedge \mathit{next}(x',y') \wedge q_{\ell i\downarrow}[y',y] \wedge \mathrm{Lvl}_{\mathtt{N}+1}(y).$$

In total analogy, we define $q^{i\text{-th bit}}_{=1}[x, y]$ by using $q_{ri\downarrow}$ instead of $q_{\ell i\downarrow}$. Any match π of the query $q^{i\text{-th bit}}_{=b}[x, y]$ instantiates the variables x and y in a quasi-computation tree $\mathcal{Q}$ according to one of the following two scenarios: either $\pi(x) = \pi(y)$ or $\pi(x)$ and $\pi(y)$ are leaves in two consecutive enriched configuration trees inside the quasi-computation tree and both of these leaves have their i-th address bit set to b. The above intuition meets its formalisation in:

Lemma 6.20 Let $\mathcal{Q}$ be a quasi-computation tree and let $M_{q^{i\text{-th bit}}_{=b}} \coloneqq \{(\pi(x),\pi(y)) \mid \mathcal{Q} \models_\pi q^{i\text{-th bit}}_{=b}\}$ for $b \in \{0,1\}$. Then $M_{q^{i\text{-th bit}}_{=b}}$ is equal to the union of M^b_1 and M^b_2 given below:

$$M^b_1 \coloneqq \{((\mathfrak{w}, w), (\mathfrak{w}, w))\}, \qquad M^b_2 \coloneqq \{((\mathfrak{w}, ubv), (\mathfrak{w}\mathfrak{b}, u'bv')) \mid |u| = |u'| = i-1\}.$$

Proof. We prove the statement for $b = 0$, the case for $b = 1$ is symmetric. First we show $M^0_1 \subseteq M_{q^{i\text{-th bit}}_{=0}}$. This is easy: for any leaf $\mathrm{d} = (\mathfrak{w}, w)$ we map all variables of $q^{i\text{-th bit}}_{=0}[x, y]$ into d; this is a match due to the presence of all the self-loops at the leaves. To show $M^0_2 \subseteq M_{q^{i\text{-th bit}}_{=0}}$ we take any $\mathrm{d} \coloneqq (\mathfrak{w}, w)$ and $\mathrm{e} \coloneqq (\mathfrak{w}\mathfrak{b}, v)$. Let π be a variable assignment that maps x to d, y to e, x' to $(\mathfrak{w}, \varepsilon)$, y' to $(\mathfrak{w}\mathfrak{b}, \varepsilon)$. The variables of $q_{\ell i\downarrow}[x', x]$ are mapped to $(\mathfrak{w}, w_j)$, where w_j is the prefix of w of length j following the path from $(\mathfrak{w}, \varepsilon)$ to $(\mathfrak{w}, w)$ level-by-level. We stress that $((\mathfrak{w}, w_{i-1}), (\mathfrak{w}, w_i)) \in \ell^{\mathcal{Q}}_i$ holds, which is crucial for the construction to work and that every

$(\mathfrak{w}, w_j)$ node has all ℓ- and r-loops. The variables of $q_{\ell i\downarrow}[y', y]$ are mapped analogously. After noticing that $\mathrm{d}, \mathrm{e} \in \mathrm{Lvl}^{\mathcal{Q}}_{\mathtt{N}+1}$ and that $(\pi(x'), \pi(y')) \in \mathit{next}^{\mathcal{Q}}$ holds, we conclude that π is clearly a match of $q^{i\text{-th bit}}_{=0}[x, y]$ to $\mathcal{Q}$.

Now we focus on showing that $M_{q^{i\text{-th bit}}_{=0}[x,y]} \subseteq M^0_1 \cup M^0_2$. Take any match π and note that x, y must be mapped to leaves. For $\pi(x')$ and $\pi(y')$ we consider the two cases:

1. $\pi(x') = \pi(y')$. As the roots do not have *next*-loops, $\pi(x')$ must be a leaf. This implies that all variables of $q_{\ell i\downarrow}[x', x]$ map into a single domain element (otherwise we would not reach a leaf after traversing such a path). Arguing similarly we infer that all variables of $q_{\ell i\downarrow}[y', y]$ are mapped to the same element. Thus $\pi(x) = \pi(y)$ holds.
2. $\pi(x') \neq \pi(y')$. Since all incoming *next* roles from leaves are self-loops, we conclude that $\pi(x')$ is the root of some enriched quasi-computation tree and $\pi(y')$ is the root of some corresponding quasi-successor in $\mathcal{Q}$ (by the definition of $\mathit{next}^{\mathcal{Q}}$). By the satisfaction of $q_{\ell i\downarrow}[x', x]$ we know that there exists a sequence of domain elements contributing to a path from $\pi(x')$ to $\pi(x)$ witnessing its satisfaction. Moreover, note that since the subquery $q_{\ell i\downarrow}[x', x]$ leads from the root to a leaf it implies that we necessarily cross the ℓ_i role at the $(i-1)$-th level, meaning that the i-th bit of the address of $\pi(x)$ is equal to 0. Thus we infer that $\pi(x) \in (\mathrm{Ad}^0_i)^{\mathcal{Q}}$. Reasoning analogously we conclude that $\pi(y) \in (\mathrm{Ad}^0_i)^{\mathcal{Q}}$.

□

We are now ready to present the query $q^i_{adr}[x, y]$ pairing leaves in consecutive enriched configuration trees with coinciding i-th address bit. Its correctness is established in Lemma 6.21.

$$q^i_{adr}[x, y] \coloneqq q_{main}[x, y] \wedge q^{i\text{-th bit}}_{=0}[x, z] \wedge q^{i\text{-th bit}}_{=1}[z, y].$$

Lemma 6.21 For any quasi-computation tree $\mathcal{Q}$ we have that $M_{q^i_{adr}} \coloneqq \{(\pi(x), \pi(y)) \mid \mathcal{Q} \models_\pi q^i_{adr}[x, y]\}$ is composed precisely of the leaf pairs in two consecutive enriched configuration trees of $\mathcal{Q}$ having equal i-th bit of address. Formally:

$$M_{q^i_{adr}} = M_{q_{main}} \cap \Big(\big(\mathrm{Ad}^{0\,\mathcal{Q}}_i \times \mathrm{Ad}^{0\,\mathcal{Q}}_i\big) \cup \big(\mathrm{Ad}^{1\,\mathcal{Q}}_i \times \mathrm{Ad}^{1\,\mathcal{Q}}_i\big) \Big).$$

Proof. By employing the definition of the query, Lemma 6.20 and relational calculus we conclude that $M_{q^i_{adr}} = M_{q_{main}} \cap \big(M_{q^{i\text{-th bit}}_{=0}} \circ M_{q^{i\text{-th bit}}_{=1}}\big) = M_{q_{main}} \cap \big((M^0_1 \cup M^0_2) \circ (M^1_1 \cup M^1_2)\big) = M_{q_{main}} \cap \big(M^0_1 \cup M^1_2 \cup M^0_2\big) = M^1_2 \cup M^0_2$, which finishes the proof. □

By collecting queries presented in this Section, we are finally ready to present our query

$$q_{\mathcal{M}} \coloneqq \bigwedge_{i=1}^{\mathtt{N}+1} q^i_{adr}[x, y] \wedge \mathrm{NoPHdAbv}(y) \wedge \mathbb{0}(x) \wedge \mathbb{1}(y)$$

by means of which we can conclude with the proof of Lemma 6.17.

Proof of Lemma 6.17. Let $q_{\mathcal{M}}$ as defined above and observe that its size is clearly polynomial in $\mathtt{N}$. Note that $q_{\mathcal{M}}$ satisfies our requirements: The 1st item follows from two lemmas: the fact that x and y are mapped to leaves of two consecutive enriched configuration trees follows from Lemma 6.19 and the fact that x and y are mapped to nodes having equal addresses follows from Lemma 6.21 applied for every $1 \leq i \leq \mathtt{N}+1$. The 2nd and the 3rd points hold since we supplemented our query with $\mathrm{NoPHdAbv}(y) \wedge \mathbb{0}(x) \wedge \mathbb{1}(y)$. □

Hardness, shown in this chapter, came as a quite surprise to us. The key insight of our proof (and maybe the take-home message from this section) is that the presence of Self allows us to mimic case distinction over paths (and hence the handling of disjunctive information) through concatenation, by providing the opportunity for one of the two disjuncts to idle by "circling in place". On a last note, our result also holds for plain $\mathcal{ALC}^{\mathsf{Self}}$ TBoxes, since the only ABox assertion $\mathrm{Ini}(\mathsf{a})$ can be replaced by the concept inclusion $\top \sqsubseteq \exists aux.\mathrm{Ini}$ for an auxiliary role name aux.

Corollary 6.22

Conjunctive query entailment over $\mathcal{ALC}^{\mathsf{Self}}$-TBoxes is 2ExpTime-hard. Hence, the conjunctive query entailment problem for every member of the $\mathcal{Z}$ family of DLs is 2ExpTime-hard as well.

Part II

Quasi-Forest Satisfiability of $\mathcal{ZOIQ}$

7

Deciding $\mathcal{ZOIQ}$ over Quasi-Forests

Contents

Motivation and Our Contribution

When reasoning about ontologies and complexity, Vardi [Var82] observed that measuring the user's data and the background ontology equally is not realistic, as the data tends to be huge in comparison to the ontology. To provide a more fine-grained notion of complexity, he invented the notion of the *data complexity* in which we treat the ontology (TBox) as fixed upfront and only the user's data (ABox) varies. The satisfiability problem for DLs is then usually easier in terms of data complexity (NP-complete or higher) than in terms of combined complexity (ExpTime-complete or higher). For instance, the two-variable fragment of first-order logic extended with counting [Pra09] (encoding DLs up to $\mathcal{ALCBIOQ}\mathsf{Self}$) is NP-complete, and thus the very expressive DL $\mathcal{SROIQ}$ [Kaz08], the logical core of OWL2. Regarding DLs with path expressions, not much is known, as the NP-completeness of $\mathcal{ALCI}^{\mathsf{Self}}_{\mathsf{reg}}$ [JLMS18] was established only recently. We are not aware of any results on data complexity of logics when both regular expressions and counting or nominals are allowed. Our results intend to fill these gaps.

In this chapter we study the data complexity of $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, and $\mathcal{ZOI}$, namely the maximal decidable fragments of $\mathcal{ZOIQ}$ that possess the so-called *quasi-forest model property*, a suitable generalisation of the well-known forest model property for $\mathcal{ALC}$. For the uniformity of our approach, we actually focus on the satisfiability problem for full $\mathcal{ZOIQ}$ but over quasi-forests, and establish its NP-completeness. This completes the data complexity landscape for decidable fragments of $\mathcal{ZOIQ}$ that remained open for more than a decade, and reproves known results on the $\mathcal{SR}$ family of DLs.

Overview of the Chapter and Prerequisites

We start with Section 7.1 by defining a suitable notion of quasi-forest models. As our notion of quasi-forest models is slightly different from the one from the literature, we discuss the differences in Section 7.2. Next, in Section 7.3, we provide a characterisation of how paths in quasi-forests look like and how they can be decomposed into interesting pieces. Section 7.4 provides suitable definitions of automata decorations: a toolkit for modular satisfaction of automata constraints in quasi-forests. Similarly, Section 7.5 provides a similar toolkit for dealing with number restrictions. Section 7.6 defines a relevant notion of summaries of quasi-forest models, a compact way of representing quasi-forest models of $\mathcal{ZOIQ}$-KBs. Finally, we present an algorithm to decide quasi-forest satisfiability of $\mathcal{ZOIQ}$ in Section 7.8. In addition to that, Chapter 8 shows how the algorithm from Section 7.8 can be used to establish the coNExpTime upper bound for the entailment for rooted queries in $\mathcal{ZIQ}$.

We assume that the reader is familiar with Section 3.1 and Preliminaries. Be warned! The forthcoming chapter is the most technical result the author of the thesis ever obtained.

High Level Algorithm Description

To establish NP-completeness of the satisfiability for $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, and $\mathcal{ZOI}$ in a uniform and elegant way, we focus on the satisfiability of $\mathcal{ZOIQ}$ over quasi-forests. One of the key ingredients is an exponential time algorithm for deciding if a $\mathcal{ZOIQ}$-KB has a quasi-forest model [Ort10, L. 3.4.1, Thm. 3.4.2]. While this algorithm cannot be used directly to solve the satisfiability problem in optimal time, we stress that we can still employ it as a black-box to decide quasi-forest satisfiability of KBs that have sizes independent from the ABox. In our approach, we intend to construct a quasi-forest model of an input $\mathcal{ZOIQ}$-KB in two steps, *i.e.* we construct its root part (dubbed the *clearing*) separately from its subtrees. Our algorithm first pre-computes (an exponential w.r.t. the size of the TBox but of constant size if the TBox is fixed) set of quasi-forest-satisfiable $\mathcal{ZOIQ}$-concepts that indicate possible subtrees that can be "plugged in" to the clearing of the intended model. Then it guesses (in NP) the intended clearing and verifies its consistency in PTime based on the pre-computed concepts and roles. For the feasibility of our "modular construction" a lot of bookkeeping needs to be done. Most importantly, certain *decorations* are employed to "relativise" and decide the satisfaction of *automata concepts* and *number restrictions* in an incomplete, fragmented forest.
I. The first type of decorations, given an automaton $\mathcal{A}$, aggregate information about existing paths realising $\mathcal{A}$ and starting at one of the roots of the intended model. As a single such path may visit several subtrees, we cut such paths into relevant pieces and summarise them by means of "shortcut" roles and $\mathcal{ZOIQ}$-concepts describing paths fully contained inside a single subtree.
II. The second type of decorations "localise" counting in the presence of nominals, as the nominals may have successors outside their own subtree and the clearing.

These two "small tricks", obfuscated by various technical difficulties, are the core ideas behind our quasi-forest-satisfiability algorithm. See the forthcoming sections for more details.

7.1 Quasi-Forests Models

In this section we adapt a handy notion of *quasi-forests* [CEO09, Def. 3.2], defined similarly to other forest-like structures from Section 3.1. In what follows we employ standard set-theoretic reconstruction of the notion of an $\mathbb{N}$-forest as a prefix-closed subset of $\mathbb{N}^+$ without the empty word ε.

Definition 7.1 Let $\mathbf{N}_\mathbf{I}^{\mathcal{A}}$ and $\mathbf{N}_\mathbf{I}^{\mathcal{T}}$ be finite subsets of $\mathbf{N_I}$, Root $\in \mathbf{N_C}$, and *child*, *edge*, *id* $\in \mathbf{N_R}$. An interpretation $\mathcal{I}$ is an **$(\mathbf{N}_\mathbf{I}^{\mathcal{A}}, \mathbf{N}_\mathbf{I}^{\mathcal{T}})$-quasi-forest** if its domain $\Delta^{\mathcal{I}}$ is an $\mathbb{N}$-forest,

$$\begin{aligned}
\mathrm{Root}^{\mathcal{I}} &= \Delta^{\mathcal{I}} \cap \mathbb{N} = \{\mathsf{a}^{\mathcal{I}} \mid \mathsf{a} \in \mathbf{N_I}\} = \{\mathsf{a}^{\mathcal{I}} \mid \mathsf{a} \in (\mathbf{N}_\mathbf{I}^{\mathcal{A}} \cup \mathbf{N}_\mathbf{I}^{\mathcal{T}})\},\\
\mathit{child}^{\mathcal{I}} &= \{(\mathrm{d}, \mathrm{d}\cdot n) \mid \mathrm{d}, \mathrm{d}\cdot n \in \Delta^{\mathcal{I}}, n \in \mathbb{N}\},\\
\mathit{edge}^{\mathcal{I}} &= \textstyle\bigcup_{r \in \mathbf{N_R}} r^{\mathcal{I}} \cup (r^-)^{\mathcal{I}},\\
\mathit{id}^{\mathcal{I}} &= \{(\mathrm{d}, \mathrm{d}) \mid \mathrm{d} \in \Delta^{\mathcal{I}}\},
\end{aligned}$$

and for all roles $r^{\mathcal{I}}$ and all pairs $(\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}$ at least one of the following conditions hold:

(i) both d and e belong to $\mathrm{Root}^{\mathcal{I}}$,

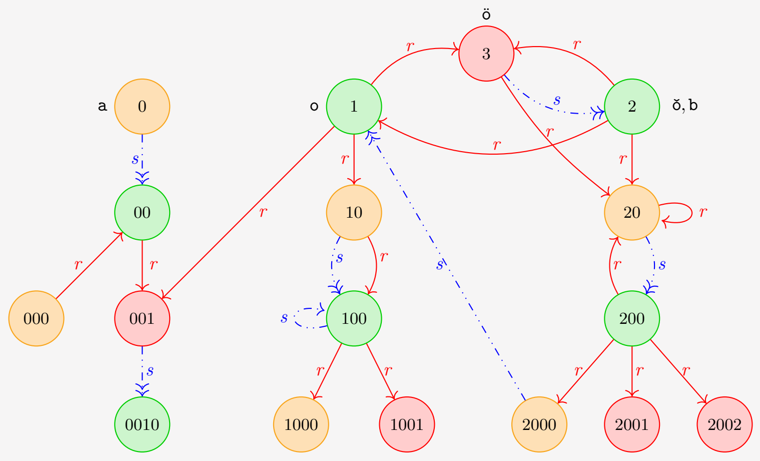
(ii) one of d or e is equal to $\mathsf{o}^{\mathcal{I}}$ for some name $\mathsf{o} \in \mathbf{N}_{\mathbf{I}}^{\mathcal{T}}$,

(iii) $(\mathrm{d}, \mathrm{e}) \in id^{\mathcal{I}} \cup child^{\mathcal{I}} \cup (child^{-})^{\mathcal{I}}$.

For convenience, we refer to pairs $(\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}$ satisfying the second condition as **backlinks**, and the ones satisfying $(\mathrm{d}, \mathrm{e}) \in id^{\mathcal{I}}$ as **self-loops**. The **clearing** of $\mathcal{I}$ is the restriction of $\mathcal{I}$ to $\mathrm{Root}^{\mathcal{I}}$. We call a quasi-forest **N-bounded** if all nodes have at most N children.

The names from $\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}$ are dubbed *nominals*, and will be usually denoted with decorated letters o. Their interpretations are usually referred as *nominal roots*. An example quasi-forest is presented in Example 7.2 below. In total analogy to Section 3.1, we employ suitable notions from graph theory such as *node*, *root*, *child*, *parent*, or *descendant*, defined as expected in accordance with $\Delta^{\mathcal{I}}$, $child^{\mathcal{I}}$ and $\mathrm{Root}^{\mathcal{I}}$. For instance, d is a descendant of c whenever $(\mathrm{c}, \mathrm{d}) \in (child^{\mathcal{I}})^{+}$.

Example 7.2. An example 3-bounded $(\{\mathsf{a}, \mathsf{b}\}, \{\mathsf{o}, \ddot{\mathsf{o}}, \breve{\mathsf{o}}\})$-quasi-forest $\mathcal{I}$ is depicted below. For readability we have omitted the interpretations of Root, *child*, *id*, and *edge*.



The element 1 has a unique child 10. The roots of $\mathcal{I}$ are 0, 1, 2, and 3. The nominal roots are 1,2, and 3. The pairs $(1, 001)$, $(2000, 1)$, and $(3, 20)$ are example backlinks.

We next lift the notion of quasi-forests to models of $\mathcal{ZOIQ}$-KBs. Indeed:

Definition 7.3 A **quasi-forest model** of a $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ is an $(\mathsf{ind}(\mathcal{A}), \mathsf{ind}(\mathcal{T}))$-quasi-forest satisfying $\mathcal{K}$, where $\mathsf{ind}(\mathcal{A})$ and $\mathsf{ind}(\mathcal{T})$ are the sets of all individual names from $\mathcal{A}$ and $\mathcal{T}$.

We remark that our notion of quasi-forests slightly differs from the original definition of quasi-forests by Calvanese et al. [CEO09, Def. 3.2]. We show however, in the next section, that the differences between these two definitions are negligible. Here comes the last interesting notion required in this section. While it is not strictly related with the forthcoming algorithm, it will be useful in Chapter 10 to establish that $\mathcal{ZOI}$ and $\mathcal{ZOQ}$ are finitely controllable [BK22].

Definition 7.4 Let $\mathcal{I}$ be an $(\mathbf{N}_{\mathbf{I}}^{\mathcal{A}}, \mathbf{N}_{\mathbf{I}}^{\mathcal{T}})$-quasi-forest, and $\mathscr{A}$ be an NFA. Call a path $\rho_1 \ldots \rho_n$ **downward** if for all indices $i < n$ we have that either ρ_{i+1} is a child of ρ_i, or $\rho_{i+1} = \rho_i$. A path is **nominal-downward** if it has the form $\rho{\cdot}\mathsf{o}^{\mathcal{I}}{\cdot}\hat{\rho}$ for some nominal $\mathsf{o} \in \mathbf{N}_{\mathbf{I}}^{\mathcal{T}}$, a downward path ρ, and a path $\hat{\rho}$. An element d **downward realises** an NFA $\mathscr{A}$ in $\mathcal{I}$ if there exists a downward or nominal-downward $\mathscr{A}_{\mathsf{q},\mathsf{q}'}$-path starting from d. We say that $\mathscr{A}$ is **downward realisable** in $\mathcal{I}$ if for all pairs of states q, q' of $\mathscr{A}$ and all elements d in $(\exists\mathscr{A}_{\mathsf{q},\mathsf{q}'})^{\mathcal{I}}$, the element d downward realise $\mathscr{A}_{\mathsf{q},\mathsf{q}'}$.

The results by Calvanese et al. [CEO09, Prop. 3.3] and Ortiz [Ort10, Lemma 3.4.1, Thm. 3.4.2] advocate the use of quasi-forest models when reasoning in the $\mathcal{Z}$ family of DLs. We state it below.

Lemma 7.5 Consider a knowledge-base $\mathcal{K}$ (in Scott's normal form) written in $\mathcal{ZOQ}$, $\mathcal{ZOI}$, or $\mathcal{ZIQ}$. Then there exists a polynomial-time computable number N, which is exponential in $|\mathcal{T}|$, such that:

- $\mathcal{K}$ is satisfiable if and only if $\mathcal{K}$ has an N-bounded quasi-forest model, and
- for all P2RPQs q, we have $\mathcal{K} \not\models q$ if and only if there exists an N-bounded quasi-forest model of $\mathcal{K}$ violating q.

Moreover, if $\mathcal{K}$ is written either in $\mathcal{ZOQ}$ or $\mathcal{ZOI}$, the quasi-forest models $\mathcal{I}$ of $\mathcal{K}$ guaranteed in the above statements have the property that each NFA $\mathcal{A}$ that appears in $\mathcal{K}$ is downward realisable in $\mathcal{I}$.

We call the quasi-forest (counter)models of $\mathcal{ZOIQ}$-KBs **canonical** whenever they are N-bounded for the number N guaranteed by Lemma 7.5. Thus, the lemma above tell us that for the deciding the satisfiability and query entailment over $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, and $\mathcal{ZOI}$, only the class of canonical quasi-forest models is relevant. We say that a $\mathcal{ZOIQ}$-KB is **quasi-forest satisfiable** whenever it has a canonical quasi-forest model. The quasi-forest satisfiability problem is defined accordingly. The following follows from the work of Ortiz [Ort10, Lemma 3.4.1, Thm. 3.4.2]. The proof will be given in the next section.

Lemma 7.6 The quasi-forest satisfiability problem for $\mathcal{ZOIQ}$-KBs is ExpTime-complete. In particular, this implies that the satisfiability of $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, and $\mathcal{ZOI}$-KBs is ExpTime-complete.

7.2 Comparing the Notions of Quasi-Forests Models

In this section we discuss differences between our definition of a quasi-forest model (Definition 7.1 and Definition 7.3) and the one proposed by Calvanese et al. [CEO09, Def. 3.2] in their seminal paper. As the original definition of Calvanese et al. contains minor errors[1] we adapt the presentation from the PhD thesis of Ortiz [Ort10, Def. 3.2.2]. We strongly recommend the reader to skip this section at first reading.

Definition 7.7 (Rephrased Def. 3.2.2 from the PhD Thesis of M. Ortiz) Let $\mathbf{N_I^{\mathcal{K}}}$ be a finite subset of $\mathbf{N_I}$. An interpretation $\mathcal{I}$ is an **$\mathbf{N_I^{\mathcal{K}}}$-Ortiz-forest** if its domain $\Delta^{\mathcal{I}}$ is a connected $\mathbb{N}$-forest satisfying $\Delta^{\mathcal{I}} \cap \mathbb{N} = \{\mathsf{a}^{\mathcal{I}} \mid \mathsf{a} \in \mathbf{N_I}\} = \{\mathsf{a}^{\mathcal{I}} \mid \mathsf{a} \in \mathbf{N_I^{\mathcal{K}}}\}$, and for all roles $r^{\mathcal{I}}$ and all pairs $(\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}$ at least one of the following conditions hold:

(I) at least one of d or e is $\mathbf{N_I^{\mathcal{K}}}$-named,

(II) d = e, d is a child of e, or e is a child of d.

An **Ortiz-forest model** of a $\mathcal{ZOIQ}$-KB $\mathcal{K}$ is an $\mathsf{ind}(\mathcal{K})$-Ortiz-forest satisfying $\mathcal{K}$. An Ortiz-forest is **proper** if interprets all "special" names (namely Root, *child*, *id*, and *edge*) as $\emptyset$.

Clearly every quasi-forest can be seen as an Ortiz-forest, but not necessarily vice versa. Let us highlight the crucial differences between these two notions. One obvious difference is that Ortiz-forests do not assume "special" concept and role names, namely Root, *child*, *id*, and *edge*. Such concepts and roles can be however easily incorporated. The second difference is more important. Given a $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$, we see that Ortiz-forest-models treat all individual names from $\mathcal{K}$ as nominals, while our quasi-forests make a distinction between nominals (that come from the TBox $\mathcal{T}$) and other individual names (that come from the ABox $\mathcal{A}$). This leads to a stricter definition of a forest, as the "backlinks to non-nominal roots" are, in contrast to Ortiz-forests, not allowed in quasi-forests.

Quasi-forests with finitely many non-empty roles can be described in $\mathcal{ZOIQ}$ in the following sense.

[1]The authors forgot to indicate that the roots of their quasi-forests are precisely the interpretations of named individuals. Without this assumption, every interpretation can be seen as a forest: just make every element a root.

Lemma 7.8 Let $\mathbf{N}_\mathbf{I}^\mathcal{A}$ and $\mathbf{N}_\mathbf{I}^\mathcal{T}$ be finite sets of individual names, and R be a finite set of role names. One can compute a $\mathcal{ZOIQ}$-KB $\mathcal{F}$ (of size polynomial w.r.t. $(|\mathbf{N}_\mathbf{I}^\mathcal{A}|+|\mathbf{N}_\mathbf{I}^\mathcal{T}|+|\mathsf{R}|)$), such that for any Ortiz-forest $\mathcal{I}$ that interprets all the role names outside $\mathsf{R} \cup \{id, child, edge\}$ as empty sets, we have $\mathcal{I} \models \mathcal{F}$ if and only if $\mathcal{I}$ is an $(\mathbf{N}_\mathbf{I}^\mathcal{A}, \mathbf{N}_\mathbf{I}^\mathcal{T})$-quasi-forest.

Proof. We provide a rather straightforward axiomatisation of $(\mathbf{N}_\mathbf{I}^\mathcal{A}, \mathbf{N}_\mathbf{I}^\mathcal{T})$-quasi-forests. The desired $\mathcal{ZOIQ}$-KB $\mathcal{F}$ is composed of all the axioms stated below.

- $(\mathbf{N}_\mathbf{I}^\mathcal{A} \cup \mathbf{N}_\mathbf{I}^\mathcal{T})$-named elements are precisely the roots of an $(\mathbf{N}_\mathbf{I}^\mathcal{A}, \mathbf{N}_\mathbf{I}^\mathcal{T})$-quasi-forest.

$$\text{Root} \equiv \bigsqcup_{\mathsf{a} \in (\mathbf{N}_\mathbf{I}^\mathcal{A} \cup \mathbf{N}_\mathbf{I}^\mathcal{T})} \{\mathsf{a}\}$$

- The role name *id* is interpreted as the identity relation.

$$\top \sqsubseteq ((\leq 1\ id).\top) \sqcap \exists id.\mathsf{Self}$$

- The role name *edge* is interpreted as the union of interpretation of all role names from R, *child*, *id*, and their inverses.

$$\top \sqsubseteq \forall\left(edge \setminus \left(\bigcup_{r \in \mathsf{R} \cup \{child, id\}} (r \cup r^-)\right)\right).\bot, \quad \top \sqsubseteq \forall\left(\left(\bigcup_{r \in \mathsf{R} \cup \{child, id\}} (r \cup r^-)\right) \setminus edge\right).\bot$$

- Backlinks are allowed only between non-root elements and nominal roots.

$$\neg\text{Root} \sqsubseteq \forall\big(edge \setminus (child \cup child^- \cup id)\big).\left(\bigsqcup_{\mathsf{o} \in \mathbf{N}_\mathbf{I}^\mathcal{T}} \{\mathsf{o}\}\right)$$

- The interpretation of *child* forms a forest, *i.e.* the following three conditions hold.
  - Every element has at most one parent.

$$\top \sqsubseteq (\leq 1\ child^-).\top$$

  - Roots are precisely the elements with no parents.

$$\text{Root} \equiv \forall child^-.\top$$

  - Every element reaches a root via the inverse of *child* relation in a finite number of steps.

$$\top \sqsubseteq \exists (child^-)^*.\text{Root}$$

It can be readily verified that the above axiomatisation does its job. □

With the above axiomatisation, we infer the ExpTime-completeness of quasi-forest satisfiability of $\mathcal{ZOIQ}$.

Corollary 7.9 (Lemma 7.6 reformulated.)

For a given number N encoded in binary, deciding whether a $\mathcal{ZOIQ}$-KB has an N-bounded quasi-forest model is ExpTime-complete. In particular, deciding whether a KB written in $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, or $\mathcal{ZOI}$ has a quasi-forest model is ExpTime-complete.

Proof. Let $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ be an input $\mathcal{ZOIQ}$-KB. If $\mathcal{K}$ is in $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, or $\mathcal{ZOI}$, we let N be the exponential bound (w.r.t. $|\mathcal{T}|$) on the branching of quasi-forests, infeed from Lemma 7.5. Otherwise, let N be as given as an input. We stress that N is given in binary encoding.

Let $\mathcal{F}$ be the $\mathcal{ZOIQ}$-KB from Lemma 7.8, written for $\mathbf{N}_{\mathbf{I}}^{\mathcal{A}} \coloneqq \mathsf{ind}(\mathcal{A})$, $\mathbf{N}_{\mathbf{I}}^{\mathcal{T}} \coloneqq \mathsf{ind}(\mathcal{T})$, and the set R composed of all role names occurring in $\mathcal{K}$. It remains to check whether $\mathcal{K}' \coloneqq (\mathcal{K} \cup \mathcal{F} \cup \{\top \sqsubseteq (\leq \mathrm{N}\ edge).\top)\}$ has an Ortiz-forest model. By Lemma 7.8 we see that $\mathcal{K}$ has an N-bounded quasi-forest model if and only if $\mathcal{K}'$ has an N-bounded quasi-forest model if and only if $\mathcal{K}'$ has an N-bounded Ortiz-forest model (dubbed *canonical* [Ort10, Def. 3.2.2]). We next follow the automata-based construction by Ortiz. First, we turn [Ort10, Prop. 3.1.5] $\mathcal{K}'$ into an equivalent $\mathcal{ZOIQ}$-concept C. Next, we construct [Ort10, Prop. 3.3.8, Cor. 3.3.9, L. 3.4.1] in time linear w.r.t. $|\mathrm{C}|$, a suitable tree automaton (called the *fully-enriched tree automaton*) $\mathcal{A}$ whose non-emptiness guarantees the existence of a (canonical) Ortiz-forest model for $\mathcal{K}'$. Based on the properties of the constructed automaton $\mathcal{A}$ [Ort10, L. 3.4.1], its non-emptiness of $\mathcal{A}$ can be then decided in exponential time [BLMV08, Cor. 4.3], providing the desired algorithm. The matching lower bound follows from $\mathcal{ALC}$ [BHLS17, Thm. 5.13]. $\square$

We next establish the *quasi-forest countermodel property* for $\mathcal{DL}$ being either $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, or $\mathcal{ZOI}$, *i.e.* the remaining part of Lemma 7.5. Namely, we want to show that for any $\mathcal{DL}$-KB $\mathcal{K}$ and a P2RPQ q we have that $\mathcal{K} \not\models q$ if and only if there exists a quasi-forest model of $\mathcal{K}$ with an exponential branching that violates q. This allow us to transfer previous results on Ortiz-forest models (the analogue of the first part of Lemma 7.5 but for Ortiz-forests) to our slightly more restricted setting.

Let us start from the case of $\mathcal{ZIQ}$, which follows from existing works [CEO14, Thm. 3.10].

Lemma 7.10 (Follows from works of Calvanese et. al) Lemma 7.5 holds for $\mathcal{ZIQ}$.

Proof. Let $\mathcal{K} \coloneqq (\mathcal{A}, \mathcal{T})$ be a $\mathcal{ZIQ}$-KB and q be a P2RPQ such that $\mathcal{K} \not\models q$ (we take $q \coloneqq \bot$ in case we are interested only in the satisfiability problem). As $\mathcal{K}$ is in $\mathcal{ZIQ}$, we have $\mathsf{ind}(\mathcal{T}) = \emptyset$. Moreover, we assume w.l.o.g. that both $\mathcal{K}$ and q do not use "special" role and concept names.

By the result of Calvanese et al. [CEO14, Thm. 3.10] there exists the so-called $k_{\mathcal{T},q}$-canonical model $\mathcal{I}$ of $\mathcal{K}$ that violates q, where $k_{\mathcal{T},q}$ is a number bounded exponentially w.r.t. joint sizes of q and $\mathcal{T}$ [CEO14, Def. 3.9]. W.l.o.g. we can assume that $\mathcal{I}$ interprets special roles and concepts as empty sets. A careful inspection [CEO14, Def. 3.7] reveals that $\mathcal{I}$ is simply an $\mathsf{ind}(\mathcal{A})$-Ortiz-forest that is $k_{\mathcal{T},q}$-bounded and that does not allow for "backlinks". Hence, by reinterpreting the concept name Root, and role names *edge*, *child*, *id* as given in Definition 7.1 we get the desired exponentially-branching $(\mathsf{ind}(\mathcal{A}), \mathsf{ind}(\mathcal{T}))$-quasi-forest countermodel for $\mathcal{K}$ and q. $\square$

To establish Lemma 7.5 for $\mathcal{ZOQ}$ and $\mathcal{ZOI}$, we first design a suitable notion of *unravelling*, similar to the one from Section 5.1. We then employ *pruning* to make sure that the branching of the resulting forest is "small". To achieve our goals, the following handy notion of Θ-paths in quasi-forests is crucial.

Definition 7.11 Let Θ be a subset of $\{\mathsf{I}, \mathcal{Q}, \mathcal{O}\}$, $\mathbf{N}_{\mathbf{I}}^{\mathcal{A}}$ and $\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}$ be finite sets of individual names, $\mathcal{I}$ be an interpretation, and ρ be a word in $(\Delta^{\mathcal{I}})^*$. We call ρ a **Θ-path** if all the conditions below hold.

- ρ is an undirected path in $\mathcal{I}$ if $\mathsf{I} \in \Theta$, and a directed path in $\mathcal{I}$ otherwise.
- ρ starts from some $(\mathbf{N}_{\mathbf{I}}^{\mathcal{A}} \cup \mathbf{N}_{\mathbf{I}}^{\mathcal{T}})$-named element.
- There is no index $i > 1$ for which ρ_i is $\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}$-named.
- There is no index i for which $\rho_i = \rho_{i+1}$.
- If $\{\mathcal{Q}, \mathsf{I}\} \subseteq \Theta$ then there is no index i for which $\rho_i = \rho_{i+2}$.

We employ Θ-paths as the domain of the forthcoming notion of unravelling. The idea behind the first item of Definition 7.11 is that directionality of paths depend on the presence of the inverse operator I in the underlying logic. The second item of Definition 7.11 is needed to manually assign roles between the named elements. The third item of Definition 7.11 guarantees that we do not copy nominals (if they are present). The fourth item of Definition 7.11 is useful for correct definition of self-loops. Finally, the last item of Definition 7.11 is needed in order to establish downward realisability of automata for the logic $\mathcal{ZOI}$ (note that this condition simply "clones" a parent of a node and "makes" such a copy a fresh child).

We are now ready to present the definition of our unravellings. For succinctness, we identify the names of logics $\mathcal{Z}\Theta$ with the letter $\mathcal{Z}$ concatenated with the letters from Θ, *e.g.* we identify $\mathcal{Z}\{\mathsf{I},\mathcal{O}\}$ with $\mathcal{ZOI}$.

Definition 7.12 Let $\mathbf{N_I^{\mathcal{A}}}$ and $\mathbf{N_I^{\mathcal{T}}}$ be finite sets of individual names, Θ be a subset of $\{\mathsf{I},\mathcal{O},\mathcal{Q}\}$, and $\mathcal{I}$ be an interpretation. The **$(\mathbf{N_I^{\mathcal{A}}},\mathbf{N_I^{\mathcal{T}}})$-$\mathcal{Z}\Theta$-unravelling** of $\mathcal{I}$ is the interpretation $\mathcal{J}$ defined as:

1. The domain $\Delta^{\mathcal{J}}$ of $\mathcal{J}$ is composed of all Θ-paths in $\mathcal{I}$.
2. For all individual names, if $\mathsf{a} \in (\mathbf{N_I^{\mathcal{A}}} \cup \mathbf{N_I^{\mathcal{T}}})$ then $\mathsf{a}^{\mathcal{J}} := \mathsf{a}^{\mathcal{I}}$. Otherwise $\mathsf{a}^{\mathcal{J}} := \mathsf{b}^{\mathcal{I}}$ for some $\mathsf{b} \in \mathbf{N_I}$.
3. We put $\mathrm{A}^{\mathcal{J}} := \{\rho \mid \mathsf{last}(\rho) \in \mathrm{A}^{\mathcal{I}}\}$ and $\mathrm{Root}^{\mathcal{J}} = \{\mathsf{a}^{\mathcal{J}} \mid \mathsf{a} \in \mathbf{N_I}\}$ for all names $\mathrm{A} \in \mathbf{N_C} \setminus \{\mathrm{Root}\}$.
4. We interpret role names $r \in \mathbf{N_R} \setminus \{\mathit{id}, \mathit{child}, \mathit{edge}\}$ as the intersection of $\Delta^{\mathcal{J}} \times \Delta^{\mathcal{J}}$ and the union of sets $\mathsf{R}^r_{\mathrm{Root}}$, $\mathsf{R}^r_{\mathsf{Self}}$, R^r, $\mathsf{R}^r_{\{\mathsf{I},\mathcal{O}\}}$, $\mathsf{R}^r_{\mathsf{I}}$, $\mathsf{R}^r_{\mathcal{O}}$ defined below.

$$\begin{aligned}
\mathsf{R}^r_{\mathrm{Root}} &:= & r^{\mathcal{I}} \cap (\mathrm{Root}^{\mathcal{J}} \times \mathrm{Root}^{\mathcal{J}}).\\
\mathsf{R}^r_{\mathsf{Self}} &:= & \Big\{(\rho,\rho) \mid (\mathsf{last}(\rho),\mathsf{last}(\rho)) \in r^{\mathcal{I}}\Big\}.\\
\mathsf{R}^r &:= & \Big\{(\rho,\rho\cdot \mathrm{d}) \mid (\mathsf{last}(\rho),\mathrm{d}) \in r^{\mathcal{I}}\Big\}.\\
\mathsf{R}^r_{\mathsf{I}} &:= & \Big\{(\rho\cdot\mathrm{d},\rho) \mid (\mathrm{d},\mathsf{last}(\rho)) \in r^{\mathcal{I}}, \mathsf{I} \in \Theta\Big\}.\\
\mathsf{R}^r_{\mathcal{O}} &:= & \Big\{(\rho,\mathsf{o}^{\mathcal{J}}) \mid (\mathsf{last}(\rho),\mathsf{o}^{\mathcal{I}}) \in r^{\mathcal{I}}, \mathsf{o} \in \mathbf{N_I^{\mathcal{T}}}, \mathcal{O} \in \Theta\Big\}.\\
\mathsf{R}^r_{\{\mathsf{I},\mathcal{O}\}} &:= & \{(\mathsf{o}^{\mathcal{J}},\rho) \mid (\mathsf{o}^{\mathcal{I}},\mathsf{last}(\rho)) \in r^{\mathcal{I}}, \mathsf{o} \in \mathbf{N_I^{\mathcal{T}}}, \{\mathsf{I},\mathcal{O}\} \subseteq \Theta\Big\}.
\end{aligned}$$

The interpretation of "special" role names as defined follows.

$$\mathit{id}^{\mathcal{J}} := \{(\rho,\rho) \mid \rho \in \Delta^{\mathcal{J}}\},\ \mathit{child}^{\mathcal{J}} := \{(\rho,\rho\cdot\mathrm{d}) \mid \rho,\rho\cdot\mathrm{d} \in \Delta^{\mathcal{J}}\},\ \mathit{edge}^{\mathcal{J}} := \textstyle\bigcup_{r \in \mathbf{N_R}\setminus\{\mathit{edge}\}} r^{\mathcal{J}} \cup (r^{-})^{\mathcal{J}}.$$

Intuitively: (i) $\mathsf{R}^r_{\mathrm{Root}}$ handles root-to-root connections, (ii) $\mathsf{R}^r_{\mathsf{Self}}$ handles self-loops, (iii) R^r handles parent-to-child roles, (iv) $\mathsf{R}^r_{\mathsf{I}}$ handles child-to-parent roles in the presence of inverses, (v) $\mathsf{R}^r_{\mathcal{O}}$ handles "backlinks" in the presence of nominals, and (vi) $\mathsf{R}^r_{\{\mathsf{I},\mathcal{O}\}}$ handles the inverses of "backlinks".

The following observation follows immediately from the above definition.

Fact 7.13. Let $\mathbf{N_I^{\mathcal{A}}}$ and $\mathbf{N_I^{\mathcal{T}}}$ be finite subsets of $\mathbf{N_I}$, and $\mathcal{I}$ be a proper $(\mathbf{N_I^{\mathcal{A}}} \cup \mathbf{N_I^{\mathcal{T}}})$-Ortiz-forest. Then for every $\Theta \subseteq \{\mathsf{I},\mathcal{O},\mathcal{Q}\}$ the $(\mathbf{N_I^{\mathcal{A}}},\mathbf{N_I^{\mathcal{T}}})$-$\mathcal{Z}\Theta$-unravelling $\mathcal{J}$ of $\mathcal{I}$ is an $(\mathbf{N_I^{\mathcal{A}}},\mathbf{N_I^{\mathcal{T}}})$-quasi-forest with last being a homomorphism from $\mathcal{J}$ to $\mathcal{I}$ (and in particular, for all P2RPQs q we have that $\mathcal{I} \not\models q$ implies $\mathcal{J} \not\models q$).

A bit more difficult task is to prove that the above unravellings preserve satisfaction of KBs written in the corresponding logics. Recall that we only consider TBoxes in the following Scott's normal form:

$$\mathrm{A} \equiv \{\mathsf{o}\},\quad \mathrm{A} \equiv \mathrm{B} \quad \mathrm{A} \equiv \neg\mathrm{B},\quad \mathrm{A} \equiv \mathrm{B} \sqcap \mathrm{B}',\quad \mathrm{A} \equiv \exists r.\mathsf{Self},\quad s = s',\quad \mathrm{A} \equiv (\geqslant n\ r).\top,\quad \mathrm{A} \equiv \exists \mathcal{A}_{\mathsf{q},\mathsf{q}'}.\top$$

where $\mathrm{A},\mathrm{B},\mathrm{B}'$ are concept names, r is a role name, s and s' are simple roles, and $\mathcal{A}$ is an NFA, and o is an individual name. In the case of $\mathcal{ZOQ}$, the inverse operator is disallowed in the definition of simple roles (and hence also from the alphabet of all NFAs appearing in the KB). In the case of $\mathcal{ZOI}$, we assume that $n = 1$ in all number restrictions $(\geqslant n\ r).\top$. We first show that our unravellings from Definition 7.12 preserve the satisfaction of knowledge-bases written in $\mathcal{ZOQ}$ and $\mathcal{ZOI}$. Indeed:

Lemma 7.14 Let $\mathcal{K} := (\mathcal{A},\mathcal{T})$ be a $\mathcal{ZOQ}$-KB in Scott's normal form, $\mathcal{I}$ be a proper Ortiz-forest model of $\mathcal{K}$, and $\mathcal{J}$ be the $(\mathsf{ind}(\mathcal{A}),\mathsf{ind}(\mathcal{T}))$-$\mathcal{ZOQ}$-unravelling of $\mathcal{I}$. Then $\mathcal{J}$ is a quasi-forest model of $\mathcal{K}$ that downward realises all NFAs from $\mathcal{K}$.

Proof sketch. We first observe, by the second item of Definition 7.12, that for all elements d from $\mathcal{J}$ and concept names $\mathrm{A} \in \mathbf{N_C} \setminus \{\mathrm{Root}\}$ we have that:

$$(\heartsuit)\colon \mathrm{d} \in \mathrm{A}^{\mathcal{J}} \text{ if and only if } \mathsf{last}(\mathrm{d}) \in \mathrm{A}^{\mathcal{I}}.$$

From ($\heartsuit$) it is immediate to see that $\mathcal{J}$ satisfies all the GCIs φ from $\mathcal{K}$ of the form $\varphi := \mathrm{A} \equiv \neg\mathrm{B}$, $\varphi := \mathrm{A} \equiv \mathrm{B}$, or $\varphi := \mathrm{A} \equiv \mathrm{B}\sqcap\mathrm{B}'$. Moreover, Equation ($\heartsuit$) simplifies the proofs of the satisfaction of GCIs of the form $\mathrm{A} \equiv \mathrm{C}$, reducing them to the proof of the fact that for all elements d of $\mathcal{J}$ we have that $\mathrm{d} \in \mathrm{C}^{\mathcal{J}}$ if and only if $\mathsf{last}(\mathrm{d}) \in \mathrm{C}^{\mathcal{I}}$. Indeed, assuming the above equivalence holds we have: $\mathrm{d} \in \mathrm{C}^{\mathcal{J}}$ if and only if (by the above assumption) $\mathsf{last}(\mathrm{d}) \in \mathrm{C}^{\mathcal{I}}$ if and only if (by the modelhood of $\mathcal{I}$) $\mathsf{last}(\mathrm{d}) \in \mathrm{A}^{\mathcal{I}}$ if and only if (by ($\heartsuit$)) $\mathrm{d} \in \mathrm{A}^{\mathcal{J}}$. This yields $\mathcal{J} \models (\mathrm{A} \equiv \mathrm{C})$.

Second, we have that for all pairs $(\mathrm{d}, \mathrm{e}) \in edge^{\mathcal{J}}$ and role names $r \in \mathbf{N_C} \setminus \{id, edge, child\}$:

$$(\clubsuit)\colon\ (\mathrm{d}, \mathrm{e}) \in r^{\mathcal{J}} \text{ if and only if } (\mathsf{last}(\mathrm{d}), \mathsf{last}(\mathrm{e})) \in r^{\mathcal{I}}.$$

Note that if $(\mathrm{d}, \mathrm{e}) \in edge^{\mathcal{J}}$ then, due to the way we interpret role names in $\mathcal{ZOQ}$-unravellings, there are only four possible options: (i) both d, e are roots of $\mathcal{J}$, or (ii) d = e, or (iii) e is a child of d, or (iv) e is a nominal. In each of these cases, we conclude the Equation ($\clubsuit$) by the definitions of sets $\mathsf{R}^r_{\mathrm{Root}}$, $\mathsf{R}^r_{\mathsf{Self}}$, R^r, and $\mathsf{R}^r_{\mathcal{O}}$. Based on Equation ($\clubsuit$) we conclude the satisfaction of the GCIs of the form $s = s'$ (for simple roles s, s') by $\mathcal{J}$. Similarly, by employing the definition of the set $\mathsf{R}^r_{\mathsf{Self}}$, we conclude the satisfaction of GCIs of the form $\mathrm{A} \equiv \exists r.\mathsf{Self}$ by $\mathcal{J}$. The satisfaction of GCIs of the form $\mathrm{A} \equiv \{\mathsf{o}\}$ follows from Equation ($\heartsuit$) and the fact that $\mathsf{o}^{\mathcal{J}}$ is defined as $\mathsf{o}^{\mathcal{I}}$ for all nominals $\mathsf{o} \in \mathbf{N}_{\mathbf{I}}^{\mathcal{T}}$. Finally, a combination of Equations ($\heartsuit$), ($\clubsuit$), and the definition of $\mathsf{R}^r_{\mathrm{Root}}$ implies the satisfaction of the ABox $\mathcal{A}$ by $\mathcal{J}$.

For the remaining two "shapes" of GCIs, namely $\mathrm{A} \equiv (\geqslant n\ r).\top$ and $\mathrm{A} \equiv \exists\mathcal{A}_{\mathsf{q},\mathsf{q}'}.\top$ it suffices to show that d in $\mathcal{J}$ belongs to the interpretation of concept on the right hand size of $\equiv$, if and only if, $\mathsf{last}(\mathrm{d})$ belongs to such a concept in $\mathcal{I}$. Take any element $\mathrm{d} \in \Delta^{\mathcal{J}}$. We have that:

- Satisfaction of automata constraints.
  Suppose $\mathrm{d} \in (\exists\mathcal{A}_{\mathsf{q},\mathsf{q}'}.\top)^{\mathcal{J}}$, as witnesses by an $\mathcal{A}_{\mathsf{q},\mathsf{q}'}$-path $\rho := \rho_1 \ldots \rho_n$ with $\rho_1 := \mathrm{d}$. Then it can be readily verified, by employing Equations ($\heartsuit$) and ($\clubsuit$), that $\mathsf{last}[\rho] := \mathsf{last}(\rho_1) \ldots \mathsf{last}(\rho_n)$ is an $\mathcal{A}_{\mathsf{q},\mathsf{q}'}$-path in $\mathcal{I}$. This yields that $\mathsf{last}(\mathrm{d}) \in (\exists\mathcal{A}_{\mathsf{q},\mathsf{q}'}.\top)^{\mathcal{I}}$. For the opposite implication, suppose that $\mathsf{last}(\mathrm{d}) \in (\exists\mathcal{A}_{\mathsf{q},\mathsf{q}'}.\top)^{\mathcal{I}}$, as witnessed by an $\mathcal{A}_{\mathsf{q},\mathsf{q}'}$-path $\rho := \rho_1 \ldots \rho_n$ with $\rho_1 := \mathsf{last}(\mathrm{d})$. We define the word $\varrho := \varrho_1 \ldots \varrho_n$, which we call the *lift* of ρ, inductively as follows. For i=1 we put $\varrho_1 := \mathrm{d}$. For all $i > 1$ we put $\varrho_{i+1} := \rho_{i+1}$ if ρ_{i+1} is a nominal or if both ρ_{i+1} ρ_i are roots of $\mathcal{I}$, and $\varrho_{i+1} := \varrho_i{\cdot}\rho_{i+1}$ otherwise. Once more, it can be readily verified that ϱ is an $\mathcal{A}_{\mathsf{q},\mathsf{q}'}$-path in $\mathcal{J}$ that starts from d. Hence, $\mathrm{d} \in (\exists\mathcal{A}_{\mathsf{q},\mathsf{q}'}.\top)^{\mathcal{J}}$. This also establishes that the NFA $\mathcal{A}_{\mathsf{q},\mathsf{q}'}$ is downward realisable in $\mathcal{J}$.
- Satisfaction of number restrictions.
  It suffices to show that for any role name r that is not "special" we have that d in $\mathcal{J}$ and $\mathsf{last}(\mathrm{d})$ in $\mathcal{I}$ have equal number of r-successors. First, by design of $\mathcal{J}$, note that pairwise-different nominals from $\mathcal{I}$ become unique pairwise-different elements in $\mathcal{J}$. Thus, by the definitions of $\mathsf{R}^r_{\mathrm{Root}}$ and $\mathsf{R}^r_{\mathcal{O}}$ we have that d has precisely the same number of nominal r-successors as $\mathsf{last}(\mathrm{d})$ in $\mathcal{I}$. Moreover, d carries an r-self-loop if and only if $\mathsf{last}(\mathrm{d})$ does, as witnessed by the definition of $\mathsf{R}^r_{\mathsf{Self}}$. Finally, observe that all remaining r-successors of d are present in $\mathcal{J}$ due to the definition of R^r. Such r-successors have the form $\mathrm{d}{\cdot}\mathrm{e}$ for some $\mathrm{e} \in \Delta^{\mathcal{I}}$ and $\mathrm{d}{\cdot}\mathrm{e}$ is a directed path in $\mathcal{I}$. As every sequence in $\Delta^{\mathcal{J}}$ has a unique end, every such r-successor $\mathrm{d}{\cdot}\mathrm{e}$ yields a unique element $\mathsf{last}(\mathrm{d}{\cdot}\mathrm{e})$ equal to e that is an r-successor of $\mathsf{last}(\mathrm{d})$. Conversely, for different r-successors e and e′ of $\mathsf{last}(\mathrm{d})$ in $\mathcal{I}$ there are unique r-successors $\mathrm{d}{\cdot}\mathrm{e}$ and $\mathrm{d}{\cdot}\mathrm{e}'$ of d in $\mathcal{J}$. Thus, the total number of non-nominal r-successors of d and $\mathsf{last}(\mathrm{d})$ is indeed equal. This concludes the proof.

Hence, the quasi-forest model $\mathcal{J}$ is as desired. □

The analogue of the above lemma for $\mathcal{ZOI}$ can be established in a nearly identical way.

Lemma 7.15 Let $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ be a $\mathcal{ZOI}$-KB in Scott's normal form, $\mathcal{I}$ be a proper Ortiz-forest model of $\mathcal{K}$, and $\mathcal{J}$ be the $(\mathsf{ind}(\mathcal{A}), \mathsf{ind}(\mathcal{T}))$-$\mathcal{ZOI}$-unravelling of $\mathcal{I}$. Then $\mathcal{J}$ is an quasi-forest model of $\mathcal{K}$ that downward realises all NFAs from $\mathcal{K}$.

Proof idea. The proof is nearly identical to the proof of Lemma 7.14. The only difference is that to argue that Equation (♣) holds, we additionally rely on the sets $\mathsf{R}^{r}_{\mathsf{I}}$, and $\mathsf{R}^{r}_{\{\mathsf{I},\mathcal{O}\}}$. □

We remark that unravellings from Definition 7.12 in general do not preserve $\mathcal{ZOIQ}$-KBs that are not in $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, or $\mathcal{ZOI}$. The main culprit is the presence of "anonymous nominals", *i.e.* elements that are not nominals but can be uniquely identified by means of GCIs in $\mathcal{ZOIQ}$.

Observe that Lemma 7.14 and Lemma 7.15 do not provide information on the degree of the resulting unravelled model. This is intentional, as degrees of named elements are rarely preserved by the unravelling.

Example 7.16. Let $\mathcal{K} \coloneqq (\{\mathrm{A}(\mathsf{a})\}, \emptyset)$ be a $\mathcal{Z}$-KB, and consider any Ortiz-forest model $\mathcal{I}$ of $\mathcal{K}$ with the domain $\Delta^{\mathcal{I}} \coloneqq \{0^i \mid i \in \mathbb{N}\}$. Suppose that $\mathcal{I}$ interprets a role name r as $r^{\mathcal{I}} \coloneqq \{(\varepsilon, 0^i) \mid i \in \mathbb{N}\}$, *i.e.* every element has an r-backlink to the root of $\mathcal{I}$. As a is not a nominal, such backlinks will not be preserved in any unravelling $\mathcal{J}$ of $\mathcal{I}$. This results in the interpretation of a having infinitely many r-successors in $\mathcal{J}$.

On the other hand, consider a $\mathcal{ZOIQ}$-KB $\mathcal{K} \coloneqq (\mathcal{A}, \mathcal{T})$, an Ortiz-forest model $\mathcal{I}$ of $\mathcal{K}$, and its $\mathcal{ZOIQ}$-unravelling $\mathcal{J}$. By analysing Lemma 7.14 and the notion of paths from Definition 7.11 we see that for any element d with $\mathsf{last}(\mathrm{d})$ not being a non-nominal root of $\mathcal{I}$ (*i.e.* the root that is not $\mathsf{ind}(\mathcal{T})$-named), the degree of d is bounded by the degree of $\mathsf{last}(\mathrm{d})$ plus the size of $\mathsf{ind}(\mathcal{A})$. Hence, if the branching of $\mathcal{I}$ is bounded exponentially w.r.t. $|\mathcal{K}|$, then so is the degree of every node of $\mathcal{J}$, possibly with the exception of nodes of $\mathcal{J}$, for which underlying sequence ends on a non-nominal root of $\mathcal{I}$. In order to reduce the branching of $\mathcal{J}$ to some number exponential w.r.t. $|\mathcal{K}|$, we design a suitable notion of *pruning*.

We start by selecting certain successors of non-nominal roots that are intended to "survive". Intuitively, we select these elements that are required to fulfil automata constraints and number restrictions.

Definition 7.17 Let $\mathcal{K} \coloneqq (\mathcal{A}, \mathcal{T})$ be a $\mathcal{ZOIQ}$-KB in Scott's normal form with N being the **counting bound of $\mathcal{K}$**, *i.e.* maximal among numbers appearing in number restrictions (or 1 if $\mathcal{K}$ has no number restrictions). Let $\mathcal{I}$ be an Ortiz-forest model of $\mathcal{K}$.

- For every GCI $\mathrm{A} \equiv \exists\mathcal{A}_{\mathsf{q},\mathsf{q}'}.\top$ from $\mathcal{K}$ and a root $\mathrm{d} \in \mathrm{A}^{\mathcal{I}}$, the **optimal $(\mathcal{A}_{\mathsf{q},\mathsf{q}'}, \mathrm{d})$-path** is the alphabetically-minimal $\mathcal{A}_{\mathsf{q},\mathsf{q}'}$-path starting from d with minimal number of roots of $\mathcal{I}$. Given such a path ρ, its member e is **important** if ρ has the form $\rho'{\cdot}\mathrm{e}'{\cdot}\mathrm{e}{\cdot}\rho''$ for some root e′.
- For every GCI $\mathrm{A} \equiv (\geqslant n\ r).\top$ from $\mathcal{K}$ and a root $\mathrm{d} \in \mathrm{A}^{\mathcal{I}}$, we call its r-successors **important** if they are among N alphabetically-minimal r-successors of d.

The set $\mathrm{Imp}_{\mathcal{K}}(\mathcal{I})$ of **$\mathcal{K}$-important elements from $\mathcal{I}$** is composed of (i) all important elements from optimal $(\mathcal{A}_{\mathsf{q},\mathsf{q}'}, \mathrm{d})$-paths over all roots $\mathrm{d} \in \mathrm{A}^{\mathcal{I}}$ of $\mathcal{I}$ and GCIs $\mathrm{A} \equiv \exists\mathcal{A}_{\mathsf{q},\mathsf{q}'}.\top$ of $\mathcal{K}$, and (ii) all important r-successors of all roots and GCIs $\mathrm{A} \equiv (\geqslant n\ r).\top$ from $\mathcal{K}$.

It is a relatively straightforward exercise to employ an analogue of the well-known pumping lemma from automata theory to establish the following lemma.

Lemma 7.18 Let $\mathcal{I}$, $\mathcal{K}$, N, and $\mathrm{Imp}_{\mathcal{K}}(\mathcal{I})$ be as in Definition 7.17. Then the size of the set $\mathrm{Imp}_{\mathcal{K}}(\mathcal{I})$ of $\mathcal{K}$-important elements from $\mathcal{I}$ can be bounded by $|\mathcal{K}|^4 + |\mathcal{K}|^2 \cdot 2^{|\mathcal{K}|}$.

Proof. In what follows we will separately calculate the number of important elements that are due to the GCIs involving counting and the GCIs involving automata. For the first kind of elements, we see that there are at most $|\mathcal{K}|$ such GCIs, and for each of at most $|\mathcal{K}|$ roots of $\mathcal{I}$ we selected at most N elements. This yields the upper bound $|\mathcal{K}|^2 \cdot 2^{|\mathcal{K}|}$ (note that N is encoded in binary). For the remaining important elements, we deal with them as follows. As there are at most $|\mathcal{K}|$ GCIs in $\mathcal{K}$, and at most $|\mathcal{K}|$ roots in $\mathcal{K}$, it suffices to show that for a fixed GCI $\mathrm{A} \equiv \exists\mathcal{A}_{\mathsf{q},\mathsf{q}'}.\top$ from $\mathcal{K}$, and a fixed root $\mathrm{d} \in \mathrm{A}^{\mathcal{I}}$, we have that the total number of important elements in the optimal $(\mathcal{A}_{\mathsf{q},\mathsf{q}'}, \mathrm{d})$-path ρ starting from d is bounded by $|\mathcal{K}|^2$. By following the definition of important elements, it suffices to show that such a ρ contains at most $|\mathcal{K}|$ many occurrences of each of the $|\mathcal{K}|$ roots d of $\mathcal{I}$. This follows from the pumping lemma. Indeed, towards a contradiction suppose that the total number of occurrences of d in ρ is greater than the number of states of $\mathcal{A}$. Then, by the pigeonhole principle, there exists a state q'' and a decomposition $\rho' {\cdot} \mathrm{d} {\cdot} \rho'' {\cdot} \mathrm{d} {\cdot} \rho'''$ for which $\rho' {\cdot} \mathrm{d}$ and $\rho' {\cdot} \mathrm{d} {\cdot} \rho'' {\cdot} \mathrm{d}$ both realise $\mathcal{A}_{\mathsf{q},\mathsf{q}''}$. But then the path $\rho' {\cdot} \mathrm{d} \rho'''$ has fewer occurrences of roots than $\rho' {\cdot} \mathrm{d} {\cdot} \rho'' {\cdot} \mathrm{d} {\cdot} \rho'''$ while it also realises the NFA $\mathcal{A}_{\mathsf{q},\mathsf{q}'}$. A contradiction with ρ being "optimal". □

We can now finally present the desired definition of *pruning*, defined below.

Definition 7.19 Let $\mathcal{DL} \in \{\mathcal{ZOI}, \mathcal{ZOQ}\}$ be a logic, $\mathcal{K}$ be a $\mathcal{DL}$-KB in Scott's normal form, $\mathcal{I}$ be a proper Ortiz-forest model of $\mathcal{K}$, and $\mathcal{J}$ be the $(\mathsf{ind}(\mathcal{A}), \mathsf{ind}(\mathcal{T}))$-$\mathcal{DL}$-unravelling of $\mathcal{I}$. By the **pruning $\mathcal{J}'$ of $\mathcal{J}$** we mean the substructure of $\mathcal{J}$ induced by all the sequences of $\Delta^{\mathcal{J}}$ that does not contain a subword in the set $\mathbb{N} \cdot (\Delta^{\mathcal{I}} \setminus \mathrm{Imp}_{\mathcal{K}}(\mathcal{I}))$.

The following observation follows immediately from the above definition.

Observation 7.20. For $\mathcal{DL}$, $\mathcal{K}$, $\mathcal{I}$, $\mathcal{J}$, $\mathcal{J}'$ as in Definition 7.19 we have that $\mathcal{J}'$ is a $(\mathsf{ind}(\mathcal{A}), \mathsf{ind}(\mathcal{T}))$-quasi-forest with last being a homomorphism from $\mathcal{J}'$ to $\mathcal{I}$ (and in particular, for all P2RPQs q we have that $\mathcal{I} \not\models q$ implies $\mathcal{J} \not\models q$). Moreover, if the degree of every node in $\mathcal{I}$ is bounded by some number N exponential w.r.t. $\mathcal{K}$ then the degree of every node in $\mathcal{J}$ is also bounded exponentially w.r.t. $|\mathcal{K}|$.

Proof sketch. We mostly rely on Fact 7.13. The first statement follows from the fact that $\mathcal{J}'$ is the restriction of a $(\mathsf{ind}(\mathcal{A}), \mathsf{ind}(\mathcal{T}))$-quasi-forest $\mathcal{J}$ to some prefix-closed set with some suffix-closed set removed. For the second statement we employ the fact that $\mathsf{last} \colon \Delta^{\mathcal{J}} \to \Delta^{\mathcal{I}}$ is a homomorphism, and so is its restriction to the domain of $\mathcal{J}'$. Finally, for the third statement we see that (i) if $\mathsf{last}(\mathrm{d})$ is not a non-nominal root with its degree bounded by N then the degree of d is bounded by $\mathrm{N}{+}|\mathsf{ind}(\mathcal{A})|$, and (ii) if $\mathsf{last}(\mathrm{d})$ is a non-nominal root then by design the set of its children is not greater than $|\mathrm{Imp}_{\mathcal{K}}(\mathcal{I})|$, which is exponential w.r.t. $|\mathcal{K}|$ by Lemma 7.18). In both cases, the degree of every node node in $\mathcal{J}'$ is clearly exponential w.r.t. $|\mathcal{K}|$. □

Note that the refinement of the above argument yields the dependence only on the TBox, not on the whole KB. This follows from the downward realisability of NFA, and one can find an upgraded proof in the Appendix of our IJCAI submission [Bed24a] (available on ArXiV). As the final result of this section we show that pruning of unravellings preserve modelhoods of KBs.

Lemma 7.21 For $\mathcal{DL}, \mathcal{K}, \mathcal{I}, \mathcal{J}, \mathcal{J}'$ as in Definition 7.19 we have that $\mathcal{J}' \models \mathcal{K}$, and that $\mathcal{J}'$ downward realises all NFAs from $\mathcal{K}$.

Proof sketch. If $\mathcal{I} \models \mathcal{K}$ then by Lemma 7.14 and Lemma 7.15 we have that $\mathcal{J} \models \mathcal{K}$. As $\mathcal{K}$ is in Scott's normal form, we see that all its GCIs except the ones of the form $\mathrm{A} \equiv ({\geqslant} n\ r).\top$ and $\mathrm{A} \equiv \exists \mathcal{A}_{\mathsf{q},\mathsf{q}'}.\top$ can be presented as universal first-order formulae, and thus are preserved under taking substructures. We establish the satisfaction of remaining GCIs below.

- Suppose that $\mathcal{J} \models \mathrm{A} \equiv ({\geqslant} n\ r).\top$. As $\mathcal{J}'$ is a substructure of $\mathcal{J}$, for all elements $\mathrm{d} \in \Delta^{\mathcal{J}'}$ we have d satisfies A in $\mathcal{J}'$ if and only if it satisfies it in $\mathcal{J}$. Moreover, if $\mathrm{d} \in \Delta^{\mathcal{J}'}$ does not satisfy $({\geqslant} n\ r).\top$ in $\mathcal{J}$ then it clearly does not satisfy this concept in any substructure of $\mathcal{J}$. Hence, it suffices to show that if $\mathrm{d} \in \Delta^{\mathcal{J}'}$ satisfies $({\geqslant} n\ r).\top$ in $\mathcal{J}$ then it does satisfy it in $\mathcal{J}'$. If $\mathsf{last}(\mathrm{d})$ is not a non-nominal root, then the set of its r-successors is not affected by the pruning, and thus the satisfaction of $({\geqslant} n\ r).\top$ is preserved. If $\mathsf{last}(\mathrm{d})$ is a non-nominal root, then in the construction of $\mathrm{Imp}_{\mathcal{K}}(\mathcal{I})$ we dubbed at least n r-successors of $\mathsf{last}(\mathrm{d})$ in $\mathcal{I}$ important. Let S be the set of such important r-successors of $\mathsf{last}(\mathrm{d})$ in $\mathcal{I}$. We define the injection $\mathfrak{f}$ from S to the domain of $\mathcal{J}'$ as follows: (i) $\mathfrak{f}(\mathsf{last}(\mathrm{d})) = \mathrm{d}$ (in case $\mathsf{last}(\mathrm{d})$ belongs to S), (ii) $\mathfrak{f}(m) = m$ if m is a nominal root, and (iii) $\mathfrak{f}(\mathrm{e}) = \mathrm{d}{\cdot}\mathrm{e}$ otherwise. Hence, d in $\mathcal{J}'$ has at least n r-successors, as desired.
- Suppose that $\mathcal{J} \models \mathrm{A} \equiv \exists \mathcal{A}_{\mathsf{q},\mathsf{q}'}.\top$. Reasoning analogously to the previous case, it suffices to show that for elements d in $\mathcal{J}'$ that satisfy $\exists \mathcal{A}_{\mathsf{q},\mathsf{q}'}.\top$ in $\mathcal{J}$, still satisfy it in $\mathcal{J}'$. We claim

that it suffices to focus on elements d with $\mathsf{last}(\mathrm{d})$ being a root of $\mathcal{I}$. Indeed, suppose that d is an element with $\mathsf{last}(\mathrm{d}) \not\in \mathbb{N}$, and consider a downward (or nominal downward) $\mathcal{A}_{\mathsf{q},\mathsf{q}'}$-path $\rho \coloneqq \rho_1 \dots \rho_n$ starting from d in $\mathcal{J}$ (that exists by the satisfaction of $\exists\mathcal{A}_{\mathsf{q},\mathsf{q}'}.\top$). If ρ does not contain elements ending on a root of $\mathcal{I}$, then ρ is fully contained in $\mathcal{J}'$, witnessing the satisfaction of $\exists\mathcal{A}_{\mathsf{q},\mathsf{q}'}.\top$. Otherwise, let $\rho \coloneqq \rho' {\cdot} \mathrm{e} {\cdot} \rho''$, where e is the first element in ρ satisfying $\mathsf{last}(\rho) \in \mathbb{N}$. Note that by the same reason as above, $\rho'\mathrm{e}$ is fully contained in $\mathcal{J}'$. Then, by compositionality of NFAs, there exists a state q'' of $\mathcal{A}$ such that $\rho'{\cdot}\mathrm{e}$ realise $\mathcal{A}_{\mathsf{q},\mathsf{q}''}$ and $\mathrm{e}{\cdot}\rho''$ realise $\mathcal{A}_{\mathsf{q}'',\mathsf{q}'}$. Hence, it suffices to establish that e satisfies $\exists\mathcal{A}_{\mathsf{q}'',\mathsf{q}'}.\top$, because then any downward or nominal downward $\mathcal{A}_{\mathsf{q}'',\mathsf{q}'}$-path ϱ starting from the element e makes $\rho'\varrho$ a downward or nominal downward $\mathcal{A}_{\mathsf{q}',\mathsf{q}'}$-path.

Let d be then the element with $\mathsf{last}(\mathrm{d}) \in \mathbb{N}$ that satisfies $\exists\mathcal{A}_{\mathsf{q},\mathsf{q}'}.\top$. Then $\mathsf{last}(\mathrm{d})$ is not only a root of $\mathcal{I}$ but it also satisfies $\exists\mathcal{A}_{\mathsf{q},\mathsf{q}'}.\top$. Hence, let $\rho \coloneqq \rho_1 \dots \rho_n$ be the optimal $(\mathcal{A}_{\mathsf{q},\mathsf{q}'}, \mathsf{last}(\mathrm{d}))$-path that witnesses the satisfaction of $\exists\mathcal{A}_{\mathsf{q},\mathsf{q}'}.\top$ by $\mathsf{last}(\mathrm{d})$. Similarly to the proof of Lemma 7.14, we employ the lift of ρ, namely the word $\varrho \coloneqq \varrho_1 \dots \varrho_n$, defined inductively as follows. For $i{=}1$ we put $\varrho_1 \coloneqq \mathrm{d}$. For all $i > 1$ we put $\varrho_{i+1} \coloneqq \rho_{i+1}$ if ρ_{i+1} is a nominal or if both ρ_{i+1} ρ_i are roots of $\mathcal{I}$, and $\varrho_{i+1} \coloneqq \varrho_i {\cdot} \rho_{i+1}$ otherwise. Note that ϱ_{i+1} for ρ_i being a root of $\mathcal{I}$ is guaranteed to exist by the design of $\mathrm{Imp}_{\mathcal{K}}(\mathcal{I})$ (otherwise ϱ_{i+1} belongs to the domain of $\mathcal{J}'$ by the construction of $\mathcal{J}$). We readily verify that ϱ is a downward (or nominal downward) $\mathcal{A}_{\mathsf{q},\mathsf{q}'}$-path starting from d, and thus d satisfies $\exists\mathcal{A}_{\mathsf{q},\mathsf{q}'}.\top$.

This completes the proof that $\mathcal{J}' \models \mathcal{K}$. ☐

We can now sum up all the work of this section to establish the remaining part of Lemma 7.5.

Lemma 7.22 Lemma 7.5 holds for $\mathcal{ZOI}$ and $\mathcal{ZOQ}$.

Proof. Let $\mathcal{DL} \in \{\mathcal{ZOQ}, \mathcal{ZOI}\}$ be a description logic, $\mathcal{K}$ be a $\mathcal{DL}$-KB, and q be a P2RPQ such that $\mathcal{K} \not\models q$ (we take $q \coloneqq \bot$ in case we are interested only in the satisfiability problem). The results of Ortiz [Ort10, Prop. 3.3.8] yield an Ortiz-forest model $\mathcal{I}$ of $\mathcal{K}$ that violates q and has the branching bounded exponentially w.r.t. $|\mathcal{T}|$. W.l.o.g. we may assume that $\mathcal{I}$ is proper, *i.e.* interprets special roles and concepts as empty sets. We apply $\mathcal{DL}$-unravellings (Definition 7.12) in order to obtain $\mathcal{J}$ from $\mathcal{I}$. By Fact 7.13 we know that $\mathcal{J} \not\models q$ and that $\mathcal{J}$ is an $(\mathsf{ind}(\mathcal{A}), \mathsf{ind}(\mathcal{T}))$-quasi-forest. By Lemma 7.15 and Lemma 7.14 we know that $\mathcal{J} \models \mathcal{K}$ and that it downward realises all NFAs from $\mathcal{K}$. We then apply Definition 7.19 to construct the pruning $\mathcal{J}'$ of $\mathcal{J}$. By Observation 7.20 we have that $\mathcal{J}'$ violates q and it is an $(\mathsf{ind}(\mathcal{A}), \mathsf{ind}(\mathcal{T}))$-quasi-forest. By Lemma 7.21 we have that $\mathcal{J}'$ satisfies $\mathcal{K}$ and downward realises all NFAs from $\mathcal{K}$. Hence, $\mathcal{J}'$ is the desired quasi-forest (counter)model. ☐

7.3 Basic Paths in Quasi-Forests

The main goal of this section is to provide a characterisation of how paths in quasi-forests look like and how they can be decomposed into interesting pieces.

Definition 7.23 A path ρ in $\mathcal{I}$ is **nameless** if it does not contain any *named* elements (*i.e.* no element of ρ has the form $\mathsf{a}^{\mathcal{I}}$ for some $\mathsf{a} \in \mathbf{N_I}$). Similarly, ρ is called an **a-subtree path** if all the members of ρ are descendants of $\mathsf{a}^{\mathcal{I}}$.

It is easy to see that in quasi-forests, nameless paths are precisely subtree paths. Indeed:

Observation 7.24. Let $\mathcal{I}$ be an $(\mathbf{N_I^{\mathcal{A}}}, \mathbf{N_I^{\mathcal{T}}})$-quasi-forest and ρ be a path in $\mathcal{I}$. Then ρ is nameless if and only if there is a name $\mathsf{a}^{\mathcal{I}}$ for which ρ is an a-subtree path.

Proof. Recall that the roots of $\mathcal{I}$ are precisely the named elements of $\mathcal{I}$. Thus if ρ is an a-subtree path then it is clearly nameless. For the other direction assume that $\rho := \rho_1 \dots \rho_k$ is nameless. By the above observation, ρ_1 is not a root of $\mathcal{I}$, and hence there exists a name $\mathsf{a} \in (\mathbf{N_I^{\mathcal{A}}} \cup \mathbf{N_I^{\mathcal{T}}})$ for which ρ_1 is a descendant of $\mathsf{a}^{\mathcal{I}}$. We will show inductively that for all $1 \leq i \leq k$ the element ρ_i is a descendant of $\mathsf{a}^{\mathcal{I}}$. The base case follows by our choice of a. Now suppose that ρ_i is a descendant of $\mathsf{a}^{\mathcal{I}}$, *i.e.* there exists a (possibly empty) word $\bar{\rho}$ and a number n such that ρ_i has the form $\mathsf{a}^{\mathcal{I}} n \bar{\rho}$. Let us show that ρ_{i+1} is also a descendant of $\mathsf{a}^{\mathcal{I}}$. As ρ is a path in $\mathcal{I}$, there exists a (possibly inverted) role name r for which $(\rho_i, \rho_{i+1}) \in r^{\mathcal{I}}$ holds. Thus, by definition of a quasi-forest we have the following options: (i) $\rho_i = \rho_{i+1}$ or there is an integer m for which either (ii) $\rho_{i+1} = \rho_i m$ or (iii) $\rho_i = \rho_{i+1} m$. In the first two cases ρ_{i+1} is obviously a descendant of $\mathsf{a}^{\mathcal{I}}$. In the last case, note that $|\rho_{i+1}| \geq 2$ (otherwise ρ_{i+1} would be named) and is a prefix of ρ_i. Hence, ρ_{i+1} is a descendant of $\mathsf{a}^{\mathcal{I}}$. This concludes the induction, and finishes the proof. □

In the following definition we introduce a bunch of different categories of paths present in quasi-forests.

Definition 7.25 Let $\mathcal{I}$ be an $(\mathbf{N_I^{\mathcal{A}}}, \mathbf{N_I^{\mathcal{T}}})$-quasi-forest, $\mathsf{a}, \mathsf{b} \in (\mathbf{N_I^{\mathcal{A}}} \cup \mathbf{N_I^{\mathcal{T}}})$, $\mathsf{o}, \ddot{\mathsf{o}} \in \mathbf{N_I^{\mathcal{T}}}$, and ρ be a path in $\mathcal{I}$. We call ρ:

- **(a, b)-direct** if $\rho = \mathsf{a}^{\mathcal{I}} \cdot \mathsf{b}^{\mathcal{I}}$.
- **a-inner** if $\rho = \mathsf{a}^{\mathcal{I}} \cdot \bar{\rho}$ for some a-subtree path $\bar{\rho}$.
- **a-roundtrip** if $\rho = \mathsf{a}^{\mathcal{I}} \cdot \bar{\rho} \cdot \mathsf{a}^{\mathcal{I}}$ for some a-subtree path $\bar{\rho}$.
- **(a, o)-inout** if $\rho = \bar{\rho} \cdot \mathsf{o}^{\mathcal{I}}$ for some a-inner path $\bar{\rho}$.
- **(a, o)-outin** if the reverse of ρ is (a, o)-inout.
- **(a, o)-inner** if $\rho = \mathsf{o}^{\mathcal{I}} \cdot \bar{\rho}$ for some a-subtree path $\bar{\rho}$.
- **$(\mathsf{a}, \mathsf{o}, \ddot{\mathsf{o}})$-bypass** if $\rho = \mathsf{o}^{\mathcal{I}} \cdot \bar{\rho} \cdot \ddot{\mathsf{o}}^{\mathcal{I}}$ for an a-subtree path $\bar{\rho}$.

If ρ falls into one of the above seven categories, we call ρ **basic**. Paths $\rho := \bar{\rho} \cdot \mathrm{d}$ for a nameless $\bar{\rho}$ and a root d are called **outer**. Finally, ρ is **decomposable** whenever there exists a growing sequence of indices $i_1 < i_2 < \dots < i_k$ with $i_1{=}1$ and $i_k{=}|\rho|$ such that for all $j < k$ the path $\rho_{i_j} \dots \rho_{i_{(j+1)}}$ is basic.

The categories of paths introduced in Definition 7.25 are visualised in Figure 7.1 below.

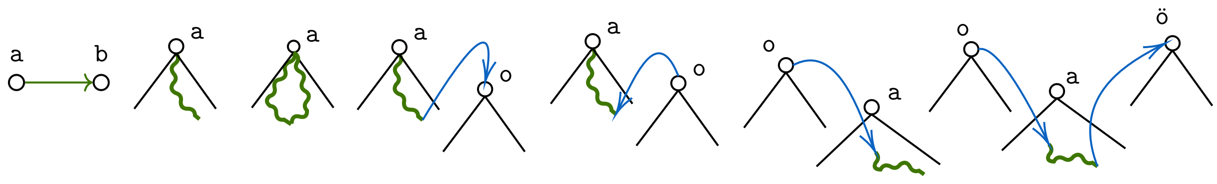


Figure 7.1: Basic paths in order of their introduction in Definition 7.25.

Our next goal is to establish that every path starting from a named individual is decomposable. The following observation focuses on paths that contain only a single named individual.

Observation 7.26. Let $\mathcal{I}$ be an $(\mathbf{N_I^{\mathcal{A}}}, \mathbf{N_I^{\mathcal{T}}})$-quasi-forest, $\mathsf{a} \in (\mathbf{N_I^{\mathcal{A}}} \cup \mathbf{N_I^{\mathcal{T}}})$, and $\rho := \mathsf{a}^{\mathcal{I}} \bar{\rho}$ be a path in $\mathcal{I}$, where $\bar{\rho}$ is nameless. Then ρ is basic. More precisely, ρ is either (i) a-inner or (ii) there exists a name $\mathsf{o} \in \mathbf{N_I^{\mathcal{T}}}$ such that $\mathsf{o}^{\mathcal{I}} = \mathsf{a}^{\mathcal{I}}$ and ρ is (b, o)-inner for some $\mathsf{b} \in (\mathbf{N_I^{\mathcal{A}}} \cup \mathbf{N_I^{\mathcal{T}}})$.

Proof. By Observation 7.24 there exists a name $\mathsf{b} \in (\mathbf{N_I^{\mathcal{A}}} \cup \mathbf{N_I^{\mathcal{T}}})$ for which $\bar{\rho}$ is a b-subtree-path. If $\mathsf{a}^{\mathcal{I}} = \mathsf{b}^{\mathcal{I}}$ then ρ is a-inner. Otherwise, by analysing how interpretation of roles is defined in quasi-forests, there exists a name $\mathsf{o} \in \mathbf{N_I^{\mathcal{T}}}$ such that $\mathsf{o}^{\mathcal{I}} = \mathsf{a}^{\mathcal{I}}$. Then ρ is (b, o)-inner. □

As the second important observation we focus on paths with two named individuals.

Observation 7.27. Let $\mathcal{I}$ be an $(\mathbf{N}_\mathbf{I}^\mathcal{A}, \mathbf{N}_\mathbf{I}^\mathcal{T})$-quasi-forest, $\mathtt{a}, \mathtt{b} \in (\mathbf{N}_\mathbf{I}^\mathcal{A} \cup \mathbf{N}_\mathbf{I}^\mathcal{T})$, and $\rho := \mathtt{a}^\mathcal{I}\bar{\rho}\mathtt{b}^\mathcal{I}$ be a path in $\mathcal{I}$, where $\bar{\rho}$ is nameless. Then ρ is basic.

