Satisfiability and Query Answering in Description Logics with Global and Local Cardinality Constraints

Franz Baader[1], Bartosz Bednarczyk[12], and Sebastian Rudolph[1]

[1] Faculty of Computer Science, TU Dresden, Germany
firstname.lastname@tu-dresden.de
[2] Institute of Computer Science, University of Wrocław, Poland
bartosz.bednarczyk@cs.uni.wroc.pl

Abstract

We introduce and investigate the expressive description logic (DL) $\mathcal{ALCSCC}^{++}$, in which the global and local cardinality constraints introduced in previous papers can be mixed. On the one hand, we prove that this does not increase the complexity of satisfiability checking and other standard inference problems. On the other hand, the satisfiability problem becomes undecidable if inverse roles are added to the languages. In addition, even without inverse roles, conjunctive query entailment in this DL turns out to be undecidable. We prove that decidability of querying can be regained if global and local constraints are not mixed and the global constraints are appropriately restricted. The latter result is based on a locally-acyclic model construction, and it reduces query entailment to ABox consistency in the restricted setting, i.e., to ABox consistency w.r.t. restricted cardinality constraints in $\mathcal{ALCSCC}$, for which we can show an ExpTime upper bound.

1 Introduction

Description Logics (DLs) [7] are a well-investigated family of logic-based knowledge representation languages, which are frequently used to formalize ontologies for application domains such as biology and medicine [14]. To define the important notions of such an application domain as formal concepts, DLs state necessary and sufficient conditions for an individual to belong to a concept. These conditions can be Boolean combinations of atomic properties required for the individual (expressed by concept names) or properties that refer to relationships with other individuals and their properties (expressed as role restrictions). Using an example from [8], the concept of a motor vehicle can be formalized by the concept description

$$\mathit{Vehicle} \sqcap \exists \mathit{part}.\mathit{Motor},$$

which uses the concept names *Vehicle* and *Motor* and the role name *part* as well as the concept constructors conjunction ($\sqcap$) and existential restriction ($\exists r.C$). The concept inclusion (CI)

$$\mathit{Motor\text{-}vehicle} \sqsubseteq \mathit{Vehicle} \sqcap \exists \mathit{part}.\mathit{Motor}$$

then states that every motor vehicle needs to belong to this concept description. Numerical constraints on the number of role successors (so-called number restrictions) have been used early on in DLs [10, 16, 15]. For example, using number restrictions, motorcycles can be constrained to being motor vehicles with exactly two wheels:

$$\mathit{Motorcycle} \sqsubseteq \mathit{Motor\text{-}vehicle} \ \sqcap (\leqslant 2\, \mathit{part}.\mathit{Wheel}) \sqcap (\geqslant 2\, \mathit{part}.\mathit{Wheel}).$$

The exact complexity of reasoning in $\mathcal{ALCQ}$, the DL that has all Boolean operations and number restrictions of the form $(\leqslant n\, r.C)$ and $(\geqslant n\, r.C)$ as concept constructors, was determined by Stephan Tobies [23, 25]: it is PSpace-complete without CIs and ExpTime-complete w.r.t. CIs, independently of whether the numbers occurring in the number restrictions are encoded in unary or binary. Note that, using unary coding of numbers, the number n is assumed to contribute n to the size of the input, whereas with binary coding the size of the number n is $\log n$. Thus, for large numbers, using binary coding is more realistic.

Whereas number restrictions are local in the sense that they consider role successors of an individual under consideration (e.g. the wheels that are part of a particular motor vehicle), cardinality restrictions on concepts (CRs) [6, 24] are global, i.e., they consider all individuals in an interpretation. For example, the cardinality restriction

$$(\leqslant 45000000\, (\mathit{Car} \sqcap \exists \mathit{registered\text{-}in}.\mathit{German\text{-}district}))$$

states that at most 45 million cars are registered all over Germany. Such cardinality restrictions can be seen as quantitative extensions of CIs since a CI of the form $C \sqsubseteq D$ can be expressed by the CR $(\leqslant 0\, (C \sqcap \neg D))$. The availability of CRs increases the complexity of reasoning: as mentioned above, consistency in $\mathcal{ALCQ}$ w.r.t. CIs is ExpTime-complete, but consistency w.r.t. CRs is NExpTime-complete if the numbers occurring in the CRs are assumed to be encoded in binary [24]. With unary coding of numbers, consistency stays ExpTime-complete even w.r.t. CRs [24]. However, as the above example considering 45 million cars indicates, unary coding does not yield a realistic measure for the input size if numbers with large values are employed.

In two previous publications we have, on the one hand, extended the DL $\mathcal{ALCQ}$ by more expressive number restrictions using cardinality and set constraints expressed in the quantifier-free fragment of Boolean Algebra with Presburger Arithmetic (QFBAPA) [17]. In the resulting DL $\mathcal{ALCSCC}$, which was introduced and investigated in [1], cardinality and set constraints are applied locally, i.e., they refer to the role successors of an individual under consideration. For example, we can state that the number of cylinders of a motor must coincide with the number of spark plugs in this motor, without fixing what this number actually is, using the following $\mathcal{ALCSCC}$ CI:

$$\mathit{Motor} \sqsubseteq \mathit{succ}(|\mathit{part} \cap \mathit{Cylinder}| = |\mathit{part} \cap \mathit{SparkPlug}|).$$

It was shown in [1] that pure concept satisfiability in $\mathcal{ALCSCC}$ is a PSpace-complete problem, and concept satisfiability w.r.t. a general TBox is ExpTime-complete. This shows that the more expressive number restrictions do not increase the complexity of reasoning since reasoning in $\mathcal{ALCQ}$ has the same complexity, as mentioned above.

On the other hand, we have extended the terminological formalism of the well-known description logic $\mathcal{ALC}$[1] from CIs not only to CRs, but to more general cardinality constraints expressed in QFBAPA [8], which we called extended cardinality constraints (ECBoxes). These

[1]The DL $\mathcal{ALC}$ is the fragment of $\mathcal{ALCQ}$ in which only number restrictions of the form $(\leqslant 0\, r.\neg C)$ (written $\forall r.C$) and $(\geqslant 1\, r.C)$ (written $\exists r.C$) are available.

constraints are global since they refer to all individuals in the interpretation domain. An example of a constraint expressible this way, but not expressible using CRs is

$$2 \cdot |\mathit{Car} \sqcap \exists \mathit{registered\text{-}in}.\mathit{German\text{-}district} \sqcap \exists \mathit{fuel}.\mathit{Diesel}|$$
$$\leq |\mathit{Car} \sqcap \exists \mathit{registered\text{-}in}.\mathit{German\text{-}district} \sqcap \exists \mathit{fuel}.\mathit{Petrol}|,$$

which states that, in Germany, cars running on petrol outnumber cars running on diesel by a factor of at least two. It was shown in [8] that reasoning w.r.t. ECBoxes is still in NExpTime even if the numbers occurring in the constraints are encoded in binary. The NExpTime lower bound follows from the result of Tobies [24] CRs mentioned above. This complexity can be lowered to ExpTime if a restricted form of cardinality constraints (RCBoxes) is used. Such RCBoxes are still powerful enough to express statistical knowledge bases [19].

An obvious way to generalize these two approaches is to combine the two extensions, i.e., to consider extended cardinality constraints, but now on $\mathcal{ALCSCC}$ concepts rather than just $\mathcal{ALC}$ concepts. This combination was investigated in [2, 3], where a NExpTime upper bound was established for reasoning in $\mathcal{ALCSCC}$ w.r.t. ECBoxes. It is also shown in [2, 3] that reasoning w.r.t. RCBoxes stays in ExpTime also for $\mathcal{ALCSCC}$.

Here we go one step further by allowing for a tighter integration of global and local constraints. The resulting logic, which we call $\mathcal{ALCSCC}^{++}$, allows, for example, to relate the number of role successors of a given individual with the overall number of elements of a certain concept. For example, the $\mathcal{ALCSCC}^{++}$ concept description[2]

$$\mathit{sat}(|\mathit{likes} \cap \mathit{Car}| = |\mathit{Car}|)$$

describes car lovers, i.e., individuals that like *all* cars, independently of whether these cars are related to them by some role or not. More generally, DLs that 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) can, for instance, be used to check the correctness of statistical statements. For example, if a German car company claims that they have produced more than N cars in a certain year, and $P\%$ of the tires used for their cars were produced by Betteryear, this may be contradictory to a statement of Betteryear that they have sold less than M tires in Germany. Such statistical information may, of course, also influence the answers to queries. If we know that the car company VMW uses only tires from Betteryear or Badmonth, but the statistical information allows us to conclude that another car company has actually bought all the tires sold by Betteryear, then we know that the cars sold by VMW all have tires produced by Badmonth. This motivates investigating DLs with expressive cardinality constraints, and to consider not just standard inferences such as satisfiability checking for these DLs, but also query answering.

In the present paper, we show that, from a worst-case complexity point of view, the extended expressivity of $\mathcal{ALCSCC}^{++}$ comes for free if we consider classical reasoning problems. Concept satisfiability in $\mathcal{ALCSCC}^{++}$ has the same complexity as in $\mathcal{ALC}$ and $\mathcal{ALCSCC}$ with global cardinality constraints: it is NExpTime-complete. However, if we add inverse roles, then concept satisfiability becomes undecidable. In addition, for effective conjunctive query answering this logic turns out to be too expressive. We show that conjunctive query entailment w.r.t. $\mathcal{ALCSCC}^{++}$ knowledge bases is, in fact, undecidable. In contrast, we can show that conjunctive query entailment w.r.t. (an extension of) $\mathcal{ALCSCC}$ RCBoxes is decidable and, in fact, only ExpTime-complete. To proof this result, we first show that standard ABox reasoning in

[2] To distinguish between constraint expressions in $\mathcal{ALCSCC}$ and in $\mathcal{ALCSCC}^{++}$, which have a different semantics, we use different keywords for them.

this setting is ExpTime-complete. Then, we reduce query entailment over arbitrary structures to query entailment over locally acyclic graphs, based on an appropriate model construction, which proceeds in three steps. Once this is achieved, the ExpTime upper bound for conjunctive query entailment is shown by a reduction to ABox reasoning, adapting the approach used by Lutz in [18] for $\mathcal{ALCHQ}$.

We assume the reader to be sufficiently familiar with all the standard notions of description logics [7, 9, 22].

2 The logic $\mathcal{ALCSCC}^{++}$

As in [1, 8, 2, 3], we use the quantifier-free fragment of Boolean Algebra with Presburger Arithmetic (QFBAPA) [17] to express our constraints. We start with a brief introduction of QFBAPA (see [17] and [1] for more details).

In the logic QFBAPA, one can build *set terms* by applying Boolean operations (intersection $\cap$, union $\cup$, and complement $\cdot^c$) to set variables as well as the constants $\emptyset$ and $\mathcal{U}$. Set terms s, t can then be used to state *set constraints*, which are equality and inclusion constraints of the form $s = t, s \subseteq t$, where s, t are set terms. *Presburger Arithmetic (PA) expressions* are built from integer constants and set cardinalities $|s|$ using addition as well as multiplication with an integer constant.[3] They can be used to form *cardinality constraints* of the form $k = \ell, k < \ell, N \,\mathsf{dvd}\, \ell$, where k, ℓ are PA expressions, N is an integer constant, and dvd stands for divisibility. A *QFBAPA formula* is a Boolean combination of set and cardinality constraints using the Boolean operations $\wedge, \vee, \neg$.

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 $s = t$ ($s \subseteq t$) if $\sigma(s) = \sigma(t)$ ($\sigma(s) \subseteq \sigma(t)$). It is further extended to a mapping from PA expressions to integers by interpreting $|s|$ as the cardinality of the finite set $\sigma(s)$, and 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 $N \,\mathsf{dvd}\, \ell$ 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 the 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. In [17] it is shown that the satisfiability problem for QFBAPA formulae is NP-complete.

We are now ready to define our new logic, which we call $\mathcal{ALCSCC}^{++}$ to indicate that it is an extension of the logic $\mathcal{ALCSCC}$ introduced in [1]. When defining the semantics of $\mathcal{ALCSCC}^{++}$, we restrict the attention to *finite* interpretations to ensure that cardinalities of concept descriptions are always well-defined non-negative integers.

Definition 1 ($\mathcal{ALCSCC}^{++}$)**.** *Given disjoint* finite *sets N_C and N_R of* concept names *and* role names*, respectively, $\mathcal{ALCSCC}^{++}$* concept descriptions *(short:* concepts*) are inductively defined as follows:*

- *Every concept name $A \in N_C$ is an $\mathcal{ALCSCC}^{++}$ concept.*
- *If C, D are $\mathcal{ALCSCC}^{++}$ concepts, then so are $C \sqcap D$ (*conjunction*), $C \sqcup D$ (*disjunction*), and $\neg C$ (*negation*).*

[3] The definition of QFBAPA in [17] also allows for integer variables, which we do not use when integrating QFBAPA into our DL.

- *If Con is a set constraint or a cardinality constraint that uses role names and already defined* $\mathcal{ALCSCC}^{++}$ *concepts in place of set variables, then sat(Con) is an* $\mathcal{ALCSCC}^{++}$ *concept. We call sat(Con) a* constraint expression.

As usual, we use $\top$ *(*top*) and* $\bot$ *(*bottom*) as abbreviations for* $A \sqcup \neg A$ *and* $A \sqcap \neg A$*, respectively, where* A *is an arbitrary concept name.*

A finite interpretation *of* N_C *and* N_R *consists of a finite, non-empty set* $\Delta^\mathcal{I}$ *and a mapping* $\cdot^\mathcal{I}$ *that maps every concept name* $A \in N_C$ *to a subset* $A^\mathcal{I}$ *of* $\Delta^\mathcal{I}$ *and every role name* $r \in N_R$ *to a binary relation* $r^\mathcal{I}$ *over* $\Delta^\mathcal{I}$*. For a given element* $d \in \Delta^\mathcal{I}$ *we define*

$$r^\mathcal{I}(d) := \{e \in \Delta^\mathcal{I} \mid (d, e) \in r^\mathcal{I}\}.$$

The substitution $\sigma_d^\mathcal{I}$ *assigns the finite set* $\Delta^\mathcal{I}$ *to* $\mathcal{U}$*, the empty set to* $\emptyset$*, and the sets* $r^\mathcal{I}(d)$ *to* r *and* $A^\mathcal{I}$ *to* A*, where* $r \in N_R$ *and* $A \in N_C$ *are viewed as set variables.*

The interpretation function $\cdot^\mathcal{I}$ *and the substitutions* $\sigma_d^\mathcal{I}$ *for* $d \in \Delta^\mathcal{I}$ *are inductively extended to* $\mathcal{ALCSCC}^{++}$ *concepts by interpreting the Boolean operators as usual:*

- $\sigma_d^\mathcal{I}(C \sqcap D) = (C \sqcap D)^\mathcal{I} = C^\mathcal{I} \cap D^\mathcal{I}$,
- $\sigma_d^\mathcal{I}(C \sqcup D) = (C \sqcup D)^\mathcal{I} = C^\mathcal{I} \cup D^\mathcal{I}$,
- $\sigma_d^\mathcal{I}(\neg C) = (\neg C)^\mathcal{I} = \Delta^\mathcal{I} \setminus C^\mathcal{I}$,

and the constraint expressions as follows:

- $\sigma_d^\mathcal{I}(sat(Con)) = sat(Con)^\mathcal{I} = \{d \in \Delta^\mathcal{I} \mid$ *the substitution* $\sigma_d^\mathcal{I}$ *satisfies Con*$\}$.[4]

The $\mathcal{ALCSCC}^{++}$ *concept description* C *is* satisfiable *if there is a finite interpretation* $\mathcal{I}$ *such that* $C^\mathcal{I} \neq \emptyset$*.*

Note that the interpretation of concepts as set variables in $\mathcal{ALCSCC}^{++}$ is global in the sense that it does not depend on d, i.e., $\sigma_d^\mathcal{I}(C) = C^\mathcal{I} = \sigma_e^\mathcal{I}(C)$ for all $d, e \in \Delta^\mathcal{I}$. In contrast, the interpretation of role names r as set variables is local since only the r-successors of d are considered by $\sigma_d^\mathcal{I}(r)$. In $\mathcal{ALCSCC}$, also the interpretation of concepts as set variables is local since in the semantics of $\mathcal{ALCSCC}$ the substitution $\sigma_d^\mathcal{I}$ considers only the elements of $C^\mathcal{I}$ that are role successors of d for some role name in N_R (see [1]). To reflect this difference in the semantics also on the syntactic level, we use the keyword *succ* (for *succ*sessor) in place of *sat* for constraint expressions in $\mathcal{ALCSCC}$, and call these expressions *successor expressions*. For the sake of completeness, we now give a detailed definition of the DL $\mathcal{ALCSCC}$ as well as of $\mathcal{ALCSCC}$ TBoxes, ABoxes, and ECBoxes (see also [1, 8] and [2, 3]).

Definition 2 ($\mathcal{ALCSCC}$)**.** *Given disjoint* finite *sets* N_C *and* N_R *of* concept names *and* role names*, respectively,* $\mathcal{ALCSCC}$ concept descriptions *(short:* concepts*) are inductively defined as follows:*

- *Every concept name* $A \in N_C$ *is an* $\mathcal{ALCSCC}$ *concept.*
- *If* C, D *are* $\mathcal{ALCSCC}$ *concepts, then so are* $C \sqcap D$ *(*conjunction*),* $C \sqcup D$ *(*disjunction*), and* $\neg C$ *(*negation*).*
- *If Con is a set constraint or a cardinality constraint that uses role names and already defined* $\mathcal{ALCSCC}$ *concepts in place of set variables, then succ(Con) is an* $\mathcal{ALCSCC}$ *concept. We call succ(Con) a* successor expression.

[4] Note that, by induction, we can assume that $\sigma_d^\mathcal{I}$ is defined on the set variables (i.e., role names and concepts) occurring in *Con*.