Proof. If $\bar{\rho}$ is empty then ρ is $(\mathtt{a},\mathtt{b})$-direct. Otherwise, we apply Observation 7.26 to $\mathtt{a}^\mathcal{I}\bar{\rho}$ and consider the following two cases:

- $\mathtt{a}^\mathcal{I}\bar{\rho}$ is $\mathtt{a}$-inner.
  If $\mathtt{a}^\mathcal{I} = \mathtt{b}^\mathcal{I}$ then ρ is an $\mathtt{a}$-roundtrip. Otherwise, by analysing how the interpretation of roles is defined in quasi-forests, there is an $\mathtt{o} \in \mathbf{N}_\mathbf{I}^\mathcal{T}$ such that $\mathtt{b}^\mathcal{I} = \mathtt{o}^\mathcal{I}$ and ρ is $(\mathtt{a},\mathtt{o})$-inout.
- There is a name $\mathtt{o} \in \mathbf{N}_\mathbf{I}^\mathcal{T}$ such that $\mathtt{o}^\mathcal{I} = \mathtt{a}^\mathcal{I}$ and $\mathtt{a}^\mathcal{I}\bar{\rho}$ is $(\mathtt{c},\mathtt{o})$-inner for some $\mathtt{c} \in (\mathbf{N}_\mathbf{I}^\mathcal{A} \cup \mathbf{N}_\mathbf{I}^\mathcal{T})$.
  The following facts again follow by analysing how the interpretation of roles is defined in quasi-forests. If $\mathtt{b}^\mathcal{I} = \mathtt{c}^\mathcal{I}$ then ρ is $(\mathtt{b},\mathtt{o})$-outin. If $\mathtt{a}^\mathcal{I} = \mathtt{c}^\mathcal{I}$ then $\mathtt{a}^\mathcal{I}\bar{\rho}$ is $\mathtt{a}$-inner and we can proceed as in the previous case. Otherwise $\mathtt{a}^\mathcal{I} \neq \mathtt{c}^\mathcal{I} \neq \mathtt{b}^\mathcal{I}$. By inequality $\mathtt{c}^\mathcal{I} \neq \mathtt{b}^\mathcal{I}$ we infer the existence $\ddot{\mathtt{o}} \in \mathbf{N}_\mathbf{I}^\mathcal{T}$ for which $\mathtt{b}^\mathcal{I} = \ddot{\mathtt{o}}^\mathcal{I}$ holds. Then ρ is simply a $(\mathtt{c},\mathtt{o},\ddot{\mathtt{o}})$-bypass.

□

We can now employ the above two observations to establish our desired characterisation.

Lemma 7.28 If $\mathcal{I}$ is a quasi-forest then every path starting from a named element is decomposable.

Proof. Consider a sequence ϱ be the sequence composed indices of all positions in ρ that are named elements, in order of their occurrence in ρ. Define ϱ_+ to be ϱ if the last position in ϱ is equal to $|\rho|$, and to be $\varrho \cdot |\rho|$ otherwise. We claim that ϱ_+ witnesses the fact that ρ is decomposable. Indeed, take any index $j < |\varrho_+|$ and consider the fragment of ρ composed of all elements of ρ between the positions $(\varrho_+)_j$ and $(\varrho_+)_{j+1}$. If $\rho_{(\varrho_+)_{j+1}}$ is named, then such a fragment of ρ is basic by Observation 7.27. Otherwise, such a fragment of ρ is basic by Observation 7.26. Thus ρ is indeed decomposable. □

Basic paths are expressible in $\mathcal{ZOIQ}$ by means of regular expressions with tests. Their construction involves nominal tests $\{\mathtt{a}\}?$, "descendant of $\mathtt{a}$" tests $\exists(child^-)^+.\{\mathtt{a}\}?$, and role names *edge* and *child*.

Lemma 7.29 Let τ be a "category" of paths from Definition 7.25 (including nameless and outer paths). Then there exists a constant-size regular expression $\mathcal{R}_\tau$ and an NFA $\mathcal{A}_\tau$ (where tests involve only individual names explicitly mentioned in τ) that realise, for all quasi-forests, exactly the paths of the category τ.

Proof. For each "category" of paths τ from Definition 7.25 we are going to present a suitable regular expression with test describing such a path.

- $(\mathtt{a},\mathtt{b})$-direct path
$$\{\mathtt{a}\}?\ edge\ \{\mathtt{b}\}?$$
- $\mathtt{a}$-inner path
$$\{\mathtt{a}\}?\ \left(edge\ [\exists(child^-)^+.\{\mathtt{a}\}]?\ \right)^+$$
- $\mathtt{a}$-roundtrip
$$\{\mathtt{a}\}?\ \left(edge\ [\exists(child^-)^+.\{\mathtt{a}\}]?\ \right)^+\ edge\ \{\mathtt{a}\}?$$
- $(\mathtt{a},\mathtt{o})$-inout
$$\{\mathtt{a}\}?\ \left(edge\ [\exists(child^-)^+.\{\mathtt{a}\}]?\ \right)^+\ edge\ \{\mathtt{o}\}?$$
- $(\mathtt{a},\mathtt{o})$-outin
$$\{\mathtt{o}\}?\ \left(edge\ [\exists(child^-)^+.\{\mathtt{a}\}]?\ \right)^+\ edge\ \{\mathtt{a}\}?$$
- $(\mathtt{a},\mathtt{o})$-inner
$$\{\mathtt{o}\}?\ \left(edge\ [\exists(child^-)^+.\{\mathtt{a}\}]?\ \right)^+$$

- $(\mathtt{a}, \mathtt{o}, \ddot{\mathtt{o}})$-bypass
$$\{\mathtt{o}\}?\ \mathit{edge}\ \left([\exists(\mathit{child}^{-})^{+}.\{\mathtt{a}\}]?\mathit{edge}\right)^{+}\ \{\ddot{\mathtt{o}}\}?$$
- nameless path
$$(\neg\mathrm{Root})?\,(\mathit{edge}\ (\neg\mathrm{Root})?)^{*}$$
- outer path
$$(\neg\mathrm{Root})?\,(\mathit{edge}\ (\neg\mathrm{Root})?)^{*}\ \mathit{edge}\ \mathrm{Root}?$$

Correctness of the above regular expressions follows immediately by the semantics of $\mathcal{ZOIQ}$ and Definition 7.25. Given a regular expression $\mathcal{R}_{\tau}$, the desired NFA $\mathcal{A}_{\tau}$ is then obtained by the usual transformation [Sip13, Lemma 1.55] from regular expression into NFAs. □

We push the content of Lemma 7.29 even further, establishing Lemma 7.30.

Lemma 7.30 Let τ be a "category" of paths from Definition 7.25 (including nameless and outer paths), $\mathcal{A}_{\tau}$ and $\mathcal{R}_{\tau}$ be as in Lemma 7.29. Consider a regular language $\mathcal{L}$ given either by an NFA $\mathcal{A}$ or a regular expression $\mathcal{R}$. Then one can compute in polynomial time an NFA $(\mathcal{A} \bowtie \mathcal{A}_{\tau})$ and a regular expression $(\mathcal{R} \bowtie \mathcal{R}\tau)$, both of sizes polynomial, respectively, in $|\mathcal{A}|$ and in $|\mathcal{R}|$, such that:

- all concept tests present in the alphabet of $(\mathcal{A} \bowtie \mathcal{A}_{\tau})$ and $(\mathcal{R} \bowtie \mathcal{R}\tau)$ involve only individual names that are explicitly mentioned in τ, and
- both $(\mathcal{A} \bowtie \mathcal{A}_{\tau})$ and $(\mathcal{R} \bowtie \mathcal{R}\tau)$ realise, for all quasi-forests, precisely the paths of the category τ realising $\mathcal{L}$.

Proof sketch. We focus only on the case when $\mathcal{L}$ is given as an NFA. Let us first explain why we can do so.[2] Suppose that $\mathcal{L}$ is given as a regular expression $\mathcal{R}$, and convert $\mathcal{R}$ into an NFA $\mathcal{A}$ of size polynomial w.r.t. $|\mathcal{R}|$ with the classical Thompson's construction [Tho68]. As observed in the proof of the star height lemma by Gruber and Holzer [GH08, Thm. 6], when converting $\mathcal{R}$ into $\mathcal{A}$, the underlying graph of $\mathcal{A}$ is of tree-width at most 2. Our forthcoming construction of $\mathcal{A} \bowtie \mathcal{A}_{\tau}$ does not increase the tree-width of the graph of the resulting automaton, and increases the alphabet and the total number of states of the resulting NFA only by a linear factor. Hence, employing a corollary by Gruber [Gru12, Cor. 3.1], this implies that the graph of $\mathcal{A} \bowtie \mathcal{A}_{\tau}$ has the "undirected cycle rank" (known also as *tree-depth*) bounded by $k := 2\log_2(|\mathcal{A} \bowtie \mathcal{A}_{\tau}|)$, which clearly is in $\mathcal{O}(\log_2(|\mathcal{R}|))$. By the results of Gruber and Holzer [GH13, Thm. 9], we know that $\mathcal{A} \bowtie \mathcal{A}_{\tau}$ can be turned back into a regular expression $(\mathcal{R} \bowtie \mathcal{R}\tau)$ of size $|\Sigma| \cdot 4^{k} \cdot n$, where Σ is the alphabet of $\mathcal{A} \bowtie \mathcal{A}_{\tau}$ and n is the total number of states of $\mathcal{A} \bowtie \mathcal{A}_{\tau}$. Thus, $(\mathcal{R} \bowtie \mathcal{R}\tau)$ is the desired regular expression of size polynomial w.r.t. $|\mathcal{R}|$.

Suppose that an automaton $\mathcal{A}$ is given, and let $\mathtt{Q}$ be its state-set, and $\mathtt{T}$ be its transition relation. As the construction is similar for each category of paths, henceforth we focus on $(\mathtt{a}, \mathtt{o}, \ddot{\mathtt{o}})$-bypassses only (as this is the most difficult case to deal with). We recall that our goal to construct the automaton $\mathcal{A} \bowtie \mathcal{A}_{(\mathtt{a},\mathtt{o},\ddot{\mathtt{o}})\text{-bypass}}$ that realises precisely $\mathcal{A}$-paths that are $(\mathtt{a}, \mathtt{o}, \ddot{\mathtt{o}})$-bypasses, namely match the regular expression

$$\{\mathtt{o}\}?\ \mathit{edge}\ \left([\exists(\mathit{child}^{-})^{+}.\{\mathtt{a}\}]?\mathit{edge}\right)^{+}\ \{\ddot{\mathtt{o}}\}?$$

introduced in the previous lemma. Our construction of the forthcoming automaton is fully guided by the shape of the above regular expression. As the first step we construct an automaton $\mathcal{B}$ that realises all $\mathcal{A}$-paths that match the regular expression

$$\left([\exists(\mathit{child}^{-})^{+}.\{\mathtt{a}\}]?\mathit{edge}\right)^{*}.$$

To do so, we initially expand the set of states $\mathtt{Q}$ of $\mathcal{A}$ with fresh states of the form $\mathtt{q}_{\delta}$ for all $\delta \in \mathtt{T}$. Then, we "split" every transition δ in $\mathtt{T}$ into two "parts". Suppose that δ leads from $\mathtt{q}$ to $\mathtt{q}'$ after

[2] I would like to thank Hermann Gruber for the proof [Gru23] of this fact.

reading the letter $\mathtt{a}$. We thus (i) replace δ in $\mathtt{T}$ with the transition that transforms $\mathtt{q}$ into $\mathtt{q}_\delta$ after reading $\mathtt{a}$, and (ii) append the transition $(\mathtt{q}_\delta, [\exists(\mathit{child}^-)^+.\{\mathtt{a}\}]?, \mathtt{q}')$ to $\mathtt{T}$. Call the resulting NFA $\mathcal{B}$. It is easy to see that the $\mathcal{B}$-paths are precisely the desired $\mathcal{A}$-paths realising the regular expression $\left([\exists(\mathit{child}^-)^+.\{\mathtt{a}\}]?\mathit{edge}\right)^*$. We next aim at realising $\left([\exists(\mathit{child}^-)^+.\{\mathtt{a}\}]?\mathit{edge}\right)^+$, *i.e.* we want to replace Kleene's star with the Kleene's plus in the previous construction.

Let us decompose $\mathcal{B} := (\Sigma, \mathtt{Q}_\mathcal{B}, \mathtt{I}_\mathcal{B}, \mathtt{F}_\mathcal{B}, \mathtt{T}_\mathcal{B})$, where all the components of $\mathcal{B}$ are as usual. We define $\mathcal{C}$ as a variant of the "product" automaton of $\mathcal{B}$ with itself, namely $\mathcal{B} := (\Sigma, \mathtt{Q}_\mathcal{B} \times \{1,2\}, \mathtt{I}_\mathcal{B} \times \{1\}, \mathtt{F}_\mathcal{B} \times \{2\}, \mathtt{T}_\mathcal{C})$, and the transition relation is defined as the sum of (a) $\{((\mathtt{q},2),\mathtt{a},(\mathtt{q}',2)) \mid (\mathtt{q},\mathtt{a},\mathtt{q}') \in \mathtt{T}_\mathcal{B}\}$, (b) $\{((\mathtt{q},1),\mathtt{a},(\mathtt{q}',1)) \mid (\mathtt{q},\mathtt{a},\mathtt{q}') \in \mathtt{T}_\mathcal{B}, \mathtt{a} \text{ is a test}\}$, and (c) $\{((\mathtt{q},1),\mathtt{a},(\mathtt{q}',2)) \mid (\mathtt{q},\mathtt{a},\mathtt{q}') \in \mathtt{T}_\mathcal{B}, \mathtt{a} \text{ is a role}\}$. The intuition is that in the runs of $\mathcal{C}$ we first traverse the first copy of $\mathcal{B}$, and after reading the "edge" we jump to the second copy and stay there till the end. Clearly the NFA $\mathcal{C}$ realises precisely the $\mathcal{A}$-paths matching the mentioned regular expression.

With nearly the same construction, we can design an automaton $\mathcal{D}$ that realise precisely the $\mathcal{A}$-paths matching the regular expression:

$$\mathit{edge}\ \left([\exists(\mathit{child}^-)^+.\{\mathtt{a}\}]?\mathit{edge}\right)^+ .$$

The trick here is that instead of taking two copies of $\mathcal{B}$, we take a copy of $\mathcal{A}$ and a copy of $\mathcal{C}$, and forward all transitions concerning roles from $\mathcal{A}$ to $\mathcal{C}$. More formally, after decomposing $\mathcal{C}$ as $(\Sigma, \mathtt{Q}_\mathcal{C}, \mathtt{I}_\mathcal{C}, \mathtt{F}_\mathcal{C}, \mathtt{T}_\mathcal{C})$, we let $\mathcal{B} := (\Sigma, \mathtt{Q} \cup \mathtt{Q}_\mathcal{C}, \mathtt{I}, \mathtt{F}_\mathcal{C}, \mathtt{T}_\mathcal{C})$, where $\mathtt{T}_\mathcal{C}$ is the union of (a) $\mathtt{T}_\mathcal{B}$, (b) $\{(\mathtt{q},\mathtt{a},\mathtt{q}') \mid (\mathtt{q},\mathtt{a},\mathtt{q}') \in \mathtt{T}, \mathtt{a} \text{ is a test}\}$, and (c) $\{(\mathtt{q},\mathtt{a},(\mathtt{q}',1)) \mid (\mathtt{q},\mathtt{a},\mathtt{q}') \in \mathtt{T}, \mathtt{a} \text{ is a test}\}$. Once more, it can be readily checked that the resulting automaton does its job. Now the desired automaton $\mathcal{A} \bowtie \mathcal{A}_{(\mathtt{a},\mathtt{o},\ddot{\mathtt{o}})\text{-bypass}}$ is obtained from $\mathcal{D}$ by extending its state set with two states $\mathtt{q}_s, \mathtt{q}_e$, making $\mathtt{q}_s$ the initial state, making $\mathtt{q}_e$ the final state, and appending new transitions of the form $(\mathtt{q}_s, \{\mathtt{o}\}?, \mathtt{q})$ and $(\mathtt{q}', \{\ddot{\mathtt{o}}\}?, \mathtt{q}_e)$ for all initial states $\mathtt{q}$ of $\mathcal{D}$, and all final states $\mathtt{q}'$ of $\mathcal{D}$. This guarantees that the realised paths starts in $\mathtt{o}$ and ends in $\ddot{\mathtt{o}}$ as desired. □

7.4 Automata decorations

Prior to moving to quite technical details, we start by discussing the main goals of this section.

7.4.1 Automata decorations: Overview

Recall that the *clearing* of a quasi-forest $\mathcal{I}$ is the substructure of $\mathcal{I}$ induced by its roots. In this section we employ Lemma 7.28 to devise a method of decorating the clearing of $\mathcal{I}$ with additional information about the basic paths realising a given NFA $\mathcal{A}$. This will help us to decide the satisfaction of automata concepts of the form $\exists\mathcal{A}_{\mathtt{q},\mathtt{q}'}.\top$ in quasi-forests in a modular way, *i.e.* independently for the clearing of a quasi-forest and all of its subtrees. To make the word "modular" more precise, observe that:

Observation 7.31. Let $\mathcal{I}$ be an $(\mathbf{N}_\mathbf{I}^\mathcal{A}, \mathbf{N}_\mathbf{I}^\mathcal{T})$-quasi-forest and let $\mathrm{d} \in (\exists\mathcal{A}.\top)^\mathcal{I}$ for an NFA $\mathcal{A}$. Then there is an $\mathcal{A}$-path ρ starting from d such that: (a) d is a root of $\mathcal{I}$, (b) ρ is nameless, or (c) $\rho = \bar{\rho}\cdot\mathrm{e}\cdot\hat{\rho}$, where $\bar{\rho}$ is nameless, e is a root of $\mathcal{I}$ and either (i) e is a nominal root, or (ii) all elements of $\bar{\rho}$ are descendants of e.

Hence, any path witnessing an automata concept either (i) starts from a root, (ii) is fully contained inside subtrees, or (iii) walks inside a subtree and then jumps to a root (so the remaining path is outer). This observation transfers to paths realising NFA in the following way. For a given NFA $\mathcal{A}$ with the state set $\mathtt{Q}$, the **$\mathcal{A}$-reachability-concepts $\mathbf{C}_\rightsquigarrow(\mathcal{A})$** are defined as the set $\{\mathrm{Reach}^\mathcal{A}_{\mathtt{q},\mathtt{q}'} \mid \mathtt{q},\mathtt{q}' \in \mathtt{Q}\}$. Our intention behind such concepts is that $\mathrm{Reach}^\mathcal{A}_{\mathtt{q},\mathtt{q}'}$ labels precisely the roots of quasi-forests, satisfying $\exists\mathcal{A}_{\mathtt{q},\mathtt{q}'}.\top$. The forthcoming crucial lemma explains the desired "modularity" condition. It simply says that if the reachability concepts $\mathrm{Reach}^\mathcal{A}_{\mathtt{q},\mathtt{q}'}$ are interpreted as intended, then the verification of whether an element d satisfies the $\exists\mathcal{A}_{\mathtt{q},\mathtt{q}'}.\top$ concept reduces to (i) a test whether d is labelled with $\mathrm{Reach}^\mathcal{A}_{\mathtt{q},\mathtt{q}'}$ if d is a root, or (ii) testing existence of a certain path which fully contained in the subtree of d, possibly with the exception of the last element which can also be the root of d or a nominal. For the statement of the lemma, we rely on the NFAs $\mathcal{A}_{\text{nmls}}$ and $\mathcal{A}_{\text{outr}}$ from Lemma 7.30 that detect, respectively, nameless and outer paths.

Lemma 7.32 Let $\mathcal{I}$ be an $(\mathbf{N}_\mathbf{I}^{\mathcal{A}}, \mathbf{N}_\mathbf{I}^{\mathcal{T}})$-quasi-forest, and $\mathcal{A}$ be an NFA with the set of states $\mathtt{Q}$. Suppose that $\mathcal{I}$ interprets all the concepts $\text{Reach}^{\mathcal{A}}_{\mathtt{q},\mathtt{q}'}$ from $\mathbf{C}_{\rightsquigarrow}(\mathcal{A})$ equally to $\text{Root} \sqcap \exists\mathcal{A}_{\mathtt{q},\mathtt{q}'}.\top$. Then for all states $\mathtt{q}, \mathtt{q}' \in \mathtt{Q}$, the concept $\exists\mathcal{A}_{\mathtt{q},\mathtt{q}'}.\top$ is interpreted in $\mathcal{I}$ equally to the union of concepts:

- $\text{Root} \sqcap \text{Reach}_{\mathtt{q},\mathtt{q}'}$,
- $\neg\text{Root} \sqcap \exists(\mathcal{A}_{\mathtt{q},\mathtt{q}'} \bowtie \mathcal{A}_{\text{nmls}}).\top$, and
- $\neg\text{Root} \sqcap \bigsqcup_{\hat{\mathtt{q}}\in\mathtt{Q}} \exists(\mathcal{A}_{\mathtt{q},\hat{\mathtt{q}}} \bowtie \mathcal{A}_{\text{outr}}).(\text{Root} \sqcap \text{Reach}_{\hat{\mathtt{q}},\mathtt{q}'})$.

Proof. As we interpret $\text{Reach}^{\mathcal{A}}_{\mathtt{q},\mathtt{q}'}$ equally to $\text{Root}\sqcap\exists\mathcal{A}_{\mathtt{q},\mathtt{q}'}.\top$, it suffices to establish the equality:

$$\text{LHS} := (\neg\text{Root} \sqcap \exists\mathcal{A}_{\mathtt{q},\mathtt{q}'}.\top)^{\mathcal{I}} = \left(\neg\text{Root} \sqcap \left[(\exists\mathcal{C}.\top) \sqcup \bigsqcup_{\hat{\mathtt{q}}\in\mathtt{Q}} \exists\mathcal{C}_{\mathtt{q},\hat{\mathtt{q}}}.\text{Reach}_{\hat{\mathtt{q}},\mathtt{q}'}\right]\right)^{\mathcal{I}} := \text{RHS},$$

where automata $\mathcal{C}$ are defined in the statement of the lemma. We establish two concept inclusions. The inclusion RHS $\subseteq$ LHS is immediate because (by construction) the languages of $\mathcal{C}$ are subsets of the language of the automaton $\mathcal{A}_{\mathtt{q},\mathtt{q}'}$. For the other direction, suppose that $\mathrm{d} \in \text{LHS}$ and let ρ be any path from d that realises $\mathcal{A}_{\mathtt{q},\mathtt{q}'}$. If ρ is nameless then ρ realises $\mathcal{A}_{\text{nmls}}$, and thus $\mathrm{d} \in (\exists\mathcal{C}.\top)^{\mathcal{I}}$. Otherwise, let i be the position of the first named element in ρ. Then there exists a state $\hat{\mathtt{q}}$ such that the path $\rho_1 \ldots \rho_i$ realises $\mathcal{A}_{\mathtt{q},\hat{\mathtt{q}}}$, and $\rho_i \ldots \rho_{|\rho|}$ realises $\mathcal{A}_{\hat{\mathtt{q}},\mathtt{q}'}$. Moreover, $\rho_1 \ldots \rho_i$ realises $\mathcal{A}_{\text{outr}}$, and hence by the proper interpretation of the reachability concepts we have $\rho_i \in (\text{Reach}^{\mathcal{A}}_{\hat{\mathtt{q}},\mathtt{q}'})$. Thus $\mathrm{d} \in (\exists\mathcal{C}_{\mathtt{q},\hat{\mathtt{q}}}.\text{Reach}_{\hat{\mathtt{q}},\mathtt{q}'})^{\mathcal{I}}$, which concludes the proof. □

We call the concept union from Lemma 7.32 the **$\mathbf{N}_\mathbf{I}^{\mathcal{T}}$-relativisation** of $\exists\mathcal{A}_{\mathtt{q},\mathtt{q}'}.\top$, and denote it with $\text{rel}(\mathbf{N}_\mathbf{I}^{\mathcal{T}}, \mathcal{A}_{\mathtt{q},\mathtt{q}'})$. We say that $\mathrm{d} \in \Delta^{\mathcal{I}}$ **virtually satisfies** $\exists\mathcal{A}_{\mathtt{q},\mathtt{q}'}.\top$ whenever d satisfies $\text{rel}(\mathbf{N}_\mathbf{I}^{\mathcal{T}}, \mathcal{A}_{\mathtt{q},\mathtt{q}'})$. Thus, Lemma 7.32 tells us that the notion of satisfaction and virtual satisfaction coincide whenever the reachability concepts are interpreted as desired.

7.4.2 Automata decorations: Decorations

In the reminder of the section we introduce several $\mathcal{ZOIQ}$-concepts and roles, dubbed *automata decorations*, intended to guarantee the desired interpretation of the reachability concepts. The following concepts from $\mathbf{C}(\mathbf{N}_\mathbf{I}^{\mathcal{T}}, \mathcal{A})$ "bookkeep" the information about relevant basic paths realising NFA starting from a given root.

Definition 7.33 Given an NFA $\mathcal{A}$ with the state-set $\mathtt{Q}$, the set of **$(\mathbf{N}_\mathbf{I}^{\mathcal{T}}, \mathcal{A})$-concepts $\mathbf{C}(\mathbf{N}_\mathbf{I}^{\mathcal{T}}, \mathcal{A})$** is composed of the following concept names (for all states $\mathtt{q}, \mathtt{q}' \in \mathtt{Q}$ and individual names $\mathtt{o}, \ddot{\mathtt{o}} \in \mathbf{N}_\mathbf{I}^{\mathcal{T}}$):

$$\text{D}^{\mathcal{A}}_{\mathtt{q},\mathtt{q}'},\ \text{I}^{\mathcal{A}}_{\mathtt{q},\mathtt{q}'},\ \text{RT}^{\mathcal{A}}_{\mathtt{q},\mathtt{q}'},\ \text{IO}^{\mathcal{A},\mathtt{o}}_{\mathtt{q},\mathtt{q}'},\ \text{OI}^{\mathcal{A},\mathtt{o}}_{\mathtt{q},\mathtt{q}'},\ \text{I}^{\mathcal{A},\mathtt{o}}_{\mathtt{q},\mathtt{q}'},\ \text{By}^{\mathcal{A},\mathtt{o},\ddot{\mathtt{o}}}_{\mathtt{q},\mathtt{q}'}.$$

A quasi-forest $\mathcal{I}$ **properly interprets** $\mathbf{C}(\mathbf{N}_\mathbf{I}^{\mathcal{T}}, \mathcal{A})$ if $\text{C}^{\mathcal{I}} = \{\mathtt{a}^{\mathcal{I}} \mid \mathtt{a} \in \mathbf{N}_\mathbf{I}, \text{cond}(\text{C}, \mathtt{a})\}$ for concepts C and conditions $\text{cond}(\text{C}, \mathtt{a})$ as indicated below.

conc. C	$\mathrm{cond}(\mathrm{C}, \mathtt{a})$: "There is a path $\rho \models \mathcal{A}_{\mathtt{q},\mathtt{q}'}$ such that
$\mathrm{D}^{\mathcal{A}}_{\mathtt{q},\mathtt{q}'}$	ρ is $(\mathtt{a}, \mathtt{b})$-direct for some $\mathtt{b}^{\mathcal{I}}$".
$\mathrm{I}^{\mathcal{A}}_{\mathtt{q},\mathtt{q}'}$	ρ is $\mathtt{a}$-inner".
$\mathrm{RT}^{\mathcal{A}}_{\mathtt{q},\mathtt{q}'}$	ρ is an $\mathtt{a}$-rountrip".
$\mathrm{IO}^{\mathcal{A},\mathtt{o}}_{\mathtt{q},\mathtt{q}'}$	ρ is an $(\mathtt{a}, \mathtt{o})$-inout".
$\mathrm{OI}^{\mathcal{A},\mathtt{o}}_{\mathtt{q},\mathtt{q}'}$	ρ is an $(\mathtt{a}, \mathtt{o})$-outin".
$\mathrm{I}^{\mathcal{A},\mathtt{o}}_{\mathtt{q},\mathtt{q}'}$	ρ is $(\mathtt{a}, \mathtt{o})$-inner".
$\mathrm{By}^{\mathcal{A},\mathtt{o},\ddot{\mathtt{o}}}_{\mathtt{q},\mathtt{q}'}$	ρ is an $(\mathtt{a}, \mathtt{o}, \ddot{\mathtt{o}})$-bypass".

Note that the size of $\mathbf{C}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \mathcal{A})$ is clearly polynomial w.r.t. $|\mathcal{A}| \cdot |\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}|$.

While the above definition may seem to be complicated, we strongly encourage the reader to analyse the figure below in order to gain extra intuitions on how the concepts from $\mathbf{C}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \mathcal{A})$ work.

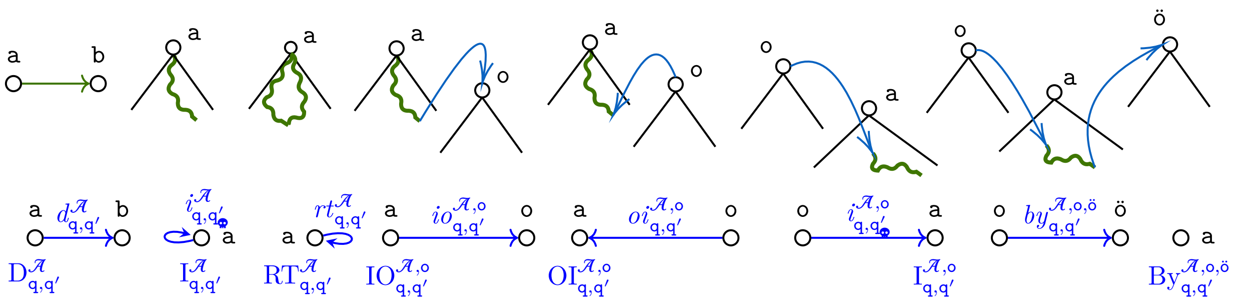

a
b
a
a
a
o
a
o
o
a
o
a
ö
$d^{\mathcal{A}}_{\mathtt{q},\mathtt{q}'}$
$i^{\mathcal{A}}_{\mathtt{q},\mathtt{q}'}$
$rt^{\mathcal{A}}_{\mathtt{q},\mathtt{q}'}$
$io^{\mathcal{A},\mathtt{o}}_{\mathtt{q},\mathtt{q}'}$
$oi^{\mathcal{A},\mathtt{o}}_{\mathtt{q},\mathtt{q}'}$
$i^{\mathcal{A},\mathtt{o}}_{\mathtt{q},\mathtt{q}'}$
$by^{\mathcal{A},\mathtt{o},\ddot{\mathtt{o}}}_{\mathtt{q},\mathtt{q}'}$
$\mathrm{D}^{\mathcal{A}}_{\mathtt{q},\mathtt{q}'}$
$\mathrm{I}^{\mathcal{A}}_{\mathtt{q},\mathtt{q}'}$
$\mathrm{RT}^{\mathcal{A}}_{\mathtt{q},\mathtt{q}'}$
$\mathrm{IO}^{\mathcal{A},\mathtt{o}}_{\mathtt{q},\mathtt{q}'}$
$\mathrm{OI}^{\mathcal{A},\mathtt{o}}_{\mathtt{q},\mathtt{q}'}$
$\mathrm{I}^{\mathcal{A},\mathtt{o}}_{\mathtt{q},\mathtt{q}'}$
$\mathrm{By}^{\mathcal{A},\mathtt{o},\ddot{\mathtt{o}}}_{\mathtt{q},\mathtt{q}'}$

Figure 7.2: Basic paths and their corresponding decorations in quasi-forests, in order of their introduction.

Throughout the chapter, we often employ a *fresh* individual name 👻 (the *ghost variable*), to stress that the constructed concepts are independent from $\mathbf{N}_{\mathbf{I}}^{\mathcal{A}}$ (but may still depend on 👻 or $\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}$). With C[👻/a] we denote the result of substituting $\mathtt{a}$ for all occurrences of 👻 in C. Relying on NFAs from Lemma 7.30, we can construct $\mathcal{ZOIQ}$-concepts that describe the intended behaviour of $(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \mathcal{A})$-concepts. For instance, $\mathrm{RT}^{\mathcal{A}}_{\mathtt{q},\mathtt{q}'}$ can be defined as $\exists(\mathcal{A}_{\mathtt{q},\mathtt{q}'} \bowtie \mathcal{A}_{\tau}).\top$, where $\mathcal{A}_{\tau}$ is the NFA from Lemma 7.30 describing 👻-roundtrips.

Lemma 7.34 For all C from $\mathbf{C}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \mathcal{A})$, there exists a $\mathcal{ZOIQ}$-concept desc(C) (of size polynomial w.r.t. $|\mathcal{A}| \cdot |\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}|$) that uses only individual names from $\mathbf{N}_{\mathbf{I}}^{\mathcal{T}} \cup \{$👻$\}$ with the property that "for all $(\mathbf{N}_{\mathbf{I}}^{\mathcal{A}}, \mathbf{N}_{\mathbf{I}}^{\mathcal{T}})$-quasi-forests $\mathcal{I}$ and $\mathtt{a} \in \mathbf{N}_{\mathbf{I}}$ we have that $\mathrm{cond}(\mathrm{C}, \mathtt{a})$ is satisfied in $\mathcal{I}$ iff $\mathtt{a}^{\mathcal{I}}$ is in (desc(C)[👻/a])$^{\mathcal{I}}$".

Proof. Let $\mathtt{T} \subseteq \mathtt{Q} \times \mathbf{N}_{\mathbf{R}}^{\mathrm{sp}} \times \mathtt{Q}$ be the transition relation of $\mathcal{A}$. We define desc(C)-concepts as:

- $\mathrm{desc}(\mathrm{D}^{\mathcal{A}}_{\mathtt{q},\mathtt{q}'}) \coloneqq \{$👻$\} \sqcap \bigsqcup_{(\mathtt{q}, r, \mathtt{q}') \in \mathtt{T}} \exists r.\mathrm{Root}$
- $\mathrm{desc}(\mathrm{I}^{\mathcal{A}}_{\mathtt{q},\mathtt{q}'}) \coloneqq \exists(\mathcal{A}_{\mathtt{q},\mathtt{q}'} \bowtie \mathcal{A}_{\tau}).\top$, where $\mathcal{A}_{\tau}$ is the NFA from Lemma 7.30 describing 👻-inner paths.
- $\mathrm{desc}(\mathrm{RT}^{\mathcal{A}}_{\mathtt{q},\mathtt{q}'}) \coloneqq \exists(\mathcal{A}_{\mathtt{q},\mathtt{q}'} \bowtie \mathcal{A}_{\tau}).\top$, where $\mathcal{A}_{\tau}$ is the NFA from Lemma 7.30 describing 👻-roundtrips.
- $\mathrm{desc}(\mathrm{IO}^{\mathcal{A},\mathtt{o}}_{\mathtt{q},\mathtt{q}'}) \coloneqq \exists(\mathcal{A}_{\mathtt{q},\mathtt{q}'} \bowtie \mathcal{A}_{\tau}).\top$, where $\mathcal{A}_{\tau}$ is the NFA from Lemma 7.30 describing (👻, $\mathtt{o}$)-inouts.
- $\mathrm{desc}(\mathrm{OI}^{\mathcal{A},\mathtt{o}}_{\mathtt{q},\mathtt{q}'}) \coloneqq \exists(\mathcal{A}_{\mathtt{q},\mathtt{q}'} \bowtie \mathcal{A}_{\tau}).\top$, where $\mathcal{A}_{\tau}$ is the NFA from Lemma 7.30 describing (👻, $\mathtt{o}$)-outins.
- $\mathrm{desc}(\mathrm{I}^{\mathcal{A},\mathtt{o}}_{\mathtt{q},\mathtt{q}'}) \coloneqq \exists(\mathcal{A}_{\mathtt{q},\mathtt{q}'} \bowtie \mathcal{A}_{\tau}).\top$, where $\mathcal{A}_{\tau}$ is the NFA from Lemma 7.30 describing (👻, $\mathtt{o}$)-inner paths.
- $\mathrm{desc}(\mathrm{By}^{\mathcal{A},\mathtt{o},\ddot{\mathtt{o}}}_{\mathtt{q},\mathtt{q}'}) \coloneqq \exists edge^{*}\left(\{\mathtt{o}\} \sqcap \exists(\mathcal{A}_{\mathtt{q},\mathtt{q}'} \bowtie \mathcal{A}_{\tau}).\top\right)$, where $\mathcal{A}_{\tau}$ is the NFA from Lemma 7.30 describing (👻, $\mathtt{o}$, $\ddot{\mathtt{o}}$)-bypasses.

Their correctness follow immediately by Lemma 7.30 and the semantics of $\mathcal{ZOIQ}$. □

Define comdsc($\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \mathcal{A}$) as the conjunction of C$\leftrightarrow$desc(C) over all concept names C in $\mathbf{C}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \mathcal{A})$ (here A$\leftrightarrow$B abbreviates (A$\sqcap$B)$\sqcup$($\neg$A$\sqcap\neg$B)). We invoke Lemma 7.34 to rephrase the notion of proper interpretation of $\mathbf{C}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \mathcal{A})$ in the language of satisfaction of $\mathcal{ZOIQ}$-concepts by the clearings of forests.

Corollary 7.35

With comdsc($\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \mathcal{A}$) defined as $\bigsqcap_{\mathrm{C} \in \mathbf{C}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \mathcal{A})}(\mathrm{C} \leftrightarrow \mathrm{desc}(\mathrm{C}))$ we have that an $(\mathbf{N}_{\mathbf{I}}^{\mathcal{A}}, \mathbf{N}_{\mathbf{I}}^{\mathcal{T}})$-quasi-forest $\mathcal{I}$ properly interprets $\mathbf{C}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \mathcal{A})$ if and only if $\mathsf{a}^{\mathcal{I}} \in \mathrm{comdsc}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \mathcal{A})\,[\text{👻}/\mathsf{a}]^{\mathcal{I}}$ for all names $\mathsf{a} \in (\mathbf{N}_{\mathbf{I}}^{\mathcal{A}} \cup \mathbf{N}_{\mathbf{I}}^{\mathcal{T}})$.

The information provided by the proper interpretation of $(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \mathcal{A})$-concepts do not suffice yet to ensure the proper interpretation of the reachability concepts by the clearings of quasi-forests. The main reason is that the concepts from $(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \mathcal{A})$ speak only about the basic paths. As some automata constraints cannot be satisfied by basic paths, *e.g.* $\exists(\{\mathsf{o}\}?\mathit{edge}\{\ddot{\mathsf{o}}\}?\mathit{edge}\{\breve{\mathsf{o}}\}?).\top$, more work needs to be done. To be able to compose basic paths into bigger pieces, another "layer of decoration" is needed: this time with fresh roles representing the endpoints of basic paths from Definition 7.25, as summarised by Figure 7.2. For instance, whenever there is an (a, o)-inout ρ realising $\mathcal{A}_{\mathsf{q},\mathsf{q}'}$ in a quasi-forest $\mathcal{I}$, the roots $\mathsf{a}^{\mathcal{I}}$ and $\mathsf{o}^{\mathcal{I}}$ are going to be linked by the role $io^{\mathcal{A},\mathsf{o}}_{\mathsf{q},\mathsf{q}'}$ (and similarly for other categories of basic paths that start and end in roots). Such roles can be seen as "aggregated paths". The "aggregated paths" will be unfolded afterwards into real paths, and the satisfaction of automata constraints by them will be verified by means of the *guided automata* (introduced in Definition 7.37). However, there exist cases of basic paths where the last element is unnamed, more precisely (a, o)-inner and a-inner paths. In their cases we treat $\mathsf{a}^{\mathcal{I}}$ as the "virtual end" of ρ (and hence we link $\mathsf{a}^{\mathcal{I}}$ and $\mathsf{o}^{\mathcal{I}}$). To ensure that such an "aggregated path" can no longer be extended, we decorate the corresponding role with the dead-end symbol ☠. When constructing the guided automaton, the roles carrying ☠ will lead to states with no outgoing transitions (dead ends).

Definition 7.36 Given an NFA $\mathcal{A}$ with the state-set Q, the set of **$(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \mathcal{A})$-roles $\mathbf{R}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \mathcal{A})$** is composed of the following role names (for all states $\mathsf{q}, \mathsf{q}' \in \mathsf{Q}$ and individual names $\mathsf{o}, \ddot{\mathsf{o}} \in \mathbf{N}_{\mathbf{I}}^{\mathcal{T}}$):

$$d^{\mathcal{A}}_{\mathsf{q},\mathsf{q}'},\ i^{\mathcal{A}}_{\mathsf{q},\mathsf{q}'_{☠}},\ rt^{\mathcal{A}}_{\mathsf{q},\mathsf{q}'},\ io^{\mathcal{A},\mathsf{o}}_{\mathsf{q},\mathsf{q}'},\ oi^{\mathcal{A},\mathsf{o}}_{\mathsf{q},\mathsf{q}'},\ i^{\mathcal{A},\mathsf{o}}_{\mathsf{q},\mathsf{q}'_{☠}},\ by^{\mathcal{A},\mathsf{o},\ddot{\mathsf{o}}}_{\mathsf{q},\mathsf{q}'}.$$

An $(\mathbf{N}_{\mathbf{I}}^{\mathcal{A}}, \mathbf{N}_{\mathbf{I}}^{\mathcal{T}})$-quasi-forest $\mathcal{I}$ **properly interprets** $\mathbf{R}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \mathcal{A})$ if its roles consist of pairs p of roots of $\mathcal{I}$ satisfying the conditions stated below.

Role name r	Pair p	Condition
$d^{\mathcal{A}}_{\mathsf{q},\mathsf{q}'}$	$(\mathsf{a}^{\mathcal{I}}, \mathsf{b}^{\mathcal{I}})$	$\mathsf{a}^{\mathcal{I}}\mathsf{b}^{\mathcal{I}} \models \mathcal{A}_{\mathsf{q},\mathsf{q}'}$
$i^{\mathcal{A}}_{\mathsf{q},\mathsf{q}'_{☠}}$	$(\mathsf{a}^{\mathcal{I}}, \mathsf{a}^{\mathcal{I}})$	$\mathsf{a}^{\mathcal{I}}$ is in $(\mathrm{I}^{\mathcal{A}}_{\mathsf{q},\mathsf{q}'})^{\mathcal{I}}$
$rt^{\mathcal{A}}_{\mathsf{q},\mathsf{q}'}$	$(\mathsf{a}^{\mathcal{I}}, \mathsf{a}^{\mathcal{I}})$	$\mathsf{a}^{\mathcal{I}}$ is in $(\mathrm{RT}^{\mathcal{A}}_{\mathsf{q},\mathsf{q}'})^{\mathcal{I}}$
$io^{\mathcal{A},\mathsf{o}}_{\mathsf{q},\mathsf{q}'}$	$(\mathsf{a}^{\mathcal{I}}, \mathsf{o}^{\mathcal{I}})$	$\mathsf{a}^{\mathcal{I}}$ is in $(\mathrm{IO}^{\mathcal{A},\mathsf{o}}_{\mathsf{q},\mathsf{q}'})^{\mathcal{I}}$
$oi^{\mathcal{A},\mathsf{o}}_{\mathsf{q},\mathsf{q}'}$	$(\mathsf{o}^{\mathcal{I}}, \mathsf{a}^{\mathcal{I}})$	$\mathsf{a}^{\mathcal{I}}$ is in $(\mathrm{OI}^{\mathcal{A},\mathsf{o}}_{\mathsf{q},\mathsf{q}'})^{\mathcal{I}}$
$i^{\mathcal{A},\mathsf{o}}_{\mathsf{q},\mathsf{q}'_{☠}}$	$(\mathsf{o}^{\mathcal{I}}, \mathsf{a}^{\mathcal{I}})$	$\mathsf{a}^{\mathcal{I}}$ is in $(\mathrm{I}^{\mathcal{A},\mathsf{o}}_{\mathsf{q},\mathsf{q}'})^{\mathcal{I}}$
$by^{\mathcal{A},\mathsf{o},\ddot{\mathsf{o}}}_{\mathsf{q},\mathsf{q}'}$	$(\mathsf{o}^{\mathcal{I}}, \ddot{\mathsf{o}}^{\mathcal{I}})$	$(\mathrm{By}^{\mathcal{A},\mathsf{o},\ddot{\mathsf{o}}}_{\mathsf{q},\mathsf{q}'})^{\mathcal{I}}$ is non-empty

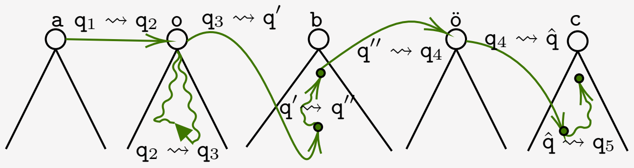
The size of $\mathbf{R}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \mathcal{A})$ is polynomial w.r.t. $|\mathcal{A}| \cdot |\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}|$.

We stress that we interpreted role names from $\mathbf{R}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \mathcal{A})$ based on the concepts from $\mathbf{C}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \mathcal{A})$ rather than on the existence of certain paths realising $\mathcal{A}$. This allows us to verify their proper interpretation, independently from the verification of the proper interpretation of $\mathbf{C}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \mathcal{A})$-concepts. The intuition behind the roles from $\mathbf{R}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \mathcal{A})$ is that they can be seen as "shortcuts" aggregating fragments of runs of $\mathcal{A}$. To retrieve proper runs, we invoke the $\mathcal{A}$-guided automaton that operates solely on such "shortcuts". The definition comes next, supplemented with Example 7.38.

Definition 7.37 For a PSA $\mathcal{A}$ with the state-set $\mathtt{Q}$, we define the **$\mathcal{A}$-guided PSA $\mathcal{B}$**. The set of states $\mathtt{Q}'$ of $\mathcal{B}$ consists of all $\mathtt{q} \in \mathtt{Q}$ and their fresh copies $\mathtt{q}_{☠}$. The transitions in $\mathcal{B}$ have the form $(\mathtt{q}, r, \mathtt{q}')$ for all $\mathtt{q}, \mathtt{q}' \in \mathtt{Q}'$, and all r from $\mathbf{R}(\mathbf{N_I^{\mathcal{T}}}, \mathcal{A})$ labelled with the ordered pair $(\mathtt{q}, \mathtt{q}')$. Note that by design, all states decorated with ☠ have no outgoing transitions.

The guided automaton $\mathcal{B}$ is of size polynomial in $|\mathcal{A}|\cdot|\mathbf{N_I^{\mathcal{T}}}|$. It traverses only the clearings of quasi-forests.

Example 7.38. Let $\rho := \mathtt{a}^{\mathcal{I}}\cdot\mathtt{o}^{\mathcal{I}}\cdot\rho_1\cdot\mathtt{o}^{\mathcal{I}}\rho_2\cdot\ddot{\mathtt{o}}^{\mathcal{I}}\cdot\rho_3 \models \mathcal{A}_{\mathtt{q}_1,\mathtt{q}_5}$ be a path in a quasi-forest $\mathcal{I}$ (as depicted below), and let $\mathcal{A}$ be an NFA with states indicated by the decorated letters $\mathtt{q}$. Suppose (i) $\mathtt{a}^{\mathcal{I}}\cdot\mathtt{o}^{\mathcal{I}} \models \mathcal{A}_{\mathtt{q}_1,\mathtt{q}_2}$ is $(\mathtt{a},\mathtt{o})$-direct, (ii) $\mathtt{o}^{\mathcal{I}}\cdot\rho_1\cdot\mathtt{o}^{\mathcal{I}} \models \mathcal{A}_{\mathtt{q}_2,\mathtt{q}_3}$ is an $\mathtt{o}$-roundtrip, (iii) $\mathtt{o}^{\mathcal{I}}\rho_2\cdot\ddot{\mathtt{o}}^{\mathcal{I}} \models \mathcal{A}_{\mathtt{q}_3,\mathtt{q}_4}$ is an $(\mathtt{b},\mathtt{o},\ddot{\mathtt{o}})$-bypass, and (iv) $\ddot{\mathtt{o}}^{\mathcal{I}}\cdot\rho_3 \models \mathcal{A}_{\mathtt{q}_4,\mathtt{q}_5}$ is $(\mathtt{c},\ddot{\mathtt{o}})$-inner. Clearly, $\mathtt{a}^{\mathcal{I}} \in (\exists\mathcal{A}_{\mathtt{q}_1,\mathtt{q}_5}.\top)^{\mathcal{I}}$.



$$\begin{tikzcd}
\mathtt{a} \arrow[r, "d^{\mathcal{A}}_{\mathtt{q}_1,\mathtt{q}_2}"] & \mathtt{o} \arrow[r, "rt^{\mathcal{A}}_{\mathtt{q}_2,\mathtt{q}_3}"] & \mathtt{o} \arrow[r, "by^{\mathcal{A},\mathtt{o},\ddot{\mathtt{o}}}_{\mathtt{q}_2,\mathtt{q}_3}"] & \ddot{\mathtt{o}} \arrow[r, "i^{\mathcal{A},\ddot{\mathtt{o}}}_{\mathtt{q}_4,(\mathtt{q}_5)_{☠}}"] & \mathtt{c}
\end{tikzcd}$$

Consider now the $\mathcal{A}$-guided PSA $\mathcal{B}$, and observe that $\mathtt{a}^{\mathcal{I}}$ is in $(\exists\mathcal{B}_{\mathtt{q}_1,(\mathtt{q}_5)_{☠}}.\top)^{\mathcal{I}}$, as witnessed by the path $\mathtt{a}^{\mathcal{I}}\cdot\mathtt{o}^{\mathcal{I}}\cdot\mathtt{o}^{\mathcal{I}}\cdot\ddot{\mathtt{o}}^{\mathcal{I}}\cdot\mathtt{c}^{\mathcal{I}}$ and the word $d^{\mathcal{A}}_{\mathtt{q}_1,\mathtt{q}_2}\cdot rt^{\mathcal{A}}_{\mathtt{q}_2,\mathtt{q}_3}\cdot by^{\mathcal{A},\mathtt{o},\ddot{\mathtt{o}}}_{\mathtt{q}_2,\mathtt{q}_3}\cdot i^{\mathcal{A},\ddot{\mathtt{o}}}_{\mathtt{q}_4,(\mathtt{q}_5)_{☠}}$ (depicted above).

The following Lemma 7.39 reveals the desired property of $\mathcal{B}$. Its proof relies on either (a) shortening a path $\rho \models \mathcal{A}_{\mathtt{q},\mathtt{q}'}$ to its subsequence composed of all named elements, or (b) replacing any two consecutive elements in $\rho \models \mathcal{B}_{\mathtt{q},\mathtt{q}'}$ with the corresponding paths guaranteed by the roles from Definition 7.36.

Lemma 7.39 Let $\mathcal{A}$ be an NFA with the corresponding $\mathcal{A}$-guided $\mathcal{B}$, and let $\mathcal{I}$ be an $(\mathbf{N_I^{\mathcal{A}}}, \mathbf{N_I^{\mathcal{T}}})$-quasi-forest that properly interprets $\mathbf{C}(\mathbf{N_I^{\mathcal{T}}}, \mathcal{A})$ and $\mathbf{R}(\mathbf{N_I^{\mathcal{T}}}, \mathcal{A})$. For all states $\mathtt{q}, \mathtt{q}'$ of $\mathcal{A}$ we have the equality: $(\mathrm{Root} \sqcap \exists\mathcal{A}_{\mathtt{q},\mathtt{q}'}.\top)^{\mathcal{I}} = (\mathrm{Root} \sqcap (\exists\mathcal{B}_{\mathtt{q},\mathtt{q}'}.\top \sqcup \exists\mathcal{B}_{\mathtt{q},\mathtt{q}'_{☠}}.\top))^{\mathcal{I}}$.

For brevity, let LHS $:= (\mathrm{Root} \sqcap \exists\mathcal{A}_{\mathtt{q},\mathtt{q}'}.\top)^{\mathcal{I}}$, and RHS $:= (\mathrm{Root} \sqcap (\exists\mathcal{B}_{\mathtt{q},\mathtt{q}'}.\top \sqcup \exists\mathcal{B}_{\mathtt{q},\mathtt{q}'_{☠}}.\top))^{\mathcal{I}}$. We establish the equality of LHS and RHS by proving two concept inclusions.

Proof of LHS $\subseteq$ RHS. Take an element $\mathrm{d} \in$ LHS, and let $\rho := \rho_1 \ldots \rho_n$ be any path witnessing $\mathrm{d} \in (\exists\mathcal{A}_{\mathtt{q},\mathtt{q}'}.\top)^{\mathcal{I}}$. Note that d is named, and ρ starts from d. As $\rho \models \mathcal{A}_{\mathtt{q},\mathtt{q}'}$, there is a word $w := r_1 \ldots r_{n-1}$ and a sequence of states $\mathtt{q}_1, \ldots, \mathtt{q}_n$ such that $\mathtt{q}_0 = \mathtt{q}$, $\mathtt{q}_n = \mathtt{q}'$, and for all $i < n$ we have: $(\mathtt{q}_i, r_i, \mathtt{q}_{i+1}) \in \mathtt{T}$, where $\mathtt{T}$ is the transition relation of $\mathcal{A}$. Consider a subsequence ϱ of ρ composed of all roots of $\mathcal{I}$ that appear in ρ, in order of their appearance. Given an index $1 \leq i \leq |\varrho|$, let ι_i denote the corresponding index of ϱ_i in ρ. Now observe that the following properties hold for all $1 \leq i \leq |\varrho|$:

- The path $\rho_{\iota_i} \ldots \rho_{\iota_{(i+1)}}$ is basic, and it realises $\mathcal{A}_{\mathtt{q}_{\iota_i},\mathtt{q}_{\iota_{(i+1)}}}$.
- There exists a role name $s_i \in \mathbf{R}(\mathbf{N_I^{\mathcal{T}}}, \mathcal{A})$ such that s_i is indexed by the ordered pair of states $(\mathtt{q}_{\iota_i}, \mathtt{q}_{\iota_{(i+1)}})$, does not carry ☠, and $(\rho_{\iota_i}, \rho_{\iota_{(i+1)}}) \in (s_i)^{\mathcal{I}}$. This follows by the above item and tables in Definitions 7.33–7.36. Employing Observation 7.27, it suffices to consider the following cases:
  - If $\rho_{\iota_i} \ldots \rho_{\iota_{(i+1)}}$ is $(\mathtt{a},\mathtt{b})$-direct then $s_i = d^{\mathcal{A}}_{\mathtt{q}_{\iota_i},\mathtt{q}_{\iota_{(i+1)}}}$.
  - If $\rho_{\iota_i} \ldots \rho_{\iota_{(i+1)}}$ is an $\mathtt{a}$-roundtrip then $s_i = rt^{\mathcal{A}}_{\mathtt{q}_{\iota_i},\mathtt{q}_{\iota_{(i+1)}}}$.
  - If $\rho_{\iota_i} \ldots \rho_{\iota_{(i+1)}}$ is an $(\mathtt{a},\mathtt{o})$-inout then $s_i = io^{\mathcal{A},\mathtt{o}}_{\mathtt{q}_{\iota_i},\mathtt{q}_{\iota_{(i+1)}}}$.
  - If $\rho_{\iota_i} \ldots \rho_{\iota_{(i+1)}}$ is an $(\mathtt{a},\mathtt{o})$-outin then $s_i = oi^{\mathcal{A},\mathtt{o}}_{\mathtt{q}_{\iota_i},\mathtt{q}_{\iota_{(i+1)}}}$.
  - If $\rho_{\iota_i} \ldots \rho_{\iota_{(i+1)}}$ is an $(\mathtt{a},\mathtt{o},\ddot{\mathtt{o}})$-bypass then $s_i = by^{\mathcal{A},\mathtt{o},\ddot{\mathtt{o}}}_{\mathtt{q}_{\iota_i},\mathtt{q}_{\iota_{(i+1)}}}$.
- Thus $\rho_{\iota_i}\rho_{\iota_{(i+1)}} = \varrho_i\varrho_{i+1}$ realises $\mathcal{B}_{\mathtt{q}_{\iota_i},\mathtt{q}_{\iota_{(i+1)}}}$.

Invoking induction, we see establish that ϱ realises $\mathcal{B}_{\mathtt{q}_1,\mathtt{q}_{\iota_{|\varrho|}}}$. Observe that if the last element of ρ is the root of $\mathcal{I}$, then $\mathtt{q}_{\iota_{|\varrho|}} = \mathtt{q}_n$. This implies that $\varrho \models \mathcal{B}_{\mathtt{q},\mathtt{q}'}$, which entails $\mathrm{d} \in (\exists\mathcal{B}_{\mathtt{q},\mathtt{q}'}.\top)^{\mathcal{I}}$. Otherwise, we show that $\mathrm{d} \in (\exists\mathcal{B}_{\mathtt{q},\mathtt{q}'_{\skull}}.\top)^{\mathcal{I}}$. Note that one of the two options holds:

- There exist names $\mathsf{o} \in \mathbf{N}_{\mathbf{I}}^{\mathcal{T}}$, and $\mathsf{a} \in (\mathbf{N}_{\mathbf{I}}^{\mathcal{A}} \cup \mathbf{N}_{\mathbf{I}}^{\mathcal{T}})$ for which $\rho_{\iota_{|\varrho|}} = \mathsf{o}^{\mathcal{I}}$ holds and ρ_n is a descendant of $\mathsf{a}^{\mathcal{I}}$. Then the path $\rho_{\iota_{|\varrho|}} \dots \rho_n$ is (a,o)-inner and realises $\mathcal{A}_{\mathtt{q}_{\iota_{|\varrho|}},\mathtt{q}'}$. Hence, $(\mathsf{o}^{\mathcal{I}},\mathsf{a}^{\mathcal{I}}) \in (i^{\mathcal{A},\mathsf{o}}_{\mathtt{q}_{\iota_{|\varrho|}},\mathtt{q}'_{\skull}})^{\mathcal{I}}$, and the path $\varrho{\cdot}\mathsf{a}^{\mathcal{I}}$ realises $\mathcal{B}_{\mathtt{q},\mathtt{q}'_{\skull}}$.
- There exists a name $\mathsf{a} \in (\mathbf{N}_{\mathbf{I}}^{\mathcal{A}} \cup \mathbf{N}_{\mathbf{I}}^{\mathcal{T}})$ for which $\rho_{\iota_{|\varrho|}} = \mathsf{a}^{\mathcal{I}}$ and ρ_n is a descendant of $\mathsf{a}^{\mathcal{I}}$. Then the path $\rho_{\iota_{|\varrho|}} \dots \rho_n$ is a-inner and realises $\mathcal{A}_{\mathtt{q}_{\iota_{|\varrho|}},\mathtt{q}'}$. Hence, $(\mathsf{a}^{\mathcal{I}},\mathsf{a}^{\mathcal{I}}) \in (i^{\mathcal{A}}_{\mathtt{q}_{\iota_{|\varrho|}},\mathtt{q}'_{\skull}})^{\mathcal{I}}$, and the path $\varrho{\cdot}\mathsf{a}^{\mathcal{I}}$ realises $\mathcal{B}_{\mathtt{q},\mathtt{q}'_{\skull}}$.

Hence, $\mathrm{d} \in (\exists\mathcal{B}_{\mathtt{q},\mathtt{q}'_{\skull}}.\top)^{\mathcal{I}}$, which finishes the proof. □

Proof of RHS $\subseteq$ LHS. Let the *fusion* $\rho \odot \varrho$ of paths $\rho \coloneqq \rho_1 \dots \rho_n$ and $\varrho \coloneqq \varrho_1 \dots \varrho_m$ with $\rho_n = \varrho_1$ as the path $\rho_1 \dots \rho_n \varrho_2 \dots \varrho_m$ (the first element of ϱ is omitted). By Definitions 7.33–7.36 we know that for all role names $r \in \mathbf{R}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}},\mathcal{A})$ carrying the ordered pair of states $(\mathtt{q},\mathtt{q}')$ we have that:

- If r does not carry $\skull$, then for all $(\mathrm{d},\mathrm{e}) \in r^{\mathcal{I}}$ there exists a path $\mathrm{d}{\cdot}\rho{\cdot}\mathrm{e}$ realising $\mathcal{A}_{\mathtt{q},\mathtt{q}'}$.
- If r carries $\skull$, then for all $(\mathrm{d},\mathrm{e}) \in r^{\mathcal{I}}$ there exists a path $\mathrm{d}{\cdot}\rho{\cdot}\mathrm{e}'$ realising $\mathcal{A}_{\mathtt{q},\mathtt{q}'}$ for some descendant e' of e.

Given such a role name r and a pair $(\mathrm{d},\mathrm{e}) \in r^{\mathcal{I}}$, we fix one such corresponding path ρ and write $\mathrm{unfold}(\mathrm{d},r,\mathrm{e})$ to denote it.

Take $\mathrm{d} \in$ RHS and let $\rho \coloneqq \rho_1 \dots \rho_n$ be the path witnessing that d is either in $(\exists\mathcal{B}_{\mathtt{q},\mathtt{q}'}.\top)^{\mathcal{I}}$ or $(\exists\mathcal{B}_{\mathtt{q},\mathtt{q}'_{\skull}}.\top))^{\mathcal{I}}$. This provides us a word $w \coloneqq r_1 \dots r_{n-1}$ and a sequence of states $\mathtt{q}_1, \dots, \mathtt{q}_n$ such that $\mathtt{q}_0 = \mathtt{q}$, $\mathtt{q}_n = \mathtt{q}'$ (or $\mathtt{q}'_{\skull}$), and for all $i < n$ we have: $(\mathtt{q}_i, r_i, \mathtt{q}_{i+1}) \in \mathtt{T}$, where $\mathtt{T}$ is the transition relation of $\mathcal{B}$. Observe that only $\mathtt{q}_n$ may carry $\skull$ by the design of $\mathcal{B}$. This implies that for any two consecutive elements ρ_i, ρ_{i+1} we have that $\mathrm{unfold}(\rho_i, r_i, \rho_{i+1})$ starts from ρ_i and ends at ρ_{i+1}. Moreover, $\mathrm{unfold}(\rho_i, r_i, \rho_{i+1})$ realises $\mathcal{A}_{\mathtt{q}_i,\mathtt{q}_{i+1}}$. Thus, by induction, the path $\mathrm{unfold}(\rho_1, r_1, \rho_2) \odot \dots \odot \mathrm{unfold}(\rho_{n-1}, r_{n-1}, \rho_n)$ realises $\mathcal{A}_{\mathtt{q},\mathtt{q}'}$, and establishes $\mathrm{d} \in$ RHS. □

With the proviso that a quasi-forest $\mathcal{I}$ properly interprets both $\mathbf{C}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}},\mathcal{A})$ and $\mathbf{R}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}},\mathcal{A})$, Lemma 7.39 guarantees the desired interpretation of reachability concepts introduced at the beginning of the section.

Definition 7.40 Let $\mathcal{I}, \mathcal{A}, \mathcal{B}$ be as in Lemma 7.39, and $\mathtt{Q}$ be the state-set of $\mathcal{A}$. The set of **$\mathcal{A}$-reachability-concepts $\mathbf{C}_{\rightsquigarrow}(\mathcal{A})$** is $\{\mathrm{Reach}^{\mathcal{A}}_{\mathtt{q},\mathtt{q}'} \mid \mathtt{q},\mathtt{q}' \in \mathtt{Q}\}$. $\mathcal{I}$ **properly interprets** $\mathbf{C}_{\rightsquigarrow}(\mathcal{A})$ if

$$(\mathrm{Reach}^{\mathcal{A}}_{\mathtt{q},\mathtt{q}'})^{\mathcal{I}} = (\mathrm{Root} \sqcap (\exists\mathcal{B}_{\mathtt{q},\mathtt{q}'}.\top \sqcup \exists\mathcal{B}_{\mathtt{q},\mathtt{q}'_{\skull}}.\top))^{\mathcal{I}}$$

for all states $\mathtt{q},\mathtt{q}' \in \mathtt{Q}$. We call $\mathcal{I}$ **virtually $(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}},\mathcal{A})$-decorated** if it properly interprets $\mathbf{R}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}},\mathcal{A})$, and $\mathbf{C}_{\rightsquigarrow}(\mathcal{A})$, and **$(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}},\mathcal{A})$-decorated** if it additionally properly interprets $\mathbf{C}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}},\mathcal{A})$.

Revisit Lemma 7.32 for the definition of the $\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}$-relativisation $\mathrm{rel}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}},\mathcal{A}_{\mathtt{q},\mathtt{q}'})$ of $\exists\mathcal{A}_{\mathtt{q},\mathtt{q}'}.\top$, and recall that a quasi-forest virtually satisfies $\exists\mathcal{A}_{\mathtt{q},\mathtt{q}'}.\top$ if it satisfies its relativisation. Based on Lemma 7.32 we infer:

Lemma 7.41 Let $\mathcal{A}$ be an NFA, and let $\mathcal{I}$ be an $(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}},\mathcal{A})$-decorated $(\mathbf{N}_{\mathbf{I}}^{\mathcal{A}},\mathbf{N}_{\mathbf{I}}^{\mathcal{T}})$-quasi-forest. For all elements d in $\mathcal{I}$, we have that if d virtually satisfies $\exists\mathcal{A}_{\mathtt{q},\mathtt{q}'}.\top$ then d satisfies $\exists\mathcal{A}_{\mathtt{q},\mathtt{q}'}.\top$.

Proof. By Lemma 7.32 it suffices to show that $\mathcal{I}$ interprets all the concepts $\mathrm{Reach}^{\mathcal{A}}_{\mathtt{q},\mathtt{q}'}$ from $\mathbf{C}_{\rightsquigarrow}(\mathcal{A})$ equally to $\mathrm{Root} \sqcap \exists\mathcal{A}_{\mathtt{q},\mathtt{q}'}.\top$. But as $\mathcal{I}$ is $(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}},\mathcal{A})$-decorated, this follows by Lemma 7.39. □

We conclude by showing an algorithmic result concerning (virtually) $(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}},\mathcal{A})$-decorated quasi-forests, which relies on the fact that the evaluation of regular path queries over databases can be done in PTime.

Lemma 7.42 Let $\mathcal{I}$ be a finite interpretation with $\mathrm{Root}^{\mathcal{I}} = \Delta^{\mathcal{I}}$, and $\mathcal{A}$ be an NFA. We can then verify in time polynomial w.r.t. $(|\mathcal{A}| \cdot |\mathcal{I}|)$ whether $\mathcal{I}$ is virtually $(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \mathcal{A})$-decorated.

Proof. Testing whether $\mathcal{I}$ properly interprets $\mathbf{R}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \mathcal{A})$ can be clearly done in time polynomial w.r.t. $(|\mathcal{A}| \cdot |\mathcal{I}|)$, simply by analysing the shape of $\mathcal{I}$ and the transition relation of $\mathcal{A}$ (in the case of direct paths). To verify that $\mathcal{I}$ properly interprets $\mathbf{C}_{\rightsquigarrow}(\mathcal{A})$, we consider any concept $\mathrm{Reach}^{\mathcal{A}}_{\mathsf{q},\mathsf{q}'}$ from $\mathbf{C}_{\rightsquigarrow}(\mathcal{A})$ and an element $\mathrm{d} \in \Delta^{\mathcal{I}}$. Then we iterate through all the elements $\mathrm{e} \in \Delta^{\mathcal{I}}$ and we verify whether for some e either $\mathcal{I} \models \mathcal{B}_{\mathsf{q},\mathsf{q}'}(\mathsf{a}, \mathsf{b})$ or $\mathcal{I} \models \mathcal{B}_{\mathsf{q},\mathsf{q}'_{\skull}}(\mathsf{a}, \mathsf{b})$ hold (where a and b are, respectively, the individual names for d and e). This can be achieved in time polynomial w.r.t. $|\mathcal{I}| \cdot |\mathcal{A}|$ (as $\mathcal{B}$ is of size polynomial w.r.t. $|\mathcal{A}|$) with the classical methods of evaluating regular path queries over databases [MW95, Lemma 3.1]. If some of the RPQs is satisfied, we check if d is labelled with $\mathrm{Reach}^{\mathcal{A}}_{\mathsf{q},\mathsf{q}'}$ (and if not, we reject the input). If none of the RPQs is satisfied but d is labelled with $\mathrm{Reach}^{\mathcal{A}}_{\mathsf{q},\mathsf{q}'}$ we also reject the input. After reaching the end of the loop, we accept the input. This concludes the algorithm. $\square$

Lemma 7.42 tells us that if $\mathcal{I}$ is the clearing of some $(\mathbf{N}_{\mathbf{I}}^{\mathcal{A}}, \mathbf{N}_{\mathbf{I}}^{\mathcal{T}})$-quasi-forest $\mathcal{J}$ that properly interprets $\mathbf{C}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \mathcal{A})$-concepts, then we can verify if $\mathcal{J}$ is $(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \mathcal{A})$-decorated in PTime.

7.5 Counting decorations

We want to “relativise” number restrictions in the presence of nominals, so that in the suitably decorated quasi-forest models, the satisfaction of concepts of the form $(\geq n\ r).\top$ by the clearing can be decided solely based on the decoration of the clearing. Observe that for a given a root d of a quasi-forest $\mathcal{I}$ and a role name r, the set of r-successors of d can be divided into three groups: (a) the clearing, (b) the children of d, and (c) the descendants of roots (but only in case d is a nominal). To relativise counting, we decorate each element of the clearing with the total number of their r-successors in categories (a) and (b), as well as the information, for each nominal o, on (c) how many of their descendants are r-successors of o.

Definition 7.43 Fix $r \in \mathbf{N_R}$, $n \in \mathbb{N}$, and a finite $\mathbf{N}_{\mathbf{I}}^{\mathcal{T}} \subseteq \mathbf{N_I}$. The set of **$(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \geq n\ r)$-counting-concepts $\mathbf{C}_{\#}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \geq n\ r)$** consists of concept names $\mathrm{Clrng}^{r}_{\mathsf{t}}$, $\mathrm{Chld}^{r}_{\mathsf{t}}$, $\mathrm{Des}^{r,\mathsf{o}}_{\mathsf{t}}$ for $\mathsf{o} \in \mathbf{N}_{\mathbf{I}}^{\mathcal{T}}$ and thresholds t of the form “$=m$” for $0 \leq m \leq n$ or “$\geq n+1$”. All integers appearing in thresholds are encoded in binary. An $(\mathbf{N}_{\mathbf{I}}^{\mathcal{A}}, \mathbf{N}_{\mathbf{I}}^{\mathcal{T}})$-quasi-forest $\mathcal{I}$ is **$(\geq n\ r)$-semi-decorated** if all roots d have unique thresholds $\mathsf{t}^{\mathrm{d}}_{\mathrm{cl}}$, $\mathsf{t}^{\mathrm{d}}_{\mathrm{ch}}$, and $\mathsf{t}^{\mathrm{d}}_{\mathsf{o}}$ for all $\mathsf{o} \in \mathbf{N}_{\mathbf{I}}^{\mathcal{T}}$, for which d satisfies the concept

$$\mathrm{Clrng}^{r}_{\mathsf{t}^{\mathrm{d}}_{\mathrm{cl}}} \sqcap \mathrm{Chld}^{r}_{\mathsf{t}^{\mathrm{d}}_{\mathrm{ch}}} \sqcap \textstyle\bigsqcap_{\mathsf{o} \in \mathbf{N}_{\mathbf{I}}^{\mathcal{T}}} \mathrm{Des}^{r,\mathsf{o}}_{\mathsf{t}^{\mathrm{d}}_{\mathsf{o}}}.$$

The above concepts are called the **$(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \geq n\ r)$-descriptions of** d. We stress that their size is polynomial w.r.t. $|\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}| \cdot \log_2(n)$.

Similarly to the previous section, we next introduce the notion of proper satisfaction. Additional intuitions regarding Definition 7.43 are provided in Example 7.46. We strongly encourage the reader to read it.

Definition 7.44 An $(\mathbf{N}_{\mathbf{I}}^{\mathcal{A}}, \mathbf{N}_{\mathbf{I}}^{\mathcal{T}})$-quasi-forest $\mathcal{I}$ **properly interprets** $\mathbf{C}_{\#}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \geq n\ r)$ if $\mathcal{I}$ is $(\geq n\ r)$-semi-decorated and for all roots $\mathsf{a}^{\mathcal{I}}$ of $\mathcal{I}$ and $\mathsf{o} \in \mathbf{N}_{\mathbf{I}}^{\mathcal{T}}$, the root $\mathsf{a}^{\mathcal{I}}$ belongs to:

- $(\mathrm{Clrng}^{r}_{\mathsf{t}})^{\mathcal{I}}$ if and only if $|\{\mathrm{d} \colon \mathrm{d} \in \mathrm{Root}^{\mathcal{I}}, (\mathsf{a}^{\mathcal{I}}, \mathrm{d}) \in r^{\mathcal{I}}\}|\ \mathsf{t}$,
- $(\mathrm{Chld}^{r}_{\mathsf{t}})^{\mathcal{I}}$ if and only if $|\{\mathrm{d} \colon (\mathsf{a}^{\mathcal{I}}, \mathrm{d}) \in r^{\mathcal{I}}, (\mathsf{a}^{\mathcal{I}}, \mathrm{d}) \in \mathit{child}^{\mathcal{I}}\}|\ \mathsf{t}$,
- $(\mathrm{Des}^{r,\mathsf{o}}_{\mathsf{t}})^{\mathcal{I}}$ if and only if $|\{\mathrm{d} \colon (\mathsf{o}^{\mathcal{I}}, \mathrm{d}) \in r^{\mathcal{I}}, (\mathsf{a}^{\mathcal{I}}, \mathrm{d}) \in (\mathit{child}^{+})^{\mathcal{I}}\}|\ \mathsf{t}$.

In this case we also say that $\mathcal{I}$ is **$(\geq n\ r)$-decorated**.

Based on the above definitions, we can easily express the intended behaviour of $\mathbf{C}_{\#}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, {\geq}n\ r)$-concepts in $\mathcal{ZOIQ}$, using $(=n\ r).\mathrm{C}$ as an abbreviation of $({\geq}n\ r).\mathrm{C} \sqcap \neg({\geq}n{+}1\ r).\mathrm{C}$. As an example: in the most difficult case we describe $(\mathrm{Des}^{r,\mathtt{o}}_{=42})^{\mathcal{I}}$ with $\{\text{ᗣ}\} \sqcap \exists edge^{*}.\big(\{\mathtt{o}\} \sqcap (=42\ r).[\exists(child^{-})^{+}.\{\text{ᗣ}\}]\big)$.

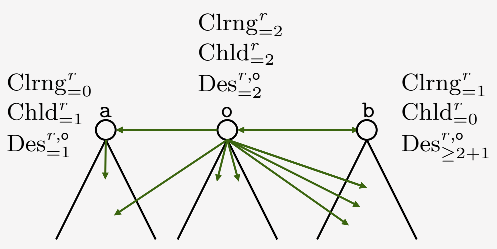
Lemma 7.45 For every $(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, {\geq}n\ r)$-description C there is a $\mathcal{ZOIQ}$-concept desc(C) (of size polynomial w.r.t. $|\mathrm{C}|$) that uses only individual names from $\mathbf{N}_{\mathbf{I}}^{\mathcal{T}} \cup \{\text{ᗣ}\}$ for a fresh ᗣ, such that for all $(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, {\geq}n\ r)$-semi-decorated $(\mathbf{N}_{\mathbf{I}}^{\mathcal{A}}, \mathbf{N}_{\mathbf{I}}^{\mathcal{T}})$-quasi-forests $\mathcal{I}$: $\mathcal{I}$ is $({\geq}n\ r)$-decorated if and only if for every root $\mathtt{a}^{\mathcal{I}}$ of $\mathcal{I}$ and its $(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, {\geq}n\ r)$-description C we have $\mathtt{a}^{\mathcal{I}} \in (\mathrm{desc}(\mathrm{C})[\text{ᗣ}/\mathtt{a}])^{\mathcal{I}}$.

Proof. The proof of the lemma relies on a straightforward translation of an $(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, r, n)$-description and Definition 7.44 into number restrictions, as follows:

- $\mathrm{Clrng}^{r}_{\mathfrak{t}}$ can be expressed with $\{\text{ᗣ}\} \sqcap (\mathfrak{t}\ r).\mathrm{Root}$.
- $\mathrm{Chld}^{r}_{\mathfrak{t}}$ can be expressed with $\{\text{ᗣ}\} \sqcap (\mathfrak{t}\ (r \cap child)).\top$.
- $\mathrm{Des}^{r,\mathtt{o}}_{\mathfrak{t}}$ can be expressed with $\{\text{ᗣ}\} \sqcap \exists edge^{*}.\big(\{\mathtt{o}\} \sqcap (\mathfrak{t}\ r).[\exists(child^{-})^{+}.\{\text{ᗣ}\}]\big)$.

The desired concept desc(C) is obtained by taking the conjunction of the above concepts. □

Example 7.46. Let us consider an $(\{\mathtt{a}, \mathtt{b}\}, \{\mathtt{o}\})$-quasi-forest $\mathcal{I}$ sketched below and a role name r depicted as a green arrow. As suggested by the drawing, (i) $\mathtt{a}^{\mathcal{I}}$ has no r-successors among the clearing of $\mathcal{I}$ and precisely one r-successor among its children, (ii) $\mathtt{o}^{\mathcal{I}}$ has two r-successors among the clearing of $\mathcal{I}$, it has precisely two r-successor among its children, one r-successor that is a descendant of $\mathtt{a}^{\mathcal{I}}$, and three r-successors that are descendants of $\mathtt{b}^{\mathcal{I}}$, and (iii) $\mathtt{b}^{\mathcal{I}}$ has precisely one r-successor inside the clearing of $\mathcal{I}$ and no other r-successors.


$\mathrm{Clrng}^{r}_{=2}$
$\mathrm{Chld}^{r}_{=2}$
$\mathrm{Des}^{r,\mathtt{o}}_{=2}$
$\mathrm{Clrng}^{r}_{=0}$
$\mathrm{Chld}^{r}_{=1}$
$\mathrm{Des}^{r,\mathtt{o}}_{=1}$
a
o
b
$\mathrm{Clrng}^{r}_{=1}$
$\mathrm{Chld}^{r}_{=0}$
$\mathrm{Des}^{r,\mathtt{o}}_{\geq 2+1}$

Suppose now that $\mathcal{I}$ is $(\{\mathtt{o}\}, {\geq}2\ r)$-decorated. This implies:

- $\mathtt{a}^{\mathcal{I}} \in (\mathrm{Clrng}^{r}_{=0} \sqcap \mathrm{Chld}^{r}_{=1} \sqcap \mathrm{Des}^{r,\mathtt{o}}_{=1})^{\mathcal{I}}$.
- $\mathtt{o}^{\mathcal{I}} \in (\mathrm{Clrng}^{r}_{=2} \sqcap \mathrm{Chld}^{r}_{=2} \sqcap \mathrm{Des}^{r,\mathtt{o}}_{=2})^{\mathcal{I}}$.
- $\mathtt{b}^{\mathcal{I}} \in (\mathrm{Clrng}^{r}_{=1} \sqcap \mathrm{Chld}^{r}_{=0} \sqcap \mathrm{Des}^{r,\mathtt{o}}_{\geq 2+1})^{\mathcal{I}}$.

Lemma 7.45 can be seen an analogue of Lemma 7.34, namely it rephrases the notion of proper satisfaction in the language of $\mathcal{ZOIQ}$-concepts. Call a quasi-forest $\mathcal{I}$ **virtually $(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, {\geq}n\ r)$-decorated** if $\mathcal{I}$ is $(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, {\geq}n\ r)$-semi-decorated, and if it properly interprets concepts of the form $\mathrm{Clrng}^{r}_{\mathfrak{t}}$. We have:

Lemma 7.47 For a finite interpretation $\mathcal{I}$ we can decide in polynomial time w.r.t. $|\mathcal{I}|$ whether $\mathcal{I}$ is virtually $(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, {\geq}n\ r)$-decorated.

Proof. It suffices to verify that every element in $\mathrm{Root}^{\mathcal{I}}$ is labelled by precisely one concept of the form (a) $\mathrm{Clrng}^{r}_{\mathfrak{t}}$, (b) $\mathrm{Chld}^{r}_{\mathfrak{t}}$, and (c) $\mathrm{Des}^{r,\mathtt{o}}_{\mathfrak{t}}$ (for all nominals $\mathtt{o} \in \mathbf{N}_{\mathbf{I}}^{\mathcal{T}}$). This can clearly be done in PTime, by analysing the representation of $\mathcal{I}$ stored in the memory. To verify that the concepts of the form $\mathrm{Clrng}^{r}_{\mathfrak{t}}$ are properly interpreted by $\mathcal{I}$, we iterate over all elements $\mathrm{d} \in \mathrm{Root}^{\mathcal{I}}$ and by iterating over $r^{\mathcal{I}}$ we count the total number of r-successors of d that belong to $\mathrm{Root}^{\mathcal{I}}$. Afterwards we compare such a number with a given threshold $\mathfrak{t}$. This is again, easily implementable in polynomial time. □

We employ the numbers indicated in all relevant thresholds from $(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, {\geq}n\ r)$-descriptions of the roots of $({\geq}n\ r)$-decorated quasi-forests to decide their satisfaction of number restriction. We first describe it with an example. Suppose that we want to verify if a non-nominal root d from $\mathcal{I}$ satisfies $(\geq 3\ r)$ based on its labels $\mathrm{Clrng}^{r}_{=1}$ and $\mathrm{Chld}^{r}_{\geq 3+1}$. It suffices to check if $1 + (3 + 1)$ is at least 3. For the nominals roots o, we additionally take all the concepts $\mathrm{Des}^{r,\mathsf{o}}_{\mathfrak{t}}$ into account, that are spread across the whole clearing. An element d from a finite $\mathcal{I}$ **virtually satisfies** $(\geq n\ r).\top$, whenever d satisfies $(\geq n\ r).\top$ in every $(\geq n\ r)$-decorated $(\mathbf{N}^{\mathcal{A}}_{\mathbf{I}}, \mathbf{N}^{\mathcal{T}}_{\mathbf{I}})$-quasi-forest with the clearing equal to $\mathcal{I}$. We show:

Lemma 7.48 For a finite virtually $(\mathbf{N}^{\mathcal{T}}_{\mathbf{I}}, \geq n\ r)$-decorated interpretation $\mathcal{I}$ with $\Delta^{\mathcal{I}} = \mathrm{Root}^{\mathcal{I}}$ and $\mathrm{d} \in \mathrm{Root}^{\mathcal{I}}$ we can test in time polynomial in $|\mathcal{I}|$ whether d virtually satisfies $(\geq n\ r).\top$.

Proof. Consider two cases:

- d is not a nominal. Then all r-successors of d belong to the clearing and to the children of d. By $(\geq n\ r)$-decoration of $\mathcal{I}$, there are unique thresholds $\mathfrak{t}^{\mathrm{d}}_{\mathrm{cl}}$, $\mathfrak{t}^{\mathrm{d}}_{\mathrm{ch}}$, for which d satisfies $\mathrm{Clrng}^{r}_{\mathfrak{t}^{\mathrm{d}}_{\mathrm{cl}}} \sqcap \mathrm{Chld}^{r}_{\mathfrak{t}^{\mathrm{d}}_{\mathrm{ch}}}$. Let m be the sum of the numbers appearing in $\mathfrak{t}^{\mathrm{d}}_{\mathrm{cl}}$, $\mathfrak{t}^{\mathrm{d}}_{\mathrm{ch}}$. Clearly m is the total number of r-successors of d. We then check if $m \geq n$ holds. This procedure can be clearly implemented in PTIME.
- d is a nominal, and let $\mathrm{d} = \mathsf{o}^{\mathcal{I}}$. Then all r-successors of d belong to the clearing and the subtrees of the clearing. Let $\mathrm{d}_1, \ldots, \mathrm{d}_k$ be all the elements of $\mathrm{Root}^{\mathcal{I}}$, and let $\mathfrak{t}^{\mathrm{d}_i}_{\mathsf{o}}$ be the unique thresholds so that d satisfies $\mathrm{Des}^{r;\mathsf{o}}_{\mathfrak{t}^{\mathrm{d}_i}_{\mathsf{o}}}$. Let m be the sum of all numbers appearing in $\mathfrak{t}^{\mathrm{d}_i}_{\mathsf{o}}$ plus the value in a unique threshold $\mathfrak{t}^{\mathrm{d}}_{\mathrm{cl}}$ for which d satisfies $\mathrm{Clrng}^{r}_{\mathfrak{t}^{\mathrm{d}}_{\mathrm{cl}}}$. Note that m is the total number of r-successors of d (children of d are also its descendants so we do not double-count them). We then check if $m \geq n$ holds. Similarly to the previous case, such a procedure can be clearly implemented in PTIME.

$\square$

7.6 Elegant Quasi-Forest Models and Their Summaries

In this section we benefit from various decorations introduced in the previous sections to design a succinct way of representing quasi-forest models of $\mathcal{ZOIQ}$-KBs, dubbed *summaries*. From now on we will focus only on KBs in Scott's normal form and on certain class of models introduced next.

Definition 7.49 Let $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ be a $\mathcal{ZOIQ}$-KB in Scott's normal form and let $\mathcal{I}$ be its model. We call $\mathcal{I}$ **elegant** if

(i) $\mathcal{I}$ is a canonical quasi-forest model of $\mathcal{K}$,

(ii) $\mathcal{I}$ is $(\mathsf{ind}(\mathcal{T}), \geq n\ r)$-decorated for all number restrictions $(\geq n\ r).\top$ from $\mathcal{T}$,

(iii) $\mathcal{I}$ is $(\mathsf{ind}(\mathcal{T}), \mathscr{A})$-decorated for all automata $\mathscr{A}$ from $\mathcal{T}$, and

(iv) $\mathcal{I}$ interprets all concept and role names that do not appear in $\mathcal{K}$, the set $\{\mathrm{Root}, \mathit{edge}, \mathit{child}, \mathit{id}\}$, and in mentioned decorations, as the empty set.

Invoking the normal form lemma from the Preliminaries, Lemma 7.5, as well as the definitions of proper interpretation, we establish the following property. Queries are only needed for the application section.

Lemma 7.50 For every $\mathcal{ZOIQ}$-TBox $\mathcal{T}$ we can compute in PTIME a $\mathcal{ZOIQ}$-TBox $\mathcal{T}'$ in Scott's normal form that possibly such that for every ABox $\mathcal{A}$ we have:

- $(\mathcal{A}, \mathcal{T})$ is quasi-forest satisfiable if and only if $(\mathcal{A}, \mathcal{T}')$ has an elegant model, and
- for every P2RPQ q using only concepts and roles present in $\mathcal{T}$ we have that $(\mathcal{A}, \mathcal{T})$ has a canonical quasi-forest model violating q if and only if $(\mathcal{A}, \mathcal{T}')$ has an elegant model violating q.

Proof. Note that in quasi-forests the universal role $\overline{\top}$ is interpreted equally to edge^{*}. Hence, w.l.o.g. we can assume that $\mathcal{T}$ does not contain $\overline{\top}$. We next invoke Lemma 2.24 and Lemma 2.25

to compute (in polynomial time) the desired $\mathcal{ZOIQ}$-TBox $\mathcal{T}'$ in Scott's normal form which is a conservative extension of $\mathcal{T}$.

Take an ABox $\mathcal{A}$, a query q and suppose that there is a quasi-forest model of $(\mathcal{A}, \mathcal{T})$ that violates q (in case we are interested in satisfiability only, we take $q := \bot$). By the semantics of $\mathcal{ZOIQ}$ we can assume w.l.o.g. that $\mathcal{I}$ interprets all concept and role names that are not special and that do not appear in $\mathcal{T}$ as empty sets. By the fact that $\mathcal{T}'$ is a conservative extension of $\mathcal{T}'$ we know that there is a quasi-forest model $\mathcal{I}'$ of $(\mathcal{A}, \mathcal{T}')$ that violates q ($\mathcal{I}'$ is just $\mathcal{I}$ with the altered interpretation of fresh concept and role names). We then alter the interpretation of concepts and roles that appear in automata and counting decorations, precisely in the way stated in Definition 7.33, Definition 7.36, Definition 7.40, and Definition 7.44. Call the resulting interpretation $\mathcal{I}''$. As all the concepts involved in decorations are not present in $\mathcal{A}$, $\mathcal{T}$, and q, and that $\mathcal{I}''$ and $\mathcal{I}$ are identical modulo the interpretation of fresh concepts and roles, we have that $\mathcal{I}''$ is a model of $(\mathcal{A}, \mathcal{T})$ that violates q. Hence, $\mathcal{I}''$ is the desired elegant model. □

Summaries are simply full descriptions of clearings of elegant models.

Definition 7.51 Let $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ be a $\mathcal{ZOIQ}$-KB in Scott's normal form. A **$\mathcal{K}$-summary $\mathcal{S}$** is any $\subseteq$-minimal ABox satisfying, for all names $\mathsf{a}, \mathsf{b} \in \mathsf{ind}(\mathcal{K})$, all the conditions listed below.

(I) $\mathcal{S}$ contains either $\mathsf{a} \approx \mathsf{b}$ or $\neg(\mathsf{a} \approx \mathsf{b})$.

(II) For all concept names A appearing in $\mathcal{K}$ we have that $\mathcal{S}$ contains either $\mathrm{A}(\mathsf{a})$ or $\neg\mathrm{A}(\mathsf{a})$.

(III) For all NFA $\mathscr{A}$ from $\mathcal{K}$, and all concept names A from $\mathbf{C}(\mathsf{ind}(\mathcal{T}), \mathscr{A}) \cup \mathbf{C}_{\leadsto}(\mathscr{A})$ we have that $\mathcal{S}$ contains either $\mathrm{A}(\mathsf{a})$ or $\neg\mathrm{A}(\mathsf{a})$.

(IV) For all role names r appearing in $\mathcal{K}$ we have that $\mathcal{S}$ contains either $r(\mathsf{a}, \mathsf{b})$ or $\neg r(\mathsf{a}, \mathsf{b})$.

(V) For all NFA $\mathscr{A}$ from $\mathcal{K}$, and all role names r from $\mathbf{R}(\mathsf{ind}(\mathcal{T}), \mathscr{A})$ we have that $\mathcal{S}$ contains either $r(\mathsf{a}, \mathsf{b})$ or $\neg r(\mathsf{a}, \mathsf{b})$ (note that a and b may be equal!).

(VI) For all number restrictions $(\geq n\ r).\top$ from $\mathcal{K}$, all $\mathsf{o} \in \mathsf{ind}(\mathcal{T})$, there are thresholds $\mathfrak{t}^{\mathsf{a}}_{\mathrm{cl}}$, $\mathfrak{t}^{\mathsf{a}}_{\mathrm{ch}}$, and $\mathfrak{t}^{\mathsf{a}}_{\mathsf{o}}$ in $\{=m, \geq n{+}1 \mid 0 \leq m \leq n\}$ for which $\mathcal{S}$ contains $\mathrm{Clrng}^{r}_{\mathfrak{t}^{\mathsf{a}}_{\mathrm{cl}}}(\mathsf{a})$, $\mathrm{Chld}^{r}_{\mathfrak{t}^{\mathsf{a}}_{\mathrm{ch}}}(\mathsf{a})$, and $\mathrm{Des}^{r,\mathsf{o}}_{\mathfrak{t}^{\mathsf{a}}_{\mathsf{o}}}(\mathsf{a})$.

(VII) $\mathrm{Root}(\mathsf{a}) \in \mathcal{S}$ and $\mathit{edge}(\mathsf{a}, \mathsf{b}) \in \mathcal{S}$.

If only a $\mathcal{ZOIQ}$-TBox $\mathcal{T}$ is given, we define **$\mathcal{T}$-ghost-summaries** as ({👻 ≈ 👻}, $\mathcal{T}$)-summaries.

Invoking the previously-established bounds on the number and sizes of concepts and roles occurring in decorations (*i.e.* the ones stated at the end of Definitions 7.33, 7.36, and 7.43), we infer:

Lemma 7.52 Let $\mathcal{K}$ and $\mathcal{T}$ be, respectively, a $\mathcal{ZOIQ}$-KB and a $\mathcal{ZOIQ}$-TBox in Scott's normal form. We have that every $\mathcal{K}$-summary is of size polynomial in $|\mathcal{K}|$. Moreover, there are exponentially (in $|\mathcal{T}|$) many $\mathcal{T}$-ghost-summaries and each of them is of size polynomial w.r.t. $|\mathcal{T}|$.

Proof. The proof goes via the analysis of axioms stated in Definition 7.51. We focus on the case of $\mathcal{K}$-summaries, as the case of $\mathcal{T}$-ghost-summaries is just a corollary. Observe that:

- Clearly, there are at most $|\mathsf{ind}(\mathcal{K})|^2 \leq |\mathcal{K}|^2$ axioms of the form (I) in a single summary, and at most $2^{|\mathsf{ind}(\mathcal{T})|} \leq 2^{|\mathcal{K}|}$ ways to select them.
- If $\mathcal{K}$ contains c different concept names, then there are at most $c \cdot |\mathsf{ind}(\mathcal{K})| \leq 2 \cdot |\mathcal{K}|^2$ axioms of the form (II) in a single summary, and at most $2^{c \cdot |\mathsf{ind}(\mathcal{K})|} \leq 2^{2|\mathcal{K}|^2}$ ways to select them. If $\mathcal{K}$ contains r different role names, then there are at most $4 \cdot r \cdot |\mathsf{ind}(\mathcal{K})| \leq 4 \cdot |\mathcal{K}|^2$ axioms of the form (IV) in a single summary, and at most $2^{4r \cdot |\mathsf{ind}(\mathcal{K})|} \leq 2^{4|\mathcal{K}|^2}$ ways to select them.
- For axioms of the form (III) and (V) we first fix an NFA $\mathscr{A}$, and observe that the sets $\mathbf{C}(\mathsf{ind}(\mathcal{T}), \mathscr{A})$, $\mathbf{C}_{\leadsto}(\mathscr{A})$, and $\mathbf{R}(\mathsf{ind}(\mathcal{T}), \mathscr{A})$ can be respectively bounded by $7 \cdot |\mathsf{ind}(\mathcal{T})| \cdot |\mathsf{Q}|^2$, $|\mathsf{Q}|^2$, and $7 \cdot |\mathsf{ind}(\mathcal{T})| \cdot |\mathsf{Q}|^2$, where Q is the state-set of $\mathscr{A}$. Hence, for a fixed NFA $\mathscr{A}$ and a fixed pair a, b of individual names there are at most $7 \cdot |\mathsf{ind}(\mathcal{T})| \cdot |\mathsf{Q}|^2 + |\mathsf{Q}|^2 + 7 \cdot |\mathsf{ind}(\mathcal{T})| \cdot |\mathsf{Q}|^2 \leq 15|\mathcal{K}|^3$ axioms of the form (III) and (V) in a single summary, and at most $2^{15|\mathcal{K}|^3}$ ways

to select them. Thus, the total number of axioms of the form (III) and (V) in a single summary can be bounded by $|\mathcal{K}| \cdot 4 \cdot |\mathsf{ind}(\mathcal{K})| \cdot 15|\mathcal{K}|^3 \leq 60|\mathcal{K}|^5$ (as there are at most $|\mathcal{K}|$ NFAs in $\mathcal{K}$, and at most $4 \cdot |\mathsf{ind}(\mathcal{K})|$ possible arguments to plug in), and they give rise to at most $2^{60|\mathcal{K}|^5}$ possible summaries.

- Finally, we discuss the axioms of the form (VI). Fix a number restriction $(\geq n\ r).\top$ and an individual name $\mathsf{a} \in \mathsf{ind}(\mathcal{K})$. Observe that every of $|\mathsf{ind}(\mathcal{T})|+2$ thresholds can be encoded on $\log_2(n)$-bits (as n is given in binary). Hence, for such a fixed number restriction, a single summary contains axioms of size bounded by $|\mathsf{ind}(\mathcal{K})| \cdot (|\mathsf{ind}(\mathcal{T})|+2) \cdot \log_2(n) \leq |\mathcal{K}|^3$, which gives raise to at most $2^{|\mathcal{K}|^3}$ possible selection of summaries. We replace $|\mathcal{K}|^3$ with $|\mathcal{K}|^4$ when considering all number restrictions that appear in $\mathcal{K}$.

By the rule of product, we conclude that every $\mathcal{K}$-summary is of size polynomial w.r.t. $|\mathcal{K}|$, and that there are exponentially many (w.r.t. $|\mathcal{K}|$) $\mathcal{K}$-summaries. The second part of the lemma follows from the fact that every $\mathcal{T}$-ghost-summary is a $\mathcal{K}$-summary for $\mathcal{K}$ of size linear in $|\mathcal{T}|$. □

While a $\mathcal{K}$-summary can be easily extracted from the clearing of any elegant model of $\mathcal{K}$, the converse direction requires several additional assumptions, dubbed *clearing-consistency* and *subtree-consistency*.

7.7 Consistent Summaries of Quasi-Forest Models

The notion of *clearing-consistency* ensures that a given summary $\mathcal{S}$ is a good candidate for the clearing of some elegant model of $\mathcal{K}$, namely (i) $\mathcal{S}$ does not violate "local" constraints from $\mathcal{K}$, (ii) $\mathcal{S}$ is virtually decorated for decorations involving number restrictions and automata, and that (iii) every element from $\mathcal{S}$ virtually satisfies all concepts involving automata or number restrictions from $\mathcal{K}$. In the definition below we write $\mathcal{I}_{\mathcal{S}}$ to denote the minimal interpretation (if exists) that corresponds to and satisfies the ABox $\mathcal{S}$.[3]

Definition 7.53 Let $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ be a $\mathcal{ZOIQ}$-KB in Scott's normal form and let $\mathcal{S}$ be a $\mathcal{K}$-summary. Let $\mathcal{T}_{\mathrm{loc}}$ be the TBox composed of all GCIs from $\mathcal{T}$ except for the ones concerning NFAs and number restrictions. We say that $\mathcal{S}$ is **clearing-consistent** if it satisfies all the conditions below.

- $\mathcal{I}_{\mathcal{S}}$ satisfies $(\mathcal{A}, \mathcal{T}_{\mathrm{loc}})$.
- For all GCIs $\mathrm{A} \equiv \exists\mathcal{A}_{\mathsf{q},\mathsf{q}'}.\top$ from $\mathcal{T}$ we have that (i) $\mathcal{I}_{\mathcal{S}}$ is virtually $(\mathsf{ind}(\mathcal{T}), \mathcal{A})$-decorated, and (ii) for all $\mathsf{a} \in \mathsf{ind}(\mathcal{K})$, $\mathrm{A}(\mathsf{a}) \in \mathcal{S}$ iff a virtually satisfies $\exists\mathcal{A}_{\mathsf{q},\mathsf{q}'}.\top$ in $\mathcal{I}_{\mathcal{S}}$ (namely $\mathrm{Reach}^{\mathcal{A}}_{\mathsf{q},\mathsf{q}'}(\mathsf{a}) \in \mathcal{S}$).
- For all GCIs $\mathrm{A} \equiv (\geq n\ r).\top$ from $\mathcal{T}$ we have that (i) $\mathcal{I}_{\mathcal{S}}$ is virtually $(\mathsf{ind}(\mathcal{T}), \geq n\ r)$-decorated, and (ii) for all $\mathsf{a} \in \mathsf{ind}(\mathcal{K})$, we have that $\mathrm{A}(\mathsf{a}) \in \mathcal{S}$ iff a virtually satisfies $(\geq n\ r).\top$ in $\mathcal{I}_{\mathcal{S}}$.

Based on the previously-established lemmas on testing virtual satisfaction of concepts (namely Lemma 7.42 and Lemma 7.48) we can conclude that deciding clearing consistency can be done in PTime.

Lemma 7.54 If $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ is a $\mathcal{ZOIQ}$-KB in Scott's normal form, and $\mathcal{S}$ is a $\mathcal{K}$-summary, we can decide in time polynomial w.r.t. $|\mathcal{K}| \cdot |\mathcal{S}|$ whether $\mathcal{S}$ is clearing consistent.

Proof. Note that the construction of $\mathcal{I}_{\mathcal{S}}$ can be done in time polynomial w.r.t. $|\mathcal{S}|$ (see the footnote describing its construction). Thus it suffices to analyse Definition 7.53 and see that each of the "bullets" can be verified in time polynomial w.r.t. $|\mathcal{K}| \cdot |\mathcal{S}|$. As $\mathcal{T}_{\mathrm{loc}}$ (via an obvious translation) belongs to the two-variable fragment of first-order logic, for which the model-checking problem is decidable in PTime [GO99, Prop. 4.2], the first item can be verified in PTime as desired. For the second bullet, we invoke Lemma 7.42, while for the last bullet we

[3]Formally, (a) the domain elements of $\mathcal{I}_{\mathcal{S}}$ are the equivalence classes $[\mathsf{a}]_{\approx}$ of names from $\mathsf{ind}(\mathcal{S})$ in the equality equivalence relation $\approx$, renamed afterwards to be positive integers, (b) we interpret all names $\mathsf{a} \in \mathsf{ind}(\mathcal{S})$ as $[\mathsf{a}]_{\approx}$ and all the remaining individual names as any arbitrary selected element, (c) for all concept names A, we have that $[\mathsf{a}]_{\approx}$ belongs to the interpretation of A if and only if $\mathrm{A}(\mathsf{a}) \in \mathcal{S}$, and (c) for all role names r we have that $([\mathsf{a}]_{\approx}, [\mathsf{b}]_{\approx})$ belongs to the interpretation of r if and only if $r(\mathsf{a}, \mathsf{b}) \in \mathcal{S}$. It is an easy exercise to see that whenever $\mathcal{S}$ is a summary then the interpretation $\mathcal{I}_{\mathcal{S}}$ is well-defined.

rely on Lemma 7.47 and Lemma 7.48. This concludes the proof. □

The second required notion is the *subtree-consistency*. To decide whether a given summary $\mathcal{S}$ extends to a model of $\mathcal{K}$, we need to check, for all elements d of $\mathcal{S}$, the existence of a suitable tree satisfying (a relativised) $\mathcal{K}$ and fulfilling all the premises given by the decorations of d. This is achieved by crafting a suitable $\mathcal{ZOIQ}$-KB and testing whether it has a quasi-forest model of a suitably bounded branching. To make such a KB dependent only on the TBox, we are going to use the ghost variable 👻 in place of the intended element d. In case the following definition is too difficult to read, please revisit Corollary 7.35 and the text right after the proof of Lemma 7.32 as well as Definition 7.43 and Lemma 7.45.

Definition 7.55 Let $\mathcal{T}$ be a $\mathcal{ZOIQ}$-TBox in Scott's normal form and let $\mathcal{S}$ be a $\mathcal{T}$-ghost-summary. We say that $\mathcal{S}$ is **$\mathcal{T}$-subtree-consistent** if the $\mathcal{ZOIQ}$-KB $\mathcal{K}_{\mathcal{S},👻} \coloneqq (\mathcal{S}, \mathcal{T}_{\text{loc}} \cup \mathcal{T}_{\text{aut}} \cup \mathcal{T}_{\text{cnt}})$ is quasi-forest satisfiable. The TBox $\mathcal{T}_{\text{loc}}$ is as in Definition 7.53, while $\mathcal{T}_{\text{aut}}$ and $\mathcal{T}_{\text{cnt}}$ are defined below.

- For all axioms $\mathrm{A} \equiv \exists \mathcal{A}_{\mathsf{q},\mathsf{q}'}.\top$ from $\mathcal{T}$, the TBox $\mathcal{T}_{\text{aut}}$ consists of:
$$\{👻\} \sqsubseteq \text{comdsc}(\mathsf{ind}(\mathcal{T}), \mathcal{A}), \quad \mathrm{A} \equiv \text{rel}(\mathsf{ind}(\mathcal{T}), \mathcal{A}_{\mathsf{q},\mathsf{q}'}).$$
- For all axioms of the form $\mathrm{A} \equiv (\geq n\ r).\top$ from $\mathcal{T}$, the TBox $\mathcal{T}_{\text{cnt}}$ consists of
$$\neg\text{Root} \sqcap \mathrm{A} \equiv \neg\text{Root} \sqcap (\geq n\ r).\top,$$
$$\{👻\} \sqsubseteq \text{desc}\left(\text{Chld}^{r}_{\mathsf{t}^{👻}_{\text{ch}}} \sqcap \textstyle\bigsqcap_{\mathsf{o} \in \mathsf{ind}(\mathcal{T})} \text{Des}^{r;\mathsf{o}}_{\mathsf{t}^{👻}_{\mathsf{o}}}\right),$$
for the unique concepts $\text{Chld}^{r}_{\mathsf{t}^{👻}_{\text{ch}}}$, $\text{Des}^{r;\mathsf{o}}_{\mathsf{t}^{👻}_{\mathsf{o}}}$ satisfied by 👻 in $\mathcal{I}_{\mathcal{S}}$.

The crucial property concerning the notion of $\mathcal{T}$-subtree-consistency and the set $\mathbf{S}_{\mathcal{T}}$ of all $\mathcal{T}$-subtree-consistent $\mathcal{T}$-ghost-summaries is given next. Its proof relies on the bounds on the concepts from Lemma 7.34, Lemma 7.32, and Definition 7.43, as well as the exponential time algorithm for deciding quasi-forest-satisfiability of $\mathcal{ZOIQ}$-KBs from Lemma 7.6.

Lemma 7.56 Let $\mathcal{T}$ be a $\mathcal{ZOIQ}$-TBox in Scott's normal form (constructed from some $\mathcal{ZOIQ}$-TBox $\mathcal{T}'$ with an algorithm described in Preliminaries). Then one can decide in time exponential w.r.t. $|\mathcal{T}|$ whether a given $\mathcal{T}$-ghost-summary is $\mathcal{T}$-subtree-consistent. Moreover, the set $\mathbf{S}_{\mathcal{T}}$ is of size exponential w.r.t. $|\mathcal{T}|$ and can be computed in time exponential w.r.t. $|\mathcal{T}|$.

Proof sketch. We first establish that $\mathcal{K}_{\mathcal{S},👻} \coloneqq (\mathcal{S}, \mathcal{T}_{\text{loc}} \cup \mathcal{T}_{\text{aut}} \cup \mathcal{T}_{\text{cnt}})$, defined in Definition 7.55, is of size polynomial w.r.t. $|\mathcal{T}|$. The fact that the ABox $|\mathcal{S}|$ is polynomial w.r.t. $|\mathcal{T}|$ follows from Lemma 7.52 (note that such a lemma also tell us that there are exponentially many of such summaries). Next, as $\mathcal{T}_{\text{loc}}$ is a subset of $\mathcal{T}$, its size is clearly polynomially bounded w.r.t. $|\mathcal{T}|$. In order to bound the size of $\mathcal{T}_{\text{aut}}$, we deal separately with each NFA $\mathcal{A}$ of $\mathcal{T}$ (note that there are at most $|\mathcal{T}|$ of them). By a combination of Lemma 7.34 and the last line of Definition 7.33 we see that comdsc and rel are of size polynomial w.r.t. $|\mathcal{T}|$. Finally, to provide the bound on the size of $\mathcal{T}_{\text{cnt}}$ we rely on Lemma 7.45. Hence, $\mathcal{K}_{\mathcal{S},👻}$ is of size polynomial w.r.t. $|\mathcal{T}|$. We can now replace all the NFAs present in $\mathcal{K}_{\mathcal{S},👻}$ with regular expressions in order to test its quasi-forest-satisfiability in time exponential w.r.t. $|\mathcal{T}|$ by Lemma 7.6 (this is needed as the original syntax of $\mathcal{ZOIQ}$ does not allow for automata in regular path expressions). While the usual transformation from a given NFA $\mathcal{A}$ into a regular expression $\mathcal{R}$ may lead to to $\mathcal{R}$ having the size exponential w.r.t. $|\mathcal{A}|$, we claim that this is not the case here. Indeed, by investigating the definition of $\mathcal{K}_{\mathcal{S},👻}$ we see NFA appearing in $\mathcal{K}_{\mathcal{S},👻}$ either come from the TBox $t\mathcal{T}$, or have the form $\mathcal{A}_{\mathsf{q},\mathsf{q}'}$ or $\mathcal{A}_{\mathsf{q},\mathsf{q}'} \bowtie \mathcal{B}$ for some NFA $\mathcal{A}$ from $\mathcal{T}$ and $\mathcal{B}$ describing some "category" of basic paths. Recall that in the construction of $\mathcal{A}$ in $\mathcal{T}$, we obtained it by transformation of some regular expression $\mathcal{R}$ present in $\mathcal{T}'$. Hence, we can simply replace $\mathcal{A}$ with the original regular expression $\mathcal{R}$ (which is clearly of polynomial size w.r.t. $|\mathcal{T}|$). In order to construct a regular expression corresponding to $\mathcal{A}_{\mathsf{q},\mathsf{q}'}$ under a proviso that a regular expression $\mathcal{R}$ (of size polynomial w.r.t. $|\mathcal{T}|$) corresponds to the NFA $\mathcal{A}$, we invoke the proof described by Hermann Gruber [Gru23] in his TCS StackExchange Post. Finally, to deal with the NFA of the form $\mathscr{A}_{\mathsf{q},\mathsf{q}'} \bowtie \mathscr{B}$ we first convert $\mathscr{A}_{\mathsf{q},\mathsf{q}'}$ into a regular expression and then invoke the second part of Lemma 7.30. Hence, the resulting KB indeed has size polynomial in $|\mathcal{T}|$.

Towards the construction of an algorithm enumerating $\mathcal{T}$-subtree-consistent $\mathcal{T}$-ghost-summaries, we first construct all $\mathcal{T}$-summaries by enumerating axioms from Definition 7.51 (enumerating thresholds in case of number restrictions) and then verifying whether the constructed ABoxes fulfil the definition of a $\mathcal{T}$-summary (which can be easily done in PTIME). Now it remains to construct suitable $\mathcal{T}_{\mathrm{loc}}$, $\mathcal{T}_{\mathrm{aut}}$, $\mathcal{T}_{\mathrm{cnt}}$. The construction of $\mathcal{T}_{\mathrm{loc}}$ is simply by dropping irrelevant axioms from $\mathcal{T}$. To construct $\mathcal{T}_{\mathrm{aut}}$, for every NFA $\mathscr{A}$ that appears in $\mathcal{T}$ we rely on Lemma 7.34 and the definitions of comdsc and rel. Finally, to construct $\mathcal{T}_{\mathrm{cnt}}$ we proceed with every number restriction from $\mathcal{T}$ and apply Lemma 7.45. This yields the polynomial-time construction of an arbitrary $\mathcal{K}_{\mathcal{S},\text{👻}}$, which quasi-forest satisfiability can be checked in time exponential w.r.t. $|\mathcal{T}|$. This concludes the proof. □

We now lift the definition of subtree-consistency from ghost summaries to arbitrary $\mathcal{K}$-summaries by producing a ghost summary per each individual name a mentioned in $\mathcal{K}$. The idea is simple: we first restrict $\mathcal{S}$ to nominals and the selected name a, and then replace a with the ghost variable 👻 (we must be a bit more careful if a is itself a nominal). This produces a ghost summary, for which the notion of subtree-consistency is already well-defined. The main benefit of testing subtree-consistency of such a summary is that this guarantees the existence of a quasi-forest containing a subtree rooted at 👻 that we can afterwards "plug in" to a in $\mathcal{S}$ in order to produce a full model of $\mathcal{K}$ from $\mathcal{S}$. This is formalised next.

Definition 7.57 Let $\mathcal{T}$ and $\mathbf{S}_{\mathcal{T}}$ be as in Lemma 7.56. For a $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ and a $\mathcal{K}$-summary $\mathcal{S}$ we say that $\mathcal{S}$ is **consistent** if $\mathcal{S}$ is clearing-consistent and for every name $\mathsf{a} \in \mathsf{ind}(\mathcal{K})$ the **$(\mathcal{T}, \mathcal{S}, \mathsf{a})$-ghost-summary** $\mathcal{S}_{\mathsf{a}}$ (as defined below) belongs to $\mathbf{S}_{\mathcal{T}}$.

We define $\mathcal{S}_{\mathsf{a}}$ as the sum of the following three ABoxes: (i) $\mathcal{S}$ restricted to $\mathsf{ind}(\mathcal{T})$, (ii) $\mathcal{S}$ restricted to $\mathsf{ind}(\mathcal{T}) \cup \{\mathsf{a}\}$ with all occurrences of a replaced with 👻, and (iii) $\{\mathsf{a} \approx \text{👻}, \text{👻} \approx \mathsf{a}\}$ whenever $\mathsf{a} \in \mathsf{ind}(\mathcal{T})$.

As we already said when introducing relevant notions of consistency, a model of a $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ can be obtained by "combining" a clearing-consistent summary $\mathcal{S}$ with relevant subtrees provided by the subtree-consistency of $(\mathcal{T}, \mathcal{S}, \mathsf{a})$-ghost-summaries for all $\mathsf{a} \in \mathsf{ind}(\mathcal{K})$. We obtain:

Lemma 7.58 A $\mathcal{ZOIQ}$-KB in Scott's normal form is quasi-forest satisfiable if and only if there exists a consistent $\mathcal{K}$-summary.

Proof. We split the proof into two sublemmas: Lemma 7.59 and Lemma 7.60. □

Lemma 7.59 If a $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ in Scott's normal form is quasi-forest satisfiable then there exists a consistent $\mathcal{K}$-summary.

Proof. Suppose that $\mathcal{K}$ is quasi-forest satisfiable, and take an elegant quasi-forest model $\mathcal{I}$ of $\mathcal{K}$ guaranteed by Lemma 7.50. In what follows we define a $\mathcal{K}$-summary $\mathcal{S}$ based on the clearing of $\mathcal{I}$, and then establish the consistency of $\mathcal{S}$. We let $\mathcal{S}$ to be composed of the following axioms:

(I) $\mathsf{a} \approx \mathsf{b}$ for all $\mathsf{a}, \mathsf{b} \in \mathsf{ind}(\mathcal{K})$ with $\mathsf{a}^{\mathcal{I}} = \mathsf{b}^{\mathcal{I}}$, and $\neg(\mathsf{a} \approx \mathsf{b})$ for all $\mathsf{a}, \mathsf{b} \in \mathsf{ind}(\mathcal{K})$ with $\mathsf{a}^{\mathcal{I}} \neq \mathsf{b}^{\mathcal{I}}$.

(II) $\mathrm{A}(\mathsf{a})$ (resp. $\neg\mathrm{A}(\mathsf{a})$) for all NFA $\mathscr{A}$ from $\mathcal{K}$, all concept names A appearing in

$$\mathcal{K} \cup \mathbf{C}(\mathsf{ind}(\mathcal{T}), \mathscr{A}) \cup \mathbf{C}_{\rightsquigarrow}(\mathscr{A}) \cup \{\mathrm{Root}\},$$

and all individual names $\mathsf{a} \in \mathsf{ind}(\mathcal{K})$ satisfying $\mathsf{a}^{\mathcal{I}} \in \mathrm{A}^{\mathcal{I}}$ (resp. $\mathsf{a}^{\mathcal{I}} \notin \mathrm{A}^{\mathcal{I}}$).

(III) $r(\mathsf{a}, \mathsf{b})$ (resp. $\neg r(\mathsf{a}, \mathsf{b})$) for all NFA $\mathscr{A}$ from $\mathcal{K}$, all role names r appearing in

$$\mathcal{K} \cup \mathbf{R}(\mathbf{N}_{\mathbf{I}}^{\mathcal{T}}, \mathscr{A}) \cup \{\mathit{edge}\},$$

and all individual names $\mathsf{a}, \mathsf{b} \in \mathsf{ind}(\mathcal{K})$ satisfying $(\mathsf{a}^{\mathcal{I}}, \mathsf{b}^{\mathcal{I}}) \in r^{\mathcal{I}}$ (resp. $(\mathsf{a}^{\mathcal{I}}, \mathsf{b}^{\mathcal{I}}) \notin r^{\mathcal{I}}$).

(IV) $\mathrm{A}(\mathsf{a})$ for

$$\mathrm{A} \in \{\mathrm{Clrng}^{r}_{\mathsf{t}^{\mathsf{a}}_{\mathrm{cl}}}, \mathrm{Chld}^{r}_{\mathsf{t}^{\mathsf{a}}_{\mathrm{ch}}}, \mathrm{Des}^{r,\mathsf{o}}_{\mathsf{t}^{\mathsf{a}}_{\mathsf{o}}} \mid \mathsf{o} \in \mathsf{ind}(\mathcal{T})\},$$

ranging over the number restrictions $(\geq n\ r).\top$ from $\mathcal{K}$ and unique thresholds $\mathsf{t}^{\mathsf{a}}_{\mathrm{cl}}$, $\mathsf{t}^{\mathsf{a}}_{\mathrm{ch}}$, $\mathsf{t}^{\mathsf{a}}_{\mathsf{o}}$ for which $\mathsf{a}^{\mathcal{I}} \in \mathrm{A}^{\mathcal{I}}$.

Following the items of Definition 7.51 we see that $\mathcal{S}$ is indeed a $\mathcal{K}$-summary. It remains to establish the clearing-consistency and subtree-consistency of $\mathcal{K}$. We start from the former one.

- It is immediate to see that $\mathcal{I}_{\mathcal{S}} \models (\mathcal{A}, \mathcal{T}_{\mathrm{loc}})$ due to the fact that $\mathcal{I} \models (\mathcal{A}, \mathcal{T}_{\mathrm{loc}})$.
- Take any GCI $\mathrm{A} \equiv \exists \mathscr{A}_{\mathsf{q},\mathsf{q}'}.\top$ from $\mathcal{T}$. As $\mathcal{I}$ is $\mathscr{A}$-decorated, we see that $\mathcal{I}_{\mathcal{S}}$ is virtually $(\mathsf{ind}(\mathcal{T}), \mathscr{A})$-decorated. By proper interpretation of Reach-concepts we can invoke Lemma 7.32 to conclude the sequence of equivalences: $\mathrm{A}(\mathsf{a}) \in \mathcal{S}$ iff $\mathsf{a}^{\mathcal{I}} \in \mathrm{A}^{\mathcal{I}}$ iff $\mathsf{a}^{\mathcal{I}} \in (\exists \mathscr{A}_{\mathsf{q},\mathsf{q}'}.\top)^{\mathcal{I}}$ iff $\mathsf{a}^{\mathcal{I}} \in (\mathrm{Reach}^{\mathscr{A}}_{\mathsf{q},\mathsf{q}'})^{\mathcal{I}}$ iff $\mathrm{Reach}^{\mathscr{A}}_{\mathsf{q},\mathsf{q}'}(\mathsf{a}) \in \mathcal{S}$, as desired.
- Take any GCIs $\mathrm{A} \equiv (\geq n\ r).\top$ from $\mathcal{T}$. As the interpretation $\mathcal{I}$ is elegant (and thus $(\mathsf{ind}(\mathcal{T}), \geq n\ r)$-decorated), we clearly have that $\mathcal{I}_{\mathcal{S}}$ is virtually $(\mathsf{ind}(\mathcal{T}), \geq n\ r)$-decorated. Applying Lemma 7.45 and the fact that $\mathcal{I} \models \mathcal{K}$ we conclude that the sequence of equivalences: $\mathsf{a}^{\mathcal{I}} \in ((\geq n\ r).\top)^{\mathcal{I}}$ iff $\mathsf{a}^{\mathcal{I}} \in \mathrm{A}^{\mathcal{I}}$ iff $\mathrm{A}(\mathsf{a}) \in \mathcal{S}$ iff $\mathsf{a}^{\mathcal{I}_{\mathcal{S}}}$ virtually satisfies $(\geq n\ r).\top$.

We next establish the subtree-consistency of $\mathcal{S}$. To do so, take any $\mathsf{a} \in \mathsf{ind}(\mathcal{K})$, and consider the $\mathcal{T}$-summary $\mathcal{S}_{\mathsf{a}}$ as defined in Definition 7.57. We need to show that the $\mathcal{ZOIQ}$-KB $\mathcal{K}_{\mathcal{S}_{\mathsf{a}},\text{👻}} := (\mathcal{S}_{\mathsf{a}}, \mathcal{T}_{\mathrm{loc}} \cup \mathcal{T}_{\mathrm{aut}} \cup \mathcal{T}_{\mathrm{cnt}})$, as defined in Definition 7.55, has a quasi-forest model. We claim that the subinterpretation $\mathcal{J}$ constructed from $\mathcal{I}$ by restricting its domain to the roots named with some name from $(\{\mathsf{a}\} \cup \mathsf{ind}(\mathcal{T}))$ and their descendants (and interpreting $\text{👻}^{\mathcal{J}} := \mathsf{a}^{\mathcal{J}}$) is a model of $\mathcal{K}_{\mathcal{S}_{\mathsf{a}},\text{👻}}$. As $\mathcal{J}$ is a submodel of $\mathcal{I}$, we know that $\mathcal{J} \models (\mathcal{S}_{\mathsf{a}}, \mathcal{T}_{\mathrm{loc}})$. For the satisfaction of $\mathcal{T}_{\mathrm{aut}}$ we invoke Corollary 7.35 and Lemma 7.32 and the fact that $\mathcal{J}$ is a submodel of $\mathcal{I}$. For the satisfaction of $\mathcal{T}_{\mathrm{cnt}}$ we again rely on the fact that $\mathcal{J}$ is a submodel of $\mathcal{I}$ (so the non-root elements do not violate number restrictions) and on Lemma 7.45.

This concludes the proof of the consistency of $\mathcal{S}$. □

We now establish the other (more difficult) direction of Lemma 7.58. The proof is given below.

Lemma 7.60 For every $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ in Scott's normal form we have that the existence of a consistent $\mathcal{K}$-summary implies the quasi-forest satisfiability of $\mathcal{K}$.

Proof. Let $\mathcal{S}$ be a consistent $\mathcal{K}$-summary, and let $\mathcal{I}_{\mathcal{S}}$ be the corresponding interpretation. The interpretation $\mathcal{I}_{\mathcal{S}}$ is going to serve as the clearing of the intended model of $\mathcal{K}$. For individual names $\mathsf{a} \in \mathsf{ind}(\mathcal{K})$ labelling pairwise-different elements of $\mathcal{I}_{\mathcal{S}}$ (*i.e.* from different equivalence class of $\approx$) we let $\mathcal{I}_{\mathsf{a}}$ to be a quasi-forest model of the $\mathcal{ZOIQ}$-KB $\mathcal{K}_{\mathcal{S}_{\mathsf{a}},\text{👻}} := (\mathcal{S}_{\mathsf{a}}, \mathcal{T}_{\mathrm{loc}} \cup \mathcal{T}_{\mathrm{aut}} \cup \mathcal{T}_{\mathrm{cnt}})$, where $\mathcal{S}_{\mathsf{a}}$ is the $(\mathcal{T}, \mathcal{S}, \mathsf{a})$-ghost-summary as in Definition 7.57 and $\mathcal{K}_{\mathcal{S}_{\mathsf{a}},\text{👻}}$ is as in Definition 7.55. Note that $\mathcal{I}_{\mathsf{a}}$ exists by the consistency of $\mathcal{S}$, and w.l.o.g. we can assume that $\mathcal{I}_{\mathsf{a}}$ interprets all concept and role names that do not appear in $\mathcal{S}$ as empty sets. We stress that by the design of $\mathcal{S}_{\mathsf{a}}$ we have that both $\mathcal{I}_{\mathcal{S}}$ and $\mathcal{I}_{\mathsf{a}}$ are identical when restricted to nominals. Hence, in what follows, we are going to "merge" $\mathcal{I}_{\mathcal{S}}$ and all the $\mathcal{I}_{\mathsf{a}}$ to obtain the desired quasi-forest model of $\mathcal{K}$. By the *merge* of an interpretation $\mathcal{I}$ with $\mathcal{I}_{\mathsf{a}}$ we mean the interpretation $\mathcal{J}$ such that:

- $\Delta^{\mathcal{J}} = \Delta^{\mathcal{I}} \cup \{\mathsf{a}^{\mathcal{I}} \cdot w \mid \text{👻}^{\mathcal{I}} \cdot w \in \Delta^{\mathcal{I}_{\mathsf{a}}}\}$.
- Individual names are interpreted as in $\mathcal{I}$.
- For concept name $\mathrm{A} \in \mathbf{N_C}$ we put $\mathrm{A}^{\mathcal{J}} := \mathrm{A}^{\mathcal{I}} \cup \{\mathsf{a}^{\mathcal{J}} \cdot w \mid \text{👻}^{\mathcal{I}_{\mathsf{a}}} \cdot w \in \mathrm{A}^{\mathcal{I}_{\mathsf{a}}}\}$.
- For role names $r \in \mathbf{N_R}$ we interpret r as

$$r^{\mathcal{I}} \cup \{(\mathsf{a}^{\mathcal{J}} \cdot w, \mathsf{a}^{\mathcal{J}} \cdot v) \mid (\text{👻}^{\mathcal{I}_{\mathsf{a}}} \cdot w, \text{👻}^{\mathcal{I}_{\mathsf{a}}} \cdot v) \in r^{\mathcal{I}_{\mathsf{a}}}\} \cup \{(\mathsf{a}^{\mathcal{J}} \cdot w, \mathsf{o}^{\mathcal{J}}) \mid (\text{👻}^{\mathcal{J}_{\mathsf{a}}} \cdot w, \mathsf{o}^{\mathcal{I}_{\mathsf{a}}}) \in r^{\mathcal{I}_{\mathsf{a}}}, \mathsf{o} \in \mathsf{ind}(\mathcal{T})\}.$$

Intuitively, we "plug in" the subtree of 👻 in $\mathcal{I}_{\mathsf{a}}$ into the place of the individual a in $\mathcal{I}$, making sure that all the freshly added elements are appropriately connected to the nominals. Moreover, note that $\mathcal{J}$ and $\mathcal{I}$ are identical when restricted to the named elements.

Let $\mathcal{J}$ be the merge of $\mathcal{I}_{\mathcal{S}}$ with all the $\mathcal{I}_{\mathsf{a}}$ (iterated in any fixed order). We claim that $\mathcal{J}$ is a quasi-forest model of $\mathcal{K}$. While the fact that $\mathcal{J}$ is a quasi-forest follows by the fact that all the components of the merge are also quasi-forests, the modelhood of $\mathcal{J}$ remains to be shown. As $\mathcal{I}_{\mathcal{S}}$ satisfies $\mathcal{A}$ (by the clearing consistency), and all the applications of the merge operation yield interpretations with identical clearing, we conclude that $\mathcal{J} \models \mathcal{A}$.

Now consider any GCI from $\mathcal{T}_{\text{loc}}$ (*i.e.* the ones that do not involve automata nor number restrictions). By the design of the notion of merging we see for any element d from $\mathcal{J}$ there exists an element d' in $\mathcal{I}$ or one of the interpretations $\mathcal{I}_{\mathsf{a}}$ such that d and d' satisfy precisely the same atomic concepts. Indeed, if d is a root of $\mathcal{J}$ we take $\mathrm{d}' := \mathrm{d}$ from $\mathcal{I}$. Otherwise, there is an individual name a for which $\mathrm{d} = \mathsf{a}^{\mathcal{J}} w$ and we put $\mathrm{d}' := \ghost^{\mathcal{I}_{\mathsf{a}}} w$ from $\mathcal{I}_{\mathsf{a}}$. By the fact that $\mathcal{I}$ and all the $\mathcal{I}_{\mathsf{a}}$ satisfy $\mathcal{T}_{\text{loc}}$, we conclude the satisfaction of the GCIs from $\mathcal{T}_{\text{loc}}$ having one of the forms: $\mathrm{A} \equiv \{\mathsf{o}\}$, $\mathrm{A} \equiv \mathrm{B}$, $\mathrm{A} \equiv \neg\mathrm{B}$, and $\mathrm{A} \equiv \mathrm{B} \sqcap \mathrm{B}'$. A similar characterisation can be provided for the roles in $\mathcal{J}$. This, together with the fact that $\mathcal{I}$ and all the $\mathcal{I}_{\mathsf{a}}$ satisfy $\mathcal{T}_{\text{loc}}$, we infer the satisfaction of the GCIs from $\mathcal{T}_{\text{loc}}$ having one of the forms: $\mathrm{A} \equiv \exists r.\mathsf{Self}$ or $s = s'$.

We next address the satisfaction of GCIs from $\mathcal{T}$ of the form $\mathrm{A} \equiv \exists \boldsymbol{\mathcal{A}}_{\mathsf{q},\mathsf{q}'}.\top$. Note that $\mathcal{I}_{\mathcal{S}}$ is virtually $\boldsymbol{\mathcal{A}}$-decorated (by design of $\mathcal{S}$) and the elements from the clearing virtually satisfies $\mathrm{A} \equiv \exists \boldsymbol{\mathcal{A}}_{\mathsf{q},\mathsf{q}'}.\top$. Moreover, we have that the interpretation of $\ghost$ in every interpretation $\mathsf{a}^{\mathcal{I}_{\mathsf{a}}}$ satisfies $\mathrm{comdsc}(\mathsf{ind}(\mathcal{T}), \boldsymbol{\mathcal{A}})$, which is due to the construction of $\mathcal{K}_{\mathcal{S}_{\mathsf{a}},\ghost}$. It is not hard to see that this implies $\mathsf{a}^{\mathcal{J}} \in \mathrm{comdsc}(\mathsf{ind}(\mathcal{T}), \boldsymbol{\mathcal{A}})\,[\ghost/\mathsf{a}]^{\mathcal{J}}$. Hence, $\mathcal{J}$ is properly $\boldsymbol{\mathcal{A}}$-decorated and by Lemma 7.32 it suffices to verify whether $\mathcal{J}$ is a model of $\mathrm{A} \equiv \mathrm{rel}(\mathsf{ind}(\mathcal{T}), \boldsymbol{\mathcal{A}}_{\mathsf{q},\mathsf{q}'})$. We verify the equality $\mathrm{A}^{\mathcal{J}} = (\mathrm{rel}(\mathsf{ind}(\mathcal{T}), \boldsymbol{\mathcal{A}}_{\mathsf{q},\mathsf{q}'}))^{\mathcal{J}}$ separately for the clearing and the other elements. When considering the elements of the clearing, the desired equality follows from the fact that $\mathcal{S}$ is clearing-consistent, as indicated by the second item of Definition 7.53. For the remaining elements, we invoke the fact that all the $\mathcal{I}_{\mathsf{a}}$ are models of $\mathrm{A} \equiv \mathrm{rel}(\mathsf{ind}(\mathcal{T}), \boldsymbol{\mathcal{A}}_{\mathsf{q},\mathsf{q}'})$ and the merging process preserves all the relevant paths between $\ghost$, its descendants and nominals. More formally, any path from $\mathcal{I}_{\mathsf{a}}$ witnessing the satisfaction of some automata constraint in $\mathcal{I}_{\mathsf{a}}$ is also the desired witnessing path in $\mathcal{J}$ (modulo renaming all occurrences of $\ghost$ with a in the description of its elements).

Finally, we establish satisfaction of all the GCIs having the form $\mathrm{A} \equiv (\geq n\ r).\top$. First, by the design of $\mathcal{S}$, we know that $\mathcal{I}_{\mathcal{S}}$ (and hence also $\mathcal{J}$) is virtually $(\geq n\ r)$-decorated and it virtually satisfies $(\geq n\ r).\top$. Second, by the definition of the merge, we see that our construction of $\mathcal{J}$ preserves the total number of r-successors for all non-root elements. Hence, by the fact that all $\mathcal{I}_{\mathsf{a}}$ satisfy the GCI $\neg\mathrm{Root} \sqcap \mathrm{A} \equiv \neg\mathrm{Root} \sqcap (\geq n\ r).\top$, we conclude that every element outside the clearing of $\mathcal{J}$ satisfies A if and only if it satisfies $(\geq n\ r).\top$. Finally, we proceed with the elements of the clearing of $\mathcal{I}$. As we know that $\mathcal{J}$ is virtually $(\geq n\ r)$-decorated and it virtually satisfies $(\geq n\ r).\top$, we see that to establish the satisfaction of the GCI $\mathrm{Root} \sqcap \mathrm{A} \equiv \mathrm{Root} \sqcap (\geq n\ r).\top$ it suffices to show that $\mathcal{J}$ is $(\geq n\ r)$-decorated. To do so, we invoke Lemma 7.45. The proper interpretation of the concepts having the form $\mathrm{Clrng}^{r}_{\mathfrak{t}}$ for all the roots of $\mathcal{I}_{\mathcal{S}}$ (and hence, by induction, also $\mathcal{J}$) is by the clearing consistency of $\mathcal{S}$ (more precisely by Item (ii) of Definition 7.53 and the notion of virtual satisfaction). For the remaining concepts, we take any individual name $\mathsf{a} \in \mathsf{ind}(\mathcal{K})$ and observe that $\ghost$ in $\mathcal{I}_{\mathsf{a}}$ satisfies $\mathrm{desc}\left(\mathrm{Chld}^{r}_{\mathfrak{t}^{\mathsf{a}}_{\mathrm{ch}}} \sqcap \bigsqcap_{\mathsf{o} \in \mathsf{ind}(\mathcal{T})} \mathrm{Des}^{r,\mathsf{o}}_{\mathfrak{t}^{\mathsf{a}}_{\mathsf{o}}}\right)$ for all the relevant concepts A with $\mathrm{A}(\mathsf{a})$ present in $\mathcal{S}$. As our construction of $\mathcal{J}$ preserves r-successors of non-root elements of each $\mathcal{I}_{\mathsf{a}}$ while merging (in particular, all r-connections to nominals), it is easy to see that the satisfaction of the above concept implies that $\mathsf{a}^{\mathcal{J}}$ satisfies $(\mathrm{desc}(\mathrm{C})[\ghost/\mathsf{a}])^{\mathcal{J}}$. Thus $\mathcal{J}$ is indeed $(\geq n\ r)$-decorated.

We can now conclude that $\mathcal{J}$ is indeed a quasi-forest model of $\mathcal{K}$, as desired. $\square$

7.8 Deciding Existence of Quasi-Forest Models

Based on Lemma 7.58 as well as the other lemmas presented in the previous section, we can finally design an algorithm for deciding whether an input $\mathcal{ZOIQ}$-KB is quasi-forest satisfiable.

Procedure 5: Deciding Quasi-Forest Satisfiability in $\mathcal{ZOIQ}$

```
Input: A $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$.
Output: True if and only if $\mathcal{K}$ is quasi-forest satisfiable.
1 Turn $\mathcal{T}$ into Scott's normal form.   // In PTime w.r.t. $|\mathcal{T}|$ due to Lemma 7.50.
2 Compute the set $\mathbf{S}_{\mathcal{T}}$.   // Implementable in ExpTime w.r.t. $|\mathcal{T}|$. Consult Lemma 7.56.
3 Guess a $\mathcal{K}$-summary $\mathcal{S}$.   // NP in $|\mathcal{K}|$ by Lemma 7.52.
4 Return False if $\mathcal{S}$ is clearing-consistent.   // Implementable in PTime w.r.t. $|\mathcal{K}|\cdot|\mathcal{S}|$ by Lemma 7.54.
5 Foreach $\mathsf{a} \in \mathsf{ind}(\mathcal{K})$ do compute the $(\mathcal{T}, \mathcal{S}, \mathsf{a})$-ghost-summary $\mathcal{S}_{\mathsf{a}}$ and return False if $\mathcal{S}_{\mathsf{a}} \notin \mathbf{S}_{\mathcal{T}}$.
    // Implementable in PTime w.r.t. $|\mathcal{K}| \cdot (|\mathcal{K}|+|\mathbf{S}_{\mathcal{T}}|)$.
6 Return True.
```

We next establish correctness and calculate the running time of our algorithm. In the proof we rely on all the lemmas and definitions presented in Section 7.6.

Lemma 7.61 Procedure 5 returns True if and only if the input $\mathcal{ZOIQ}$-KB $\mathcal{K}$ has an elegant model. Moreover, there exist polynomial function p and an exponential function e for which Procedure 5 can be implemented with a nondeterministic Turing machine of running time bounded, for every input $\mathcal{K} := (\mathcal{A}, \mathcal{T})$, by $\mathrm{e}(|\mathcal{T}|) + \mathrm{p}(|\mathcal{K}|) + \mathrm{p}(|\mathcal{K}|) \cdot \mathrm{e}(|\mathcal{T}|)$.

Proof. Correctness of our algorithm follows from Lemma 7.50 and Lemma 7.58. For its implementation, we stress that:

- The first step of the algorithm can be done in time polynomial w.r.t. $|\mathcal{T}|$ by Lemma 7.50.
- The second step of the algorithm can be done in time exponential w.r.t. $|\mathcal{T}|$ by lemma 7.56.
- The third step can be implemented in non-deterministic polynomial time, due to Lemma 7.52.
- The fourth step can be implemented in time polynomial w.r.t. $|\mathcal{K}|\cdot|\mathcal{S}|$ by Lemma 7.54, and thus in time polynomial w.r.t. $|\mathcal{K}|$.
- In the second-to-last step, for each name $\mathsf{a} \in \mathsf{ind}(\mathcal{K})$ we construct the $(\mathcal{T}, \mathcal{S}, \mathsf{a})$-ghost-summary $\mathcal{S}_{\mathsf{a}}$. This can clearly be done in time polynomial w.r.t. $|\mathcal{T}| + |\mathcal{S}|$, and thus polynomial w.r.t. $|\mathcal{K}|$. We then iterate through the set $\mathbf{S}_{\mathcal{T}}$ (which is exponential w.r.t. $|\mathcal{T}|$ by Lemma 7.56) and check whether $\mathcal{S}_{\mathsf{a}}$ belongs to it or not. This costs us the time exponential w.r.t. $|\mathcal{T}|$. Hence, the whole step can be executed in time exponential w.r.t. $|\mathcal{K}|$.

Hence, there exists a polynomial function p and an exponential function e such that the running time of the above algorithm can be bounded by $\mathrm{e}(|\mathcal{T}|) + \mathrm{p}(|\mathcal{K}|) + \mathrm{p}(|\mathcal{K}|) \cdot \mathrm{e}(|\mathcal{T}|)$ for every input $\mathcal{K} := (\mathcal{A}, \mathcal{T})$. This concludes the proof. □

We stress that by replacing the 3rd step of Procedure 5 by a loop that iterates through all possible $\mathcal{K}$-summaries, would yield a deterministic exponential time algorithm for quasi-forest satisfiability of $\mathcal{ZOIQ}$. In our case however, it is more convenient to keep the algorithm as it is. Suppose now that a TBox $\mathcal{T}$ is fixed and only an ABox $\mathcal{A}$ is given as the input. Then the first two steps of Procedure 5 are independent from the input. The same holds for the verification of whether a given ghost summary belongs to the pre-computed set $\mathbf{S}_{\mathcal{T}}$ from line 5. We present the desired algorithm below.

Procedure 6: Deciding Quasi-Forest Satisfiability in $\mathcal{ZOIQ}$ w.r.t. data complexity

```
Input: An ABox $\mathcal{A}$.
Parameters: $\mathcal{ZOIQ}$-TBox $\mathcal{T}$ already in Scott's normal form and a pre-computed set $\mathbf{S}_{\mathcal{T}}$.
Output: True if and only if $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ is quasi-forest satisfiable.
1 Guess a $\mathcal{K}$-summary $\mathcal{S}$.   // NP in $|\mathcal{K}|$ by Lemma 7.52.
2 Return False if $\mathcal{S}$ is clearing-consistent.   // Implementable in PTime w.r.t. $|\mathcal{K}|\cdot|\mathcal{S}|$ by Lemma 7.54.
3 Foreach $\mathsf{a} \in \mathsf{ind}(\mathcal{K})$ do compute the $(\mathcal{T}, \mathcal{S}, \mathsf{a})$-ghost-summary $\mathcal{S}_{\mathsf{a}}$ and return False if $\mathcal{S}_{\mathsf{a}} \notin \mathbf{S}_{\mathcal{T}}$.
    // Implementable in PTime w.r.t. $|\mathcal{K}| \cdot (|\mathcal{K}|+|\mathbf{S}_{\mathcal{T}}|)$, and thus in PTime w.r.t. $|\mathcal{K}|$ if $\mathbf{S}_{\mathcal{T}}$ is fixed.
4 Return True.
```

We employ Lemma 7.61, and based on Procedure 6 we establish the forthcoming Theorem 7.62.

Theorem 7.62

For every $\mathcal{ZOIQ}$-TBox $\mathcal{T}$, there is an NP procedure (parametrised by $\mathcal{T}$) that takes an ABox $\mathcal{A}$ as the input and decides whether the $\mathcal{ZOIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ has a quasi-forest model.

Proof. For the TBox $\mathcal{T}$ we turn it into Scott's normal form with Lemma 7.50, and then compute $\mathbf{S}_{\mathcal{T}}$ with Lemma 7.56. We now treat both $\mathcal{T}$ and $\mathbf{S}_{\mathcal{T}}$ as parameters and construct a Turing machine realising Procedure 6. Correctness of Procedure 6 now follows from correctness of Procedure 5. As for the running time, we use the fact that $\mathcal{T}$ is fixed from the beginning, and thus the value $\mathrm{e}(|\mathcal{T}|)$ is treated as constant. Hence, the function $\mathrm{e}(|\mathcal{T}|) + \mathrm{p}(|\mathcal{K}|) + \mathrm{p}(|\mathcal{K}|) \cdot \mathrm{e}(|\mathcal{T}|)$ bounding the running time of our algorithm, becomes polynomial in terms of $|\mathcal{A}|$, as desired. $\square$

Recall that the maximal decidable fragments of $\mathcal{ZOIQ}$ have the elegant model property (namely that every satisfiable knowledge base has an elegant model) as explained by Lemma 7.5 and Lemma 7.50. Thus by employing the above algorithm we can conclude:

Theorem 7.63

The satisfiability problem for $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, and $\mathcal{ZOI}$ is NP-complete w.r.t. the data complexity.

It is know that the decidable expressive logics from the $\mathcal{SR}$ family, the logical core of OWL2, can be equivalently rewritten into the $\mathcal{Z}$ family [CEO09, Prop. 5.1]. More precisely, for any ABox $\mathcal{A}$ and any TBox $\mathcal{T}$ in $\mathcal{SRIQ}$, $\mathcal{SROQ}$ or $\mathcal{SROI}$ one can compute a TBox $\mathcal{T}'$, respectively, in $\mathcal{ZIQ}$, $\mathcal{ZOQ}$ or $\mathcal{ZOI}$ such that $(\mathcal{A}, \mathcal{T})$ is satisfiable if and only if $(\mathcal{A}, \mathcal{T}')$ is. Hence, Theorem 7.63 reproves known results about the $\mathcal{SR}$ family, previously obtained by Kazakov [Kaz08, Lemma 10, Table 1] by a translation from $\mathcal{SROIQ}$ to the two-variable fragment of first-order logic extended with counting quantifiers [Pra09, Thm. 1].

Corollary 7.64

The satisfiability problem for $\mathcal{SRIQ}$, $\mathcal{SROQ}$, and $\mathcal{SROI}$ is NP-complete w.r.t. the data complexity.

Recall that *regular path queries* are queries of the form $\mathscr{R}(x, y)$, where $\mathscr{R}$ is a regular expression (with tests). One of the key features of the $\mathcal{Z}$ family is that they have a built-in support for regular path expressions. Thus, by including the GCIs $\top \sqsubseteq \neg\exists\mathscr{R}.\top$ to an input TBox $\mathcal{T}$, one can ensure that all the models of $\mathcal{T} \cup \{\top \sqsubseteq \neg\exists\mathscr{R}.\top\}$ *do not* satisfy the query $\mathscr{R}(x, y)$. Hence, as a corollary we obtain:

Corollary 7.65

The entailment problem of regular path queries over tamed $\mathcal{ZOIQ}$-KBs is coNP-complete w.r.t. the data complexity. In particular, this applies to $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, $\mathcal{ZOI}$, and to the corresponding fragments of OWL2, namely $\mathcal{SRIQ}$, $\mathcal{SROQ}$, and $\mathcal{SROI}$.

8

Applications in Rooted Query Entailment

Contents

Motivation and Our Contribution

In this section we focus on the rooted query entailment problem, *i.e.* a restriction of the classical query entailment problem where the input query is assumed to be connected and contains at least one answer variable. Such a natural restriction of the query entailment problem often reduces its complexity. The canonical example may be the well-known DL $\mathcal{ALCI}$. It enjoys 2ExpTime-complete [Lut08a, Thm. 2] conjunctive query entailment problem, while its complexity in rooted case drops to coNExpTime [Lut08a, Thm. 1]. Despite the clear complexity-theoretic motivation towards studying the complexity of restricted classes of queries, there is a more serious application of this line of work. Quite recently, Jung et al. [JLPW22, p. 3] established a tight correspondence between the entailment of (unions of) rooted conjunctive queries and the so-called weak separability problem, one of the fundamental problems in concept (machine) learning.

In this chapter we first modify the algorithm presented in Section 7.8 to infer a coNExpTime upper bound for the entailment of unions of rooted conjunctive queries in $\mathcal{ZIQ}$ (consult Lemma 8.1 to understand why our results do not generalise to logics with nominals). We conclude with a novel coNExpTime lower bound for $\mathcal{ALC}$ extended with the Self operator. We stress that by the correspondence between rooted entailment and weak separability, our results yield novel lower and upper bounds for the later problem.

Overview of the Chapter and Prerequisites

We start with an extra preliminaries (Section 8.1). We then establish coNExpTime upper bound for the rooted query entailment in $\mathcal{ZIQ}$ (Section 8.2). We conclude with the matching coNExpTime lower bound for $\mathcal{ALC}$ extended with the Self operator, where the proof overview is given in Section 8.3.1.

We assume that the reader is familiar with the content of Chapter 7, especially its last section concerning our algorithm for deciding quasi-forest satisfiability of $\mathcal{ZOIQ}$. For the lower bound section, it may be useful to read Chapter 6 first, although such an extra background is not required.

8.1 Additional Preliminaries for Rooted Query Entailment

We recall the standard definitions regarding rooted queries and the entailment problem for rooted query. We also discuss a suitable version of the tiling problem employed in our forthcoming hardness proof.

8.1.1 Rooted Entailment of Queries

We call a conjunctive query (CQ) *connected* if its query structure (defined as in Section 2.4) is connected. A CQ is *rooted* if it is connected and contains at least one individual name.[1] The definition of "being rooted" is lifted to unions of CQs by requiring that each of its disjuncts is a rooted. In this chapter we are interested in the following problem:

Entailment of rooted (unions of) conjunctive queries over $\mathcal{DL}$-KBs

Parameters:	Description logic $\mathcal{DL}$
Input:	A $\mathcal{DL}$-knowledge base $\mathcal{K}$, and a *rooted* (union of) conjunctive queries q
Question:	Does $\mathcal{K}$ entail q?

We stress that the rooted entailment problem is a special case of the classical query entailment problem (with a proviso that the individual names are allowed in the query in addition to variables). Hence, the extensions of $\mathcal{ALC}$ with ExpTime-complete UCQ entailment problem (see *e.g.* Corollary 5.7) also enjoy ExpTime-complete entailment problem for (unions of) rooted conjunctive queries. We would also like to point out that whenever a description logic $\mathcal{DL}$ allows for nominals, then the additional assumptions of rootedness and connectivity do not influence the complexity of the underlying decision problem.[2] Indeed:

Lemma 8.1 Let $\mathcal{DL}$ extend $\mathcal{ALCO}$. Then there exists a LogSpace reduction between the rooted entailment problem of CQs over $\mathcal{DL}$-KBs and the CQ entailment problem over $\mathcal{DL}$-KBs.

Proof. Based on the discussion above, it suffices to reduce the query entailment problem to the rooted entailment. Let $\mathcal{K}$ be an $\mathcal{DL}$-knowledge-base, q be a conjunctive query, $\mathtt{a}$, A and r be, respectively, a fresh individual, concept and role name, that do not appear in $\mathcal{K}$ and q. It is not difficult to see that $\mathcal{K} \models q$ if and only if $\mathcal{K} \cup \{\top \sqsubseteq \exists r.\{\mathtt{a}\}\}$ entails the rooted conjunctive query $q \wedge \bigwedge_{x \in \mathrm{Var}(q)} r(x, \mathtt{a})$. This concludes the proof. $\square$

8.1.2 Torus Tiling Problem

In our reduction we employ a variant of a tiling problem [Wan61]. An auxiliary definition comes first.

Definition 8.2 A **torus tiling system** is a triple $\mathfrak{D} := (\mathrm{T}, \mathrm{H}, \mathrm{V})$, where T is a set of *tiles*, and $\mathrm{H} \subseteq \mathrm{T} \times \mathrm{T}$ and $\mathrm{V} \subseteq \mathrm{T} \times \mathrm{T}$ are, respectively, horizontal and vertical *matching relations*. We say that $\mathfrak{D}$ **covers** $\mathbb{Z}_{2^n} \times \mathbb{Z}_{2^n}$ for a positive $n \in \mathbb{N}$ if there exists a mapping $\xi \colon \mathbb{Z}_{2^n} \times \mathbb{Z}_{2^n} \to \mathrm{T}$ such that for all pairs $(x, y) \in \mathbb{Z}_{2^n} \times \mathbb{Z}_{2^n}$ the following conditions are satisfied ($\oplus_{2^n}$ denotes addition modulo 2^n):
(THor) $(\xi(x, y), \xi(x \oplus_{2^n} 1, y)) \in \mathrm{H}$, and
(TVer) $(\xi(x, y), \xi(x, y \oplus_{2^n} 1)) \in \mathrm{V}$.
The tiles t, t' are H-*compatible* whenever $(\mathrm{t}, \mathrm{t}') \in \mathrm{H}$, and V-*compatible* whenever $(\mathrm{t}, \mathrm{t}') \in \mathrm{V}$.

[1] In contrast to all the other chapters of this thesis, here we explicitly consider queries with answer variables.
[2] We would like to thank Jean Jung for this observation.

The **torus tiling problem** [Lut02, Def. 4.13] is properly defined below. Its NExpTime-completeness can be shown in a standard manner, and was first established in the PhD thesis of Lutz [Lut02, Cor. 4.15].

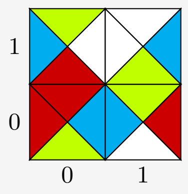
(Exponential) Torus Tiling Problem

Input: Positive integer n (encoded in unary), a torus tiling system $\mathscr{D} := (\mathrm{T}, \mathrm{H}, \mathrm{V})$, and an initial condition $\bar{\mathrm{c}} := (\bar{\mathrm{c}}_0, \ldots, \bar{\mathrm{c}}_{n-1}) \in \mathrm{T}^n$.

Question: Does $(\mathscr{D}, \bar{\mathrm{c}})$ have a *solution*, *i.e.* is there a function ξ that covers $\mathbb{Z}_{2^n} \times \mathbb{Z}_{2^n}$ such that for all $i < n$ we have $\xi(i, 0) = \bar{\mathrm{c}}_i$?

For brevity, whenever a map ξ satisfies $\xi(i, 0) = \bar{\mathrm{c}}_i$ for all $i < n$, we say that ξ is *compatible* with $\bar{\mathrm{c}}$.

Example 8.3. Let $\mathscr{D}$ be the torus tiling system defined as follows. Let T be composed of all four-bordered tiles made of colours: 🟦, 🟥, □, 🟩, *e.g.* T contains ⊠. We let H (resp. V) be composed of all pairs of tiles $(\mathrm{t}, \mathrm{t}') \in \mathrm{T}^2$ for which the right border of t has the same colour as the left border of t', *e.g.* (⊠, (⊠) $\in$ H. Finally, we define V as the set of all pairs of tiles $(\mathrm{t}, \mathrm{t}') \in \mathrm{T}^2$ for which the upper border of t has the same colour as the lower border of t', *e.g.* ((⊠, ⊠) $\in$ V. Then $\mathscr{D}$ covers $\mathbb{Z}_2 \times \mathbb{Z}_2$ as witnessed by the mapping $\xi := \{(0,0) \mapsto$ ⊠, $(1,0) \mapsto$ ⊠, $(0,1) \mapsto$ ⊠, $(1,1) \mapsto$ ⊠$\}$. Moreover, assuming $\bar{\mathrm{c}} :=$ (⊠), we have that ξ is a solution to $(\mathscr{D}, \bar{\mathrm{c}})$. Such a solution ξ is visualised below.



8.2 Entailment of Rooted Queries: Upper Bound

As an application of our techniques from the previous chapter, we show how Procedure 5 can be adapted to derive coNExpTime-completeness of the entailment problem of unions of rooted conjunctive queries over $\mathcal{ZIQ}$-KBs. We thus generalise the results on $\mathcal{SHIQ}$ by Lutz [Lut08b, Thm. 2] (as transitivity of roles can be simulated with the Kleene's star in $\mathcal{ZIQ}$). We focus on the dual problem: "Given a (union of) rooted CQs q and $\mathcal{ZIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$, is there a model of $\mathcal{K}$ that violates q?" and show its NExpTime-completeness. As we work with $\mathcal{ZIQ}$, by Lemma 7.50 we can assume that the input KB $\mathcal{K}$ is in Scott's normal form, and the intended model $\mathcal{I}$ violating q is elegant (in particular, is N-bounded for an N exponential in $|\mathcal{T}|$). The crucial observation is that whenever q matches $\mathcal{I}$, all the elements from the image of a match are at the depth at most $|q|$ (as q has at least one individual name and is connected). Hence, it suffices to construct an "initial segment" of $\mathcal{I}$ of depth at most $|q|$ and degree at most N, and check if (i) q does not match it, and (ii) it can be extended to the full model of $\mathcal{K}$. This is formalised next.

Definition 8.4 Let $\mathrm{R}, \mathrm{C}, \mathrm{D} \in \mathbb{N}$. An **$(\mathrm{R}, \mathrm{C}, \mathrm{D})$-forest $\mathcal{F}$** is a prefix-closed set of non-empty words from $\mathbb{Z}_{\mathrm{R}} \cdot (\mathbb{Z}_{\mathrm{C}})^+$ of length at most D. The number R indicates the total number of roots of $\mathcal{F}$, C denotes the maximal number of children per each element, and D indicates the maximal depth.

Treating $\mathcal{F}$ as a set of individual names, we define an **$(\mathcal{F}, \mathcal{K})$-initial segment** of a $\mathcal{ZOIQ}$-KB $\mathcal{K}$ as any summary $\mathcal{S}$ of $\mathcal{K} \cup \{\mathsf{a} \not\approx \mathsf{b} \mid \mathsf{a}, \mathsf{b} \in \mathcal{F}, \mathsf{a} \neq \mathsf{b}\}$ such that:

- For every $\mathsf{a} \in \mathsf{ind}(\mathcal{K})$ there is $\mathsf{b} \in \mathcal{F} \cap \mathbb{N}$ with $\mathsf{a} \approx \mathsf{b}$ in $\mathcal{S}$, and for every $\mathsf{b} \in \mathcal{F} \cap \mathbb{N}$ there is $\mathsf{a} \in \mathsf{ind}(\mathcal{K})$ with $\mathsf{a} \approx \mathsf{b}$ in $\mathcal{S}$.
- $((=0\ \mathit{child}).\top)(\mathsf{a})$ for all $\mathsf{a} \in \mathcal{F}$ that are not leaves of $\mathcal{F}$.
- $\neg r(\mathsf{a}, \mathsf{b})$ for all role names r appearing in $\mathcal{K}$ and all $\mathsf{a} \neq \mathsf{b}$ from $\mathcal{F}$ such that a is not a child of b in $\mathcal{F}$ (or vice-versa).

The above conditions are needed to represent a forest $\mathcal{F}$ inside the clearing of the intended models of $\mathcal{S}$. The first item of Definition 8.4 guarantees the proper behaviour of roots. The second item guarantees that the children of elements in the initial segment are precisely the ones that are explicitly mentioned there. Finally, the last item ensures that the "tree-likeness" of the initial segment is not violated.

Lemma 8.5 Let $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ be a $\mathcal{ZIQ}$-KB in Scott's normal form and let $q := \bigvee_i q_i$ be a union of rooted CQs. We have that $\mathcal{K} \not\models q$ if and only if there is an $(\mathcal{F}, \mathcal{K})$-initial segment $\mathcal{S}$ such that (i) $\mathcal{F}$ is an $(\mathrm{R}, \mathrm{C}, \mathrm{D})$-forest for an R bounded by $|\mathsf{ind}(\mathcal{K})|$, C exponential in $|\mathcal{T}|$, and D bounded by $|q|$, (ii) $(\mathcal{S}, \mathcal{T})$ is quasi-forest satisfiable, and (iii) $\mathcal{I}_{\mathcal{S}} \not\models q$.

Proof sketch. Note that by Lemma 7.58 the condition (ii) from the statement of the lemma is equivalent to saying that "$\mathcal{S}$ is a consistent summary of $\mathcal{K} \cup \{\mathsf{a} \not\approx \mathsf{b} \mid \mathsf{a}, \mathsf{b} \in \mathcal{F}, \mathsf{a} \neq \mathsf{b}\}$".

Suppose that $\mathcal{K} \not\models q$. Then, by Lemma 7.50, there exists an elegant model $\mathcal{I} \models \mathcal{K}$ with $\mathcal{I} \not\models q$. Take $\mathcal{J}$ be the restriction of $\mathcal{I}$ to all words of length at most $|q|$. Observe that: (i) $\mathcal{J}$ is a quasi-forest, (ii) the total number of roots in $\mathcal{J}$ is bounded by $|\mathsf{ind}(\mathcal{K})|$, (iii) the degree of every element is bounded by some N exponential w.r.t. $|\mathcal{T}|$ (due to the fact that $\mathcal{I}$ is elegant), (iv) $\mathcal{J}$ is of size exponential w.r.t. $|\mathcal{T}|$, (v) $\mathcal{J} \not\models q$. Note that the domain of $\mathcal{J}$, after a trivial renaming, can be seen as a $(\mathrm{R}, \mathrm{C}, \mathrm{D})$-forest $\mathcal{F}$ for parameters as indicated by the above conditions (ii), (iii), and D bounded by $|q|$. Hence, we can alter the interpretation of individual names in $\mathcal{J}$ so that the individual names from $\mathcal{F}$ are interpreted as the corresponding domain elements of $\mathcal{J}$. What remains to be done is to define a suitable summary $\mathcal{S}$. We construct $\mathcal{S}$ of $\mathcal{J}$ in exactly the same way as we did in Lemma 7.59 when constructing a consistent summary out of a quasi-forest model, and hence we prefer not to repeat the construction here. The only difference is that we include in $\mathcal{S}$ the axioms of the form $((=0\ \mathit{child}).\top)\,(\mathsf{a})$ for all the individual names $\mathsf{a} \in \mathcal{F}$ that are not leaves of $\mathcal{F}$. By design, $\mathcal{S}$ is an $(\mathcal{F}, \mathcal{K})$-initial segment and $\mathcal{I}_{\mathcal{S}} \not\models q$. It remains to show that $\mathcal{S}$ is consistent, but this is fairly straightforward. We modify $\mathcal{I}$ in the following way: (i) we interpret individual names from $\mathcal{F}$ in a way we did for $\mathcal{J}$, (ii) we remove from $\mathit{child}^{\mathcal{I}}$ all pairs of the form $(\mathsf{a}^{\mathcal{I}}, \mathsf{b}^{\mathcal{I}})$ for $\mathsf{a}, \mathsf{b} \in \mathcal{F}$, (iii) we include in $\mathrm{Root}^{\mathcal{I}}$ the interpretations of all names from $\mathcal{F}$. Note that the constructed structure is no longer a quasi-forest, but this can be easily fixed. To do so, we rename all the members of $\mathrm{Root}^{\mathcal{I}}$ to make them natural numbers. In addition to that we also rename all the elements of the form $w{\cdot}\mathrm{d}{\cdot}v$. for d being equal to the interpretation of some leaf of $\mathcal{F}$, to $\mathrm{d}{\cdot}v$. After such a renaming, the interpretation $\mathcal{I}$ becomes a quasi-forest again. Based on the modified interpretation $\mathcal{I}$, we can show that $\mathcal{S}$ is clearing-consistent and subtree-consistent in precisely the same way as we did in the proof of Lemma 7.59. Thus $\mathcal{S}$ is a consistent summary, which finishes the proof.

For the reverse direction, we apply Lemma 7.58 to infer the existence of a quasi-forest model $\mathcal{I}$ of $\mathcal{K} \cup \{\mathsf{a} \not\approx \mathsf{b} \mid \mathsf{a}, \mathsf{b} \in \mathcal{F}, \mathsf{a} \neq \mathsf{b}\}$ such that (i) $\mathcal{I}$ restricted to $\mathrm{Root}^{\mathcal{I}}$ is an $(\mathrm{R}, \mathrm{C}, \mathrm{D})$-forest (after a suitable renaming), (ii) $\mathcal{I}$ restricted to $\mathrm{Root}^{\mathcal{I}}$ does not satisfy q (this follows from the fact that all quasi-forest models of $(\mathcal{S}, \mathcal{T})$ have identical clearing, modulo the symbols that do not appear in $\mathcal{K}$ and $\mathcal{F}$), and (iii) only the roots of $\mathcal{I}$ that are leaves in $\mathcal{F}$ may have positive number of children. We next modify $\mathcal{I}$ in order to obtain a new interpretation $\mathcal{J}$ as follows:

- Remove from $\mathrm{Root}^{\mathcal{I}}$ all elements that do not interpret individuals being the roots of $\mathcal{F}$.
- Append to $\mathit{child}^{\mathcal{I}}$ all pairs of the form $(\mathrm{d}, \mathrm{e}) \in \Delta^{\mathcal{I}}$, where there are $\mathsf{a}, \mathsf{b} \in \mathcal{F}$ with $\mathrm{d} = \mathsf{a}^{\mathcal{I}}$ and $\mathrm{e} = \mathsf{b}^{\mathcal{I}}$ such that a is a parent of b in $\mathcal{F}$.
- Interpret all individual names outside $\mathsf{ind}(\mathcal{K})$ equally to some fixed name from $\mathsf{ind}(\mathcal{K})$.
- We rename $\mathcal{J}$ to make its domain a forest. For instance, we first rename all the elements d of $\Delta^{\mathcal{J}}$ to positive integers. Then, we rename each d with the unique child^{*}-path ρ (*i.e.* a word in $\mathrm{Root}^{\mathcal{J}} {\cdot} (\Delta^{\mathcal{J}})^{*}$) leading from the unique ancestor of d labelled with Root to d.

It can be readily verified that $\mathcal{J} \models \mathcal{K}$. By construction, it is also its quasi-forest model. Finally, we show that $\mathcal{J} \not\models q$. Indeed, by the fact that q is rooted, any match of q will contain only the elements from $\mathcal{J}$ that are words of length at most $|q|$, yielding a match in $\mathcal{I}$. □

Note that the forest $\mathcal{F}$ described above is exponential w.r.t. $|\mathcal{K}|$, and thus the initial segment $\mathcal{S}$ can be "guessed" in NExpTime. This allow us to provide the following adaptation of Procedure 5.

Procedure 7: Deciding Rooted Query Entailment in $\mathcal{ZIQ}$

```
Input: A $\mathcal{ZIQ}$-KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ and a union $q := \bigvee_{i=1}^{n} q_i$ of rooted conjunctive queries.
Output: True if and only if $\mathcal{K} \not\models q$.
1 Turn $\mathcal{T}$ into Scott's normal form.              // In PTime w.r.t. $|\mathcal{T}|$ due to Lemma 7.50.
2 Guess $(\mathcal{F}, \mathcal{K})$-initial segment $\mathcal{S}$ of $\mathcal{K}$ as in Lemma 8.5.      // Of size exponential w.r.t. $|\mathcal{K}|$ by
    Lemma 7.52 and the design of $\mathcal{F}$, and the guessing is implementable in NExp.
3 Foreach $q_i \in q$ return False if $\mathcal{I}_{\mathcal{S}} \models q_i$.  // Implementable in $|\mathcal{S}|^{|q|}$, and thus in Exp w.r.t. $|q| \cdot |\mathcal{K}|$.
4 Use Procedure 5 to verify that $(\mathcal{S}, \mathcal{T})$ is quasi-forest satisfiable and return True if this is the case.
    // Implementable in time $\mathrm{e}(|\mathcal{T}|) + \mathrm{p}(|\mathcal{S}| + |\mathcal{T}|) + \mathrm{p}(|\mathcal{S}| + |\mathcal{T}|) \cdot \mathrm{e}(|\mathcal{T}|)$ for some exponential function e and
    a polynomial function p, and thus in exponential time w.r.t. $|\mathcal{K}| \cdot |q|$.
```

It can be readily verified that Procedure 7 can be implemented with a nondeterministic exponential time Turing machine, and that it answers `True` if and only if $\mathcal{K}$ does not entail the input query. This is formally established by the following lemma.

Lemma 8.6 Procedure 7 returns `True` if and only if the input $\mathcal{ZOIQ}$-KB $\mathcal{K}$ does not entail the input query q. Moreover, Procedure 7 can can be implemented with a nondeterministic Turing machine of running time bounded by some function bounded exponentially w.r.t. $|q|$ and $|\mathcal{K}|$.

Proof. Correctness of Procedure 7 follows by the fact that the steps 2–4 described above verify precisely the conditions (i)—(iii) from Lemma 8.5, shown to be equivalent to $\mathcal{K} \not\models q$. For the running time of the algorithm, it suffices to bound the running time of each of the steps by some exponential function w.r.t. $|\mathcal{K}| \cdot |q|$. Indeed:

- The first step of the algorithm can be done in time polynomial w.r.t. $|\mathcal{T}|$ by Lemma 7.50.
- The second step requires a bit of work. As the desired parameters R, C and D are bounded, respectively, by $|\mathsf{ind}(\mathcal{K})|$, some exponential function in $|\mathcal{T}|$, and $|q|$, we see that their values can be guessed in time polynomial w.r.t. $|\mathcal{K}| \cdot |q|$. This implies that the desired $(\mathrm{R}, \mathrm{C}, \mathrm{D})$-forest $\mathcal{F}$ has size exponential w.r.t. $|\mathcal{K}| \cdot |q|$, and also that it can be guessed in time exponential w.r.t. $|\mathcal{K}| \cdot |q|$, *e.g.* by first guessing the roots and then proceeding in a top-down manner. The construction of the initial segment can be then done in time polynomial w.r.t. $|\mathcal{K}| + |\mathcal{F}|$ (and thus in time exponential w.r.t. $|\mathcal{K}| \cdot |q|$) by Lemma 7.52 (adding the extra axioms required by Definition 8.4 can be handled easily). Thus, the second step of Procedure 7 can be implemented in non-deterministic exponential time.
- The verification whether the interpretation $\mathcal{I}_{\mathcal{S}}$ satisfies the query q_i can be done easily. It suffices to iterate through all the possible variable assignments $\eta \colon \mathrm{Var}(q_i) \to \mathcal{F}$ (there are at most $|\mathcal{S}|^{|q_i|}$ of them), and for each of them evaluate the query based on the content of $\mathcal{S}$ (*e.g.* to verify whether $r(x, y)$ holds under the assignment $x \mapsto \mathsf{a}$, $y \mapsto y$ we check if $r(\mathsf{a}, \mathsf{b})$ belongs to $\mathcal{S}$). Note that as $|\mathcal{S}|$ is exponential w.r.t. $|q| \cdot |\mathcal{K}|$, the value $|\mathcal{S}|^{|q_i|}$ is exponential w.r.t. $|q| \cdot |\mathcal{K}|$ as well.
- For the remaining step of Procedure 7, we invoke Lemma 7.61. It tells us that Procedure 5 can be implemented with a nondeterministic Turing machine in such a way that for the input $\mathcal{K}' := (\mathcal{A}', \mathcal{T}')$ it runs in time bounded by $\mathrm{e}(|\mathcal{T}'|) + \mathrm{p}(|\mathcal{K}'|) + \mathrm{p}(|\mathcal{K}'|) \cdot \mathrm{e}(|\mathcal{T}'|)$ for some functions e and p, that are, respectively, exponential and polynomial. Thus, by assigning $\mathcal{K}' := (\mathcal{S}, \mathcal{T})$ and $\mathcal{T}' := \mathcal{T}$ from the input, we can bound the running time of the 4-th step by $\mathrm{e}(|\mathcal{T}|) + \mathrm{p}(|\mathcal{S}| + |\mathcal{T}|) + \mathrm{p}(|\mathcal{S}| + |\mathcal{T}|) \cdot \mathrm{e}(|\mathcal{T}|)$. This is clearly exponential w.r.t. $|\mathcal{K}| \cdot |q|$ due to the fact that the size of $\mathcal{S}$ is only exponential in terms of $|q| \cdot |\mathcal{K}|$.