An $\mathcal{ALCSCC}$ concept inclusion (CI) *is of the form* $C \sqsubseteq D$ *where* C, D *are* $\mathcal{ALCSCC}$ *concepts, and an* $\mathcal{ALCSCC}$ TBox *is a finite set of* $\mathcal{ALCSCC}$ *CIs. An* $\mathcal{ALCSCC}$ ABox *is a finite set of* concept assertions $C(a)$ *and* role assertions $r(a, b)$ *where* C *is an* $\mathcal{ALCSCC}$ *concept,* r *is a role name, and* a, b *are individual names from a set* N_I *of such names, which is disjoint with* N_C *and* N_R*. We define* extended cardinality constraints on $\mathcal{ALCSCC}$ concepts *as follows:*

- $\mathcal{ALCSCC}$ cardinality terms *are built from integer constants and concept cardinalities* $|C|$ *for* $\mathcal{ALCSCC}$ *concepts* C *using addition and multiplication with integer constants;*
- extended $\mathcal{ALCSCC}$ cardinality constraints *are of the form* $k = \ell, k < \ell, N\ \mathsf{dvd}\ \ell$*, where* k, ℓ *are* $\mathcal{ALCSCC}$ *cardinality terms and* N *is an integer constant;*
- *an* extended $\mathcal{ALCSCC}$ cardinality box (ECBox) *is a Boolean combination of extended* $\mathcal{ALCSCC}$ *cardinality constraints.*

A finite interpretation *of* N_C *and* N_R *consists of a finite, non-empty set* $\Delta^\mathcal{I}$ *and a mapping* $\cdot^\mathcal{I}$ *that maps every concept name* $A \in N_C$ *to a subset* $A^\mathcal{I}$ *of* $\Delta^\mathcal{I}$*, every role name* $r \in N_R$ *to a binary relation* $r^\mathcal{I}$ *over* $\Delta^\mathcal{I}$*, and ever individual name* $a \in N_I$ *to an element* $a^\mathcal{I}$ *of* $\Delta^\mathcal{I}$*. For a given element* $d \in \Delta^\mathcal{I}$ *we define*

$$r^\mathcal{I}(d) := \{e \in \Delta^\mathcal{I} \mid (d, e) \in r^\mathcal{I}\} \quad \textit{and} \quad ars^\mathcal{I}(d) := \bigcup_{r \in N_R} r^\mathcal{I}(d).$$ [5]

The substitution $\tau_d^\mathcal{I}$ *assigns the finite set* $ars^\mathcal{I}(d)$ *to* $\mathcal{U}$*, the empty set to* $\emptyset$*, and the sets* $r^\mathcal{I}(d)$ *to* r *and* $A^\mathcal{I} \cap ars^\mathcal{I}(d)$ *to* A*, where* $r \in N_R$ *and* $A \in N_C$ *are viewed as set variables.*

The interpretation function $\cdot^\mathcal{I}$ *and the substitutions* $\tau_d^\mathcal{I}$ *for* $d \in \Delta^\mathcal{I}$ *are inductively extended to* $\mathcal{ALCSCC}$ *concepts by interpreting the Boolean operators as usual:*

- $(C \sqcap D)^\mathcal{I} = C^\mathcal{I} \cap D^\mathcal{I}$ *and* $\tau_d^\mathcal{I}(C \sqcap D) = C^\mathcal{I} \cap D^\mathcal{I} \cap ars^\mathcal{I}(d)$.
- $(C \sqcup D)^\mathcal{I} = C^\mathcal{I} \cup D^\mathcal{I}$ *and* $\tau_d^\mathcal{I}(C \sqcup D) = (C^\mathcal{I} \cup D^\mathcal{I}) \cap ars^\mathcal{I}(d)$.
- $(\neg C)^\mathcal{I} = \Delta^\mathcal{I} \setminus C^\mathcal{I}$ *and* $\tau_d^\mathcal{I}(\neg C) = (\Delta^\mathcal{I} \setminus C^\mathcal{I}) \cap ars^\mathcal{I}(d)$.

and the successor expressions as follows:

- $succ(Con)^\mathcal{I} = \{d \in \Delta^\mathcal{I} \mid \text{the substitution } \tau_d^\mathcal{I} \text{ satisfies } Con\}$ *and*
- $\tau_d^\mathcal{I}(succ(Con)) = succ(Con)^\mathcal{I} \cap ars^\mathcal{I}(d)$.

The finite interpretation $\mathcal{I}$ *is a* model *of the* $\mathcal{ALCSCC}$ *TBox* $\mathcal{T}$ *if it satisfies all the CIs* $C \sqsubseteq D$ *in* $\mathcal{T}$*, which is the case if* $C^\mathcal{I} \subseteq D^\mathcal{I}$ *holds. It is a* model *of the* $\mathcal{ALCSCC}$ *ABox* $\mathcal{A}$ *if it satisfies all the assertions in* $\mathcal{A}$*, where* $\mathcal{I}$ *satisfies the concept assertion* $C(a)$ *if* $a^\mathcal{I} \in C^\mathcal{I}$ *holds, and the role assertion* $r(a, b)$ *if* $(a^\mathcal{I}, b^\mathcal{I}) \in r^\mathcal{I}$ *holds. Concept cardinalities within an ECBox* $\mathcal{E}$ *are interpreted in the obvious way, i.e.,* $|C|^\mathcal{I} := |C^\mathcal{I}|$*. Cardinality terms and cardinality constraints as well as their Boolean combination are then interpreted as in QFBAPA. The finite interpretation* $\mathcal{I}$ *is a* model *of an ECBox* $\mathcal{E}$ *if it satisfies the Boolean formula* $\mathcal{E}$ *according to this semantics.*

The $\mathcal{ALCSCC}$ *concept description* C *is* satisfiable *w.r.t. the ECBox* $\mathcal{E}$ *if there is a model* $\mathcal{I}$ *of* $\mathcal{E}$ *such that* $C^\mathcal{I} \neq \emptyset$*. The ABox* $\mathcal{A}$ *is* consistent *w.r.t.* $\mathcal{E}$ *if there is a model* $\mathcal{I}$ *of* $\mathcal{E}$ *that is also a model of* $\mathcal{A}$*.*

The following examples illustrates the difference between the semantics of constraint expressions in $\mathcal{ALCSCC}^{++}$ and successor expressions in $\mathcal{ALCSCC}$.

[5] *ars* stands for "all role successors."

Example 3. If A is a concept name and r is a role name, then the following is an $\mathcal{ALCSCC}^{++}$ concept description:

$$E := sat(|A| \geq 4) \sqcap sat(A \subseteq r) \sqcap sat(|r| \leq 3).$$

The first constraint expression requires that the overall size of the concept A is at least four. Thus, if $\mathcal{I}$ is an interpretation with $|A^{\mathcal{I}}| \leq 3$, then no element of $\Delta^{\mathcal{I}}$ can belong to $sat(|A| \geq 4)^{\mathcal{I}}$. Otherwise, every element of $\Delta^{\mathcal{I}}$ belongs to $sat(|A| \geq 4)^{\mathcal{I}}$. The second constraint says that every element of A must be an r successor of the given individual. Thus, $sat(A \subseteq r)^{\mathcal{I}}$ consists of those elements of $\Delta^{\mathcal{I}}$ that are connected, via the role r, with every element of $A^{\mathcal{I}}$. The third constraint is satisfied by those element of $\Delta^{\mathcal{I}}$ that have at most three r successors. Thus, the third and the second constraint put together require that $A^{\mathcal{I}}$ has at most three elements, which contradicts the first constraint. Thus, we have seen that the concept E is actually unsatisfiable.

Using the syntax for $\mathcal{ALCSCC}$ introduced in [1], we can write the following $\mathcal{ALCSCC}$ concept description

$$E' := succ(A \subseteq r) \sqcap succ(|r| \leq 3),$$

and state the global constraint $|A| \geq 4$ in an ECBox. But now we have that E' is satisfiable w.r.t. this ECBox since the constraints in E' are local. In fact, the first constraint in E' is satisfied by individuals for which every role successor that belongs to A is also an r successors of this individual. Together with the second constraint, this only implies that an individual that belongs to E' has at most three role successors belonging to A, but this does not constrain the overall number of elements of A, and thus does not contradict the statement in the ECBox, which is global. For example, an interpretation $\mathcal{I}$ consisting of four individuals belonging to A, none of which has any role successors, is a model of the global constraint $|A| \geq 4$, and every of its elements belongs to E'. In contrast, none of the individuals in $\mathcal{I}$ belongs to the $\mathcal{ALCSCC}^{++}$ concept E since the second constraint of E is clearly violated.

The local successor constraints of $\mathcal{ALCSCC}$ can clearly be simulated in $\mathcal{ALCSCC}^{++}$ by using $C \cap (\bigcup_{r \in N_R} r)$ instead of C when formulating the constraints. Thus, $\mathcal{ALCSCC}$ concepts can be expressed by $\mathcal{ALCSCC}^{++}$ concepts. In addition, extended cardinality constraints (ECBoxes), as introduced above, are expressible within $\mathcal{ALCSCC}^{++}$ concept descriptions, as are nominals, the universal role, and role negation. Recall that a *nominal* is of the form $\{a\}$ where $a \in N_I$, and is interpreted as the singleton set $\{a^{\mathcal{I}}\}$ by any finite interpretation $\mathcal{I}$. The *universal role* u is interpreted as $u^{\mathcal{I}} = \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, *role conjunction* as $(r \sqcap s)^{\mathcal{I}} = r^{\mathcal{I}} \cap s^{\mathcal{I}}$, and *role negation* as $(\neg r)^{\mathcal{I}} = (\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}) \setminus r^{\mathcal{I}}$.

Proposition 4. *$\mathcal{ALCSCC}^{++}$ concepts can polynomially express nominals, role conjunctions, and $\mathcal{ALCSCC}$ ECBoxes, and thus also ABoxes, $\mathcal{ALC}$ ECBoxes and $\mathcal{ALCSCC}$ TBoxes. In addition, they have the same expressivity as concepts of $\mathcal{ALCSCC}$ extended with the universal role or with role negation, whereas both of these features are not expressible in plain $\mathcal{ALCSCC}$.*

Proof. ECBoxes correspond to Boolean combinations of concepts of the form $sat(Con)$ where Con contains only concept descriptions as set variables. Since the concepts occurring in Con are interpreted globally when viewed as set variables, such a constraint expression $sat(Con)$ is satisfied either by no element of $\Delta^{\mathcal{I}}$ or by all of them. Consequently, their effect is to enforce the constraint on the whole interpretation domain if they are conjoined to a concept description.

Nominals are concepts that must be interpreted as singleton sets. Given a concept name A, we can enforce that it is interpreted as a singleton set using the constraint expression $sat(|A| = 1)$. Regarding role conjunction, the constraint $sat(\top \subseteq sat(t = r \cap s))$ ensures that, for every individual d, its t successors are exactly the individuals that are both its r and s successors.

The constraint $sat(\top \subseteq sat(u = \mathcal{U}))$ ensures that u is the universal role since it says that the u-successors of every individual are all the elements of the interpretation domain. Conversely, if the universal role is available, then every individual has all individuals as a role successors, and thus the difference between the semantics of $\mathcal{ALCSCC}$ and $\mathcal{ALCSCC}^{++}$ goes away.

Regarding role negation, for given role names $r, \overline{r}$, the constraint $sat(\top \subseteq sat(r \cap \overline{r} \subseteq \emptyset))$ enforces that, for every individual, the sets of its r and $\overline{r}$ successors are disjoint. In addition, the constraint $sat(\top \subseteq sat(|r| + |\overline{r}| = |\mathcal{U}|))$ says that elements of the domain that are not r successors of a given individual must be $\overline{r}$ successors. Thus, we can express in $\mathcal{ALCSCC}^{++}$ that the role $\overline{r}$ is interpreted as the complement of r, i.e. $\overline{r}^{\mathcal{I}} = \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \setminus r^{\mathcal{I}}$ for every finite interpretation $\mathcal{I}$. Conversely, role negation allows us to express the universal role in $\mathcal{ALCSCC}$: the $\mathcal{ALCSCC}$ constraint $sat(r \cup \neg r = u)$ is satisfied by an individual d if the set of its u successors consists of it r and its $\neg r$ successors, and thus all elements of the interpretation domain. Thus, conjoining such constraint at every place where u is used ensures that u really acts as the universal role.

Inexpressibility of role negation and of the universal role in $\mathcal{ALCSCC}$ can easily be shown using the fact that models of $\mathcal{ALCSCC}$ TBoxes are closed under disjoint union of finite interpretations, whereas this is not the case in the presence of role negation or the universal role. □

3 Satisfiability of $\mathcal{ALCSCC}^{++}$ concept descriptions

In the following we consider an $\mathcal{ALCSCC}^{++}$ concept description E and show how to test E for satisfiability by reducing this problem to the problem of testing satisfiability of QFBAPA formulae. Since the reduction is exponential and satisfiability in QFBAPA is in NP, this yields a NExpTime upper bound for satisfiability of $\mathcal{ALCSCC}^{++}$ concept descriptions. This bound is optimal since consistency of extended cardinality constraints in $\mathcal{ALC}$, as introduced in [8], is already NExpTime hard, and can be expressed as an $\mathcal{ALCSCC}^{++}$ satisfiability problem by Proposition 4.

Our NExpTime algorithm combines ideas from the satisfiability algorithm for $\mathcal{ALCSCC}$ concept descriptions [1] and the consistency procedure for $\mathcal{ALC}$ ECBoxes [8]. In particular, we use the notion of a type, as introduced in [8]. This notion is also similar to the Venn regions employed in [1]. Given a set of concept descriptions $\mathcal{M}$, the *type* of an individual in an interpretation consists of the elements of $\mathcal{M}$ to which the individual belongs. Such a type t can also be seen as a concept description C_t, which is the conjunction of all the elements of t. We assume in the following that E is an arbitrary, but fixed $\mathcal{ALCSCC}^{++}$ concept and $\mathcal{M}_E$ consists of *all subdescriptions* of the concept description E as well as the *negations of these subdescriptions*. In Example 3, the set $\mathcal{M}_E$ consists of

$$E, \neg E, sat(|A| \geq 4), \neg sat(|A| \geq 4), sat(A \subseteq r), \neg sat(A \subseteq r), sat(|r| \leq 3), \neg sat(|r| \leq 3), A, \neg A.$$

Definition 5. *A subset t of $\mathcal{M}_E$ is a* type *for E if it satisfies the following properties:*

1. *for every concept description $\neg C \in \mathcal{M}_E$, either C or $\neg C$ belongs to t;*
2. *for every concept description $C \sqcap D \in \mathcal{M}_E$, we have that $C \sqcap D \in t$ iff $C \in t$ and $D \in t$;*
3. *for every concept description $C \sqcup D \in \mathcal{M}_E$, we have that $C \sqcup D \in t$ iff $C \in t$ or $D \in t$.*

We denote the set of all types for E with types(E). Given an interpretation $\mathcal{I}$ and a domain element $d \in \Delta^{\mathcal{I}}$, the type of d w.r.t. E *is the set* $t_{\mathcal{I}}^{E}(d) := \{C \in \mathcal{M}_E \mid d \in C^{\mathcal{I}}\}$.

It is easy to show that the type of an individual really satisfies the conditions stated in the definition of a type. In our example, the following are the only types containing E:

$$t_1 \;:=\; \{E, sat(|A| \geq 4), sat(A \subseteq r), sat(|r| \leq 3), A\}, \tag{1}$$
$$t_2 \;:=\; \{E, sat(|A| \geq 4), sat(A \subseteq r), sat(|r| \leq 3), \neg A\}. \tag{2}$$

Due to Condition (1) in the definition of types, concept descriptions $C_t, C_{t'}$ induced by different types $t \neq t'$ are disjoint, and all concept descriptions in $\mathcal{M}_E$ can be obtained as the union of the concept descriptions induced by the types containing them, i.e., we have

$$C^{\mathcal{I}} = \bigcup_{t \text{ type with } C \in t} C_t^{\mathcal{I}}$$

for all $C \in \mathcal{M}_E$ and finite interpretations $\mathcal{I}$. Since the concepts induced by types are disjoint, the following holds for all finite interpretations $\mathcal{I}$:

$$|C^{\mathcal{I}}| = \sum_{t \text{ type with } C \in t} |C_t^{\mathcal{I}}| \quad \text{and} \quad |C_t^{\mathcal{I}}| = |\bigcap_{C \in t} C^{\mathcal{I}}|,$$

where the latter identity is an immediate consequence of the definition of C_t as the conjunction of all the elements of t. In our example, we have $|E^{\mathcal{I}}| = |C_{t_1}^{\mathcal{I}}| + |C_{t_2}^{\mathcal{I}}|$.

Given a type t, the constraints occurring in the top-level Boolean structure of t induce a QFBAPA formula ψ_t, in which the concepts C and roles r occurring in these constraints are replaced by set variables X_C and X_r^t, respectively. In our example, t_1 and t_2 contain the same constraints, and the associated QFBAPA formulae are clearly unsatisfiable:

$$\psi_{t_i} = |X_A| \geq 4 \wedge X_A \subseteq X_r^{t_i} \wedge |X_r^{t_i}| \leq 3 \text{ for } i = 1, 2.$$

Note that set variables corresponding to concepts are independent of the type t, i.e., they are shared by all types, whereas the set variables corresponding to roles are different for different types. This corresponds to the fact that roles are evaluated locally, but concepts are evaluated globally in the semantics of $\mathcal{ALCSCC}^{++}$. In order to ensure that the Boolean structure of concepts is respected by the set variables, we introduce the formula

$$\beta = \bigwedge_{C \sqcap D \in \mathcal{M}_E} X_{C \sqcap D} = X_C \cap X_D \wedge \bigwedge_{C \sqcup D \in \mathcal{M}_E} X_{C \sqcup D} = X_C \cup X_D \wedge \bigwedge_{\neg C \in \mathcal{M}_E} X_{\neg C} = (X_C)^c.$$

Overall, we translate the $\mathcal{ALCSCC}^{++}$ concept E into the QFBAPA formula

$$\delta_E := (|X_E| \geq 1) \wedge \beta \wedge \bigwedge_{t \in \text{types}(E)} (|\bigcap_{C \in t} X_C| = 0) \vee \psi_t.$$

Intuitively, to satisfy E, we need to have at least one element in it, which explains the first conjunct. The third conjunct together with β ensures that, for any type that is realized (i.e., has elements), the constraints of this type are satisfied.

In our example, β ensures that $X_E = \bigcap_{C \in t_1} X_C \cup \bigcap_{C \in t_2} X_C$ is satisfied. Together with $|X_E| \geq 1$ this implies that there is an $i \in \{1, 2\}$ such that $|\bigcap_{C \in t_i} X_C| > 0$ must hold. But then we need to satisfy ψ_{t_i}, which is impossible since this QFBAPA formula is unsatisfiable. Thus, we have seen that δ_E is not solvable, which corresponds to the fact E that is unsatisfiable.

The following two lemmas state that solvability of δ_E and satisfiability of E are indeed equivalent.

Lemma 6. *If the $\mathcal{ALCSCC}^{++}$ concept description E is satisfiable, then the QFBAPA formula δ_E is also satisfiable.*

Proof. Assume that the finite interpretation $\mathcal{I}$ satisfies E, i.e., there is a $d_0 \in \Delta^{\mathcal{I}}$ such that $d_0 \in E^{\mathcal{I}}$. We define $\sigma(X_C) := C^{\mathcal{I}}$ for all concepts $C \in \mathcal{M}_E$. Then we have $d_0 \in \sigma(X_E)$, and thus σ satisfies the cardinality constraint $|X_E| \geq 1$. In addition, σ clearly satisfies β. For example, $\sigma(X_{C \sqcap D}) = (C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}} = \sigma(X_C) \cap \sigma(X_D) = \sigma(X_C \cap X_D)$. For every type t we have $C_t^{\mathcal{I}} = \bigcap_{C \in t} C^{\mathcal{I}} = \bigcap_{C \in t} \sigma(X_C) = \sigma(\bigcap_{C \in t} X_C)$, and thus $\sigma(|\bigcap_{C \in t} X_C|) = 0$ iff $C_t^{\mathcal{I}} = \emptyset$.

Let t by a type such that $\sigma(|\bigcap_{C \in t} X_C|) \neq 0$. Then there is an individual $d \in \Delta^{\mathcal{I}}$ such that $d \in C_t^{\mathcal{I}}$. The semantics of $\mathcal{ALCSCC}^{++}$ then implies that we can extend σ to a solution of ψ_t by interpreting the set variables with superscript t using the role successors of d:

$$\sigma(X_r^t) := \{e \mid (d, e) \in r^{\mathcal{I}}\}.$$

If t is a type such that $\sigma(|\bigcap_{C \in t} X_C|) = 0$, then it is not necessary for σ to satisfy ψ_t. We can thus extend σ to the set variables with superscript t in an arbitrary way, e.g. by interpreting all of them as the empty set. Overall, this show that we can use an interpretation satisfying E to define a solution σ of δ_E. □

Next, we show that the converse of Lemma 6 holds as well.

Lemma 7. *If the QFBAPA formula δ_E is satisfiable, then the $\mathcal{ALCSCC}^{++}$ concept description E is also satisfiable.*

Proof. Assume that there is a solution σ of δ_E. We claim that, for every element $e \in \sigma(\mathcal{U})$, there is a unique type t_e such that $e \in \bigcap_{C \in t_e} \sigma(X_C)$. In fact, we can define t_e as

$$t_e := \{C \in \mathcal{M}_E \mid e \in \sigma(X_C)\}.$$

Since σ satisfies β, the set t_e is indeed a type. For example, assume that $C \sqcup D \in t_e$. Then $e \in \sigma(X_{C \sqcup D}) = \sigma(X_C) \cup \sigma(X_D)$ iff $e \in \sigma(X_C)$ or $e \in \sigma(X_D)$ iff $C \in t_e$ or $D \in t_e$. Satisfaction of the other conditions in the definition of a type can be shown similarly. Regarding uniqueness, assume that t is a type different from t_e. Then there is an element $C \in \mathcal{M}_E$ such that (modulo removal of double negation) $C \in t_e$ and $\neg C \in t$. But then $e \in \sigma(X_C)$ implies $e \notin \sigma((X_C)^c) = \sigma(X_{\neg C})$, and thus $e \notin \bigcap_{D \in t} \sigma(X_D)$.

Let

$$T_\sigma := \{t \mid t \text{ type with } \sigma(|\bigcap_{C \in t} X_C|) \neq 0\}$$

be the set of all types that are realized by σ. Note that, by what we have shown above, we have $T_\sigma = \{t_e \mid e \in \sigma(\mathcal{U})\}$.

We now define a finite interpretation $\mathcal{I}$ and show that it satisfies E. The interpretation domain consists of copies of the realized types, where the number of copies is determined by σ:

$$\Delta^{\mathcal{I}} := \{(t, j) \mid t \in T_\sigma \text{ and } 1 \leq j \leq \sigma(|\bigcap_{C \in t} X_C|)\}.$$

Since for every element $e \in \sigma(\mathcal{U})$ there is a unique type t_e such that $e \in \bigcap_{C \in t_e} \sigma(X_C)$, there is a bijection π from $\sigma(\mathcal{U})$ to $\Delta^{\mathcal{I}}$ such that $\pi(e) = (t, j)$ implies that $t = t_e$.

For concept names A we define

$$A^{\mathcal{I}} := \{(t, j) \in \Delta^{\mathcal{I}} \mid A \in t\}$$

and for role names r

$$r^{\mathcal{I}} := \{((t,j),\pi(e)) \mid (t,j) \in \Delta^{\mathcal{I}} \wedge e \in \sigma(X_r^t)\}.$$

Since σ solves the constraint $X_E \geq 1$, there is a $d_0 \in \sigma(X_E)$. Let t_0 be the unique type such that $d_0 \in \bigcap_{C\in t_0} \sigma(X_C)$. Then we have $\sigma(|\bigcap_{C\in t_0} X_C|) \neq 0$, and thus $(t_0, 1) \in \Delta^{\mathcal{I}}$. To show that $\mathcal{I}$ satisfies E, it is sufficient to show that $(t_0, 1) \in E^{\mathcal{I}}$.

For this, we show the following more general claim: for all concept descriptions $C \in \mathcal{M}_E$ and all $(t,j) \in \Delta^{\mathcal{I}}$ we have

$$(t,j) \in C^{\mathcal{I}} \text{ iff } C \in t. \tag{3}$$

We show (3) by *induction on the structure of* C:

- Let $C = A$ for $A \in N_C$. Then (3) is an immediate consequence of the definition of $A^{\mathcal{I}}$ for concept names A.
- Let $C = \neg D$. Then induction yields $(t,j) \in D^{\mathcal{I}}$ iff $D \in t$. By contraposition, this is the same as $(t,j) \notin D^{\mathcal{I}}$ iff $D \notin t$. By Condition 1 in the definition of types and the semantics of negation, this is in turn equivalent to $(t,j) \in (\neg D)^{\mathcal{I}}$ iff $\neg D \in t$.
- Let $C = D_1 \sqcap D_2$. Then induction yields $(t,j) \in D_1^{\mathcal{I}}$ iff $D_1 \in t$ and $(t,j) \in D_2^{\mathcal{I}}$ iff $D_2 \in t$. From this, we obtain $(t,j) \in (D_1 \sqcap D_2)^{\mathcal{I}}$ iff $D_1 \sqcap D_2 \in t$ using Condition 2 in the definition of types and the semantics of conjunction.
- The case where $C = D_1 \sqcup D_2$ can be handled similarly, using Condition 3 in the definition of types and the semantics of disjunction.
- $C = sat(Con)$ be a constraint expression. First, assume that $C \in t$. Then the translation Con' of Con using set variables X_D and X_r^t is a conjunct in ψ_t. In addition, since $(t,j) \in \Delta^{\mathcal{I}}$, we have $\sigma(|\bigcap_{D\in t} X_D|) \neq 0$. Consequently, σ satisfies this translation Con'. Thus, to show that $(t,j) \in C^{\mathcal{I}}$, it is sufficient to show that the following holds:

  1. $\pi(\sigma(X_r^t)) = r^{\mathcal{I}}(t,j)$ and
  2. $\pi(\sigma(X_D)) = D^{\mathcal{I}}$ for all concepts D occurring in the constraint c.

  The first statement is an immediate consequence of the definition of the interpretation of the roles in $\mathcal{I}$.

  To show the second statement, first assume that $e \in \sigma(X_D)$. Then $\pi(e) = (t_e, j')$ where t_e is the unique type such that $e \in \bigcap_{F\in t_e} \sigma(X_F)$. Thus, $e \in \sigma(X_D)$ implies that $D \in t_e$. By induction, we obtain $\pi(e) = (t_e, j') \in D^{\mathcal{I}}$. Second, assume that $\pi(e) = (t_e, j') \in D^{\mathcal{I}}$. Then induction yields $D \in t_e$, and thus $e \in \sigma(X_D)$.

  Conversely, assume that $C \notin t$. Then $\neg\,\mathrm{succ}(Con) \in t$, and thus the translation $\neg Con'$ of $\neg Con$ using set variables X_D and X_r^t is a conjunct in ψ_t. We can now proceed as in the first case, but with $\neg Con$ and $\neg Con'$ in place of Con and Con'.

This completes the proof of (3) and thus the proof of the lemma. □

We have shown that the question of whether an $\mathcal{ALCSCC}^{++}$ concept description E is satisfiable can be reduced to checking whether the corresponding QFBAPA formula δ_E is satisfiable. Since the size of δ_E is exponential in the size of E, this yields the following complexity result.

Theorem 8. *Satisfiability of $\mathcal{ALCSCC}^{++}$ concept descriptions is* NExpTime*-complete independently of whether the numbers occurring in these descriptions are encoded in unary or binary.*

Proof. Since satisfiability of QFBAPA formulae can be decided within NP even for binary coding of numbers [17], it is sufficient to show that the size of the QFBAPA formula δ_E is at most exponential in the size of E. This is an easy consequence of the fact that there are at most exponentially many types t since the cardinality of $\mathcal{M}_E$ is linear in the size of E. This implies that the conjunction over all types in δ_E has only exponentially many conjuncts. The conjunct for a type t is of the form $(|\bigcap_{C\in t} X_C| = 0) \vee \psi_t$. Since every type contains only linearly many concepts, and these concepts have linear size, both $(|\bigcap_{C\in t} X_C| = 0)$ and ψ_t is of polynomial size. Obviously, $(|X_E| \geq 1)$ has linear size, and the formula β has polynomial size since $\mathcal{M}_E$ contains linearly many elements of linear size.

The NExpTime lower bound is inherited from consistency of $\mathcal{ALC}$ ECBoxes [8] due to Proposition 4. As argued in [8], this lower bound already holds if numbers are encoded in unary since one can use small ECBoxes to generate large numbers from small ones. □

Thanks to Proposition 4, the NExpTime upper bound carries over to satisfiability of $\mathcal{ALCSCC}^{++}$ knowledge bases, which may feature an ABox, a TBox and an ECBox.

4 Restricted Cardinality Constraints and ABoxes in $\mathcal{ALCSCC}$

In Definition 2, we have introduced the DL $\mathcal{ALCSCC}$ and ECBoxes. As mentioned above, NExpTime hardness already holds for consistency of $\mathcal{ALCSCC}$ ECBoxes, and Theorem 8 yields the matching upper bound since ECBoxes can be expressed by $\mathcal{ALCSCC}^{++}$ concepts by Proposition 4. The same proposition also states that ABoxes can be expressed by $\mathcal{ALCSCC}^{++}$ concepts, which yields a NExpTime upper bound also for consistency of $\mathcal{ALCSCC}$ ABoxes w.r.t. $\mathcal{ALCSCC}$ ECBoxes.

For the sub-logic $\mathcal{ALC}$ of $\mathcal{ALCSCC}$, a restricted notion of cardinality boxes, called RCBoxes, was introduced in [8], and it was shown that this restriction lowers the complexity of the consistency problem from NExpTime to ExpTime. In [2, 3] it was shown that the same is true for $\mathcal{ALCSCC}$. Here we prove that this result can be extended to consistency of $\mathcal{ALCSCC}$ ABoxes w.r.t. $\mathcal{ALCSCC}$ RCBoxes. In the presence of ECBoxes, this extension is irrelevant since ECBoxes can express nominals, and thus also ABoxes. However, this is not the case for RCBoxes. Below, we actually consider an extension of RCBoxes, which were called ERCBoxes in [21].

Definition 9 (RCBoxes)**.** Semi-restricted $\mathcal{ALCSCC}$ cardinality constraints *are of the form*

$$N_1|C_1| + \cdots + N_k|C_k| + M \leq N_{k+1}|C_{k+1}| + \cdots + N_{k+\ell}|C_{k+\ell}|, \tag{4}$$

where C_i are $\mathcal{ALCSCC}$ concept descriptions, N_i are integer constants for $1 \leq i \leq k+\ell$, and M is a non-negative integer constant. An extended restricted $\mathcal{ALCSCC}$ cardinality box (ERCBox) *is a positive Boolean combination of semi-restricted $\mathcal{ALCSCC}$ cardinality constraints.*

An interpretation $\mathcal{I}$ is a model of the semi-restricted $\mathcal{ALCSCC}$ cardinality constraint (4) if

$$N_1|C_1^{\mathcal{I}}| + \cdots + N_k|C_k^{\mathcal{I}}| + M \leq N_{k+1}|C_{k+1}^{\mathcal{I}}| + \cdots + N_{k+\ell}|C_{k+\ell}^{\mathcal{I}}|.$$

The notion of a model is extended to ERCBoxes using the usual interpretation of conjunction and disjunction in propositional logic.

Note that $\mathcal{ALCSCC}$ ECBoxes can express both ERCBoxes and ABoxes. The restricted cardinality boxes (RCBoxes) introduced in [8, 2, 3] differ from ERCBoxes in that the number

M in constraints of the form (4) must be zero, and that only conjunction of such constraints is allowed. Since ExpTime-hardness already holds for consistency of RCBoxes in $\mathcal{ALCSCC}$ without an ABox [8, 2, 3], we obtain the following complexity lower bound. Actually, the hardness proof does not require large number, and thus ExpTime-hardness even holds for unary coding of numbers.

Proposition 10. *The consistency of $\mathcal{ALCSCC}$ ERCBoxes w.r.t. $\mathcal{ALCSCC}$ ABoxes is* ExpTime*-hard, independently of whether numbers are encoded in unary or binary.*

Following the approach in [2, 3] for consistency of $\mathcal{ALCSCC}$ RCBoxes, we show the ExpTime upper bound for numbers encoded in binary using type elimination, where the notion of augmented type from [1] is used, and a second step for removing types is added to take care of the ERCBox, similarly to what is done in [8]. In addition, the ABox individuals are taken into account by making them elements of exactly one augmented type.

The ExpTime upper bound for our procedure on the one hand depends on the following lemma, which applies in our setting due to the special form of semi-restricted cardinality constraints. It is an extension of Lemma 10 in [8].

Lemma 11. *Let ϕ be a system of linear inequalities consisting of $A \cdot \boldsymbol{v} \geq \boldsymbol{b}$ and $\boldsymbol{v} \geq \boldsymbol{0}$, where A, B are matrices of integer coefficients, $\boldsymbol{b}$ is a vector of non-negative integer parameters, and $\boldsymbol{v}$ is the variable vector.*

1. *The solutions of ϕ are closed under addition.*
2. *If $\{v_1, \ldots, v_k\}$ is a set of variables such that, for all v_i $(1 \leq i \leq k)$, ϕ has a solution $\boldsymbol{c}_i$ in which the ith component $c_i^{(i)}$ is not 0, then there is a non-negative integer solution $\boldsymbol{c}$ of ϕ such that, for all $i, 1 \leq i \leq k$, the ith component $c^{(i)}$ of $\boldsymbol{c}$ satisfies $c^{(i)} \geq 1$.*
3. *Deciding whether ϕ has a non-negative integer solution can be done in polynomial time.*

Proof. (1) Let $\boldsymbol{c}, \boldsymbol{d}$ be two solutions of ϕ. Since all components of these vectors are non-negative, this is clearly also the case for their sums. In addition, we have

$$A \cdot (\boldsymbol{c} + \boldsymbol{d}) = A \cdot \boldsymbol{c} + A \cdot \boldsymbol{d} \geq \boldsymbol{b} + \boldsymbol{b} \geq \boldsymbol{b},$$

where the first inequality holds since $\boldsymbol{c}, \boldsymbol{d}$ are solutions of ϕ, and the last inequality holds since the components of $\boldsymbol{b}$ are non-negative.

(2) Given solutions $\boldsymbol{c}_i$ as described in the second part of the lemma, the solution $\boldsymbol{c}$ satisfying the stated properties can be obtained as their sum.

(3) It is well-known that solvability in the rational numbers of a system of inequalities of the form stated in the lemma can be decided in polynomial time [13]. In addition, if ϕ has a rational solution $\boldsymbol{c}$, then it also has an integer solution. In fact, let D be the least common multiple (lcm) of the denominators of the components of $\boldsymbol{c}$. Then $D \cdot \boldsymbol{c}$ is an integer vector that is a solution of ϕ due to closure under addition of solutions, as stated in the first part of the lemma. □

Another important ingredient of our ExpTime procedure are augmented types, which have been introduced in [1] to show that satisfiability in $\mathcal{ALCSCC}$ w.r.t. concept inclusions is in ExpTime. We use the notion of a type as introduced in Definition 5 (see also Definition 3 of [2]), but extended such that it takes the ABox $\mathcal{A}$ and the ERCBox $\mathcal{R}$ into account, i.e., the set $\mathcal{M}(\mathcal{R}, \mathcal{A})$ of all relevant concept descriptions contains all subdescriptions of the concept descriptions occurring in $\mathcal{R}$ or $\mathcal{A}$ as well as their negations. In addition, for every individual name $b \in \mathsf{Ind}_\mathcal{A}$ (where $\mathsf{Ind}_\mathcal{A}$ denotes the set of individual name occurring on $\mathcal{A}$), the set $\mathcal{M}(\mathcal{R}, \mathcal{A})$ contains this name and its negation.