By aggregating all the running times given above, we conclude the proof. □

Based on Lemma 8.6 we can now conclude the main result of this section.

Theorem 8.7
The entailment problem for unions of rooted CQs over $\mathcal{ZIQ}$-KBs is coNExpTime-complete.

The coNExpTime-hardness of rooted entailment holds already for $\mathcal{ALCI}$ [Lut08a, Thm. 1]. Interestingly enough, in the next section we establish a novel lower bound for $\mathcal{ALC}$Self.

8.3 Entailment of Rooted Queries: A Novel Lower Bound

The goal of this section is to present a novel coNExpTime lower bound for the entailment of rooted conjunctive queries in $\mathcal{ALC}$Self. We start by providing a short overview.

8.3.1 Overview of the Lower Bound Proof

Our lower bound proof is inspired by two hardness proofs: the coNExpTime-hardness proof of the entailment of rooted queries in $\mathcal{ALCI}$ by Lutz [Lut08a, Sec. 3], and the 2ExpTime-hardness proof of the query entailment problem for $\mathcal{ALC}$ with the Self operator by Bednarczyk and Rudolph [BR22]. Despite the obvious similarities between our approaches, we believe that our proof is highly non-trivial.

We reduce from the NExpTime-complete torus[3] tiling problem introduced in Section 8.1.2. More precisely, given an instance of the torus tiling problem $(\mathfrak{D}, \bar{\mathrm{c}})$ we construct a rooted conjunctive query q and an $\mathcal{ALC}$Self-KB $\mathcal{K}$ (both of size polynomial w.r.t. $|\mathfrak{D}|+|\bar{\mathrm{c}}|$) such that $\mathcal{K} \not\models q$ if and only if $(\mathfrak{D}, \bar{\mathrm{c}})$ has a solution. The rough idea is that the models of $\mathcal{K}$ encode exponential tori (possibly incorrectly) labelled with tiles, while every match of the query q detects mismatches in tiling. After completing all the missing details, this yields coNExpTime-hardness of the entailment problem of rooted CQ for $\mathcal{ALC}$Self.

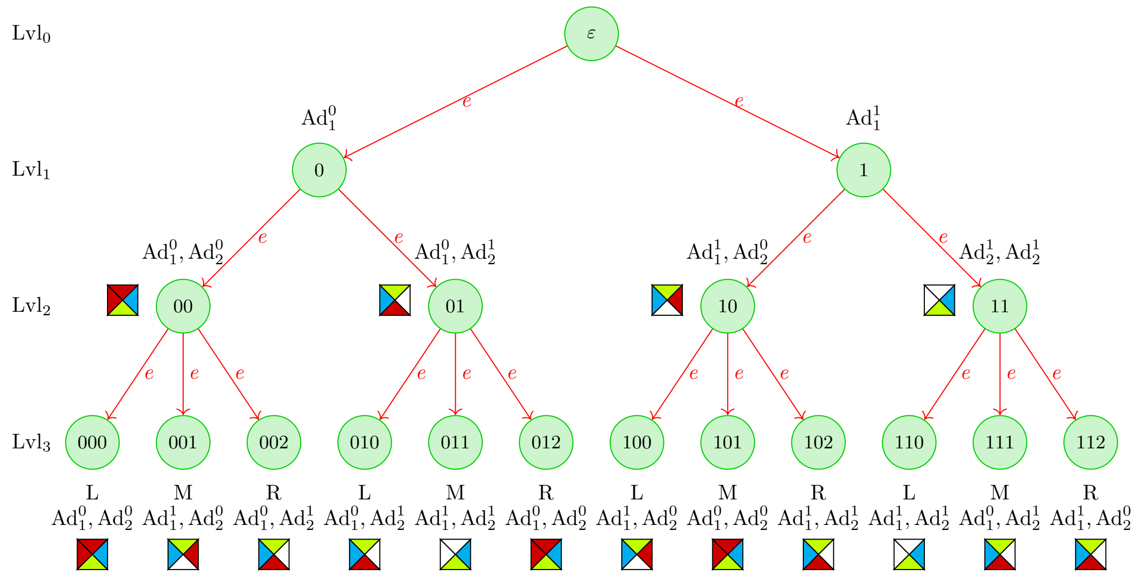


Figure 8.1: A visualisation of an example treepod for $n = 1$ that represents the solution ξ from Example 8.3. For brevity, we wrote ⊠ instead of $\mathrm{C}_{⊠}$ (and similarly for other tiles).

Step I: Representing tori and tilings. Suppose that all the ingredients of the input, namely an integer n, a torus tiling system $\mathfrak{D} \coloneqq (\mathrm{T}, \mathrm{H}, \mathrm{V})$, and an initial condition $\bar{\mathrm{c}} \coloneqq (\bar{\mathrm{c}}_0, \ldots, \bar{\mathrm{c}}_{n-1}) \in \mathrm{T}^n$, are given. Similarly to the plethora of other reductions from the tiling problem for modal and description logics we represent the exponentially large torus $\mathbb{Z}_{2^n} \times \mathbb{Z}_{2^n}$ using a binary tree of height $2n$. Such trees can be easily expressed with an $\mathcal{ALC}$-concept of polynomial size w.r.t. n that employs the (edge) role name e. Note that every leaf of such a tree naturally represents a position $(x, y) \in 2^n \times 2^n$. Such an "address" (x, y) is also represented by means of concepts $\mathrm{Ad}_1^{b_1}, \ldots, \mathrm{Ad}_n^{b_n}$ and $\mathrm{Ad}_{n+1}^{b_{n+1}}, \ldots, \mathrm{Ad}_{2n}^{b_{2n}}$, where $b_1 \ldots b_n$ and $b_{n+1} \ldots b_{2n}$ are binary representations of the positions x and y on n bits. In addition to that, we label every leaf with precisely one concept of the form C_{t} for $\mathrm{t} \in \mathrm{T}$ indicating the selection of tiles. This provides the

[3]The use of tori rather than rectangles is handy, as in tori every element has the horizontal and the vertical neighbour.

correspondence between the decorated binary trees and the tile assignments $\xi \colon (2^n \times 2^n) \to \mathrm{T}$. The main challenge is to guarantee that the assigned tiles do not violate the matching relations.

Step II: Localisation of successors and treepods. We shift a bit the difficulty of the problem by reducing the problem of detecting the mismatches in tiling to the certain equality test. The first ingredient is to "localise" horizontal and vertical successors of all the elements. Given the trees described in the previous step, we equip every one of its leaves with three extra children labelled, respectively, with concepts L, M, R. Such extra children, similarly to their parents, also carry the tile concepts. The intended role of the extra children of an element representing some position (x, y), is that they are intended to represent, respectively, the positions (x, y), $(x \oplus_{2^n} 1, y)$, $(x, y \oplus_{2^n} 1)$. Call the resulting structures *treepods*.

In treepods, the verification of horizontal and vertical matching relations becomes easy as the consistency check can be implemented locally. For instance, whenever the tiles t, t' are not H-compatible, our ontology would contain the GCI $\mathrm{C_t} \sqsubseteq \neg\exists e.(\mathrm{M} \sqcap \mathrm{C_{t'}})$. This approach works, however, only under a naïve assumption that all the nodes representing the same position of a torus carry precisely the same tile. This "tile equality test" will be achieved with the query. Before defining the query, yet another gadget is needed.

Step III: Tentacles. To prepare the query implementing the aforementioned (in)equality check, we design an alternative way of encoding positions and tiles in treepods. This is achieved by *tentacles*, namely the paths endowed with various self-loops replacing the leaves of treepods. Consult the following example.

Example 8.8. Let $n = 1$ and suppose that $\mathscr{D}$ contains only the tiles ⊠,⊠, ⊠, and ⊠. Then the tentacle that is going to be attached to the node 100 from the treepod depicted in Figure 8.1 is presented below. In its construction we employ fresh role names ad_1^0, ad_1^1, ad_2^0, ad_2^1, $r_{⊠}$, $r_{⊠}$, $r_{⊠}$, $r_{⊠}$. The symbol $\star$ in the picture below serves as the wildcard for all the mentioned role names. Note that the leaves (and only them) of tentacles are decorated with e-self-loops.

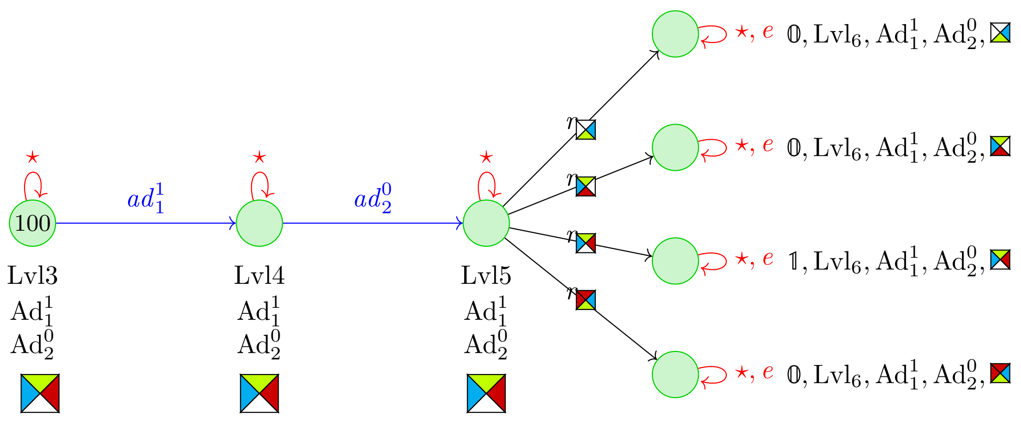

$\star$
$\star$
$\star$
100
ad_1^1
ad_2^0
Lvl3
Lvl4
Lvl5
Ad_1^1
Ad_2^0
Ad_1^1
Ad_2^0
Ad_1^1
Ad_2^0
$r_{⊠}$
$r_{⊠}$
$r_{⊠}$
$r_{⊠}$
$\star, e$ $\mathbb{0}, \mathrm{Lvl}_6, \mathrm{Ad}_1^1, \mathrm{Ad}_2^0,$ ⊠
$\star, e$ $\mathbb{0}, \mathrm{Lvl}_6, \mathrm{Ad}_1^1, \mathrm{Ad}_2^0,$ ⊠
$\star, e$ $\mathbb{1}, \mathrm{Lvl}_6, \mathrm{Ad}_1^1, \mathrm{Ad}_2^0,$ ⊠
$\star, e$ $\mathbb{0}, \mathrm{Lvl}_6, \mathrm{Ad}_1^1, \mathrm{Ad}_2^0,$ ⊠

Figure 8.2: The tentacle attached to the node 100 of the treepod from Figure 8.1.

Recall that every leaf d of a treepod (of height $2n$) is labelled with the selection of concepts $\mathrm{Ad}_1^{b_1}, \ldots, \mathrm{Ad}_n^{b_n}$ and $\mathrm{Ad}_{n+1}^{b_{n+1}}, \ldots, \mathrm{Ad}_{2n}^{b_{2n}}$ representing its address from $\mathbb{Z}_2^n \times \mathbb{Z}_{2^n}$ is binary. In the construction of tentacles, we enrich d with the outgoing path composed of fresh roles $ad_1^{b_1}, \ldots, ad_n^{b_n}, ad_{n+1}^{b_{n+1}}, \ldots, ad_{2n}^{b_{2n}}$. Such a path leads to a node that has an r_t-successor for every tile $\mathrm{t} \in \mathrm{T}$. Precisely one of such r_t-successors is labelled with the concept $\mathbb{1}$, indicating the tile labelling the node d. The remaining r_t-successors are labelled with $\mathbb{0}$. All the elements of a tentacle are endowed with various r_t- and ad_i^j-self-loops. Additionally, the leaves of tentacles carry e-self-loops. By replacing every leaf of a treepod with the corresponding tentacle, we obtain *jellyfishes*. From a bird's eye view, jellyfishes look as depicted by Figure 8.3.

Summarising, jellyfishes provide an alternative way of encoding address of nodes and their tiles by means of attached outgoing paths labelled by various roles. For instance, the node 100 from the treepod depicted in Figure 8.1 is labelled with Ad_1^1, Ad_2^0 and $\mathrm{C}_{⊠}$, while in the corresponding jellyfish is it equipped with the outgoing $ad_1^1 ad_2^0 r_{⊠}$-path leading to an element labelled with $\mathbb{1}$. We exploit such paths in the query.

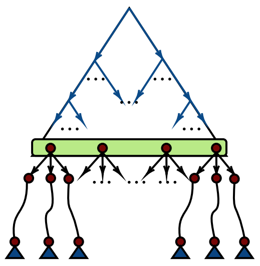


Figure 8.3: A high-level visualisation of a jellyfish.

Step IV: The query. Rather than presenting the desired query finding mismatches in tilings in full details, we provide some of its important subqueries and illustrate them with examples. In what follows, we will always consider the jellyfish obtained by replacing the treepod from Figure 8.1 with the corresponding tentacles (for instance the node 100 is replaced with the tentacle from Figure 8.2). For brevity, we once more employ the path syntax of conjunctive queries (consult Section 2.4 if needed). Recall that by a path-shaped conjunctive query we understand an expression of the form

$$(\mathrm{A}_0?; r_1; \mathrm{A}_1?; r_2; \mathrm{A}_2?; \ldots; \mathrm{A}_{n-1}?; r_n; \mathrm{A}_n?)(x_0, x_n)$$

with all $r_i \in \mathbf{N_R}$ and $\mathrm{A}_i \in \mathbf{N_C} \cup \{\top\}$, serving as a shorthand for

$$\bigwedge_{i=0}^{n} \mathrm{A}_i(x_i) \wedge \bigwedge_{i=1}^{n} r_i(x_{i-1}, x_i).$$

Whenever A_i happens to be $\top$, it will be removed from the expression; this does not create ambiguities. We write r^n as the abbreviation or $r; \ldots; r$, where r is repeated n times. As only the starting and the final variables of path queries matters, we write $q(x, y)$ to denote the query obtained by renaming the variable with the lowest index to x and the variable with the highest index to y. The alternative syntax for path-shaped CQs is just syntactic sugar and our queries should not be mistaken *e.g.* for regular path queries.

Example 8.9. Consider the jellyfish $\mathcal{Y}$ described above, and the following query $q_{\downarrow}$:

$$\big(\mathrm{Lvl}_0?e; \mathrm{Lvl}_1?e; \mathrm{Lvl}_2?e; \mathrm{Lvl}_3?[ad_1^0; ad_1^1]\mathrm{Lvl}_4?[ad_2^0; ad_2^1]; \mathrm{Lvl}_5?[r_{\boxtimes}; r_{\boxtimes}; r_{\boxtimes}; r_{\boxtimes}]\mathrm{Lvl}_6?\big)\,(x, y).$$

We strongly encourage the reader to play with the query to understand how it matches jellyfishes. We can readily verify that (i) for any match η for $q_{\downarrow}$ and $\mathcal{Y}$ we have that $\eta(x)$ is the root of $\mathcal{Y}$ and $\eta(y)$ is a leaf of $\mathcal{Y}$, and (ii) for every leaf d of $\mathcal{Y}$ there is a match η for $q_{\downarrow}$ and $\mathcal{Y}$ that maps y to d and x to the root of $\mathcal{Y}$.

It is especially important to observe how our query "traverses" $\mathcal{Y}$. Having a match η for $q_{\downarrow}$ and $\mathcal{Y}$, imagine that while reading the consecutive parts of the query, the element $\eta(x)$ moves towards the element $\eta(y)$. When scanning any of the subexpressions of $q_{\downarrow}$ indicated with square brackets, note that for each role r our element will either stay in place (by employing an r-self-loop) or it moves forward by following an r-edge.

We employ the self-loops that occur in jellyfishes as a handy way of introducing the disjunction of paths to our queries. This intuition becomes even more prominent with the following example, where we present the query relating two leaves such that our query either (i) loops at the leaf, then moves to the root, and then goes down to the another leaf, or (ii) moves to the root, goes to the another leaf and then loops.

Example 8.10. Consider the jellyfish $\mathcal{Y}$ from the previous example, and the following query $q^1_{\mathrm{eq}}(x, x')$:

$$\mathbb{0}(x) \wedge \mathrm{Lvl}_6(x) \wedge \big(e; e; e; ad_1^0; [ad_2^0; ad_2^1]; [r_{\boxtimes}; r_{\boxtimes}; r_{\boxtimes}; r_{\boxtimes}]\big)\,(y, x) \wedge$$

$$\wedge \big(e; e; e; ad_1^0; [ad_2^0; ad_2^1]; [r_{\boxtimes}; r_{\boxtimes}; r_{\boxtimes}; r_{\boxtimes}]\big)\,(y, z) \wedge \mathrm{Lvl}_6(z) \wedge$$

$$\mathrm{Lvl}_6(z) \wedge \big(e; e; e; ad_1^1; [ad_2^0; ad_2^1]; [r_{\boxtimes}; r_{\boxtimes}; r_{\boxtimes}; r_{\boxtimes}]\big)\,(y', z) \wedge$$

$$\wedge\Big(e; e; e; ad_1^1; [ad_2^0; ad_2^1]; [r_{\boxtimes}; r_{\boxtimes}; r_{\boxtimes}; r_{\boxtimes}]\Big)\big(y', x'\big) \wedge \mathrm{Lvl}_6(x') \wedge \mathbb{1}(x').$$

We stress that the role name ad_1^1 does not appear in the first two lines of q_{eq}^1, and that the role name ad_1^0 does not appear in the last two lines of q_{eq}^1. Consider a match η for q_{eq}^1 and $\mathcal{Y}$. Clearly, $\eta(x) \neq \eta(x')$.

Two observations first. First, by the presence of concepts Lvl_6 we know that $\eta(x)$, $\eta(x')$, and $\eta(z)$ are leaves of $\mathcal{Y}$. Second, as $\eta(y)$ and $\eta(y')$ have outgoing e-path of length three, by the design of treepods and jellyfishes, only two options are possible: each of $\eta(y)$ and $\eta(y')$ is either a leaf or the root of $\mathcal{Y}$. We would like to show that either both $\eta(x)$ and $\eta(x')$ are labelled with Ad_1^0, or both of them are labelled with Ad_1^1. We strongly encourage the reader to analyse the reasoning sketched below. Consider the following cases:

- $\eta(y)$ is the root of $\mathcal{Y}$.
  Then by the fact that $\eta(x)$ is a leaf of $\mathcal{Y}$ and the first line of the query q_{eq}^1, we infer that the root-to-leaf path linking $\eta(y)$ and $\eta(x)$ employs the role ad_1^0. Thus, $\eta(x)$ is labelled with Ad_1^0. Similarly, we obtain that $\eta(z)$ is a leaf of $\mathcal{Y}$ labelled with Ad_1^0 (note that it is possible that $\eta(z)$ is equal to $\eta(x)$). By the fact that $\eta(z)$ is labelled with Ad_1^0, there is no path leading from the root of $\mathcal{Y}$ to $\eta(z)$ that matches the third line of the query q_{eq}^1. Hence, the whole such path "collapses into a self-loop", yielding $\eta(y') = \eta(z)$. Reasoning similarly, we infer that $\eta(y') = \eta(x')$ holds. Thus, $\eta(x')$ is labelled with Ad_1^0 (as $\eta(z)$ is).
- $\eta(y)$ is a leaf of $\mathcal{Y}$.
  Observe that then this implies the equality $\eta(y) = \eta(x) = \eta(z)$. We next consider the node $\eta(y')$. We see that $\eta(y')$ cannot be a leaf of $\mathcal{Y}$. Indeed, with a similar reasoning this would imply the equality $\eta(y') = \eta(x') = \eta(z)$, leading to a contradiction with the fact that $\eta(x) \neq \eta(x')$. Hence, $\eta(y')$ the root of $\mathcal{Y}$. By the same reasoning as in the previous case (and by relying on the last two lines of the query q_{eq}^1), we infer that $\eta(z)$ is labelled with Ad_1^1. This in turn yields that $\eta(x')$ is labelled with Ad_1^1. By the equality $\eta(x) = \eta(z)$, we conclude that $\eta(x)$ is also labelled with Ad_1^1.

As we observed in Example 8.10, the presented query maps the variables x and x' to different leaves of a jellyfish that have their first bit of the address equal. Reasoning similarly, we produce analogous queries also for the other bits of addresses and for the assignment of tiles. By joining all these queries together we obtain the desired query detecting (in)equality of the tiling assignment. This concludes our reduction.

8.3.2 Treepods

$(\mathcal{D}, \bar{\mathrm{c}})$-treepods are tree-based structures (with the role e indicating the edge relation) of depth $2|\bar{\mathrm{c}}|+1$ that are mostly binary trees with an exception for the second-to-last level where each element have three children. The depth of an element, namely its distance from the root, is indicated by the satisfaction of Lvl_i-concepts. In treepods, every node at depth $i \leq 2|\bar{\mathrm{c}}|$ is decorated with concepts $\mathrm{Ad}_1^{b_1}, \ldots, \mathrm{Ad}_i^{b_i}$ for some word $b_1 \ldots b_i \in \{0,1\}^i$ that naturally encodes (in binary) the *address* of a node, *i.e.* its number according to the prefix of a treepod (note that less significant bits carry the higher number). Treating the numbers represented by first $|\bar{\mathrm{c}}|$ $\mathrm{Ad}_i^{b_i}$-concepts and the number represented by the last $|\bar{\mathrm{c}}|$ $\mathrm{Ad}_i^{b_i}$-concepts separately, every node at depth $2|\bar{\mathrm{c}}|$ *encodes* some position (x, y) in $\mathbb{Z}_{2^{|\bar{\mathrm{c}}|}} \times \mathbb{Z}_{2^{|\bar{\mathrm{c}}|}}$. Any node at depth $2|\bar{\mathrm{c}}|$ encoding (x, y) is additionally equipped with three different e-successors, labelled respectively with $\mathrm{L}, \mathrm{M}, \mathrm{R}$, and they are constrained to encode in binary, respectively, the positions (x, y), $(x \oplus_{2^n} 1, y)$, and $(x, y \oplus_{2^n} 1)$. This way, every node at depth $2|\bar{\mathrm{c}}|$ has "local" access to its vertical and horizontal successor. Finally, each leaf of a $(\mathcal{D}, \bar{\mathrm{c}})$-treepod is decorated with precisely one "tile concept" $\mathrm{C_t}$ for $\mathrm{t} \in \mathrm{T}$ in a way that respect matching relations of the tiling system $\mathcal{D}$. Consult Figure 8.1 and Definition 8.11.

Definition 8.11 Let $\mathcal{D} := (\mathrm{T}, \mathrm{H}, \mathrm{V})$ be a torus tiling system, and $\bar{\mathrm{c}} := (\bar{\mathrm{c}}_0, \ldots, \bar{\mathrm{c}}_{n-1}) \in \mathrm{T}^n$ be an initial condition. A **$(\mathcal{D}, \bar{\mathrm{c}})$-treepod** $\mathcal{T} := (\Delta^{\mathcal{T}}, \cdot^{\mathcal{T}})$ is an interpretation fulfilling all the requirements below.

- $\Delta^{\mathcal{T}} = \{0,1\}^{\leq 2n} \cup \{0,1\}^{=2n} \cdot \{0,1,2\}$.
- $e^{\mathcal{T}} = \{(w, w0), (w, w1) \mid |w| < 2n\} \cup \{(w, w0), (w, w1), (w, w2) \mid |w| = 2n\}$.
- $\mathrm{Lvl}_i^{\mathcal{T}} = \{w \in \Delta^{\mathcal{T}} \mid |w| = i\}$.
- $\mathrm{L}^{\mathcal{T}} = \{0,1\}^{=2n} \cdot \{0\}$, $\mathrm{M}^{\mathcal{T}} = \{0,1\}^{=2n} \cdot \{1\}$, $\mathrm{R}^{\mathcal{T}} = \{0,1\}^{=2n} \cdot \{2\}$.
- $(\mathrm{Ad}_i^b)^{\mathcal{T}} \cap \{0,1\}^{\leq 2n} = \{w \in \Delta^{\mathcal{T}} \mid |w| \geq i$ and its i-th letter is $b\}$.
- For all $w := w_1 \ldots w_{2n} \in \{0,1\}^{=2n}$, encoding a position $(x, y) \in \mathbb{Z}_{2^n} \times \mathbb{Z}_{2^n}$, we have that:

- $w0$ encodes the position (x, y), *i.e.* $w0 \in \bigcap_{i=1}^{2n} (\mathrm{Ad}_i^{w_i})^{\mathcal{T}}$,
- $w1$ encodes the position $(x \oplus_{2^n} 1, y)$, *i.e.* $w1 \in \bigcap_{i=1}^{n} (\mathrm{Ad}_i^{u_i})^{\mathcal{T}} \cap \bigcap_{i=n+1}^{2n} (\mathrm{Ad}_i^{w_i})^{\mathcal{T}}$,
- $w2$ encodes the position $(x, y \oplus_{2^n} 1)$, *i.e.* $w2 \in \bigcap_{i=1}^{n} (\mathrm{Ad}_i^{w_i})^{\mathcal{T}} \cap \bigcap_{i=n+1}^{2n} (\mathrm{Ad}_i^{u_i})^{\mathcal{T}}$,

where $u := u_1 \dots u_{2n} \in \{0, 1\}^{=2n}$ encodes the position $(x \oplus_{2^n} 1, y \oplus_{2^n} 1) \in \mathbb{Z}_{2^n} \times \mathbb{Z}_{2^n}$.

- The concepts $\mathrm{C}_{\mathrm{t}}^{\mathcal{T}}$ for $\mathrm{t} \in \mathrm{T}$ are pairwise disjoint and $\bigcup\{\mathrm{C}_{\mathrm{t}}^{\mathcal{T}} \mid \mathrm{t} \in \mathrm{T}\} = \mathrm{Lvl}_{2n+1}^{\mathcal{T}} \cup \mathrm{Lvl}_{2n}^{\mathcal{T}}$.
- For all $w \in \mathrm{Lvl}_{2n}^{\mathcal{T}}$ and all $\mathrm{t}_0, \mathrm{t}_1, \mathrm{t}_2 \in \mathrm{T}$ such that $wi \in \mathrm{C}_{\mathrm{t}_i}^{\mathcal{T}}$ we have $(\mathrm{t}_0, \mathrm{t}_1) \in \mathrm{H}$ and $(\mathrm{t}_0, \mathrm{t}_2) \in \mathrm{V}$. For brevity we say that w **carries a tile** t whenever $w \in \mathrm{C}_{\mathrm{t}}^{\mathcal{T}}$.
- For any $0 \leq i < n$ we have that all the leaves encoding the position $(i, 0)$ belong to $\mathrm{C}_{\bar{\mathrm{c}}_i}^{\mathcal{T}}$.

A $(\mathcal{D}, \bar{\mathrm{c}})$-treepod $\mathcal{T}$ is **proper** if there is no position $(x, y) \in \mathbb{Z}_{2^{|\bar{\mathrm{c}}|}} \times \mathbb{Z}_{2^{|\bar{\mathrm{c}}|}}$ and no two leaves $w, w' \in \Delta^{\mathcal{T}}$ encoding (x, y) such that w and w' carry different tiles from T.

With every proper $(\mathcal{D}, \bar{\mathrm{c}})$-treepod $\mathcal{T}$ we associate a map $\xi \colon \mathbb{Z}_{2^{|\bar{\mathrm{c}}|}} \times \mathbb{Z}_{2^{|\bar{\mathrm{c}}|}} \to \mathrm{T}$ defined as $\xi(x, y) := \mathrm{t}_{(x,y)}$, where $\mathrm{t}_{(x,y)}$ is the unique[4] tile carried by any leaf of $\mathcal{T}$ encoding the position (x, y). In this case we say that $\mathcal{T}$ *represents* ξ. The following Fact 8.12 is a direct consequence of our design choices and Definition 8.11.

Fact 8.12. Fix $\mathcal{D} := (\mathrm{T}, \mathrm{H}, \mathrm{V})$ and $\bar{\mathrm{c}} \in \mathrm{T}^n$ as in Definition 8.11. Then (a) for any proper $(\mathcal{D}, \bar{\mathrm{c}})$-treepod $\mathcal{T}$, the function $\xi \colon \mathbb{Z}_{2^n} \times \mathbb{Z}_{2^n} \to \mathrm{T}$ represented by $\mathcal{T}$ is a solution to $(\mathcal{D}, \bar{\mathrm{c}})$, and (b) for any function $\xi \colon \mathbb{Z}_{2^n} \times \mathbb{Z}_{2^n} \to \mathrm{T}$ compatible with $\bar{\mathrm{c}}$ that covers $\mathbb{Z}_{2^n} \times \mathbb{Z}_{2^n}$ there exists a proper $(\mathcal{D}, \bar{\mathrm{c}})$-treepod representing ξ.

In the rest of this section we show how to define $(\mathcal{D}, \bar{\mathrm{c}})$-treepods in $\mathcal{ALC}$ with concepts of size polynomial w.r.t $(|\mathcal{D}| + |\bar{\mathrm{c}}|)$. The properness conditions will be enforced afterwards with a help of a query. We stress that, as usual in this type of reductions, we cannot formalise $(\mathcal{D}, \bar{\mathrm{c}})$-treepods up to isomorphism, but the axiomatisation provided is sufficient for our purposes in a sense made formally precise in Lemma 8.13.

1. Concepts $\mathrm{Treepod}^i_{(\mathcal{D},\bar{\mathrm{c}})}$ axiomatise subtrees of $(\mathcal{D}, \bar{\mathrm{c}})$-treepods starting at depth $0 \leq i \leq 2|\bar{\mathrm{c}}|{+}1$.

   (Treepod$[i]$**)** $\mathrm{Treepod}^i_{(\mathcal{D},\bar{\mathrm{c}})} := \mathrm{Succ}^i_{(\mathcal{D},\bar{\mathrm{c}})} \sqcap \mathrm{CorrLvl}^i_{(\mathcal{D},\bar{\mathrm{c}})} \sqcap \mathrm{CorrDir}^i_{(\mathcal{D},\bar{\mathrm{c}})} \sqcap \mathrm{CorrAd}^i_{(\mathcal{D},\bar{\mathrm{c}})} \sqcap \mathrm{CorrT}^i_{(\mathcal{D},\bar{\mathrm{c}})}$,

   where the concepts $\mathrm{Succ}_{(\mathcal{D},\bar{\mathrm{c}})}$, $\mathrm{CorrLvl}_{(\mathcal{D},\bar{\mathrm{c}})}$, $\mathrm{CorrDir}_{(\mathcal{D},\bar{\mathrm{c}})}$, $\mathrm{CorrAd}_{(\mathcal{D},\bar{\mathrm{c}})}$, $\mathrm{CorrT}_{(\mathcal{D},\bar{\mathrm{c}})}$ will be defined soon.

2. Any element from a $(\mathcal{D}, \bar{\mathrm{c}})$-treepod located at depth $0 \leq i < 2|\bar{\mathrm{c}}|$ has two e-successors: a left one (with the $(i{+}1)$-th bit of its address off) and a right one (with the $(i{+}1)$-th bit of its address on). Moreover, elements at depth $2|\bar{\mathrm{c}}|$ have three e-successors, labelled respectively with L, M, and R.

   (Succ$[2|\bar{\mathrm{c}}|{+}1]$**)** $\mathrm{Succ}^{2|\bar{\mathrm{c}}|+1}_{(\mathcal{D},\bar{\mathrm{c}})} := \top$

   (Succ$[2|\bar{\mathrm{c}}|]$**)** $\mathrm{Succ}^{2|\bar{\mathrm{c}}|}_{(\mathcal{D},\bar{\mathrm{c}})} := \bigsqcap_{\mathrm{D} \in \{\mathrm{L},\mathrm{M},\mathrm{R}\}} (\exists e.\mathrm{D}) \ \sqcap \ \forall e. \left(\mathrm{Treepod}^{2|\bar{\mathrm{c}}|+1}_{(\mathcal{D},\bar{\mathrm{c}})}\right)$

   (Succ$[i]$**)** $\mathrm{Succ}^i_{(\mathcal{D},\bar{\mathrm{c}})} := \exists e.\mathrm{Ad}^0_{i+1} \ \sqcap \ \exists e.\mathrm{Ad}^1_{i+1} \ \sqcap \ \forall e.\mathrm{Treepod}^{i+1}_{(\mathcal{D},\bar{\mathrm{c}})}$

3. Any element at depth i is labelled with the concept Lvl_i (and with no Lvl_j for all other j).

   (Lvl$[i]$**)** $\mathrm{CorrLvl}^i_{(\mathcal{D},\bar{\mathrm{c}})} := \mathrm{Lvl}_i \sqcap \bigsqcap_{1 \leq j \leq 4|\bar{\mathrm{c}}|+2, j \neq i} \neg\mathrm{Lvl}_j$

4. Only the leaves of $(\mathcal{D}, \bar{\mathrm{c}})$-treepods are labelled with (precisely one of) "direction concepts" L, M, R.

   (Num$[2|\bar{\mathrm{c}}|{+}1]$**)** $\mathrm{CorrDir}^{2|\bar{\mathrm{c}}|+1}_{(\mathcal{D},\bar{\mathrm{c}})} := [\mathrm{L} \sqcap \neg\mathrm{M} \sqcap \neg\mathrm{R}] \sqcup [\neg\mathrm{L} \sqcap \mathrm{M} \sqcap \neg\mathrm{R}] \sqcup [\neg\mathrm{L} \sqcap \neg\mathrm{M} \sqcap \mathrm{R}]$

   (Num$[i]$**)** $\mathrm{CorrDir}^i_{(\mathcal{D},\bar{\mathrm{c}})} := \neg\mathrm{L} \sqcap \neg\mathrm{M} \sqcap \neg\mathrm{R}$

5. The elements of $(\mathcal{D}, \bar{\mathrm{c}})$-treepods at depth $0 \leq i \leq 2|\bar{\mathrm{c}}|$ are appropriately labelled with "address concepts" encoding the position w.r.t the prefix ordering of the treepod. When considering elements at depth $2|\bar{\mathrm{c}}|$ encoding the positions (x, y), we additionally take care of their e-successors to make them encode the positions (x, y), $(x \oplus_{2^n} 1, y)$, and $(x, y \oplus_{2^n} 1)$. This is achieved by axiomatising the usual properties of addition under binary encodings of numbers [BHLS17, p. 127].

   (Adr$[2|\bar{\mathrm{c}}|{+}1]$**)** $\mathrm{CorrAd}^{2|\bar{\mathrm{c}}|+1}_{(\mathcal{D},\bar{\mathrm{c}})} := \bigsqcap_{i=1}^{2|\bar{\mathrm{c}}|} \bigsqcup_{b=0}^{1} \left(\mathrm{Ad}_i^b \sqcap \neg\mathrm{Ad}_i^{1-b}\right)$

   (Adr$[2|\bar{\mathrm{c}}|]$**)** $\mathrm{CorrAd}^{2|\bar{\mathrm{c}}|}_{(\mathcal{D},\bar{\mathrm{c}})} := \bigsqcap_{i=1}^{2|\bar{\mathrm{c}}|} \bigsqcup_{b=0}^{1} \left(\mathrm{Ad}_i^b \sqcap \neg\mathrm{Ad}_i^{1-b}\right)$

   $\sqcap \bigsqcap_{\mathrm{D} \in \{\mathrm{L},\mathrm{M},\mathrm{R}\}} \bigsqcap_{j=1}^{2|\bar{\mathrm{c}}|} \left[(\forall e.[\mathrm{D} \to (\mathrm{Ad}_i^0 \sqcap \neg\mathrm{Ad}_i^1)] \sqcup \forall e.[\mathrm{D} \to (\neg\mathrm{Ad}_i^0 \sqcap \mathrm{Ad}_i^1)]) \right]$

[4] This is well-defined by disjointness of concepts $\mathrm{C}_{\mathrm{t}}^{\mathcal{T}}$ for $\mathrm{t} \in \mathrm{T}$, and properness of $\mathcal{T}$.

$\sqcap \bigsqcap_{i=1}^{2|\bar{c}|} \bigsqcap_{b=0}^{1} \mathrm{Ad}_i^b \to \forall e. \left(\mathrm{L} \to [\mathrm{Ad}_i^b \sqcap \neg \mathrm{Ad}_i^{1-b}]\right)$
$\sqcap \downarrow_{(\mathcal{D},\bar{c})}^{\oplus_\mathrm{H}} \sqcap \bigsqcap_{i=|\bar{c}|+1}^{2|\bar{c}|} \bigsqcap_{b=0}^{1} \mathrm{Ad}_i^b \to \forall e. \left(\mathrm{M} \to [\mathrm{Ad}_i^b \sqcap \neg \mathrm{Ad}_i^{1-b}]\right)$
$\sqcap \downarrow_{(\mathcal{D},\bar{c})}^{\oplus_\mathrm{V}} \sqcap \bigsqcap_{i=1}^{|\bar{c}|} \bigsqcap_{b=0}^{1} \mathrm{Ad}_i^b \to \forall e. \left(\mathrm{R} \to [\mathrm{Ad}_i^b \sqcap \neg \mathrm{Ad}_i^{1-b}]\right)$,
where the concepts $\downarrow_{(\mathcal{D},\bar{c})}^{\oplus_\mathrm{H}}$ and $\downarrow_{(\mathcal{D},\bar{c})}^{\oplus_\mathrm{V}}$ implement binary addition modulo $2^{|\bar{c}|}$ [BHLS17, p. 127]:

$$\downarrow_{(\mathcal{D},\bar{c})}^{\oplus_\mathrm{H}} := \Big[\bigsqcap_{i=1}^{|\bar{c}|} \mathrm{Ad}_i^1 \sqcap \forall e.(\mathrm{M} \to \mathrm{Ad}_i^0)\Big] \sqcup$$
$$\sqcup \bigsqcup_{i=1}^{|\bar{c}|} \Big(\mathrm{Ad}_i^0 \sqcap \forall e.(\mathrm{M} \to \mathrm{Ad}_i^1) \sqcap \bigsqcap_{j=1}^{i-1} (\mathrm{Ad}_j^0 \leftrightarrow \forall e.(\mathrm{M} \to \mathrm{Ad}_j^0)) \sqcap \bigsqcap_{j=i+1}^{|\bar{c}|} (\mathrm{Ad}_j^1 \sqcap \forall e.(\mathrm{M} \to \mathrm{Ad}_j^0))\Big)$$

$$\downarrow_{(\mathcal{D},\bar{c})}^{\oplus_\mathrm{V}} := \Big[\bigsqcap_{i=|\bar{c}|+1}^{2|\bar{c}|} \mathrm{Ad}_i^1 \sqcap \forall e.(\mathrm{R} \to \mathrm{Ad}_i^0)\Big] \sqcup$$
$$\sqcup \bigsqcup_{i=|\bar{c}|+1}^{2|\bar{c}|} \Big(\mathrm{Ad}_i^0 \sqcap \forall e.(\mathrm{R} \to \mathrm{Ad}_i^1) \sqcap \bigsqcap_{j=|\bar{c}|+1}^{i-1} (\mathrm{Ad}_j^0 \leftrightarrow \forall e.(\mathrm{R} \to \mathrm{Ad}_j^0)) \sqcap \bigsqcap_{j=i+1}^{2|\bar{c}|} (\mathrm{Ad}_j^1 \sqcap \forall e.(\mathrm{M} \to \mathrm{Ad}_j^0))\Big)$$

(Adr[i]) $\mathrm{CorrAd}_{(\mathcal{D},\bar{c})}^i := \bigsqcap_{j=1}^{i} \bigsqcup_{b=0}^{1} \left([\mathrm{Ad}_j^b \sqcap \neg \mathrm{Ad}_j^{1-b}] \sqcap \forall e.[\mathrm{Ad}_j^b \sqcap \neg \mathrm{Ad}_j^{1-b}]\right) \sqcap \bigsqcap_{j=i+1}^{2|\bar{c}|} [\neg \mathrm{Ad}_j^0 \sqcap \neg \mathrm{Ad}_j^1]$.

6. The last family of concepts ensures that the "tile concepts" are appropriately assigned to the leaves of treepods, fulfilling the last three items of Definition 8.11. Below we assume that $\mathcal{D}$ is decomposed as $(\mathrm{T}, \mathrm{H}, \mathrm{V})$ and that $b_1^x \ldots b_{|\bar{c}|}^x$ is the binary representation of the number x on $|\bar{c}|$ bits.

(Tiles[i]) $\mathrm{CorrT}_{(\mathcal{D},\bar{c})}^i := \bigsqcap_{\mathrm{t} \in \mathrm{T}} \neg \mathrm{C_t}$

(Tiles[$2|\bar{c}|+1$]) $\mathrm{CorrT}_{(\mathcal{D},\bar{c})}^{2|\bar{c}|+1} := \bigsqcup_{\mathrm{t} \in \mathrm{T}} \left(\mathrm{C_t} \sqcap \bigsqcap_{\mathrm{t}' \in \mathrm{T} \setminus \{\mathrm{t}\}} \neg \mathrm{C_{t'}}\right) \sqcap \bigsqcap_{x=0}^{|\bar{c}|-1} \left(\bigsqcap_{i=1}^{|\bar{c}|} [\mathrm{Ad}_i^{b_i^x} \sqcap \mathrm{Ad}_{|\bar{c}|+i}^0]\right) \to \mathrm{C}_{\bar{c}_x}$

(Tiles[$2|\bar{c}|$]) $\mathrm{CorrT}_{(\mathcal{D},\bar{c})}^{2|\bar{c}|} := \bigsqcup_{\mathrm{t} \in \mathrm{T}} \left(\mathrm{C_t} \sqcap (\forall e.(\mathrm{L} \to \mathrm{C_t})) \sqcap \bigsqcap_{\mathrm{t}' \in \mathrm{T} \setminus \{\mathrm{t}\}} \neg \mathrm{C_{t'}}\right) \sqcap$
$\sqcap \bigsqcap_{(\mathrm{t},\mathrm{t}') \in \mathrm{T}^2 \setminus \mathrm{H}} \neg(\exists e.[\mathrm{L} \sqcap \mathrm{C_t}] \sqcap \exists e.[\mathrm{M} \sqcap \mathrm{C_{t'}}]) \sqcap \bigsqcap_{(\mathrm{t},\mathrm{t}') \in \mathrm{T}^2 \setminus \mathrm{V}} \neg(\exists e.[\mathrm{L} \sqcap \mathrm{C_t}] \sqcap \exists e.[\mathrm{R} \sqcap \mathrm{C_{t'}}])$

The following lemma distils the essential properties of our construction. Its proof is immediate.

Lemma 8.13 Let $\mathcal{D}$ be a torus tiling system, and $\bar{c}$ be an initial condition. Then for all $0 \leq i \leq 2|\bar{c}|+1$:

(I) The concept $\mathrm{Treepod}_{(\mathcal{D},\bar{c})}^i$ is of size polynomial w.r.t $(|\bar{c}|+|\mathcal{D}|)$.

(II) For all $(\mathcal{D},\bar{c})$-treepods $\mathcal{T}$ and all $\mathrm{d} \in \mathrm{Lvl}_i^{\mathcal{T}}$, we have $\mathrm{d} \in (\mathrm{Treepod}_{(\mathcal{D},\bar{c})}^i)^{\mathcal{T}}$.

(III) If $i \leq 2|\bar{c}|$, then for all pointed models $(\mathcal{I}, \mathrm{d})$ of $\mathrm{Treepod}_{(\mathcal{D},\bar{c})}^i$ with $\mathrm{d} \in (\mathrm{Ad}_1^{b_1})^{\mathcal{T}} \cap \ldots \cap (\mathrm{Ad}_i^{b_i})^{\mathcal{T}}$ for some binary word $w := b_1 \ldots b_i$ in $\{0,1\}^i$ there exists a $(\mathcal{D},\bar{c})$-treepod $\mathcal{T}$ and a homomorphism $\mathfrak{h}$, from $\mathcal{T}\restriction_{\{wu \mid u \in \{0,1,2\}^*\}}$ (*i.e.* the subtree of $\mathcal{T}$ rooted at w) *into* $\mathcal{I}$, satisfying $\mathfrak{h}(w) = \mathrm{d}$.

Proof sketch. The proof is by careful inspection of the semantics of the logic $\mathcal{ALC}$, our axiomatisation given above, the definition of $(\mathcal{D},\bar{c})$-treepods (Definition 8.11), basic properties of binary addition, and routine induction, where the inductive assumption is as stated above. □

8.3.3 Tentacles and Jellyfishes

For $\mathcal{D}$ and $\bar{c}$ as in the previous section, we introduce $(\mathcal{D},\bar{c})$-*tentacles*, whose sole purpose is to provide an alternative, query-friendly way of incorporating addresses and tiles for the leaves of $(\mathcal{D},\bar{c})$-treepods. We explain them with an example. Suppose d is a leaf of a $(\mathcal{D},\bar{c})$-treepod, decorated with the "address concepts" $\mathrm{Ad}_1^{b_1}, \ldots, \mathrm{Ad}_{2n}^{b_{2n}}$ (encoding the address of d) as well as the tile concept $\mathrm{C_t}$ (encoding the tile carried by d). We make our encoding redundant by enriching d with an outgoing $(ad_1^{b_1} \ldots ad_{2n}^{b_{2n}})$-path (where $ad_i^{b_i}$ are *role* names) leading to some element $\mathrm{d_t}$, representing the selected tile t in an usual way.

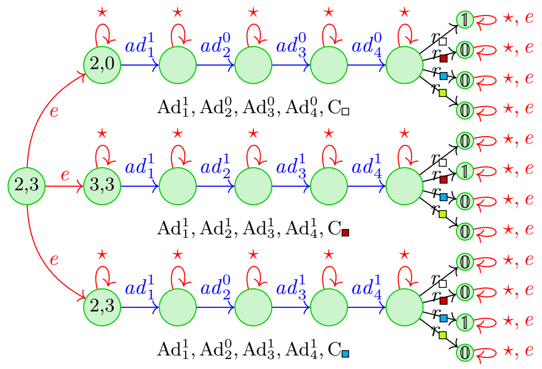

2,0
2,3
3,3
2,3
ad_1^1 ad_2^0 ad_3^0 ad_4^0
$\mathrm{Ad}_1^1, \mathrm{Ad}_2^0, \mathrm{Ad}_3^0, \mathrm{Ad}_4^0, \mathrm{C}_{□}$
ad_1^1 ad_2^1 ad_3^1 ad_4^1
$\mathrm{Ad}_1^1, \mathrm{Ad}_2^1, \mathrm{Ad}_3^1, \mathrm{Ad}_4^1, \mathrm{C}_{🟥}$
ad_1^1 ad_2^0 ad_3^1 ad_4^1
$\mathrm{Ad}_1^1, \mathrm{Ad}_2^0, \mathrm{Ad}_3^1, \mathrm{Ad}_4^1, \mathrm{C}_{🟦}$
e
$\star$
$\star, e$
$\mathbb{1}$
$\mathbb{0}$

Figure 8.4: A visualisation of three tentacles attached to the node representing the position $(2, 3)$ in treepods of height 4 (and assuming that all the tiles are among □, 🟦, 🟥, and 🟩).

Such an element $\mathrm{d_t}$ has pairwise-different $r_{\mathrm{t}'}$-successors for roles $r_{\mathrm{t}'}$ for tiles $\mathrm{t}' \in \mathrm{T}$, where the r_{t}-successor of $\mathrm{d_t}$ is labelled with $\mathbb{1}$, and all the other successors of $\mathrm{d_t}$ are labelled with $\mathbb{0}$.

For reasons that become clear only after introducing the query, all elements[5] from tentacles are decorated with all kinds of self-loops. Consult the above example to gain more intuitions.

Definition 8.14 Let $\mathcal{D} := (\mathrm{T}, \mathrm{H}, \mathrm{V})$ be a torus tiling system, $\bar{\mathrm{c}} := (\bar{\mathrm{c}}_0, \ldots, \bar{\mathrm{c}}_{n-1}) \in \mathrm{T}^n$ be an initial condition, $w := b_1 \ldots b_{2n}$ be a binary word encoding some position $(x, y) \in \mathbb{Z}_{2^n} \times \mathbb{Z}_{2^n}$ and $\mathrm{t} \in \mathrm{T}$ be a tile. A **$(\mathcal{D}, \bar{\mathrm{c}}, w, \mathrm{t})$-tentacle** $\mathcal{T} := (\Delta^{\mathcal{T}}, \cdot^{\mathcal{T}})$ is an interpretation fulfilling all the requirements below.

- $\Delta^{\mathcal{T}} = \{i \mid 2n{+}1 \leq i \leq 4n{+}1\} \cup (\{4n{+}2\} \times \mathrm{T})$,
- $(\mathrm{Lvl}_{4n+2})^{\mathcal{T}} = \{4n{+}2\} \times \mathrm{T}$ and $(\mathrm{Lvl}_i)^{\mathcal{T}} = \{i\}$ for all $2n{+}1 \leq i \leq 4n{+}1$,
- $(\mathrm{Ad}_i^{b_i})^{\mathcal{T}} = \Delta^{\mathcal{T}}$ and $(\mathrm{Ad}_i^{1-b_i})^{\mathcal{T}} = \emptyset$ for all $1 \leq i \leq 2n$,
- $(\mathrm{C_t})^{\mathcal{T}} = \{i, (4n{+}2, \mathrm{t}) \mid 2n{+}1 \leq i \leq 4n{+}1\}$ and $(\mathrm{C}_{\mathrm{t}'})^{\mathcal{T}} = \{(4n{+}2, \mathrm{t}')\}$ for all $\mathrm{t}' \in \mathrm{T} \backslash \{\mathrm{t}\}$
- $\mathbb{1}^{\mathcal{T}} = \{(4n{+}2, \mathrm{t})\}$ and $\mathbb{0}^{\mathcal{T}} = (\{4n{+}2\} \times \mathrm{T}) \setminus \mathbb{1}^{\mathcal{T}}$,
- $e^{\mathcal{T}} = \{(\mathrm{d}, \mathrm{d}) \mid \mathrm{d} \in (\{4n{+}2\} \times \mathrm{T})\}$,
- $(ad_i^{b_i})^{\mathcal{T}} = \{(2n{+}i, 2n{+}i{+}1)\} \cup \{(\mathrm{d}, \mathrm{d}) \mid \mathrm{d} \in \Delta^{\mathcal{T}}\}$ for all $1 \leq i \leq 2n$,
- $(ad_i^{1-b_i})^{\mathcal{T}} = \{(\mathrm{d}, \mathrm{d}) \mid \mathrm{d} \in \Delta^{\mathcal{T}}\}$ for all $\mathrm{t}' \in \mathrm{T}$ and all $1 \leq i \leq 2n$,
- $(r_{\mathrm{t}})^{\mathcal{T}} = \{(4n{+}1, (4n{+}2, \mathrm{t}))\} \cup \{(\mathrm{d}, \mathrm{d}) \mid \mathrm{d} \in \Delta^{\mathcal{T}}\}$ for all $\mathrm{t} \in \mathrm{T}$.

A **$(\mathcal{D}, \bar{\mathrm{c}})$-tentacle** is an $(\mathcal{D}, \bar{\mathrm{c}}, w, \mathrm{t})$-tentacle for some w and t.

We next provide an axiomatisation of $(\mathcal{D}, \bar{\mathrm{c}})$-tentacles, in total analogy to what we did in the previous section. We made our axiomatisation sufficiently formal by means of the forthcoming Lemma 8.15. For brevity, let $\mathsf{R} := \{r_{\mathrm{t}}, ad_i^b \mid \mathrm{t} \in \mathrm{T}, 1 \leq i \leq 2|\bar{\mathrm{c}}|, 0 \leq b \leq 1\}$ be the set of all roles used in our axiomatisation.

1. Concepts $\mathrm{Tent}^i_{(\mathcal{D},\bar{\mathrm{c}})}$, for $0 \leq i \leq 2|\bar{\mathrm{c}}|{+}1$, axiomatise substructures of $(\mathcal{D}, \bar{\mathrm{c}})$-tentacles restricted to elements having their first coordinate smaller or equal to $2|\bar{\mathrm{c}}|{+}i{+}1$.

   (Tent[i]) $\mathrm{Tent}^i_{(\mathcal{D},\bar{\mathrm{c}})} := \mathrm{UniqueAdr}^i_{(\mathcal{D},\bar{\mathrm{c}})} \sqcap \mathrm{Next}^i_{(\mathcal{D},\bar{\mathrm{c}})} \sqcap \mathrm{CorrLvlsLoops}^i_{(\mathcal{D},\bar{\mathrm{c}})} \sqcap \mathrm{CorrTil}^i_{(\mathcal{D},\bar{\mathrm{c}})}$.

2. The first element of a tentacle has a unique address. Addresses are propagated through successors.

   (UniqueAdr[i]) $\mathrm{UniqueAdr}^i_{(\mathcal{D},\bar{\mathrm{c}})} := \bigsqcap_{j=1}^{2|\bar{\mathrm{c}}|} \bigsqcup_{b=0}^{1} \Big([\mathrm{Ad}_j^b \sqcap \neg \mathrm{Ad}_j^{1-b}] \sqcap \bigsqcap_{r \in \mathsf{R}} \forall r.[\mathrm{Ad}_j^b \sqcap \neg \mathrm{Ad}_j^{1-b}] \Big)$.

3. Successors of elements are properly enforced.

   (Next[$2|\bar{\mathrm{c}}|{+}1$]) $\mathrm{Next}^{2|\bar{\mathrm{c}}|+1}_{(\mathcal{D},\bar{\mathrm{c}})} := \top$,

   (Next[$2|\bar{\mathrm{c}}|$]) $\mathrm{Next}^{2|\bar{\mathrm{c}}|}_{(\mathcal{D},\bar{\mathrm{c}})} := \bigsqcup_{\mathrm{t} \in \mathrm{T}} \exists r_{\mathrm{t}}.[\mathrm{C_t} \sqcap \mathrm{Lvl}_{4|\bar{\mathrm{c}}|+2} \sqcap \mathrm{Tent}^{4|\bar{\mathrm{c}}|+2}_{(\mathcal{D},\bar{\mathrm{c}})}]$,

[5] For the design of the query we only require self-loops on elements from the last level. Loops on the other elements are needed however to keep the resulting concept size polynomial.

(Next[i]) $\mathrm{Next}^{i}_{(\mathcal{D},\bar{c})} \coloneqq \bigsqcap_{b=0}^{1}[\mathrm{Ad}^{b}_{i} \rightarrow (\forall ad^{1-b}_{i}.\neg\mathrm{Lvl}_{2|\bar{c}|+1+(i+1)} \sqcap \exists ad^{b}_{i}.\mathrm{Lvl}_{2|\bar{c}|+1+(i+1)})] \sqcap$ $\sqcap \forall ad^{0}_{i}.\forall ad^{1}_{i}.(\mathrm{Lvl}_{2|\bar{c}|+1+(i+1)} \rightarrow \mathrm{Tent}^{i+1}_{(\mathcal{D},\bar{c})})$.[6]

4. Levels and self-loops are properly enforced.

   (CorrLvlsLoops[i]) $\mathrm{CorrLvlsLoops}^{i}_{(\mathcal{D},\bar{c})} \coloneqq \bigsqcup_{2|\bar{c}|+1 \leq j \leq |\bar{c}|+2} \left(\mathrm{Lvl}_{j} \sqcap \bigsqcap_{k \neq j} \neg\mathrm{Lvl}_{k}\right) \sqcap \bigsqcap_{r \in \mathsf{R}} \exists r.\mathsf{Self} \sqcap$ $\sqcap \mathrm{Lvl}_{4|\bar{c}|+2} \rightarrow \exists e.\mathsf{Self}$.

5. There is a unique tile carried by the first element of a tentacle, and such a tile is propageted further to the very last element and is indicated by the $\mathbb{1}$ concept.

   (CorrTil[$2|\bar{c}|+1$]) $\mathrm{CorrTil}^{2|\bar{c}|+1}_{(\mathcal{D},\bar{c})} \coloneqq \top$,

   (CorrTil[$2|\bar{c}|$]) $\mathrm{CorrTil}^{2|\bar{c}|}_{(\mathcal{D},\bar{c})} \coloneqq \bigsqcap_{\mathrm{t} \in \mathrm{T}} \left(\mathrm{C}_{\mathrm{t}} \rightarrow \forall r_{\mathrm{t}}.[\mathrm{Lvl}_{4|\bar{c}|+2} \rightarrow \mathbb{1}] \sqcap \bigsqcap_{\mathrm{t}' \in \mathrm{T}\setminus\{\mathrm{t}\}} \forall r_{\mathrm{t}'}.[\mathrm{Lvl}_{4|\bar{c}|+2} \rightarrow \mathbb{0}]\right)$,

   (CorrTil[i]) $\mathrm{CorrTil}^{i}_{(\mathcal{D},\bar{c})} \coloneqq \bigsqcup_{\mathrm{t} \in \mathrm{T}} \left(\mathrm{C}_{\mathrm{t}} \sqcap \bigsqcap_{\mathrm{t}' \in \mathrm{T}\setminus\{\mathrm{t}\}} \neg\mathrm{C}_{\mathrm{t}'} \sqcap \bigsqcap_{r \in \mathsf{R}} \forall r.\mathrm{C}_{\mathrm{t}}\right)$.

The following lemma distils the essential properties of our construction. Its proof is immediate.

Lemma 8.15 Let $\mathcal{D}$ be a torus tiling system, and $\bar{c}$ be an initial condition. Then for all $0 \leq i \leq 2|\bar{c}|+1$:

(I) The concept $\mathrm{Tent}^{i}_{(\mathcal{D},\bar{c})}$ is of size polynomial w.r.t $(|\bar{c}|+|\mathcal{D}|)$.

(II) For all $(\mathcal{D},\bar{c})$-tentacles $\mathcal{T}$ and all $\mathrm{d} \in \mathrm{Lvl}^{\mathcal{T}}_{2|\bar{c}|+1+i}$, we have $\mathrm{d} \in (\mathrm{Tent}^{i}_{(\mathcal{D},\bar{c})})^{\mathcal{T}}$.

(III) If $i \leq 2|\bar{c}|$, then for all pointed models $(\mathcal{I},\mathrm{d})$ of $\mathrm{Tent}^{i}_{(\mathcal{D},\bar{c})}$ with $\mathrm{d} \in (\mathrm{Ad}^{b_1}_{1})^{\mathcal{T}} \cap \ldots \cap (\mathrm{Ad}^{b_{2|\bar{c}|}}_{2|\bar{c}|})^{\mathcal{T}} \cap (\mathrm{C}_{\mathrm{t}})^{\mathcal{T}}$ for some binary word $w \coloneqq b_1 \ldots b_{2|\bar{c}|}$ in $\{0,1\}^{2|\bar{c}|}$ and a tile $\mathrm{t} \in \mathrm{T}$, there exists a $(\mathcal{D},\bar{c},w,\mathrm{t})$-tentacle $\mathcal{T}$ and a homomorphism $\mathfrak{h}$, from $\mathcal{T}{\restriction}_{\{j \mid 2|\bar{c}|+1+i \leq j \leq 4|\bar{c}|+1\} \cup (\{4|\bar{c}|+2\} \times \mathrm{T})}$ (*i.e.* the substructure of $\mathcal{T}$ "starting" from $2|\bar{c}|+1+i$) *into* $\mathcal{I}$, satisfying $\mathfrak{h}(2|\bar{c}|+1+i) = \mathrm{d}$.

Proof sketch. The proof is by careful inspection of the semantics of the logic $\mathcal{ALC}^{\mathsf{Self}}$, our axiomatisation given above, the definition of $(\mathcal{D},\bar{c})$-tentacle (Definition 8.14), and routine induction, in which the inductive assumption is as stated above. □

As the final step of our model construction we introduce $(\mathcal{D},\bar{c})$-*jellyfishes* (see Figure 8.3 for a visualisation). These are interpretations obtained by replacing the leaves in $(\mathcal{D},\bar{c})$-treepods by the corresponding $(\mathcal{D},\bar{c},w,\mathrm{t})$-tentacles. Formally:

Definition 8.16 Let $\mathcal{D} \coloneqq (\mathrm{T},\mathrm{H},\mathrm{V})$ be a torus tiling system and $\bar{c} \coloneqq (\bar{c}_0,\ldots,\bar{c}_{n-1}) \in \mathrm{T}^n$ be an initial condition. A **$(\mathcal{D},\bar{c})$-jellyfish** is an interpretation $\mathcal{Y} \coloneqq (\Delta^{\mathcal{Y}},\cdot^{\mathcal{Y}})$ satisfying:

- $\Delta^{\mathcal{Y}} = \Delta \cup \bigcup_{w \in \{0,1\}^{=2n}\cdot\{0,1,2\}} \Delta_w$, where
  - $\Delta \coloneqq \{0,1\}^{\leq 2n} \cup \{0,1\}^{=2n} \cdot \{0,1,2\}$,
  - $\Delta_w \coloneqq \{w,(w,i),(w,(4n+2,\mathrm{t})) \mid \mathrm{t} \in \mathrm{T}, 2n+2 \leq i \leq 4n+1\}$,
- $\mathcal{Y}$ restricted to Δ is a $(\mathcal{D},\bar{c})$-treepod,
- For every $w \in \{0,1\}^{=2n} \cdot \{0,1,2\}$ we have that $\mathcal{Y}$ restricted to Δ is a $(\mathcal{D},\bar{c})$-tentacle (modulo renaming domain elements via $w \mapsto 2n+1$, $(w,i) \mapsto i$ and $(w,(4n+2,\mathrm{t})) \mapsto (4n+2,\mathrm{t})$).

We say that $\mathcal{Y}$ is **proper** if the corresponding treepod ($\mathcal{Y}$ restricted to Δ is proper).

It is straightforward to show that any treepod can be extended to a jellyfish.

Fact 8.17. Let $(\mathcal{D},\bar{c})$ be as in Definition 8.16, and $\mathcal{I}$ be a $(\mathcal{D},\bar{c})$-treepod $\mathcal{T}$. Then there exists a $(\mathcal{D},\bar{c})$-jellyfish $\mathcal{Y}$ that contains $\mathcal{T}$ as an induced substructure. Moreover, $\mathcal{Y}$ is proper if and only if $\mathcal{T}$ is proper.

[6]It is more natural employ the concept $\forall ad^{0}_{i}.(\mathrm{Lvl}_{2|\bar{c}|+1+(i+1)} \rightarrow \mathrm{Tent}^{i+1}_{(\mathcal{D},\bar{c})}) \sqcap \forall ad^{1}_{i}.(\mathrm{Lvl}_{2|\bar{c}|+1+(i+1)} \rightarrow \mathrm{Tent}^{i+1}_{(\mathcal{D},\bar{c})})$ in place of our concept definition. This would however result in an exponential blowup, which we want to avoid.

Proof. It suffices replace every leaf from $\mathcal{T}$ that represents the address w and that carries a tile t with an $(\mathcal{D}, \bar{\mathrm{c}}, w, \mathrm{t})$-tentacle. $\square$

Take $\mathrm{Yelly}_{(\mathcal{D},\bar{\mathrm{c}})} \coloneqq \mathrm{Treepod}^0_{(\mathcal{D},\bar{\mathrm{c}})} \sqcap \forall e^{2|\bar{\mathrm{c}}|+1}.\mathrm{Tent}^0_{(\mathcal{D},\bar{\mathrm{c}})}$, where $\forall e^1.\mathrm{C} \coloneqq \forall e.\mathrm{C}$ and $\forall e^{i+1}.\mathrm{C} \coloneqq \forall e^i.(\forall e.\mathrm{C})$ for all positive integers i and concepts C. The following is a consequence of Lemma 8.13 and Lemma 8.15.

Lemma 8.18 Let $\mathcal{D}$ be a torus tiling system, and $\bar{\mathrm{c}}$ be an initial condition. Then:

(I) The concept $\mathrm{Yelly}_{(\mathcal{D},\bar{\mathrm{c}})}$ is of size polynomial w.r.t $(|\bar{\mathrm{c}}|+|\mathcal{D}|)$.

(II) For all $(\mathcal{D},\bar{\mathrm{c}})$-jellyfishes $\mathcal{Y}$ we have $\varepsilon \in (\mathrm{Yelly}_{(\mathcal{D},\bar{\mathrm{c}})})^{\mathcal{Y}}$.

(III) For all pointed models $(\mathcal{I}, \mathrm{d})$ of $\mathrm{Yelly}_{(\mathcal{D},\bar{\mathrm{c}})}$, there exists a $(\mathcal{D},\bar{\mathrm{c}})$-jellyfish $\mathcal{Y}$ and a homomorphism $\mathfrak{h}$ from $\mathcal{Y}$ *into* $\mathcal{I}$ satisfying $\mathfrak{h}(\varepsilon) = \mathrm{d}$.

Proof. It suffices to apply Lemma 8.13 and Lemma 8.15 for the case of $i = 0$. $\square$

Having the notion of jellyfishes ready, we employ Fact 8.17 (relating treepods and jellyfishes) in order to reformulate Fact 8.12 (relating proper treepods and solutions to the tiling problem) in the language of proper jellyfishes. The following fact is vital for the design of the forthcoming query.

Fact 8.19. Let $\mathcal{D} \coloneqq (\mathrm{T}, \mathrm{H}, \mathrm{V})$ and $\bar{\mathrm{c}} \in \mathrm{T}^n$ be an instance of the torus tiling problem. A proper $(\mathcal{D},\bar{\mathrm{c}})$-jellyfish $\mathcal{Y}$ exists if and only if $(\mathcal{D},\bar{\mathrm{c}})$ has a solution. Moreover, a jellyfish $\mathcal{Y}$ is *not* proper if and only if there exists elements $\mathrm{d}, \mathrm{e} \in \Delta^{\mathcal{Y}}$ satisfying all the criteria below.

- The elements d and e are leaves of $\mathcal{Y}$ (namely they belong to $\mathrm{Lvl}^{\mathcal{Y}}_{4n+2}$) satisfying $\mathrm{d} \in \mathbb{0}^{\mathcal{Y}}$ and $\mathrm{e} \in \mathbb{1}^{\mathcal{Y}}$.
- The elements d and e represent the same positions of a torus, namely for all $1 \leq i \leq 2n$ the equivalence $\mathrm{d} \in (\mathrm{Ad}^0_i)^{\mathcal{Y}}$ if and only if $\mathrm{e} \in (\mathrm{Ad}^0_i)^{\mathcal{Y}}$ holds. Note that we can alternatively say that for all $1 \leq i \leq 2n$, d has an ancestor d' that has a child d'' satisfying $(\mathrm{d}', \mathrm{d}'') \in (ad^0_i)^{\mathcal{J}}$ if and only if there exists an ancestor e' of e that has a child e'' satisfying $(\mathrm{e}', \mathrm{e}'') \in (ad^0_i)^{\mathcal{J}}$.
- For all tiles $\mathrm{t} \in \mathrm{T}$ the equivalence $\mathrm{d} \in (\mathrm{C_t})^{\mathcal{Y}}$ if and only if $\mathrm{e} \in (\mathrm{C_t})^{\mathcal{Y}}$ holds. Alternatively, for all tiles $\mathrm{t} \in \mathrm{T}$ we can say that $(\mathrm{d}', \mathrm{d}) \in (r_{\mathrm{t}})^{\mathcal{J}}$ if and only if $(\mathrm{e}', \mathrm{e}) \in (r_{\mathrm{t}})^{\mathcal{J}}$, where d' and e' are parents of d and e.

Proof. The existence of a proper $(\mathcal{D},\bar{\mathrm{c}})$-jellyfish is equivalent to the existence of a proper treepod (b y Fact 8.17), which is in turn equivalent to the existence of a solution to $(\mathcal{D},\bar{\mathrm{c}})$ (by Fact 8.12). If a $(\mathcal{D},\bar{\mathrm{c}})$-jellyfish $\mathcal{Y}$ is not proper, then by definition (see Definition 8.16), its underlying treepod $\mathcal{T}$ is not proper. Then, by Definition 8.11, there exists a position $(x, y) \in \mathbb{Z}_{2^n} \times \mathbb{Z}_{2^n}$, tiles $\mathrm{t}, \mathrm{t}' \in \mathrm{T}$ (where $n \coloneqq |\bar{\mathrm{c}}|$ and T is a set of tiles from $\mathcal{D}$), and leaves w, w' of $\mathcal{T}$ encoding (x, y) such that $w \in \mathrm{C}^{\mathcal{T}}_{\mathrm{t}}$, $w' \in \mathrm{C}^{\mathcal{T}}_{\mathrm{t}'}$. Let $b_1, \ldots, b_{2n}$ be the binary encoding of (x, y) on $2n$ bits. Then we have w and w' both belong to $\bigcap_{i=1}^{2n} (\mathrm{Ad}^{b_i}_i)^{\mathcal{T}}$. By the construction of jellyfishes, we see that w and w' have an outgoing $ad^{b_1}_1 \ldots ad^{b_{2n}}_1$ paths, leading, respectively, to the elements d' and e'. Note that all the elements on this path are labelled concepts $\mathrm{Ad}^{b_1}_1, \ldots, \mathrm{Ad}^{b_{2n}}_1$. In the case of w (resp. w') such elements are additionally labelled with $\mathrm{C_t}$ (resp. $\mathrm{C_{t'}}$). Let d (resp. e) be the $r_{\mathrm{t}'}$-successor of d' (resp. e'). By the previous observation, we infer $\mathrm{d} \in \mathbb{0}^{\mathcal{Y}}$ and $\mathrm{e} \in \mathbb{1}^{\mathcal{Y}}$. Hence, d and e are fulfil of the requirements of Fact 8.19. $\square$

8.3.4 Towards queries detecting mismatches: the top-down query

Henceforth we introduce an auxiliary query called the *top-down query*, which intended role is to traverse the jellyfishes in a top-down manner, and to simulate disjunction of paths in a rather intricate sense. We provide a characterisation of matches of the top-down query over jellyfishes as well as discuss its modifications.

A bit of extra notation. First of all, recall that for convenience and brevity we employ a path syntax for path queries. Second, given a path query q, we write $q(x, y)$ to denote the query obtained by replacing the variable with the lowest index by x, and the variable with the highest index with y. Whenever we write the conjunction $q \wedge q'$ of two path queries, we assume that they have disjoint sets of variables. This holds unless indicated otherwise with the bracket notation, *e.g.* in the case of $q(x,y) \wedge q(y,z)$ we assume that q and q' share a common variable y. Third, for a set of role names $\mathsf{R} \subseteq \mathbf{N_R}$ and a path query q, we write $q[\![\mathsf{R}]\!]$ for the query obtained by removing (or alternatively, replacing with $\top?$) all roles present in R. For convenience, brackets are removed if R is a singleton set. For instance, by $(\mathrm{A}?; r_1; \mathrm{B}?; r_2; \mathrm{C}?; r_3)[\![r_1]\!]$ we mean the query $\mathrm{A}?; \mathrm{B}?; r_2; \mathrm{C}?; r_3$, and by $(\mathrm{A}?; r_1; \mathrm{B}?; r_2; \mathrm{C}?; r_3)[\![\{r_1, r_3\}]\!]$ we mean the query $\mathrm{A}?; \mathrm{B}?; r_2; \mathrm{C}?$.

As the most important ingredient of the forthcoming query, we introduce the *top down query*.

Definition 8.20 Let $\mathcal{D} := (\mathrm{T}, \mathrm{H}, \mathrm{V})$ be a torus tiling system and $\bar{\mathrm{c}} := (\bar{\mathrm{c}}_0, \ldots, \bar{\mathrm{c}}_{n-1}) \in \mathrm{T}^n$ be an initial condition. For convenience, we enumerate T as $\mathrm{t}_1, \ldots, \mathrm{t}_{|\mathrm{T}|}$. We define the **top-down query $q_{\downarrow}$ for $(\mathcal{D}, \bar{\mathrm{c}})$** as the following path conjunctive query:

$$q_{\downarrow} := e^{2n+1}; ad_1^0; ad_1^1; \ldots; ad_{2n}^0; ad_{2n}^1; r_{\mathrm{t}_1}; r_{\mathrm{t}_2}; \ldots; r_{\mathrm{t}_{|\mathrm{T}|}}; \mathrm{Lvl}_{4n+2}?$$

In the forthcoming constructions we consider prefixes and suffixes of $q_{\downarrow}$. For brevity, let $s_0 := e^{2n+1}$, $s_i := ad_i^0; ad_i^1$ for all $1 \le i \le 2n$, and $s_{2n+1} := r_{\mathrm{t}_1}; r_{\mathrm{t}_2}; \ldots; r_{\mathrm{t}_{|\mathrm{T}|}}$. Thus the query $q_{\downarrow}$ can be rewritten as $s_0; s_1; \ldots; s_{2n+1}; \mathrm{Lvl}_{2n+1}?$. For all $0 \le i \le 2n{+}1$ we define:

$$q_{\downarrow}^{i+} := s_0; \ldots; s_i \qquad q_{\downarrow}^{i-} := s_i; s_{i+1}; \ldots; s_{2n+1}; \mathrm{Lvl}_{4n+2}?$$

Note that the queries $q_{\downarrow}^{0-}$, $q_{\downarrow}^{(2n+1)+}\mathrm{Lvl}_{4n+2}?$, and $q_{\downarrow}$ are equal.

Note that the size of $q_{\downarrow}$ is clearly polynomial w.r.t. $|\mathcal{D}|{+}|\bar{\mathrm{c}}|$. The intention behind the query $q_{\downarrow}$ is that it either traverses the jellyfishes in a top-down manner, or it loops at one of its leaves. Consult also Example 8.9 for a more concrete example. Such intuitions meet their formalisation in the following (quite technical) lemmas, established via a routine induction.

Lemma 8.21 Let $\mathcal{D}$, $\bar{\mathrm{c}}$, n, and $q_{\downarrow}$ be as in Definition 8.20, and let $\mathcal{Y}$ be a $(\mathcal{D}, \bar{\mathrm{c}})$-jellyfish. For all indices $0 \le i \le 2n{+}1$, the set $\mathrm{M}_i^+ := \{(\eta(x), \eta(y)) \mid \mathcal{Y} \models_\eta q_{\downarrow}^{i+}(x,y)\}$ is equal to

$$\{(\mathrm{d}, \mathrm{d}) \mid \mathrm{d} \in \mathrm{Lvl}_{4n+2}^{\mathcal{Y}}\} \cup \{(\varepsilon, \mathrm{e}) \mid \mathrm{e} \in \textstyle\bigcup_{k=0}^{i} \mathrm{Lvl}_{2n+k+1}^{\mathcal{Y}}\}.$$

Proof. The proof is by induction, where the inductive assumption is given in the statement of the lemma. We start from the base case of $i = 0$. Observe that $(\heartsuit)$: "the only elements of $\mathcal{Y}$ that have an outgoing e^{2n+1}-path are the leaves of $\mathcal{Y}$ and the root of $\mathcal{Y}$". Indeed, this follows from the definition of jellyfishes based on the following facts: (i) the leaves of $\mathcal{Y}$ are equipped with an e-self-loop, (ii) no elements in $\mathrm{Lvl}_j^{\mathcal{Y}}$ for $2n{+}1 \le j \le 4n{+}1$ have e-successors, (iii) (by induction) the longest outgoing e^*-path from the elements d from $\mathrm{Lvl}_j^{\mathcal{Y}}$ for $0 \le j \le 2n$ has length $2n{+}1{-}j$ and the elements e^{2n+1-j} reachable from d belong to $\mathrm{Lvl}_{2n+1}^{\mathcal{Y}}$. We prove that:

- $\mathrm{M}_0^+ \subseteq \{(\mathrm{d}, \mathrm{d}) \mid \mathrm{d} \in \mathrm{Lvl}_{4n+2}^{\mathcal{Y}}\} \cup \{(\varepsilon, \mathrm{e}) \mid \mathrm{e} \in \mathrm{Lvl}_{2n+1}^{\mathcal{Y}}\}$.
  By $(\heartsuit)$ we have that $\eta(x)$ is either the root of $\mathcal{Y}$ or a leaf of $\mathcal{Y}$. In the first case, $\eta(y)$ belongs to Lvl_{2n+1} by item (iii) of $(\heartsuit)$. Otherwise, as all the outgoing roles from leaves are self-loops, we conclude that $\eta(x) = \eta(y)$.
- $\mathrm{M}_0^+ \supseteq \{(\mathrm{d}, \mathrm{d}) \mid \mathrm{d} \in \mathrm{Lvl}_{4n+2}^{\mathcal{Y}}\} \cup \{(\varepsilon, \mathrm{e}) \mid \mathrm{e} \in \mathrm{Lvl}_{2n+1}^{\mathcal{Y}}\}$.
  Take (d, e) from the RHS of the above expression, and it suffices to construct a match η for $q_{\downarrow}^{i+}(x,y)$ that maps x to d and y to e. If $\mathrm{d} = \mathrm{e}$ then the function η that maps all the variables from $q_{\downarrow}^{i+}$ to d is the desired match. Otherwise, the function η that maps each variable x_i to the i-th element prefix of e is the desired match.

Suppose that $i > 0$. We consider the query $q_{\downarrow}^{i+}(x,y)$ and present it as $q_{\downarrow}^{(i-1)+}(x,u) \wedge s_i(u,y)$. We establish the following two inclusions:

- $\mathrm{M}_i^+ \subseteq \{(\mathrm{d}, \mathrm{d}) \mid \mathrm{d} \in \mathrm{Lvl}_{4n+2}^{\mathcal{Y}}\} \cup \{(\varepsilon, \mathrm{e}) \mid \mathrm{e} \in \mathrm{Lvl}_{2n+i+1}^{\mathcal{Y}}\}$.
  Based on the inductive assumption on M_{i-1}^+ we have two possible options: either $\eta(x)$ is a leaf of $\mathcal{Y}$ or $\eta(x) = \varepsilon$. In the first case, by the inductive assumption, we get $\eta(u) = \eta(x)$.

Observe that the only r-successor of $\eta(u)$, for all role names r that appear in s_i, is $\eta(u)$ itself. Thus $\eta(u) = \eta(y)$, implying $\eta(u) = \eta(y)$. Otherwise, $\eta(u)$ belongs to $\bigcup_{k=0}^{i-1} \mathrm{Lvl}^{\mathcal{Y}}_{2n+k+1}$. Depending on whether $\eta(u)$ belongs to $\mathrm{Lvl}^{\mathcal{Y}}_{2n+i}$ or not we have two cases. If $\eta(u) \notin \mathrm{Lvl}^{\mathcal{Y}}_{2n+i}$ then the only r-successor of $\eta(u)$ for role names r present in s_i is $\eta(u)$ itself. Thus all the variables from $s_i(u, y)$ are mapped to $\eta(u)$, yielding $\eta(y) \in \bigcup_{k=0}^{i-1} \mathrm{Lvl}^{\mathcal{Y}}_{2n+i+1}$ as desired. Otherwise, $\eta(u) \notin \mathrm{Lvl}^{\mathcal{Y}}_{2n+i}$. If $\eta(y) = \eta(u)$ then we are done, so suppose that the opposite. Then we can present the query $s_i(u, y)$ as $\alpha(u, v) \wedge r(v, z) \wedge \beta(z, y)$ for some path subqueries α and β of s_i and a role name r present in s_i, such that the match η maps all the variables from α to $\eta(u)$ (it could be that α is empty) and $\eta(v) \neq \eta(z)$. By analysing the definition of jellyfishes we see that $\eta(z)$ is a child of $\eta(v)$ (and $\eta(u)$). More formally, $\eta(z) \in \mathrm{Lvl}^{\mathcal{Y}}_{2n+i+1}$. Moreover, all role names from β appear only as self-loops on elements from $\mathrm{Lvl}^{\mathcal{Y}}_{2n+i+1}$. This implies that all the variables from β are mapped via η to the same element, in particular $\eta(z) = \eta(y)$. Thus, $\eta(y)$ belongs to $\mathrm{Lvl}^{\mathcal{Y}}_{2n+i+1}$ (and hence to the "big sum").

- $\mathrm{M}_i^{+} \supseteq \{(\mathrm{d}, \mathrm{d}) \mid \mathrm{d} \in \mathrm{Lvl}^{\mathcal{Y}}_{4n+2}\} \cup \{(\varepsilon, \mathrm{e}) \mid \mathrm{e} \in \mathrm{Lvl}^{\mathcal{Y}}_{2n+i+1}\}$.
  Similarly to the base case, we take (d, e) from the RHS of the above expression and construct a match η for $q_{\downarrow}^{i+}(x, y)$ that maps x to d and y to e. If $\mathrm{d} = \mathrm{e}$ then the function η that maps all the variables from $q_{\downarrow}^{i+}$ to d is the desired match. Otherwise, we consider two cases depending on whether e belongs to Lvl_{2n+i+1} or not. If it does not, we invoke the inductive assumption to derive match η for $\mathcal{Y}$ and $q_{\downarrow}^{(i-1)+}(x, u)$ that maps x to the root of $\mathcal{Y}$ and u to e. By extending η to map all the variables from $s_i(u, y)$ to e we get the desired match. If e belongs to Lvl_{2n+i+1}, let c be its parent. Similarly to the previous case, we invoke the inductive assumption to derive match η for $\mathcal{Y}$ and $q_{\downarrow}^{(i-1)+}(x, u)$ that maps x to the root of $\mathcal{Y}$ and u to c. We can present $s_i(u, y)$ as $\alpha(u, v) \wedge r(v, z) \wedge \beta(z, y)$, where r is the unique role name from s_i for which e is the r-successor of c. Next, we extend η in a way that it maps all the variables from α to c and all the variables from β to e. By the presence of self-loops on c and e, the constructed map is the desired match.

This concludes the induction, and hence completes the proof. □

We next discuss the "suffix" subquery of the top-down query.

Lemma 8.22 Let $\mathcal{D}$, $\bar{\mathrm{c}}$, n, and $q_{\downarrow}$ be as in Definition 8.20, and let $\mathcal{Y}$ be a $(\mathcal{D}, \bar{\mathrm{c}})$-jellyfish. For all indices $0 \leq i \leq 2n{+}1$ we have that:

$$\mathrm{M}_0^{-} := \{(\eta(x), \eta(y)) \mid \mathcal{Y} \models_{\eta} q_{\downarrow}^{0-}(x, y)\} \text{ equals } \{(\mathrm{d}, \mathrm{d}), (\varepsilon, \mathrm{d}) \mid \mathrm{d} \in \mathrm{Lvl}^{\mathcal{Y}}_{4n+2}\},$$
$$\mathrm{M}_i^{-} := \{(\eta(x), \eta(y)) \mid \mathcal{Y} \models_{\eta} q_{\downarrow}^{i-}(x, y)\} \text{ equals } \left\{(\mathrm{d}, \mathrm{d}{\cdot}w) \in \left(\bigcup_{j=2n+i}^{4n+2} \mathrm{Lvl}_j\right) \times \mathrm{Lvl}_{4n+2}\right\}.$$

Proof sketch. For the equality concerning the set M_0^{-} we simply use the fact that the queries $q_{\downarrow}^{0-}$ and $q_{\downarrow}^{(2n+1)+}\mathrm{Lvl}_{4n+2}?$ are equal. Then the desired equality follows from Lemma 8.21. The remaining part of the proof goes via a routine induction, where the inductive assumption is stated above. We essentially repeat the second part of the proof of Lemma 8.21. □

We conclude the section by discussing a modification of the top-down queries, in which one of the subqueries s_i is reduced to a single role name. We define such queries below, and then prove their correctness.