Definition 12. *Let $\mathcal{A}$ be an $\mathcal{ALCSCC}$ ABox and $\mathcal{R}$ be an $\mathcal{ALCSCC}$ ERCBox. A subset t of $\mathcal{M}(\mathcal{R},\mathcal{A})$ is a* type for $\mathcal{R}$ and $\mathcal{A}$ *if it satisfies the following properties:*

1. *for every concept description $\neg C \in \mathcal{M}(\mathcal{R},\mathcal{A})$, either C or $\neg C$ belongs to t;*
2. *for every individual name $b \in \mathcal{M}(\mathcal{R},\mathcal{A})$, either b or $\neg b$ belongs to t;*
3. *for every concept description $C \sqcap D \in \mathcal{M}(\mathcal{R},\mathcal{A})$, we have that $C \sqcap D \in t$ iff $C \in t$ and $D \in t$;*
4. *for every concept description $C \sqcup D \in \mathcal{M}(\mathcal{R},\mathcal{A})$, we have that $C \sqcup D \in t$ iff $C \in t$ or $D \in t$.*

Intuitively, a type containing $b \in \mathsf{Ind}_\mathcal{A}$ is supposed to represent the individual b. Our type elimination procedure will ensure that, for every individual b exactly one type is available. However, ERCBoxes do not allow us to express that this type should be realized by only one element of the model. In our model construction, we will actually have several individuals that realize such a type, and choose one of them to actually interpret the individual b. With respect to membership in concepts, this "chosen" individual and its copies behave the same. However, to satisfy role assertions we must ensure that role successors are always the chosen individuals. This can be achieved by adding an appropriate cardinality constraint when defining augmented types (see below).

Augmented types consider not just the concepts to which a single individual belongs, but also the Venn regions to which its role successors belong. Basically, we define the notion of a Venn region as in [1, 2, 3], but extend it by (i) always considering the set of all set variables X_D for subdescriptions D occurring in $\mathcal{R}$ or $\mathcal{A}$ and X_r for $r \in N_R$ rather than just the ones occurring in the given QFBAPA formula; and (ii) additionally considering set variables X_b for all individuals $b \in \mathsf{Ind}_\mathcal{A}$.

Definition 13 (Venn region)**.** *Let $\mathcal{A}$ be an $\mathcal{ALCSCC}$ ABox and $\mathcal{R}$ be an $\mathcal{ALCSCC}$ ERCBox, and let $X_1, \dots, X_k$ be an enumeration of all set variables X_C for subdescriptions C occurring in $\mathcal{R}$ or $\mathcal{A}$, X_r for $r \in N_R$, and X_a for individual names $a \in \mathsf{Ind}_\mathcal{A}$. A* Venn region for $\mathcal{R}$ and $\mathcal{A}$ *is of the form*

$$X_1^{c_1} \cap \ldots \cap X_k^{c_k},$$

where c_i is either empty or c for $i = 1, \dots, k$.

Again, a Venn region containing b says that this element corresponds to the individual $b \in \mathsf{Ind}_\mathcal{A}$. But now QFBAPA allows us to formulate constraints on the cardinality of the sets X_b. In particular, by adding $|X_b| \leq 1$ we can ensure that there is only one role successor that belongs to a type containing b.

Given a type t for $\mathcal{R}$ and $\mathcal{A}$, we consider the corresponding QFBAPA formula ϕ_t, which is induced by the (possibly negated) successor constraints occurring in t. We conjoin to this formula the set constraint

$$X_{r_1} \cup \ldots \cup X_{r_n} = \mathcal{U},$$

where $N_R = \{r_1, \dots, r_n\}$,[6] as well as the cardinality constraints

$$|X_b| \leq 1$$

for $b \in \mathsf{Ind}_\mathcal{A}$. In case $a \in \mathsf{Ind}_\mathcal{A}$ belongs to t, we consider all role assertions $r_1(a, b_1), \dots, r_k(a, b_\ell)$ with a in the first component in $\mathcal{A}$, and add the conjuncts

$$|X_{b_1} \cap X_{r_1}| \geq 1, \dots, |X_{b_\ell} \cap X_{r_\ell}| \geq 1.$$

For the resulting formula ϕ'_t, we compute the number N_t that bounds the number of Venn regions that need to be non-empty in a solution of ϕ'_t (see Lemma 1 in [2]).

[6] Without loss of generality we assume that N_R contains only the role names occurring in $\mathcal{R}$ and $\mathcal{A}$.

Definition 14. *Let $\mathcal{R}$ be an $\mathcal{ALCSCC}$ ERCBox and $\mathcal{A}$ be an $\mathcal{ALCSCC}$ ABox. An* augmented type (t, V) *for $\mathcal{R}$ and $\mathcal{A}$ consists of a type t for $\mathcal{R}$ and $\mathcal{A}$ together with a set of Venn region V such that $|V| \leq N_t$ and the formula ϕ'_t has a solution in which exactly the Venn regions in V are non-empty.*

The existence of a solution of ϕ'_t in which exactly the Venn regions in V are non-empty can obviously be checked (within NP) by adding to ϕ'_t conjuncts that state non-emptiness of the Venn regions in V and the fact that the union of these Venn regions is the universal set (see the description of the PSpace algorithm in the proof of Theorem 1 in [1]). Another easy to show observation is that there are only exponentially many augmented types (see the accompanying technical report of [1] for a proof of the following lemma).

Lemma 15. *Let $\mathcal{R}$ be an $\mathcal{ALCSCC}$ ERCBox and $\mathcal{A}$ be an $\mathcal{ALCSCC}$ ABox. The set of augmented types for $\mathcal{R}$ and $\mathcal{A}$ contains at most exponentially many elements in the size of $\mathcal{R}$ and $\mathcal{A}$, and it can be computed in exponential time.*

The type elimination procedure checking the consistency of $\mathcal{ALCSCC}$ RCBoxes introduced in [2] starts with the set of all augmented types, and then successively eliminates augmented types

(i) whose Venn regions are not realized by the currently available augmented types, or

(ii) whose first component is forced to be empty by the constraints in $\mathcal{R}$.

To make the first reason for elimination more precise, assume that $\mathbb{A}$ is a set of augmented types and that v is a Venn region. In the following, let D denote an $\mathcal{ALCSCC}$ concept and b an individual name. The Venn region v yields a set of concept descriptions S_v that contains, for every set variable X_D (X_b) occurring in v, the element D (b) in case v contains X_D (X_b) and the element $\neg D$ ($\neg b$) in case v contains X^c_D (X^c_b). It is easy to see that S_v is actually a subset of $\mathcal{M}(\mathcal{R}, \mathcal{A})$ (modulo removal of double negation).

Definition 16. *Let $\mathbb{A}$ be a set of augmented types and v a Venn region, We say that v is* realized by $\mathbb{A}$ *if there is an augmented type $(t, V) \in \mathbb{A}$ such that $S_v \subseteq t$.*

The fact that both Venn regions and types contain every concept or individual (set variable) either positively or negatively implies that, modulo elimination of double negation, we actually have $S_v = t$ whenever $S_v \subseteq t$. Note that, for some Venn regions v, there may not be a type t such that $S_v \subseteq t$ since in the definition of Venn regions we do not consider the Boolean structure of concepts (e.g., a Venn region may contain $X_{C \sqcap D}$ positively, but X_D negatively). However this will not be a problem since in our proofs we will always work with Venn regions that are contained in types.

Also note that the condition that Venn regions must be realized also takes care of role assertions. In fact, consider an augmented type (t, V) and assume that the type t contains a and $r(a, b) \in \mathcal{A}$. Then ϕ'_t contains the conjuncts $|X_b \cap X_r| \geq 1$ and $|X_b| \leq 1$. Consequently, V contains a Venn region v in which X_r and X_b occur positively, and thus $b \in S_v$. If this Venn region is realized by the augmented type (s, W), then s must contain b (i.e., represent the individual b). Intuitively, this ensures that a has r-successor b. In order to show this formally, however, some more work is needed since we must ensure that a is actually linked to the copy chosen to represent b rather than just to a type containing b (see the proof of Lemma 18 below).

We are now ready to formulate our algorithm. We assume without loss of generality that $\mathcal{A}$ is non-empty, and thus contains at least one individual. In addition, we assume that $\mathcal{R}$ is a conjunction of semi-restricted constraints, which we call a *conjunctive ERCBox*. We will argue later why is is sufficient to restrict the attention to conjunctive ERCBoxes.

Algorithm 17. *Let $\mathcal{R}$ be a conjunctive $\mathcal{ALCSCC}$ ERCBox and $\mathcal{A} \neq \emptyset$ be an $\mathcal{ALCSCC}$ ABox. First, we compute the set $\mathcal{M}(\mathcal{R}, \mathcal{A})$ consisting of all subdescriptions of $\mathcal{R}$ and $\mathcal{A}$ as well as the negations of these subdescriptions, together with the set of all individual names occurring in $\mathcal{A}$ and their negations. Based on this set $\mathcal{M}(\mathcal{R}, \mathcal{A})$, we compute the set $\widehat{\mathbb{A}}$ of all augmented types for $\mathcal{R}$ and $\mathcal{A}$. We now decide consistency of $\mathcal{A}$ w.r.t. $\mathcal{R}$ by performing the following three steps:*

1. *Compute all maximal subsets $\mathbb{A}$ of $\widehat{\mathbb{A}}$ such that*

   (a) *for every individual $b \in \mathsf{Ind}_{\mathcal{A}}$, there is exactly one augmented type $(t, V) \in \mathbb{A}$ with $b \in t$,*

   (b) *if $(t, V) \in \mathbb{A}$ and $b \in t$ for an individual $b \in \mathsf{Ind}_{\mathcal{A}}$, then $C \in t$ for all concept assertion $C(b) \in \mathcal{A}$,*

   To achieve this, in a first step, we can remove all augmented types that do not satisfy condition (1b). In case there is an individual $b \in \mathsf{Ind}_{\mathcal{A}}$ such that all types containing b have been removed, then the algorithm fails*. Otherwise, choose for every $b \in \mathsf{Ind}_{\mathcal{A}}$ exactly one of the remaining augmented types whose first component contains b and remove all the other augmented types containing b.*

   Check whether the following two steps succeed for one of the sets $\mathbb{A}$ computed this way.

2. *If there is an individual $b \in \mathsf{Ind}_{\mathcal{A}}$ such that $\mathbb{A}$ does not contain an augmented type (t, V) such that $b \in t$, then the algorithm* fails *for the current set of augmented types. Otherwise, it checks whether $\mathbb{A}$ contains an element (t, V) such that not all the Venn regions in V are realized by $\mathbb{A}$. If there is no such element (t, V) in $\mathbb{A}$, then continue with the next step. Otherwise, let (t, V) be such an element, and set $\mathbb{A} := \mathbb{A} \setminus \{(t, V)\}$. Continue with this step, but now using the new current set of augmented types.*

3. *Let $T_{\mathbb{A}} := \{t \mid \text{there is } V \text{ such that } (t, V) \in \mathbb{A}\}$, and let $\phi_{T_{\mathbb{A}}}$ be obtained from $\mathcal{R}$ by replacing each $|C|$ in $\mathcal{R}$ with $\sum_{t \in T_{\mathbb{A}} \text{ s.t. } C \in t} v_t$ and adding $v_t \geq 0$ for each $t \in T_{\mathbb{A}}$. Check whether $T_{\mathbb{A}}$ contains an element t such that $\phi_{T_{\mathbb{A}}} \wedge v_t \geq 1$ has no solution. If this is the case for t, then remove all augmented types of the form $(t, \cdot)$ from $\mathbb{A}$, and continue with the previous step. If no type t is removed in this step, then the algorithm* succeeds.

Before proving that this algorithm runs in exponential time, we show that it is sound and complete.

Lemma 18 (Soundness)**.** *Let $\mathcal{R}$ be a conjunctive $\mathcal{ALCSCC}$ ERCBox and $\mathcal{A} \neq \emptyset$ an $\mathcal{ALCSCC}$ ABox. If Algorithm 17 succeeds on input $\mathcal{R}$ and $\mathcal{A}$, then $\mathcal{A}$ is consistent w.r.t. $\mathcal{R}$.*

Proof. Assume that the algorithm succeeds on input $\mathcal{R}$ and $\mathcal{A}$, and let $\mathbb{A}$ be the final set of augmented types when the algorithm stops successfully. Note that $\mathbb{A} \neq \emptyset$ since there is at least one individual b in $\mathcal{A}$, and thus the algorithm would have failed for an empty set of augmented types. We show how $\mathbb{A}$ can be used to construct a model $\mathcal{I}$ of $\mathcal{R}$ and $\mathcal{A}$.

For this construction, we first consider the formula $\phi_{T_{\mathbb{A}}}$, which is obtained from $\mathcal{R}$ by replacing each $|C|$ in $\mathcal{R}$ with $\sum_{t \in T_{\mathbb{A}} \text{ s.t. } C \in t} v_t$ and adding $v_t \geq 0$ for each $t \in T_{\mathbb{A}}$. Note that, due to the special form of conjunctive ERCBoxes, we know that this yields a system of linear inequalities of the form $A \cdot \boldsymbol{v} \geq \boldsymbol{b}$, $\boldsymbol{v} \geq \boldsymbol{0}$. Since the algorithm has terminated successfully, we know for all $t \in T_{\mathbb{A}}$ that the formula $\phi_{T_{\mathbb{A}}} \wedge v_t \geq 1$ has a solution. By Lemma 11 this implies that $\phi_{T_{\mathbb{A}}}$ has a solution in which all variables v_t for $t \in T_{\mathbb{A}}$ have a value ≥ 1 and all variables v_t with $t \notin T_{\mathbb{A}}$ have value 0. In addition, given an arbitrary number $N \geq 1$, we know that there is a solution σ_N of $\phi_{T_{\mathbb{A}}}$ such that $\sigma_N(v_t) \geq 1$ and $N | \sigma_N(v_t)$ holds for all $t \in T_{\mathbb{A}}$. To see this, note that we can just multiply with N a given solution satisfying the properties mentioned in the previous sentence.

We use the augmented types in $\mathbb{A}$ to determine the right N:

- For each augmented type (t, V), we know that the formula ϕ'_t has a solution where exactly the Venn regions in V are non-empty (see Definition 14). Assume that this solution assigns a set of cardinality $k_{(t,V)}$ to the universal set.

- For each $t \in T_{\mathbb{A}}$, let n_t be the cardinality of the set $\{V \mid (t, V) \in \mathbb{A}\}$, i.e., the number of augmented types in $\mathbb{A}$ that have t as their first component.

We now define N as

$$N := (\max\{k_{(t,V)} \mid (t, V) \in \mathbb{A}\}) \cdot \prod_{t \in T_{\mathbb{A}}} n_t,$$

and use the solution σ_N of $\phi_{T_{\mathbb{A}}}$ to construct a finite interpretation $\mathcal{I}$ as follows. The domain of $\mathcal{I}$ is defined as

$$\Delta^{\mathcal{I}} := \{(t, V)^i \mid (t, V) \in \mathbb{A} \text{ and } 1 \leq i \leq \sigma_N(v_t)/n_t\}.$$

Note that $\sigma_N(v_t)/n_t$ is a natural number since $N|\sigma_N(v_t)$ implies $n_t|\sigma_N(v_t)$. In addition, $\Delta^{\mathcal{I}} \neq \emptyset$ because $\mathbb{A} \neq \emptyset$ and $\sigma_N(v_t)/n_t \geq 1$ since $\sigma_N(v_t) \geq 1$. Moreover, for each type $t \in T_{\mathbb{A}}$, the set $\{(t, V)^i \mid (t, V)^i \in \Delta^{\mathcal{I}}\}$ has cardinality $\sigma_N(v_t)$.

The interpretation of the concept names A is based on the occurrence of these names in the first component of an augmented type, i.e.,

$$A^{\mathcal{I}} := \{(t, V)^i \in \Delta^{\mathcal{I}} \mid A \in t\}.$$

Individual names are treated similarly, however we need to ensure that an individual name is interpreted by a single element of $\Delta^{\mathcal{I}}$, and not by a set of cardinality > 1. First, note that, due to step (1) and the failure condition in step (2), for each individual name $a \in \mathsf{Ind}_{\mathcal{A}}$, $\mathbb{A}$ contains exactly one augmented type (t, V) such that $a \in t$. Let us denote this augmented type with (t_a, V_a). The interpretations domain may contain several copies of (t_a, V_a), but we interpret a using the first one, i.e., we define

$$a^{\mathcal{I}} := (t_a, V_a)^1.$$

Defining the interpretation of the role names is a bit more tricky. Obviously, it is sufficient to define, for each role name $r \in N_R$ and each $d \in \Delta^{\mathcal{I}}$, the set $r^{\mathcal{I}}(d)$. Thus, consider an element $(t, V)^i \in \Delta^{\mathcal{I}}$. Since (t, V) is an augmented type in $\mathbb{A}$, the formula ϕ'_t has a solution σ in which exactly the Venn regions in V are non-empty, and which assigns a set of cardinality $m := k_{(t,V)}$ to the universal set. In addition, each Venn region $w \in V$ is realized by an augmented type $(t^w, V^w) \in \mathbb{A}$. Assume that the solution σ assigns the finite set $\{d_1, \ldots, d_m\}$ to the set term $\mathcal{U}$. We consider an injective mapping π of $\{d_1, \ldots, d_m\}$ into $\Delta^{\mathcal{I}}$ such that the following holds for each element d_j of $\{d_1, \ldots, d_m\}$: if d_j belongs to the Venn region $w \in V$, then

- $\pi(d_j) = (t^w, V^w)^\ell$ for some $1 \leq \ell \leq \sigma_N(v_{t^w})/n_{t^w}$;

- if w contains X_b for an individual name $b \in \mathsf{Ind}_{\mathcal{A}}$ positively, then $\ell = 1$.