Definition 8.23 Let $\mathcal{D}$, T, $\bar{\mathrm{c}}$, n, and $q_{\downarrow}$ be as in Definition 8.20. Let i be between 1 and $2n$, j be 0 or 1, $\mathrm{t} \in \mathrm{T}$ be a tile, and $\mathsf{R}_{\mathrm{t}} := \{r_{\mathrm{t}'} \mid \mathrm{t}' \in \mathrm{T}, \mathrm{t} \neq \mathrm{t}'\}$. By the **$ad_i^j$-refined top-down query** we mean the query $q_{ad_i^j} := q_{\downarrow}[\![ad_i^{1-j}]\!]$, and by the **$r_{\mathrm{t}}$-refined top-down query** we mean the query $q_{r_{\mathrm{t}}} := q_{\downarrow}[\![\mathsf{R}_{\mathrm{t}}]\!]$.

The main property of the refined top-down queries is established below.

Lemma 8.24 For all the parameters as in Definition 8.23, and a $(\mathcal{D}, \bar{\mathrm{c}})$-jellyfish $\mathcal{Y}$ we have that:

$$\mathrm{M}_{ad_i^j} := \{(\eta(x), \eta(y)) \mid \mathcal{Y} \models_\eta q_{ad_i^j}(x, y)\} \text{ is equal to } \{(\mathrm{d}, \mathrm{d}), (\varepsilon, \mathrm{e}) \mid \mathrm{d}, \mathrm{e} \in \mathrm{Lvl}_{4n+2}^{\mathcal{Y}}, \mathrm{e} \in (\mathrm{Ad}_i^j)^{\mathcal{Y}}\},$$
$$\mathrm{M}_{r_\mathrm{t}} := \{(\eta(x), \eta(y)) \mid \mathcal{Y} \models_\eta q_{r_\mathrm{t}}(x, y)\} \text{ is equal to } \{(\mathrm{d}, \mathrm{d}), (\varepsilon, \mathrm{e}) \mid \mathrm{d}, \mathrm{e} \in \mathrm{Lvl}_{4n+2}^{\mathcal{Y}}, \mathrm{e} \in (\mathrm{C}_\mathrm{t})^{\mathcal{Y}}\}.$$

Proof. As the two proofs of equality of the above sets are analogous, we focus only on the case of the query $q_{ad_i^j}$. We present $q_{ad_i^j}(x, y)$ as the query $q_{\downarrow}^{(i-1)+}(x, u) \wedge ad_i^j(u, v) \wedge q_{\downarrow}^{(i+1)-}(v, y)$. Similarly to the previous proof, we establish two inclusions.

- $\mathrm{M}_{ad_i^j} \subseteq \{(\mathrm{d}, \mathrm{d}), (\varepsilon, \mathrm{e}) \mid \mathrm{d}, \mathrm{e} \in \mathrm{Lvl}_{4n+2}^{\mathcal{Y}}, \mathrm{e} \in (\mathrm{Ad}_i^j)^{\mathcal{Y}}\}$.
  Consider a match η for $\mathcal{Y}$ and $q_{ad_i^j}$. By the property ($\heartsuit$) from the proof of Lemma 8.21 there are only two possible options: either $\eta(x)$ is a leaf of $\mathcal{Y}$ or $\eta(x)$ is the root of $\mathcal{Y}$. In the first case, namely if $\eta(x)$ is a leaf of $\mathcal{Y}$, we again employ the fact that the only r-successor of $\eta(x)$, for role names appearing in the query, is $\eta(x)$ itself. This implies that $\eta(x) = \eta(y)$ as desired. Otherwise, we rely on the previously established lemmas. With Lemma 8.21 we infer that $\eta(u) \in \bigcup_{k=0}^{i-1} \mathrm{Lvl}_{2n+k+1}^{\mathcal{Y}}$. With Lemma 8.22 we infer that $\eta(v) \in \bigcup_{j=2n+i+1}^{4n+2} \mathrm{Lvl}_j^{\mathcal{Y}}$ and that $\eta(y)$ is a descendant of $\eta(v)$. Thus, the only possibility for η to be a match is when $\eta(u) \in \mathrm{Lvl}_{2n+i}^{\mathcal{Y}}$, and $\eta(v) \in \mathrm{Lvl}_{2n+i+1}^{\mathcal{Y}}$. Due to the presence of the atom $ad_i^j(u, v)$ in the query, and the fact that $\eta(u)$ is a parent of $\eta(v)$ in $\mathcal{Y}$, we conclude that $\eta(v) \in (\mathrm{Ad}_i^j)^{\mathcal{Y}}$. By the design of $\mathcal{Y}$, we thus have that all the descendants of $\eta(v)$ belong to $(\mathrm{Ad}_i^j)^{\mathcal{Y}}$. In particular, this implies that $\eta(y) \in (\mathrm{Ad}_i^j)^{\mathcal{Y}}$, as desired.
- $\mathrm{M}_{ad_i^j} \supseteq \{(\mathrm{d}, \mathrm{d}), (\varepsilon, \mathrm{e}) \mid \mathrm{d}, \mathrm{e} \in \mathrm{Lvl}_{4n+2}^{\mathcal{Y}}, \mathrm{e} \in (\mathrm{Ad}_i^j)^{\mathcal{Y}}\}$.
  Similarly to the previous proofs of this section, we take (d, e) from the RHS of the above expression and construct a match η for $q_{ad_i^j}(x, y)$ that maps x to d and y to e. If $\mathrm{d} = \mathrm{e}$ then the function η that maps all the variables from the query $q_{ad_i^j}(x, y)$ to d is the desired match. Otherwise, we employ the results from Lemma 8.21. Take c_2 to be the ancestor of e satisfying Lvl_{2n+i+1}, and let c_1 be its parent. By the fact that $\mathrm{e} \in (\mathrm{Ad}_i^j)^{\mathcal{Y}}$, we know that $(\mathrm{c}_1, \mathrm{c}_2) \in (ad_i^j)^{\mathcal{Y}}$. By Lemma 8.21 we infer a match η_1 for $\mathcal{Y}$ and $q_{\downarrow}^{(i-1)+}(x, u)$ such that $\eta_1(x) = \varepsilon$ and $\eta_1(u) = \mathrm{c}_1$. Similarly, there exists a match η_2 for $\mathcal{Y}$ and $q_{\downarrow}^{(i+1)-}(v, y)$ such that $\eta_2(v) = \mathrm{c}_2$ and $\eta_2(y) = \mathrm{e}$. Hence, by combining η_1 and η_2 we obtain the desired match for $\mathcal{Y}$ and $q_{\downarrow}^{(i-1)+}(x, u) \wedge ad_i^j(u, v) \wedge q_{\downarrow}^{(i+1)-}(v, y)$.

□

8.3.5 The conjunctive query detecting mismatches

We conclude the chapter by designing a conjunctive query with distinguished variables x, y whose matches over jellyfishes $\mathcal{Y}$ identify the elements d and e witnessing the non-properness of $\mathcal{Y}$ (as explained by Fact 8.19). As the first step, by employing the top-down query, we easily design the *leaf-root-leaf* query $q_{\mathrm{lrl}}(x, y, z) := q_{\downarrow}(y, x) \wedge \mathrm{Lvl}_0(y) \wedge q_{\downarrow}(y, z)$ with three distinguished variables x, y, z that matches all possible leaf-root-leaf triples.

Corollary 8.25

Let $\mathcal{D}$, $\bar{\mathrm{c}}$, n, and $q_{\downarrow}$ be as in Definition 8.20, and let $\mathcal{Y}$ be a $(\mathcal{D}, \bar{\mathrm{c}})$-jellyfish. Then, for the leaf-root-leaf query $q_{\mathrm{lrl}}(x, y, z) := q_{\downarrow}(y, x) \wedge \mathrm{Lvl}_0(y) \wedge q_{\downarrow}(y, z)$, we have the equality:

$$\mathrm{M} := \{(\eta(x), \eta(y), \eta(z)) \mid \mathcal{Y} \models_\eta q_{\mathrm{lrl}}(x, y, z)\} \text{ is equal to } \{(\mathrm{d}, \varepsilon, \mathrm{e}) \mid \mathrm{d}, \mathrm{e} \in \mathrm{Lvl}_{4n+2}^{\mathcal{Y}}\}.$$

Proof. If η is a match for $q_{\mathrm{lrl}}(x, y, z)$ and $\mathcal{Y}$, then clearly $\eta(y) = \varepsilon$ due to the presence of the atom $\mathrm{Lvl}_0(y)$ in the query. Now it suffices to apply twice the definition of M_0^- from Lemma 8.21. □

Observe that in any match η of the $q_{\mathrm{lrl}}(x,y,z) \wedge \mathbb{0}(x) \wedge \mathbb{1}(z)$ over jellyfishes $\mathcal{Y}$, the elements $\eta(x)$ and $\eta(z)$ satisfy the first condition of Fact 8.19. Hence, it remains to design queries expressing the second and the third condition of Fact 8.19 and then append them to our query.

Definition 8.26 Let $\mathcal{D}$, $\bar{\mathrm{c}}$, n, and $q_{\downarrow}$ be as in Definition 8.20. For all $1 \leq i \leq 2n$ we define the ***i*-th bit equal query** $q^i_{\mathrm{eq}}(x,z) \coloneqq q_{ad^0_i}(y,x) \wedge q_{ad^0_i}(y,u) \wedge q_{ad^1_i}(y',u) \wedge q_{ad^1_i}(y',z)$.

We already provided intuitions regarding the 1-th bit query for the case of $n=1$ in Example 8.10. Here the idea is exactly the same; the only difference is the total of number roles involved in $q_{\downarrow}$. The idea is that $q^i_{\mathrm{eq}}(x,z)$ relates leaves of jellyfishes $\mathcal{Y}$ that agree on the i-th bit of their address, namely in any match η we either have that both $\eta(x)$ and $\eta(z)$ satisfy the concept Ad^0_i, or both of them satisfy the concept Ad^1_i.

Lemma 8.27 Let $\mathcal{D}$, $\bar{\mathrm{c}}$, and q^i_{eq} be as in Definition 8.26, and $\mathcal{Y}$ be a $(\mathcal{D},\bar{\mathrm{c}})$-jellyfish. Then the set $\mathrm{M}^i_{\mathrm{eq}} \coloneqq \{(\eta(x),\eta(z)) \mid \mathcal{Y} \models_\eta q^i_{\mathrm{eq}}(x,z)\}$ is equal to $\bigcup_{j=0}^{1}((\mathrm{Ad}^j_i)^{\mathcal{Y}} \cap \mathrm{Lvl}^{\mathcal{Y}}_{4n+2}) \times ((\mathrm{Ad}^j_i)^{\mathcal{Y}} \cap \mathrm{Lvl}^{\mathcal{Y}}_{4n+2})$.

Proof. We first establish that $\bigcup_{j=0}^{1}((\mathrm{Ad}^j_i)^{\mathcal{Y}} \cap \mathrm{Lvl}^{\mathcal{Y}}_{4n+2}) \times ((\mathrm{Ad}^j_i)^{\mathcal{Y}} \cap \mathrm{Lvl}^{\mathcal{Y}}_{4n+2})$ is a subset of $\mathrm{M}^i_{\mathrm{eq}}$. Take any (d,e) from LHS, and assume w.l.o.g. that d and e belong to $(\mathrm{Ad}^0_i)^{\mathcal{Y}} \cap \mathrm{Lvl}^{\mathcal{Y}}_{4n+2}$. We are now going to construct the intended match η. By Lemma 8.24 there is a match η_1 for $q_{ad^0_i}(y,x)$ that maps y to the root of $\mathcal{Y}$ and x to d. Moreover, there exists a η_2 for $q_{ad^0_i}(y,u)$ that maps y to the root of $\mathcal{Y}$ and u to e. We let η map all the variables from $q_{ad^0_i}(y,u)$ to their image via η_1, all the variables from $q_{ad^1_i}(y',u)$ to their image via η_2, all the remaining variables to e (note that a function that maps all the variables to the same leaf of $\mathcal{Y}$ is a match for $\mathcal{Y}$ and $q_{ad^1_i}$ by Lemma 8.24). Thus η is as desired.

To establish the reverse inclusion, consider a match η for q^i_{eq} and $\mathcal{Y}$. Similarly to what was described in Example 8.10 we see that $\eta(x)$, $\eta(z)$, and $\eta(u)$ are leaves of $\mathcal{Y}$, while $\eta(y)$ and $\eta(y')$ are either the root of $\mathcal{Y}$ or a leaf of $\mathcal{Y}$ (as they have outgoing e-paths of length $2n{-}1$). Consider the following cases:

- $\eta(y)$ is the root of $\mathcal{Y}$.
  Then, by Lemma 8.24 we have that both $\eta(x)$ and $\eta(u)$ satisfy Ad^0_i. Thus, $\eta(y')$ cannot be the root of $\mathcal{Y}$ as it would imply, by Lemma 8.24 again, that $\eta(u)$ satisfies Ad^1_i. Hence, $\eta(y')$ is a leaf of $\mathcal{Y}$, which once more by Lemma 8.24 implies that $\eta(y) = \eta(u)$ and $\eta(y') = \eta(z)$. This implies the equality $\eta(z) = \eta(u)$, and thus $\eta(z)$ satisfies the concept Ad^0_i as well.
- $\eta(y')$ is the root of $\mathcal{Y}$.
  The proof is then symmetric to the previous case (modulo swapping the numbers 0 and 1 in concepts). We arrive at the conclusion that $\eta(x)$ and $\eta(z)$ both satisfy the concept Ad^1_i.
- Both $\eta(y)$ and $\eta(y')$ are leaves of $\mathcal{Y}$.
  Then by Lemma 8.24 we have that all the distinguished variables are mapped to the same leaf of $\mathcal{Y}$.

Thus, in all such cases we established that $(\eta(x),\eta(z))$ belongs to either to the set $((\mathrm{Ad}^0_i)^{\mathcal{Y}} \cap \mathrm{Lvl}^{\mathcal{Y}}_{4n+2}) \times ((\mathrm{Ad}^0_i)^{\mathcal{Y}} \cap \mathrm{Lvl}^{\mathcal{Y}}_{4n+2})$ or to $((\mathrm{Ad}^1_i)^{\mathcal{Y}} \cap \mathrm{Lvl}^{\mathcal{Y}}_{4n+2}) \times ((\mathrm{Ad}^1_i)^{\mathcal{Y}} \cap \mathrm{Lvl}^{\mathcal{Y}}_{4n+2})$ as desired. $\square$

The above query ensures the satisfaction of the second condition from Fact 8.19. Note that its size is clearly polynomial w.r.t. the given parameters. For the remaining condition (namely the satisfaction of precisely the same tiles), we employ a nearly identical query. The main difference is that the disjunction that is hidden in the query is no longer binary but $|\mathrm{T}|$-ary, where T is the underlying set of tiles.

Definition 8.28 Let $\mathcal{D}, \bar{\mathrm{c}}, \mathrm{T}, n$, and $q_{\downarrow}$ be as in Definition 8.20, and let us enumerate T as $\mathrm{t}_1, \mathrm{t}_2, \ldots, \mathrm{t}_{|\mathrm{T}|}$.

We define the **tile equality query** $q^{\mathrm{til}}_{\mathrm{eq}}(x, z)$ as

$$\bigwedge_{i=1}^{|\mathrm{T}|} \Big(q_{r_{\mathrm{t}_i}}(y_i, u_{i-1}) \land q_{r_{\mathrm{t}_i}}(y_i, u_i) \Big),$$

after replacing the variables u_0 and u_n, respectively, with x and z.

Note that the size of $q^{\mathrm{til}}_{\mathrm{eq}}$ is polynomial w.r.t. join sizes of $\mathcal{D}$, $\bar{\mathrm{c}}$, and T. We formalise the behaviour of the tile equality query with the following statement. Its proof is nearly the same as the proof of Lemma 8.27.

Corollary 8.29

Let $\mathcal{D}$, $\bar{\mathrm{c}}$, and $q^{\mathrm{til}}_{\mathrm{eq}}$ be as in Definition 8.28, and $\mathcal{Y}$ be a $(\mathcal{D}, \bar{\mathrm{c}})$-jellyfish. Then the set $\mathrm{M}^{\mathrm{til}}_{\mathrm{eq}} := \{(\eta(x), \eta(z)) \mid \mathcal{Y} \models_\eta q^{\mathrm{til}}_{\mathrm{eq}}(x, z)\}$ is equal to $\bigcup_{i=1}^{|\mathrm{T}|} ((\mathrm{C}_{\mathrm{t}_i})^{\mathcal{Y}} \cap \mathrm{Lvl}^{\mathcal{Y}}_{4n+2}) \times ((\mathrm{C}_{\mathrm{t}_i})^{\mathcal{Y}} \cap \mathrm{Lvl}^{\mathcal{Y}}_{4n+2})$.

This establishes the last item of Fact 8.19. We can now conclude our reduction.

Lemma 8.30 Consider $\mathcal{D}$, $\bar{\mathrm{c}}$ be the input for the torus tiling problem. Let $q_{\mathrm{lrl}}(x, y, z)$, q^{i}_{eq}, and $q^{\mathrm{til}}_{\mathrm{eq}}$ be queries defined in Corollary 8.25, Definition 8.26, and Definition 8.28. We have that $\mathcal{Y}$ is proper if and only if it does not satisfy the query $q_{\mathrm{main}} := q_{\mathrm{lrl}}(x, y, z) \land \mathbb{0}(x) \land \mathbb{1}(z) \land \bigwedge_{i=1}^{|\bar{\mathrm{c}}|} q^{i}_{\mathrm{eq}}(x, z) \land q^{\mathrm{til}}_{\mathrm{eq}}(x, z)$.

Proof. Let $\mathcal{Y}$ be a $(\mathcal{D}, \bar{\mathrm{c}})$-jellyfish and suppose that it is not proper. By Fact 8.19 there exists leaves d, e of $\mathcal{Y}$ that satisfy all three conditions from Fact 8.19. It now suffices to construct a match η for q_{main} and $\mathcal{Y}$ that maps x to d and z to e. As d and e are both leaves of $\mathcal{Y}$, the mapping $x \mapsto \mathrm{d}, z \mapsto \mathrm{e}$ can be extended to a match for $\mathcal{Y}$ and $q_{\mathrm{lrl}}(x, y, z)$ by Corollary 8.25. For all $1 \leq i \leq 2n$, as d satisfies Ad^0_i if and only if e satisfies Ad^0_i, the mapping $x \mapsto \mathrm{d}, z \mapsto \mathrm{e}$ can be extended to a match for $\mathcal{Y}$ and all of $q^{i}_{\mathrm{eq}}(x, z)$. Finally, as d and e are labelled with the same tile predicate, we can extend the mapping $x \mapsto \mathrm{d}, z \mapsto \mathrm{e}$ a match for $\mathcal{Y}$ and all of $q^{\mathrm{til}}_{\mathrm{eq}}(x, z)$. As we assume that all the subqueries have only the variables x and z in common, the union of the aforementioned query matches becomes a query match for $\mathcal{Y}$ and q_{main}.

For the other direction, take any match η for $\mathcal{Y}$ and q_{main}. We claim that the elements $\mathrm{d} = \eta(x)$ and $\mathrm{e} = \eta(z)$ satisfy the conditions from Fact 8.19. Indeed, the first condition is guaranteed by the Corollary 8.25 and the fact $q_{\mathrm{lrl}}(x, y, z)$ evaluates to true under η. The second condition, for all $1 \leq i \leq 2n$ is guaranteed by the fact that $q^{i}_{\mathrm{eq}}(x, z)$ evaluates to true under η and by Lemma 8.27. Finally, the third conditions holds, as guaranteed by Corollary 8.29 and the fact that the query $q^{\mathrm{til}}_{\mathrm{eq}}(x, z)$ evaluates to true under η. This concludes the proof. □

The query q_{main} defined above is connected and is clearly of polynomial size w.r.t. $|\mathcal{D}|+|\bar{\mathrm{c}}|$. Hence, we can prove the final theorem of the chapter.

Theorem 8.31

The entailment problem for rooted conjunctive queries for $\mathcal{ALC}^{\mathsf{Self}}$ is coNExpTime-hard. This holds already over $\mathcal{ALC}^{\mathsf{Self}}$-KBs with a single individual name and with no TBox (but with non-atomic concepts allowed in the ABox).

Proof. It suffices to show that the non-entailment problem is NExpTime-hard. To do so, we reduce from the torus tiling problem (see Section 8.1.2). Let $(\mathcal{D}, \bar{\mathrm{c}})$ be an instance of the torus tiling problem, and let $\mathcal{K} := \{\mathrm{Yelly}_{(\mathcal{D},\bar{\mathrm{c}})}(\mathsf{aux})\}$ be an $\mathcal{ALC}^{\mathsf{Self}}$-KB, where aux is a fresh individual name aux. We consider a rooted (and connected) conjunctive query q obtained by replacing the variable y in the query q_{main} defined in Lemma 8.30 with aux. Note that both $\mathcal{K}$ and q are both of size polynomial w.r.t. the input.

We claim that $\mathcal{K} \not\models q$ if and only if $(\mathcal{D}, \bar{\mathrm{c}})$ has a solution. For one direction, suppose that $(\mathcal{D}, \bar{\mathrm{c}})$ has a solution. Then by Fact 8.19 there exists a proper $(\mathcal{D}, \bar{\mathrm{c}})$-jellyfish $\mathcal{Y}$. W.l.o.g. we may assume that $\mathcal{Y}$ interprets $\mathtt{aux}$ as its root. Hence, by Lemma 8.18 we have that $\mathcal{Y} \models \mathcal{K}$. As $\mathcal{Y}$ is proper, by Lemma 8.30 we conclude that $\mathcal{Y} \not\models q$. Hence, $\mathcal{K} \not\models q$, as desired. For the other direction, suppose that $\mathcal{K} \not\models q$. By Part (III) of Lemma 8.18 there exists a $(\mathcal{D}, \bar{\mathrm{c}})$-jellyfish $\mathcal{Y}$ and a homomorphism $\mathfrak{h}$ from $\mathcal{Y}$ *into* $\mathcal{I}$ satisfying $\mathfrak{h}(\varepsilon) = \mathtt{aux}^{\mathcal{I}}$. As query matches are preserved under (inverse) homomorphisms, we conclude that $\mathcal{Y} \not\models q$. This implies, by Lemma 8.30, that $\mathcal{Y}$ is proper. Hence, by Fact 8.19, we conclude that $(\mathcal{D}, \bar{\mathrm{c}})$ has a solution. □

Note that by the correspondence between weak separability problem [JLPW22, p. 3] and the entailment of (unions of) rooted conjunctive queries, our result applies to such a machine-learning-inspired problem.

Part III

Entailment of Queries in Expressive Fragments of $\mathcal{ZOIQ}$

9

Entailment of Positive Two-Way RPQs in Tamed $\mathcal{ZOIQ}$: Combined Complexity

Contents

Motivation and Our Contribution

As we discussed in the previous sections, the query entailment problem is one of the most fundamental problems considered in logic-based knowledge representation. Consequently, the positive two-way regular path queries were advertised [CGLV00] as a common ground between the query languages from the classical, relational database theory such as the (unions of) conjunctive queries, and the more modern query languages for graph databases, including regular path queries. Thus, establishing the exact complexity of the query entailment problem for expressive description logics seems to be a very natural task.

In this section we consider tamed $\mathcal{ZOIQ}$, a common umbrella for $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, and $\mathcal{ZOI}$. Recall that tamed $\mathcal{ZOIQ}$ has the quasi-forest countermodel property, and hence it allows for the employment of tree-automata-based toolkit. Such an approach was taken in the original work by Calvanese et al. [CEO09]. Their approach is not ideal. First of all, the automata construction is very heavy and requires careful treatment. Second, it is exponentially less optimal in the case when the numbers in number restrictions are encoded in binary. Finally, automata-based approaches are difficult to implement, and hence not realistic to use in practice. Hence, an alternative approach is needed. Leveraging said model-theoretic property of the considered DLs, we provide a novel reduction from the query entailment problem $\mathcal{K} \models q$ to the problem of checking unsatisfiability of some other knowledge base $\mathcal{K}_{\neg q}$, written in $\mathcal{ZOIQ}$. This advocates the use of the satisfiability problem as the main reasoning problem to study. To design $\mathcal{K}_q$, we design a certain "proof-like calculus" for deriving query matches in quasi-forest models. Our calculus results in the KB $\mathcal{K}_q$ of size exponential w.r.t. $|\mathcal{K}|$ (even assuming the binary encoding of numbers in number restrictions), which even becomes polynomial when treating the query q as fixed beforehand. Its application allows us to exponentially reduce the P2RPQ entailment problem over tamed $\mathcal{ZOIQ}$ to the satisfiability problem, exponentially improving previous results [CEO09, Thm. 4.3] concerning $\mathcal{ZIQ}$ and $\mathcal{ZOQ}$ (which relied on the unary encoding of counters in number restrictions). The obtained results allow to correspondingly improve a number of previous results on query containment and can be transferred to DLs from the $\mathcal{SR}$ family.

Overview of the Chapter and Prerequisites

We start with a short extra preliminaries in Section 9.1. We then define our proof-based calculus in Section 9.2, starting from a general overview. In Section 9.3 we provide the desired reduction from the query entailment problem to the satisfiability problem. This is then followed by very technical proofs of soundness and completeness of our calculus. We conclude with Section 9.4 not only wrapping-up all the relevant results and applications but also discussing several interesting open problems.

We assume that the reader is familiar with the notion of quasi-forest models from Section 7.1, the syntax and semantics of $\mathcal{ZOIQ}$ from Section 2.10, as well as the section concerning automata (in particular inverse automata, see Section 2.6), decision problems (Section 2.5), and queries (Section 2.8) from Preliminaries.

9.1 Extra Preliminaries: Simple Positive Two-Way RPQs

In what follows we consider the class of positive two-way regular path queries (P2RPQs), as defined in Section 2.8. We call a P2RPQ q *simple* if all its axioms are of the form $\mathcal{A}(x, y)$ for some variables x, y, and an NFA $\mathcal{A}$ over $\mathbf{N_R}$. The following lemma can be shown via a routine renaming.

Lemma 9.1 For any tamed $\mathcal{ZOIQ}$-KB $\mathcal{K}$ and a P2RPQ q we can compute in polynomial time w.r.t. $|\mathcal{K}|+|q|$ a tamed $\mathcal{ZOIQ}$-KB $\mathcal{K}'$ and a *simple* P2RPQ q' such that $\mathcal{K} \models q$ if and only if $\mathcal{K}' \models q$.

Proof sketch. To find the desired query we employ the following rewriting procedure.

(i) We replace any answer variable $\mathsf{a} \in \mathbf{N_I}$ that occur in q with a fresh variable x_{a}, append the atom $\mathrm{A}_{\mathsf{a}}(x_0)$ to the query, and the GCI $\mathrm{A}_{\mathsf{a}} \equiv \{\mathsf{a}\}$ to the knowledge base.

(ii) We replace each atom of the form $\mathrm{C}(x)$ with an atom $r_{\mathrm{C}}(x, x)$ for a fresh role name r_{C} and we append the following axioms to the knowledge-base: $\mathrm{C} \equiv \exists r_{\mathrm{C}}.\mathsf{Self}$ and $\top \sqsubseteq (\leqslant 1\ r_{\mathrm{C}}).\top$. Note that the extra axioms ensure that in every model of the extended knowledge-base, the elements satisfying C are precisely the elements equipped with an r_{C}-self-loop.

(iii) For every atom $\mathcal{R}(x, y)$ from the query, we (a) replace any simple role s that is not a role name and appears in q with a fresh role name r and append an axiom $s = r$ to the knowledge base, and (b) replace any test C? with a fresh role name r_{C}, and similarly to the previous case, we append the following axioms to the knowledge base: $\mathrm{C} \equiv \exists r_{\mathrm{C}}.\mathsf{Self}$ and $\top \sqsubseteq (\leqslant 1\ r_{\mathrm{C}}).\top$. Note that now the rewritten regular expression is built from role names using $\cup$, $\circ$, and *. Hence, with the classical methods we can translate the resulting regular expression into the corresponding NFA.

It is immediate that see the resulting knowledge base and the query are as desired. □

9.2 Annotating (Partial) Query Matches

In the following, suppose that a tamed $\mathcal{ZOIQ}$-KB $\mathcal{K}$ is given, and consider a P2RPQ q for which we want to decide whether $\mathcal{K} \models q$ holds or not. Note that q, by turning q into DNF, it can be rewritten into an equivalent disjunction $\bigvee_{1 \leq i \leq n} q_i$, where the number of disjuncts n can be exponential w.r.t. $|q|$ but each of the queries q_i is only of polynomial size w.r.t. $|q|$. By Lemma 9.1 we may assume that each of the queries q_i is simple. Let $[q]$ denote the set $\{q_1, \ldots, q_n\}$. Then we know that testing whether $\mathcal{K} \not\models q$ amounts to determining whether $\mathcal{K}$ has a model $\mathcal{I}$ which is simultaneous a countermodel (*i.e.* $\mathcal{I} \not\models q_i$) for all the queries q_i from $[q]$. By tameness of $\mathcal{K}$, we can assume that the desired countermodel $\mathcal{I}$ is a quasi-forest.

9.2.1 Proof Overview

We now give an intuitive description of our technique to detect (or refute) matches of q. Given a quasi-forest model $\mathcal{I}$ of $\mathcal{K}$, we iteratively, deterministically annotate all its domain elements d with fresh concept names $\mathrm{Q_M}$ that indicate which "parts" M of some query q_i from $[q]$ match into $\mathcal{I}$ and how d participates in these partial matches. To this end, we employ descriptions M of query parts which contain information about (i) the query variables matched, (ii) (the existence of) paths realising certain state transitions in the query's automata, and (iii) optionally, the "position" of the current d in the query match by use of a marker $\bullet$ acting like an additional, distinguished query variable. In the annotation process, the $\mathrm{Q_M}$s will be assigned to domain elements d based on their local environment and known annotations $\mathrm{Q_{M'}}$ for "smaller" partial queries to the same element d or its direct (role) neighbourhood. As an exception, non-localised query matches M not containing $\bullet$ will be "broadcast", *i.e.* uniformly attached to all domain elements. This way, in the annotation process, $\mathrm{Q_M}$s for larger and larger partial queries are assigned, until finally (after reaching the unique fixpoint) also the full matches for any q_i are recognised by annotations Q_{q_i}. This process will accurately detect partial and full query matches (with the proviso that $\mathcal{I}$ is a quasi-forest). The annotation process is realised by virtue of a (tamed) $\mathcal{ZOIQ}$-KB $\mathcal{K}_q$, the size of which is exponential in q, but only polynomial w.r.t. $|\mathcal{K}|$.

Over the next few pages, we will stepwise introduce the KB $\mathcal{K}_q$, interleaved with some necessary definitions, starting from the relevant notion of a *partial query.* In the following we may equate a conjunctive two-way regular path query q_i with the set of its atoms.

Definition 9.2 Let C2RPQ q_i be of the form $\bigwedge_{i=1}^{n} \mathcal{A}(x_i, y_i)$. A **partial query** for q_i and a subset $\mathrm{X} \subseteq \{x_1, y_1, \ldots, x_m, y_m\}$ of variables from q_i is a set of M consisting of:

- all atoms $\mathcal{A}(x, y)$ from q_i for which $\{x, y\} \subseteq \mathrm{X}$,
- for every atom $\mathcal{A}(x, y)$ from q_i with $x \in X$ but $y \notin X$, one of the atoms $\mathcal{A}^{-}_{\mathsf{q},\mathsf{q}'}(y, \bullet)$ or $\mathcal{A}^{-}_{\mathsf{q},\mathsf{q}'}(y, \mathsf{o})$ where q is a final state of $\mathcal{A}$ and $\mathsf{o} \in \mathsf{ind}(\mathcal{K})$,
- if X is empty, exactly one atom of the form $\mathcal{A}_{\mathsf{q},\mathsf{q}'}(\bullet, \mathsf{o})$ or $\mathcal{A}^{-}_{\mathsf{q},\mathsf{q}'}(\bullet, \mathsf{o})$ (called **nominal-anchored path**) or $\mathcal{A}_{\mathsf{q},\mathsf{q}'}(\bullet, \bullet)$ or $\mathcal{A}^{-}_{\mathsf{q},\mathsf{q}'}(\bullet, \bullet)$ (called **round trip**) for some $\mathcal{A}$ from q_i and $\mathsf{o} \in \mathsf{ind}(\mathcal{K})$.

A partial query is called **monodic**, whenever X is a singleton set. A partial query is called **local** if it contains $\bullet$ and it is called **global** otherwise.

9.2.2 Annotating quasi-forest models

In the forthcoming paragraphs we provide the definitions of suitable axioms of the to-be-constructed $\mathcal{K}_q$.

Nominal-Anchored Paths First, we want to detect role paths that start in the to-be-annotated individual, end in named individual $\mathsf{o}^{\mathcal{I}}$, and realise state transitions in one of the query's automata. Let o be any individual name in $\mathcal{K}$, let $\mathcal{A}^{\pm}$ be either $\mathcal{A}$ or $\mathcal{A}^{-}$ for any automaton $\mathcal{A}$ occurring in q with states $\mathsf{q}, \mathsf{q}', \mathsf{q}''$. We let $\mathcal{K}_q$ contain the axiom

$$Q_{\mathcal{A}^{\pm}_{\mathsf{q},\mathsf{q}}(\bullet,\mathsf{o})}(\mathsf{o}) \tag{9.1}$$

and whenever $\mathcal{A}^{\pm}$ has a transition $(\mathsf{q}, r, \mathsf{q}')$ we also have the axiom:

$$\exists r.Q_{\mathcal{A}^{\pm}_{\mathsf{q}',\mathsf{q}''}(\bullet,\mathsf{o})} \sqsubseteq Q_{\mathcal{A}^{\pm}_{\mathsf{q},\mathsf{q}''}(\bullet,\mathsf{o})} \tag{9.2}$$

Round Trips Next, we are concerned with paths which start and end in the to-be-annotated individual. Assuming an automaton $\mathcal{A}$ with states $\mathsf{q}, \mathsf{q}', \mathsf{q}'', \mathsf{q}'''$ and transition set $\mathtt{T}$ as well as an individual name o, we add the axioms:

$$\top \sqsubseteq Q_{\mathcal{A}^-_{\mathsf{q},\mathsf{q}}(\bullet,\bullet)} \tag{9.3}$$

$$Q_{\mathcal{A}_{\mathsf{q},\mathsf{q}'}(\bullet,\bullet)} \sqcap Q_{\mathcal{A}_{\mathsf{q}',\mathsf{q}''}(\bullet,\bullet)} \sqsubseteq Q_{\mathcal{A}_{\mathsf{q},\mathsf{q}''}(\bullet,\bullet)} \tag{9.4}$$

$$Q_{\mathcal{A}_{\mathsf{q},\mathsf{q}'}(\bullet,\mathsf{o})} \sqcap Q_{\mathcal{A}^-_{\mathsf{q}'',\mathsf{q}'}(\bullet,\mathsf{o})} \sqsubseteq Q_{\mathcal{A}_{\mathsf{q},\mathsf{q}''}(\bullet,\bullet)} \tag{9.5}$$

$$\text{for } (\mathsf{q},r,\mathsf{q}') \in \mathtt{T}: \qquad \exists r.\mathsf{Self} \sqsubseteq Q_{\mathcal{A}_{\mathsf{q},\mathsf{q}'}(\bullet,\bullet)} \tag{9.6}$$

$$\text{for } \{(\mathsf{q},r,\mathsf{q}'),(\mathsf{q}'',s,\mathsf{q}''')\} \subseteq \mathtt{T}: \qquad \exists(r \cap s^-).Q_{\mathcal{A}_{\mathsf{q}',\mathsf{q}''}(\bullet,\bullet)} \sqsubseteq Q_{\mathcal{A}_{\mathsf{q},\mathsf{q}'''}(\bullet,\bullet)} \tag{9.7}$$

Initialising monodic partial queries Next we determine domain elements to which separate query variables could possibly be mapped in a match.

Definition 9.3 We call a monodic partial query M **original**, if $\mathsf{q} = \mathsf{q}'$ for every $\mathcal{A}_{\mathsf{q},\mathsf{q}'}(x,\bullet) \in \mathrm{M}$ and every $\mathcal{A}^-_{\mathsf{q},\mathsf{q}'}(x,\bullet) \in \mathrm{M}$. For M an original monodic partial query for q and $\{x\}$, the set Prec_M of **precondition concepts** consists of:

- $Q_{\mathcal{A}_{\mathsf{q},\mathsf{q}'}(\bullet,\mathsf{o})}$ for each atom $\mathcal{A}_{\mathsf{q},\mathsf{q}'}(x,\mathsf{o})$ in M,
- $Q_{\mathcal{A}^-_{\mathsf{q},\mathsf{q}'}(\bullet,\mathsf{o})}$ for each atom $\mathcal{A}^-_{\mathsf{q},\mathsf{q}'}(x,\mathsf{o})$ in M,
- $\bigsqcup_{\substack{\mathsf{q}\text{ initial}\\ \mathsf{q}'\text{ final}}} Q_{\mathcal{A}_{\mathsf{q},\mathsf{q}'}(\bullet,\bullet)}$ for each atom $\mathcal{A}(x,x)$ in M.

Now we can initialise all original monodic partial queries M by adding the following axioms to $\mathcal{K}_q$:

$$\bigsqcap \mathrm{Prec}_\mathrm{M} \sqsubseteq Q_\mathrm{M} \tag{9.8}$$

Updating partial queries Partial queries can be updated by roundtrips: Let M be a partial query containing some $\mathcal{A}^\pm_{\mathsf{q},\mathsf{q}'}(x,\bullet)$. Then we realise the update by adding the following axioms to $\mathcal{K}_q$:

$$Q_\mathrm{M} \sqcap Q_{\mathcal{A}^\pm_{\mathsf{q}',\mathsf{q}''}(\bullet,\bullet)} \sqsubseteq Q_{\mathrm{M}\setminus\{\mathcal{A}^\pm_{\mathsf{q},\mathsf{q}'}(x,\bullet)\}\cup\{\mathcal{A}^\pm_{\mathsf{q},\mathsf{q}''}(x,\bullet)\}} \tag{9.9}$$

Furthermore, a partial query can "progress" when making a "step" in the model, moving from the considered d to some role neighbour. In such a step, all the "unready" paths (corresponding to atoms with $\bullet$) must be updated in a synchronous manner. Given a partial query M for q_i and nonempty X as well as a set $\{r_1,\ldots,r_k\}$ of (possibly inverted) role names, assume M′ can be obtained from M by replacing each atom of the form $\mathcal{A}^\pm_{\mathsf{q},\mathsf{q}'}(x,\bullet)$ by an atom $\mathcal{A}^\pm_{\mathsf{q},\mathsf{q}''}(x,\bullet)$ such that $\mathcal{A}^\pm$ has a transition $(\mathsf{q}',r,\mathsf{q}'')$ for any $r \in \{r_1,\ldots,r_k\}$. Then add the following axiom to the KB $\mathcal{K}_q$:

$$\exists(r_1^- \cap r_2^- \cap \ldots \cap r_m^-).Q_\mathrm{M} \sqsubseteq Q_{\mathrm{M}'} \tag{9.10}$$

Joining partial queries When two partial queries corresponding to the same query q_i "meet" in a domain element d, they can – under certain circumstances – be "glued together" into a "bigger" partial query of q_i.

Definition 9.4 Let M_1 be a partial query for q_i and X_1 and let M_2 be a partial query for q_i and X_2. We call M_1 and M_2 **joinable** if

- $\mathrm{X}_1 \neq \emptyset$, $\mathrm{X}_2 \neq \emptyset$, $\mathrm{X}_1 \cap \mathrm{X}_2 = \emptyset$, and
- for each $\mathcal{A}(x,y) \in q_i \setminus (\mathrm{M}_1 \cup \mathrm{M}_2)$ with $\{x,y\} \subseteq \mathrm{X}_1 \cup \mathrm{X}_2$, there are states $\mathsf{q},\mathsf{q}',\mathsf{q}''$ of $\mathcal{A}$ with q initial and q'' final, such that either $\{\mathcal{A}_{\mathsf{q},\mathsf{q}'}(x,\bullet), \mathcal{A}^-_{\mathsf{q}'',\mathsf{q}'}(y,\bullet)\} \in \mathrm{M}_1 \cup \mathrm{M}_2$ or $\{\mathcal{A}_{\mathsf{q},\mathsf{q}'}(x,\mathsf{o}), \mathcal{A}^-_{\mathsf{q}'',\mathsf{q}'}(y,\mathsf{o})\} \in \mathrm{M}_1 \cup \mathrm{M}_2$ for some $\mathsf{o} \in \mathbf{N_I}$.

For joinable M_1 and M_2, their **join**, denoted $\mathrm{M}_1 \bowtie \mathrm{M}_2$, is the partial query obtained from $\mathrm{M}_1 \cup \mathrm{M}_2$ by replacing any pair of atoms $\mathcal{A}_{\mathsf{q},\mathsf{q}'}(x,*)$ and $\mathcal{A}^-_{\mathsf{q}'',\mathsf{q}'}(y,*)$ by $\mathcal{A}(x,y)$, where $*$ is either an individual name o or $\bullet$.

We implement the join operation for every pair M_1, M_2 of joinable partial queries for a $q_i \in [q]$ by extending $\mathcal{K}_q$ with:

$$Q_{\mathrm{M}_1} \sqcap Q_{\mathrm{M}_2} \sqsubseteq Q_{\mathrm{M}_1 \bowtie \mathrm{M}_2} \tag{9.11}$$

Broadcasting global partial queries Whenever a partial query M does not have any occurrences of $\bullet$, which would tie it to a specific element, it will be "broadcast" to all domain elements:

$$\exists \top . Q_{\mathrm{M}} \sqsubseteq Q_{\mathrm{M}} \tag{9.12}$$

This concludes the definition of $\mathcal{K}_q$. Revisiting our initial intuition of $\mathcal{K}_q$'s purpose of deterministically annotating a model $\mathcal{I}$ of $\mathcal{K}$, the following lemma formalises this by singling out one unique annotated model $\mathcal{I}^*$ for every $\mathcal{I}$. We will call the models $\mathcal{I}^*$ from Lemma 9.5 **Q-minimal**.

Lemma 9.5 Let $\mathcal{K}$ be a $\mathcal{ZOIQ}$-KB and q a simple P2RPQ. For every model $\mathcal{I}$ of $\mathcal{K}$ there exists a unique model $\mathcal{I}^*$ of $\mathcal{K} \cup \mathcal{K}_q$ which extends $\mathcal{I}$ and satisfies $Q_{\mathrm{M}}^{\mathcal{I}^*} \subseteq Q_{\mathrm{M}}^{\mathcal{J}}$ for every Q_{M} in $\mathcal{K}_q$ and for every model $\mathcal{J}$ of $\mathcal{K} \cup \mathcal{K}_q$ that extends $\mathcal{I}$.

Proof sketch. Observe that all axioms of $\mathcal{K}_q$ can be easily expressed in monadic Datalog. Hence, for every model $\mathcal{I}$ of $\mathcal{K}$, any concept membership $\mathrm{d} \in Q_{\mathrm{M}}^{\mathcal{I}^*}$ holding in the corresponding Q-minimal model $\mathcal{I}^*$ can be derived from concept and role memberships in $\mathcal{I}$ through a finite sequence of "applications" of axioms from $\mathcal{K}_q$. □

9.3 Reduction to Satisfiability

Now we establish technical results relating the presence of Q_{M}-annotations in models and the actual semantic satisfaction of the corresponding partial query M. Note that any partial query M can be seen as a (set representation of a) C2RPQ, assuming that $\bullet$ is just an ordinary variable.

Definition 9.6 Given an interpretation $\mathcal{I}$, a domain element $\mathrm{d} \in \Delta^{\mathcal{I}}$, and a partial query M, we say that M is **satisfied** or **holds** in d, written $\mathcal{I}, \mathrm{d} \models \mathrm{M}$, if there is a match η for M in $\mathcal{I}$ with $\eta(\bullet) = \mathrm{d}$. M is **satisfied** or **holds tightly** in d, written $\mathcal{I}, \mathrm{d} \Vvdash \mathrm{M}$, if η is such that every $\mathcal{A}^{\pm}_{\mathsf{q}_1,\mathsf{q}_2}(x, \bullet) \in \mathrm{M}$ is realized by a path $\eta(x) \rightsquigarrow \mathrm{d}$ not containing $\mathsf{o}^{\mathcal{I}}$ for any individual name o.

Note that $\models$ and $\Vvdash$ coincide whenever M is global or does not contain variables. Note also that, as a consequence of this definition, if M is global, then Q_{M} holds (tightly) everywhere or nowhere throughout the domain. With this notion in place, the next two lemmas can be seen as soundness and completeness results regarding the deduction calculus for (partial) query matches embodied by $\mathcal{K}_q$.

Lemma 9.7 Let $\mathcal{I}$ be a Q-minimal model of $\mathcal{K} \cup \mathcal{K}_q$ and let $\mathrm{d} \in \Delta^{\mathcal{I}}$. Let M be any partial query for any C2RPQ $q_i \in [q]$. Then $\mathrm{d} \in Q_{\mathrm{M}}^{\mathcal{I}}$ implies $\mathcal{I}, \mathrm{d} \models \mathrm{M}$.

Lemma 9.8 Let $\mathcal{I}$ be a quasi-forest model of $\mathcal{K} \cup \mathcal{K}_q$ and let $\mathrm{d} \in \Delta^{\mathcal{I}}$. Let M be any partial query for any C2RPQ $q_i \in [q]$. Then $\mathcal{I}, \mathrm{d} \Vvdash \mathrm{M}$ implies $\mathrm{d} \in Q_{\mathrm{M}}^{\mathcal{I}}$.

The proofs of the two lemmas above will be given in separate sections. Soundness is proven by induction over the length of the derivation for $\mathrm{d} \in Q_{\mathrm{M}}^{\mathcal{I}}$. Completeness is shown by induction over the *spread* of the match for M, *i.e.* the number of variables (excluding $\bullet$) plus the sum of the lengths of all paths realising the query atoms. Thanks to these correspondences, we can essentially rule out models with matches of any $q_1, \ldots, q_n$ by forcing the extensions of $Q_{q_1}, \ldots, Q_{q_n}$ to be empty. Therefore, let

$$\mathcal{K}_{\neg q} = \mathcal{K} \cup \mathcal{K}_q \cup \{Q_{q_i} \sqsubseteq \bot \mid q_i \in [q]\}. \tag{9.13}$$

We establish some syntactic and semantic properties of $\mathcal{K}_{\neg q}$.

Fact 9.9. The size of $\mathcal{K}_{\neg q}$ is single exponential in the total size of $\mathcal{K}$ and q. It is polynomial, if the number of atoms in q is bounded by a constant.

Proof sketch. Routine calculations. □

We next see that our construction preserves "tameness" of the underlying KB.

Fact 9.10. If $\mathcal{K}$ is written in tamed $\mathcal{ZOIQ}$, then $\mathcal{K}_{\neg q}$ is in tamed $\mathcal{ZOIQ}$ as well.

Proof. First note that $\mathcal{K}_{\neg q}$ is in $\mathcal{ZOIQ}$. For tameness, consider a model $\mathcal{I}$ of $\mathcal{K}_{\neg q}$. As $\mathcal{I}$ is a model of $\mathcal{K}$ and the latter is tame, there exists a quasi-forest model $\mathcal{J}$ of $\mathcal{K}$ and a homomorphism $\mathfrak{h}\colon \mathcal{J} \to \mathcal{I}$. Now, we can construct a quasi-forest model $\mathcal{J}'$ of $\mathcal{K}_{\neg q}$ by extending $\mathcal{J}$, letting $\mathrm{d} \in Q_{\mathrm{M}}^{\mathcal{J}'}$ if $\mathfrak{h}(\mathrm{d}) \in Q_{\mathrm{M}}^{\mathcal{I}}$ for all concept names Q_{M} occurring in $\mathcal{K}$. □

Based on Fact 9.9 and Fact 9.10 (and with the proviso that our method is sound and complete) we infer the main theorem of this section.

Theorem 9.11

Let $\mathcal{K}$ be a tamed $\mathcal{ZOIQ}$-KB and let q be a simple P2RPQ over $\mathcal{K}$. Then $\mathcal{K} \models q$ if and only if $\mathcal{K}_{\neg q}$ is unsatisfiable.

Proof. It suffices to show that $\mathcal{K} \not\models q$ if and only if $\mathcal{K}_{\neg q}$ satisfiable. For the "only if" direction, suppose that $\mathcal{K} \not\models q$. Then there exists an interpretation $\mathcal{I}$ with $\mathcal{I} \models \mathcal{K}$ and $\mathcal{I} \not\models q$. Consider $\mathcal{I}^*$, the Q-minimal model of $\mathcal{K} \cup \mathcal{K}_q$ extending $\mathcal{I}$. If some $Q_{q_i}^{\mathcal{I}^*}$ were nonempty, then there would be a query match of q into $\mathcal{I}$ by Lemma 9.7, contradicting our assumption. Therefore, $\mathcal{I}^*$ satisfies the GCI $Q_{q_i} \sqsubseteq \bot$ for all queries $q_i \in [q]$. Thus $\mathcal{I}^* \models \mathcal{K}_{\neg q}$.
For the reverse implication, let $\mathcal{K}_{\neg q}$ be satisfiable. As $\mathcal{K}_{\neg q}$ is in tamed $\mathcal{ZOIQ}$, we know that there is a quasi-forest model $\mathcal{I}$ of $\mathcal{K}_{\neg q}$. The satisfaction of the GCIs $Q_{q_i} \sqsubseteq \bot$ by $\mathcal{I}$ implies the emptiness of $Q_{q_i}^{\mathcal{I}}$ for every query $q_i \in [q]$. We claim that $\mathcal{I} \not\models q$. Towards a contradiction suppose that $\mathcal{I} \models q_i$ for some $q_i \in [q]$. Then, Lemma 9.8 this implies that $\mathrm{d} \in Q_{q_i}^{\mathcal{I}}$ for all domain elements $\mathrm{d} \in \Delta^{\mathcal{I}}$. A contradiction. Hence, $\mathcal{I}$ is the desired countermodel for $\mathcal{K}$ and q. □

Theorem 9.11 yields the desired reduction from the query entailment problem for tamed $\mathcal{ZOIQ}$, and, in particular, in its decidable fragments with ExpTime-complete satisfiability problem. Based on this theorem, we can now prove our central result:

Theorem 9.12

The entailment problem for the class of positive two-way regular path queries over tamed $\mathcal{ZOIQ}$-KBs is 2ExpTime-complete, even if the numbers appearing in number restrictions are encoded in binary. Moreover, for any P2RPQ q there is an exponential time algorithm (parametrised by q) that takes a tamed $\mathcal{ZOIQ}$-KB $\mathcal{K}$ as the input and checks whether $\mathcal{K} \models q$ holds.

Proof. By our discussion at the beginning of Section 9.2 we can assume that q is a disjunction (of exponential size w.r.t. the input query) of simple C2RPQ (of polynomial size w.r.t. the input query). By Theorem 9.11, entailment of a simple P2RPQ q from a tamed $\mathcal{ZOIQ}$-KB $\mathcal{K}$ can be reduced to checking unsatisfiability of $\mathcal{K}_{\neg q}$, which can be computed in output-polynomial time and the size of which is exponential (polynomial, if the number of atoms in q is bounded). By Lemma 7.6, (un)satisfiability of $\mathcal{K}_{\neg q}$ can be checked in exponential size w.r.t. the size of $\mathcal{K}_{\neg q}$. Thus we obtain the desired 2ExpTime and ExpTime upper bounds. The matching lower bounds are well-known even for much weaker logics, *e.g.* $\mathcal{ALC}$Self from Chapter 6. □

We next prove soundness and completeness of our method. The proofs are quite tedious and (unfortunately) not as polished as the rest of the thesis, so we recommend the reader to skip them for first reading. Their revised version will be available at some point in the journal versions of the corresponding papers. For brevity, in the forthcoming proof we will often write $\mathrm{d} \leadsto^{\rho} \mathrm{d}'$ whenever there exists a path ρ from d and ρ'. We say that a path ρ realises a transition $\mathsf{q} \rightarrow^{\rho} \mathsf{q}'$ whenever $\rho \models \mathcal{A}_{\mathsf{q},\mathsf{q}'}$, as defined before.

9.3.1 Proof of Lemma 9.7

We prove the Lemma by induction over the length of the derivation sequence for $\mathrm{d} \in Q_{\mathrm{M}}$, which uses rules from $\mathcal{K}_q$. By induction hypothesis, Lemma 9.7 holds for all previously derived concept memberships. In order to complete the proof, we will make a case distinction and in each case we will provide a match η witnessing $\mathcal{I}, \mathrm{d} \models \mathrm{M}$.

Let us start with the case when $\mathrm{M} = \{\mathcal{A}^{\pm}_{\mathsf{q}_1,\mathsf{q}_2}(\bullet, \circ)\}$.

- The concept membership $\mathrm{d} \in Q^{\mathcal{I}}_{\mathcal{A}^{\pm}_{\mathsf{q},\mathsf{q}}(\bullet,\circ)}$ has been derived via Rule 9.1. Thus $\mathrm{d} = \mathsf{o}^{\mathcal{I}}$ and $\mathcal{A}^{\pm}_{\mathsf{q},\mathsf{q}}(\bullet, \circ)$ is trivially realised by an empty path from d to d. Hence, we can set $\eta = \{\bullet \mapsto \mathrm{d}\}$.
- The concept membership $\mathrm{d} \in Q^{\mathcal{I}}_{\mathcal{A}^{\pm}_{\mathsf{q},\mathsf{q}''}(\bullet,\circ)}$ has been derived via Rule 9.2. Then we know that d has an r-successor d' satisfying $\mathrm{d}' \in Q^{\mathcal{I}}_{\mathcal{A}^{\pm}_{\mathsf{q}',\mathsf{q}''}(\bullet,\circ)}$. By induction hypothesis there exists a variable assignment η' witnessing that $\mathcal{A}^{\pm}_{\mathsf{q}',\mathsf{q}''}(\bullet, \circ)$ holds in d', *i.e.* there is a path ρ from d' to $\mathsf{o}^{\mathcal{I}}$ realising a transition $\mathsf{q}' \rightarrow^{\rho} \mathsf{q}''$. This path can be extended to some path ρ' by prepending the path $\mathrm{d} \leadsto^{r} \mathrm{d}'$. Since there is a transition $(\mathsf{q}, r, \mathsf{q}')$ in $\mathcal{A}^{\pm}$, the path ρ' from d to $\mathsf{o}^{\mathcal{I}}$ indeed realises a transition $\mathsf{q} \rightarrow^{\rho'} \mathsf{q}''$ in $\mathcal{A}$. So we set $\eta = \{\bullet \mapsto \mathrm{d}\}$.

The next cases we consider are round-trips.

- The concept membership $\mathrm{d} \in Q^{\mathcal{I}}_{\mathcal{A}^{-}_{\mathsf{q},\mathsf{q}}(\bullet,\bullet)}$ has been derived via Rule 9.3. Then $\mathcal{A}^{-}_{\mathsf{q},\mathsf{q}}(\bullet, \bullet)$ is trivially realised by an empty path from d to d. Thus we can set $\eta = \{\bullet \mapsto \mathrm{d}\}$.
- The concept membership $\mathrm{d} \in Q^{\mathcal{I}}_{\mathcal{A}_{\mathsf{q},\mathsf{q}''}(\bullet,\bullet)}$ has been derived via Rule 9.4. Then $\mathrm{d} \in Q^{\mathcal{I}}_{\mathcal{A}_{\mathsf{q},\mathsf{q}'}(\bullet,\bullet)}$ as well as $\mathrm{d} \in Q^{\mathcal{I}}_{\mathcal{A}_{\mathsf{q}',\mathsf{q}''}(\bullet,\bullet)}$. By the induction hypothesis, there exist paths ρ and ρ' both from d to d, which realise transitions $\mathsf{q} \rightarrow^{\rho} \mathsf{q}'$ and $\mathsf{q}' \rightarrow^{\rho'} \mathsf{q}''$. Thus a transition $\mathsf{q} \rightarrow^{\rho\rho'} \mathsf{q}''$ is also realised by the path $\rho{\cdot}\rho'$ from d to d and we can define a match $\eta = \{\bullet \mapsto \mathrm{d}\}$.
- The concept membership $\mathrm{d} \in Q^{\mathcal{I}}_{\mathcal{A}_{\mathsf{q},\mathsf{q}''}(\bullet,\bullet)}$ has been derived via Rule 9.5. Then $\mathrm{d} \in Q^{\mathcal{I}}_{\mathcal{A}_{\mathsf{q},\mathsf{q}'}(\bullet,\circ)}$ as well as $\mathrm{d} \in Q^{\mathcal{I}}_{\mathcal{A}^{-}_{\mathsf{q}'',\mathsf{q}'}(\bullet,\circ)}$. By the induction hypothesis there is a path ρ' from d to $\mathsf{o}^{\mathcal{I}}$ realising a transition $\mathsf{q} \rightarrow^{\rho'} \mathsf{q}'$ from $\mathcal{A}$ and there is a path ρ'' from d to $\mathsf{o}^{\mathcal{I}}$ realising a transition $\mathsf{q}'' \rightarrow^{\rho''} \mathsf{q}'$ from $\mathcal{A}^{-}$. Let $(\rho'')^{-}$ be the inverse path of ρ'', *i.e.* the path is reversed and all roles are replaced by their inverses. Thus the path ρ consisting of $\mathrm{d} \leadsto^{\rho'} \mathsf{o}^{\mathcal{I}} \leadsto^{(\rho'')^{-}} \mathrm{d}$ realises a transition $\mathsf{q} \rightarrow^{\rho} \mathsf{q}''$ in $\mathcal{A}$. So we set $\eta = \{\bullet \mapsto \mathrm{d}\}$.
- The concept membership $\mathrm{d} \in Q^{\mathcal{I}}_{\mathcal{A}_{\mathsf{q},\mathsf{q}'}(\bullet,\bullet)}$ has been derived via Rule 9.6. Since $\mathrm{d} \in (\exists r.\mathsf{Self})^{\mathcal{I}}$, we know that there is a path $\mathrm{d} \leadsto^{r} \mathrm{d}$, that by assumption realises the transition $\mathsf{q} \rightarrow^{r} \mathsf{q}'$ in $\mathcal{A}$. Thus we set $\eta = \{\bullet \mapsto \mathrm{d}\}$.
- The concept membership $\mathrm{d} \in Q^{\mathcal{I}}_{\mathcal{A}_{\mathsf{q},\mathsf{q}'''}(\bullet,\bullet)}$ has been derived via Rule 9.7. Thus there exists an element $\mathrm{d}' \in \Delta^{\mathcal{I}}$, reached from d via r and s^{-}. Such an element satisfies $\mathrm{d}' \in Q^{\mathcal{I}}_{\mathcal{A}_{\mathsf{q}',\mathsf{q}''}(\bullet,\bullet)}$, thus by the induction hypothesis there is a variable assignment η', witnessing that $\mathcal{A}_{\mathsf{q}',\mathsf{q}''}(\bullet, \bullet)$ holds at d'. Thus, there is a path ρ', leading from d' to d', realising a transition $\mathsf{q}' \rightarrow^{\rho'} \mathsf{q}''$ in $\mathcal{A}$. Since automaton $\mathcal{A}$ has transitions $(\mathsf{q}, r, \mathsf{q}')$ and $(\mathsf{q}'', s, \mathsf{q}''')$, the path ρ defined as $\mathrm{d} \leadsto^{r} \mathrm{d}' \leadsto^{\rho'} \mathrm{d}' \leadsto^{r} \mathrm{d}$ realises a transition $\mathsf{q} \rightarrow^{\rho} \mathsf{q}'''$ in $\mathcal{A}$. Hence we take $\eta = \{\bullet \mapsto \mathrm{d}\}$.

Now we will focus on partial queries M with variables.

- Assume that the concept membership $\mathrm{d} \in Q^{\mathcal{I}}_{\mathrm{M}}$ has been derived via Rule 9.8 for some original monodic partial query M for a variable x. Then we also know that $\mathrm{d} \in (\bigsqcap \mathrm{Prec}_{\mathrm{M}})^{\mathcal{I}}$, so for each $Q_{M'}$ from the preconditions satisfied in d the variable assignment $\eta' = \{\bullet \mapsto \mathrm{d}\}$ is a match. Now we show that $\eta = \{\bullet \mapsto \mathrm{d}, x \mapsto \mathrm{d}\}$ witnesses that M holds at d. Indeed, considering any atom from M, we will show that it is realised by η. If this atom is of the form $\mathcal{A}_{\mathsf{q},\mathsf{q}'}(x,\mathsf{o})$, note that, due to the fact that d satisfies the precondition, d also belongs to $Q^{\mathcal{I}}_{\mathcal{A}_{\mathsf{q},\mathsf{q}'}(\bullet,\mathsf{o})}$. Then, by induction hypothesis, there is a role path ρ from $\eta'(\bullet) = \mathrm{d} = \eta(x)$ to $\mathsf{o}^{\mathcal{I}}$ realising a transition $\mathsf{q} \to^{\rho} \mathsf{q}'$ in $\mathcal{A}$ and hence witnessing $\mathcal{A}_{\mathsf{q},\mathsf{q}'}(x,\mathsf{o})$. The argument transfers easily to the case when an atom is of the form $\mathcal{A}^{-}_{\mathsf{q},\mathsf{q}'}(x,\mathsf{o})$. Finally, let us consider an atom $\mathcal{A}(x,x)$ from M. But then $\mathrm{d} \in Q^{\mathcal{I}}_{\mathcal{A}_{\mathsf{q},\mathsf{q}'}(\bullet,\bullet)}$ (for an initial state q and a final state q′ of $\mathcal{A}$), therefore, by induction hypothesis, a path ρ accepted by $\mathcal{A}$ from $\eta'(\bullet) = \mathrm{d} = \eta(x)$ to itself exists, witnessing $\mathcal{A}(x,x)$. Thus a match η for M holds at d.

- The concept membership $\mathrm{d} \in Q^{\mathcal{I}}_{\mathrm{M}}$ has been derived via Rule 9.9. Then M contains some $\mathcal{A}^{\pm}_{\mathsf{q},\mathsf{q}'}(x,\bullet)$. From the induction hypothesis follows that there exists a variable assignment η witnessing that M holds at d. We claim that η also witnesses that $\mathrm{M}' = \mathrm{M} \setminus \{\mathcal{A}^{\pm}_{\mathsf{q},\mathsf{q}'}(x,\bullet)\} \cup \{\mathcal{A}^{\pm}_{\mathsf{q},\mathsf{q}''}(x,\bullet)\}$ holds in d. All but one atom of M′ are contained in M and hence must be realized by induction hypothesis, so it remains to prove that the only new automaton atom $\mathcal{A}^{\pm}_{\mathsf{q},\mathsf{q}''}(x,\bullet)$ is satisfied. By the induction hypothesis there is a path ρ' from $\eta(x)$ to $\eta(\bullet) = \mathrm{d}$ realising a transition $\mathsf{q} \to^{\rho'} \mathsf{q}'$ in $\mathcal{A}^{\pm}$ (due to the assumption that M contains $\{\mathcal{A}^{\pm}_{\mathsf{q},\mathsf{q}'}(x,\bullet)\}$). Also, since $\mathrm{d} \in Q^{\mathcal{I}}_{\mathcal{A}^{\pm}_{\mathsf{q}',\mathsf{q}''}(\bullet,\bullet)}$ there is a round-trip ρ'', from d to d, realising a transition $\mathsf{q}' \to^{\rho''} \mathsf{q}''$ in $\mathcal{A}^{\pm}$. Hence the path $\rho = \rho'\rho''$ leads from $\eta(x)$ to $\eta(\bullet) = \mathrm{d}$ and realises a transition $\mathsf{q} \to^{\rho} \mathsf{q}''$ in $\mathcal{A}^{\pm}$, showing our claim.
- The concept membership $\mathrm{d} \in Q^{\mathcal{I}}_{\mathrm{M}'}$ has been derived via Rule 9.10. Then d has a predecessor $\mathrm{d}' \in Q^{\mathcal{I}}_{\mathrm{M}}$, reaching d via any of $r_1, r_2, \ldots, r_m$. Let η' be a variable assignment, given by the induction hypothesis, witnessing that M holds at d′. We obtain a variable assignment η from η' by setting $\eta(\bullet) = \mathrm{d}$. Now it remains to show that all the automata atoms are satisfied. Let $\mathcal{A}^{\pm}_{\mathsf{q},\mathsf{q}''}(x,\bullet)$ be an arbitrary atom from M′ and let $\mathcal{A}^{\pm}_{\mathsf{q},\mathsf{q}'}(x,\bullet)$ be the corresponding atom from M. By the induction hypothesis there exists a path ρ from $\eta'(x)$ to $\eta'(\bullet) = \mathrm{d}'$ realising a transition $\mathsf{q} \to^{\rho} \mathsf{q}'$ in $\mathcal{A}^{\pm}$. But there is also a role r leading from d′ to d, such that there is a transition $(\mathsf{q}', r, \mathsf{q}'')$ in $\mathcal{A}^{\pm}$. Thus an obtained path $\rho' = \rho r$, such that $\eta'(x) \rightsquigarrow^{\rho} \mathrm{d}' \rightsquigarrow^{r_i} \mathrm{d}$, witnesses a transition $\mathsf{q} \to^{\rho'} \mathsf{q}''$ in $\mathcal{A}^{\pm}$. Hence all atoms from M′ correspond to appropriate paths under η'.
- The concept membership $\mathrm{d} \in Q^{\mathcal{I}}_{\mathrm{M}}$ has been derived via Rule 9.11 for a join M of two joinable partial queries M_1 and M_2. Both of $\mathrm{d} \in Q^{\mathcal{I}}_{\mathrm{M}_1}$ and $\mathrm{d} \in Q^{\mathcal{I}}_{\mathrm{M}_2}$ were derived earlier, thus by the induction hypothesis there are two matches η_1 and η_2 witnessing, respectively, that M_1 and M_2 hold at d. We let $\eta = \eta_1 \cup \eta_2$, which is unambiguous since η_1 and η_2 coincide on $\bullet$ and do not share any other variables. We will argue that η witnesses that M holds at d. First of all, observe that if $\mathrm{M}_1 \cup \mathrm{M}_2$ contains some atom of the form $\mathcal{A}(x,x)$ or $\mathcal{A}^{\pm}_{\mathsf{q},\mathsf{q}'}(x,\mathsf{o})$ for any variable X, this atom must already have been present in M_1 or M_2, hence have been realized by η_1 or η_2 by induction hypothesis, therefore it is also realized by η. To complete the proof we need to ensure that also atoms of the form $\mathcal{A}(x,y)$ are realised. If $\mathcal{A}(x,y)$ occurs in M_1 or M_2, then its realisation is guaranteed by the induction hypothesis (the same argument as before). If not, then it was created from two atoms $\mathcal{A}_{\mathsf{q},\mathsf{q}'}(x,*)$ and $\mathcal{A}^{-}_{\mathsf{q}'',\mathsf{q}'}(y,*)$. We distinguish two possibilities. (i) Assume $* = \mathsf{o} \in \mathbf{N_I}$. In this case we know by induction hypothesis that each of $\mathcal{A}_{\mathsf{q},\mathsf{q}'}(x,\mathsf{o})$ and $\mathcal{A}^{-}_{\mathsf{q}'',\mathsf{q}'}(y,\mathsf{o})$ is realized by η_1 or η_2 and hence by η. That is, there is a path ρ' from $\eta(x)$ to $\mathsf{o}^{\mathcal{I}}$ realising a transition $\mathsf{q} \to^{\rho'} \mathsf{q}'$ from $\mathcal{A}$ and there is a path ρ'' from $\eta(y)$ to $\mathsf{o}^{\mathcal{I}}$ realising a transition $\mathsf{q}'' \to^{\rho''} \mathsf{q}'$ from $\mathcal{A}^{-}$. Let $(\rho'')^{-}$ be the inverse path of ρ'', *i.e.* the path is reversed and all roles are replaced by their inverses. Thus the path ρ consisting of $\eta(x) \rightsquigarrow^{\rho'} \mathsf{o}^{\mathcal{I}} \rightsquigarrow^{(\rho'')^{-}} \eta(y)$ realises a transition $\mathsf{q} \to^{\rho} \mathsf{q}''$ in $\mathcal{A}$. As q is initial and q″ is final in $\mathcal{A}$, $\mathcal{A}(x,y)$ is realized by η. (ii) The case $* = \bullet$ is analogous to the previous with $\mathsf{o}^{\mathcal{I}}$ replaced by d.

  Thus η is indeed a match witnessing that $\mathrm{M} = \mathrm{M}_1 \bowtie \mathrm{M}_2$ holds at d.
- The concept membership has been derived via Rule 9.11. Then some $\overline{\top}$-predecessor d′ of d (*i.e.* an arbitrary d′ from $\mathrm{d}^{\mathcal{I}}$) belongs to $Q^{\mathcal{I}}_{\mathrm{M}}$. Let us take a variable assignment η, given by the induction hypothesis witnessing global, i.e. bullet-free, query M holding in d′. Since the query does not refer to bullet and all automata atoms from M are satisfied via a variable assignment η, then an assignment $\eta \setminus \{\bullet \mapsto \mathrm{d}'\} \cup \{\bullet \mapsto \mathrm{d}\}$ witnesses that M also holds at d.

This concludes the inspection of all cases and hence the proof of Lemma 9.7

9.3.2 Proof of Lemma 9.8

We will prove this result by induction over the *spread* of the match for M, *i.e.* the number of variables (excluding $\bullet$) plus the sum of the lengths of all paths realizing the query atoms. We assume that there is a match η witnessing that a partial query M holds tightly at d and we want to show $\mathrm{d} \in Q^{\mathcal{I}}_{\mathrm{M}}$.
We start with the cases where M does not contain variables (other than $\bullet$).

1. $\mathrm{M} = \{\mathcal{A}^{\pm}_{\mathsf{q}_1,\mathsf{q}_2}(\bullet, \mathsf{o})\}$. Then the spread of M is simply the length of the path from d to $\mathsf{o}^{\mathcal{I}}$.
   a) The spread is zero. Then $\mathrm{d} = \mathsf{o}^{\mathcal{I}}$ and $\mathsf{q}_1 = \mathsf{q}_2$ (as our automata do not have epsilon transitions), thus the membership of d in $Q^{\mathcal{I}}_{\mathrm{M}}$ is guaranteed by Rule 9.1.
   b) The spread is positive. Since there is a match of M at d, there is a non-empty path ρ from d to $\mathsf{o}^{\mathcal{I}}$, realising a transition $\mathsf{q}_1 \to^{\rho} \mathsf{q}_2$ in $\mathcal{A}^{\pm}$. Thus, there must be a role r and an r-successor d' of d, such that ρ corresponds to the path $\mathrm{d} \leadsto^{r} \mathrm{d}' \leadsto^{\rho'} \mathsf{o}$. Hence there exists a state q' and a transition of the form $(\mathsf{q}_1, r, \mathsf{q}')$ in $\mathcal{A}^{\pm}$ as well as a match of a partial query $\mathrm{M}' = \{\mathcal{A}^{\pm}_{\mathsf{q}',\mathsf{q}_2}(\bullet, \mathsf{o})\}$ in d', guaranteed by the existence of the path ρ'. Since the spread of the match for M' is smaller than the spread for M, we can use the induction hypothesis and infer that $\mathrm{d}' \in Q^{\mathcal{I}}_{\mathrm{M}'}$. Finally, by applying Rule 9.2 we can conclude that $\mathrm{d} \in Q^{\mathcal{I}}_{\mathrm{M}}$.
2. $\mathrm{M} = \{\mathcal{A}_{\mathsf{q}_1,\mathsf{q}_2}(\bullet, \bullet)\}$, i.e, the partial query M represents a round-trip. We represent
   a) The spread is zero. This is only possible when $\mathsf{q}_1 = \mathsf{q}_2$. Thus the concept membership $\mathrm{d} \in Q^{\mathcal{I}}_{\mathrm{M}}$ follows trivially from Rule 9.3.
   b) The spread is one. Then a path ρ from d to d realising a match must be a self-loop wrt. some role r such that $\mathcal{A}$ has a transition $(\mathsf{q}_1, r, \mathsf{q}_2)$. Thus the concept membership $\mathrm{d} \in Q^{\mathcal{I}}_{\mathrm{M}}$ follows immediately from Rule 9.6.
   c) The spread is greater than one. Let ρ be the corresponding path from d to d realising a match. We consider three cases separately:
      i. d occurs as an intermediate element on the path, i.e. ρ corresponds to $\mathrm{d} \leadsto^{\rho'} \mathrm{d} \leadsto^{\rho''} \mathrm{d}$. Then there exists a state q' in $\mathcal{A}$ such that $\mathcal{A}$ realises state transitions $\mathsf{q}_1 \to^{\rho'} \mathsf{q}'$ and $\mathsf{q}' \to^{\rho''} \mathsf{q}_2$. Consequently, the partial queries $\{\mathcal{A}_{\mathsf{q}_1,\mathsf{q}'}(\bullet, \bullet)\}$ and $\{\mathcal{A}_{\mathsf{q}',\mathsf{q}_2}(\bullet, \bullet)\}$ hold at d. Since both have smaller spread than M, we can invoke the induction hypothesis and conclude $\mathrm{d} \in Q^{\mathcal{I}}_{\{\mathcal{A}_{\mathsf{q}_1,\mathsf{q}'}(\bullet,\bullet)\}}$ as well as $\mathrm{d} \in Q^{\mathcal{I}}_{\{\mathcal{A}_{\mathsf{q}',\mathsf{q}_2}(\bullet,\bullet)\}}$. Then $\mathrm{d} \in Q^{\mathcal{I}}_{\mathrm{M}}$ follows via Rule 9.4.
      ii. Some $\mathsf{o}^{\mathcal{I}}$ occurs as an intermediate element on the path, i.e. ρ corresponds to $\mathrm{d} \leadsto^{\rho'} \mathsf{o}^{\mathcal{I}} \leadsto^{\rho''} \mathrm{d}$. Then there exists a state q' in $\mathcal{A}$ such that $\mathcal{A}$ realises state transitions $\mathsf{q}_1 \to^{\rho'} \mathsf{q}'$ and $\mathsf{q}' \to^{\rho''} \mathsf{q}_2$. Consequently, the partial queries $\{\mathcal{A}_{\mathsf{q}_1,\mathsf{q}'}(\bullet, \mathsf{o})\}$ and $\{\mathcal{A}^{-}_{\mathsf{q}_2,\mathsf{q}'}(\bullet, \mathsf{o})\}$ hold at d. Since both have smaller spread than M, we can invoke the induction hypothesis and conclude $\mathrm{d} \in Q^{\mathcal{I}}_{\{\mathcal{A}_{\mathsf{q}_1,\mathsf{q}'}(\bullet,\mathsf{o})\}}$ as well as $\mathrm{d} \in Q^{\mathcal{I}}_{\{\mathcal{A}^{-}_{\mathsf{q}_2,\mathsf{q}'}(\bullet,\mathsf{o})\}}$. Then $\mathrm{d} \in Q^{\mathcal{I}}_{\mathrm{M}}$ follows via Rule 9.5.
      iii. None of the above cases applies. Since the length of the path ρ is at least two, we can divide it into the following pieces (where the path ρ' could be possibly empty): $\mathrm{d} \leadsto^{r} \mathrm{d}'' \leadsto^{\rho'} \mathrm{d}' \leadsto^{r} \mathrm{d}$. Assume that automaton $\mathcal{A}$, starting in q_1 enters the state q' after reading r and then progresses to q'' after reading ρ', *i.e.* $\mathcal{A}$ realises the transitions $\mathsf{q}_1 \to^{r} \mathsf{q}'$, $\mathsf{q}' \to^{\rho'} \mathsf{q}''$, and $\mathsf{q}'' \to^{r} \mathsf{q}_2$. Note that there are no named individuals on the path ρ, hence (since $\mathcal{I}$ is a quasi-forest model by assumption) all the domain elements on the path are inside the same tree-part of the model. As ρ traverses the tree without crossing d in between, it follows that $\mathrm{d}'' = \mathrm{d}'$. Consequently, $\{\mathcal{A}_{\mathsf{q}',\mathsf{q}''}(\bullet, \bullet)\}$ holds in d', therefore, by induction hypothesis, $\mathrm{d}' \in Q^{\mathcal{I}}_{\{\mathcal{A}_{\mathsf{q}',\mathsf{q}''}(\bullet,\bullet)\}}$. On the other hand, $(\mathrm{d}, \mathrm{d}') \in (r \cap r^{-})^{\mathcal{I}}$, hence Rule 9.7 yields $\mathrm{d} \in Q^{\mathcal{I}}_{\mathrm{M}}$.

Now we consider a partial query M over variables V with V non-empty. Recall that η is a match witnessing that M holds tightly in d. For each query atom At from M we fix a path ρ_{At} in $\mathcal{I}$ that realises At under η, by providing a state transition from the initial $\mathsf{q}^{\mathrm{i}}_{At}$ to the final $\mathsf{q}^{\mathrm{f}}_{At}$. In case ρ_{At} contains named individuals, we fix one such $\mathsf{o}^{\mathcal{I}}_{At}$ and denote the state of $\mathcal{A}$ associated with $\mathsf{o}^{\mathcal{I}}_{At}$ by $\mathsf{q}^{\mathrm{m}}_{At}$. We consider two main cases:

1. V can be divided into two disjoint non-empty subsets V_1 and V_2, such that
   - all variables X occurring in some atom $\mathcal{A}^{\pm}_{\mathsf{q},\mathsf{q}'}(x, \bullet) \in \mathrm{M}$ must occur jointly in V_1, and

- for every atom $At = \mathcal{A}(x, y)$ with X and y from different V_i, the corresponding path ρ_{At} contains some named individuals.

In this case, we are going to divide M into two smaller partial queries M_1 and M_2, both holding tightly in d with smaller spread, such that $\mathrm{M} = \mathrm{M}_1 \bowtie \mathrm{M}_2$. Then by induction hypothesis we can infer that both $\mathrm{d} \in Q^{\mathcal{I}}_{\mathrm{M}_1}$ and $\mathrm{d} \in Q^{\mathcal{I}}_{\mathrm{M}_2}$ and, by applying Rule 9.11, conclude that $\mathrm{d} \in Q_{\mathrm{M}}$. It remains to find M_1 and M_2. To this end, let M_i contain

- all atoms from M containing variables only from V_i (including possibly $\bullet$),
- for every $At = \mathcal{A}(x, y) \in \mathrm{M}$ with $x \in V_i$ and $y \notin V_i$ the atom $\mathcal{A}_{\mathsf{q}^{\mathrm{i}}_{At},\mathsf{q}^{\mathrm{m}}_{At}}(x, \circ_{At})$, and
- for every $At = \mathcal{A}(x, y) \in \mathrm{M}$ with $y \in V_i$ and $x \notin V_i$ the atom $\mathcal{A}^{-}_{\mathsf{q}^{\mathrm{f}}_{At},\mathsf{q}^{\mathrm{m}}_{At}}(y, \circ_{At})$.

It can now be readily checked that $\mathrm{M} = \mathrm{M}_1 \bowtie \mathrm{M}_2$ and, by assumption, M_1 and M_2 hold tightly in d with a smaller spread than M. This concludes this case.

2. Such a division of variables is not possible. Then η maps all the variables (and, if M is local, also $\bullet$) into the same tree-shaped part of the quasi-forest model $\mathcal{I}$. To complete the proof, we will make a case distinction depending on whether a query M is global or local.

   a) M is global, *i.e.* $\bullet$-free. We are going to show that $\mathrm{d}' \in Q^{\mathcal{I}}_{\mathrm{M}}$ for some specific d', from which, due to "globality", $\mathrm{d} \in Q^{\mathcal{I}}_{\mathrm{M}}$ for (any) d follows via Rule 9.12. To this end, pick one $x \in V$ and chose $\mathrm{d}' = \eta(x)$. To show $\mathrm{d}' \in Q^{\mathcal{I}}_{\mathrm{M}}$, we will distinguish two cases:

      i. $|V| = 1$. Then M is a monodic query match and (due to the absence of $\bullet$) it is also original. We will proceed to show that $\mathrm{d}' \in (\bigsqcap \mathrm{Prec}_{\mathrm{M}})^{\mathcal{I}}$, then $\mathrm{d}' \in Q^{\mathcal{I}}_{\mathrm{M}}$ follows via Rule 9.8. For atoms of the form $\mathcal{A}^{\pm}_{\mathsf{q},\mathsf{q}'}(x, \circ)$ from M, the fact that M holds at d' entails that $\{\mathcal{A}^{\pm}_{\mathsf{q},\mathsf{q}'}(\bullet, \circ)\}$ holds at d' as well (with a smaller spread). Thus by induction hypothesis we infer $\mathrm{d}' \in Q^{\mathcal{I}}_{\{\mathcal{A}^{\pm}_{\mathsf{q},\mathsf{q}'}(\bullet,\circ)\}}$. For atoms $\mathcal{A}(x, x)$ from M, realisation of $\mathcal{A}(x, x)$ means that there exists an initial state q, a final state q' such that a $\mathcal{A}_{\mathsf{q},\mathsf{q}'}(x, x)$ is also realized. Thus $\{\mathcal{A}_{\mathsf{q},\mathsf{q}'}(\bullet, \bullet)\}$ has a match at $\eta(x) = \mathrm{d}'$. By the induction hypothesis we infer $\mathrm{d}' \in Q^{\mathcal{I}}_{\{\mathcal{A}_{\mathsf{q},\mathsf{q}'}(\bullet,\bullet)\}}$.

      ii. $|V| > 1$. In this case we will provide M_1 and M_2, both holding tightly in d' with a smaller spread than M, such that $\mathrm{M} = \mathrm{M}_1 \bowtie \mathrm{M}_2$, so the claim follows from the induction hypothesis via Rule 9.11. We let M_1 contain
         - all atoms from M containing no variables but X,
         - for every $y \neq x$ and $At = \mathcal{A}(x, y) \in \mathrm{M}$ with a nominal on ρ_{At}, the atom $\mathcal{A}_{\mathsf{q}^{\mathrm{i}}_{At},\mathsf{q}^{\mathrm{m}}_{At}}(x, \circ_{At})$,
         - for every $y \neq x$ and $At = \mathcal{A}(y, x) \in \mathrm{M}$ with a nominal on ρ_{At}, the atom $\mathcal{A}^{-}_{\mathsf{q}^{\mathrm{f}}_{At},\mathsf{q}^{\mathrm{m}}_{At}}(x, \circ_{At})$,
         - for every $y \neq x$ and $At = \mathcal{A}(x, y) \in \mathrm{M}$ with no nominal on ρ_{At}, the atom $\mathcal{A}_{\mathsf{q}^{\mathrm{i}}_{At},\mathsf{q}^{\mathrm{i}}_{At}}(x, \bullet)$,
         - for every $y \neq x$ and $At = \mathcal{A}(y, x) \in \mathrm{M}$ with no nominal on ρ_{At}, the atom $\mathcal{A}^{-}_{\mathsf{q}^{\mathrm{f}}_{At},\mathsf{q}^{\mathrm{f}}_{At}}(x, \bullet)$.