Such a bijection exists since,

- $\sigma_N(v_{t^w})/n_{t^w} \geq \max\{k_{(t',V')} \mid (t', V') \in \mathbb{A}\} \geq k_{(t,V)} = m$;

- due to the presence of the cardinality constraints $|X_b| \leq 1$ in the QFBAPA formula ϕ'_t, there is at most one individual d_j that belongs to a Venn region w containing X_b positively. Any other individual d_k belongs to a different Venn region w' not containing X_b positively, and thus $S_w \subseteq t^w$ and $S_{w'} \subseteq t^{w'}$ implies $t^w \neq t^{w'}$ since $b \in t^w$ but $b \notin t^{w'}$. This shows that choosing the index $\ell = 1$ when defining $\pi(d_j)$ is possible without getting into conflict with the required choice of the index 1 for a different individual.

We now define

$$r^{\mathcal{I}}((t,V)^i) := \{\pi(d_j) \mid d_j \in \sigma(X_r)\}.$$

First, note that this definition of the interpretation of roles in $\mathcal{I}$ satisfies the role assertions in $\mathcal{A}$. To see this, assume that $r(a,b) \in \mathcal{A}$, and let $a^{\mathcal{I}} = (t_a, V_a)^1$. Then $a \in t_a$, which implies that ϕ'_{t_a} contains the cardinality constraint $|X_b \cap X_r| \geq 1$ as well as the constraint $|X_b| \leq 1$. Consequently, V_a contains exactly one Venn region w that contains X_b and X_r positively. Consider the solution of ϕ'_{t_a} used above to define $r^{\mathcal{I}}((t_a, V_a)^1)$, and let d_j be the unique individual belonging to X_b under this solution. Then this individual also belongs to X_r under this solution, and we have $\pi(d_j) = (t^w, V^w)^1 \in r^{\mathcal{I}}((t_a, V_a)^1)$. In addition, t_w contains b since S_w contains b and $S_w \subseteq t_w$. This shows that $b^{\mathcal{I}} = (t^w, V^w)^1$, and thus that the role assertion $r(a,b)$ is satisfied by $\mathcal{I}$.

To prove that $\mathcal{I}$ also satisfies the concept assertions in $\mathcal{A}$ and the ERCBox $\mathcal{R}$, we first show the following claim:

Claim: *For all concept descriptions $C \in \mathcal{M}(\mathcal{R}, \mathcal{A})$, all augmented types $(t,V) \in \mathbb{A}$, and all $i, 1 \leq i \leq \sigma_N(v_t)/n_t$, we have $C \in t$ iff $(t,V)^i \in C^{\mathcal{I}}$.*

We prove the claim by induction on the size of C:

- The cases $C = A$, $C = \neg D$, $C = D_1 \sqcap D_2$, and $C = D_1 \sqcup D_2$ can be handled as in the proof of (3) in the proof of Lemma 7.

- Now assume that $C = succ(Con)$ for a set or cardinality constraint Con.

  - If $C \in t$, then this constraint is part of the QFBAPA formula ϕ'_t obtained from t, and thus satisfied by the solution σ of ϕ'_t used to define the role successors of $(t,V)^i$. According to this definition, there is a 1–1 correspondence between the elements of $\sigma(\mathcal{U})$ and the role successors of $(t,V)^i$. This bijection π also respects the assignment of subsets of $\sigma(\mathcal{U})$ to set variables of the form X_r (for $r \in N_R$) and X_D (for concept descriptions D) occurring in ϕ'_t, i.e.,

$$(*) \quad \begin{array}{l} d_j \in \sigma(X_r) \text{ iff } \pi(d_j) \in r^{\mathcal{I}}((t,V)^i), \\ d_j \in \sigma(X_D) \text{ iff } \pi(d_j) \in D^{\mathcal{I}}. \end{array}$$

  Once $(*)$ is shown it is easy to see that $(t,V)^i \in succ(Con)^{\mathcal{I}} = C^{\mathcal{I}}$. In fact, the translation ϕ_{Con} of Con, where r is replaced by X_r and D by X_D, is a conjunct in ϕ'_t and thus σ satisfies ϕ_c. Now $(*)$ shows that (modulo the application of the bijection π), when checking whether $(t,V)^i \in succ(Con)^{\mathcal{I}}$, roles r and concepts D in ϕ_{Con} are interpreted in the same way as the set variables X_r and X_D in the solution σ of ϕ'_t. Thus the fact that σ satisfies the conjunct ϕ_{Con} of ϕ'_t implies that the role successors of $(t,V)^i$ satisfy Con, i.e., $(t,V)^i \in succ(Con)^{\mathcal{I}}$ holds. Note that, though ϕ'_t also contains set variables of the form X_b for individual names b, this is not the case for ϕ_{Con} since individuals occur only in the ABox and not in concepts.

  For role names r, property $(*)$ is immediate by the definition of $r^{\mathcal{I}}((t,V)^i)$. Now consider a concept description D such that X_D occurs in ϕ'_t. Then D occurs in Con, and is thus smaller than C, which means that we can apply induction to it. If $d_j \in \sigma(X_D)$, then the Venn region w to which d_j belongs contains X_D positively. Consequently, S_w contains D, and the augmented type (t^w, V^w) realizing w satisfies $D \in t_w$. By induction, we obtain $\pi(d_j) = (t^w, V^w)^\ell \in D^{\mathcal{I}}$. Conversely, assume that $\pi(d_j) = (t^w, V^w)^\ell \in D^{\mathcal{I}}$, where w is the Venn region to which d_j belongs w.r.t. σ. By induction, we obtain $D \in t^w$, and thus the Venn region w contains X_D positively. Since d_j belongs to this Venn region, we obtain $d_j \in \sigma(X_D)$.

– The case where $C \notin t$ can be treated similarly. In fact, in this case the constraint $\neg Con$ is part of the QFBAPA formula ϕ'_t obtained from t, and we can employ the same argument as above, just using $\neg Con$ instead of Con.

This finishes the proof of the claim. As an easy consequence of this claim we have for all C occurring in $\mathcal{R}$ that

$$C^{\mathcal{I}} = \{(t,V)^i \mid C \in t, (t,V) \in \mathbb{A},\ \text{and}\ 1 \leq i \leq \sigma_N(v_t)/n_t\}.$$

Consequently, $|C^{\mathcal{I}}| = \sum_{t \in T_{\mathbb{A}}\ \text{s.t.}\ C \in t} \sigma_N(v_t)$, which shows that $\mathcal{I}$ satisfies $\mathcal{R}$ since σ_N solves $\phi_{T_{\mathbb{A}}}$.

Finally, assume that $C(a) \in \mathcal{A}$. Then $a^{\mathcal{I}} = (t_a, V_a)^1$ and $C \in t_a$. The claim thus yields $(t_a, V_a)^1 \in C^{\mathcal{I}}$, which shows that $\mathcal{I}$ also satisfies the concept assertions in $\mathcal{A}$. □

Next we show that the algorithm is also complete, i.e., whenever $\mathcal{A}$ is consistent w.r.t. $\mathcal{R}$, then it succeeds on this input.

Lemma 19 (Completeness)**.** *Let $\mathcal{R}$ be a conjunctive $\mathcal{ALCSCC}$ ERCBox and $\mathcal{A} \neq \emptyset$ an $\mathcal{ALCSCC}$ ABox. If $\mathcal{A}$ is consistent w.r.t. $\mathcal{R}$, then Algorithm 17 succeeds on input $\mathcal{R}$ and $\mathcal{A}$.*

Proof. Assume that $\mathcal{I}$ is a model of $\mathcal{R}$ and $\mathcal{A}$. Consider the set of all types of elements of $\mathcal{I}$, i.e., $T_{\mathcal{I}} := \{t_{\mathcal{I}}(d) \mid d \in \Delta^{\mathcal{I}}\}$, where

$$\begin{aligned} t_{\mathcal{I}}(d) := {} & \{D \in \mathcal{M}(\mathcal{R},\mathcal{A}) \mid D \text{ concept description and } d \in D^{\mathcal{I}}\} \cup \\ & \{a \in \mathcal{M}(\mathcal{R},\mathcal{A}) \cap N_I \mid a^{\mathcal{I}} = d\} \cup \{\neg a \in \mathcal{M}(\mathcal{R},\mathcal{A}) \mid a \in N_I, a^{\mathcal{I}} \neq d\}. \end{aligned}$$

It is easy to see that the elements $t_{\mathcal{I}}(d)$ of $T_{\mathcal{I}}$ are indeed types. In addition, for every $a \in \mathsf{Ind}_{\mathcal{A}}$, there is exactly one type t in $T_{\mathcal{I}}$ that contains a, which is $t_{\mathcal{I}}(a^{\mathcal{I}})$. Also note that $C(a) \in \mathcal{A}$ implies $a^{\mathcal{I}} \in C^{\mathcal{I}}$, and thus $C \in t_{\mathcal{I}}(a^{\mathcal{I}})$. This shows that the types in $T_{\mathcal{I}}$ satisfy the conditions on the sets of augmented types computed in step (1) of the algorithm. However, we still need to equip our types with Venn regions.

Consider $t := t_{\mathcal{I}}(d)$ for an element $d \in \Delta^{\mathcal{I}}$. We claim that the QFBAPA formula ϕ'_t corresponding to t has as solution the substitution σ in which the universal set $\mathcal{U}$ consists of all the role successors of d, and the other set variables are assigned sets according to the interpretations of individuals, roles, and concept descriptions in the model $\mathcal{I}$. The fact that $d \in C^{\mathcal{I}}$ for all concept descriptions $C \in t$ implies that σ satisfies ϕ_t, and the fact that $\sigma(\mathcal{U})$ consists of all the role successors of d implies that $X_{r_1} \cup \ldots \cup X_{r_n} = \mathcal{U}$ is also satisfied by σ. The constraints $|X_b| \leq 1$ for $b \in \mathsf{Ind}_{\mathcal{A}}$ are satisfied since at most one role successors of d can be equal to $b^{\mathcal{I}}$. If $a \in \mathsf{Ind}_{\mathcal{A}}$ belongs to t, then $t = t_{\mathcal{I}}(a^{\mathcal{I}})$ and thus $d = a^{\mathcal{I}}$. If $r(a,b) \in \mathcal{A}$, then $b^{\mathcal{I}}$ is an r-successor of d in $\mathcal{I}$, and thus $b^{\mathcal{I}} \in \sigma(X_b) \cap \sigma(X_r)$. This shows that σ also satisfies the cardinality constraints of the form $|X_b \cap X_r| \geq 1$ in ϕ'_t.

Now, let $\{d_1, \ldots, d_m\} = \sigma(\mathcal{U})$ be the set of all role successors of d in $\mathcal{I}$, and w_i the Venn region to which d_i belongs w.r.t. σ. By Lemma 1 in [2], there is a solution σ' of ϕ'_t such that the set V of non-empty Venn regions w.r.t. σ' has cardinality $\leq N_t$ and each of these non-empty Venn regions in V is one of the Venn regions w_i, i.e., $V \subseteq \{w_1 \ldots, w_m\}$. If $a \in \mathsf{Ind}_{\mathcal{A}}$ belongs to t and $r(a,b) \in \mathcal{A}$, then there is an i such that $d_i = b^{\mathcal{I}}$. Note that the Venn region w_i then belongs to V since otherwise σ' could not be a solution of $|X_b \cap X_r| \geq 1$.

By construction, (t,V) is an augmented type. Let $\mathbb{A}_{\mathcal{I}}$ denote the set of augmented types obtained by extending the types in $T_{\mathcal{I}}$ in this way for every $d \in \Delta^{\mathcal{I}}$. By construction, for every $t \in T_{\mathcal{I}}$ there is a set of Venn regions V such that $(t,V) \in \mathbb{A}_{\mathcal{I}}$. It is easy to see that $\mathbb{A}_{\mathcal{I}}$ satisfies the conditions (1a) and (1b) considered in the first step of the algorithm, and thus there is a set of augmented types $\mathbb{A}$ computed in this first step such that $\mathbb{A}_\mathcal{I} \subseteq \mathbb{A}$. We now perform the other steps using $\mathbb{A}$ as a starting point.

First, note that no element of $\mathbb{A}_\mathcal{I}$ can be removed in Step 3 of our algorithm. This is an easy consequence of the following observation. Let T be a set of types such that $T_\mathcal{I} \subseteq T$, and let ϕ_T be obtained from $\mathcal{R}$ by replacing each $|C|$ in $\mathcal{R}$ with $\sum_{t \in T \text{ s.t. } C \in t} v_t$ and adding $v_t \geq 0$ for each $t \in T$. Since $\mathcal{I}$ is a model of $\mathcal{R}$, it is easy to see that ϕ_T has a solution that also satisfies $v_t \geq 1$ for all $t \in T_\mathcal{I}$.

Next, we show that the Venn regions occurring in some augmented type in $\mathbb{A}_\mathcal{I}$ are realized by $\mathbb{A}_\mathcal{I}$. Thus, let (t, V) be an augmented type constructed from a type $t = t_\mathcal{I}(d)$ as described above, and let $w \in V$ be a Venn region occurring in this augmented type. Then there is a role successor d_i of d such that d_i belongs to the Venn region $w = w_i$ w.r.t. the solution σ of ϕ'_t induced by $\mathcal{I}$. We know that $d_i \in D^\mathcal{I}$ for all $D \in S_w$, and thus $S_w \subseteq t_\mathcal{I}(d_i)$. Since $\mathbb{A}_\mathcal{I}$ contains an augmented type with first component $t_\mathcal{I}(d_i)$, this shows that w is realized by $\mathbb{A}_\mathcal{I}$.

We claim that, in a the run of Algorithm 17, we always have $\mathbb{A}_\mathcal{I} \subseteq \mathbb{A}$ and $T_\mathcal{I} \subseteq T_\mathbb{A}$. Obviously, this is true when we enter the second step for the first time with the set $\mathbb{A}$ satisfying $\mathbb{A}_\mathcal{I} \subseteq \mathbb{A}$. In addition, in Step 2 of our algorithm, no element of $\mathbb{A}_\mathcal{I}$ can be removed since we have seen that the Venn regions occurring in some augmented type in $\mathbb{A}_\mathcal{I}$ are realized by $\mathbb{A}_\mathcal{I}$. Finally, we have also seen above that, in Step 3 of our algorithm, no element of $T_\mathcal{I} = T_{\mathbb{A}_\mathcal{I}}$ can be removed.

Since $\mathbb{A}_\mathcal{I}$ contains, for every $a \in \mathsf{Ind}_\mathcal{A}$, an augmented type (t, V) such that $a \in t$, the algorithm cannot fail. This completes the proof of completeness. □

We have now proved that both the positive and the negative answers given by the algorithm are correct. This allows us to show our ExpTime complexity upper bound.

Theorem 20. *Consistency of $\mathcal{ALCSCC}$ ABoxes w.r.t. $\mathcal{ALCSCC}$ ERCBoxes is an* ExpTime*-complete problem.*

Proof. Given an arbitrary, not necessarily conjunctive ERCBox $\mathcal{R}$, we consider all Boolean valuations of the semi-restricted cardinality constraints occurring in $\mathcal{R}$, and collect those that evaluate the positive Boolean structure of $\mathcal{R}$ to true. For each of these valuations ρ, we consider the conjunctive ERCBox $\mathcal{R}_\rho$ that is the conjunction of all the semi-restricted cardinality constraints evaluated to true by ρ. There are exponentially many such conjunctive ERCBoxes $\mathcal{R}_\rho$, but each of them has a size that is linearly bounded by the size of $\mathcal{R}$. In addition, $\mathcal{R}$ is satisfiable iff one of the conjunctive ERCBox $\mathcal{R}_\rho$ obtained this way is satisfiable.

Thus, it remains to prove that Algorithm 17 indeed runs in exponential time on conjunctive ERCBoxes. To see this, first note that, according to Lemma 15, there are only exponentially many augmented types, and they can be computed in exponential time. In the first step, we first need to check whether condition (1b) is satisfied for exponentially many augmented types. This can clearly be done in exponential time. Then, we consider all possible ways of choosing, for every individual a, an appropriate augmented type. Since the number of individuals is polynomial and for each one there are at most exponentially many augmented types containing this individual in the first component, there are only exponentially many sets that can be generated by a combination of these choices.

For each of the sets generated in the first step, the iteration between the other two steps can happen only exponentially often since in each iteration at least one augmented type is removed. A single Step 2 takes only exponential time since for each of the exponentially many augmented types (t, V), only exponentially many other augmented types need to be considered. Finally, a single Step 3 takes only exponential time. In fact, we need to consider exponentially many systems of linear inequalities $\phi_{T_\mathbb{A}} \wedge v_t \geq 1$. Each of these systems may be of exponential size, but its solvability can be tested in time that is polynomial in this size, and thus exponential in the size of the input. Lemma 11 is applicable since adding $v_t \geq 1$ does not destroy the specific form of the system required by the lemma. □

One might ask whether the approach used here to deal with individuals in ABoxes could also be used to treat nominals in concept descriptions, where a nominal is a concept that must be interpreted as a singleton set. As usual in Description Logic, we write such a nominal as $\{o\}$ where o is an individual name. The answer to the above question is, unfortunately, negative. From a technical point of view, the claim in the proof of Lemma 18 is no longer correct since for a nominal it only holds for $i = 1$, but not for $i > 1$. However, in the induction assumption we would need this for arbitrary i and not just for $i = 1$. Using a reduction from [25], it is actually easy to see that adding nominals increases the complexity of ERCBox consistency from ExpTime to NExpTime even for $\mathcal{ALC}$. As usual, we use $\mathcal{O}$ in the name of the DL to indicate the presence of nominals.

Proposition 21. *Consistency of conjunctive $\mathcal{ALCO}$ ERCBoxes is* NExpTime*-complete.*

Proof. Membership in NExpTime follows from the fact that $\mathcal{ALCO}$ ERCBox can be expressed using $\mathcal{ALC}$ ECBoxes, whose consistency problem was shown to be in NExpTime in [8].