         Moreover, we let M_2 contain
         - all atoms from M not containing the variable X,
         - for every $y \neq x$ and $At = \mathcal{A}(x, y) \in \mathrm{M}$ with a nominal on ρ_{At}, the atom $\mathcal{A}^{-}_{\mathsf{q}^{\mathrm{f}}_{At},\mathsf{q}^{\mathrm{m}}_{At}}(y, \circ_{At})$,
         - for every $y \neq x$ and $At = \mathcal{A}(y, x) \in \mathrm{M}$ with a nominal on ρ_{At}, the atom $\mathcal{A}_{\mathsf{q}^{\mathrm{i}}_{At},\mathsf{q}^{\mathrm{m}}_{At}}(y, \circ_{At})$,
         - for every $y \neq x$ and $At = \mathcal{A}(x, y) \in \mathrm{M}$ with no nominal on ρ_{At}, the atom $\mathcal{A}^{-}_{\mathsf{q}^{\mathrm{f}}_{At},\mathsf{q}^{\mathrm{i}}_{At}}(y, \bullet)$,
         - for every $y \neq x$ and $At = \mathcal{A}(y, x) \in \mathrm{M}$ with no nominal on ρ_{At}, the atom $\mathcal{A}_{\mathsf{q}^{\mathrm{i}}_{At},\mathsf{q}^{\mathrm{f}}_{At}}(y, \bullet)$.

         It can now be readily checked that $\mathrm{M} = \mathrm{M}_1 \bowtie \mathrm{M}_2$ and, by assumption, M_1 and M_2 hold tightly in d' with a smaller spread than M. This concludes this case.

   b) M is local, *i.e.* it contains $\bullet$. Then we need to consider the following cases:

      i. There exists an atom $At = \mathcal{A}^{\pm}_{\mathsf{q}_1,\mathsf{q}_2}(x, \bullet) \in \mathrm{M}$ such that the corresponding path ρ_{At} crosses d at least twice, *i.e.* ρ_{At} has the shape $\eta(x) \rightsquigarrow^{\rho'} \mathrm{d} \rightsquigarrow^{\rho''} \mathrm{d}$ with ρ'' non-empty. Let q' be the state of $\mathcal{A}^{\pm}_{\mathsf{q}_1,\mathsf{q}_2}$ associated with the intermediate occurrence of d (*i.e.* after reading ρ'). Then the partial round-trip query $\{\mathcal{A}^{\pm}_{\mathsf{q}',\mathsf{q}_2}(\bullet, \bullet)\}$ holds in d and therefore, by induction hypothesis $\mathrm{d} \in Q^{\mathcal{I}}_{\{\mathcal{A}^{\pm}_{\mathsf{q}',\mathsf{q}_2}(\bullet,\bullet)\}}$. On the other hand, thanks to ρ' we know that $\mathcal{A}^{\pm}_{\mathsf{q}_1,\mathsf{q}'}(x, \bullet)\}$

is realized under η and therefore also $\mathrm{M}' = \mathrm{M} \setminus \{\mathcal{A}^{\pm}_{\mathsf{q}_1,\mathsf{q}_2}(x,\bullet)\} \cup \mathcal{A}^{\pm}_{\mathsf{q}_1,\mathsf{q}'}(x,\bullet)\}$ holds at d. Obviously, M' has smaller spread than M so by induction hypothesis $\mathrm{d} \in Q^{\mathcal{I}_{\mathrm{M}'}}$. But from $\mathrm{d} \in Q^{\mathcal{I}_{M'}}$ and $\mathrm{d} \in Q^{\mathcal{I}}_{\{\mathcal{A}^{\pm}_{\mathsf{q}',\mathsf{q}_2}(\bullet,\bullet)\}}$ follows $\mathrm{d} \in Q^{\mathcal{I}}_{M}$ via Rule 9.9.

ii. No atom of the form $At = \mathcal{A}^{\pm}_{\mathsf{q}_1,\mathsf{q}_2}(x,\bullet) \in \mathrm{M}$ is realised by a path with a premature occurrence of d.

A. there exists an $x \in V$ with $\mathrm{d} = \eta(x)$. We distinguish two cases:

(α) $|V| = 1$. Then M must be an original monodic partial query (since all atoms $\mathcal{A}^{\pm}_{\mathsf{q}_1,\mathsf{q}_2}(x,\bullet)$ must be realised by empty paths by assumption, from which follows $\mathsf{q}_1 = \mathsf{q}_2$). We will proceed to show that $\mathrm{d} \in (\bigsqcap \mathrm{Prec}_M)^{\mathcal{I}}$, then $\mathrm{d} \in Q^{\mathcal{I}}_{\mathrm{M}}$ follows via Rule 9.8. For atoms of the form $\mathcal{A}^{\pm}_{\mathsf{q},\mathsf{q}'}(x,\mathsf{o})$ from M, the fact that M holds at d entails that $\{\mathcal{A}^{\pm}_{\mathsf{q},\mathsf{q}'}(\bullet,\mathsf{o})\}$ holds at d as well (with a smaller spread). Thus by induction hypothesis we infer $\mathrm{d} \in Q^{\mathcal{I}}_{\{\mathcal{A}^{\pm}_{\mathsf{q},\mathsf{q}'}(\bullet,\mathsf{o})\}}$. For atoms $\mathcal{A}(x,x)$ from M, realisation of $\mathcal{A}(x,x)$ means that there exists an initial state q, a final state q' such that a $\mathcal{A}_{\mathsf{q},\mathsf{q}'}(x,x)$ is also realized. Thus $\{\mathcal{A}_{\mathsf{q},\mathsf{q}'}(\bullet,\bullet)\}$ has a match at $\eta(x) = \mathrm{d}$. By the induction hypothesis we infer $\mathrm{d} \in Q^{\mathcal{I}}_{\{\mathcal{A}_{\mathsf{q},\mathsf{q}'}(\bullet,\bullet)\}}$.

(β) $|V| > 1$. In this case we will provide M_1 and M_2, both holding tightly in d with a smaller spread than M, such that $\mathrm{M} = \mathrm{M}_1 \bowtie \mathrm{M}_2$, so the claim follows from the induction hypothesis via Rule 9.11. We let M_1 contain

– all atoms from M containing no variables but X,

– for every $y \neq x$ and $At = \mathcal{A}(x,y) \in \mathrm{M}$ with a nominal on ρ_{At}, the atom $\mathcal{A}_{\mathsf{q}^{\mathrm{i}}_{At},\mathsf{q}^{\mathrm{m}}_{At}}(x,\mathsf{o}_{At})$,

– for every $y \neq x$ and $At = \mathcal{A}(y,x) \in \mathrm{M}$ with a nominal on ρ_{At}, the atom $\mathcal{A}^{-}_{\mathsf{q}^{\mathrm{f}}_{At},\mathsf{q}^{\mathrm{m}}_{At}}(x,\mathsf{o}_{At})$,

– for every $y \neq x$ and $At = \mathcal{A}(x,y) \in \mathrm{M}$ with no nominal on ρ_{At}, the atom $\mathcal{A}_{\mathsf{q}^{\mathrm{i}}_{At},\mathsf{q}^{\mathrm{i}}_{At}}(x,\bullet)$

– for every $y \neq x$ and $At = \mathcal{A}(y,x) \in \mathrm{M}$ with no nominal on ρ_{At}, the atom $\mathcal{A}^{-}_{\mathsf{q}^{\mathrm{f}}_{At},\mathsf{q}^{\mathrm{f}}_{At}}(x,\bullet)$.

Moreover, we let M_2 contain

– all atoms from M not containing the variable X,

– for every $y \neq x$ and $At = \mathcal{A}(x,y) \in \mathrm{M}$ with a nominal on ρ_{At}, the atom $\mathcal{A}^{-}_{\mathsf{q}^{\mathrm{f}}_{At},\mathsf{q}^{\mathrm{m}}_{At}}(y,\mathsf{o}_{At})$,

– for every $y \neq x$ and $At = \mathcal{A}(y,x) \in \mathrm{M}$ with a nominal on ρ_{At}, the atom $\mathcal{A}_{\mathsf{q}^{\mathrm{i}}_{At},\mathsf{q}^{\mathrm{m}}_{At}}(y,\mathsf{o}_{At})$,

– for every $y \neq x$ and $At = \mathcal{A}(x,y) \in \mathrm{M}$ with no nominal on ρ_{At}, the atom $\mathcal{A}^{-}_{\mathsf{q}^{\mathrm{f}}_{At},\mathsf{q}^{\mathrm{i}}_{At}}(y,\bullet)$,

– for every $y \neq x$ and $At = \mathcal{A}(y,x) \in \mathrm{M}$ with no nominal on ρ_{At}, the atom $\mathcal{A}_{\mathsf{q}^{\mathrm{i}}_{At},\mathsf{q}^{\mathrm{f}}_{At}}(y,\bullet)$.

It can now be readily checked that $\mathrm{M} = \mathrm{M}_1 \bowtie \mathrm{M}_2$ and, by assumption, M_1 and M_2 hold tightly in d with its spread smaller than M. This concludes this case.

B. there are no $x \in V$ with $\mathrm{d} = \eta(x)$. Recall that $\mathrm{d} = \eta(\bullet)$ as well as all $\eta(x)$ with $x \in V$ occur in the same tree part of the quasi-forest model $\mathcal{I}$. Consider the minimal subtree Υ containing all these domain elements, oriented such that d is the root. Consider cases:

(α) d has exactly one child d'. Note that for any atom $At = \mathcal{A}^{\pm}_{\mathsf{q},\mathsf{q}'}(x,\bullet) \in \mathrm{M}$, the corresponding path ρ_{At} is free of named individuals by assumption and has no premature occurrence of d. Thus, any such paths reaching $\eta(\bullet)$ from some $\eta(x)$ must have the same domain element d' as the penultimate node. This means that whenever we have an atom of the form $At = \mathcal{A}^{\pm}_{\mathsf{q},\mathsf{q}'}(x,\bullet)$ the path ρ_{At} realising it can be written as $\eta(x) \rightsquigarrow^{\rho'} \mathrm{d}' \rightsquigarrow^{r_{At}} \mathrm{d}$ for some role r_{At} where $\mathcal{A}^{\pm}$ realises state transitions $\mathsf{q} \rightarrow^{\rho'} \mathsf{q}''$ and $\mathsf{q}'' \rightarrow^{r_{At}} \mathsf{q}'$. Let M' be the set of atoms obtained from M by replacing each atom of the form $\mathcal{A}^{\pm}_{\mathsf{q},\mathsf{q}'}(x,\bullet)$ with $\mathcal{A}^{\pm}_{\mathsf{q},\mathsf{q}''}(x,\bullet)$. Thus M' holds tightly at d'. Since its spread is smaller that the spread for M, we invoke the induction hypothesis and infer $\mathrm{d}' \in Q^{\mathcal{I}}_{\mathrm{M}'}$. By using Rule 9.10 we conclude $\mathrm{d} \in Q^{\mathcal{I}}_{\mathrm{M}}$.

(β) d has more than one child. Then pick one child d' and let V_1 contain all variables from V which are mapped to d' or one of its descendants. Let $V_2 = V \setminus V_1$. Note that by construction, for any atom $At = \mathcal{A}(x,y) \in \mathrm{M}$ with $x \in V_i$ and $y \notin V_i$, the corresponding path ρ_{At} must either contain some named individual $\mathsf{o}^{\mathcal{I}}_{At}$ (with

$\mathtt{q}^{\mathrm{i}}_{At}$,$\mathtt{q}^{\mathrm{m}}_{At}$, and $\mathtt{q}^{\mathrm{f}}_{At}$ as defined above) or, otherwise it must contain d. In the latter case, let $\mathtt{q}^{\mathrm{i}}_{At}$,$\mathtt{q}^{\mathrm{m}}_{At}$, and $\mathtt{q}^{\mathrm{f}}_{At}$ be defined as for the named-individual-crossing paths, just with $\mathtt{q}^{\mathrm{m}}_{At}$ being the state associated with d.

We will now provide M_1 and M_2, both holding tightly in d with its spread smaller than M, such that $\mathrm{M} = \mathrm{M}_1 \bowtie \mathrm{M}_2$, so the claim follows from the induction hypothesis via Rule 9.11. For $i \in \{1, 2\}$, we let M_i contain

- all atoms from M containing only variables from V_i,
- for every $x \in V_i$ and $y \notin V_i$ with $At = \mathcal{A}(x, y) \in \mathrm{M}$ and a named individual on ρ_{At}, the atom $\mathcal{A}_{\mathtt{q}^{\mathrm{i}}_{At},\mathtt{q}^{\mathrm{m}}_{At}}(x, \circ_{At})$,
- for every $x \in V_i$ and $y \notin V_i$ with $At = \mathcal{A}(y, x) \in \mathrm{M}$ and a named individual on ρ_{At}, the atom $\mathcal{A}^{-}_{\mathtt{q}^{\mathrm{f}}_{At},\mathtt{q}^{\mathrm{m}}_{At}}(x, \circ_{At})$,
- for every $x \in V_i$ and $y \notin V_i$ with $At = \mathcal{A}(x, y) \in \mathrm{M}$ and no named individual on ρ_{At}, the atom $\mathcal{A}_{\mathtt{q}^{\mathrm{i}}_{At},\mathtt{q}^{\mathrm{m}}_{At}}(x, \bullet)$, and
- for every $x \in V_i$ and $y \notin V_i$ with $At = \mathcal{A}(y, x) \in \mathrm{M}$ and no named individual on ρ_{At}, the atom $\mathcal{A}^{-}_{\mathtt{q}^{\mathrm{f}}_{At},\mathtt{q}^{\mathrm{m}}_{At}}(x, \bullet)$.

It can now be readily checked that $\mathrm{M} = \mathrm{M}_1 \bowtie \mathrm{M}_2$ and, by assumption, M_1 and M_2 hold tightly in d with smaller spread than M. This concludes this case.

This finishes all cases, and therefore the proof of Lemma 9.8.

9.4 Conclusions and Extra Results

In this chapter we have established tight complexity bounds for expressive querying in very expressive DLs under the assumption of succinct (*i.e.* binary) encoding of number restrictions. Arguing along the lines of the original work of Calvanese et al. [CEO09], we can leverage our findings to strengthen their results on query containment as well as the $\mathcal{SR}$ family of DLs as follows:

Theorem 9.13

One can decide in time doubly-exponential w.r.t. the input KB $\mathcal{K}$ and queries q, q' whether q' is contained by q modulo $\mathcal{K}$ (*i.e.* whether every model of $\mathcal{K}$ satisfying q' also satisfies q) whenever:

(i) $\mathcal{K}$ is written in $\mathcal{ZOI}$ or $\mathcal{ZOQ}$, and the queries q, q' are P2RPQs.

(ii) $\mathcal{K}$ is written in $\mathcal{ZIQ}$, the query q is a P2RPQ, and q' is a CQ.

The complexity drops to a single exponential time if the number of atoms occurring in q is bounded by a constant.

Proof. It follows by a reduction from the containment problem to the satisfiability problem, stated right above Theorem 4.3 in the original work on $\mathcal{ZOIQ}$ by Calvanese et. al [CEO09]. □

As already discussed in Section 7.8, the decidable expressive logics from the $\mathcal{SR}$ family, the logical core of OWL2, can be equivalently rewritten into the $\mathcal{Z}$ family [CEO09, Prop. 5.1]. As the rewriting yields an exponential blow-up, we conclude the following theorem:

Theorem 9.14

For KBs $\mathcal{K}$ written in $\mathcal{SRIQ}$, $\mathcal{SROQ}$ or $\mathcal{SROI}$, and P2RPQs q we can decide whether $\mathcal{K} \models q$ in triply-exponential time w.r.t. $|\mathcal{K}|+|q|$.

Finally, our results apply to the query entailment problem for the two-variable guarded fragment extended with counting quantifiers, via a relatively easy translation [BR19, Sec. 6] into the DL $\mathcal{ZIQ}$.

There are plenty of open questions left for future work. One research direction is to lift our theorems, stated in terms of combined complexity, to the case of data complexity. We are confident that we have a solution to do so, and plan to write and publish the result after a successful defence of this dissertation. Another, very ambitious task is to try to establish the decidability of the entailment problem in case of the finite model reasoning. Some serious progress was made by Gutowski et al. [GGIM22] who proved the *finite* entailment of conjunctive regular path queries for $\mathcal{ALC}$ is 2ExpTime-complete. Lifting their results even to the case of plain $\mathcal{Z}$ seems to be very difficult. Finally, one can move to the more expressive classes of query languages. While the extension of our query formalism by nesting [BCOv14] seems straightforward without impacting complexities, our technique seems not readily extendable to capture more elaborate query languages like Monadically Defined Queries [RK13] or Regular Queries [RRV17].

10

Finite Controllability in the $\mathcal{Z}$ family of DLs

Contents

Motivation and Our Contribution

As mentioned at the end of the previous chapter (as well as in the Preliminaries), dealing with the query entailment problem under the finite model semantics is a challenging task. The main obstacle is the fact that most of the existing techniques are automata-based and inherently rely on the presence of infinite forest-like models of knowledge-bases. Thus, reasoning about finite models requires a completely different set of techniques. Unfortunately, we currently lack the toolkit for attacking these kinds of problems. We hence lower our expectations a bit, and aim for a more modest goal. In contrast to the previous chapter, we will be interested in the entailment of positive existential queries (*i.e.* we drop Kleene's star from the syntax of P2RPQs). Such queries are simply positive boolean combinations of conjunctive queries, still very relevant in DLs as they are a premier query language for reasoning with classical, relational databases. Under this setting, we would like to obtain tight results on the *finite* entailment problem for the most expressive members of the $\mathcal{Z}$ family, namely $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, and $\mathcal{ZOI}$. As we already solved the case $\mathcal{ZIQ}$ in the previous chapters (consult Section 5.4 if needed), it suffices to work with $\mathcal{ZOQ}$ and $\mathcal{ZOI}$. It turns out that (except for our result from the previous chapters) nearly nothing is known about finite model reasoning in the $\mathcal{Z}$ family of DLs. The notable exception is the relatively recent work of Calvanese et al. [COv16, Thm. 6], in which the authors established the finite model property of $\mathcal{ZOI}$, *i.e.* that every satisfiable $\mathcal{ZOI}$-KB also has a finite model. Thus the finite and unrestricted satisfiability problems for $\mathcal{ZOI}$ coincide. Regarding the finite query entailment problem, to the best of our knowledge, nothing was known even about plain $\mathcal{Z}$.

Our contribution. In this chapter we show that both $\mathcal{ZOQ}$ and $\mathcal{ZOI}$ are *finitely controllable*, *i.e.* for any $\mathcal{ZOQ}$-KB or $\mathcal{ZOI}$-KB $\mathcal{K}$ and any PEQ q it holds that $\mathcal{K}$ entails q if and only if $\mathcal{K}$ finitely entails q. Alternatively, we can say that the existence of a countermodel for $\mathcal{K}$ and q coincides with the existence of a *finite* countermodel for $\mathcal{K}$ and q. This allows one to reuse the existing algorithms for the previous section and infer that the finite PEQ entailment problem for $\mathcal{ZOI}$ and $\mathcal{ZOQ}$ are 2ExpTime-complete. Similarly to the previous section, our results also transfer to the expressive members of the $\mathcal{SR}$ family of DLs (assuming that non-simple roles do not appear in queries) as well as to the query containment problem. Our proof is based on an intricate finite model construction which starts from an (infinite) quasi-forest model, distinguishes in this model some finite pattern fragments, and carefully forms a new finite model out of some finite number of copies of those fragments. The construction itself resembles the construction already used in Section 5.2 (in fact, the construction from Section 5.2 is an adaptation of the forthcoming construction).

Overview of the Chapter and Prerequisites

We assume that the reader is familiar with the notion of quasi-forest models from Section 7.1, the syntax and semantics of $\mathcal{ZOIQ}$ from Section 2.10, as well as the section concerning automata (Section 2.6), decision problems (Section 2.5), and "local" queries (Section 2.4) from the Preliminaries.

As we establish a single result here, the whole chapter is composed of a single section. In Section 10.1.1 we introduce certain simplifications of KBs, queries and models. We also define the relevant notions of types. In Section 10.1.2 we introduce the vital notion of components. We employ components in Section 10.1.3 to construct a finite structure $\mathcal{J}$ out of the infinite countermodel $\mathcal{I}$ for a KB $\mathcal{K}$ and a query q. In Section 10.1.4 we show that $\mathcal{J} \models \mathcal{K}$. In Section 10.1.5. Concerning the content of this chapter, there are no immediate open problems to solve.

10.1 Finite Controllability of $\mathcal{ZOQ}$ and $\mathcal{ZOI}$

The whole chapter is devoted to a single proof of the following theorem.

Theorem 10.1
The logics $\mathcal{ZOQ}$ and $\mathcal{ZOI}$ are finitely controllable (for the class of positive existential queries).

As an immediate corollary, by the known complexity results concerning the satisfiability and query entailment for the members of the Z family of DLs (see Lemma 7.6 and Theorem 9.12) we have:

Corollary 10.2
The finite KB satisfiability problems for $\mathcal{ZOQ}$ and $\mathcal{ZOI}$ are ExpTime-complete, while their finite PEQ entailment problems are 2ExpTime-complete (even when the numbers in number restrictions are encoded in binary).

In DB theory, one of the most prominent reasoning problems is query containment, which received also some attention from the DL community [COv11, BLW12]. Arguing along the lines of [CEO09] we can lift our findings to *finite* query containment problem. Missing definitions are in [CEO09]. Up to our knowledge it is the first result on *finite containment* of regular queries in the local ones, in the DL setting.

10.1.1 Warming Up

We next proceed with the proof of Theorem 10.1. Until the end of this chapter we fix a satisfiable KB $\mathcal{K} := (\mathcal{A}, \mathcal{T})$ written either in $\mathcal{ZOQ}$ or $\mathcal{ZOI}$ KB, a positive existential query q and an interpretation $\mathcal{I}$ being a countermodel for $\mathcal{K}$ and q, *i.e.* $\mathcal{I} \models \mathcal{K}$ but $\mathcal{I} \not\models q$.

Simplifications In what follows we introduce a few simplifications concerning $\mathcal{K}$, q, and $\mathcal{I}$.

(I) First, we assume that $\mathcal{K}$ has the empty ABox. This is w.l.o.g. as all statements from the ABox can be internalised in TBox in the presence of nominals. Moreover, we assume that the TBox contains all the concept and role names that appear in the query q (they can be inserted to $\mathcal{T}$ in some dummy way, if necessary). Finally, by Lemma 2.26 we assume that $\mathcal{T}$ is in Scott's normal form.

(II) We assume that for every GCI $\mathrm{A} \equiv \exists \mathcal{A}.\top$ from $\mathcal{K}$ and every state q of $\mathcal{A}$, the KB $\mathcal{K}$ contains the GCI $\mathrm{A}_{\mathsf{q}} \equiv \exists \mathcal{A}_{\mathsf{q}}.\top$, where A_{q} is a fresh concept name, and $\mathcal{A}_{\mathsf{q}}$ is the NFA obtained by changing the initial state of $\mathcal{A}$ to q.

(III) By rewriting q into DNF we can assume w.l.o.g. that q is a UCQ $q \coloneqq \bigvee_i q_i$. We let K be the maximal number of variables across the queries q_i.

(IV) Finally, we enforce some additional conditions on $\mathcal{I}$. As the first step, we can assume by Lemma 7.5 that $\mathcal{I}$ is a quasi-forest model that is N-bounded by some positive integer N. Moreover, we assume that $\mathcal{I}$ downward realise all the automata present in $\mathcal{K}$. We always assume that if the sequence $\mathrm{d} \cdot n$ belongs to $\Delta^{\mathcal{I}}$ for some $n \in \mathbb{N}$ then for all positive integers $n' < n$ we have $d \cdot n' \in \Delta^{\mathcal{I}}$. This can be achieved with a simple renaming, and allow us to refer to $\mathrm{d}{\cdot}n$ as the *n-th child* of d. Moreover, w.l.o.g. we assume that $\mathcal{I}$ interprets all the concept and role names that do not appear in $\mathcal{K}$ as $\emptyset$.

When reasoning about trees and forest we employ the following terminology. A *descendant* of $\mathrm{d} \in \Delta^{\mathcal{I}}$ from $\mathcal{I}$ is any node of the form $\mathrm{d}w \in \Delta^{\mathcal{I}}$ for $w \in \mathbb{N}^{+}$. The substructure $\mathit{Subtree}_{\mathcal{I}}(\mathrm{d})$, called the **subtree of** d, is the substructure $\mathcal{I}$ induced by d and its descendants. Sometimes we also look at descendants at a certain distance from d. Hence, we denote with $\mathit{Subtree}^{\leq n}_{\mathcal{I}}(\mathrm{d})$ the substructure of $\mathcal{I}$ induced by d and all of its descendants of the form $\mathrm{d}w$ for some $w \in \mathbb{N}^{+}$ of length at most n.

We will find a finite countermodel $\mathcal{J}$ for $\mathcal{K}$ and q by distinguishing certain substructures of $\mathcal{I}$ and carefully linking together some number of their copies. The construction and its correctness proof will be nearly the same for $\mathcal{ZOQ}$ and $\mathcal{ZOI}$; we will pinpoint the minor differences when necessary. In our construction a vital role is played by various notions of types. We introduce them next.

Types Recall that given an interpretation $\mathcal{I}$ and a subset $\Delta_0 \subseteq \Delta^{\mathcal{I}}$ we denote by $\mathcal{I}\restriction_{\Delta_0}$ the **restriction** of $\mathcal{I}$ to Δ_0, that is the structure with the domain Δ_0, mapping each concept name A to $\Delta_0 \cap \mathrm{A}^{\mathcal{I}}$, each role name p to $p^{\mathcal{I}} \cap \Delta_0 \times \Delta_0$, and mapping each individual name o to $\mathsf{o}^{\mathcal{I}}$ if $\mathsf{o}^{\mathcal{I}} \in \Delta_0$, leaving $\mathsf{o}^{\mathcal{I}}$ undefined otherwise. For $\mathrm{d} \in \Delta^{\mathcal{I}}$ we define its **atomic type**, denoted $\mathit{atp}_{\mathcal{I}}(\mathrm{d})$, as the isomorphism type (as usually defined in model theory) of $\mathcal{I}\restriction_{(\{\mathrm{d}\} \cup \mathsf{Nom}_{\mathcal{I}})}$, where $\mathsf{Nom}_{\mathcal{I}} = \{\mathsf{o}^{\mathcal{I}} \mid \mathsf{o} \in \mathbf{N_I}\}$. Note that structures have equal isomorphism type if and only if they are isomorphic. For quasi-forest models we also want to know how the structure consisting of the elements being at most K steps below d looks like (recall that K is the maximal number of variables in the q_i). Formally, the **downward type** of d, written: $\mathit{dtp}_{\mathcal{I}}(\mathrm{d})$, is the isomorphism type of $\mathcal{I}\restriction_{(\mathit{Subtree}^{\leq \mathrm{K}}_{\mathcal{I}}(\mathrm{d}) \cup \mathsf{Nom}_{\mathcal{I}})}$, where We will call the downward types of the nominals the **nominal (downward) types**. We denote the sets of downward and nominal types appearing in $\mathcal{I}$ resp. with $\mathit{DTP}_{\mathcal{I}}$ and $\mathit{NTP}_{\mathcal{I}}$. The said sets are finite due to finite branching of $\mathcal{I}$ and the fact $\mathcal{I}$ interprets only finitely many concept and role names as non-empty sets.

10.1.2 Preparing building blocks

Our next step is to define *components* that will serve as basic building blocks in the construction of $\mathcal{J}$. For each downward type $\pi \in \mathit{DTP}_{\mathcal{I}}$ we fix a domain element $\mathrm{d}_\pi \in \Delta^{\mathcal{I}}$ having such a downward type. We are going to select a finite subset of $\mathit{Subtree}_{\mathcal{I}}(\mathrm{d}_\pi)$, which will be *sibling*- and *parent*-closed (up to d_π), *i.e.* for each element from such a subset either all of its children belong to it or none of them (resp., if an element d is not equal d_π then its parent belongs to the subset). Such a subset will be the domain of a building block called a called the *π-pre-component* $\mathit{PreCmp}_{\mathcal{I}}(\pi)$. We create them as follows.

Definition 10.3 Let $\pi \in \mathit{DTP}_{\mathcal{I}}$ be a downward type. A **π-pre-component** $\mathit{PreCmp}_{\mathcal{I}}(\pi)$ is the smallest set of domain elements of $\mathcal{I}$ that is sibling- and parent-closed such that:

- It contains the domain of $\mathit{Subtree}^{\leq \mathrm{K}}_{\mathcal{I}}(\mathrm{d}_\pi)$.
- For every GCI $\mathrm{A} \equiv \exists \mathcal{A}.\top$ from $\mathcal{K}$ for which $\mathrm{d}_\pi \in \mathrm{A}^{\mathcal{I}}$ there exists a downward path $\rho \coloneqq \mathrm{d}_1, \mathrm{d}_2, \ldots, \mathrm{d}_k$ realising $\mathcal{A}$ and starting from d_π with the minimal possible number of nominal elements on it. We consider two cases:
  - If ρ is nominal-free and d_k is a member of $\mathit{PreCmp}_{\mathcal{I}}(\pi)$.
  - Otherwise, for the smallest j for which d_j is a nominal, $\mathit{PreCmp}_{\mathcal{I}}(\pi)$ contains d_{j-1}.

The **component** for the downward type π is the interpretation $\mathcal{C}_\pi := \mathcal{I}\restriction_{PreCmp_{\mathcal{I}}(\pi)}$. Note that components are interpretations based on finite trees, and hence we employ the usual terminology from graph theory. In particular, we will speak about their leaves. However, d_π will be called the **origin** of $\mathcal{C}_\pi$, rather than its root, to avoid confusion with the roots of the interpretation $\mathcal{I}$.

10.1.3 Assembling a finite model $\mathcal{J}$

The interpretation $\mathcal{J}$ will be composed of a finite number of copies of the components $\mathcal{C}_\pi$, carefully connected to preserve satisfaction of $\mathcal{K}$ and non-satisfaction of q. We will use sub- and super-scripts to distinguish such copies. As expected, a unique copy of $\mathcal{C}_\pi$ for every nominal type π from $NTP_{\mathcal{I}}$ will be used—this guarantees the uniqueness of nominals. We remark that the scheme of joining components is somehow similar to the one used recently by Danielski and Kieroński [DK19] to build small finite models for finitely-satisfiable formulae in the extension of the Unary Negation Fragment.

Let L be the maximal number of leaves across all the components, and let M be the maximal number of children of a node in $\mathcal{I}$ (note that L and M are finite by design and the finite branching of $\mathcal{I}$).

The domain The domain of $\mathcal{J}$ is

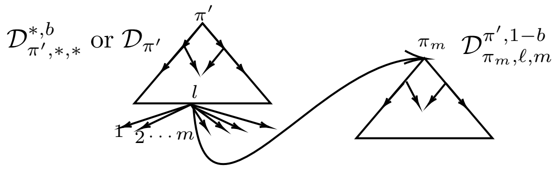
$$\Delta^{\mathcal{J}} := \bigcup_{\pi \in NTP_{\mathcal{I}}} \Delta_\pi \quad \cup \quad \bigcup \Delta^{\pi',b}_{\pi,\ell,m},$$

where the second union ranges over $\pi \in DTP_{\mathcal{I}} \setminus NTP_{\mathcal{I}}$, $\pi' \in DTP_{\mathcal{I}}$, $1{\le}\ell{\le}L$, $1{\le}m{\le}M$ and $b \in \{0,1\}$. The presence of various indices may look cryptic but we will clarify it soon, when describing the "linking process".

Concepts and roles inside a single copy The above sums are disjoint and the elements of the decorated $\Delta^{*,*}_{\pi,*,*}$ are just disjoint copies of the corresponding Δ_π. For every π, π', ℓ, m, b as above, we make $\mathcal{D}^{\pi',b}_{\pi,\ell,m} := \mathcal{J}\restriction_{\Delta^{\pi',b}_{\pi,\ell,m}}$ isomorphic to $\mathcal{C}_\pi$; similarly for every $\pi \in NTP_{\mathcal{I}}$ we make $\mathcal{D}_\pi := \mathcal{J}\restriction_{\Delta_\pi}$ isomorphic to $\mathcal{C}_\pi$. We naturally define the **pattern function** $\mathfrak{f}\colon \Delta^{\mathcal{J}} \to \Delta^{\mathcal{I}}$ that maps an element from $\Delta^{\mathcal{J}}$ to its corresponding element in $\Delta^{\mathcal{I}}$. Note that $\mathfrak{f}(\mathsf{o}^{\mathcal{J}}) = \mathsf{o}^{\mathcal{I}}$ for all $\mathsf{o} \in \mathbf{N_I}$.

Roles between copies It remains to define the roles between various components of $\mathcal{J}$.

- Connections involving nominal elements.
  Roles between $\mathrm{d} \in \Delta^{\mathcal{J}}$ and $\mathsf{o}^{\mathcal{J}} \in \mathsf{Nom}_{\mathcal{J}}$ are defined according to $\mathfrak{f}$. For all role names $r \in \mathbf{N_R}$ we put $(\mathrm{d}, \mathsf{o}^{\mathcal{J}}) \in r^{\mathcal{J}}$ if and only if $(\mathfrak{f}(\mathrm{d}), \mathfrak{f}(\mathsf{o}^{\mathcal{J}})) \in r^{\mathcal{I}}$. This way the map $\mathfrak{f}$ preserves atomic types.
- Linking different components.
  For every component, its leaves will be connected to the origins of other components, in order to provide the leaves their "local witnesses". Our strategy is as follows. For every type $\pi \in DTP_{\mathcal{I}} \setminus NTP_{\mathcal{I}}$ the origin of $\mathcal{D}^{\pi',1-b}_{\pi,\ell,m}$ will serve as the m-th child of the ℓ-th leaf of any of the components $\mathcal{D}^{*,b}_{\pi',*,*}$, or, in the case of $b = 1$ and $\pi' \in NTP_{\mathcal{I}}$, of the component $\mathcal{D}_{\pi'}$.[1] This explains our naming scheme.


$\mathcal{D}^{*,b}_{\pi',*,*}$ or $\mathcal{D}_{\pi'}$
π'
l
1
2 ⋯ m
π_m
$\mathcal{D}^{\pi',1-b}_{\pi_m,\ell,m}$

  Formally, let d' be the ℓ-th leaf of some component $\mathcal{D}^{*,b}_{\pi',*,*}$ or $\mathcal{D}_{\pi'}$ (in the latter case set $b := 1$), let $\mathrm{d}_1, \ldots, \mathrm{d}_k$ be the children of $\mathfrak{f}(\mathrm{d}')$ in $\mathcal{I}$ with downward types (in $\mathcal{I}$) $\pi_1, \ldots, \pi_k$. For all $m = 1, \ldots, k$ we link d' with the origin of $\mathcal{D}^{\pi',1-b}_{\pi_m,\ell,m}$ in the same way (*i.e.* by the same atomic roles) as $\mathfrak{f}(\mathrm{d}')$ is joined with d_m in $\mathcal{I}$. Repeat this step for all leaves and all components.
- For any pair of elements which we have not explicitly connected in the above steps, we leave it unconnected (we do not join them by any role).

This finishes the definition of $\mathcal{J}$. As the next step we establish correctness of our construction.

[1]The index b is not crucial here, but using it allows us to avoid *e.g.* the need to link leaves of a component with its origin.

10.1.4 Preservation of knowledge base satisfiability

For the correctness of our construction, we establish that our construction is modelhood preserving.

Lemma 10.4 The interpretation $\mathcal{J}$ constructed in Section 10.1.3 is a model of $\mathcal{K}$.

The rest of this section is dedicated to the proof of the above lemma. As $\mathcal{K}$ is in Scott's normal form, it suffices to consider the following cases.

1. Preservation of GCIs of the form $\mathrm{A} \equiv \{\mathsf{o}\}$, $\mathrm{A} \equiv \mathrm{B}$, $\mathrm{A} \equiv \neg\mathrm{B}$, $\mathrm{A} \equiv \mathrm{B} \sqcup \mathrm{B}'$, $\mathrm{A} \equiv \exists s.\mathsf{Self}$.
   This follows from the fact that the satisfaction of such GCIs depends on the atomic type of a domain element. In our construction we have that the atomic types of any $\mathrm{d} \in \Delta^{\mathcal{J}}$ and $\mathfrak{f}(\mathrm{d})$ are equal, and thus such GCIs are fulfilled by $\mathcal{J}$.
2. $\mathrm{A} \equiv (\geqslant n\ s).\top$. Take any element $\mathrm{d} \in \Delta^{\mathcal{J}}$, and note that it suffices to show that d and $\mathfrak{f}(\mathrm{d})$ have equal number of s-successors. As such GCIs are only available in $\mathcal{ZOQ}$, the possible s-successors of d are only its children and (possibly) nominals. Let us consider two cases.
   - If d is not a leaf of its component, then the set of its s-successors is limited only to its children and, possibly, to nominal elements. By design, $\mathcal{J}$ restricted to these elements is isomorphic to $\mathcal{I}$ restricted to $\mathfrak{f}(\mathrm{d})$, its children and the nominal elements of $\mathcal{I}$.
   - If d is a leaf then in our construction (1st item of the linking process), note that we link d with nominals in exactly the same way as $\mathfrak{f}(\mathrm{d})$ is linked to them in $\mathcal{I}$. Moreover if $\mathfrak{f}(\mathrm{d})$ has ℓ children $\mathrm{e}_1, \ldots \mathrm{e}_\ell$ in $\mathcal{I}$ then d is connected to exactly ℓ other domain elements (origins) $\mathrm{d}_1, \mathrm{d}_2, \ldots, \mathrm{d}_\ell$ in a way that $(\mathrm{d}, \mathrm{d}_i)$ are in exactly the same atomic roles in $\mathcal{J}$ as $(\mathfrak{f}(\mathrm{d}), \mathrm{e}_i)$ are in $\mathcal{I}$ — see the 2nd point of the construction.
3. $\mathcal{J} \models \mathrm{A} \sqsubseteq \exists\mathcal{A}.\mathrm{B}$ for all GCIs $\mathrm{A} \sqsubseteq \exists\mathcal{A}.\mathrm{B}$ that appear in $\mathcal{K}$.
   We first show by induction on k that for any origin e in $\mathcal{J}$ and any GCI having the form $\mathrm{C} \sqsubseteq \exists\mathcal{A}'.\mathrm{D}$ such that $\mathrm{e} \in \mathrm{C}^{\mathcal{J}}$ the following condition holds:

   "if $\mathrm{d} := \mathfrak{f}(\mathrm{e})$ has a downward path witnessing $\exists\mathcal{A}'.\mathrm{D}$ with at most k occurrences of nominals on it (not counting the first occurrence of e, in case e is nominal) then $\mathrm{e} \in (\exists\mathcal{A}'.\mathrm{D})^{\mathcal{J}}$".

   Note that d is an origin of a component in $\mathcal{I}$, and (since $\mathfrak{f}$ retains atomic types) we have $\mathrm{d} \in \mathrm{C}^{\mathcal{I}}$.

   If $k = 0$ then during the construction of the component with the origin d we chose a path starting from d and witnessing $\exists\mathcal{A}'.\mathrm{D}$ with no nominal elements on it. Such a path is then included inside the component of d, and thus its isomorphic copy is contained in the component of e. Thus $\mathrm{e} \in \mathrm{C}^{\mathcal{J}}$.

   Consider now $k > 0$. Let $\rho := \rho' {\cdot} \mathrm{d}' {\cdot} \rho''$ be the downward path for d and $\exists\mathcal{A}'.\mathrm{D}$ chosen during the construction of the component of d with d′ being the first occurrence of a nominal. An isomorphic copy ζ' of ρ' is contained in the component of e by our construction. Assume that in the run of the NFA $\mathcal{A}'$ realising ρ we have that $\mathcal{A}$ enters a state q when reading (first time) a role-letter leading to d′. Then, recalling that we appended extra GCIs to $\mathcal{K}$, we have $A'_{\mathsf{q}} \equiv \exists\mathcal{A}'_{\mathsf{q}}.\mathrm{D}$, for some concept name A'_{q} among the GCIs. Clearly, $\mathrm{d}' \in (\exists\mathcal{A}'_{\mathsf{q}}.\mathrm{D})^{\mathcal{I}}$ and thus $\mathrm{d}' \in (A'_{\mathsf{q}})^{\mathcal{I}}$. Since d′ is a nominal then there is exactly one element e′ in $\mathcal{J}$ such that $\mathfrak{f}(\mathrm{e}') = \mathrm{d}'$. Again, since $\mathfrak{f}$ retains atomic types we have $\mathrm{e}' \in (\mathrm{A}'_{\mathsf{q}})^{\mathcal{J}}$. By the inductive assumption $\mathrm{e}' \in (\exists\mathcal{A}'_{\mathsf{q}}.\mathrm{D})^{\mathcal{J}}$. Let ζ'' be a path starting from e′ and witnessing the satisfaction of $\exists\mathcal{A}'_{\mathsf{q}}.\mathrm{D}$. Note that the role-connections between the last node of ζ' and e′ are identical to the role-connections between the last node of ρ' and d′ (see item "connections involving nominal elements"), so we can finally compose a path starting from e and witnessing the satisfaction of $\exists\mathcal{A}'.\mathrm{B}$ as $\zeta'\zeta''$, so $\mathrm{e} \in (\exists\mathcal{A}'.\mathrm{D})^{\mathcal{J}}$. This finishes our inductive proof.

   Take now any e in $\mathcal{J}$ such that $\mathrm{e} \in A^{\mathcal{J}}$. We show that $\mathrm{e} \in (\exists\mathcal{A}.\mathrm{B})^{\mathcal{J}}$. If e is an origin then this follows from the property inductively proved above. If not, let $\mathcal{D}$ be the component of e, $\mathrm{d} := \mathfrak{f}(\mathrm{e})$, and $\mathcal{C}$ be the component of d. Note that $\mathcal{C}$ and $\mathcal{D}$ are isomorphic. Since $\mathfrak{f}$ retains atomic types we have that $\mathrm{d} \in \mathrm{A}^{\mathcal{J}}$. Take any downward path ρ starting from d and witnessing the satisfaction of $\exists\mathcal{A}.\mathrm{B}$. If this path is fully contained in $\mathcal{C}$ then its isomorphic copy, starting from e is contained in $\mathcal{D}$ and then $\mathrm{e} \in (\exists\mathcal{A}.\mathrm{B})^{\mathcal{J}}$. Otherwise, let as write $\rho = \rho' {\cdot} \mathrm{d}' {\cdot} \rho''$, where d′ is the first element not belonging to $\mathcal{C}$ (observe that d′ may, but need not to be nominal). Then an isomorphic copy ζ' of ρ', starting at e, belongs to $\mathcal{D}$. Assume that $\mathcal{A}$, when accepting a word corresponding to ρ and reading (for the first time) the role-letter leading to the element d' enters a state q. By our normal form we have a GCI $\mathrm{A}_{\mathsf{q}} \equiv \exists \mathcal{A}_{\mathsf{q}}.\mathrm{B}$. Clearly $\mathrm{d}' \in (\exists \mathcal{A}_{\mathsf{q}}.\mathrm{B})^{\mathcal{I}}$ and thus $\mathrm{d}' \in \mathrm{A}_{\mathsf{q}}^{\mathcal{I}}$. By our scheme of joining the components the last element of ζ' is joined to some origin e' in $\mathcal{J}$ isomorphically to how the last element of ρ' is joined to d' in $\mathcal{I}$. Moreover the downward types of d' and $\mathfrak{f}(\mathrm{e}')$ are the same. This in particular means that the atomic types of d' and e' are the same, so $\mathrm{e}' \in \mathrm{A}_{\mathsf{q}}^{\mathcal{J}}$. Recalling that e' is an origin we have already proved that $\mathrm{e}' \in (\exists \mathcal{A}_{\mathsf{q}}.\mathrm{B})^{\mathcal{J}}$. Let ζ'' be a witness path for e' and $\exists \mathcal{A}_{\mathsf{q}}.\mathrm{B}$. It is readily verified that the path $\zeta'\zeta''$ is a path starting from e' and witnessing the satisfaction of $\exists \mathcal{A}.\mathrm{B}$, which implies that $\mathrm{e} \in (\exists \mathcal{A}.\mathrm{B})^{\mathcal{J}}$. This finishes the proof.

4. $\mathcal{J} \models \exists \mathcal{A}.\mathrm{B} \sqsubseteq \mathrm{A}$. The proof goes by induction on k where the inductive assumption states:

   "for any $\mathrm{C}, \mathrm{D}, \exists \mathcal{A}'$ such that the TBox $\mathcal{T}$ contains $\mathrm{C} \equiv \exists \mathcal{A}'.\mathrm{D}$ and for any $\mathrm{d} \in (\exists \mathcal{A}'.\mathrm{D})^{\mathcal{J}}$ for which there is a path ζ witnessing the satisfaction of $\exists \mathcal{A}'.\mathrm{D}$ on which the total number of component changes is at most k we have $\mathrm{d} \in \mathrm{C}^{\mathcal{J}}$."

   In the case when $k = 0$ note that the whole path $\zeta \coloneqq \mathrm{d}_1 (= \mathrm{d}), \mathrm{d}_2, \ldots, \mathrm{d}_n$ witnessing the satisfaction of $\exists \mathcal{A}'.\mathrm{D}$ in $\mathcal{J}$ is contained in a single component. Hence, the path $\mathfrak{f}(\zeta) = \mathfrak{f}(\mathrm{d}_1), \mathfrak{f}(\mathrm{d}_2), \ldots, \mathfrak{f}(\mathrm{d}_n)$ is a witness path for $\mathfrak{f}(\mathrm{d})$ and $\exists \mathcal{A}'.\mathrm{D}$. Thus $\mathfrak{f}(\mathrm{d}) \in \mathrm{C}^{\mathcal{I}}$. As $\mathfrak{f}$ preserves atomic types, we get $\mathrm{d} \in \mathrm{C}^{\mathcal{J}}$.

   Otherwise take any $k > 0$ and assume that the hypothesis holds for all smaller k. Let $\zeta \coloneqq \mathrm{d}_1 (= \mathrm{d}), \mathrm{d}_2, \ldots, \mathrm{d}_n$ once again be the desired path witnessing that $\exists \mathcal{A}'.\mathrm{D}$ holds, and let e be the first domain element on ζ being in a different component than d. Assume that $\mathcal{A}'$ in the run witnessing that $\zeta \models \mathcal{A}'$ ends in the state q after reading a role leading to e. As our TBox is in normal form it also contains $\mathrm{C}_{\mathsf{q}} \equiv \exists \mathcal{A}'_{\mathsf{q}}.\mathrm{D}$. Let ζ' be the suffix of ζ starting at e. Clearly ζ' starts from e and witnesses the satisfaction of $\exists \mathcal{A}'_{\mathsf{q}}.\mathrm{D}$ with one component change less than ζ. So, by the inductive assumption, we have that $\mathrm{e} \in (\mathrm{C}_{\mathsf{q}})^{\mathcal{J}}$. Thus we also have that $\mathfrak{f}(\mathrm{e}) \in (\mathrm{C}_{\mathsf{q}})^{\mathcal{I}}$ (since $\mathfrak{f}$ preserves atomic types). So there is a path ρ' in $\mathcal{I}$ such that it is a witness path for $\mathfrak{f}(\mathrm{e})$ and $\exists \mathcal{A}'_{\mathsf{q}}.\mathrm{D}$. Note that, by the construction of components, the path $\mathfrak{f}(\mathrm{d}), \mathfrak{f}(\mathrm{d}_2), \ldots \mathfrak{f}(\mathrm{d}_j) = \mathfrak{f}(\mathrm{e})$ is isomorphic to the j-element prefix of ζ. Indeed, there are two cases: (1) e is not a nominal element and then we know that d_{j-1} and e are linked by exactly the same atomic roles as $\mathfrak{f}(\mathrm{d}_{j-1})$ and $\mathfrak{f}(\mathrm{e})$: it follows from the 2nd item of the linking process; or (2) e is a nominal and then we see that the role-connections between d_{j-1} and e are identical to the ones between $\mathfrak{f}(\mathrm{d}_{j-1})$ and $\mathfrak{f}(\mathrm{e})$ by the 1st item of the linking process. This implies that there is a run of $\mathcal{A}'$ from its initial state that starts from $\mathfrak{f}(\mathrm{d})$ and ends on $\mathfrak{f}(\mathrm{d}_j) = \mathfrak{f}(\mathrm{e})$ in state q. By concatenating the mentioned path and ζ' we obtain a witness path for $\mathfrak{f}(\mathrm{d})$ and $\exists \mathcal{A}'.\mathrm{D}$. Hence, $\mathfrak{f}(\mathrm{d}) \in \mathrm{C}^{\mathcal{I}}$, which (by the preservation of atomic types by $\mathfrak{f}$) allows us to conclude that $\mathrm{d} \in \mathrm{C}^{\mathcal{J}}$.

This concludes the proof that the constructed interpretation $\mathcal{J}$ is indeed a model of $\mathcal{K}$.

10.1.5 Preservation of query (non) entailment

We next establish that $\mathcal{J}$ does not satisfy q.

Lemma 10.5 The interpretation $\mathcal{J}$ constructed in Section 10.1.3 does not satisfy the query q.

The rest of this section is dedicated to the proof of the above lemma. Towards a contradiction, assume $\mathcal{J} \models q$. Thus, there is a CQ q_i such that $\mathcal{J} \models q_i$ and let η be a match witnessing it.

Query graphs Let Δ_{q_j} be the image of η. We will define a homomorphism from $\mathcal{J}{\restriction}_{\Delta_{q_j}}$ to $\mathcal{I}$, which will provide us with a match of q_j in $\mathcal{I}$, contradicting our initial assumption that $\mathcal{I} \not\models q$. For convenience we treat separately the role-connections among non-nominal elements of Δ_{q_j} and the role-connections involving at least one nominal element. Let us denote with $\Delta^*_{q_j}$ the set $\Delta_{q_j} \setminus \mathsf{Nom}_{\mathcal{J}}$ and let $G^*_{q_j}$ be the **Gaifman graph** of $\mathcal{J}{\restriction}_{\Delta^*_{q_j}}$ (*i.e.* the graph, whose nodes are the domain elements, and an undirected edge between a pair of nodes is present if the nodes are connected by some atomic role). Note that the edges of $G^*_{q_j}$ correspond to parent-child role-connections inside components of $\mathcal{J}$ or the role-connections between leaves of components and (non-nominal) origins of some other components. Let $G_1, \ldots, G_m$ be the connected components of $G^*_{q_j}$. To avoid notational clatter we will denote a graph and the set of its nodes with the same letter.

Active, Lower, and Upper Components We next construct a homomorphism $\mathfrak{h}_k$ from $\mathcal{J}\restriction_{G_k}$ into $\Delta^{\mathcal{I}}$, for $k = 1, \ldots, m$. Setting additionally $\mathfrak{h}_0 \colon \mathsf{Nom}_{\mathcal{J}} \cap \Delta_{q_j} \to \mathsf{Nom}_{\mathcal{I}}$ in the natural way: $\mathfrak{h}_0(\mathsf{o}^{\mathcal{J}}) \coloneqq \mathsf{o}^{\mathcal{I}}$, we will get the desired homomorphism $\mathfrak{h} \coloneqq \bigcup_{k=0}^{m} \mathfrak{h}_k$ from Δ_{q_j} into $\Delta^{\mathcal{I}}$. Consider a single G_k. Call the components of $\mathcal{J}$ containing nodes of G_k **active** (for G_k). If there is only one active component then as $\mathfrak{h}_k$ we take the restriction of $\mathfrak{f}$ to G_k. Otherwise call an active component **upper** (resp. **lower**) if at least one of its leaves (resp. its origin) belongs to G_k. Observe that for an active component it cannot be the case that both its origin and a leaf belong to G_k (since the path leading from the origin to a leaf has at least $K{+}1$ nodes and G_k is connected and has at most K nodes), and that each active component is either upper or lower (since different components may be joined only by edges between leaves and origins). The origins of the lower components are not nominal elements. By our strategy of joining the components it must be the case that either all the upper components are of the form $\mathcal{D}^{*,b}_{\pi,*,*}$ for some fixed $\pi \in \mathit{DTP}_{\mathcal{I}} \setminus \mathit{NTP}_{\mathcal{I}}$ and $b \in \{0, 1\}$, meaning that they are all isomorphic to $\mathcal{C}_\pi$ or there is only one upper component $\mathcal{D}_\pi$ for some $\pi \in \mathit{NTP}_{\mathcal{I}}$, which is then isomorphic to $\mathcal{C}_\pi$. For every d' of G_k belonging to an upper component we set $\mathfrak{h}_k(\mathrm{d}') \coloneqq \mathfrak{f}(\mathrm{d}')$. Note that the images of all such d' are members of the single component $\mathcal{C}_\pi$ in $\mathcal{I}$. It remains to define $\mathfrak{h}_k$ for the elements of the lower components.

Defining the desired homomorphisms Consider any lower component $\mathcal{D}^{\pi,b}_{\pi',i,j}$ and its origin e' (of type π'). Let d' be the i-th leaf of an upper component. In our process d' has been connected (via some role) to e' and role connections are identical to the role-connections between the i-th leaf of $\mathcal{C}_\pi$ and its j-th child e in $\mathcal{I}$. Denoting e^* the origin of $\mathcal{C}_{\pi'}$, by the definition of downward types we have that $\mathcal{I}\restriction_{\mathit{Subtree}^{\leq \mathrm{K}}_{\mathcal{I}}(\mathrm{e})}$ is isomorphic to the upper part $\mathit{Subtree}^{\leq K}_{\mathcal{I}}(e^*)$ of $\mathcal{C}_{\pi'}$. Let $\mathfrak{g} \colon \mathit{Subtree}^{\leq K}_{\mathcal{I}}(\mathrm{e}^*) \to \mathit{Subtree}^{\leq \mathrm{K}}_{\mathcal{I}}(\mathrm{e})$ be the appropriate isomorphism. For all d from $\mathcal{D}^{\pi,b}_{\pi',i,j}$ we set $\mathfrak{h}_k(\mathrm{d}) \coloneqq \mathfrak{g}(\mathfrak{f}(\mathrm{d}))$. Let $\mathfrak{h} \coloneqq \bigcup_{k=0}^{m} \mathfrak{h}_k$.

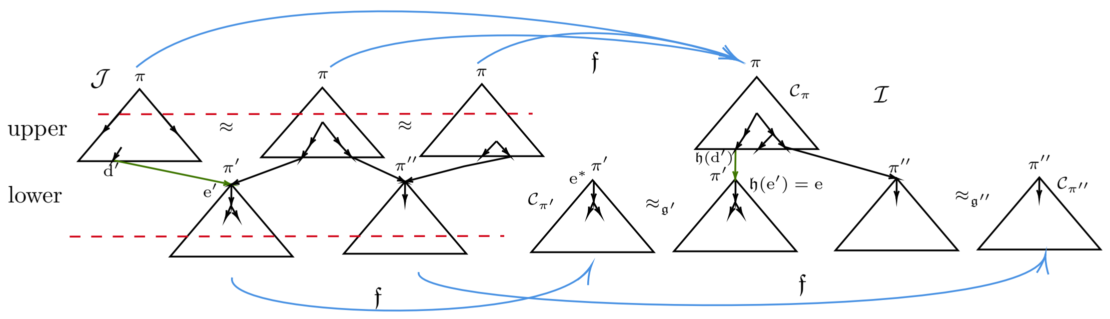


Figure 10.1: Constructing the homomorphism $\mathfrak{h}_k$. For the upper components $\mathfrak{h}_k$ coincides with $\mathfrak{f}$, for the lower ones it coincides with $\mathfrak{g} \cdot \mathfrak{f}$, for the appropriate $\mathfrak{g}$.

Correctness We explain that $\mathfrak{h}$ is indeed a homomorphism. Concept preservation follows from the fact that $\mathfrak{f}$ and the $\mathfrak{g}$s used in the definition of the $\mathfrak{h}_k$ preserve atomic types and that the nominal elements of $\mathcal{J}$ have the same atomic types as the corresponding nominal elements of $\mathcal{I}$. Assume now that $(\mathrm{d}', \mathrm{e}') \in r^{\mathcal{J}}$ for some $\mathrm{d}', \mathrm{e}' \in \Delta_{q_j}$ and we will show $(\mathfrak{h}(\mathrm{d}'), \mathfrak{h}(\mathrm{e}')) \in r^{\mathcal{I}}$. We consider the following three cases:

1. $\mathrm{d}', \mathrm{e}' \in \Delta^*_{q_j}$ and they belong to the same component. Then $(\mathfrak{h}(\mathrm{d}'), \mathfrak{h}(\mathrm{e}')) \in r^{\mathcal{I}}$ holds, since $\mathfrak{f}$ acts as a partial isomorphism when restricted to a single component and $\mathfrak{g}$ is a partial isomorphism in $\mathcal{I}$.
2. $\mathrm{d}', \mathrm{e}' \in \Delta^*_{q_j}$ and they belong to different components. Then d' is a leaf of a component and e' is an origin of another component, or vice versa; (w.l.o.g. we focus on former case). Looking at Figure 10.1 we can see that $\mathfrak{h}(\mathrm{e}')(= \mathfrak{f}(\mathfrak{g}(\mathrm{e}'))$ for the appropriate $\mathfrak{g}$) is a child of $\mathfrak{h}(\mathrm{d}')(= \mathfrak{f}(\mathrm{d}'))$. By the construction of $\mathcal{J}$ the connection between d' and e' in $\mathcal{J}$ is isomorphic to the connection between $\mathfrak{h}(\mathrm{d}')$ and $\mathfrak{h}(\mathrm{e}')$ in $\mathcal{I}$ (from which the claim follows).

3. At least one of $\mathrm{d}', \mathrm{e}' \in \Delta_{q_j} \setminus \Delta^*_{q_j}$ $(\subseteq \mathsf{Nom}_{\mathcal{J}})$.
   Follows from the fact that when defining $\mathcal{J}$ we always join every element d' with the nominal elements in $\mathcal{J}$ in the same way as $\mathfrak{f}(\mathrm{d}')$ is joined with the corresponding nominals in $\mathcal{I}$, and that $\mathfrak{f}(\mathrm{d}')$ and $\mathfrak{g}(\mathfrak{f}(\mathrm{d}'))$ are joined with nominals in $\mathcal{I}$ in the same way (equal downward types!).

This concludes the proof that $\mathcal{J} \not\models q$, and also the proof of:

Corollary 10.6

The interpretation $\mathcal{J}$ is a *finite* countermodel for $\mathcal{K}$ and q.

10.2 Applications and Open Problems

In this section we presented a novel model-theoretic construction that allowed us to establish that the DLs $\mathcal{ZOQ}$ and $\mathcal{ZOI}$ are finitely-controllable for the class of positive existential queries. We believe that this kind of a construction may have many further applications. One application is the notion of scattered unravellings, presented in Section 5.2. We also recently used a similar construction (together with Benno Fünfstück) to establish that a certain restriction of the guarded fragment is expressively complete in the sense of Van Benthem characterisation theorems. What is more, the technique seems to be applicable also to the case of μ-calculus with graded modalities and nominals.

Similarly to the previous section, by arguing along the lines of the original work of Calvanese et al. [CEO09], we can leverage our findings to strengthen their results on query containment as well as the $\mathcal{SR}$ family of DLs as follows (compare with Theorem 9.13)

Corollary 10.7

One can decide in time doubly-exponential w.r.t. the input KB $\mathcal{K}$ whether a P2RPQ q' is finitely contained by a CQ q modulo $\mathcal{K}$ (*i.e.* whether every finite model of $\mathcal{K}$ satisfying q' also satisfies q) for KBs $\mathcal{K}$ written in either $\mathcal{ZIQ}$ or $\mathcal{ZOQ}$.

Once more, similarly to the previous section, we can lift [CEO09, Prop. 5.1] the results on the $\mathcal{Z}$ family of DLs to the $\mathcal{SR}$ family. Compare to Theorem 9.14.

Corollary 10.8

For KBs $\mathcal{K}$ written in $\mathcal{SROQ}$ or $\mathcal{SROI}$, and PEQs q that employs only simple roles, we can decide whether $\mathcal{K} \models q$ holds in the finite in triply-exponential time w.r.t. $|\mathcal{K}|+|q|$.

Part IV

A Step Beyond the $\mathscr{Z}$ Family of DLs

11

Beyond Regularity: Exploring Non-Regular Extensions of $\mathcal{ALC}_{\mathsf{reg}}$

Contents

Motivation

We recall that the logic $\mathcal{ALC}_{\mathsf{reg}}$, the core of the $\mathcal{Z}$ family of DLs, was already studied in 1979 by the formal-verification community [FL79], under the name of Propositional Dynamic Logic (PDL). Relationship between (extensions of) PDL and $\mathcal{ALC}_{\mathsf{reg}}$ were investigated by De Giacomo and Lenzerini [DL94]. Note that the spectrum of recognizable word languages is relatively wide. Hence the question of whether regular constraints in path expressions of $\mathcal{ALC}_{\mathsf{reg}}$ can be lifted to more expressive classes of languages received a lot of attention from researchers. We call such extensions of PDL *non-regular*. After the first undecidability proof of satisfiability of $\mathcal{ALC}_{\mathsf{reg}}$ by context-free languages [HPS81b, Cor. 2.2], several decidable cases were identified. For instance, Koren and Pnueli [KP83, Sec. Decidability] proved that $\mathcal{ALC}_{\mathsf{reg}}$ extended with the simplest non-regular language $r^{\#}s^{\#} := \{r^n s^n \mid n \in \mathbb{N}\}$ for *fixed* roles r, s is decidable; while combining it with $s^{\#}r^{\#}$ leads to undecidability [HPS81a, Thm. 3.2]. This surprises at first glance, but as it was shown later [LLS07, Thm. 18], PDL extended with a broad class of input-driven context-free languages, called *visibly pushdown languages* [AM09, Sec. 5], remains decidable. This generalises all previously known decidability results, and partially explains the reason behind known failures (*e.g.* the languages $r^{\#}s^{\#}$ and $s^{\#}r^{\#}$ cannot be both visibly-pushdown under the same partition of the alphabet). Three years ago, the decidability boundary was pushed even further [BL21, Ex. 1], by allowing for mixing modalities in visibly-pushdown expressions (for instance, allowing the user to specify that "for all positive integers $n \in \mathbb{N}$, all t-successors of r^n-reachable elements can s^n-reach an element fulfilling φ"). Despite the presence of a plethora of various results concerning non-regular extensions of PDL [KP83, HP84, HS96, HR93, BL21], to the best of our knowledge the extensions of non-regular PDL with popular features supported by W3C ontology languages are yet to be investigated. Such extensions include, among others, *nominals* (constants), *inverse roles* (inverse programs), *functionality* or *counting* (deterministic programs or graded modalities), and the Self operator (self-loops). The honourable exception is the unpublished undecidability result for $\mathcal{ALC}_{\mathsf{reg}}$ extended with the language $\{r^n s (r^-)^n \mid n \in \mathbb{N}\}$, where r^- denotes the converse of r, from Göller's thesis [Göl08] (answering an open problem of Demri [Dem07, Probléme ouvert 29]). The lack of results on entailment of non-regular queries over ontologies is also intriguing, taking into account positive results for conjunctive visibly-pushdown queries in the setting of relational-databases [LL15, Thm. 2].

Our Contribution and Overview of the Chapter

In this chapter of the thesis we contribute to the further understanding of the aforementioned questions by proving various undecidability results. Section 11.2 establishes undecidability of the concept satisfiability of $\mathcal{ALC}_{\mathsf{vpl}}$ extended with the seemingly innocent Self operator. Section 11.3 establishes undecidability of the concept satisfiability of $\mathcal{ALC}_{\mathsf{vpl}}$ extended with nominals. More specifically, the undecidability arises already if the only non-regular language present in concepts is $r^{\#}s^{\#}$. Finally, Section 11.4 establishes undecidability of the query entailment problem over $\mathcal{ALC}$-TBoxes, in which our queries can employ atoms involving the language $r^{\#}s^{\#}$. We conclude with a list of open problem (Section 11.5).

11.1 Extra Preliminaries

This section is dedicated for providing missing definitions concerning visibly pushdown languages. We also discuss related undecidability results that follow from the literature.

Visibly-pushdown languages. The class $\mathbb{VPL}$ of **visibly-pushdown languages** [AM09, Sec. 5] (VPLs) is a well-behaved family of context-free languages, in which the usage of the stack in the underlying automaton model is input-driven. For the exposition of VPLs we follow Löding et al. [LLS07, Sec. 2.2].

Definition 11.1 A **pushdown alphabet** Σ is an alphabet equipped with a partition $(\Sigma_c, \Sigma_i, \Sigma_r)$. The elements of Σ_c, Σ_i, and Σ_r are called, respectively, **call** letters, **internal** letters, and **return** letters. A **visibly-pushdown automaton** (VPA) $\mathcal{A}$ over a pushdown alphabet Σ is a tuple $(\mathtt{Q}, \mathtt{I}, \mathtt{F}, \Gamma, \mathtt{T})$, where $\mathtt{Q}$ is a finite set of states, $\mathtt{I}$ is a finite subset of *initial states*, $\mathtt{F}$ is a finite subset of *final states*, Γ is a finite stack alphabet that contains a *bottom-of-stack* symbol $\lhd$, and $\mathtt{T}$ is a transition relation of type $\mathtt{T} \subseteq (\mathtt{Q} \times \Sigma_c \times \mathtt{Q} \times (\Gamma \setminus \lhd)) \quad \cup \quad (\mathtt{Q} \times \Sigma_r \times \Gamma \times \mathtt{Q})) \quad \cup \quad (\mathtt{Q} \times \Sigma_i \times \mathtt{Q})$.

The next definition concerns configurations and runs of visibly automata.

Definition 11.2 A **configuration** of a VPA $\mathcal{A}$ is a pair $(\mathtt{q}, \sigma) \in \mathtt{Q} \times (\Gamma \setminus \lhd)^* \lhd$ of a state $\mathtt{q}$ and a stack content σ. For a letter $\mathtt{a}$ and a configuration $(\mathtt{q}, \sigma)$ we say that $(\mathtt{q}', \sigma')$ is an $\mathtt{a}$-successor of $(\mathtt{q}, \sigma)$, and denote this fact with $(\mathtt{q}, \sigma) \to_{\mathtt{a}} (\mathtt{q}', \sigma')$, if some of the following cases hold:

- $\mathtt{a} \in \Sigma_c$, $\sigma' = \gamma\sigma$ and there is a transition $(\mathtt{q}, \mathtt{a}, \mathtt{q}', \gamma) \in \mathtt{T}$.
- $\mathtt{a} \in \Sigma_i$, $\sigma' = \sigma$ and there is a transition $(\mathtt{q}, \mathtt{a}, \mathtt{q}') \in \mathtt{T}$.
- $\mathtt{a} \in \Sigma_r$, either (i) $\sigma = \gamma\sigma'$ and there is a transition $(\mathtt{q}, \mathtt{a}, \gamma, \mathtt{q}') \in \mathtt{T}$, or (ii) $\sigma = \sigma' = \lhd$ and $(\mathtt{q}, \mathtt{a}, \lhd, \mathtt{q}') \in \mathtt{T}$.

Given a word $\mathtt{w} := \mathtt{a}_1 \ldots \mathtt{a}_n$, a **run** of $\mathcal{A}$ on $\mathtt{w}$ is a sequence $(\mathtt{q}_0, \lhd) \to_{\mathtt{a}_1} (\mathtt{q}_1, \sigma_1) \to_{\mathtt{a}_2} \ldots \to_{\mathtt{a}_n} (\mathtt{q}_n, \sigma_n)$ where $\mathtt{q}_0 \in \mathtt{I}$. We call $\mathtt{w}$ **accepted** by $\mathcal{A}$ if there is a run of $\mathcal{A}$ on $\mathtt{w}$ in which the last configuration contains a final state.

The language $\mathcal{L}(\mathcal{A})$ of $\mathcal{A}$ is composed of all words accepted by $\mathcal{A}$. A language $\mathcal{L}$ (*i.e.* a set of words) over Σ is **visibly-pushdown** if there is a VPA $\mathcal{A}$ over Σ for which $\mathcal{L}(\mathcal{A}) = \mathcal{L}$.

Example 11.3. Suppose that r is a call letter and s is a return letter. Then the languages $r^{\#}s^{\#} := \{r^n s^n \mid n \in \mathbb{N}\}$ and $r^{\#}s^{>\#} := \{r^n s^{n+m+1} \mid n, m \in \mathbb{N}\}$ are visibly-pushdown. Under such a choice of r and s, the language $s^{\#}r^{\#}$ *is not* visibly-pushdown. Moreover, every regular language is visibly-pushdown.

The logic $\mathcal{ALC}_{\mathsf{vpl}}$. Throughout this chapter of the thesis, Σ_{all} is presented as a pushdown alphabet

$$\Sigma_{\mathsf{vpl}} := ((\mathbf{N_R})_c, (\mathbf{N_R})_i \cup \{C? \mid C \in \mathbf{N_C}\}, (\mathbf{N_R})_r),$$

where the sets $(\mathbf{N_R})_c, (\mathbf{N_R})_i, (\mathbf{N_R})_r$ form a partition of $\mathbf{N_R}$. Hence, we define $\mathcal{ALC}_{\mathsf{vpl}}$ as the restriction of $\mathcal{ALC}_{\mathsf{all}}$ to visibly-pushdown languages over finite subsets of Σ_{vpl}, in which languages in existential and universal restrictions are represented by means of nondeterministic VPA. The logic $\mathcal{ALC}_{\mathsf{vpl}}$ generalises many other logics with non-regular path expressions, and has a 2ExpTime-complete [LLS07, Thm. 18–19] concept satisfiability problem. As a special case of $\mathcal{ALC}_{\mathsf{vpl}}$, we also consider a, rather minimalistic, extension of $\mathcal{ALC}_{\mathsf{reg}}$ with the language $r^{\#}s^{\#}$ for one *fixed* call letter $r \in (\mathbf{N_R})_c$ and one *fixed* return letter $s \in (\mathbf{N_R})_r$. We denote it here by $\mathcal{ALC}_{\mathsf{reg}}^{r^{\#}s^{\#}}$. The concept satisfiability problem for $\mathcal{ALC}_{\mathsf{reg}}^{r^{\#}s^{\#}}$ was shown to be decidable [KP83, Sec. Decidability] already 40 years ago by Koren and Pnueli, but its extensions with popular features like nominals or functionality are still unexplored.

Tree models. Löding et al. established that $\mathcal{ALC}_{\mathsf{vpl}}$ possesses a tree-model property. An interpretation $\mathcal{I}$ is *tree-like* if its domain is a prefix-closed subset of $\mathbb{N}^*$, and for all $r \in \mathbf{N_R}$ and $\mathrm{d}, \mathrm{e} \in \Delta^{\mathcal{I}}$ the condition "if $(\mathrm{d}, \mathrm{e}) \in r^{\mathcal{I}}$, then $\mathrm{e} = \mathrm{d}n$ for some $n \in \mathbb{N}$" holds. An interpretation is *single-role* if any two domain elements are connected by at most one role. The following lemma follows immediately from the proof of Proposition 8 by Löding et al. [LLS07].

Corollary 11.4 (Consequence of the proof of Prop. 8 of [LLS07])

Every satisfiable $\mathcal{ALC}_{\mathsf{vpl}}$-TBox $\mathcal{T}$ has a single-role tree-like model.[a] Moreover, for any query q preserved under homomorphisms we have that $\mathcal{T} \not\models q$ if and only if there exists a single-role tree-like model $\mathcal{I} \models \mathcal{T}$ such that $\mathcal{I} \not\models q$.

[a]The original work considers concepts only. However, all their results transfer immediately to the case of TBoxes, as TBoxes can be internalised in concepts in the presence of regular expressions [BCM+03, p. 186]. The queries are not mentioned either: the so-called tree model property is established with a suitable notion of unravelling, which produces interpretations that can be then homomorphically mapped to the original interpretations (entailing the preservation of the non-satisfaction of query).

Undecidability results for extensions of $\mathcal{ALC}_{\mathsf{vpl}}$ that follow from the literature. Other popular (not yet mentioned) features supported by W3C ontology languages are *inverse roles* and *role hierarchies*, defined as in Preliminaries. For bibliographical purposes, we would like to use the extra space given here to briefly discuss how these features result in the undecidability of the respective extensions of $\mathcal{ALC}_{\mathsf{vpl}}$. It was shown by Stefan Göller in his PhD thesis [Göl08, Prop. 2.32] that $\mathcal{ALC}_{\mathsf{reg}}$ extended with the single visibly-pushdown language $\{r^n s (r^-)^n \mid n \in \mathbb{N}\}$ is undecidable.

Corollary 11.5

The concept satisfiability problem for $\mathcal{ALC}_{\mathsf{vpl}}$ with inverses is undecidable, even if the only allowed non-regular language is $\{r^n s (r^-)^n \mid n \in \mathbb{N}\}$ for fixed $r, s \in \mathbf{N_R}$.

We next discuss the extension of $\mathcal{ALC}_{\mathsf{vpl}}$ with role hierarchies. Fix r and r' to be call letters, and s and s' to be return letters. Suppose that an interpretation $\mathcal{I}$ satisfies all of the statements $s \subseteq r'$, $r' \subseteq s$, $s' \subseteq r$, and $r \subseteq s'$. Clearly, for all elements $\mathrm{d} \in \Delta^{\mathcal{I}}$ and concepts C we have that $\mathrm{d} \in (\exists s^{\#} r^{\#}.\mathrm{C})^{\mathcal{I}}$ if and only if $\mathrm{d} \in (\exists r'^{\#} s'^{\#}.\mathrm{C})^{\mathcal{I}}$. Thus $\mathcal{ALC}_{\mathsf{vpl}}$ with role-hierarchies can express concepts of $\mathcal{ALC}_{\mathsf{reg}}$ extended with both non-regular languages $r^{\#}s^{\#}$ and $s^{\#}r^{\#}$. By undecidability of the concept satisfiability [KP83, Sec. Decidability] of the latter we conclude:

Corollary 11.6

The concept satisfiability problem for $\mathcal{ALC}_{\mathsf{vpl}}$ with role-hierarchies is undecidable, even if the only allowed non-regular languages are $r^{\#}s^{\#}$ and $r'^{\#}s'^{\#}$ for fixed call letters r, r' and return letters s, s'.

11.2 Negative results I: The Seemingly innocent Self operator

We start our series of negative results, by showing (in our opinion) a rather surprising undecidability result. Henceforth we employ the Self operator, a modelling feature supported by two profiles of the OWL 2 Web Ontology Language [HKS06, KRH08]. Recall that the Self operator allows us to specify the situation when an element is related to it*self* by a binary relationship, *i.e.* we interpret the concept $\exists r.\mathsf{Self}$ in an interpretation $\mathcal{I}$ as the set of all those elements d for which (d,d) belongs to $r^{\mathcal{I}}$. We provide a reduction from the undecidable problem of non-emptiness of the intersection of deterministic one-counter automata (DOCA) [Val73, p. 75]. Such an automaton model is similar to pushdown automata, but its stack alphabet is single-letter only. The Self operator will be especially useful to introduce "disjunction" to paths.

Let Σ be an alphabet and $\mathtt{w} := (\mathtt{a}_1, \mathtt{b}_1) \ldots (\mathtt{a}_n, \mathtt{b}_n)$ be a word over $\Sigma \times \{c, r, i\}$. We call the word $\pi_1(\mathtt{w}) := \mathtt{a}_1 \ldots \mathtt{a}_n$ from Σ^* the *projection* of $\mathtt{w}$. An important property of DOCAs is that they can be made visibly pushdown in the following sense.

Lemma 11.7 For any deterministic one-counter automaton $\mathcal{A}$ over the alphabet Σ, we can construct a visibly-pushdown automaton $\tilde{\mathcal{A}}$ over a pushdown alphabet $\tilde{\Sigma} := (\Sigma \times \{c\}, (\Sigma \times \{i\}) \cup \{\mathtt{x}\}, \Sigma \times \{r\})$ with a fresh internal letter $\mathtt{x}$ such that all words in $\mathcal{L}(\tilde{\mathcal{A}})$ are of the form $\tilde{\mathtt{a}_1}\mathtt{x}\tilde{\mathtt{a}_2}\mathtt{x} \ldots \mathtt{x}\tilde{\mathtt{a}_n}$ for $\tilde{\mathtt{a}_1}, \ldots, \tilde{\mathtt{a}_n} \in \Sigma \times \{c, i, r\}$, and

$$\mathcal{L}(\mathcal{A}) = \{\pi_1(\tilde{\mathtt{w}}) \mid \tilde{\mathtt{w}} := \tilde{\mathtt{a}_1} \ldots \tilde{\mathtt{a}_n},\ \ \tilde{\mathtt{a}_1}\mathtt{x} \ldots \mathtt{x}\tilde{\mathtt{a}_n} \in \mathcal{L}(\tilde{\mathcal{A}})\}.$$

Proof sketch. Alur and Madhusudan proved [AM09, Thm. 5.2] that for any context-free language $\mathcal{L}$ over Σ there exists a VPL $\hat{\mathcal{L}}$, over the pushdown alphabet $(\Sigma \times \{c\}, (\Sigma \times \{i\}), \Sigma \times \{r\})$, for which $\mathcal{L} = \{\pi_1(\mathtt{w}) \mid \mathtt{w} \in \hat{\mathcal{L}}\}$ holds.[1] Suppose now that a one-counter automaton $\mathcal{A}$ is given. By means of the previous construction, we obtain a visibly-pushdown automaton $\hat{\mathcal{A}} := (\mathtt{Q}, \mathtt{I}, \mathtt{F}, \Gamma, \mathtt{T})$ for which $\mathcal{L}(\mathcal{A}) = \{\pi_1(\mathtt{w}) \mid \mathtt{w} \in \mathcal{L}(\hat{\mathcal{A}})\}$ holds. What remains to be done is to "insert" the *internal* letter $\mathtt{x}$ after every position of a word accepted by $\hat{\mathcal{A}}$. As reading internal letters by visibly-pushdown automata do not affect the content of their stacks, we may proceed as in standard constructions from the theory of regular languages [Sip13, Ex. 1.31]. As the first step, we expand the set of states $\mathtt{Q}$ with fresh states of the form $\mathtt{q}_\delta$ for all $\delta \in \mathtt{T}$. As the second step, we "split" every transition δ in $\mathtt{T}$ into two "parts". Suppose that δ leads from $\mathtt{q}$ to $\mathtt{q}'$ after reading the letter $\mathtt{a}$. We thus (i) replace δ in $\mathtt{T}$ with the transition that transforms $\mathtt{q}$ into $\mathtt{q}_\delta$ after reading $\mathtt{a}$ (and has the same effect on the stack as δ has), and (ii) append the transition $(\mathtt{q}_\delta, \mathtt{x}, \mathtt{q}')$ to $\mathtt{T}$. Call the resulting automaton $\tilde{\mathcal{A}}$. It can now be readily verified that $\mathcal{L}(\tilde{\mathcal{A}}) = \{\hat{\mathtt{a}_1}\mathtt{x}\hat{\mathtt{a}_2}\mathtt{x} \ldots \mathtt{x}\hat{\mathtt{a}_n} \mid \hat{\mathtt{w}} := \hat{\mathtt{a}_1} \ldots \hat{\mathtt{a}_n},\ \hat{\mathtt{w}} \in \mathcal{L}(\hat{\mathcal{A}})\}$ holds, and thus, by the relationship between $\mathcal{A}$ and $\hat{\mathcal{A}}$, the automaton $\tilde{\mathcal{A}}$ is as desired. □

Let us fix a finite alphabet $\Sigma \subseteq \mathbf{N_R}$. We also fix two deterministic one-counter automata $\mathcal{A}_1$ and $\mathcal{A}_2$ over Σ, and let $\mathcal{C}_1$ and $\mathcal{C}_2$ be deterministic one-counter automata recognizing the complement of the languages of $\mathcal{A}_1$ and $\mathcal{A}_2$ (they exist as DOCA are closed under complement [Val73, p. 76]). Finally, we apply Lemma 11.7 to construct their visibly-pushdown counterparts $\tilde{\mathcal{A}}_1$, $\tilde{\mathcal{A}}_2$, $\tilde{\mathcal{C}}_1$, $\tilde{\mathcal{C}}_2$ over *the same* pushdown alphabet $\tilde{\Sigma}$. We stress that the letter $\mathtt{x}$, playing the role of a "separator", is identical for all of the aforementioned visibly-pushdown automata. Moreover, note that the non-emptiness of $\mathcal{L}(\tilde{\mathcal{A}}_1) \cap \mathcal{L}(\tilde{\mathcal{A}}_2)$ is not equivalent to the non-emptiness of $\mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_1)$, as the projection of a letter $\mathtt{a} \in \tilde{\Sigma}$ may be used by $\mathcal{A}_1$ and $\mathcal{A}_2$ in different contexts (*e.g.* both as a call or as a return).

We are going to encode words accepted by one-counter automata by means of word-like interpretations.

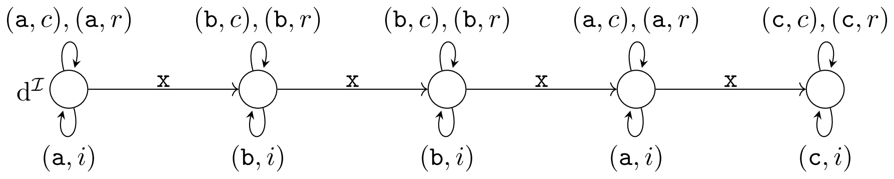
Definition 11.8 A pointed interpretation $(\mathcal{I}, \mathrm{d})$ is **Σ-friendly** if for every element $\mathrm{e} \in \Delta^{\mathcal{I}}$ that is $\mathtt{x}^*$-reachable from d in $\mathcal{I}$ there exists a unique letter $\mathtt{a} \in \Sigma$ so that e carries all $\tilde{\mathtt{a}}$-self-loops for all $\tilde{\mathtt{a}} \in \tilde{\Sigma}$ with $\pi_1(\tilde{\mathtt{a}}) = \mathtt{a}$, and no self-loops for all other letters in $\tilde{\Sigma}$ (also including $\mathtt{x}$).

[1] The main proof idea here is to take an input DOCA, and decorate the letters on its transitions with c, i, and r, depending on the counter action of the transition.

Observe that Σ-friendly interpretations can be axiomatised with an $\mathcal{ALC}^{\mathsf{Self}}$-concept $\mathrm{C}^{\Sigma}_{\mathrm{fr}}$:

$$\mathrm{C}^{\Sigma}_{\mathrm{fr}} := \forall \mathtt{x}^*. \bigsqcup_{\mathtt{a}\in\Sigma} \bigsqcap_{\mathtt{b}\neq\mathtt{a},\mathtt{b}\in\Sigma,\pi_1(\tilde{\mathtt{a}})=\mathtt{a},\pi_1(\tilde{\mathtt{b}})=\mathtt{b}} \Big([\exists\tilde{\mathtt{a}}.\mathsf{Self}] \sqcap \neg[\exists\tilde{\mathtt{b}}.\mathsf{Self}] \sqcap \neg[\exists\mathtt{x}.\mathsf{Self}]\Big).$$

Moreover, every $\mathtt{x}^*$-path ρ in a Σ-friendly $(\mathcal{I}, \mathrm{d})$ *represents* a word in Σ^* in the following sense: the i-th letter of such a word is $\mathtt{a}$ if and only if the i-th element of the path carries an $(\mathtt{a}, c)$-self-loop. This is well-defined, by the fact that every $\mathtt{x}^*$-reachable element in Σ-friendly $(\mathcal{I}, \mathrm{d})$ carries a $(\mathtt{a}, c)$-self-loop for a unique letter $\mathtt{a} \in \Sigma$. Consult Figure 11.1 for a visualization.



$(\mathtt{a}, c), (\mathtt{a}, r)$
$(\mathtt{b}, c), (\mathtt{b}, r)$
$(\mathtt{b}, c), (\mathtt{b}, r)$
$(\mathtt{a}, c), (\mathtt{a}, r)$
$(\mathtt{c}, c), (\mathtt{c}, r)$
$\mathrm{d}^{\mathcal{I}}$
x
x
x
x
$(\mathtt{a}, i)$
$(\mathtt{b}, i)$
$(\mathtt{b}, i)$
$(\mathtt{a}, i)$
$(\mathtt{c}, i)$

Figure 11.1: An example Σ-friendly $(\mathcal{I}, \mathrm{d})$ encoding the word $\mathtt{abbac}$.

As a special class of Σ-friendly interpretations we consider Σ-metawords.

Definition 11.9 We say that a pointed interpretation $(\mathcal{I}, \mathrm{d})$ is a **Σ-metaword** if it is a Σ-friendly interpretation of the domain $\mathbb{Z}_n$ for some positive $n \in \mathbb{N}$, the role name $\mathtt{x}$ is interpreted as the set $\{(i, i{+}1) \mid 0 \leq i \leq n{-}2\}$, and all other role names are either interpreted as $\emptyset$ or are subsets of the diagonal $\{(i, i) \mid i \in \mathbb{Z}_n\}$ (or, put differently, they appear only as self-loops).

The example Σ-friendly $(\mathcal{I}, \mathrm{d})$ from Figure 11.1 is actually a Σ-metaword. Note that for every word $\mathtt{w} \in \Sigma^+$ there is a Σ-metaword representing $\mathtt{w}$. A crucial observation regarding Σ-metawords is as follows.

Observation 11.10. Let $\ell \in \{1, 2\}$, Σ-metaword $(\mathcal{I}, \mathrm{d})$, and $\tilde{\mathtt{w}}$ be in the language of $\tilde{\mathcal{A}}_\ell$. Then the element d can $\{\tilde{\mathtt{w}}\}$-reach an element e via a path ρ if and only if for all odd indices i we have $\rho_i = \rho_{i+1}$ and for all even indices i we have $\rho_i + 1 = \rho_{i+1}$.

As the next step of the construction, we are going to decorate Σ-friendly interpretations with extra information on whether or not words represented by paths are accepted by $\mathcal{A}_1$. This is achieved by means of the following concept

$$\mathrm{C}_{\mathcal{A}_1} := \mathrm{C}^{\Sigma}_{\mathrm{fr}} \sqcap \forall\mathcal{L}(\tilde{\mathcal{A}}_1).\mathrm{Acc}_{\mathcal{A}_1} \sqcap \forall\mathcal{L}(\tilde{\mathcal{C}}_1).\neg\mathrm{Acc}_{\mathcal{A}_1},$$

for a fresh concept name $\mathrm{Acc}_{\mathcal{A}_1}$. We define the concept $\mathrm{C}_{\mathcal{A}_2}$ analogously. We have that:

Lemma 11.11 Fix $\ell \in \{1, 2\}$. If $\mathrm{C}_{\mathcal{A}_\ell}$ is satisfied by a pointed interpretation $(\mathcal{I}, \mathrm{d})$, then $(\mathcal{I}, \mathrm{d})$ is Σ-friendly and for every element $\mathrm{e} \in \Delta^{\mathcal{I}}$ that is $\mathtt{x}^*$-reachable from d via a path ρ we have $\mathrm{e} \in (\mathrm{Acc}_{\mathcal{A}_\ell})^{\mathcal{I}}$ if and only if the Σ-word represented by ρ belongs to $\mathcal{L}(\mathcal{A}_\ell)$. Moreover, after reinterpreting the concept name $\mathrm{Acc}_{\mathcal{A}_\ell}$, every Σ-metaword becomes a model of $\mathrm{C}_{\mathcal{A}_\ell}$.

Proof. The proof relies on Observation 11.10. Suppose that $(\mathcal{I}, \mathrm{d})$ is a pointed interpretation and $\mathrm{d} \in (\mathrm{C}_{\mathcal{A}_\ell})^{\mathcal{I}}$. We know that $(\mathcal{I}, \mathrm{d})$ is Σ-friendly by the satisfaction of $\mathrm{C}^{\Sigma}_{\mathrm{fr}}$. For the remainder of the proof, consider any element $\mathrm{e} \in \Delta^{\mathcal{I}}$ that is $\mathtt{x}^*$-reachable from d, say, via a path $\rho := \rho_1 \ldots \rho_n$. Let $\mathtt{w}$ be the word represented by ρ. This implies, that for every index i, the element ρ_i in $\mathcal{I}$ is equipped with a family of self-loops involving (a decorated) letter $\mathtt{w}_i$. We consider two cases:

- Assume that $\mathrm{e} \in (\mathrm{Acc}_{\mathcal{A}_\ell})^{\mathcal{I}}$. We will show that $\mathtt{w} \in \mathcal{L}(\mathcal{A}_\ell)$. Ad absurdum, suppose that $\mathtt{w} \notin \mathcal{L}(\mathcal{A}_\ell)$. Then, by definition of $\mathcal{C}_\ell$, we have that $\mathtt{w}$ belongs to $\mathcal{L}(\mathcal{C}_\ell)$. By the construction of $\tilde{\mathcal{C}}_\ell$ there exists a sequence $\star_1, \ldots, \star_n \in \{c, i, r\}$, for which the word

$\mathtt{u} := (\mathtt{w}_1, \star_1)\mathtt{x} \ldots \mathtt{x}(\mathtt{w}_n, \star_n)$ is accepted by $\tilde{\mathcal{C}}_\ell$. But then the path $\rho' := \rho_1\rho_1\rho_2\rho_2 \ldots \rho_n\rho_n$ witnesses $\{\mathtt{u}\}$-reachability (and thus $\mathcal{L}(\tilde{\mathcal{C}}_\ell)$-reachability) of e from d. By the satisfaction of $\forall\mathcal{L}(\tilde{\mathcal{C}}_\ell).\neg\mathrm{Acc}_{\mathcal{A}_\ell}$ by $(\mathcal{I}, \mathrm{d})$ we infer $\mathrm{e} \in (\neg\mathrm{Acc}_{\mathcal{A}_\ell})^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus (\mathrm{Acc}_{\mathcal{A}_\ell})^{\mathcal{I}}$. A contradiction.

- Assume that $\mathtt{w} \in \mathcal{L}(\mathcal{A}_\ell)$. We proceed analogously to the previous case. By the construction of $\tilde{\mathcal{A}}_\ell$ there exists a sequence $\star_1, \ldots, \star_n \in \{c, i, r\}$, for which the word $\mathtt{u} := (\mathtt{w}_1, \star_1)\mathtt{x} \ldots \mathtt{x}(\mathtt{w}_n, \star_n)$ is accepted by $\tilde{\mathcal{A}}_\ell$. Once again, the path $\rho' := \rho_1\rho_1\rho_2\rho_2 \ldots \rho_n\rho_n$ witnesses $\{\mathtt{u}\}$-reachability (and thus $\mathcal{L}(\tilde{\mathcal{A}}_\ell)$-reachability) of e from d. Due to the satisfaction of $\forall\mathcal{L}(\tilde{\mathcal{A}}_\ell).\mathrm{Acc}_{\mathcal{A}_\ell}$ by $(\mathcal{I}, \mathrm{d})$ we infer $\mathrm{e} \in (\mathrm{Acc}_{\mathcal{A}_\ell})^{\mathcal{I}}$, as desired.

For the last statement of the proof, take a Σ-metaword $\mathcal{I}$ that represents a word $\mathtt{w} \in \Sigma^*$. We alter the interpretation of the concept name $\mathrm{Acc}_{\mathcal{A}_\ell}$ in $\mathcal{I}$ so that $(\mathrm{Acc}^{\mathcal{I}}_{\mathcal{A}_\ell}) = \{i{-}1 \mid \mathtt{w}_1 \ldots \mathtt{w}_i \in \mathcal{L}(\mathcal{A}_\ell)\}$. It follows that $\forall\mathcal{L}(\tilde{\mathcal{A}}_\ell).\mathrm{Acc}_{\mathcal{A}_\ell} \sqcap \forall\mathcal{L}(\tilde{\mathcal{C}}_\ell).\neg\mathrm{Acc}_{\mathcal{A}_\ell}$ is indeed satisfied by $(\mathcal{I}, 0)$. □

Equipped with Lemma 11.11, we are ready to prove correctness of our reduction.

Lemma 11.12 $\mathrm{C}_{\mathcal{A}_1} \sqcap \mathrm{C}_{\mathcal{A}_2} \sqcap \exists\mathtt{x}^*.(\mathrm{Acc}_{\mathcal{A}_1} \sqcap \mathrm{Acc}_{\mathcal{A}_2})$ is satisfiable if and only if $\mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2) \neq \emptyset$.

Proof. For one direction, take $\mathtt{w} \in \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$. Let $(\mathcal{I}, \mathrm{d})$ be a Σ-metaword representing $\mathtt{w}$. By Lemma 11.11 we decorate $\mathcal{I}$ with concepts $\mathrm{Acc}_{\mathcal{A}_1}$ and $\mathrm{Acc}_{\mathcal{A}_2}$ so that $(\mathcal{I}, \mathrm{d}) \models \mathrm{C}_{\mathcal{A}_1} \sqcap \mathrm{C}_{\mathcal{A}_2}$, and the interpretations of concepts $\mathrm{Acc}_{\mathcal{A}_\ell}$ contain precisely the elements k for which the k-letter prefix of $\mathtt{w}$ belongs to $\mathcal{L}(\mathcal{A}_\ell)$. In particular, this means that $(|\mathtt{w}|{-}1) \in (\mathrm{Acc}_{\mathcal{A}_1} \sqcap \mathrm{Acc}_{\mathcal{A}_2})^{\mathcal{I}}$. As $(|\mathtt{w}|{-}1)$ is $\mathtt{x}^*$-reachable from 0, we conclude the satisfaction of $\exists\mathtt{x}^*.(\mathrm{Acc}_{\mathcal{A}_1} \sqcap \mathrm{Acc}_{\mathcal{A}_2})$ by $(\mathcal{I}, 0)$. Hence, the concept from the statement of Lemma 11.12 is indeed satisfiable.

For the other direction, assume that $(\mathcal{I}, \mathrm{d})$ is a model of $\mathrm{C}_{\mathcal{A}_1} \sqcap \mathrm{C}_{\mathcal{A}_2} \sqcap \exists\mathtt{x}^*.(\mathrm{Acc}_{\mathcal{A}_1} \sqcap \mathrm{Acc}_{\mathcal{A}_2})$. Then there exists an $\mathtt{x}^*$-path ρ from d to some $\mathrm{e} \in (\mathrm{Acc}_{\mathcal{A}_1} \sqcap \mathrm{Acc}_{\mathcal{A}_2})^{\mathcal{I}}$. Hence, by Lemma 11.11, for all $\ell \in \{1, 2\}$ the word represented by ρ belongs to $\mathcal{L}(\mathcal{A}_\ell)$, and thus $\mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$. □

By the undecidability of the non-emptiness problem for intersection of one-counter languages [Val73, p. 75], we conclude Theorem 11.13.

Theorem 11.13

The concept satisfiability problem for $\mathcal{ALC}^{\mathsf{Self}}_{\mathsf{vpl}}$ is undecidable, even if only visibly-pushdown languages that are encodings of DOCA languages are allowed in concepts.

There is nothing special about deterministic one-counter automata used in the proof. In fact, any automaton model would satisfy our needs as long as it would (i) have an undecidable non-emptiness problem for the intersection of languages, (ii) enjoy the analogue of Lemma 11.7, and (iii) be closed under complement. We leave it is an open problem to see if there exists a *single* visibly-pushdown language $\mathcal{L}$ that makes the concept satisfiability of $\mathcal{ALC}^{\mathsf{Self}}_{\mathsf{reg}}$ extended with $\mathcal{L}$ undecidable. For instance, the decidability status of $\mathcal{ALC}^{r\#s\#}_{\mathsf{reg}}$ with Self is open.

11.3 Negative results II: Nominals meet $r^{\#}s^{\#}$

We next provide an undecidability proof for the concept satisfiability problem for $\mathcal{ALCO}^{r\#s\#}_{\mathsf{reg}}$. To achieve this, we employ a slight variant of the classical domino tiling problem [Wan61].

Definition 11.14 A **domino tiling system** is a triple $\mathfrak{D} := (\mathrm{Col}, \mathrm{T}, \square)$, where Col is a finite set of **colours**, $\mathrm{T} \subseteq \mathrm{Col}^4$ is a set of 4-sided **tiles**, and $\square \in \mathrm{Col}$ is a distinguished colour called *white*. For brevity, we call a tile $(\mathrm{c}_l, \mathrm{c}_d, \mathrm{c}_r, \mathrm{c}_u) \in \mathrm{T}$ (i) *left-border* if $\mathrm{c}_l = \square$, (ii) *down-border* if $\mathrm{c}_d = \square$, (iii) *right-border* if $\mathrm{c}_r = \square$, and (iv) *up-border* if $\mathrm{c}_u = \square$. We also say that tiles $\mathrm{t} := (\mathrm{c}_l, \mathrm{c}_d, \mathrm{c}_r, \mathrm{c}_u)$ and $\mathrm{t}' := (\mathrm{c}'_l, \mathrm{c}'_d, \mathrm{c}'_r, \mathrm{c}'_u)$ from T are (i) **H-compatible** if $\mathrm{c}_r = \mathrm{c}'_l$, and (ii) **V-compatible** if $\mathrm{c}_u = \mathrm{c}'_d$. We say that $\mathfrak{D}$ **covers** $\mathbb{Z}_n \times \mathbb{Z}_m$ (where n and m are positive integers) if there exists a mapping

$\xi\colon \mathbb{Z}_n \times \mathbb{Z}_m \to \mathrm{T}$ such that for all pairs $(x, y) \in \mathbb{Z}_n \times \mathbb{Z}_m$ with $\xi(x, y) := (\mathrm{c}_l, \mathrm{c}_d, \mathrm{c}_r, \mathrm{c}_u)$ the following conditions are satisfied:

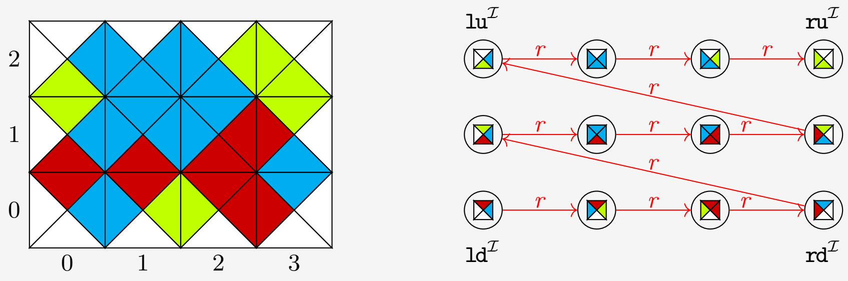
(TBorders) $x = 0$ iff $\mathrm{c}_l = \square$; $x = n{-}1$ iff $\mathrm{c}_r = \square$; $y = 0$ iff $\mathrm{c}_d = \square$; $y = m{-}1$ iff $\mathrm{c}_u = \square$;

(THori) If $(x{+}1, y) \in \mathbb{Z}_n \times \mathbb{Z}_m$ then $\xi(x, y)$ and $\xi(x{+}1, y)$ are H-compatible.

(TVerti) If $(x, y{+}1) \in \mathbb{Z}_n \times \mathbb{Z}_m$ then $\xi(x, y)$ and $\xi(x, y{+}1)$ are V-compatible.

Intuitively, $\xi\colon \mathbb{Z}_n \times \mathbb{Z}_m$ can be seen as a white-bordered rectangle of size $n \times m$ coloured by unit 4-sided tiles (with coordinates corresponding to the left, down, right, and upper colour) from T, where sides of tiles of consecutive squares have matching colours. Consider the following example:

Example 11.15. Suppose that Col $= \{$🟦, 🟥, □, 🟩$\}$ and $\mathrm{T} = \mathrm{Col}^4$. Then the map $\xi := \{(0,0) \mapsto$ ⊠, $(1,0) \mapsto$ ⊠, $(2,0) \mapsto$ ⊠, $(3,0) \mapsto$ ⊠, $(0,1) \mapsto$ ⊠, $(1,1) \mapsto$ ⊠, $(2,1) \mapsto$ ⊠, $(3,1) \mapsto$ ⊠, $(0,2) \mapsto$ ⊠, $(1,2) \mapsto$ ⊠, $(2,2) \mapsto$ ⊠, $(3,2) \mapsto$ ⊠$\}$ covers $\mathbb{Z}_4 \times \mathbb{Z}_3$, and can be visualised as follows.



W.l.o.g. we assume that T does not contain tiles having more than 2 white sides. A system $\mathcal{D}$ is *solvable* if there exist positive integers $n, m \in \mathbb{N}$ for which $\mathcal{D}$ covers $\mathbb{Z}_n \times \mathbb{Z}_m$. The problem of deciding whether an input domino tiling system is solvable is undecidable, which can be shown by a minor modification of classical undecidability proofs [PH23, Lemma 3.9][Boa97]. For a domino tiling system $\mathcal{D} := (\mathrm{Col}, \mathrm{T}, \square)$ we employ fresh concept names from $\mathrm{C}^{\mathrm{T}}_{🧩} := \{\mathrm{C}_{\mathrm{t}} \mid \mathrm{t} \in \mathrm{T}\}$ to encode mappings ξ from some $\mathbb{Z}_n \times \mathbb{Z}_m$ to T in interpretations $\mathcal{I}$ as certain r^+-paths ρ from $\mathtt{ld}^{\mathcal{I}}$ to $\mathtt{ru}^{\mathcal{I}}$ passing through $\mathtt{rd}^{\mathcal{I}}$ and $\mathtt{lu}^{\mathcal{I}}$ (where the individual names from $\mathsf{N}^{\mathrm{T}}_{🧩} := \{\mathtt{ld}, \mathtt{rd}, \mathtt{lu}, \mathtt{ru}\}$ are fresh). Consult the figure in Example 11.15.

Definition 11.16 Consider a domino tiling system $\mathcal{D} := (\mathrm{Col}, \mathrm{T}, \square)$. An interpretation $\mathcal{I}$ is a **$\mathcal{D}$-snake** whenever all seven criteria listed below are fulfilled:

(SPath) There is an r^+-path ρ that starts in $\mathtt{ld}^{\mathcal{I}}$, then passes through $\mathtt{rd}^{\mathcal{I}}$, then passes through $\mathtt{lu}^{\mathcal{I}}$ and finishes in $\mathtt{ru}^{\mathcal{I}}$. More formally, there are indices $1 < i < j < |\rho|$ such that $\rho_1 = \mathtt{ld}^{\mathcal{I}}$, $\rho_i = \mathtt{rd}^{\mathcal{I}}$, $\rho_j = \mathtt{lu}^{\mathcal{I}}$ and $\rho_{|\rho|} = \mathtt{ru}^{\mathcal{I}}$.

(SNoLoop) No $\mathsf{N}^{\mathrm{T}}_{🧩}$-named element can r^+-reach itself.

(SUniqTil) For every element d that is r^*-reachable from $\mathtt{ld}^{\mathcal{I}}$ there exists precisely one tile $\mathrm{t} \in \mathrm{T}$ such that $\mathrm{d} \in \mathrm{C}^{\mathcal{I}}_{\mathrm{t}}$ (we say that d is *labelled* by a tile t or that d *carries* t).

(SSpecTil) The $\mathsf{N}^{\mathrm{T}}_{🧩}$-named elements are unique elements r^*-reachable from $\mathtt{ld}^{\mathcal{I}}$ that are labelled by tiles with two white sides. Moreover, we have that (a) $\mathtt{ld}^{\mathcal{I}}$ carries a tile that is left-border and down-border, (b) $\mathtt{rd}^{\mathcal{I}}$ carries a tile that is right-border and down-border, (c) $\mathtt{lu}^{\mathcal{I}}$ carries a tile that is left-border and up-border, (d) $\mathtt{ru}^{\mathcal{I}}$ carries a tile that is right-border and up-border.

(SHori) For all elements d different from $\mathtt{ru}^{\mathcal{I}}$ that are r^*-reachable from $\mathtt{ld}^{\mathcal{I}}$ and labelled by some tile $\mathrm{t} := (\mathrm{c}_l, \mathrm{c}_d, \mathrm{c}_r, \mathrm{c}_u)$, there exists a tile $\mathrm{t}' := (\mathrm{c}'_l, \mathrm{c}'_d, \mathrm{c}'_r, \mathrm{c}'_u)$ for which all r-successors e of d carry the tile t' and: (i) t, t' are H-compatible, (ii) if $\mathrm{c}_d = \square$ then ($\mathrm{c}_r \neq \square$ iff $\mathrm{c}'_d = \square$), and (iii) if $\mathrm{c}_u = \square$ then $\mathrm{c}'_u = \square$.

(SLen) There exists a unique positive integer N such that all r^+-paths between $\mathtt{ld}^{\mathcal{I}}$ and $\mathtt{rd}^{\mathcal{I}}$ are of length N$-$1. Moreover, $\mathtt{rd}^{\mathcal{I}}$ is the only element $r^{\mathrm{N}-1}$-reachable from $\mathtt{ld}^{\mathcal{I}}$.

(SVerti) For all elements d that are r^*-reachable from $\mathtt{ld}^{\mathcal{I}}$ and labelled by some $\mathrm{t} \in \mathrm{T}$ that is not up-border, we have that (a) there exists a tile $\mathrm{t}' \in \mathrm{T}$ such that all elements e r^{N}-reachable (for the N guaranteed by (SLen)) from d carry t', (b) t and t' are V-compatible, (c) t is left-border (resp. right-border) if and only if t' is.

Note that tiles are not "deterministic" in the following sense: it could happen that two elements carry the same tile but tiles of their (horizontal or vertical) successors do not coincide.

If $\mathcal{I}$ satisfies all but the last two conditions of Definition 11.16, we call it a $\mathcal{D}$-*pseudosnake*. The key properties of our encoding are extracted and established in Lemmas 11.17–11.18.

Lemma 11.17 If a domino tiling system $\mathcal{D}$ is solvable then there exists a $\mathcal{D}$-snake.

Proof. Suppose that $\mathcal{D}$ covers $\mathbb{Z}_n \times \mathbb{Z}_m$ and let ξ be a mapping witnessing it. Define an interpretation $\mathcal{I}$ as follows:

(i) $\Delta^{\mathcal{I}} := \mathbb{Z}_{n \cdot m}$,

(ii) $\mathtt{ld}^{\mathcal{I}} := 0$, $\mathtt{rd}^{\mathcal{I}} := n{-}1$, $\mathtt{lu}^{\mathcal{I}} := (m{-}1) \cdot n$, $\mathtt{ru}^{\mathcal{I}} := m \cdot n - 1$, and $\mathsf{a}^{\mathcal{I}} := 0$ for all $\mathsf{a} \in \mathbf{N_I} \setminus \mathsf{N}^{\mathrm{T}}_{\text{🧩}}$.

(iii) $\mathrm{C}^{\mathcal{I}}_{\mathrm{t}} := \{(x + y \cdot n) \in \Delta^{\mathcal{I}} \mid \xi(x, y) = \mathrm{t}\}$ for all $\mathrm{t} \in \mathrm{T}$, and $\mathrm{C}^{\mathcal{I}} := \emptyset$ for $\mathrm{C} \in \mathbf{N_C} \setminus \mathrm{C}^{\mathrm{T}}_{\text{🧩}}$,

(iv) $r^{\mathcal{I}} := \{(i, i{+}1) \mid i \in \mathbb{Z}_{n \cdot m - 1}\}$, and $(r')^{\mathcal{I}} := \emptyset$ for all $r' \in \mathbf{N_R} \setminus \{r\}$.

Thus $\mathcal{I}$ is an $n{\cdot}m$ element r^+-path, labelled accordingly to ξ. As ξ respects (TBorders), (THori) and (TVerti), we can readily verify that $\mathcal{I}$ is indeed a $\mathcal{D}$-snake. The only case that requires treatment, is to verify the satisfaction of (SHori) for elements of the form $\mathrm{d} := (n{-}1) + y \cdot n$ for some $y \in \mathbb{N}$. Then, by (TBorders), d carries a right-border tile and its r-successor $\mathrm{d}' := 0 + (y{+}1) \cdot n$ carries a left-border tile. Hence, their tiles are H-compatible. □

Lemma 11.18 If there exists a $\mathcal{D}$-snake for a domino tiling system $\mathcal{D}$, then $\mathcal{D}$ is solvable.

Proof. Suppose that $\mathcal{I}$ is a $\mathcal{D}$-snake. Let $\rho := \rho_1 \ldots \rho_{|\rho|}$ be the path guaranteed by (SPath) and let N be the integer guaranteed by (SLen). We show by induction that $|\rho|$ is divisible by N. The inductive assumption states that for all integers $k \in \mathbb{N}$ with $k \cdot \mathrm{N} \leq \|\rho\|$ we have:

(i) $\rho_{k \cdot \mathrm{N}+1}$ carries a left-border tile,

(ii) There is no $2 \leq i < \mathrm{N}$ such that $\rho_{k \cdot \mathrm{N}+i}$ carries a left-border tile or a right-border tile,

(iii) $\rho_{k \cdot \mathrm{N}+\mathrm{N}}$ carries a right-border tile.

Then by Property (iii) and the fact that $\rho_{|\rho|}$ (equal to $\mathtt{ru}^{\mathcal{I}}$ by (SPath)) carries a right-border tile (by Property (d) of (SSpecTil)), we can conclude that $|\rho|$ is indeed divisible by N.

We heavily rely on the fact that every element of ρ is labelled by precisely one tile, which is due to (SUniqTil). We start with the case of $k = 0$. Then $\rho_{0 \cdot \mathrm{N}+1} = \rho_1$ is equal to $\mathtt{ld}^{\mathcal{I}}$, by (SPath). Moreover, ρ_1 is labelled with a left-border tile, by Property (a) of (SSpecTil). What is more, $\rho_{0 \cdot \mathrm{N}+\mathrm{N}} = \rho_{\mathrm{N}}$ is equal to $\mathtt{rd}^{\mathcal{I}}$ (by (SLen)), which carries a right-border tile by Property (b) of (SSpecTil). This resolves Properties (i) and (iii). To establish Property (ii), assume towards a contradiction that there is i between 2 and N for which ρ_i carries a left-border tile (the proof for a right-border tile is analogous). Take the smallest such i. By (SHori) we infer that $\rho_{(i-1)}$ carries a right-border tile. In particular, this means that $i > 2$ because the tile carried by ρ_1 is not right-border. By exhaustive application of (SHori) and the fact that the tile of ρ_1 is down-border, we deduce that the tile of $\rho_{(i-1)}$ is also down-border. Hence, by Property (b) of (SSpecTil) we have that $\rho_{(i-1)}$ is equal to $\mathtt{rd}^{\mathcal{I}}$. But then the path $\rho_{(i-1)}\rho_i \ldots \rho_{\mathrm{N}}$ witnesses r^+-reachability of $\mathtt{rd}^{\mathcal{I}}$ from itself, which is forbidden by (SNoLoop). A contradiction. For the inductive step, assume that Properties (i)–(iii) hold true for some k, and consider the case of $k{+}1$. Note that Property (i) follows from Property (iii) of the inductive assumption by (SHori). We next show that Property (ii) holds. Assume ad absurdum that there is i for which $\rho_{(k+1) \cdot \mathrm{N}+i}$ carries a left-border (resp. right-border) tile. But then, invoking Item (c) of (SVerti), we infer that $\rho_{k \cdot \mathrm{N}+i}$ is also left-border (resp. right-border). This contradicts Property (ii) of the inductive assumption. Hence Property (ii) holds true. By inductive assumption, we know that $\rho_{k\cdot\mathrm{N}+\mathrm{N}}$ carries a right-border tile. Then, we apply Property (c) of (SVerti) to infer that $\rho_{k\cdot\mathrm{N}+\mathrm{N}+\mathrm{N}} = \rho_{(k+1)\cdot\mathrm{N}+\mathrm{N}}$ is right-border, as desired. This establishes Property (iii), and concludes the induction.

Let $\mathrm{M} := |\rho|/\mathrm{N}$. By the previous claim, we know that $\mathrm{M} \in \mathbb{N}$. Consider a function $\xi \colon \mathbb{Z}_{\mathrm{N}} \times \mathbb{Z}_{\mathrm{M}} \to \mathrm{T}$ that maps all (x, y) to the unique tile carried by $\rho_{x+\mathrm{N}\cdot y+1}$. (Note that we number paths from 1!) This function is well-defined by (SUniqTil) and it satisfies (THori) and (TVerti) due to the satisfaction of (SHori) and (SVerti). The satisfaction of the first two statements of (TBorders) by ξ is guaranteed by Properties (i)–(iii) from the induction above. Finally, the last two statements of (TBorders) are due to straightforward induction that employs (SSpecTil) and the last statement of (SHori). As we proved that ξ covers $\mathbb{Z}_{\mathrm{N}} \times \mathbb{Z}_{\mathrm{M}}$, we conclude that $\mathcal{D}$ is indeed solvable. $\square$

While $\mathcal{D}$-snakes do not seem to be directly axiomatizable even in $\mathcal{ALC}_{\mathsf{vpl}}$, we at least see how to express $\mathcal{D}$-pseudosnakes in $\mathcal{ALCO}^{r\#s\#}_{\mathsf{reg}}$. The next lemma is routine.

Lemma 11.19 For every domino tiling system $\mathcal{D} := (\mathrm{Col}, \mathrm{T}, \square)$, there exists an $\mathcal{ALCO}^{r\#s\#}_{\mathsf{reg}}$-concept $\mathrm{C}^{\mathcal{D}}_{\mathcal{S}}$, that employs only the role r, individual names from $\mathsf{N}^{\mathrm{T}}_{\maltese}$ and concept names from $\mathrm{C}^{\mathrm{T}}_{\maltese}$, such that for all interpretations $\mathcal{I}$ we have that $\mathcal{I}$ is a $\mathcal{D}$-pseudosnake iff $\mathcal{I} \models \mathrm{C}^{\mathcal{D}}_{\mathcal{S}}$.

Proof. We present a rather straightforward axiomatization of the aforementioned properties, written from the point of view of the interpretation of a nominal $\mathtt{ld}$.

$$\mathrm{C}_{\text{(SPath)}} := \{\mathtt{ld}\} \sqcap \exists r^{+}.\big(\{\mathtt{rd}\} \sqcap \exists r^{+}.\big(\{\mathtt{lu}\} \sqcap \exists r^{+}.\{\mathtt{ru}\}\big)\big).$$

$$\mathrm{C}_{\text{(SNoLoop)}} := \bigsqcap_{\mathsf{a}\in\mathsf{N}^{\mathrm{T}}_{\maltese}} \forall r^{*}.\Big[\{\mathsf{a}\} \to \forall r^{+}.\neg\{\mathsf{a}\}\Big].$$

$$\mathrm{C}_{\text{(SUniqTil)}} := \forall r^{*}\Big[\bigsqcup_{\mathrm{C}\in\mathrm{C}^{\mathrm{T}}_{\maltese}}\Big(\mathrm{C} \sqcap \bigsqcap_{\mathrm{C}'\in\mathrm{C}^{\mathrm{T}}_{\maltese},\mathrm{C}'\neq\mathrm{C}} \neg\mathrm{C}'\Big)\Big].$$

$$\mathrm{C}_{\text{(SSpecTil)}} := \forall r^{*}\Big[\Big(\bigsqcup_{\mathrm{t}\in\mathrm{T}\ \text{with two white sides}} \mathrm{C}_{\mathrm{t}}\Big) \leftrightarrow \bigsqcup_{\mathsf{a}\in\mathsf{N}^{\mathrm{T}}_{\maltese}}\{\mathsf{a}\}\Big] \sqcap \mathrm{C}^{\text{down}}_{\text{(SSpecTil)}} \sqcap \mathrm{C}^{\text{up}}_{\text{(SSpecTil)}},\ \text{where}$$

$$\mathrm{C}^{\text{down}}_{\text{(SSpecTil)}} := \Big(\{\mathtt{ld}\} \sqcap \bigsqcup_{\mathrm{t}:=(\square,\square,c_r,c_u)\in\mathrm{T}} \mathrm{C}_{\mathrm{t}}\Big) \sqcap \exists r^{*}.\Big(\{\mathtt{rd}\} \sqcap \bigsqcup_{\mathrm{t}:=(c_l,\square,\square,c_u)\in\mathrm{T}} \mathrm{C}_{\mathrm{t}}\Big),$$

$$\mathrm{C}^{\text{up}}_{\text{(SSpecTil)}} := \exists r^{*}.\Big(\{\mathtt{lu}\} \sqcap \bigsqcup_{\mathrm{t}:=(\square,c_d,c_r,\square)\in\mathrm{T}} \mathrm{C}_{\mathrm{t}}\Big) \sqcap \exists r^{*}.\Big(\{\mathtt{ru}\} \sqcap \bigsqcup_{\mathrm{t}:=(c_l,c_d,\square,\square)\in\mathrm{T}} \mathrm{C}_{\mathrm{t}}\Big).$$

$$\mathrm{C}_{\text{(SHori)}} := \bigsqcap_{\mathrm{t}\in\mathrm{T}} \forall r^{*}.\Big[(\neg\{\mathtt{ru}\} \sqcap \mathrm{C}_{\mathrm{t}}) \to \Big((\exists r.\top) \sqcap \bigsqcup_{\mathrm{t}'\in\mathrm{T}\ \text{satisfying cond. (i)–(iii) of (SHori)}} \forall r.\mathrm{C}_{\mathrm{t}'}\Big)\Big].$$

We can now define $\mathrm{C}^{\mathcal{D}}_{\mathcal{S}}$ as the conjunction of all the concept definitions presented above. It follows immediately from the semantics of $\mathcal{ALCO}^{r\#s\#}_{\mathsf{reg}}$ that the presented concept definition is consistent if and only there exists an element starting a pseudosnake. $\square$

Note that the property that pseudosnakes are missing in order to be proper snakes, is the ability to measure. We tackle this issue by introducing a gadget called a "yardstick".

Definition 11.20 Let T be a finite non-empty set, let $\mathsf{N}^{\mathrm{T}}_{📏} := \{\mathtt{st}, \mathtt{md}, \mathtt{md}_{\mathrm{t}}, \mathtt{end}_{\mathrm{t}} \mid \mathrm{t} \in \mathrm{T}\}$ be composed of (pairwise different) individual names. A **T-yardstick** is any interpretation $\mathcal{I}$ satisfying all the conditions below.

(YNom) $\mathsf{N}^{\mathrm{T}}_{📏}$-named elements in $\mathcal{I}$ are pairwise different and $(r+s)^*$-reachable from $\mathtt{st}^{\mathcal{I}}$.

(YNoLoop) No $\mathsf{N}^{\mathrm{T}}_{📏}$-named element can $(r+s)^+$-reach itself.

(YMid) $\mathtt{md}^{\mathcal{I}}$ is the *unique* element with an s-successor that is r^*-reachable from $\mathtt{st}^{\mathcal{I}}$.

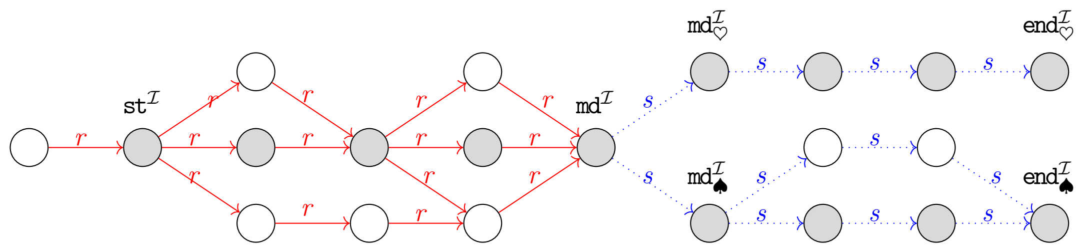
(YSuccOfMid) The s-successors of $\mathtt{md}^{\mathcal{I}}$ are precisely $\{\mathtt{md}_{\mathrm{t}} \mid \mathrm{t} \in \mathrm{T}\}$-named elements.

(YReachMidT) For every $\mathrm{t} \in \mathrm{T}$ we have that $\mathtt{md}_{\mathrm{t}}^{\mathcal{I}}$ can s^*-reach $\mathtt{end}_{\mathrm{t}}^{\mathcal{I}}$ but it cannot s^*-reach $\mathtt{end}_{\mathrm{t}'}^{\mathcal{I}}$ for all $\mathrm{t}' \neq \mathrm{t}$.

(YEqDst) The elem. $r^{\#}s^{\#}$-reachable from $\mathtt{st}^{\mathcal{I}}$ are precisely the $\{\mathtt{end}_{\mathrm{t}} \mid \mathrm{t} \in \mathrm{T}\}$-named ones.

(YNoEqDst) No $\{\mathtt{end}_{\mathrm{t}} \mid \mathrm{t} \in \mathrm{T}\}$-named element is $r^{\#}s^{\#}$-reachable from an element $(s+r)^+$-reachable from $\mathtt{st}^{\mathcal{I}}$.

An example $\{\heartsuit, \spadesuit\}$-yardstick is depicted below. A "minimal" yardstick contains the grey nodes only.


$\mathtt{st}^{\mathcal{I}}$
$\mathtt{md}^{\mathcal{I}}$
$\mathtt{md}_{\heartsuit}^{\mathcal{I}}$
$\mathtt{end}_{\heartsuit}^{\mathcal{I}}$
$\mathtt{md}_{\spadesuit}^{\mathcal{I}}$
$\mathtt{end}_{\spadesuit}^{\mathcal{I}}$
r
s

The forthcoming lemma explains the name "yardstick". Intuitively it says that in any T-yardstick $\mathcal{I}$, all s^*-paths from $\mathtt{md}^{\mathcal{I}}$ to all $\mathtt{end}_{\mathrm{t}}^{\mathcal{I}}$ have equal length.

Lemma 11.21 Let $\mathcal{I}$ be a T-yardstick. Then there exists a unique positive integer N such that: (i) for all $\mathrm{t} \in \mathrm{T}$ we have that $\mathtt{end}_{\mathrm{t}}^{\mathcal{I}}$ is s^{N}-reachable from $\mathtt{md}^{\mathcal{I}}$, and (ii) for all $\mathrm{t} \in \mathrm{T}$ we have that $\mathtt{end}_{\mathrm{t}}^{\mathcal{I}}$ is $s^{\mathrm{N}-1}$-reachable from $\mathtt{md}_{\mathrm{t}}^{\mathcal{I}}$. We will call N the *length* of $\mathcal{I}$.

Proof. Fix $\mathrm{t}_{\star} \in \mathrm{T}$. By (YEqDst) we know that $\mathtt{st}^{\mathcal{I}}$ $r^{\#}s^{\#}$-reaches $\mathtt{end}_{\mathrm{t}_{\star}}^{\mathcal{I}}$, and let $\rho := \rho_1 \ldots \rho_{2\mathrm{N}+1}$ be a path witnessing it. We claim that this is the desired length of $\mathcal{I}$. First, note that $\mathrm{N} > 0$ by (YNom). Second, by the semantics of $r^{\#}s^{\#}$, for all $i \leq \mathrm{N}$ we have $(\rho_i, \rho_{i+1}) \in r^{\mathcal{I}}$ and $(\rho_{\mathrm{N}+i}, \rho_{\mathrm{N}+i+1}) \in s^{\mathcal{I}}$. Thus $\rho_{\mathrm{N}+1}$ is r^*-reachable from $\mathtt{st}^{\mathcal{I}}$ and has an s-successor. These two facts imply (by (YMid)) that $\rho_{\mathrm{N}+1}$ is equal to $\mathtt{md}^{\mathcal{I}}$. It remains to show that all the paths leading from $\mathtt{md}^{\mathcal{I}}$ to some $\mathtt{end}_{\mathrm{t}}$ are of length N. Towards a contradiction, assume that there is $\mathrm{t}' \in \mathrm{T}$ and an integer $\mathrm{M} \neq \mathrm{N}$ such that $\mathtt{md}^{\mathcal{I}}$ s^{M}-reaches $\mathtt{end}_{\mathrm{t}'}^{\mathcal{I}}$ via a path $\rho' := \rho'_1 \ldots \rho'_{\mathrm{M}}$. We stress that $\rho'_1 = \mathtt{md}^{\mathcal{I}}$ and $\rho'_{\mathrm{M}} = \mathtt{end}_{\mathrm{t}'}^{\mathcal{I}}$ (by design of ρ'), and $\rho'_2 = \mathtt{md}_{\mathrm{t}'}^{\mathcal{I}}$ (by a conjunction of (YSuccOfMid) and (YReachMidT)). To finish the proof, we resolve the following two cases.

- Suppose that $\mathrm{M} < \mathrm{N}$. Then $\rho_{\mathrm{N}+1-\mathrm{M}}$ $(r^{\mathrm{M}}s^{\mathrm{M}})$-reaches (thus also $r^{\#}s^{\#}$-reaches) $\mathtt{end}_{\mathrm{t}'}^{\mathcal{I}}$, as witnessed by the path $\rho_{\mathrm{N}+1-\mathrm{M}} \ldots \rho_{\mathrm{N}}\rho'$. Moreover $\rho_{\mathrm{N}+1-\mathrm{M}}$ is r^+-reachable from $\mathtt{st}^{\mathcal{I}}$, witnessed by the path $\rho_1 \ldots \rho_{\mathrm{N}+1-\mathrm{M}}$ (note that its length is positive by the inequality $\mathrm{M} < \mathrm{N}$). This yields a contradiction with (YNoEqDst).
- Suppose that $\mathrm{M} > \mathrm{N}$. Consider a path $\rho_1 \ldots \rho_{\mathrm{N}}\rho'_1 \ldots \rho'_{\mathrm{N}}$. By construction, such a path witnesses the fact that $\mathtt{st}^{\mathcal{I}}$ $(r^{\mathrm{N}}s^{\mathrm{N}})$-reaches (and thus also $r^{\#}s^{\#}$-reaches) ρ'_{N}. By (YEqDst) we infer that ρ'_{N} is then $\{\mathtt{end}_{\mathrm{t}} \mid \mathrm{t} \in \mathrm{T}\}$-named. As $\rho'_2 = \mathtt{md}_{\mathrm{t}'}^{\mathcal{I}}$ s^+-reaches ρ'_{N}, we infer that $\rho'_{\mathrm{N}} = \mathtt{end}_{\mathrm{t}'}^{\mathcal{I}}$ (otherwise we would have a contradiction with (YReachMidT)). But then $\mathtt{end}_{\mathrm{t}'}^{\mathcal{I}}$ s^+-reaches itself via a path $\rho'_{\mathrm{N}} \ldots \rho_{\mathrm{M}}$, which is of positive length by the fact that $\mathrm{M} > \mathrm{N}$. This yields a contradiction with (YNoLoop).

This establishes Property (i). Property (ii) is now immediate by (YSuccOfMid). □

The next lemma proves the existence of arbitrary large yardsticks.

Lemma 11.22 For every finite non-empty set T and a positive integer N, there exists a T-yardstick of length N.

Proof. Consider the following interpretation $\mathcal{I}$ with $\Delta^{\mathcal{I}} := \mathbb{Z}_{\mathrm{N}} \cup \{\mathrm{N}\} \cup (\mathbb{Z}_{\mathrm{N}-1} \times \mathrm{T})$:

- $\mathtt{st}^{\mathcal{I}} := 0$, $\mathtt{md}^{\mathcal{I}} := \mathrm{N}$, $(\mathtt{md}_{\mathrm{t}})^{\mathcal{I}} := (0, \mathrm{t})$, $(\mathtt{end}_{\mathrm{t}})^{\mathcal{I}} := (\mathrm{N}{-}1, \mathrm{t})$ for all $\mathrm{t} \in \mathrm{T}$, and $\mathtt{a}^{\mathcal{I}} := 0$ for all names $\mathtt{a} \in \mathbf{N_I} \setminus \mathsf{N}^{\mathrm{T}}_{\text{📏}}$.
- $r^{\mathcal{I}} := \{(i, i{+}1) \mid i \in \mathbb{Z}_{\mathrm{N}}\}$, $s^{\mathcal{I}} := \{(\mathrm{N}, (0, \mathrm{t})), ((i, \mathrm{t}), (i{+}1, \mathrm{t})) \mid \mathrm{t} \in \mathrm{T}, i \in \mathbb{Z}_{\mathrm{N}-1}\}$, and $(r')^{\mathcal{I}} = \emptyset$ for all role names $r' \in \mathbf{N_R} \setminus \{r, s\}$.

An example such $\mathcal{I}$ for $\mathrm{N} = 3$ and $\mathrm{T} = \{\heartsuit, \spadesuit\}$ is depicted above (in restriction to grey nodes only). It is routine to verify that $\mathcal{I}$ satisfies all the properties in Definition 11.20, as well as conditions (i) and (ii) from the statement of Lemma 11.21. This concludes the proof. ☐

To make use of yardsticks in our proofs, we need to axiomatise them inside $\mathcal{ALCO}^{r\#s\#}_{\mathsf{reg}}$.

Lemma 11.23 There exists an $\mathcal{ALCO}^{r\#s\#}_{\mathsf{reg}}$-concept $\mathrm{C}^{\mathrm{T}}_{\text{📏}}$, that employs only role names r, s and individual names from $\mathsf{N}^{\mathrm{T}}_{\text{📏}}$, such that for all interpretations $\mathcal{I}$ we have: $\mathcal{I}$ is a T-yardstick if and only if $\mathcal{I}$ is a model of $\mathrm{C}^{\mathrm{T}}_{\text{📏}}$.

Proof. The following concepts are written from the point of view of the interpretation of $\mathtt{st}$.

$$\mathrm{C}_{\text{(YNom)}} := \bigsqcap_{\mathtt{a} \in \mathsf{N}^{\mathrm{T}}_{\text{📏}}} \exists (r+s)^* \Big[\{\mathtt{a}\} \sqcap \bigsqcap_{\mathtt{b} \in \mathsf{N}^{\mathrm{T}}_{\text{📏}} \setminus \{\mathtt{a}\}} \neg\{\mathtt{b}\}\Big].$$

$$\mathrm{C}_{\text{(YNoLoop)}} := \bigsqcap_{\mathtt{a} \in \mathsf{N}^{\mathrm{T}}_{\text{📏}}} \forall (r+s)^*.\Big[\{\mathtt{a}\} \to \forall (r+s)^+.\neg\{\mathtt{a}\}\Big].$$

$$\mathrm{C}_{\text{(YMid)}} := \Big[\exists r^*.\,((\exists s.\top) \sqcap \{\mathtt{md}\})\Big] \sqcap \Big[\forall r^*.\,((\exists s.\top) \to \{\mathtt{md}\})\Big].$$

$$\mathrm{C}_{\text{(YSuccOfMid)}} := \forall (r+s)^*.\left(\{\mathtt{md}\} \to \Big[\bigsqcap_{\mathtt{a} \in \{\mathtt{md}_{\mathrm{t}} \mid \mathrm{t} \in \mathrm{T}\}} \exists s.\{\mathtt{a}\}\Big] \sqcap \Big[\forall s. \bigsqcup_{\mathtt{a} \in \{\mathtt{md}_{\mathrm{t}} \mid \mathrm{t} \in \mathrm{T}\}} \{\mathtt{a}\}\Big]\right).$$

$$\mathrm{C}_{\text{(YReachMidT)}} := \bigsqcap_{\mathrm{t} \in \mathrm{T}} \forall (r+s)^*.\{\mathtt{md}_{\mathrm{t}}\} \to \Big[(\exists s^*.\{\mathtt{end}_{\mathrm{t}}\}) \sqcap \bigsqcap_{\mathrm{t}' \in \mathrm{T}, \mathrm{t} \neq \mathrm{t}'} \forall s^*.\neg\{\mathtt{end}_{\mathrm{t}'}\}\Big].$$

$$\mathrm{C}_{\text{(YEqDst)}} := \left(\bigsqcap_{\mathrm{t} \in \mathrm{T}} \exists r^{\#}s^{\#}.\{\mathtt{end}_{\mathrm{t}}\}\right) \sqcap \forall r^{\#}s^{\#}.\left(\bigsqcup_{\mathrm{t} \in \mathrm{T}} \{\mathtt{end}_{\mathrm{t}}\}\right).$$

$$\mathrm{C}_{\text{(YNoEqDst)}} := \forall (r+s)^+.\forall r^{\#}s^{\#}.\left(\bigsqcap_{\mathrm{t} \in \mathrm{T}} \neg\{\mathtt{end}_{\mathrm{t}}\}\right).$$

We define $\mathrm{C}^{\mathrm{T}}_{\text{📏}}$ as the conjunction of $\{\mathtt{st}\}$ and all the concept definitions presented above. By semantics of $\mathcal{ALCO}^{r\#s\#}_{\mathsf{reg}}$ we have that $\mathrm{C}^{\mathrm{T}}_{\text{📏}}$ is consistent if and only if $(\mathrm{C}^{\mathrm{T}}_{\text{📏}})^{\mathcal{I}} = \{\mathtt{st}^{\mathcal{I}}\}$. ☐

We next put pseudosnakes and yardsticks together, obtaining metricobras. The intuition behind their construction is fairly simple: (i) we take a disjoint union of a pseudosnake and a yardstick, (ii) we then connect (via the role s) every element carrying a tile t with the interpretation of the corresponding nominal $\mathtt{md}_{\mathrm{t}}$, and finally (iii) we synchronise the length of the underlying yardstick, say N, with the length of the path between the interpretations of $\mathtt{ld}$ and $\mathtt{rd}$. After such "merging", retrieving (SVerti) is easy: rather than testing if every N-reachable element from some d carries a suitable tile t (for an a priori unknown N) we can check instead whether d can $r^{\#}s^{\#}$-reach the interpretation of $\mathtt{end}_{\mathrm{t}}$.

A formal definition and a picture come next.

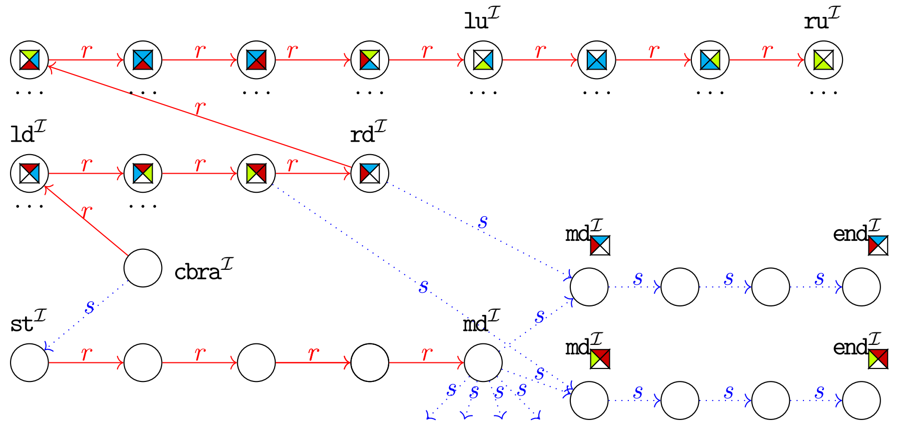


Figure 11.2: A fragment of an example $\mathcal{D}$-metricobra representing ξ from Example 11.15. The upper part corresponds to a $\mathcal{D}$-snake, and the lower part corresponds to a T-yardstick. The distances between named elements are important.

Definition 11.24 Let $\mathcal{D} := (\mathrm{Col}, \mathrm{T}, \Box)$ be a domino tiling system and cbra be an individual name. An interpretation $\mathcal{I}$ is a ***$\mathcal{D}$*-metricobra** if all the conditions below are satisfied:

(MInit) $\mathcal{I}$ is a $\mathcal{D}$-pseudosnake and a T-yardstick, and $\mathsf{cbra}^{\mathcal{I}}$ has precisely two successors: one r-successor, namely $\mathtt{ld}^{\mathcal{I}}$, and one s-successor, namely $\mathtt{st}^{\mathcal{I}}$.

(MTile) For every tile $\mathrm{t} \in \mathrm{T}$ and every element $\mathrm{d} \in \Delta^{\mathcal{I}}$ that is r^*-reachable from $\mathtt{ld}^{\mathcal{I}}$ we have that d carries a tile $\mathrm{t} \in \mathrm{T}$ if and only if d has a unique s-successor and such a successor is equal to $\mathsf{md}_{\mathrm{t}}^{\mathcal{I}}$.

(MSync) Let $\mathrm{t} \in \mathrm{T}$ be the tile labelling $\mathtt{rd}^{\mathcal{I}}$. Then (a) $\mathsf{cbra}^{\mathcal{I}}$ $r^{\#}s^{\#}$-reaches $\mathsf{end}_{\mathrm{t}}^{\mathcal{I}}$ and cannot $r^{\#}s^{\#}$-reach any of $\mathsf{end}_{\mathrm{t}'}^{\mathcal{I}}$ for $\mathrm{t}' \neq \mathrm{t}$, (b) $\mathsf{cbra}^{\mathcal{I}}$ cannot $r^{\#}s^{\#}$-reach an element that can s^{+}-reach $\mathsf{end}_{\mathrm{t}}^{\mathcal{I}}$, (c) no element r^*-reachable from $\mathtt{ld}^{\mathcal{I}}$ can $r^{\#}s^{\#}$-reach $\mathsf{end}_{\mathrm{t}}^{\mathcal{I}}$.

(MVerti) For all elements d that are r^*-reachable from $\mathtt{ld}^{\mathcal{I}}$ and are labelled by some $\mathrm{t} \in \mathrm{T}$ that is not up-border, we have that there exists a tile $\mathrm{t}' \in \mathrm{T}$ such that (a) t and t′ are V-compatible, (b) t is left-border (resp. right-border) iff t′ is, and (c) d can $r^{\#}s^{\#}$-reach $\mathsf{end}_{\mathrm{t}'}$ but cannot reach $r^{\#}s^{\#}$-reach $\mathsf{end}_{\mathrm{t}''}$ for all $\mathrm{t}'' \neq \mathrm{t}'$.

We first show that $\mathcal{D}$-metricobras are axiomatizable in $\mathcal{ALCO}_{\mathsf{reg}}^{r\#s\#}$.

Lemma 11.25 There exists an $\mathcal{ALCO}_{\mathsf{reg}}^{r\#s\#}$-concept $\mathrm{C}^{\mathcal{D}}$ such that for all interpretations $\mathcal{I}$ we have that $\mathcal{I}$ is a $\mathcal{D}$-metricobra if and only if $(\mathrm{C}^{\mathcal{D}})^{\mathcal{I}} = \{\mathsf{cbra}^{\mathcal{I}}\}$.

Proof. We present a rather straightforward axiomatization of the aforementioned properties, written from the point of view of the interpretation of cbra. Note that as the interpretation of cbra has a unique r-successor, we can simplify concepts of the form $\forall r.[\{\mathtt{ld}\} \to (\forall r^*.\mathrm{C})]$ to $\forall r^{+}.\mathrm{C}$, which we frequently do below.

$$\mathrm{C}_{\text{(MInit)}} := \exists r.\mathrm{C}^{\mathcal{D}} \sqcap \exists s.\mathrm{C}^{\mathrm{T}} \sqcap \forall r.(\{\mathtt{ld}\} \sqcap \neg\{\mathtt{st}\}) \sqcap \forall s.(\neg\{\mathtt{ld}\} \sqcap \{\mathtt{st}\}).$$

$$\mathrm{C}_{\text{(MTile)}} := \bigsqcap_{\mathrm{t}\in\mathrm{T}} \forall r^{+}.\left(\mathrm{C}_{\mathrm{t}} \leftrightarrow [\exists s.\{\mathsf{md}_{\mathrm{t}}\} \sqcap \forall s.\{\mathsf{md}_{\mathrm{t}}\}]\right).$$

$$\mathrm{C}_{\text{(MVerti)}} := \forall r^{+}. \overset{\text{not up-border}}{\bigsqcap_{\mathrm{t}\in\mathrm{T}}} \left(\mathrm{C}_{\mathrm{t}} \to \overset{\text{sat. (a),(b) of (MVerti)}}{\bigsqcup_{\mathrm{t}'\in\mathrm{T}}} \left[(\exists r^{\#}s^{\#}.\{\mathsf{end}_{\mathrm{t}'}\}) \sqcap \bigsqcap_{\mathrm{t}''\neq\mathrm{t}',\mathrm{t}''\in\mathrm{T}} \forall r^{\#}s^{\#}.\neg\{\mathsf{end}_{\mathrm{t}''}\}\right] \right).$$

$$\mathrm{C}^{\mathrm{t}}_{\text{(MSync)}} := (\exists r^{\#}s^{\#}.\{\mathsf{end}_{\mathrm{t}}\}) \sqcap \bigsqcap_{\mathrm{t}'\neq\mathrm{t}} (\forall r^{\#}s^{\#}.\neg\{\mathsf{end}_{\mathrm{t}'}\}) \sqcap (\forall r^{\#}s^{\#}.\forall s^{+}.\neg\{\mathsf{end}_{\mathrm{t}}\}) \sqcap \forall r^{+}.\forall r^{\#}s^{\#}.\neg\{\mathsf{end}_{\mathrm{t}}\}.$$

We define $C^{\mathfrak{D}}_{🐍} \coloneqq \{\texttt{cbra}\} \sqcap C_{(\text{MInit})} \sqcap C_{(\text{MTile})} \sqcap C_{(\text{MVerti})} \sqcap \bigsqcap_{t \in T}[(\exists r^{+}.(\{\texttt{rd}\} \sqcap C_t)) \rightarrow C^{t}_{(\text{MSync})}]$. Its correctness follows from the semantics of $\mathcal{ALCO}^{r\#s\#}_{\mathsf{reg}}$ and Lemmas 11.19 and 11.23. ☐

As the second step, we show that for every $\mathfrak{D}$-snake we can construct a $\mathfrak{D}$-metricobra.

Lemma 11.26 If $\mathcal{I}$ is a $\mathfrak{D}$-snake then there exists a $\mathfrak{D}$-metricobra $\mathcal{J}$.

Proof. Let $\mathcal{I}$ be a $\mathfrak{D}$-snake, N be the integer guaranteed by (SLen), and $\mathcal{I}'$ be any T-yardstick of length N (existence guaranteed by Lemma 11.22). We construct an interpretation $\mathcal{J}$ as a disjoint union of $\mathcal{I}$, $\mathcal{I}'$ and an additional domain element that we interpret as $\texttt{cbra}^{\mathcal{J}}$. We let $\mathcal{J}$ interpret all names from $\mathsf{N}^{T}_{📏}$ as in $\mathcal{I}'$, all names from $\mathsf{N}^{T}_{🧩}$ as in $\mathcal{I}$, and all other (unused in our concept definitions) names as $\texttt{cbra}^{\mathcal{J}}$. Interpretation of concept names is inherited from $\mathcal{I}$ and $\mathcal{J}$. Finally we interpret role names as in $\mathcal{I}$ and $\mathcal{I}'$ with minor corrections. More precisely: (i) we alter the interpretation of $r^{\mathcal{J}}$ to include an extra pair $(\texttt{cbra}^{\mathcal{J}}, \texttt{ld}^{\mathcal{J}})$, and (ii) we alter the interpretation of $s^{\mathcal{J}}$ to include the pair $(\texttt{cbra}^{\mathcal{J}}, \texttt{st}^{\mathcal{J}})$ and $\bigcup_{t \in T} C^{\mathcal{J}}_{t} \times \{\texttt{md}^{\mathcal{J}}_{t}\}$. Consult Figure 11.2 to see an example construction of $\mathcal{J}$ from $\mathcal{I}$ and $\mathcal{I}'$. The satisfaction of properties (MInit) and (MTile) follow from the construction of $\mathcal{J}$, while the other two properties are due to, respectively, (SLen) and (SVerti). Thus $\mathcal{J}$ is the desired metricobra. ☐

Finally, we show that every $\mathfrak{D}$-metricobra is actually a $\mathfrak{D}$-snake.

Lemma 11.27 If $\mathcal{I}$ is a $\mathfrak{D}$-metricobra then it is also a $\mathfrak{D}$-snake.

Proof. As $\mathcal{I}$ is a pseudosnake by definition, it suffices to show that it satisfies the missing conditions of Definition 11.16. Our first goal is to establish $\mathcal{I} \models$ (SLen). Note that by (SSpecTil) we have that $\texttt{rd}^{\mathcal{I}}$ is the only element r^{*}-reachable from $\texttt{ld}^{\mathcal{I}}$ that carries a down- and right-border tile, say t. Furthermore by (MTile) and (YReachMidT) we infer that $\texttt{rd}^{\mathcal{I}}$ is the only element r^{*}-reachable from $\texttt{ld}^{\mathcal{I}}$ that can s^{*}-reach $\texttt{end}^{\mathcal{I}}_{t}$. Take N to be the length of the yardstick. By Lemma 11.21 we know that $\texttt{md}^{\mathcal{I}}_{t}$ s^{N-1}-reaches $\texttt{end}^{\mathcal{I}}_{t}$, hence $\texttt{rd}^{\mathcal{I}}$ s^{N}-reaches $\texttt{end}^{\mathcal{I}}_{t}$ (and there is no other integer $M \neq N$ for which such reachability conditions hold). From (MSync) we know that $\texttt{cbra}^{\mathcal{I}}$ $r^{\#}s^{\#}$-reaches $\texttt{end}^{\mathcal{I}}_{t}$, thus by previous observations we deduce that $\texttt{cbra}^{\mathcal{I}}$ $r^{N}s^{N}$-reaches $\texttt{end}^{\mathcal{I}}_{t}$ (whence $\texttt{ld}^{\mathcal{I}}$ r^{N-1}-reaches $\texttt{rd}^{\mathcal{I}}$). This establishes the existence of a path of length N mentioned in (SLen), and we next need to show that all such paths have equal length. Consider the following cases:

- There exists $M < N$ such that $\texttt{ld}^{\mathcal{I}}$ r^{M-1}-reaches $\texttt{rd}^{\mathcal{I}}$. Then $\texttt{cbra}^{\mathcal{I}}$ $r^{M}s^{M}$-reaches some element that can s^{+}-reach $\texttt{end}^{\mathcal{I}}_{t}$. This yields a contradiction with condition (b) of (MSync).
- There exists $M > N$ such that $\texttt{ld}^{\mathcal{I}}$ r^{M-1}-reaches $\texttt{rd}^{\mathcal{I}}$. Then there is an element r^{*}-reachable from $\texttt{ld}^{\mathcal{I}}$ that can $r^{N}s^{N}$-reach $\texttt{end}^{\mathcal{I}}_{t}$. This contradictions condition (c) of (MSync).

Hence N is indeed unique. The fact that $\texttt{rd}^{\mathcal{I}}$ is the only element r^{N-1}-reachable from $\texttt{ld}^{\mathcal{I}}$ follows from the uniqueness of the tile assigned to $\texttt{rd}^{\mathcal{I}}$, see (SSpecTil) and Property (a) of (MSync). It remains now to show that $\mathcal{I} \models$ (SVerti). To do so, it suffices to observe that the following property holds. As all the s^{*}-paths from an element carrying a tile t to $\texttt{end}_{t}$ are of length N (by the previous discussion and Lemma 11.21), we can see that $r^{\#}s^{\#}$-reachability of $\texttt{end}_{t}$ is equivalent to r^{N}-reachability of some element carrying t. Then the satisfaction of (SVerti) follows immediately by (MVerti). Hence, $\mathcal{I}$ is indeed a $\mathfrak{D}$-snake. ☐

By collecting all previous lemmas we establish the correspondence between solvability of tiling systems and satisfiability of $C^{\mathfrak{D}}_{🐍}$.

Lemma 11.28 A domino tiling system $\mathfrak{D}$ is solvable if and only if $C^{\mathfrak{D}}_{🐍}$ has a model.

Proof. If $\mathcal{D}$ is solvable, then by Lemma 11.17 there exists a $\mathcal{D}$-snake, which by Lemma 11.26 implies the existence of a $\mathcal{D}$-metricobra, which is a model of $\mathrm{C}^{\mathcal{D}}_{🐍}$ (see Lemma 11.25). For the other direction, if $\mathrm{C}^{\mathcal{D}}_{🐍}$ has a model, then such a model is a $\mathcal{D}$-metricobra (by Lemma 11.25), as well as a $\mathcal{D}$-snake (by Lemma 11.27). As the existence of a $\mathcal{D}$-snake guarantees that $\mathcal{D}$ is solvable by Lemma 11.18, this finishes the proof. $\square$

By the undecidability of the tiling problem, we conclude the main theorem of this Section.

Theorem 11.29

The concept satisfiability problem for $\mathcal{ALCO}_{\mathsf{vpl}}$ is undecidable, even if the languages allowable in concepts are restricted to $\{r, s, r^+, s^+, r^*, s^*, (r+s)^*, (r+s)^+, r^{\#}s^{\#}\}$.

The logic $\mathcal{ALCO}_{\mathsf{reg}}$ is a notational variant of Propositional Dynamic Logic with nominals [KS14]. Thus the above theorem provides results also in the realm of formal verification.

11.4 Negative Results III: Entailment of Queries Involving Non-Regular atoms

We conclude the negative part of this chapter by showing that positive results regarding entailment of conjunctive queries with visibly-pushdown atoms ($\mathbb{VPL}$-CQs) in the database setting [LL15, Thm. 2] do not generalise even to $\mathcal{ALC}$-ontologies. We reduce from the *White-bordered Octant Tiling Problem*, to be defined next. Roughly speaking, the ontology used in our reduction will define a "grid" covered with tiles, while the query counterpart will serve as a tool to detect mismatches in its lower triangle (a.k.a. octant).

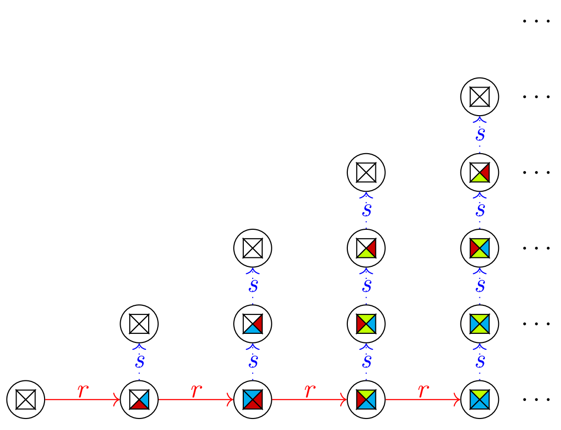


Figure 11.3: Visualization of a fragment of a T-octant interpretation.

We refer to the set $\mathbb{O} \coloneqq \{(n,m) \mid n,m \in \mathbb{N}, 0 \leq m \leq n\}$ as the **octant**. Let $\mathcal{D} \coloneqq (\mathrm{Col}, \mathrm{T}, \square)$ be a domino tiling system (defined as in Section 11.3). For our reduction it is convenient to impose several restrictions on T, namely that the all-white tile $\boxtimes$ belongs to T, and that all other tiles from T containing white colour are both left- and up-border but neither down- nor right-border.

Definition 11.30 We say that $\mathcal{D}$ **covers** the **octant** $\mathbb{O} \coloneqq \{(n,m) \mid n,m \in \mathbb{N}, 0 \leq m \leq n\}$ if there exists a mapping $\xi \colon \mathbb{O} \to \mathrm{T}$ such that for all pairs $(n,m) \in \mathbb{O}$ the following conditions are satisfied:

(OInit) $\xi(0,0) = \boxtimes$ and $\xi(1,0) \neq \boxtimes$.

(OVerti) If $(n, m{+}1) \in \mathbb{O}$ then $\xi(n,m)$ and $\xi(n,m{+}1)$ are V-compatible.

(OHori) The tiles $\xi(n,m)$ and $\xi(n{+}1,m)$ are H-compatible.

In the *White-bordered Octant Tiling Problem* we ask if an input domino tiling system $\mathcal{D}$ (with additional conditions on tiles mentioned above) covers the octant $\mathbb{O}$. Observe that:

Observation 11.31. Let $\mathcal{D}$ be a tiling system and let $\xi\colon \mathbb{O} \to \mathrm{T}$ be covering the octant. Then for all $i \in \mathbb{N}$ we have that $\xi(i,i) = \boxtimes$, and $\xi(i{+}1,i)$ is left- and up-bordered. Moreover, no position (i,j) satisfying $0 < j < i{-}1$ carries a white-border tile.

Proof. The first statement follows by routine induction. From (OInit) we have $\xi(0,0) = \boxtimes$. Suppose now that $\xi(i,i) = \boxtimes$. Then, by (OHori) and our choice of tiles, we have that $\xi(i{+}1,i)$ is left- and up-border. Thus, as $\boxtimes$ is the only tile in T that is down-border, we conclude that $(i{+}1,i{+}1)$ carries $\boxtimes$. For the second statement, suppose that there is a pair (i,j) for which $j < i{-}1$ and $\xi(i,j)$ contains a white-border tile, and take lexicographically smallest such (i,j). By design of $\mathcal{D}$, the tile $\xi(i,j)$ is left- and up-border. From (OInit) we infer $i > 1$. By (OHori), the tile $\xi(i{-}1,j)$ is also right-border, contradicting the minimality of (i,j). $\square$

We first show undecidability of the White-bordered Octant Tiling Problem, which follows by a straightforward reduction[2] from the Octant Tiling Problem [BDG+10, Sec. 3.1].

Lemma 11.32 The white-bordered Octant Tiling Problem is undecidable.

Proof. We say that a domino tiling system $\mathcal{D}$ *almost covers* the octant if there is a mapping $\xi\colon \mathbb{O} \to \mathrm{T}$ such that all pairs $(n,m) \in \mathbb{O}$ fulfil (OVerti) and (OHori). It was stated by Bresolin et al. [BDG+10, Sec. 3.1] (and follows from the classical work of van Emde Boas [Boa97]) that deciding whether $\mathcal{D}$, that does not contain white-bordered tiles, almost covers the octant is undecidable. To prove undecidability of the White-bordered Octant Tiling Problem we provide a reduction from the tiling problem of Bresolin et al. Thus, let $\mathcal{D} := (\mathrm{Col}, \mathrm{T}, \square)$ be a domino tiling system that does not have white-bordered tiles, and consider $\mathcal{D}' := (\mathrm{Col}, \mathrm{T}', \square)$ to be the tiling system obtained by putting $\mathrm{T}' := \{\boxtimes, (\mathrm{c}_l, \mathrm{c}_d, \mathrm{c}_r, \mathrm{c}_u), (\square, \mathrm{c}_d, \mathrm{c}_r, \square) \mid (\mathrm{c}_l, \mathrm{c}_d, \mathrm{c}_r, \mathrm{c}_u) \in \mathrm{T}\}$. We claim that $\mathcal{D}'$ covers the octant if and only if $\mathcal{D}$ almost covers the octant. Indeed:

- The "if" direction is relatively straightforward. Let ξ be a map witnessing that $\mathcal{D}$ almost covers the octant. We alter the tiles assigned by ξ to the diagonal to make them left- and up-border, then we shift ξ right, and fill the remaining places with $\boxtimes$. Formally, let ξ' be defined as: (i) $\xi'(i,i) := \boxtimes$ for all $i \in \mathbb{N}$, (ii) $\xi'(i,i{-}1) := (\square, \mathrm{c}_d, \mathrm{c}_r, \square)$, where $\xi(i,i) = (\mathrm{c}_l, \mathrm{c}_d, \mathrm{c}_r, \mathrm{c}_u)$, for all positive $i \in \mathbb{N}$, and (iii) $\xi'(i,j) = \xi'(i,j{-}1)$ for all remaining $i,j \in \mathbb{N}$. It can be readily verified that ξ' satisfies the required conditions.
- For the "only if" direction, let ξ' be a map witnessing that $\mathcal{D}'$ covers the octant. It suffices to take $\xi\colon (i,j) \mapsto \xi'(i{+}2,j)$. By Observation 11.31) none of the tiles assigned by ξ is white-bordered. As ξ' covers the octant, we infer that ξ (almost) covers the octant.

$\square$

Our undecidability proof relies on concepts from $\mathrm{C}^{\mathrm{T}}_{🧩}$ and the non-regular language $r^{\#}s^{\#}$. We fix and enumerate a set of tiles T as $\mathrm{t}_1, \ldots, \mathrm{t}_{\mathrm{N}}$ for $\mathrm{N} := |\mathrm{T}|$.

As the first building block, we introduce octant interpretations.

Definition 11.33 An interpretation $\mathcal{I}$ is called **T-octant** if there exists a function $\xi\colon \mathbb{O} \to \mathrm{T}$ fulling (OInit) and (OVerti) (but not necessarily (OHori)) that satisfies the following conditions: (i) $\Delta^{\mathcal{I}} = \mathbb{O}$, (ii) $r^{\mathcal{I}} = \{((n,0),(n{+}1,0)) \mid n \in \mathbb{N}\}$, (iii) $s^{\mathcal{I}} = \{((n,m),(n,m{+}1)) \mid n,m \in \mathbb{N}, m < n\}$, and (iv) $\mathrm{C}_{\mathrm{t}}^{\mathcal{I}} = \{(n,m) \mid \xi(n,m) = \mathrm{t}\}$ for all tiles $\mathrm{t} \in \mathrm{T}$. In this case $\mathcal{I}$ *represents* ξ.

Due to the fact that $\xi\colon \mathbb{O} \to \mathrm{T}$ is a function, every domain element of a T-octant carries precisely one tile. Moreover, for every such ξ we can easily find a T-octant representing ξ (just employ the above definition). For more intuitions consult Figure 11.3.

To avoid disjunction in the forthcoming query, we need to extend octant interpretations with yet another way of representing tiles, which will be based on distances. Suppose that an element (n,m)

[2]The author was asked explicitely by one of his supervisors to provide the proof of this fact.

from an octant interpretation $\mathcal{I}$ carries a tile t_i, and that the tile assigned to its horizontal predecessor $(n{-}1, m)$, if exists, is equal to t_j. We equip (n, m) with an outgoing s^+-path ρ of length N, composed of fresh elements (we employ fresh concept names In and $\overline{\mathrm{In}}$ to make a distinction between elements in octant and the ones present in "extra paths"). The i-th element of ρ will be the unique element of ρ that belongs to the interpretation of a concept Cur. Similarly, the j-th element of ρ will be the unique element of ρ that belongs to the interpretation of a concept Prev. Thus, the distance from (n, m) to an element labelled by the concept Cur (resp. Prev) uniquely determines the tile of a current node (resp. its horizontal predecessor). As a mere technicality, needed for query design, we enrich the element $(0, 0)$ with an incoming r^+-path of length N. A formalization comes next.

Definition 11.34 Let $\mathcal{I}$ be an interpretation with a domain $\mathbb{O} \times \mathbb{Z}_{\mathrm{N}+1} \cup \{(-i{-}1, 0, 0) \mid i \in \mathbb{Z}_{\mathrm{N}}\}$, and let $\mathcal{I}_{\mathbb{O}}$ be the restriction of $\mathcal{I}$ to the set $\mathbb{O} \times \{0\}$. We call $\mathcal{I}$ a **T-hyperoctant** if:

- $\mathcal{I}_{\mathbb{O}}$ is isomorphic (via a projection $(n, m, 0) \mapsto (n, m)$) to a T-octant interpretation,
- $\mathrm{In}^{\mathcal{I}} = \mathbb{O} \times \{0\}$, $\overline{\mathrm{In}}^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus \mathrm{In}^{\mathcal{I}}$, $\overline{\mathrm{Prev}}^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus \mathrm{Prev}^{\mathcal{I}}$,
- $r^{\mathcal{I}} = r^{\mathcal{I}_{\mathbb{O}}} \cup \{((-i{-}1, 0, 0), (-i, 0, 0)) \mid i \in \mathbb{Z}_{\mathrm{N}}\}$,
- $s^{\mathcal{I}} = s^{\mathcal{I}_{\mathbb{O}}} \cup \{((n, m, k), (n, m, k{+}1)) \mid (n, m) \in \mathbb{O}, k \in \mathbb{Z}_{\mathrm{N}}\}$,
- $\mathrm{C}^{\mathcal{I}}_{\mathrm{t}_k} = \mathrm{C}^{\mathcal{I}_{\mathbb{O}}}_{\mathrm{t}_k}$ and $\mathrm{Cur}^{\mathcal{I}} = \{(n, m, k) \mid (n, m, 0) \in \mathrm{C}^{\mathcal{I}}_{\mathrm{t}_k}, \mathrm{t}_k \in \mathrm{T}\}$, and
- for every $(n, m) \in \mathbb{O}$ there is precisely one positive k for which $(n, m, k) \in \mathrm{Prev}^{\mathcal{I}}$ holds, and for such a number k we have that t_k and the tile carried by $(n, m, 0)$ are H-compatible.

Note that, in addition to what is present in the definition of T-hyperoctant, we employed fresh concept names, namely In, Prev, Cur, $\overline{\mathrm{In}}$, and $\overline{\mathrm{Prev}}$. The purpose of "overlined" concepts is to help with a design of a query, as the use of negation is not allowed there. We call $\mathcal{I}$ *proper* if for all $(n, m, k) \in \mathrm{Cur}^{\mathcal{I}}$ we have $(n{+}1, m, k) \in \mathrm{Prev}^{\mathcal{I}}$. Note that properness is not definable in $\mathcal{ALC}$, but we will enforce it with a query. The map $\xi \colon \mathbb{O} \to \mathrm{T}$ *represented* by a hyperoctant $\mathcal{I}$ is the map represented by its T-octant substructure $\mathcal{I}_{\mathbb{O}}$.

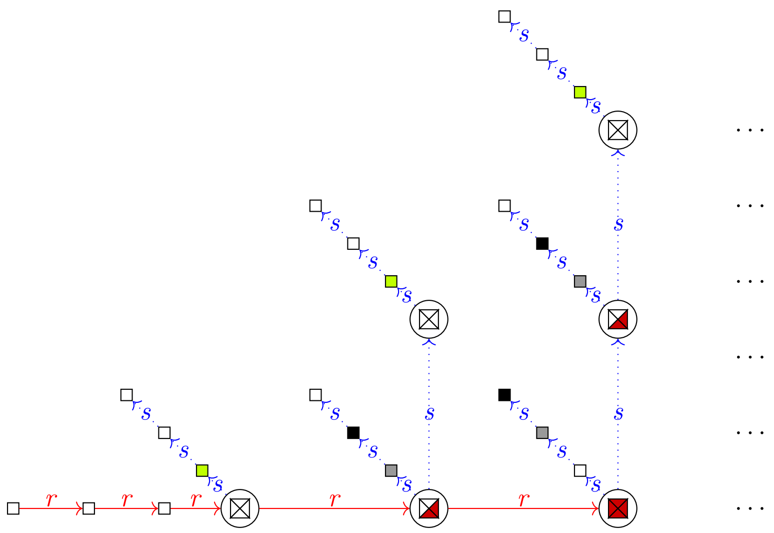


Figure 11.4: A fragment of a proper $\{\mathrm{t}_1, \mathrm{t}_2, \mathrm{t}_3\}$-hyperoctant for $\mathrm{t}_1 =$ ⊠, $\mathrm{t}_2 =$ ◪, and $\mathrm{t}_3 =$ ▣. Elements in $\mathrm{In}^{\mathcal{I}}$ are depicted as circles. Elements in $\mathrm{Prev}^{\mathcal{I}}$ are marked grey, elements in $\mathrm{Cur}^{\mathcal{I}}$ are marked black, and the lime elements belong to $\mathrm{Prev}^{\mathcal{I}} \cap \mathrm{Cur}^{\mathcal{I}}$.

We next relate proper T-hyperoctants and domino tiling systems.

Lemma 11.35 Let $\mathcal{D} \coloneqq (\mathrm{Col}, \mathrm{T}, \square)$ be a domino tilling system. For every proper T-hyperoctant $\mathcal{I}$, the map $\xi \colon \mathbb{O} \to \mathrm{T}$ represented by $\mathcal{I}$ witnesses that $\mathcal{D}$ covers the octant. If $\mathcal{D}$ covers the octant, as witnessed by a map $\xi \colon \mathbb{O} \to \mathrm{T}$, then there exists a proper T-hyperoctant $\mathcal{I}$ representing ξ.

Proof. Take a proper T-hyperoctant $\mathcal{I}$, and the map ξ represented by $\mathcal{I}$. By definition of a T-octant, we have that ξ satisfies (OInit) and (OVerti). To establish (OHori) take any $(n,m) \in \mathbb{O}$, and suppose that $\xi(n,m) = \mathrm{t}_i$ and $\xi(n{+}1,m) = \mathrm{t}_j$ hold for some integers $i,j \in \mathbb{N}$. By definition of a hyperoctant and properness of $\mathcal{I}$, we have that $(n{+}1,m,0)$ carries the tile t_j, and $(n{+}1,m,i) \in \mathrm{Prev}^{\mathcal{I}}$. Thus, by the 6th item of Definition 11.34 we infer that tiles t_i and t_j are H-compatible. The proof of the other statement is routine: take $\mathcal{I}_{\mathbb{O}}$ to be a T-octant representing ξ. W.l.o.g. assume that $\mathcal{I}_{\mathbb{O}}$ interprets all role names and concept names that are not mentioned in its definitions as empty sets. We next rename the domain of $\mathcal{I}_{\mathbb{O}}$ to $\mathbb{O} \times \{0\}$ and append fresh elements to make the domain equal to $\mathbb{O} \times \mathbb{Z}_{\mathrm{N}+1} \cup \{(-i{-}1,0,0) \mid i \in \mathbb{Z}_{\mathrm{N}}\}$. Then, we enlarge the interpretations of r, s, In, and Cur in a minimal way according to the first five items of Definition 11.34. Finally, we interpret Prev as the set composed of all triples $(n{+}1,m,k)$ for all $(n,m) \in \mathbb{O}$ carrying a tile t_k, and all triples (n,n,ℓ) for $x \in \mathbb{N}$ and ℓ denoting the index of $\boxtimes$ in T. Call the resulting interpretation $\mathcal{I}$. Clearly, $\mathcal{I}$ is T-hyperoctant due to the fact that ξ is a map and respects conditions (OInit), (OVerti), and (OHori). $\square$

We employ a $\mathbb{VPL}$-CQ $q^{\mathfrak{D}}_{\blacktriangle}(u_1,u_2,v_1,v_2,w_1,w_2,x_1,x_2,y_1,y_2,z_1,z_2)$ as a tool for detecting whether a given T-hyperoctant $\mathcal{I}$ is proper. Observe that $\mathcal{I}$ *is not* proper if and only if there is a position $(n,m) \in \mathbb{O}$ and a number $1 \leq k \leq \mathrm{N}$, for which we have $(n,m,k) \in \mathrm{Cur}^{\mathcal{I}}$ and $(n{+}1,m,k) \notin \mathrm{Prev}^{\mathcal{I}}$. This is precisely the condition that is going to be expressed with $q^{\mathfrak{D}}_{\blacktriangle}$, informally presented at Figure 11.5. The intuition behind $q^{\mathfrak{D}}_{\blacktriangle}$ is as follows.