In [25], Tobies has shown that consistency of $\mathcal{ALCQ}$ CBoxes is NExpTime-hard, using a reduction from a bounded tiling problem. Looking closer at this reduction, one sees that actually only $\mathcal{ALC}$ concept descriptions, $\mathcal{ALC}$ CIs, and cardinality restrictions of the forms $(\geq 1\, C)$, $(\leq 1\, C)$, and $(\leq 2^n{\cdot}2^n\, C)$ for $\mathcal{ALC}$ concepts C are needed. CIs and cardinality restrictions $(\geq 1\, C)$ can easily be expressed using semi-restricted cardinality constraints, as introduced in Definition 9. Using a new nominal $\{o\}$, we can express $(\leq 1\, C)$ as $|C| \leq |\{o\}|$. To express $(\leq 2^n{\cdot}2^n\, C)$, we need a new nominal and additional auxiliary new concept names: the constraints

$$|A_0| \leq |\{o\}| \wedge |A_1| \leq 2|A_0| \wedge \ldots \wedge |A_{2n}| \leq 2|A_{2n-1}|$$

ensures that the cardinality of A_{2n} is bounded by $2^{2n} = 2^n{\cdot}2^n$. □

5 Undecidability of $\mathcal{ALCISCC}^{++}$

We next observe that a seemingly harmless extension of $\mathcal{ALCSCC}^{++}$ turns the satisfiability problem undecidable. We obtain $\mathcal{ALCISCC}^{++}$ by adding *role inverses* to $\mathcal{ALCSCC}^{++}$ by additionally allowing expressions of the form r^- for any $r \in N_R$ in all places where role names are allowed to occur. The semantics of the expression r^- is defined by $(r^-)^\mathcal{I} = \{(e, d) \mid (d, e) \in r^\mathcal{I}\}$. The key insight for showing our result is that adding this feature enables us to encode multiplication of concept extensions, allowing for a reduction from Hilbert's tenth problem. We first provide an example illustrating how "class extension multiplication" can be expressed.

Example 22. *In order to express that the cardinality of a concept C coincides with the product of the cardinalities of concepts A and B, we employ two auxiliary roles r and s. We first enforce that role r connects precisely each member of A with every member of B:*

$$A \equiv \exists r.\top \qquad B \equiv \exists r^-.\top \qquad A \sqsubseteq sat(B = r) \qquad B \sqsubseteq sat(A = r^-)$$

Next, we make sure that every domain element has precisely as many outgoing r roles as outgoing s roles:

$$\top \sqsubseteq sat(|r| = |s|)$$

Moreover, the elements with incoming s roles are precisely the instances of concept C:

$$C \equiv \exists s^{-}.\top$$

Finally, no element can have more than one incoming s role (in other words, s is inverse functional):

$$\top \sqsubseteq sat(|s^{-}| \leq 1)$$

A construction very much along the lines of the given example allows us to express Hilbert's tenth problem as an $\mathcal{ALCISCC}^{++}$ concept satisfiability problem and hence establish undecidability of the latter.

Theorem 23. *Satisfiability of $\mathcal{ALCISCC}^{++}$ concept descriptions is undecidable.*

Proof. We show the claim via a reduction from Hilbert's tenth problem, i.e., the solvability of Diophantine equations. Note that any Diophantine equation D can be transformed (possibly introducing fresh auxiliary variables) into a system $\mathcal{E}$ of equations, where each equation has one of the following three forms: (i) $x = y \cdot z$, (ii) $x = y + z$, or (iii) $x = n$, for a natural number n, such that D has an integer solution if and only if $\mathcal{E}$ has a solution in the natural numbers.

Given such a system $\mathcal{E}$ of equations over a set Var of variables, we now construct an $\mathcal{ALCISCC}^{++}$ concept expression $C_{\mathcal{E}}$ containing concept names A_x for all variables x occurring in $\mathcal{E}$, such that satisfiability of $C_{\mathcal{E}}$ coincides with the existence of a natural solution for $\mathcal{E}$. We let $C_{\mathcal{E}} = \bigsqcap_{eq \in \mathcal{E}} C_{eq}$, where C_{eq} stands for

- the concept expression $C^1_{eq} \sqcap C^2_{eq} \sqcap C^3_{eq} \sqcap C^4_{eq}$ if eq is of the form $x = y \cdot z$, where
  - $C^1_{eq} = sat\big(\neg A_y \subseteq sat(|s_{eq}|{=}0)\big)$,
  - $C^2_{eq} = sat\big(A_y \subseteq sat(|s_{eq}|{=}|A_z|)\big)$,
  - $C^3_{eq} = sat\big(\top \subseteq sat(|s^{-}_{eq}|{\leq}1)\big)$, and
  - $C^4_{eq} = sat\big(A_x = sat(|s^{-}_{eq}|{\geq}1)\big)$.
- $sat(|A_x| = |A_y| + |A_z|)$ if eq is of the form $x = y + z$,
- $sat(|A_x| = n)$ if eq is of the form $x = n$.

We now show that $\mathcal{E}$ has a solution in the natural numbers if and only if $C_{\mathcal{E}}$ is satisfiable.

For the "if" direction, assume there is some finite interpretation $\mathcal{I}$ and domain element $d \in \Delta^{\mathcal{I}}$ such that $d \in C^{\mathcal{I}}_{\mathcal{E}}$. Let $\sigma : Var \to \mathbb{N}$ be the variable assignment mapping every variable x in $\mathcal{E}$ to $|A^{\mathcal{I}}_x|$. Then, clearly, σ maps every equation eq of the form (ii) or (iii) to a true statement due to $\delta \in C^{\mathcal{I}}_{eq}$. Now consider some equation $x = y + z$ of the form (i). For this, we obtain

$$\begin{aligned}
|A^{\mathcal{I}}_x| &= |(sat(|s^{-}|{\geq}1))^{\mathcal{I}}| \text{ due to } C^4_{eq}\\
&= |\{e \mid (e', e) \in s^{\mathcal{I}}_{eq}\}|\\
&= |\{(e', e) \mid (e', e) \in s^{\mathcal{I}}_{eq}\}| \text{ due to } C^3_{eq}\\
&= |s^{\mathcal{I}}_{eq}|\\
&= \textstyle\sum_{e' \in \Delta^{\mathcal{I}}} |\{e \mid (e', e) \in s^{\mathcal{I}}_{eq}\}|\\
&= \textstyle\sum_{e' \in A^{\mathcal{I}}_y} |\{e \mid (e', e) \in s^{\mathcal{I}}_{eq}\}| \text{ due to } C^1_{eq}\\
&= \textstyle\sum_{e' \in A^{\mathcal{I}}_y} |A^{\mathcal{I}}_z| \text{ due to } C^2_{eq}\\
&= |A^{\mathcal{I}}_y| \cdot |A^{\mathcal{I}}_z|,
\end{aligned}$$

which finishes the proof of the "if" direction.

For the "only if" direction, let $\sigma : Var \to \mathbb{N}$ be a variable mapping satisfying all equations in $\mathcal{E}$. We now construct a model $\mathcal{I}$ of C_{eq} as follows:

- $\Delta^{\mathcal{I}} = \{n \in \mathbb{N} \mid 1 \leq n \leq \max_{v \in Var} \sigma(v)\}$
- $A_v^{\mathcal{I}} = \{n \in \mathbb{N} \mid 1 \leq n \leq \sigma(v)\}$
- $s_{x=y \cdot z}^{\mathcal{I}} = \{(i, k \cdot \sigma(z) + i) \mid 0 \leq k \leq \sigma(y) - 1,\ 1 \leq i \leq \sigma(z)\}$

It is straightforward to check modelhood of $\mathcal{I}$. This concludes the "only if" direction and hence the proof. □

6 Query entailment in $\mathcal{ALCSCC}^{++}$

The final result of this section is the undecidability of conjunctive query entailment for $\mathcal{ALCSCC}^{++}$. To this end, we first briefly recap the notion of (Boolean) conjunctive queries and define query entailment.

In queries, we use *variables* from a countably infinite set V. A Boolean *conjunctive query* (CQ) q is a finite set of atoms of the form $r(x, y)$ or $C(z)$, where r is a role, C is concept, and $x, y, z \in V$. A CQ q is *satisfied* by $\mathcal{I}$ (written: $\mathcal{I} \models q$) if there is a *variable assignment* $\pi : V \to \Delta^{\mathcal{I}}$ (called *match*) such that $(\pi(x), \pi(y)) \in r^{\mathcal{I}}$ for every $r(x, y) \in q$ and $\pi(z) \in C^{\mathcal{I}}$ for every $C(z) \in q$. A CQ q is *(finitely) entailed* from a knowledge base $\mathcal{K}$ (written: $\mathcal{K} \models q$) if every (finite) model $\mathcal{I}$ of $\mathcal{K}$ satisfies q.

We actually show undecidability of CQ entailment for a much weaker logic, thereby providing a very restricted fragment of constant-free and equality-free two-variable first-order logic for which finite CQ entailment is already undecidable, significantly strengthening and solidifying earlier results along those lines [20]. Our proof makes use of deterministic Turing machines (DTMs). For our purposes, it is sufficient to consider only computations starting with an empty tape. For space reasons, we assume the reader to be familiar with standard notions and constructions concerning DTMs. We call a DTM *looping* if its run starting contains repeating configurations,i.e., there are two different (and hence – due to determinism – infinitely many) points in time, where the machine's tape content, head position, and state are the same. It is easy to see that the problem of determining if a given TM is looping is undecidable.

We show our undecidability result for the DL $\mathcal{ALC}^{\text{cov}}$, a slight extension of $\mathcal{ALC}$ by *role cover axioms* of the form $\text{cov}(r, s)$ for role names r and s. An interpretation $\mathcal{I}$ satisfies $\text{cov}(r, s)$ if $r^{\mathcal{I}} \cup s^{\mathcal{I}} = \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. Role cover axioms can be expressed in $\mathcal{ALCSCC}^{++}$ via $sat\big(\top \subseteq sat(|r \cup s| = |\mathcal{U}|)\big)$, hence $\mathcal{ALC}^{\text{cov}}$ is subsumed by $\mathcal{ALCSCC}^{++}$.

In what follows, assume that a DTM $\mathcal{M}$ is given. We now describe an $\mathcal{ALC}^{\text{cov}}$ TBox $\mathcal{T}$ and conjunctive query q such that $\mathcal{T} \models q$ exactly if $\mathcal{M}$ is not looping. We provide q and $\mathcal{T}$ together with the underlying intuitions of our construction. The goal of our construction is that a countermodel (i.e., an interpretation satisfying $\mathcal{T}$ but not q) corresponds to a looping configuration sequence of $\mathcal{M}$. Thereby, the domain elements represent tape cells at certain computation steps of $\mathcal{M}$. The role h connects consecutive tape cells of the same configuration, whereas the role v connects a configuration's tape cell with the same tape cell of the successor configuration.

We start by providing the query. Intuitively, the query is meant to catch the unwanted situation that two corresponding tape cells of consecutive configurations are v-connected, but the cells to their right aren't.

$$q = \exists x, y, x', y'. v(x, y) \wedge h(x, x') \wedge h(y, y') \wedge \overline{v}(x', y') \tag{5}$$

Table 1: TBox axioms for DTM implementation

$$\top \sqsubseteq \exists aux.(\mathit{TapeStart} \sqcap \mathit{InitConf} \sqcap \mathit{State}_{\mathfrak{q}_{\mathrm{ini}}}) \tag{7}$$

$$\top \sqsubseteq \exists h.\top \sqcap \exists v.\top \tag{8}$$

$$\mathit{TapeStart} \sqsubseteq \forall v.\mathit{TapeStart} \tag{9}$$

$$\mathit{InitConf} \sqsubseteq \forall h.\mathit{InitConf} \qquad \mathit{InitConf} \sqsubseteq \mathit{Symbol}_{\square} \tag{10}$$

$$\mathit{State}_{\mathfrak{q}} \sqsubseteq \forall h.\mathit{NoHeadR} \qquad \mathit{NoHeadR} \sqsubseteq \forall h.\mathit{NoHeadR} \qquad \mathit{NoHeadR} \sqsubseteq \mathit{NoHead} \tag{11}$$

$$\exists h.\mathit{State}_{\mathfrak{q}} \sqsubseteq \mathit{NoHeadL} \qquad \exists h.\mathit{NoHeadL} \sqsubseteq \mathit{NoHeadL} \qquad \mathit{NoHeadL} \sqsubseteq \mathit{NoHead} \tag{12}$$

$$\mathit{State}_{\mathfrak{q}} \sqcap \mathit{NoHead} \sqsubseteq \bot \tag{13}$$

$$\mathit{Symbol}_{\sigma} \sqcap \mathit{Symbol}_{\sigma'} \sqsubseteq \bot \qquad \mathit{State}_{\mathfrak{q}} \sqcap \mathit{State}_{\mathfrak{q}'} \sqsubseteq \bot \tag{14}$$

$$NoHead \sqcap \mathit{Symbol}_{\sigma} \sqsubseteq \forall v.\mathit{Symbol}_{\sigma} \tag{15}$$

$$\mathit{State}_{\mathfrak{q}} \sqcap \mathit{Symbol}_{\sigma} \quad \sqsubseteq \quad \forall v.(\mathit{Symbol}_{\sigma'} \sqcap \forall h.\mathit{State}_{\mathfrak{q}'}) \tag{16}$$

$$\exists h.(\mathit{State}_{\mathfrak{q}} \sqcap \mathit{Symbol}_{\sigma}) \quad \sqsubseteq \quad \forall v.(\mathit{State}_{\mathfrak{q}'} \sqcap \forall h.\mathit{Symbol}_{\sigma'}) \tag{17}$$

$$\mathit{TapeStart} \sqcap \mathit{State}_{\mathfrak{q}} \sqcap \mathit{Symbol}_{\sigma} \quad \sqsubseteq \quad \forall v.(\mathit{State}_{\mathfrak{q}'} \sqcap \mathit{Symbol}_{\sigma'}) \tag{18}$$

We proceed by giving the axioms of $\mathcal{T}$. The following covering axiom ensures that, whenever two elements are not v-connected, they must be $\overline{v}$-connected. This is needed to enable the above query to catch the described problem.

$$\mathrm{cov}(v, \overline{v}) \tag{6}$$

The remaining TBox axioms can be found in Table 1. Axiom 7 ensures (by means of an auxiliary role aux which serves no further purpose) that there is a first tape cell of the first (initial) configuration where the head of the TM is positioned in the initial state. Axiom 8 enforces that for every cell of every configuration there is both a tape cell to its right and a corresponding tape cell in the successor configuration. Axiom 9 makes sure that, for every cell that is the first on its tape, the corresponding successor configuration's tape cell is also the first. Axioms 10 propagates the information that a cell belongs to the initial configuration along the tape, and fills the tape with blanks. Axioms 11–13 (instantiated for every state $\mathfrak{q}$) make sure that in every configuration there can only be one cell where the head is positioned. Every cell can only carry one symbol and the head can be in only one state, as ensured by Axioms 14 (for distinct symbols σ, σ' and distinct states $\mathfrak{q}, \mathfrak{q}'$). Thanks to Axiom 15, symbols on head-free cells carry over to the next configuration. As specified by the DTM's transition function, the head reads a symbol σ, writes a symbol σ', changes its state from $\mathfrak{q}$ to $\mathfrak{q}'$ and moves right (Axiom 16) or left (Axiom 17) or stays in its place whenever it is supposed to move left but is already at the leftmost tape cell (Axiom 18). This finishes the description of the TBox $\mathcal{T}$, allowing us to establish the claimed property and consequenty the undecidability result.

Proposition 24. *$\mathcal{M}$ is looping iff there is a finite model $\mathcal{I}$ of $\mathcal{T}$ with $\mathcal{I} \not\models Q$.*

Theorem 25. *Finite CQ entailment over $\mathcal{ALC}^{\mathrm{cov}}$ TBoxes is undecidable.*

Proof. According to Proposition 24, the TM looping problem can be reduced to the problem if for a given $\mathcal{ALC}^{\mathrm{cov}}$ TBox $\mathcal{T}$ and conjunctive query q, there is a finite interpretation $\mathcal{I}$ with $\mathcal{I} \models \mathcal{T}$ with $\mathcal{I} \not\models q$. Note that the latter is the case exactly if $\mathcal{T}$ does not finitely entail q. □

Finally, taking into account that $\mathcal{ALCSCC}^{++}$subsumes $\mathcal{ALC}^{\mathrm{cov}}$ and only allows for finite models, we obtain the wanted result.

Corollary 26. *Conjunctive query entailment for $\mathcal{ALCSCC}^{++}$is undecidable.*

7 Decidable querying for $\mathcal{ALCSCC}$

In stark contrast to the undecidability result just presented, we prove that conjunctive query entailment by $\mathcal{ALCSCC}$ ABoxes w.r.t. $\mathcal{ALCSCC}$ ERCBoxes is only ExpTime-complete, thus not harder than deciding knowledge base consistency for plain $\mathcal{ALC}$.

Our result employs a construction by Lutz [18], but careful and non-trivial argumentation is needed to show that the idea, conceived for arbitrary models, carries over to our finite-model case. The approach reduces entailment of some CQ q to an exponential number of ExpTime inconsistency checks in the spirit of Theorem 20, resulting in an overall ExpTime procedure. In their entirety, these mentioned checks verify if some model exists that does not admit any matches of q having a specific, forest-like shape.

It remains to argue that these specific, forest-shaped query matches of q are the only ones that matter for checking entailment. To this end, we show that all other matches can be "removed" by a model transformation consisting of the following three consecutive steps: (i) forward-unraveling, resulting in possibly-infinite structures (in Section 7.1.1) then (ii) cautious collapsing to regain finiteness while keeping the model "forest-like enough" for small conjunctive queries to match only in a tree-shaped way (in Section 7.1.2) and finally (iii) enriching the model by copies of domain elements to again satisfy the global counting constraints which had possibly become violated in the course of the previous steps (in Section 7.1.3).