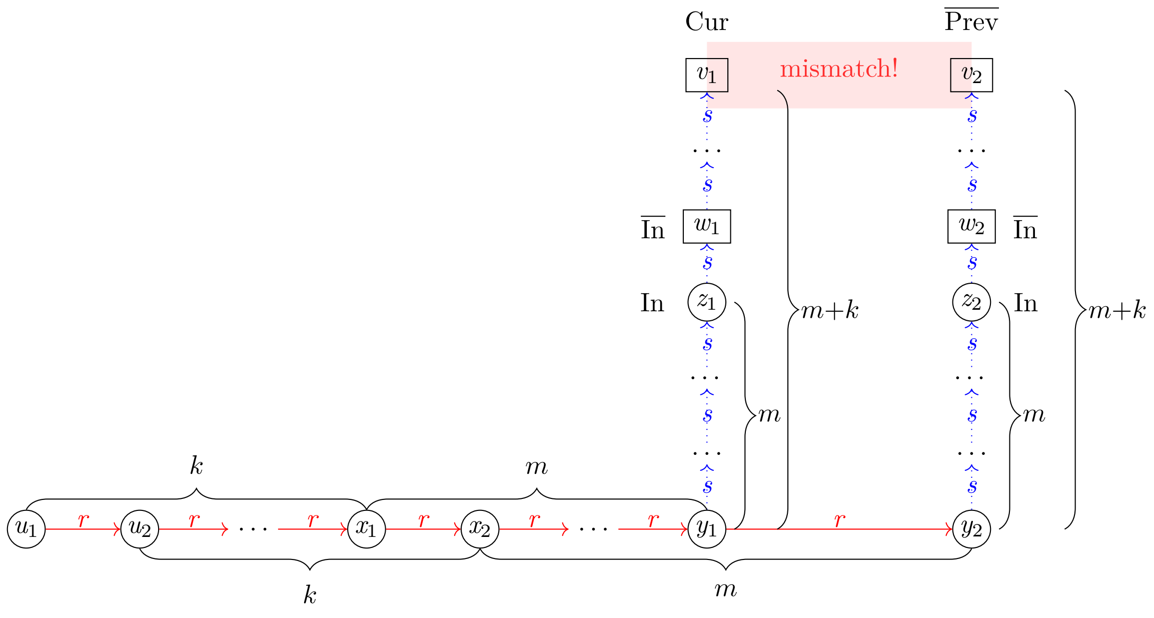


Figure 11.5: Visualisation of the query $q^{\mathfrak{D}}_{\blacktriangle}(u_1,u_2,v_1,v_2,w_1,w_2,x_1,x_2,y_1,y_2,z_1,z_2)$.

We first ensure that z_1, z_2 are mapped to some elements representing the coordinates (n,m) and (n',m') of the octant for some integers satisfying $n' = n{+}1$ and $m = m'$. The fact that they belong to the octant is handled by means of the In concept. The equality $n' = n{+}1$ is achieved by introducing variables y_1, y_2, stating their r-connectedness, and the s^*-reachability of z_1 and z_2 from, respectively, y_1 and y_2. Thus y_1, y_2 are placed "at the bottom" of the variables z_1 and z_2. Next, the equi-hight of z_1, z_2 is ensured with extra r-connected variables x_1, x_2 located to the left of y_1, and non-regular atoms $r^{\#}s^{\#}(x_1,z_1)$ and $r^{\#}s^{\#}(x_2,z_2)$, enforcing equality of the distance between x_i and y_i, and the distance between y_i and z_i, for all $i \in \{1,2\}$. Once we know that the variables z_1, z_2 are mapped by a query as desired, we need to express that they violate properness of $\mathcal{I}$. Recall that we want to establish $(n,m,k) \in \mathrm{Cur}^{\mathcal{I}}$ and $(n{+}1,m,k) \notin \mathrm{Prev}^{\mathcal{I}}$ for some k. Such elements will be represented, respectively, by variables v_1 and v_2. To express the mentioned constraint, we introduce fresh r-connected variables u_1, u_2 that are located to the left of x_1, x_2, and whose distance to u_1, u_2 will be precisely the k that we are looking for. We stress that we do not "hardcode" the value of k in the query. As the variable u_1 is free to map whenever it wants, this mimics a disjunction over possible values of k. We ensure the variables v_1, v_2 are mapped to elements outside the octant by expressing that they are r^*-reachable from the s-successors w_1, w_2 of z_1, z_2, that are labelled with $\overline{\text{In}}$. Note that just expressing that v_1, v_2 belong to $\overline{\text{In}}$ does not suffice, as the path leading from some of z_i to v_i could contain elements in In (which we implicitly forbid). With non-regular atoms $r^{\#}s^{\#}(u_1, v_1)$ and $r^{\#}s^{\#}(u_2, v_2)$ we make sure that the distance between z_1 and v_1 (respectively z_2 and v_2) is indeed k.

Despite the high technicality of our construction, we hope that after reading the above intuition and glancing at Figure 11.5, the following definition of the query $q^{\mathcal{D}}_{\blacktriangle}$ should be now understandable:

$$q^{\mathcal{D}}_{\blacktriangle} \coloneqq r(u_1, u_2) \wedge r^*(u_2, x_1) \wedge r(x_1, x_2) \wedge r^*(x_2, y_1) \wedge r(y_1, y_2) \wedge \text{Cur}(v_1) \wedge \overline{\text{Prev}}(v_2) \wedge$$
$$\bigwedge_{i=1}^{2} \left[s^*(y_i, z_i) \wedge \text{In}(z_i) \wedge s(z_i, w_i) \wedge \overline{\text{In}}(w_i) \wedge s^*(w_i, v_i) \wedge r^{\#}s^{\#}(x_i, z_i) \wedge r^{\#}s^{\#}(u_i, v_i) \right].$$

By routine case analysis with a bit of calculations, we can show that:

Lemma 11.36 Let $\mathcal{D} \coloneqq (\text{Col}, \text{T}, \square)$ be a domino tilling system and let $\mathcal{I}$ be a T-hyperoctant. Then we have that $\mathcal{I}$ is proper if and only if $\mathcal{I} \not\models q^{\mathcal{D}}_{\blacktriangle}$.

Proof. We start from the "only if" direction. Suppose that $\mathcal{I}$ is proper, but there is a match η witnessing $\mathcal{I} \models q^{\mathcal{D}}_{\blacktriangle}$. We provide a few tedious calculations. As the atoms $r(x_1, x_2)$, $r(y_1, y_2)$, and $r(u_1, u_2)$ belong to $q^{\mathcal{D}}_{\blacktriangle}$, there are $a, b, n \in \mathbb{Z}$ such that $\eta(u_1) = (a, 0, 0)$, $\eta(u_2) = (a{+}1, 0, 0)$, $\eta(x_1) = (b, 0, 0)$, $\eta(x_2) = (b{+}1, 0, 0)$, $\eta(y_1) = (n, 0, 0)$, and $\eta(y_2) = (n{+}1, 0, 0)$. By the presence of atoms $r^*(u_2, x_1)$ and $r^*(x_2, y_1)$ in the query, we infer that $a < b < n$. From atoms $s^*(y_i, z_i), \text{In}(z_i)$ present in $q^{\mathcal{D}}_{\blacktriangle}$, we can deduce that $\eta(z_1) = (n, m, 0)$, $\eta(z_2) = (n{+}1, m', 0)$ hold for some $m, m' \in \mathbb{N}$. Next, by the satisfaction of the atoms $r^{\#}s^{\#}(x_1, z_1)$ and $r^{\#}s^{\#}(x_2, z_2)$ the equations $m = n - b$ and $m' = (n{+}1) - (b{+}1)$ follow. Thus $m' = m$. As the next step, we deal with the variables w_1 and w_2. From the atoms $s(z_i, w_i)$ and $\overline{\text{In}}(w_i)$ we can deduce that $\eta(w_1) = (n, m, 1)$, $\eta(w_2) = (n{+}1, m, 1)$. Together with atoms $s^*(w_1, v_1)$ and $s^*(w_1, v_1)$, this implies that $\eta(v_1) = (n, m, k)$, and $\eta(z_2) = (n{+}1, m, k')$ hold for some positive $k, k' \in \mathbb{N}$. By the satisfaction of the atoms $r^{\#}s^{\#}(u_1, v_1)$ and $r^{\#}s^{\#}(u_2, v_2)$ the equations $m{+}k = n{-}a$ and $m{+}k' = (n{+}1) - (a{+}1)$ follow. Hence, $k = k'$ holds and by collecting all the previous equations we conclude that $\eta(v_1) = (n, m, k)$ and $\eta(v_1) = (n{+}1, m, k)$. Finally, the atoms $\text{Cur}(v_1)$ and $\overline{\text{Prev}}(v_2)$ of $q^{\mathcal{D}}_{\blacktriangle}$ imply that $(n, m, k) \in \text{Cur}^{\mathcal{I}}$ but $(n{+}1, m, k) \notin \text{Prev}^{\mathcal{I}}$. This contradicts the properness of $\mathcal{I}$.

We show the "if" direction by contraposition. Suppose that $\mathcal{I}$ is not proper, and that the properness of $\mathcal{I}$ is violated by $(n, m, k) \in \text{Cur}^{\mathcal{I}}$ and $(n{+}1, m, k) \in \overline{\text{Prev}}^{\mathcal{I}}$. Then the map $v_1 \mapsto (n, m, k)$, $v_2 \mapsto (n{+}1, m, k)$, $w_1 \mapsto (n, m, 1)$, $w_2 \mapsto (n{+}1, m, 1)$, $z_1 \mapsto (n, m, 0)$, $z_2 \mapsto (n{+}1, m, 0)$, $y_1 \mapsto (n, 0, 0)$, $y_2 \mapsto (n{+}1, 0, 0)$, $x_1 \mapsto (n{-}m, 0, 0)$, $x_2 \mapsto (n{-}m{+}1, 0, 0)$, $u_1 \mapsto (n{-}m{-}k, 0, 0)$, $u_2 \mapsto (n{-}m{-}k{+}1, 0, 0)$ is a match for $q^{\mathcal{D}}_{\blacktriangle}$. This finishes the proof.

As a side remark, note that if the value of $n{-}m$ is sufficiently small, the value of $n{-}m{-}k$ can be negative (but not smaller than $-\text{N}$). This explains at last why we appended an incoming r^*-path of length N to the element $(0, 0, 0)$ in the construction of hyperoctants. $\square$

As the final step of our construction, we define an $\mathcal{ALC}$-TBox $\mathcal{T}^{\mathcal{D}}_{\blacktriangle}$ whose intended tree-like models will contain T-hyperoctants. Most of the axioms written below are formalizations of straightforward properties satisfied by T-hyperoctants and grids. As we are aiming to design an $\mathcal{ALC}$-TBox, all the properties expressed in $\mathcal{T}^{\mathcal{D}}_{\blacktriangle}$ are interpreted "globally". Hence, to express existence of a starting point of a T-hyperoctant we employ a fresh role name *aux* and say that every element has an *aux*-successor (with the intended meaning that such a successors "starts" a T-hyperoctant). For brevity, we say that an element is inner (resp. outer) if it belongs (resp. does not belong) to the interpretation of In. For a language $\mathcal{L}$ we also say that an element e in $\mathcal{I}$ is $\mathscr{L}$-outer-reachable from d if there exists a $\mathscr{L}$-path from d to e composed solely of outer elements (with a possible exception of d).

(GStart) Every element has an *aux*-successor. Every such successor has an outgoing r^{N}-path composed of outer elements that leads to an inner element carrying a tile $\boxtimes$ that has an inner r-successor carrying a tile different from $\boxtimes$.

(GCompl) Concept name $\overline{\mathrm{In}}$ (resp. $\overline{\mathrm{Prev}}$) is interpreted as the complement of the interpretation of concept name In (resp. Prev).

(GTil) Every inner element is labelled with precisely one concept name from $\mathrm{C}^{\mathrm{T}}_{🧩}$, and there are no outer elements labelled with such concept names.

(GSuc) If an element has an inner r-successor then such a successor also has an inner r-successor. Every inner element has an inner s-successor.[3]

(GPath) Every inner element has an outgoing s^{N}-path composed solely of outer elements.

(GCur) For every inner element carrying the tile t_k we have that: (i) all s^k-outer-reachable elements are labelled with Cur, (ii) there is no s^ℓ-outer-reachable element, for $\ell \leq \mathrm{N}$ and $\ell \neq k$, that is labelled with Cur.

(GPrev) For every inner element there exists a number $k \leq \mathrm{N}$ such that: (i) all s^k-outer-reachable elements are labelled with Prev, (ii) there is no s^ℓ-outer-reachable element, for $\ell \leq \mathrm{N}$ and $\ell \neq k$, that is labelled with Prev.

(GVerti) Every pair of tiles carried by s-successors is V-compatible.

(GHori) For all tiles $\mathrm{t} \in \mathrm{T}$ and all elements carrying t that can s^ℓ-outer-reach an element labelled with Prev for some ℓ, we have that t_ℓ and t are H-compatible.

The definition of $\mathcal{T}^{\mathcal{D}}_{\blacktriangle}$ is provided in the proof of the following lemma.

Lemma 11.37 There exists an $\mathcal{ALC}$-TBox $\mathcal{T}^{\mathcal{D}}_{\blacktriangle}$ expressing the above properties.

Proof. Given $\mathcal{ALC}$-concepts C, D and a role name $r \in \mathbf{N_R}$ we employ macros $(\forall r.\mathrm{C})^n.\mathrm{D}$ and $(\exists r.\mathrm{C})^n.\mathrm{D}$ defined inductively for $n \geq 1$ as follows:

$$(\forall r.\mathrm{C})^1.\mathrm{D} \coloneqq \forall r.(\mathrm{C} \rightarrow \mathrm{D}), \qquad (\forall r.\mathrm{C})^{n+1}.\mathrm{D} \coloneqq \forall r.(\mathrm{C} \rightarrow [(\forall r.\mathrm{C})^n.\mathrm{D}]),$$

$$(\exists r.\mathrm{C})^1.\mathrm{D} \coloneqq \exists r.(\mathrm{C} \sqcap \mathrm{D}), \qquad (\exists r.\mathrm{C})^{n+1}.\mathrm{D} \coloneqq \exists r.(\mathrm{C} \sqcap [(\exists r.\mathrm{C})^n.\mathrm{D}]).$$

Our $\mathcal{T}^{\mathcal{D}}_{\blacktriangle}$ is composed of all the GCIs listed below, describing properties (GStart), (GCompl), (GTil), (GSuc), (GPath), (GCur), (GPrev), (GVerti), and (GHori) in precisely this order.

$$\begin{aligned}
\top &\sqsubseteq & (\exists aux.\top) \sqcap \forall aux.\left((\exists r.\overline{\mathrm{In}})^{\mathrm{N}}.[\exists r.\left(\mathrm{In} \sqcap \mathrm{C}_{\boxtimes} \sqcap \exists r.(\mathrm{In} \sqcap \neg \mathrm{C}_{\boxtimes})\right)]\right), \\
\top &\sqsubseteq & (\overline{\mathrm{In}} \leftrightarrow \neg \mathrm{In}) \sqcap (\overline{\mathrm{Prev}} \leftrightarrow \neg \mathrm{Prev}), \\
\top &\sqsubseteq & [(\bigsqcup_{\mathrm{t} \in \mathrm{T}} \mathrm{C}_{\mathrm{t}}) \rightarrow \mathrm{In}] \sqcap [\mathrm{In} \rightarrow (\bigsqcup_{\mathrm{t} \in \mathrm{T}} (\mathrm{C}_{\mathrm{t}} \sqcap \bigsqcap_{\mathrm{t}' \in \mathrm{T} \setminus \{\mathrm{t}\}} \neg \mathrm{C}_{\mathrm{t}'})], \\
\top &\sqsubseteq & (\forall r.[\mathrm{In} \rightarrow (\exists r.\mathrm{In})]) \sqcap (\mathrm{In} \rightarrow \exists s.\mathrm{In}), \\
\top &\sqsubseteq & \mathrm{In} \rightarrow (\exists s.\overline{\mathrm{In}})^{\mathrm{N}}.\top, \\
\top &\sqsubseteq & \bigsqcap_{1 \leq k \leq \mathrm{N}} \left(\mathrm{C}_{\mathrm{t}_k} \rightarrow \left[[(\forall s.\overline{\mathrm{In}})^k.\mathrm{Cur}] \sqcap \bigsqcap_{1 \leq \ell \leq \mathrm{N},\ \ell \neq k} [(\forall s.\overline{\mathrm{In}})^\ell.\neg \mathrm{Cur}]\right]\right), \\
\top &\sqsubseteq & \bigsqcup_{1 \leq k \leq \mathrm{N}} \left([(\forall s.\overline{\mathrm{In}})^k.\mathrm{Prev}] \sqcap \bigsqcap_{1 \leq \ell \leq \mathrm{N},\ \ell \neq k} [(\forall s.\overline{\mathrm{In}})^\ell.\neg \mathrm{Prev}]\right),
\end{aligned}$$

[3]This ensures for each $i \in \mathbb{N}$ the existence of i s-successors for the i-th element in the bottom of the octant.

$$\top \sqsubseteq \bigsqcup_{\mathrm{t}\in\mathrm{T}} \left(\mathrm{C}_{\mathrm{t}} \rightarrow [\bigsqcap_{\mathrm{t}'\in\mathrm{T},(\mathrm{t},\mathrm{t}') \text{ are not V-compatible}} \neg\exists s.\mathrm{C}_{\mathrm{t}'}] \right),$$

$$\top \sqsubseteq \bigsqcup_{\mathrm{t}\in\mathrm{T}} \left(\mathrm{C}_{\mathrm{t}} \rightarrow [\bigsqcap_{\mathrm{t}_{\ell}\in\mathrm{T},(\mathrm{t}_{\ell},\mathrm{t}) \text{ are not H-compatible}} \neg(\exists s.\overline{\mathrm{In}})^{\ell}.\mathrm{Prev}] \right).$$

It should be clear that the above GCIs formalise the required properties. □

We first see that T-hyperoctant can be extended to (counter)models of $\mathcal{T}^{\mathcal{D}}_{\blacktriangle}$ and $q^{\mathcal{D}}_{\blacktriangle}$.

Lemma 11.38 Every proper T-hyperoctant can be extended to a model of $\mathcal{T}^{\mathcal{D}}_{\blacktriangle}$ that violates $q^{\mathcal{D}}_{\blacktriangle}$.

Proof. Let $\mathcal{I}$ be a proper T-hyperoctant. Consider an interpretation $\mathcal{J}$ with the domain $\Delta^{\mathcal{J}} := \Delta^{\mathcal{I}} \cup ((\mathbb{N}\times\mathbb{N}) \setminus \mathbb{O}) \times \mathbb{Z}_{\mathrm{N}+1}$ defined as follows:

- $\mathcal{J}$ restricted to $\Delta^{\mathcal{I}}$ is isomorphic to $\mathcal{I}$,
- $s^{\mathcal{J}} := s^{\mathcal{I}} \cup \{((n,m,k),(n,m,k+1)),((n,m,0),(n,m+1,0)) \mid (n,m) \in (\mathbb{N}\times\mathbb{N})\setminus\mathbb{O}, k \in \mathbb{Z}_{\mathrm{N}}\}$,
- $r^{\mathcal{J}} := r^{\mathcal{I}}$,
- $aux^{\mathcal{J}} := \Delta^{\mathcal{J}} \times \{(-\mathrm{N},0,0)\}$, and
- For each name $\mathrm{A} \in \mathbf{N_C}$, each tuple (n,m,k) with $(n,m) \notin \mathbb{O}$ belongs $\mathrm{A}^{\mathcal{J}}$ iff $(0,0,k) \in \mathrm{A}^{\mathcal{I}}$.

Intuitively, we enlarged $\mathcal{I}$ to a grid and filled fresh places with copies of the diagonal of $\mathcal{I}$ (this is needed to fulfil the second conjunct of the axiomatisation of (GSuc)). By construction of $\mathcal{J}$ (and the fact that $\mathcal{I}$ is a T-hyperoctant) it follows $\mathcal{J}$ is a model of $\mathcal{T}^{\mathcal{D}}_{\blacktriangle}$. Call $\mathcal{J}$ *proper* if for all $(n,m,k) \in \mathrm{Cur}^{\mathcal{J}}$ we have $(n+1,m,k) \in \mathrm{Prev}^{\mathcal{J}}$. Note that as $\mathcal{I}$ is proper, the above condition holds for all triples (n,m,k) with $(n,m) \in \mathbb{O}$. For other tuples, we simply use the fact that the interpretation of concepts for (n,m,k) with $(n,m) \notin \mathbb{O}$ is inherited from the elements of the form $(0,0,k)$. Thus $\mathcal{J}$ is indeed proper. Now, without any further changes, the proof of the "only if" direction of Lemma 11.36 establishes $\mathcal{J} \not\models q^{\mathcal{D}}_{\blacktriangle}$. □

As a handy lemma used in the forthcoming proof, we need to establish that:

Lemma 11.39 Every single-role tree-like model of $\mathcal{T}^{\mathcal{D}}_{\blacktriangle}$ contains a substructure that is isomorphic to a T-hyperoctant.

Proof. The proof idea is simple but tedious. We take single-role tree-like model $\mathcal{I}$ of $\mathcal{T}^{\mathcal{D}}_{\blacktriangle}$, and select elements that will later constitute a hyperoctant branch-by-branch. Let $\mathrm{d}_{(-\mathrm{N},0,0)}$ be any element of $\mathcal{I}$ which is an aux-successor of some element of $\mathcal{I}$ (it exists as $\Delta^{\mathcal{I}} \neq \emptyset$ and $\mathcal{I}$ satisfies (GStart)). Thus there exists an r^{*}-path of elements $\mathrm{d}_{(-\mathrm{N},0,0)}, \mathrm{d}_{(-\mathrm{N}+1,0,0)}, \ldots, \mathrm{d}_{(0,0,0)}, \mathrm{d}_{(1,0,0)}$ witnessing the satisfaction of (GStart), where only $\mathrm{d}_{(0,0,0)}, \mathrm{d}_{(1,0,0)}$ belong to $\mathrm{In}^{\mathcal{J}}$ (by (GStart) and (GCompl)), $\mathrm{d}_{(0,0,0)}$ carries a tile $\boxtimes$ and $\mathrm{d}_{(1,0,0)}$ carries a tile that differs from $\boxtimes$. After employing a routine induction, we see that the first conjunct of (GSuc) yields the existence of an infinite r^{*}-path $\mathrm{d}_{(1,0,0)}, \mathrm{d}_{(2,0,0)}, \ldots$ of inner elements. By the second conjunct of (GSuc) we know that every $\mathrm{d}_{(i,0,0)}$ has an outgoing s^{i}-path $\mathrm{d}_{(i,1,0)}, \ldots, \mathrm{d}_{(i,i,0)}$ composed of inner elements. Finally, by (GPath), every element $\mathrm{d}_{(i,j,0)}$ has an outgoing s^{N}-path $\mathrm{d}_{(i,j,1)}$, …, $\mathrm{d}_{(i,j,\mathrm{N})}$ composed of outer elements. Note that by tree-likeness of $\mathcal{I}$ we have that all of selected elements $\mathrm{d}_{(i,j,k)}$ are pairwise-different, and any pair of elements is connected by at most one role. Let $\mathcal{J}$ be the restriction of $\mathcal{I}$ to the set of selected element, with the domain renamed with a map $\mathrm{d}_{(i,j,k)} \mapsto (i,j,k)$. It follows from (GTil) and (GCompl) that $\mathcal{J}$ restricted to $\mathrm{In}^{\mathcal{J}}$ is a T-octant. Other properties of T-hyperoctants can be readily verified with (GCur), (GPrev), (GVerti), (GHori). Thus $\mathcal{J}$ is a T-hyperoctant, as desired. □

The next lemma summarises our reduction.

Lemma 11.40 $\mathcal{D}$ covers the octant if and only if $\mathcal{T}^{\mathcal{D}}_{\blacktriangle} \not\models q^{\mathcal{D}}_{\blacktriangle}$.

Proof. Suppose that $\mathcal{D}$ covers the octant. Then, by Lemma 11.35, there exists a proper T-hyperoctant $\mathcal{I}$. Hence, from Lemma 11.38 we conclude existence of a model $\mathcal{J}$ of $\mathcal{T}^{\mathcal{D}}_{\blacktriangle}$ such that $\mathcal{J} \not\models q^{\mathcal{D}}_{\blacktriangle}$. Thus $\mathcal{T}^{\mathcal{D}}_{\blacktriangle} \not\models q^{\mathcal{D}}_{\blacktriangle}$. For the other direction, suppose that there exists an interpretation $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{T}^{\mathcal{D}}_{\blacktriangle}$ but $\mathcal{I} \not\models q^{\mathcal{D}}_{\blacktriangle}$. By Corollary 11.4 we can assume that $\mathcal{I}$ is single-role tree-like. From Lemma 11.39 we know that $\mathcal{I}$ contains a substructure $\mathcal{J}$ that is a T-hyperoctant. As $\mathcal{I} \not\models q^{\mathcal{D}}_{\blacktriangle}$, we know that $\mathcal{J} \not\models q^{\mathcal{D}}_{\blacktriangle}$. Thus $\mathcal{J}$ is a proper T-hyperoctant. Hence, by Lemma 11.35 we conclude that $\mathcal{D}$ covers the octant. ☐

As the octant tiling problem is undecidable, we conclude the main theorem of this section.

Theorem 11.41

The entailment problem of $\{r, s, r^*, s^*, r^{\#}s^{\#}\}$-conjunctive-queries over $\mathcal{ALC}$-TBoxes is undecidable.

Interestingly enough[4], our proof technique can be adjusted (with little effort) to derive related results for query languages involving inverses. Before moving to the undecidability result, let us define the class of *Visibly-Pushdown Path Queries* [LL15, p. 330] (VPQs) as the class of single-atom $\mathbb{VPL}$-CQs (*i.e.* $\mathbb{VPL}$-CQs without conjunction). We stress that the entailment problem of such queries is decidable. The main idea is that the non-satisfaction of a VPQ of the form $\mathcal{L}(x, y)$ can be expressed with an $\mathcal{ALC}_{\mathsf{vpl}}$-GCI $\top \sqsubseteq \neg\exists\mathcal{L}.\top$. Indeed:

Corollary 11.42

The entailment problem of VPQs over $\mathcal{ALC}_{\mathsf{vpl}}$-TBoxes is 2ExpTime-complete (assuming that the $\mathbb{VPL}$s from the query and the TBox are over the same alphabet).

Proof. The lower bound is inherited from the concept satisfiability problem for $\mathcal{ALC}_{\mathsf{vpl}}$ [LLS07, Thm. 19]. Let $\mathcal{T}$ be an $\mathcal{ALC}_{\mathsf{vpl}}$-TBox and q be a VPQ of the form $\mathcal{L}(x, y)$. The crucial observation is that by the semantics of queries we have that $\mathcal{T} \not\models q$ if and only if $\mathcal{T}' := \mathcal{T} \cup \{\top \sqsubseteq \neg\exists\mathcal{L}.\top\}$ has a model. Hence, by a well-known internalisation of TBoxes as concepts in the presence of regular expressions [BCM+03, p. 186], we can compute (in time polynomial w.r.t. $|\mathcal{T}|$) an $\mathcal{ALC}_{\mathsf{vpl}}$-concept C that is satisfiable if and only if $\mathcal{T}'$ is. Now it suffices to check the satisfiability of C. This in turn can be done in doubly-exponential time by the result of Löding et al. [LLS07, Thm. 18], finishing the proof. ☐

We next extend the class of VPQs with an inverse operator, obtaining the class of *Two-Way Visibly Pushdown Path Queries* (2VPQs). More precisely, such queries are VPQs that allow for letters of the form r^- for all role names $r \in \mathbf{N_R}$ in the underlying visibly-pushdown alphabet. When evaluating 2VPQs over interpretations $\mathcal{I}$, the role $(r^-)^{\mathcal{I}}$ is interpreted as the set-theoretic inverse of the role $r^{\mathcal{I}}$. In what follows we sketch the proof of the fact that the entailment of 2VPQs over $\mathcal{ALC}$-TBoxes is undecidable. The key ingredients of our reduction are the previously-defined $\mathcal{ALC}$-TBox $\mathcal{T}^{\mathcal{D}}_{\blacktriangle}$ and a fresh 2VPQ $q^{\mathcal{D}}_{\mathrm{2VPQ}} := \mathcal{L}_{\downarrow\to\uparrow}(x, y)$, where

$$\mathcal{L}_{\downarrow\to\uparrow} := \left\{ \mathrm{Cur}?\ \overline{\mathrm{In}}?\ \left(s^-\ \overline{\mathrm{In}}?\right)^k\ \ s\ (\mathrm{In}?\ s)^m\ (\mathrm{In}?\ r\ \mathrm{In}?)\ (s\ \mathrm{In}?)^m s\ \left(\overline{\mathrm{In}}?\ s\right)^k \overline{\mathrm{In}}?\ \overline{\mathrm{Prev}}? \mid k, m \in \mathbb{N} \right\}.$$

It is an easy exercise to construct a pushdown automaton for the language $\mathcal{L}_{\downarrow\to\uparrow}$. Such an automaton becomes visibly-pushdown under the requirement that s and s^- are, respectively, return and call symbols, and r is an internal symbol. The forthcoming Lemma 11.43 relates the queries $q^{\mathcal{D}}_{\blacktriangle}$ and $q^{\mathcal{D}}_{\mathrm{2VPQ}}$. As it can be established analogously to the other lemmas of this section, we only sketch the proof and leave some minor details to the reader.

[4]I would like to thank Anni-Yasmin Turhan for asking this question.

Lemma 11.43 Let $\mathcal{D} := (\mathrm{Col}, \mathrm{T}, \square)$ be a domino tilling system and let $\mathcal{I}$ be a T-hyperoctant. We have that $\mathcal{I} \models q^{\mathcal{D}}_{\blacktriangle}$ if and only if $\mathcal{I} \models q^{\mathcal{D}}_{2\mathrm{VPQ}}$. Moreover, by applying the construction from Lemma 11.38, every proper T-hyperoctant can be extended to a model of $\mathcal{T}^{\mathcal{D}}_{\blacktriangle}$ that violates $q^{\mathcal{D}}_{2\mathrm{VPQ}}$.

Proof sketch. We first sketch the proof of the equivalence $\mathcal{I} \models q^{\mathcal{D}}_{\blacktriangle}$ if and only if $\mathcal{I} \models q^{\mathcal{D}}_{2\mathrm{VPQ}}$. First, take any match η witnessing $\mathcal{I} \models q^{\mathcal{D}}_{\blacktriangle}$. Then it can be readily verified that the mapping $x \mapsto \eta(v_1), y \mapsto \eta(v_2)$ is a match for $\mathcal{I}$ and $q^{\mathcal{D}}_{2\mathrm{VPQ}}$. For the opposite direction, let $\mathrm{d}, \mathrm{e} \in \Delta^{\mathcal{I}}$ be domain elements for which the mapping $x \mapsto \mathrm{d}, y \mapsto \mathrm{e}$ is a match for $\mathcal{I}$ and $q^{\mathcal{D}}_{2\mathrm{VPQ}}$. Hence, e is $[\overline{\mathrm{In}}?\ \left(s^{-}\ \overline{\mathrm{In}}?\right)^{k}\ s\ (\mathrm{In}?\ s)^{m}\ (\mathrm{In}?\ r\ \mathrm{In}?)\ (s\ \mathrm{In}?)^{m} s\ \left(\overline{\mathrm{In}}?\ s\right)^{k} \overline{\mathrm{In}}?]$-reachable from d for some integers $m, k \in \mathbb{N}$. By analysing the shape of T-hyperoctants and the above expression, we infer the existence of an integer $n \in \mathbb{N}$ for which $\mathrm{d} = (n, m, k)$, and $\mathrm{e} = (n{+}1, m, k)$. Indeed, the equality of the second and third coordinates is ensured by the presence of m and k in the above path expression; the fact that the first coordinate of e is the successor value of the first coordinate of d is guaranteed by the subexpression $(\mathrm{In}?\ r\ \mathrm{In}?)$ in $q^{\mathcal{D}}_{2\mathrm{VPQ}}$. Now the match for $q^{\mathcal{D}}_{\blacktriangle}$ and $\mathcal{I}$ can be defined as: $v_1 \mapsto (n, m, k)$, $v_2 \mapsto (n{+}1, m, k)$, $w_1 \mapsto (n, m, 1)$, $w_2 \mapsto (n{+}1, m, 1)$, $z_1 \mapsto (n, m, 0)$, $z_2 \mapsto (n{+}1, m, 0)$, $y_1 \mapsto (n, 0, 0)$, $y_2 \mapsto (n{+}1, 0, 0)$, $x_1 \mapsto (n{-}m, 0, 0)$, $x_2 \mapsto (n{-}m{+}1, 0, 0)$, $u_1 \mapsto (n{-}m{-}k, 0, 0)$, $u_2 \mapsto (n{-}m{-}k{+}1, 0, 0)$.

For the remaining part of the proof, take any proper T-hyperoctant $\mathcal{I}$. We apply the construction from the proof of Lemma 11.38, and obtain a model $\mathcal{J}$ of $\mathcal{T}^{\mathcal{D}}_{\blacktriangle}$ that violates $q^{\mathcal{D}}_{\blacktriangle}$. It suffices to show that $\mathcal{J} \not\models q^{\mathcal{D}}_{2\mathrm{VPQ}}$. Towards a contradiction, suppose the opposite, and take any match η for $\mathcal{J}$ and $q^{\mathcal{D}}_{2\mathrm{VPQ}}$. Then, by the construction of $\mathcal{J}$ and the shape of $q^{\mathcal{D}}_{2\mathrm{VPQ}}$, we conclude that the image of η belongs to the T-hyperoctant part $\mathcal{I}$ of $\mathcal{J}$. By equivalence between the queries $q^{\mathcal{D}}_{2\mathrm{VPQ}}$ and $q^{\mathcal{D}}_{\blacktriangle}$ sketched above, we conclude $\mathcal{I} \models q^{\mathcal{D}}_{\blacktriangle}$. A contradiction. $\square$

The second auxiliary lemma has a proof analogous to the proof Lemma 11.40.

Lemma 11.44 Let $\mathcal{D} := (\mathrm{Col}, \mathrm{T}, \square)$ be a domino tilling system and let $\mathcal{I}$ be a T-hyperoctant. We have that $\mathcal{T}^{\mathcal{D}}_{\blacktriangle} \not\models q^{\mathcal{D}}_{2\mathrm{VPQ}}$ if and only if $\mathcal{D}$ covers the octant.

Proof. If $\mathcal{D}$ covers the octant, by Lemma 11.35 there exists a proper T-hyperoctant $\mathcal{I}$. Applying the second part of Lemma 11.43, we extend $\mathcal{I}$ to a model of $\mathcal{T}^{\mathcal{D}}_{\blacktriangle}$ violating $\mathcal{J} \not\models q^{\mathcal{D}}_{2\mathrm{VPQ}}$. Thus $\mathcal{T}^{\mathcal{D}}_{\blacktriangle} \not\models q^{\mathcal{D}}_{2\mathrm{VPQ}}$. For the reverse direction, take any model $\mathcal{I}$ of $\mathcal{T}^{\mathcal{D}}_{\blacktriangle}$ such that $\mathcal{I} \not\models q^{\mathcal{D}}_{2\mathrm{VPQ}}$. By Corollary 11.4 we can assume that $\mathcal{I}$ is single-role tree-like. From Lemma 11.39 we know that $\mathcal{I}$ contains a substructure $\mathcal{J}$ that is a T-hyperoctant. As $\mathcal{I} \not\models q^{\mathcal{D}}_{2\mathrm{VPQ}}$, we know that $\mathcal{J} \not\models q^{\mathcal{D}}_{2\mathrm{VPQ}}$, and thus, by Lemma 11.43, we know that $\mathcal{J} \not\models q^{\mathcal{D}}_{\blacktriangle}$. Hence, by Lemma 11.36, $\mathcal{J}$ is a proper T-hyperoctant. Invoking Lemma 11.35 we conclude that $\mathcal{D}$ covers the octant. $\square$

Linking Lemma 11.44 with the undecidability of the tiling problem (Lemma 11.32), we get:

Theorem 11.45

The entailment problem of 2VPQs over $\mathcal{ALC}$-TBoxes is undecidable.

Once again, let us point out that by the semantics of the query and the logics observe that $\mathcal{T}^{\mathcal{D}}_{\blacktriangle} \not\models q^{\mathcal{D}}_{2\mathrm{VPQ}}$ holds if and only if $\mathcal{T}^{\mathcal{D}}_{\blacktriangle} \cup \{\top \sqsubseteq \neg\exists\mathcal{L}_{\downarrow\rightarrow\uparrow}.\top\}$ (which is written in $\mathcal{ALC}_{\mathsf{vpl}}\mathcal{I}$, namely the extension of $\mathcal{ALC}_{\mathsf{vpl}}$ with the inverse operator) is satisfiable. Such a reduction yields undecidability of the concept satisfiability problem for $\mathcal{ALC}_{\mathsf{vpl}}\mathcal{I}$, and thus reproves the previously-established result of Göller [Göl08, Prop. 2.32] (Corollary 11.5 in Preliminaries).

We conclude the section by revisiting known results concerning query entailment in (extensions of) $\mathcal{ALC}$, and lifting them to the case of $\mathcal{ALC}_{\mathsf{vpl}}$. This contrasts with Theorem 11.41.

Corollary 11.46

The query entailment problem for the class of Positive Conjunctive Regular Path Queries over $\mathcal{ALC}_{\mathsf{vpl}}$-TBoxes is 2ExpTime-complete.

Proof sketch. Note that all the results given above transfer immediately to the case of TBoxes, as they can be internalised in concepts in the presence of regular expressions [BCM+03, p. 186]. Hence, it suffices to focus on $\mathcal{ALC}_{\mathsf{vpl}}$-concepts only. Let C be an input $\mathcal{ALC}_{\mathsf{vpl}}$-concept and q be a $\mathbb{REG}$-PEQ. Löding et al. introduced [LLS07, p. 55] a model of visibly pushdown tree automata that has the following properties (unfortunately all of them are only implicit in the paper): (a) generalises nondeterministic tree automata, (b) is closed under intersection (and an automaton recognizing the intersection of languages can be computed in polynomial time), and (c) its non-emptiness can be tested in exponential time [LLS07, Thm. 4]. They also provided [LLS07, Sec. 4.2] an automaton $\mathcal{A}_{\mathrm{C}}$ that accepts precisely (suitably) single-role tree-like models of C, and the size of $\mathcal{A}_{\mathrm{C}}$ is exponential w.r.t. the size of the concept C [LLS07, Lemma 17]. On the other hand, Gutiérrez-Basulto et al. provide [GIJM23, Lemma 8] a non-deterministic tree automaton $\mathcal{A}_{\neg q}$, of size exponential w.r.t. the sizes of C and q, that accepts all single-role tree-like structures that do not contain any matches of q.[5] It follows then that the intersection of the languages of $\mathcal{A}_{\mathrm{C}}$ and $\mathcal{A}_{\neg q}$ is non-empty if and only if $\mathrm{C} \not\models q$. As visibly pushdown tree automata are closed under intersection [LLS07, p. 55], and their non-emptiness can be solved in exponential time [LLS07, Thm. 4], we infer that the non-emptiness of $\mathcal{A}_{\mathrm{C}} \cap \mathcal{A}_{\neg q}$ can be tested in doubly-exponential time w.r.t. the sizes of C and q. The matching lower bound is inherited from the concept satisfiability. □

11.5 Open Problems

We investigated the decidability status of extensions of $\mathcal{ALC}_{\mathsf{vpl}}$ (also known as Propositional Dynamic Logic with Visibly Pushdown Programs) with popular features supported by W3C ontology languages. While undecidability of $\mathcal{ALC}_{\mathsf{vpl}}$ with inverses or role-hierarchies follows from existing work, we provided undecidability proofs of $\mathcal{ALC}_{\mathsf{vpl}}$ extended with the Self operator (Section 11.2), with nominals (Section 11.3), or non-regular queries (Section 11.4). The following questions remain open.

- Our undecidability proof for $\mathcal{ALC}_{\mathsf{vpl}}^{\mathsf{Self}}$ relied on the availability of multiple visibly-pushdown languages that are encodings on deterministic one-counter languages. Can our undecidability proof be sharpened? For instance, is the concept satisfiability of $\mathcal{ALC}_{\mathsf{reg}}^{r\#s\#}$ with Self already undecidable?
- Positive results for $\mathcal{ALC}_{\mathsf{vpl}}$ [LLS07, Thm. 18] concern the concept satisfiability problem, rather than the knowledge-base satisfiability problem. Is the later decidable for $\mathcal{ALC}_{\mathsf{vpl}}$? Classical techniques [DL94, p. 210] for incorporating ABoxes inside concepts do not work, as the class of visibly-pushdown languages is not compositional (is of "infinite memory"). Note that this problem already occurs for $\mathcal{ALC}_{\mathsf{reg}}^{r\#s\#}$.
- Is an extension of $\mathcal{ALC}_{\mathsf{vpl}}$ (or even $\mathcal{ALC}_{\mathsf{reg}}^{r\#s\#}$) with functionality decidable? De Giacomo and Lenzerini [DL94, p. 211] proposed a satisfiability-preserving translation from $\mathcal{ALC}_{\mathsf{reg}}$ with functionality to plain $\mathcal{ALC}_{\mathsf{reg}}$. Unfortunately, this reduction does not seem to be applicable to $\mathcal{ALC}_{\mathsf{vpl}}$. The reason is again that visibly-pushdown languages are not compositional. Also, functionality violates a crucial condition of "unique diamond-path property" from the decidability proof of $\mathcal{ALC}_{\mathsf{vpl}}$ [LLS07, Def. 11].
- Existing positive results on non-regular extensions of $\mathcal{ALC}_{\mathsf{reg}}$, especially these of Löding et al. [LLS07, Thm. 18], rely on the use of (potentially infinite) tree-like models. Is the *finite* satisfiability problem for $\mathcal{ALC}_{\mathsf{vpl}}$ decidable? Already the case of $\mathcal{ALC}_{\mathsf{reg}}^{r\#s\#}$ is open.

[5] In the paper of Gutiérrez-Basulto et al., the query automaton is presented in a more general setting of tree decomposition as it was designed to also work for the case of knowledge-bases (not just TBoxes). In our paper we deal with concepts only, so such an automaton works on trees as usual.

Conclusions

In this thesis we provided a series of novel, mathematically intriguing and relatively technical results concerning fundamental database-inspired problems in the realm of logic-based knowledge representation. As we do not want to repeat all the results, we recall the most important of them below.

- In Part I we devised a new meta-algorithm based on Lutz's spoiler technique that yields tight complexity results for the entailment of unions of conjunctive queries over ontologies that are forest-friendly (their models locally resemble forests). As a by-product, this allowed us to establish (in a uniform way) ExpTime-completeness of finite and unrestricted entailment of UCQs over $\mathcal{ZQ}^{-}$, *i.e.* the logic $\mathcal{ZQ}$ without the Self operator. For $\mathcal{ALC}$ extended with the Self operator, we established a novel 2ExpTime-hardness result concerning the query entailment (Chapter 6).
- In Part II we proposed a new algorithm for deciding quasi-forest satisfiability of $\mathcal{ZOIQ}$-KBs. We employed the algorithm to establish NP-completeness (w.r.t. the data complexity) of the satisfiability problem for the maximal decidable sublogics $\mathcal{ZOQ}$, $\mathcal{ZIQ}$, and $\mathcal{ZOI}$ of $\mathcal{ZOIQ}$, closing a problem open for more than a decade. As the second application of our algorithm, we established coNExpTime-completeness (w.r.t. the combined complexity) of the entailment problem of rooted queries over $\mathcal{ZIQ}$, supplemented with a novel matching lower bound for $\mathcal{ALC}$ with the Self operator.
- In Part III we designed a uniform, efficient reduction from the entailment of positive regular path queries over (tamed) $\mathcal{ZOIQ}$-KBs to their satisfiability problem. Our approach is not only significantly simpler than the existing automata-based approaches, but exponentially improves previously established upper bounds on querying (for fragments of $\mathcal{ZOIQ}$ involving counting). We transferred parts of our results to the realm of the finite model theory, by establishing that the logics $\mathcal{ZOQ}$ and $\mathcal{ZOI}$ are finitely-controllable for the class of positive existential queries.
- Finally, Part IV investigates the decidability border of non-regular extensions of the $\mathcal{Z}$ family of DLs, providing several undecidability results.

Our results are in harmony with the desired conditions of efficiency, robustness, and logic-independence described by Section 1.2 and rely on well-established notions from graph and model theory. While we left quite a lot of problems still open (as discussed at the end of relevant sections), we are confident that we made a significant progress towards understanding the complexity of various DLs as well as that our techniques can be applied in future research.

For future research directions, we are currently working on transferring our results on the data complexity of $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, and $\mathcal{ZOI}$ from Part II to the case of the data complexity of the entailment problem of positive regular path queries, investigated in Chapter 9. We plan to publish the results after the successful defence of this thesis. We also hope to make some progress on establishing at least decidability of the finite satisfiability of $\mathcal{ZIQ}$. We already have a couple of ideas to try on.

Bibliography

[AAB$^+$17] Renzo Angles, Marcelo Arenas, Pablo Barceló, Aidan Hogan, Juan Reutter, and Domagoj Vrgoč. Foundations of Modern Query Languages for Graph Databases. *ACM Computing Surveys*, 50(5), Sep 2017.

[AAB$^+$18] Renzo Angles, Marcelo Arenas, Pablo Barceló, Peter Boncz, George Fletcher, Claudio Gutierrez, Tobias Lindaaker, Marcus Paradies, Stefan Plantikow, Juan Sequeda, Oskar van Rest, and Hannes Voigt. G-CORE: A Core for Future Graph Query Languages. In *Proceedings of the 2018 International Conference on Management of Data*, SIGMOD '18, page 1421–1432, New York, NY, USA, 2018. Association for Computing Machinery.

[AHD10] Carlos Areces, Guillaume Hoffmann, and Alexandre Denis. Modal Logics with Counting. In Anuj Dawar and Ruy J. G. B. de Queiroz, editors, *Logic, Language, Information and Computation, 17th International Workshop, WoLLIC 2010, Brasilia, Brazil, July 6-9, 2010. Proceedings*, volume 6188 of *Lecture Notes in Computer Science*, pages 98–109. Springer, 2010.

[AM09] Rajeev Alur and Parthasarathy Madhusudan. Adding Nesting Structure to Words. *Journal of ACM*, 56(3), may 2009.

[Baa17] Franz Baader. A New Description Logic with Set Constraints and Cardinality Constraints on Role Successors. In Clare Dixon and Marcelo Finger, editors, *Frontiers of Combining Systems - 11th International Symposium, FroCoS 2017, Brasília, Brazil, September 27-29, 2017, Proceedings*, volume 10483 of *Lecture Notes in Computer Science*, pages 43–59. Springer, 2017.

[Bar13] Pablo Barceló. Querying Graph Databases. In *Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, PODS '13, page 175–188, New York, NY, USA, 2013. Association for Computing Machinery.

[BBL05] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the EL Envelope. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, page 364–369, San Francisco, CA, USA, 2005. Morgan Kaufmann Publishers Inc.

[BBL15] Franz Baader, Stefan Borgwardt, and Marcel Lippmann. Temporal Query Entailment in the Description Logic SHQ. *Journal of Web Semantics*, 33:71–93, 2015. Ontology-based Data Access.

[BBLW16] Franz Baader, Meghyn Bienvenu, Carsten Lutz, and Frank Wolter. Query and Predicate Emptiness in Ontology-Based Data Access. *Journal of Artificial Intelligence Research*, 56:1–59, 2016.

[BBR19] Franz Baader, Bartosz Bednarczyk, and Sebastian Rudolph. Satisfiability Checking and Conjunctive Query Answering in Description Logics with Global and Local Cardinality Constraints. In Mantas Šimkus and Grant Weddell, editors, *Proceedings of the 32nd International Workshop on Description Logics, Oslo, Norway, June 18-21, 2019*, volume 2373 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019.

[BBR20] Franz Baader, Bartosz Bednarczyk, and Sebastian Rudolph. Satisfiability and Query Answering in Description Logics with Global and Local Cardinality Constraints. In Giuseppe De Giacomo, Alejandro Catalá, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarín, and Jérôme Lang, editors, *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020*, volume 325, pages 616–623. IOS Press, 2020.

[BBV16] Michael Benedikt, Pierre Bourhis, and Michael Vanden Boom. A Step Up in Expressiveness of Decidable Fixpoint Logics. In Martin Grohe, Eric Koskinen, and Natarajan Shankar, editors, *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, pages 817–826. ACM, 2016.

[BCM$^+$03] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, 2003.

[BCOv14] Meghyn Bienvenu, Diego Calvanese, Magdalena Ortiz, and Mantas Šimkus. Nested Regular Path Queries in Description Logics. In Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014.* AAAI Press, 2014.

[BD19] Franz Baader and Filippo De Bortoli. On the Expressive Power of Description Logics with Cardinality Constraints on Finite and Infinite Sets. In Andreas Herzig and Andrei Popescu, editors, *Frontiers of Combining Systems - 12th International Symposium, FroCoS 2019, London, UK, September 4-6, 2019, Proceedings*, volume 11715 of *Lecture Notes in Computer Science*, pages 203–219. Springer, 2019.

[BDG$^+$10] Davide Bresolin, Dario Della Monica, Valentin Goranko, Angelo Montanari, and Guido Sciavicco. Undecidability of the Logic of Overlap Relation over Discrete Linear Orderings. *Electronic Notes in Theoretical Computer Science*, 262:65–81, 2010. Proceedings of the 6th Workshop on Methods for Modalities (M4M-6 2009).

[Bed20] Bartosz Bednarczyk. Satisfiability and Query Answering in Description Logics with Global and Local Cardinality Constraints. `http://bartoszjanbednarczyk.github.io/slides/ECAI2020-video.mov`, 2020. [Online; accessed 16-April-2024].

[Bed21a] Bartosz Bednarczyk. Exploiting Forwardness: Satisfiability and Query-Entailment in Forward Guarded Fragment. In Wolfgang Faber, Gerhard Friedrich, Martin Gebser, and Michael Morak, editors, *Logics in Artificial Intelligence - 17th European Conference, JELIA 2021, Virtual Event, May 17-20, 2021, Proceedings*, volume 12678 of *Lecture Notes in Computer Science*, pages 179–193. Springer, 2021.

[Bed21b] Bartosz Bednarczyk. Lutz's Spoiler Technique Revisited: A Unified Approach to Worst-Case Optimal Entailment of Unions of Conjunctive Queries in Locally-Forward Description Logics. *ArXiv*, abs/2108.05680, 2021.

[Bed21c] Bartosz Bednarczyk. Statistical EL is ExpTime-Complete. *Information Processing Letters*, 169:106113, 2021.

[Bed23] Bartosz Bednarczyk. Beyond $\mathcal{ALC}_{\mathrm{reg}}$: Exploring Non-Regular Extensions of PDL with Description Logics Features. In Sarah Alice Gaggl, Maria Vanina Martinez, and Magdalena Ortiz, editors, *Logics in Artificial Intelligence - 18th European Conference, JELIA 2023, Dresden, Germany, September 20-22, 2023, Proceedings*, volume 14281 of *Lecture Notes in Computer Science*, pages 289–305. Springer, 2023.

[Bed24a] Bartosz Bednarczyk. Data Complexity in Expressive Description Logics With Path Expressions. In *IJCAI 2024, Proceedings of the 33rd International Joint Conference on Artificial Intelligence, Jeju, South Korea, August 3-9, 2024*, 2024.

[Bed24b] Bartosz Bednarczyk. Exploring Non-Regular Extensions of Propositional Dynamic Logic with Description-Logics Features. *Logical Methods Computer Science*, 2024.

[BGO14] Vince Bárány, Georg Gottlob, and Martin Otto. Querying the Guarded Fragment. *Logical Methods in Computer Science*, 10(2), 2014.

[BHLS17] Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017.

[BHS05] Franz Baader, Ian Horrocks, and Ulrike Sattler. Description Logics as Ontology Languages for the Semantic Web. In Dieter Hutter and Werner Stephan, editors, *Mechanizing Mathematical Reasoning, Essays in Honor of Jörg H. Siekmann on the Occasion of His 60th Birthday*, volume 2605 of *Lecture Notes in Computer Science*, pages 228–248. Springer, 2005.

[BK22] Bartosz Bednarczyk and Emanuel Kieroński. Finite Entailment of Local Queries in the Z Family of Description Logics. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 5487–5494. AAAI Press, 2022.

[BKR14] Pierre Bourhis, Markus Krötzsch, and Sebastian Rudolph. How to Best Nest Regular Path Queries. In Meghyn Bienvenu, Magdalena Ortiz, Riccardo Rosati, and Mantas Šimkus, editors, *Informal Proceedings of the 27th International Workshop on Description Logics, Vienna, Austria, July 17-20, 2014*, volume 1193 of *CEUR Workshop Proceedings*, pages 404–415. CEUR-WS.org, 2014.

[BKW21] Bartosz Bednarczyk, Emanuel Kieroński, and Piotr Witkowski. Completing the Picture: Complexity of Graded Modal Logics with Converse. *Theory and Practice of Logic Programming*, 21(4):493–520, 2021.

[BL21] Florian Bruse and Martin Lange. A Decidable Non-Regular Modal Fixpoint Logic. In Serge Haddad and Daniele Varacca, editors, *32nd International Conference on Concurrency Theory, CONCUR 2021, August 24-27, 2021, Virtual Conference*, volume 203 of *LIPIcs*, pages 23:1–23:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[BLMV08] Piero A. Bonatti, Carsten Lutz, Aniello Murano, and Moshe Y. Vardi. The Complexity of Enriched Mu-Calculi. *Logical Methods Computer Science*, 4(3), 2008.

[BLW12] Meghyn Bienvenu, Carsten Lutz, and Frank Wolter. Query Containment in Description Logics Reconsidered. In Gerhard Brewka, Thomas Eiter, and Sheila A. McIlraith, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10-14, 2012*. AAAI Press, 2012.

[BO15] Meghyn Bienvenu and Magdalena Ortiz. Ontology-Mediated Query Answering with Data-Tractable Description Logics. In Wolfgang Faber and Adrian Paschke, editors, *Reasoning Web. Web Logic Rules - 11th International Summer School 2015, Berlin, Germany, July 31 - August 4, 2015, Tutorial Lectures*, volume 9203 of *Lecture Notes in Computer Science*, pages 218–307. Springer, 2015.

[Boa97] Peter van Emde Boas. The Convenience of Tilings. *Lecture Notes in Pure and Applied Mathematics*, pages 331–363, 1997.

[BOv15] Meghyn Bienvenu, Magdalena Ortiz, and Mantas Šimkus. Regular Path Queries in Lightweight Description Logics: Complexity and Algorithms. *Journal of Artificial Intelligence Research*, 53:315–374, 2015.

[BR19] Bartosz Bednarczyk and Sebastian Rudolph. Worst-Case Optimal Querying of Very Expressive Description Logics with Path Expressions and Succinct Counting. In Sarit Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 1530–1536. ijcai.org, 2019.

[BR22] Bartosz Bednarczyk and Sebastian Rudolph. The Price of Selfishness: Conjunctive Query Entailment for ALCSelf Is 2EXPTIME-Hard. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Virtual Event, February 22 - March 1, 2022*, volume 36, pages 5495–5502. AAAI Press, Jun. 2022.

[BR23] Bartosz Bednarczyk and Sebastian Rudolph. How to Tell Easy from Hard: Complexity of Conjunctive Query Entailment in Extensions of ALC. *Journal of Artificial Intelligence Research*, 123:1–23, 2023.

[BtCS15] Vince Bárány, Balder ten Cate, and Luc Segoufin. Guarded Negation. *Journal of the ACM*, 62(3):22:1–22:26, 2015.

[BW18] Julian C. Bradfield and Igor Walukiewicz. The Mu-Calculus and Model Checking. In Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem, editors, *Handbook of Model Checking*, pages 871–919. Springer, 2018.

[CDGLR04] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Logical Foundations of Peer-To-Peer Data Integration. In *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 241–251, 2004.

[CEO09] Diego Calvanese, Thomas Eiter, and Magdalena Ortiz. Regular Path Queries in Expressive Description Logics with Nominals. In Craig Boutilier, editor, *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, pages 714–720, 2009.

[CEO14] Diego Calvanese, Thomas Eiter, and Magdalena Ortiz. Answering Regular Path Queries in Expressive Description Logics via Alternating Tree-Automata. *Information and Computation*, 237:12–55, 2014.

[CGLV00] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Moshe Y. Vardi. Containment of Conjunctive Regular Path Queries with Inverse. In Anthony G. Cohn, Fausto Giunchiglia, and Bart Selman, editors, *KR 2000, Principles of Knowledge Representation and Reasoning Proceedings of the Seventh International Conference, Breckenridge, Colorado, USA, April 11-15, 2000*, pages 176–185. Morgan Kaufmann, 2000.

[CKS81] Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. Alternation. *Journal of the ACM*, 28(1):114–133, Jan 1981.

[COv11] Diego Calvanese, Magdalena Ortiz, and Mantas Šimkus. Containment of Regular Path Queries under Description Logic Constraints. In Toby Walsh, editor, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 805–812. IJCAI/AAAI, 2011.

[COv16] Diego Calvanese, Magdalena Ortiz, and Mantas Šimkus. Verification of Evolving Graph-structured Data under Expressive Path Constraints. In Wim Martens and Thomas Zeume, editors, *19th International Conference on Database Theory, ICDT 2016, Bordeaux, France, March 15-18, 2016*, volume 48 of *LIPIcs*, pages 15:1–15:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

[CV93] Surajit Chaudhuri and Moshe Y. Vardi. Optimization of Real Conjunctive Queries. In *Proceedings of the Twelfth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS '93, page 59–70, New York, NY, USA, 1993. Association for Computing Machinery.

[Dem07] Stéphane Demri. *Logiques Pour la Spécification et Vérification.* 2007.

[Die17] Reinhard Diestel. *Graph Theory: 5th edition.* Springer Graduate Texts in Mathematics. Springer-Verlag, 2017.

[DK19] Daniel Danielski and Emanuel Kieroński. Finite Satisfiability of Unary Negation Fragment with Transitivity. In Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen, editors, *44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019, August 26-30, 2019, Aachen, Germany*, volume 138 of *LIPIcs*, pages 17:1–17:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[DL94] Giuseppe De Giacomo and Maurizio Lenzerini. Boosting the Correspondence between Description Logics and Propositional Dynamic Logics. In Barbara Hayes-Roth and Richard E. Korf, editors, *Proceedings of the 12th National Conference on Artificial Intelligence, Seattle, WA, USA, July 31 - August 4, 1994, Volume 1*, pages 205–212. AAAI Press / The MIT Press, 1994.

[DL96] Giuseppe De Giacomo and Maurizio Lenzerini. TBox and ABox Reasoning in Expressive Description Logics. In Luigia Carlucci Aiello, Jon Doyle, and Stuart C. Shapiro, editors, *Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR'96), Cambridge, Massachusetts, USA, November 5-8, 1996*, pages 316–327. Morgan Kaufmann, 1996.

[DL97] Giuseppe De Giacomo and Maurizio Lenzerini. A Uniform Framework for Concept Definitions in Description Logics. *Journal of Artificial Intelligence Research*, 6:87–110, 1997.

[EH85] Ernest Allen Emerson and Joseph Y. Halpern. Decision Procedures and Expressiveness in the Temporal Logic of Branching Time. *Journal of Computer and System Sciences*, 30(1):1–24, 1985.

[ELOv09] Thomas Eiter, Carsten Lutz, Magdalena Ortiz, and Mantas Šimkus. Query Answering in Description Logics with Transitive Roles. In Craig Boutilier, editor, *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, pages 759–764, 2009.

[EOv08] Thomas Eiter, Magdalena Ortiz, and Mantas Šimkus. Reasoning Using Knots. In Iliano Cervesato, Helmut Veith, and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning, 15th International Conference, LPAR 2008, Doha, Qatar, November 22-27, 2008. Proceedings*, volume 5330 of *Lecture Notes in Computer Science*, pages 377–390. Springer, 2008.

[EOv12] Thomas Eiter, Magdalena Ortiz, and Mantas Šimkus. Conjunctive Query Answering in the Description Logic SH Using Knots. *Journal of Computer and System Sciences*, 78(1):47–85, 2012.

[FFB20] Diego Figueira, Santiago Figueira, and Edwin Pin Baque. Finite Controllability for Ontology-Mediated Query Answering of CRPQ. In Diego Calvanese, Esra Erdem, and Michael Thielscher, editors, *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020, Rhodes, Greece, September 12-18, 2020*, pages 381–391, 2020.

[FGG$^+$18] Nadime Francis, Alastair Green, Paolo Guagliardo, Leonid Libkin, Tobias Lindaaker, Victor Marsault, Stefan Plantikow, Mats Rydberg, Petra Selmer, and Andrés Taylor. Cypher: An Evolving Query Language for Property Graphs. In *Proceedings of the 2018 International Conference on Management of Data*, SIGMOD '18, page 1433–1445, New York, NY, USA, 2018. Association for Computing Machinery.

[FGG$^+$23] Nadime Francis, Amélie Gheerbrant, Paolo Guagliardo, Leonid Libkin, Victor Marsault, Wim Martens, Filip Murlak, Liat Peterfreund, Alexandra Rogova, and Domagoj Vrgoc. A Researcher's Digest of GQL (Invited Talk). In Floris Geerts and Brecht Vandevoort, editors, *26th International Conference on Database Theory, ICDT 2023, March 28-31, 2023, Ioannina, Greece*, volume 255 of *LIPIcs*, pages 1:1–1:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[FL79] Michael J. Fischer and Richard E. Ladner. Propositional Dynamic Logic of Regular Programs. *Journal of Computer and System Sciences*, 18(2):194–211, 1979.

[FLOR23] Thomas Feller, Tim S. Lyon, Piotr Ostropolski-Nalewaja, and Sebastian Rudolph. Decidability of Querying First-Order Theories via Countermodels of Finite Width. *CoRR*, abs/2304.06348, 2023.

[FLS98] Daniela Florescu, Alon Y. Levy, and Dan Suciu. Query Containment for Conjunctive Queries with Regular Expressions. In Alberto O. Mendelzon and Jan Paredaens, editors, *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 1-3, 1998, Seattle, Washington, USA*, pages 139–148. ACM Press, 1998.

[GBFF91] Manfred Gehrke, Gerrit Burkert, Peter Forster, and Enrico Franconi. Natural Language Processing and Description Logics. In *Proceedings of the Terminological Logic Users Workshop*, pages 162–164. Department of Computer Science Technische Universität Berlin, 1991.

[GGBIG$^+$19] Tomasz Gogacz, Víctor Gutiérrez-Basulto, Yazmín Ibáñez-García, Jean Christoph Jung, and Filip Murlak. On Finite and Unrestricted Query Entailment beyond SQ with Number Restrictions on Transitive Roles. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 1719–1725. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[GGIM22] Víctor Gutiérrez-Basulto, Albert Gutowski, Yazmín Ibáñez-García, and Filip Murlak. Finite Entailment of UCRPQs over ALC Ontologies. In Gabriele Kern-Isberner, Gerhard Lakemeyer, and Thomas Meyer, editors, *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning, KR 2022, Haifa, Israel, July 31 - August 5, 2022*, 2022.

[GH08] Hermann Gruber and Markus Holzer. Finite Automata, Digraph Connectivity, and Regular Expression Size. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations*, volume 5126 of *Lecture Notes in Computer Science*, pages 39–50. Springer, 2008.

[GH13] Hermann Gruber and Markus Holzer. Provably Shorter Regular Expressions from Finite Automata. *International Journal of Foundations of Computer Science*, 24(8):1255–1280, 2013.

[GH15] Hermann Gruber and Markus Holzer. From Finite Automata to Regular Expressions and Back - A Summary on Descriptional Complexity. *International Journal of Foundations of Computer Science*, 26(8):1009–1040, 2015.

[GHM$^+$08] Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, Bijan Parsia, Peter Patel-Schneider, and Ulrike Sattler. OWL 2: The Next Step for OWL. *Journal of Web Semantics*, 6(4):309–322, 2008.

[GHS08] Birte Glimm, Ian Horrocks, and Ulrike Sattler. Unions of Conjunctive Queries in SHOQ. In Gerhard Brewka and Jérôme Lang, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Eleventh International Conference, KR 2008, Sydney, Australia, September 16-19, 2008*, pages 252–262. AAAI Press, 2008.

[GIJ18] Víctor Gutiérrez-Basulto, Yazmín Angélica Ibáñez-García, and Jean Christoph Jung. Answering Regular Path Queries over SQ Ontologies. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1845–1852. AAAI Press, 2018.

[GIJM23] Víctor Gutiérrez-Basulto, Yazmín Ibáñez-García, Jean Christoph Jung, and Filip Murlak. Answering Regular Path Queries Mediated by Unrestricted SQ Ontologies. *Artificial Intelligence*, 314:103808, 2023.

[Gin68] Abraham Ginzburg. *Algebraic Theory of Automata.* Academic Press, 1968.

[GLHS08] Birte Glimm, Carsten Lutz, Ian Horrocks, and Ulrike Sattler. Conjunctive Query Answering for the Description Logic SHIQ. *Journal of Artificial Intelligence Research*, 31:157–204, 2008.

[GO99] Erich Grädel and Martin Otto. On Logics with Two Variables. *Theoretical Computer Science*, 224(1-2):73–113, 1999.

[GO07] Valentin Goranko and Martin Otto. Model Theory of Modal Logic. In Patrick Blackburn, Johan van Benthem, and Frank Wolter, editors, *Handbook of Modal Logic*, volume 3 of *Studies in logic and practical reasoning*, pages 249–329. North-Holland, 2007.

[Göl08] Stefan Göller. *Computational Complexity of Propositional Dynamic Logics.* PhD thesis, University of Leipzig, 2008.

[Grä99] Erich Grädel. On The Restraining Power of Guards. *Journal Symbolic Logic*, 64(4):1719–1742, 1999.

[Gru12] Hermann Gruber. Digraph Complexity Measures and Applications in Formal Language Theory. *Discrete Mathematics and Theoretical Computer Science*, 14(2):189–204, 2012.

[Gru23] Hermann Gruber. An Answer To: The Complexity of Conversion From a Regular Expression to a Nondeterminsitic Automata and Back After Changing Initial and Final States. https://cstheory.stackexchange.com/q/53677, 2023. [Online; accessed 16-April-2024].

[HKP$^+$12] Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, and Sebastian Rudolph. *OWL 2 Web Ontology Language Primer (Second Edition).* World Wide Web Consortium (W3C), December 2012.

[HKS06] Ian Horrocks, Oliver Kutz, and Ulrike Sattler. The Even More Irresistible SROIQ. In Patrick Doherty, John Mylopoulos, and Christopher A. Welty, editors, *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2-5, 2006*, pages 57–67. AAAI Press, 2006.

[HP84] David Harel and MS Paterson. Undecidability of PDL with L $= \{a^{2^i} \mid i \geq 0\}$. *Journal of Computer and System Sciences*, 29(3):359–365, 1984.

[HPS81a] David Harel, Amir Pnueli, and Jonathan Stavi. Further Results on Propositional Dynamic Logic of Nonregular Programs. In Dexter Kozen, editor, *Logics of Programs, Workshop, Yorktown Heights, New York, USA, May 1981*, volume 131 of *Lecture Notes in Computer Science*, pages 124–136. Springer, 1981.

[HPS81b] David Harel, Amir Pnueli, and Jonathan Stavi. Propositional Dynamic Logic of Context-Free Programs. In *22nd Annual Symposium on Foundations of Computer Science (SFCS 1981)*, pages 310–321. IEEE, 1981.

[HR93] David Harel and Danny Raz. Deciding Properties of Nonregular Programs. *SIAM Journal on Computing*, 22(4):857–874, 1993.

[HRG$^+$96] Ian Horrocks, Alan L Rector, Carole A Goble, Franz Baader, Martin Buchheit, Manfred A Jeusfeld, and Werner Nutt. A Description Logic Based Schema for the Classification of Medical Data. In Franz Baader, Martin Buchheit, Manfred A Jeusfeld, and Werner Nutt, editors, *KRDB*, volume 4 of *CEUR Workshop Proceedings*, Germany, 1996. RWTH Aachen University. Knowledge Representation Meets Databases, Proceedings of the 3rd Workshop KRDB'96, Budapest, Hungary, August 13, 1996 ; Conference date: 01-01-1824.

[HS96] David Harel and Eli Singerman. More on Nonregular PDL: Finite Models and Fibonacci-Like Programs. *Information and Computation*, 128(2):109–118, 1996.

[HS13] Steve Harris and Andy Seaborne. SPARQL 1.1 Query Language, March 2013.

[HS22] Daniel Hausmann and Lutz Schröder. Coalgebraic Satisfiability Checking for Arithmetic μ-Calculi. *CoRR*, abs/2212.11055, 2022.

[HT00] Ian Horrocks and Sergio Tessaris. Answering Conjunctive Queries over DL ABoxes: A Preliminary Report. In Franz Baader and Ulrike Sattler, editors, *Proceedings of the 2000 International Workshop on Description Logics (DL2000), Aachen, Germany, August 17-19, 2000*, volume 33 of *CEUR Workshop Proceedings*, pages 173–182. CEUR-WS.org, 2000.

[ILS14] Yazmín Angélica Ibáñez-García, Carsten Lutz, and Thomas Schneider. Finite Model Reasoning in Horn Description Logics. In Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014*. AAAI Press, 2014.

[JLMS18] Jean Christoph Jung, Carsten Lutz, Mauricio Martel, and Thomas Schneider. Querying the Unary Negation Fragment with Regular Path Expressions. In Benny Kimelfeld and Yael Amsterdamer, editors, *21st International Conference on Database Theory, ICDT 2018, March 26-29, 2018, Vienna, Austria*, volume 98 of *LIPIcs*, pages 15:1–15:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[JLPW22] Jean Christoph Jung, Carsten Lutz, Hadrien Pulcini, and Frank Wolter. Logical Separability of Labeled Data Examples Under Ontologies. *Artificial Intelligence*, 313:103785, 2022.

[JLZ20] Jean Christoph Jung, Carsten Lutz, and Thomas Zeume. On the Decidability of Expressive Description Logics with Transitive Closure and Regular Role Expressions. In Diego Calvanese, Esra Erdem, and Michael Thielscher, editors, *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020, Rhodes, Greece, September 12-18, 2020*, pages 529–538, 2020.

[Kaz08] Yevgeny Kazakov. RIQ and SROIQ are harder than SHOIQ. In Gerhard Brewka and Jérôme Lang, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Eleventh International Conference, KR 2008, Sydney, Australia, September 16-19, 2008*, pages 274–284. AAAI Press, 2008.

[Koo19] Patrick Koopmann. Ontology-Based Query Answering for Probabilistic Temporal Data. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 2903–2910. AAAI Press, 2019.

[KP83] Tmima Koren and Amir Pnueli. There Exist Decidable Context Free Propositional Dynamic Logics. In *Proceedings of the Carnegie Mellon Workshop on Logic of Programs*, pages 290–312, Berlin, Heidelberg, 1983. Springer-Verlag.

[KPS22] Clemens Kupke, Dirk Pattinson, and Lutz Schröder. Coalgebraic Reasoning with Global Assumptions in Arithmetic Modal Logics. *ACM Transactions on Computational Logic*, 23(2), Jan 2022.

[KR07] Viktor Kunčak and Martin C. Rinard. Towards Efficient Satisfiability Checking for Boolean Algebra with Presburger Arithmetic. In Frank Pfenning, editor, *Automated Deduction - CADE-21, 21st International Conference on Automated Deduction, Bremen, Germany, July 17-20, 2007, Proceedings*, volume 4603 of *Lecture Notes in Computer Science*, pages 215–230. Springer, 2007.

[KRH08] Markus Krötzsch, Sebastian Rudolph, and Pascal Hitzler. ELP: Tractable Rules for OWL 2. In Amit P. Sheth, Steffen Staab, Mike Dean, Massimo Paolucci, Diana Maynard, Timothy W. Finin, and Krishnaprasad Thirunarayan, editors, *The Semantic Web - ISWC 2008, 7th International Semantic Web Conference, ISWC 2008, Karlsruhe, Germany, October 26-30, 2008. Proceedings*, volume 5318 of *Lecture Notes in Computer Science*, pages 649–664. Springer, 2008.

[KS14] Mark Kaminski and Gert Smolka. A Goal-Directed Decision Procedure for Hybrid PDL. *Journal of Automated Reasoning*, 52(4):407–450, 2014.

[LAHS04] Carsten Lutz, Carlos Areces, Ian Horrocks, and Ulrike Sattler. Keys, Nominals, and Concrete Domains. *Journal of Artificial Intelligence Research*, 23:667–726, 2004.

[Lib04] Leonid Libkin. *Elements of Finite Model Theory.* Springer, Aug 2004.

[LL15] Martin Lange and Étienne Lozes. Conjunctive Visibly-Pushdown Path Queries. In Adrian Kosowski and Igor Walukiewicz, editors, *Fundamentals of Computation Theory - 20th International Symposium, FCT 2015, Gdańsk, Poland, August 17-19, 2015, Proceedings*, volume 9210 of *Lecture Notes in Computer Science*, pages 327–338. Springer, 2015.

[LLS07] Christof Löding, Carsten Lutz, and Olivier Serre. Propositional Dynamic Logic with Recursive Programs. *Journal of Logical and Algebraic Methods in Programming*, 73(1-2):51–69, 2007.

[LS10] Carsten Lutz and Lutz Schröder. Probabilistic Description Logics for Subjective Uncertainty. In Fangzhen Lin, Ulrike Sattler, and Mirosław Truszczyński, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010.* AAAI Press, 2010.

[Lut02] Carsten Lutz. *The Complexity of Description Logics with Concrete Domains.* PhD thesis, RWTH Aachen University, Germany, 2002.

[Lut08a] Carsten Lutz. The Complexity of Conjunctive Query Answering in Expressive Description Logics. In Alessandro Armando, Peter Baumgartner, and Gilles Dowek, editors, *Automated Reasoning, 4th International Joint Conference, IJCAR 2008, Sydney, Australia, August 12-15, 2008, Proceedings*, volume 5195 of *Lecture Notes in Computer Science*, pages 179–193. Springer, 2008.

[Lut08b] Carsten Lutz. Two Upper Bounds for Conjunctive Query Answering in SHIQ. In Franz Baader, Carsten Lutz, and Boris Motik, editors, *Proceedings of the 21st International Workshop on Description Logics (DL2008), Dresden, Germany, May 13-16, 2008*, volume 353 of *CEUR Workshop Proceedings.* CEUR-WS.org, 2008.

[McG99] Deborah L McGuinness. Ontology-Enhanced Search for Primary Care Medical Literature. In *the Proceedings of the International Medical Informatics Association Working Group*, pages 16–19, 1999.

[MT14] Marie-Laure Mugnier and Michaël Thomazo. An Introduction to Ontology-Based Query Answering with Existential Rules. In Manolis Koubarakis, Giorgos B. Stamou, Giorgos Stoilos, Ian Horrocks, Phokion G. Kolaitis, Georg Lausen, and Gerhard Weikum, editors, *Reasoning Web. Reasoning on the Web in the Big Data Era - 10th International Summer School 2014, Athens, Greece, September 8-13, 2014. Proceedings*, volume 8714 of *Lecture Notes in Computer Science*, pages 245–278. Springer, 2014.

[MW95] Alberto O. Mendelzon and Peter T. Wood. Finding Regular Simple Paths in Graph Databases. *SIAM Journal on Computing*, 24(6):1235–1258, 1995.

[Ngu20] Linh Anh Nguyen. Exptime Tableaux with Global Caching for Hybrid PDL. *Journal of Automated Reasoning*, 64(1):21–52, 2020.

[NOv16] Nhung Ngo, Magdalena Ortiz, and Mantas Šimkus. Closed Predicates in Description Logics: Results on Combined Complexity. In Chitta Baral, James Delgrande, and Frank Wolter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25-29, 2016*, pages 237–246. AAAI Press, 2016.

[Noy04] Natalya F Noy. Semantic Integration: A Survey of Ontology-Based Approaches. *ACM Sigmod Record*, 33(4):65–70, 2004.

[Ort10] Magdalena Ortiz. *Query Answering in Expressive Description Logics: Techniques and Complexity Results*. PhD thesis, TU Wien, AT, 2010.

[Ort23] Magdalena Ortiz. A Short Introduction to SHACL for Logicians. In Helle Hvid Hansen, Andre Scedrov, and Ruy J. G. B. de Queiroz, editors, *Logic, Language, Information, and Computation - 29th International Workshop, WoLLIC 2023, Halifax, NS, Canada, July 11-14, 2023, Proceedings*, volume 13923 of *Lecture Notes in Computer Science*, pages 19–32. Springer, 2023.

[ORv10] Magdalena Ortiz, Sebastian Rudolph, and Mantas Šimkus. Worst-Case Optimal Reasoning for the Horn-DL Fragments of OWL 1 and 2. In Fangzhen Lin, Ulrike Sattler, and Mirosław Truszczyński, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010*. AAAI Press, 2010.

[Ott04] Martin Otto. Modal and Guarded Characterisation Theorems over Finite Transition Systems. *Annals of Pure and Applied Logic*, 130(1-3):173–205, 2004.

[Ott12] Martin Otto. Highly Acyclic Groups, Hypergraph Covers, and the Guarded Fragment. *Journal of the ACM*, 59(1):5:1–5:40, 2012.

[Ov12] Magdalena Ortiz and Mantas Šimkus. Reasoning and Query Answering in Description Logics. In Thomas Eiter and Thomas Krennwallner, editors, *Reasoning Web. Semantic Technologies for Advanced Query Answering - 8th International Summer School 2012, Vienna, Austria, September 3-8, 2012. Proceedings*, volume 7487 of *Lecture Notes in Computer Science*, pages 1–53. Springer, 2012.

[Ov14] Magdalena Ortiz and Mantas Šimkus. Revisiting the Hardness of Query Answering in Expressive Description Logics. In Roman Kontchakov and Marie-Laure Mugnier, editors, *Web Reasoning and Rule Systems - 8th International Conference, RR 2014, Athens, Greece, September 15-17, 2014. Proceedings*, volume 8741 of *Lecture Notes in Computer Science*, pages 216–223. Springer, 2014.

[OvE08] Magdalena Ortiz, Mantas Šimkus, and Thomas Eiter. Worst-case Optimal Conjunctive Query Answering for an Expressive Description Logic without Inverses. In Dieter Fox and Carla P. Gomes, editors, *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pages 504–510. AAAI Press, 2008.

[PH23] Ian Pratt-Hartmann. *Fragments of First-Order Logic*. Oxford University Press, 2023.

[Pir12] Robert Piro. *Model-Theoretic Characterisations of Description Logics*. PhD thesis, University of Liverpool, UK, 2012.

[PP17] Rafael Peñaloza and Nico Potyka. Towards Statistical Reasoning in Description Logics over Finite Domains. In Serafín Moral, Olivier Pivert, Daniel Sánchez, and Nicolás Marín, editors, *Scalable Uncertainty Management - 11th International Conference, SUM 2017, Granada, Spain, October 4-6, 2017, Proceedings*, volume 10564 of *Lecture Notes in Computer Science*, pages 280–294. Springer, 2017.

[Pra07] Ian Pratt-Hartmann. Complexity of the Guarded Two-Variable Fragment with Counting Quantifiers. *Journal of Logic and Computation*, 17(1):133–155, 2007.

[Pra09] Ian Pratt-Hartmann. Data-Complexity of the Two-Variable Fragment with Counting Quantifiers. *Information and Computation*, 207(8):867–888, 2009.

[RG10] Sebastian Rudolph and Birte Glimm. Nominals, Inverses, Counting, and Conjunctive Queries or: Why Infinity is your Friend! *Journal of Artificial Intelligence Resesearch*, 39:429–481, 2010.

[RK13] Sebastian Rudolph and Markus Krötzsch. Flag & Check: Data Access with Monadically Defined Queries. In Richard Hull and Wenfei Fan, editors, *Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2013, New York, NY, USA - June 22 - 27, 2013*, pages 151–162. ACM, 2013.

[RKH08] Sebastian Rudolph, Markus Krötzsch, and Pascal Hitzler. Cheap Boolean Role Constructors for Description Logics. In Steffen Hölldobler, Carsten Lutz, and Heinrich Wansing, editors, *Logics in Artificial Intelligence, 11th European Conference, JELIA 2008, Dresden, Germany, September 28 - October 1, 2008. Proceedings*, volume 5293 of *Lecture Notes in Computer Science*, pages 362–374. Springer, 2008.

[RRV17] Juan Reutter, Miguel Romero, and Moshe Vardi. Regular queries on graph databases. *Theory of Computing Systems*, 61(1):31–83, 2017.

[Rud16] Sebastian Rudolph. Undecidability Results for Database-Inspired Reasoning Problems in Very Expressive Description Logics. In Chitta Baral, James Delgrande, and Frank Wolter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25-29, 2016*, pages 247–257. AAAI Press, 2016.

[Sch91] Klaus Schild. A Correspondence Theory for Terminological Logics: Preliminary Report. In John Mylopoulos and Raymond Reiter, editors, *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI 1991)*, pages 466–471. Morgan Kaufmann, 1991.

[Sch93] Andrea Schaerf. On the Complexity of the Instance Checking Problem in Concept Languages with Existential Quantification. *Journal of Intelligent Information Systems*, 2(3):265–278, 1993.

[Sip13] Michael Sipser. *Introduction to the Theory of Computation.* Course Technology, Boston, MA, Third edition, 2013.

[SV01] Ulrike Sattler and Moshe Y. Vardi. The Hybrid μ-Calculus. In Rajeev Goré, Alexander Leitsch, and Tobias Nipkow, editors, *Proc. of the 1st Int. Joint Conf. on Automated Reasoning (IJCAR 2001)*, volume 2083 of *Lecture Notes in Computer Science*, pages 76–91. Springer, 2001.

[Tam15] Kuniaki Tamura. A Small Model Theorem For the Hybrid μ-Calculus. *Journal of Logic and Computation*, 25(2):405–441, 2015.

[Tho68] Ken Thompson. Programming Techniques: Regular Expression Search Algorithm. *Communications of the ACM*, 11:419–422, 1968.

[Val73] Leslie Valiant. *Decision Procedures for Families of Deterministic Pushdown Automata.* PhD thesis, University of Warwick, 1973.

[Var82] Moshe Y. Vardi. The Complexity of Relational Query Languages (Extended Abstract). In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing (STOC 1982)*, STOC '82, page 137–146, New York, NY, USA, 1982. Association for Computing Machinery.

[Var96] Moshe Y. Vardi. Why is Modal Logic So Robustly Decidable? In Neil Immerman and Phokion G. Kolaitis, editors, *Descriptive Complexity and Finite Models, Proceedings of a DIMACS Workshop 1996, Princeton, New Jersey, USA, January 14-17, 1996*, volume 31 of

DIMACS Series in Discrete Mathematics and Theoretical Computer Science, pages 149–183. DIMACS/AMS, 1996.

[Wan61] Hao Wang. Proving Theorems by Pattern Recognition II. *Bell System Technical Journal*, 40(1):1–41, 1961.

[WJ96] Martin Weese Winfried Just. *Discovering Modern Set Theory. I: The Basics.* Discovering Modern Set Theory. American Mathematical Society, 1996.

[ZST13] Michal Zawidzki, Renate Schmidt, and Dmitry Tishkovsky. Satisfiability Problem for Modal Logic with Global Counting Operators Coded in Binary is NExpTime-Complete. *Information Processing Letters*, 113(1-2):34–38, 2013.