To the end of this Section let $\mathcal{K}_0 = (\mathcal{A}_0, \mathcal{T}_0, \mathcal{R}_0)$ be an $\mathcal{ALCSCC}$ knowledge base composed of an ABox $\mathcal{A}_0$, a Tbox $\mathcal{T}_0$ and an ERCBox $\mathcal{R}_0$. Without loss of generality we will assume that $\mathcal{K}_0$ is *normalized*, i.e. all concepts appearing in $\mathcal{T}_0$ are of depth at most one and all concepts occurring in $\mathcal{A}_0$ and $\mathcal{R}_0$ are atomic. This can be done via a routine transformations.

7.1 The construction of sufficiently tree-like models

We start with some preliminary definitions on morphism, neighbourhoods and bisimulations.

Morphisms. A *homomorphism* from an interpretation $\mathcal{I}$ to an interpretation $\mathcal{J}$ is a function $\mathfrak{h} : \mathcal{I} \to \mathcal{J}$ satisfying for all concept names A and all role names r the following properties: if $d \in A^{\mathcal{I}}$ then $\mathfrak{h}(d) \in A^{\mathcal{J}}$ and if $(d, d') \in r^{\mathcal{I}}$ then $\big(\mathfrak{h}(d), \mathfrak{h}(d')\big) \in r^{\mathcal{J}}$. An *isomorphism* is a bijection $\mathfrak{f}$ such that both $\mathfrak{f}$ and $\mathfrak{f}^{-1}$ are homomorphisms.

Neighbourhoods. For a given interpretation $\mathcal{I}$ and an element $d \in \Delta^{\mathcal{I}}$ we denote with $\mathsf{Succ}_{\mathcal{I}}(d)$ the set of role successors of d, i.e. the set $\bigcup_{r \in N_R}\{d' : (d, d') \in r^{\mathcal{I}}\}$. Note that it is possible that $d \in \mathsf{Succ}_{\mathcal{I}}(d)$. The *forward neighbourhood* (or simply *neighbourhood*) $\mathsf{N}_{\mathcal{I}}(d)$ of d is the interpretation $\mathsf{N}_{\mathcal{I}}(d) = (\Delta^{\mathsf{N}_{\mathcal{I}}(d)}, \cdot^{\mathsf{N}_{\mathcal{I}}(d)})$ such that $\Delta^{\mathsf{N}_{\mathcal{I}}(d)} = \mathsf{Succ}_{\mathcal{I}}(d) \cup \{d\}$, $A^{\mathsf{N}_{\mathcal{I}}(d)} = A^{\mathcal{I}} \cap \Delta^{\mathsf{N}_{\mathcal{I}}(d)}$ for any concept name $A \in N_C$ and $r^{\mathsf{N}_{\mathcal{I}}(d)} = r^{\mathcal{I}} \cap (\{d\} \times \Delta^{\mathcal{I}})$ for any role name $r \in N_R$.

The next definition introduces a notion of bisimulation tailored to normalized $\mathcal{ALCSCC}$ kbs.

Definition 27. *Let $\mathcal{I}, \mathcal{J}$ be interpretations with $d \in \mathcal{I}, d' \in \mathcal{J}$. We say that d and d' are* forward-neighbourhood bisimilar *(or simply* bisimilar*), denoted with $d \equiv_{\mathsf{fb}} d'$, if there exist a function $\mathfrak{f} : \mathsf{N}_{\mathcal{I}}(d) \to \mathsf{N}_{\mathcal{J}}(d')$ (called* bisimulation*) satisfying the following conditions:*

- $\mathfrak{f} : \mathsf{N}_{\mathcal{I}}(d)\big|_{\mathsf{Succ}_{\mathcal{I}}(d)} \to \mathsf{N}_{\mathcal{J}}(d')\big|_{\mathsf{Succ}_{\mathcal{J}}(d')}$ *is a bijection, and*
- *For all $d' \in \mathsf{N}_{\mathcal{I}}(d)$, for all concept names $A \in N_C$ and all role names $r \in N_R$ equivalences $d' \in A^{\mathcal{I}} \Leftrightarrow \mathfrak{f}(d') \in A^{\mathcal{J}}$ and $(d, d') \in r^{\mathcal{I}} \Leftrightarrow (\mathfrak{f}(d), \mathfrak{f}(d')) \in r^{\mathcal{J}}$ hold.*

The following observation simplifies most of the forthcoming proofs. It can be either shown by a straightforward structural induction over the shape of $\mathcal{ALCSCC}$ concepts or deduced from Proposition 2 from [5], where the notion of $\mathcal{ALCQ}t$–bisimulation was developed.

Observation 28. *Let $\mathcal{I} \models \mathcal{K}$ be a model of a normalized $\mathcal{ALCSCC}$ knowledge base $\mathcal{K}$. For any two domain elements $d, d' \in \Delta^{\mathcal{I}}$, if d and d' are bisimilar then they satisfy the same $\mathcal{ALCSCC}$ concepts of depth at most one.*

7.1.1 Forward-unravelings of finite models

For a finite interpretation $\mathcal{I}$ with $\Delta^{\mathcal{I}}_{\text{named}}$ we denote those elements $d \in \Delta^{\mathcal{I}}$ for which $a^{\mathcal{I}} = d$ holds for some individual name $a \in \mathsf{Ind}_{\mathcal{A}}$.

Definition 29. *Let $\mathcal{I}$ be a* finite *interpretation. We define a forward-unraveling $\mathcal{I}^{\to} = (\Delta^{\mathcal{I}^{\to}}, \cdot^{\mathcal{I}^{\to}})$ of $\mathcal{I}$ as a (potentially infinite) interpretation satisfying the following conditions:*

- $\Delta^{\mathcal{I}^{\to}} = (\Delta^{\mathcal{I}})^{+} \setminus \left(\Delta^{\mathcal{I}}_{\text{named}} \cdot \Delta^{\mathcal{I}}_{\text{named}} \cdot (\Delta^{\mathcal{I}})^{*}\right)$
  In words, $\Delta^{\mathcal{I}^{\to}}$ consists of all nonempty sequences of elements from $\Delta^{\mathcal{I}}$ except those, where the first two elements are named in $\mathcal{I}$.
- *For any $a \in \mathsf{Ind}_{\mathcal{A}}$, let $a^{\mathcal{I}^{\to}} = a^{\mathcal{I}}$, i.e. a is interpreted by the one-element sequence consisting of the named element $a^{\mathcal{I}}$ from $\mathcal{I}$.*[7]
- *For concept names A, we let $A^{\mathcal{I}^{\to}} = \{w \mid \mathsf{last}(w) \in A^{\mathcal{I}}\}$, where for a given element $w \in \Delta^{\mathcal{I}^{\to}}$ we use $\mathsf{last}(w)$ to denote the last*[8] *$d \in \Delta^{\mathcal{I}}$ in the sequence w.*
- *For role names r, we let $r^{\mathcal{I}^{\to}} = r^{\mathcal{I}} \cap (\Delta^{\mathcal{I}}_{\text{named}} \times \Delta^{\mathcal{I}}_{\text{named}}) \cup \{(w, wd) \mid (\mathsf{last}(w), d) \in r^{\mathcal{I}}\}$.*

The notion of forward-unravelings differs only slightly from the classical notion of unraveling. The only difference is that the sequences starting from two named individuals are excluded from the domain and that roles linking named individuals are assigned manually by the last item from Definition 29. It is not surprising that forward-unravellings preserve satisfaction of $\mathcal{ALCSCC}$ Aboxes and Tboxes as well as conjunctive query non-entailment. The proof is standard and hinges on the fact that $w \in \Delta^{\mathcal{I}^{\to}}$ and $\mathsf{last}(w) \in \Delta^{\mathcal{I}}$ satisfy the same $\mathcal{ALCSCC}$ concepts. For CQ non-entailment it is enough to see that $\mathsf{last}(\cdot)$ is a homomorphism from $\mathcal{I}^{\to}$ to $\mathcal{I}$.

[7] For convenience, we will not syntactically distinguish elements from $\Delta^{\mathcal{I}}$ and one-element sequences from $\Delta^{\mathcal{I}^{\to}}$; in particular this means $\Delta^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}^{\to}}$.

[8] We define $\mathsf{first}(w)$ analogously.

Lemma 30. *For any normalized ABox $\mathcal{A}$ and any finite interpretation $\mathcal{I}$, if $\mathcal{I} \models \mathcal{A}$ holds, then also $\mathcal{I}^{\rightarrow} \models \mathcal{A}$ holds.*

Proof. Take an arbitrary normalized ABox $\mathcal{A}$ as well as arbitrary finite interpretation $\mathcal{I}$. Assume that $\mathcal{I} \models \mathcal{A}$ holds. Note that $\Delta^{\mathcal{I}}_{\text{named}} = \Delta^{\mathcal{I}^{\rightarrow}}_{\text{named}}$ holds since we agreed that we will not syntactically distinguish elements from $\Delta^{\mathcal{I}}$ and one-element sequences. First, see that satisfaction of assertions of the form $A(a) \in \mathcal{A}$ is guaranteed due to the third point of Definition 29 and the fact that the property $\mathsf{last}(w) = w$ holds for any $w \in \Delta^{\mathcal{I}^{\rightarrow}}_{\text{named}}$. Second, we can conclude that any assertion of the form $r(a, b) \in \mathcal{A}$ is also satisfied in $\mathcal{I}^{\rightarrow}$, due to the last item of Definition 29, more precisely the fact that $r^{\mathcal{I}} \cap (\Delta^{\mathcal{I}}_{\text{named}} \times \Delta^{\mathcal{I}}_{\text{named}}) \subseteq r^{\mathcal{I}^{\rightarrow}}$ holds. Hence $\mathcal{I}^{\rightarrow} \models \mathcal{A}$. □

An important step towards proving that forward unravelings preserve normalized $\mathcal{ALCSCC}$ TBoxes is to show that any sequence $w \in \Delta^{\mathcal{I}^{\rightarrow}}$ is forward-bisimilar to $\mathsf{last}(w) \in \Delta^{\mathcal{I}}$, i.e, the element from which w originated.

Lemma 31. *Let $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ be a normalized $\mathcal{ALCSCC}$ knowledge base and let $\mathcal{I}$ be its arbitrary finite model. Then for all domain elements $d \in \Delta^{\mathcal{I}}$ and all sequences $w \in \Delta^{\mathcal{I}^{\rightarrow}}$ the implication $d = \mathsf{last}(w) \Rightarrow d \equiv_{\mathsf{fb}} w$ holds.*

Proof. We define a function $\mathfrak{f} : \mathsf{N}_{\mathcal{I}^{\rightarrow}}(w) \to \mathsf{N}_{\mathcal{I}}(d)$, which maps the neighbourhood of w in $\mathcal{I}^{\rightarrow}$ to the neighbourhood of d in $\mathcal{I}$, as $\mathfrak{f}(x) = \mathsf{last}(x)$. The definition of $\mathfrak{f}$ is sound, since $\mathsf{last}(d)$ is defined uniquely for each sequence from $(\Delta^{\mathcal{I}})^{+}$. Moreover, see that $\mathfrak{f}^{-1} : \mathsf{N}_{\mathcal{I}}(d) \to \mathsf{N}_{\mathcal{I}^{\rightarrow}}(w)$ is defined as $\mathfrak{f}^{-1}(x) = x$ for named individuals and $\mathfrak{f}^{-1}(x) = wx$ otherwise, which is also sound due to the second and the last item of Definition 29.

We will first show that $\mathfrak{f} : \mathsf{N}_{\mathcal{I}^{\rightarrow}}(w)\big|_{\mathsf{Succ}_{\mathcal{I}^{\rightarrow}}(w)} \to \mathsf{N}_{\mathcal{I}}(d)\big|_{\mathsf{Succ}_{\mathcal{I}}(d)}$ is a bijection. One can show it by proving that equations $\mathfrak{f} \circ \mathfrak{f}^{-1} = id = \mathfrak{f}^{-1} \circ \mathfrak{f}$ hold, where id is the identity function and $\circ$ is a function-composition operator. Take an arbitrary element w from $\mathsf{N}_{\mathcal{I}^{\rightarrow}}(w)$ and assume that both w, w' are named. Then $w = \mathsf{last}(w)$ and $w' = \mathsf{last}(w')$ (since we identify named individuals with one-element sequences) and the following equations hold:

$$\mathfrak{f}(\mathfrak{f}^{-1}(w')) = \mathfrak{f}(w') = \mathsf{last}(w') = w' = \mathfrak{f}^{-1}(w') = \mathfrak{f}^{-1}(\mathsf{last}(w')) = \mathfrak{f}^{-1}(\mathfrak{f}(w')).$$

Now assume that one of w, w' is not named. Then w' is in the form $w' = we$ and the presented equations $\mathfrak{f} \circ \mathfrak{f}^{-1} = id = \mathfrak{f}^{-1} \circ \mathfrak{f}$ hold again, as it is written below:

$$\mathfrak{f}(\mathfrak{f}^{-1}(e)) = \mathfrak{f}(we) = \mathsf{last}(we) = e \text{ and } we = \mathfrak{f}^{-1}(e) = \mathfrak{f}^{-1}(\mathsf{last}(we)) = \mathfrak{f}^{-1}(\mathfrak{f}(we)).$$

Hence $\mathfrak{f}$ restricted to role successors of w is a bijection. Note that for any atomic concept A we know that $w \in A^{\mathcal{I}^{\rightarrow}}$ holds iff $d \in A^{\mathcal{I}}$ holds, due to the third item of Definition 29 (and since $d = \mathsf{last}(w) = \mathfrak{f}(w)$). Thus, the only thing which remains to be done is to show that for all $w' \in \mathsf{N}_{\mathcal{I}^{\rightarrow}}(w)$ the equivalence $(w, w') \in r^{\mathcal{I}^{\rightarrow}} \Leftrightarrow (\mathfrak{f}(w), \mathfrak{f}(w')) \in r^{\mathcal{I}^{\rightarrow}}$ holds.

Let us fix an arbitrary neighbour w' of w, i.e., a domain element $w \in \Delta^{\mathcal{I}^{\rightarrow}}$ s.t. $(w, w') \in r^{\mathcal{I}^{\rightarrow}}$ holds for some role name r. Let $d' = \mathsf{last}(w') = \mathfrak{f}(w')$ be the corresponding element in $\Delta^{\mathcal{I}}$.

We distinguish two cases.

- w, w' are not named.
  Since we agreed that $\Delta^{\mathcal{I}^{\rightarrow}}_{\text{named}} = \Delta^{\mathcal{I}}_{\text{named}}$ holds, we infer that $d = \mathfrak{f}(w) = w$ and $d' = \mathfrak{f}(w') = w'$. Thus we can use the last item of Definition 29, namely the part stating that $r^{\mathcal{I}} \cap (\Delta^{\mathcal{I}}_{\text{named}} \times \Delta^{\mathcal{I}}_{\text{named}}) = r^{\mathcal{I}^{\rightarrow}} \cap (\Delta^{\mathcal{I}}_{\text{named}} \times \Delta^{\mathcal{I}}_{\text{named}})$ and conclude the mentioned property.
- At least one of w, w' is not named.
  In this case, from the second part of the third item of Definition 29 we know that w' is actually a sequence in the form $w \cdot e$. But from the same definition as above, $(w, w') = (w, we) \in r^{\mathcal{I}^{\rightarrow}}$ holds if and only if $(\mathsf{last}(w), e) = (d, e) \in r^{\mathcal{I}}$ holds, which is exactly what we wanted to prove.

Since we have shown preservation (and non-preservation) of atomic concepts and roles by $\mathfrak{f}$ and since $\mathfrak{f}$ is a bijection, we infer that $\mathfrak{f}$ is a bisimulation. Hence $w \equiv_{\mathsf{fb}} d$ holds. ☐

As an immediate consequence of Lemma 31 we obtain that any two sequences $w, w' \in \Delta^{\mathcal{I}^{\rightarrow}}$ having the same last element are forward-bisimilar, as stated below.

Lemma 32. *For any finite interpretation $\mathcal{I}$ being a model of a normalized $\mathcal{ALCSCC}$ knowledge base $\mathcal{K}$ and any sequences $w, w' \in \Delta^{\mathcal{I}^{\rightarrow}}$ with $\mathsf{last}(w) = \mathsf{last}(w')$, the property $w \equiv_{\mathsf{fb}} w'$ holds.*

Proof. By applying Lemma 31 to w and w', we infer that $w \equiv_{\mathsf{fb}} \mathsf{last}(w)$ and $w' \equiv_{\mathsf{fb}} \mathsf{last}(w')$ holds. Since the elements $\mathsf{last}(w)$ and $\mathsf{last}(w')$ are equal, we conclude that w is bisimilar to w'. ☐

Once we have shown that $w \equiv_{\mathsf{fb}} \mathsf{last}(w)$ for any $w \in \Delta^{\mathcal{I}^{\rightarrow}}$, we can employ this fact to show that forward-unraveling preserve satisfaction of normalized TBoxes.

Lemma 33. *For any normalized $\mathcal{ALCSCC}$ TBox $\mathcal{T}$ and any finite interpretation $\mathcal{I}$, the implication $\mathcal{I} \models \mathcal{T} \Rightarrow \mathcal{I}^{\rightarrow} \models \mathcal{T}$ holds.*

Proof. Let $w \in \Delta^{\mathcal{I}^{\rightarrow}}$ be an arbitrary domain element from $\mathcal{I}^{\rightarrow}$ and let $d = \mathsf{last}(w)$ be the corresponding element from $\Delta^{\mathcal{I}}$. Let $\varepsilon = C_0 \sqsubseteq C_1$ be an arbitrary GCI from the TBox $\mathcal{T}$. Note that C_0, C_1 are not necessary atomic, but since we restricted our attention to normalized knowledge bases only, we can assume that C_0 and C_1 are $\mathcal{ALCSCC}$ concepts of depth at most one. Assume that $w \in C_0^{\mathcal{I}^{\rightarrow}}$ holds. Then, to prove that $\mathcal{I}^{\rightarrow} \models \varepsilon$ holds, we need to show that $w \in C_1^{\mathcal{I}^{\rightarrow}}$ holds. Since $w \equiv_{\mathsf{fb}} d$ holds (by Lemma 31), from Observation 28 we know that d and w satisfy the same $\mathcal{ALCSCC}$ concepts of depth at most one. Hence $d \in C_0^{\mathcal{I}}$. From the fact that $\mathcal{I}$ satisfies ε we infer that $d \in \mathcal{C}_1^{\mathcal{I}}$ holds. Again, since d and w are bisimilar, they satisfy the same $\mathcal{ALCSCC}$ concepts of depth ≤ 1 and thus $w \in \mathcal{C}_1^{\mathcal{I}^{\rightarrow}}$ holds too. Due to the fact that w and ε were arbitrarily chosen, we conclude that $\mathcal{I}^{\rightarrow} \models \mathcal{T}$ holds. ☐

From the construction of forward unravelings one can immediately see that it also preserves non-entailment of conjunctive queries. Without loss of generality we can always assume that CQs contains only atomic concepts (e.g. by introducing a fresh name A_C for each concept C and putting the GCI $C \equiv A_C$ inside the TBox).

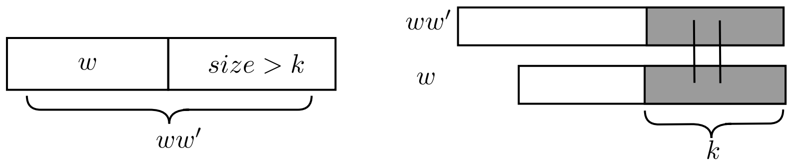
Lemma 34. *For any finite interpretation $\mathcal{I}$ and any conjunctive query q, if $\mathcal{I} \not\models q$ holds then $\mathcal{I}^{\rightarrow} \not\models q$ holds too.*

Proof. Assume that $\mathcal{I} \not\models q$ holds but $\mathcal{I}^{\rightarrow}$ entails q. Then there exists a match π of q on $\mathcal{I}^{\rightarrow}$. Note that $\mathfrak{h}(x) = \mathsf{last}(x)$ is a homomorphism $\mathcal{I}^{\rightarrow}$ to $\mathcal{I}$. Indeed, the preservation of atomic concepts by $\mathfrak{h}$ can be deduced from the third item of Definition 29, and the fact that if $(d, d') \in r^{\mathcal{I}^{\rightarrow}}$ holds then $(\mathfrak{h}(d), \mathfrak{h}(d')) \in r^{\mathcal{I}}$ holds can be inferred from the last item of Definition 29. However, in that case π' with $\pi'(x) = \mathfrak{h}(\pi(x))$ would be a match of q on $\mathcal{I}$, which contradicts the initial assumption $\mathcal{I} \not\models q$. Thus $\mathcal{I}^{\rightarrow} \not\models q$ holds. ☐

7.1.2 Loosening of finite unravelings

Unraveling removes non-forest-shaped query matches, however, $\mathcal{I}^{\rightarrow}$ does not need to be finite even if $\mathcal{I}$ is. To regain finiteness without re-introducing query matches, we are going to introduce the notion of k-*loosening.*

For a given finite interpretation $\mathcal{I}$, we say that an element $u \in \Delta^{\mathcal{I}^{\rightarrow}}$ is k*–blocked* by its prefix w, if $u = ww'$ for some w' of length longer than k, and w's and u's suffixes of length k coincide. The definition is depicted below. The definition is depicted below.



We also say that w is *minimally k–blocked* if it is k–blocked (by some prefix), but none of its prefixes is k–blocked. With $\mathsf{Bl}^{[k]}_{\mathcal{I}^{\rightarrow}}$ we denote the set of minimally k–blocked elements in $\mathcal{I}^{\rightarrow}$.

Definition 35. *For a given finite interpretation $\mathcal{I}$ we define its k–loosening $\mathcal{I}^{[k]} = (\Delta^{\mathcal{I}^{[k]}}, \cdot^{\Delta^{\mathcal{I}^{[k]}}})$ as an interpretation obtained from $\mathcal{I}^{\rightarrow}$ by exhaustively selecting minimally k–blocked elements v from $\mathsf{Bl}^{[k]}_{\mathcal{I}^{\rightarrow}}$ (k–blocked by some w), removing all of descendants of v and identifying v and w. More formally, we enumerate the set of minimally k–blocked elements $\mathsf{Bl}^{[k]}_{\mathcal{I}^{\rightarrow}} = \{v_1, v_2, \ldots, v_n\}$ and define a sequence of auxiliary interpretations $\mathcal{J}_0 = \mathcal{I}, \ldots, \mathcal{J}_n = \mathcal{I}^{[k]}$, where the i–th interpretation $\mathcal{J}^i = (\Delta^{\mathcal{J}_i}, \cdot^{\mathcal{J}_i})$ for any $i > 0$ is defined as:*

- $\Delta^{\mathcal{J}_i} = \Delta^{\mathcal{J}_{i-1}} \setminus \big(v_i \cdot (\Delta^{\mathcal{I}})^*\big)$
- ${\Delta_{\text{named}}}^{\mathcal{J}_i} = {\Delta_{\text{named}}}^{\mathcal{J}_{i-1}}$ *and for any $a \in \mathsf{Ind}_{\mathcal{A}}$ the condition $a^{\mathcal{J}_i} = a^{\mathcal{J}_{i-1}}$ is satisfied,*
- $A^{\mathcal{J}_i} = A^{\mathcal{J}_{i-1}} \cap \Delta^{\mathcal{J}_i}$ *for any concept name $A \in N_C$*
- $r^{\mathcal{J}_i} = r^{\mathcal{J}_{i-1}} \cap \big(\Delta^{\mathcal{J}_i} \times \Delta^{\mathcal{J}_i}\big) \cup \{(w, v_i') \mid (w, v_i) \in r^{\mathcal{J}_{i-1}}\}$, *for any role name $r \in N_R$, where v_i' is the element k–blocking v_i in $\mathcal{I}^{\rightarrow}$.*

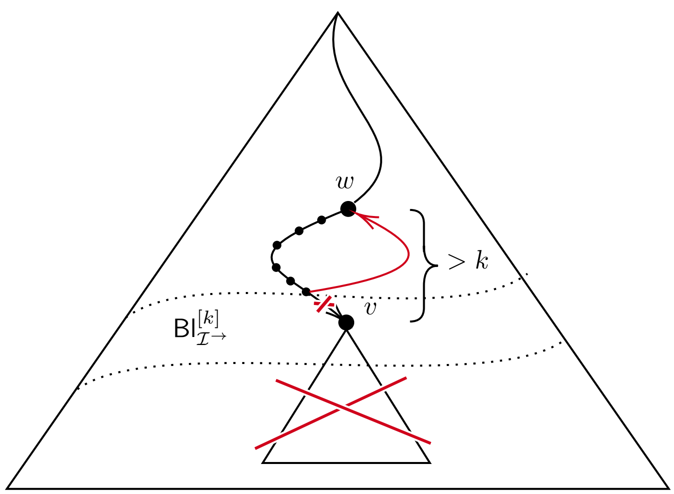


Figure 1: A single step of the construction of $\mathcal{I}^{[k]}$.

We first argue that k–loosening of a finite interpretation is also finite.

Lemma 36. *For any finite interpretation $\mathcal{I}$, its k–loosening $\mathcal{I}^{[k]}$ for any natural $k > 0$ is finite.*

Proof. Take an arbitrary finite $\mathcal{I}$ and observe that the branching of k–loosening is finite due to finiteness of $\mathcal{I}$ and each element of $\mathcal{I}^{[k]}$ has only finite number of successors (by pigeon-hole principle the blocking eventually occurs on every branch of $\mathcal{I}^{\rightarrow}$). Hence by employing (the contraposition) of the König's Lemma, we conclude that $\mathcal{I}^{[k]}$ is finite. $\square$

Like unravelings, k-loosenings preserve satisfaction of normalized Aboxes and Tboxes, as well as CQ non-entailment. However, ERCBoxes might become violated in the construction. We startfrom the ABox preservation.

Lemma 37. *For any finite $\mathcal{I}$ and any normalized ABox $\mathcal{A}$ and any natural $k > 0$, the implication if $\mathcal{I} \models \mathcal{A}$ then $\mathcal{I}^{[k]} \models \mathcal{A}$ holds.*

Proof. Assume that $\mathcal{I} \models \mathcal{A}$ holds. Then, due to Lemma 30 we know that $\mathcal{I}^{\rightarrow} \models \mathcal{A}$ holds. Observe that $\Delta^{\mathcal{I}^{[k]}}$ is a subset of $\Delta^{\mathcal{I}^{\rightarrow}}$, due to the first item of Definition 35. Moreover the sets $\Delta^{\mathcal{I}^{\rightarrow}}_{\text{named}}$ and $\Delta^{\mathcal{I}^{[k]}}_{\text{named}}$ are equal, due to the second item of Definition 35. Since the k–loosening construction does not affect the ABox part of $\mathcal{I}^{\rightarrow}$ (e.g. those elements are not k–blocked for any k, see also the second item of Definition 35) we conclude that $\mathcal{I}^{[k]}$ is a model of $\mathcal{A}$. $\square$

Towards proving the TBox preservation of k–loosening, we prepare a bisimulation argument.

Lemma 38. *Let $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ be a normalized $\mathcal{ALCSCC}$ knowledge base and let $\mathcal{I}$ be its arbitrary finite model. Then any $w \in \Delta^{\mathcal{I}^{[k]}}$ is bisimilar to $\mathsf{last}(w) \in \Delta^{\mathcal{I}}$.*

Proof. Take an arbitrary domain element $w = w_{\mathcal{I}^{[k]}} \in \Delta^{\mathcal{I}^{[k]}}$ and, since $\Delta^{\mathcal{I}^{[k]}} \subseteq \Delta^{\mathcal{I}^{\rightarrow}}$ holds (see: Definition 35), let $w_{\mathcal{I}^{\rightarrow}} = w$ be the corresponding element from $\Delta^{\mathcal{I}^{\rightarrow}}$. To show that w and $\mathsf{last}(w)$ are bisimilar, is sufficient prove that $w_{\mathcal{I}^{[k]}} \equiv_{\mathsf{fb}} w_{\mathcal{I}^{\rightarrow}}$ and use Lemma 31.

We proceed as follows. We define a function $\mathfrak{f} : \mathsf{N}_{\mathcal{I}^{[k]}}(w) \rightarrow \mathsf{N}_{\mathcal{I}^{\rightarrow}}(w)$ as $\mathfrak{f}(w') = w'$ for all $w' \in \mathsf{N}_{\mathcal{I}^{[k]}}(w) \cap \mathsf{N}_{\mathcal{I}^{\rightarrow}}(w)$ and $f(w') = w \cdot \mathsf{last}(w')$ otherwise (note that in this case w' is some of minimally k–blocked elements).

We first argue that $\mathfrak{f}$ is a function. Since $\Delta^{\mathcal{I}^{[k]}} \subseteq \Delta^{\mathcal{I}^{\rightarrow}}$ holds, we infer that $\mathfrak{f}$ is an identity function on the set $\mathsf{N}_{\mathcal{I}^{[k]}}(w) \cap \mathsf{N}_{\mathcal{I}^{\rightarrow}}(w)$, thus well-defined. The problematic case is when w' is not included in $\mathsf{N}_{\mathcal{I}^{[k]}}(w) \cap \mathsf{N}_{\mathcal{I}^{\rightarrow}}(w)$. Observe that in this case w' was identified, during the construction of $\mathcal{I}^{[k]}$, with some k–blocked element $v \in \mathsf{Bl}^{[k]}_{\mathcal{I}^{\rightarrow}}$, which originally was a successor of w. It means that v was k–blocked by w' and from the definition of k–blocked elements we infer that w' and v share the same suffix of length k. Thus w' and v share the same last element. Since v is a successor of w, then $v = w \cdot \mathsf{last}(v) = w \cdot \mathsf{last}(w')$. Hence the definition of $\mathfrak{f}$ is sound.

To see that $\mathfrak{f} : \mathsf{N}_{\mathcal{I}^{[k]}}(w)\big|_{\mathsf{Succ}_{\mathcal{I}^{[k]}}(w)} \rightarrow \mathsf{N}_{\mathcal{I}^{\rightarrow}}(w)\big|_{\mathsf{Succ}_{\mathcal{I}^{\rightarrow}}(w)}$ is a bijection, we can restrict our attention only to the elements not included in the set $\mathsf{N}_{\mathcal{I}^{[k]}}(w) \cap \mathsf{N}_{\mathcal{I}^{\rightarrow}}(w)$, since, as we already mentioned, on such set $\mathfrak{f}$ is the identity function and thus, also a bijection. Observe that $\mathfrak{f}$ is injection for any $w' \in \mathsf{N}_{\mathcal{I}^{[k]}}(w) \setminus \mathsf{N}_{\mathcal{I}^{\rightarrow}}(w)$. Indeed, if there would be w', w'' satisfying $\mathfrak{f}(w') = \mathfrak{f}(w'')$, then it would imply that they originated from the same successor of w in $\mathcal{I}^{\rightarrow}$ (since they share the same suffix), which is clearly not possible. To see that $\mathfrak{f}$ is a surjection it is enough to see that for any successor $w' = we$ of w in $\mathcal{I}^{\rightarrow}$ the function $\mathfrak{f}$ is either identity (thus $\mathfrak{f}(w') = w'$) or w' was minimally k–blocked and hance was identified with an element sharing the same last element. Hence, $\mathfrak{f}$ (restricted to appropriate sets) is a bijection.

We will prove that $\mathfrak{f}$ is a bisimulation. In the first part we will prove the following statement:

$$\forall A \in N_C \ \forall w' \in \mathsf{N}_{\mathcal{I}^{[k]}}(w) \text{ the equivalence } w' \in A^{\mathcal{I}^{[k]}} \Leftrightarrow \mathfrak{f}(w') \in A^{\mathcal{I}^{\rightarrow}} \text{ holds.}$$

Take an arbitrary concept name A and arbitrary domain element $w' \in \mathsf{N}_{\mathcal{I}^{[k]}}(w)$. If $\mathfrak{f}(w') = w'$ then the above condition trivially holds. Assume that $\mathfrak{f}(w') \neq w'$. Then $\mathfrak{f}(w') = w\mathsf{last}(w')$ and the preservation of concepts follows from Definition 29.

In the second part we will prove:

$$\forall r \in N_R \; \forall w' \in \mathsf{N}_{\mathcal{I}^{[k]}}(w) \text{ the equivalence } (w, w') \in r^{\mathcal{I}^{[k]}} \Leftrightarrow (\mathfrak{f}(w), \mathfrak{f}(w')) \in r^{\mathcal{I}^{\rightarrow}} \text{ holds.}$$

Take an arbitrary role name r and arbitrary domain element $w' \in \mathsf{N}_{\mathcal{I}^{[k]}}(w)$. Once more, if $\mathfrak{f}(w') = w'$ then the above condition trivially holds. Assume that $\mathfrak{f}(w') \neq w'$. Then again $\mathfrak{f}(w') = w\mathsf{last}(w') = v$ and v is minimally k–blocked by w'. From Definition 35 we know that $(w, v) \in r^{\mathcal{I}^{\rightarrow}}$ iff $(w, w') \in r^{\mathcal{I}^{[k]}}$, which proves the statement about (non)preservation of roles during the construction of $\mathcal{I}^{[k]}$.

We conclude that $\mathfrak{f}$ is a bisimulation and hence $w_{\mathcal{I}^{[k]}} \equiv_{\mathsf{fb}} w_{\mathcal{I}^{\rightarrow}}$ holds. □

The TBox preservation follows immediately from the previous lemma.

Lemma 39. *For any finite $\mathcal{I}$ and any normalized TBox $\mathcal{T}$ and any natural $k > 0$, the implication if $\mathcal{I} \models \mathcal{T}$ then $\mathcal{I}^{[k]} \models \mathcal{T}$ holds.*

Proof. Take an arbitrary finite interpretation $\mathcal{I}$, a normalized TBox $\mathcal{T}$ and a positive integer k. Assume that $\mathcal{I} \models \mathcal{T}$ holds. To prove that each GCI ε from $\mathcal{T}$ is also satisfied in $\mathcal{I}^{[k]}$, we apply the same reasoning as we already done for Lemma 33. Namely, it is sufficient to prove that the k–loosening construction is concept preserving but it can be concluded from Definition 27 (of bisimulation) and from Lemma 38. □

Lemma 40. *For any $k \in \mathbb{N}$, if $\mathcal{I} \not\models q$ then $\mathcal{I}^{[k]} \not\models q$.*

Proof. Assume that $\mathcal{I} \not\models q$, but $\mathcal{I}^{[k]} \models q$. In this case there exists a match π of q on $\mathcal{I}^{[k]}$. By using the same ideas as for Lemma 34 we argue that in this case π' with $\pi'(x) = \mathsf{last}(\pi(x))$ would be a match of q on $\mathcal{I}$, which contradicts with $\mathcal{I} \not\models q$. Thus $\mathcal{I}^{\rightarrow} \not\models q$ holds. □

For a given interpretation $\mathcal{J}$, an *anonymous cycle* is simply a word $w \in (\Delta^{\mathcal{J}})^{+} \cdot (\Delta^{\mathcal{J}} \setminus \Delta^{\mathcal{J}}_{\text{named}}) \cdot (\Delta^{\mathcal{J}})^{+}$, where first and the last element are the same, and for any two consecutive elements d_i, d_{i+1} of w there exists a role r witnessing $(d_i, d_{i+1}) \in r^{\mathcal{J}}$. The *girth* of $\mathcal{J}$ is the length of the smallest anonymous cycle in $\mathcal{J}$ if such a cycle exists or ∞ otherwise. The main feature of the k–loosening $\mathcal{I}^{[k]}$ is that the girth of $\mathcal{I}^{[k]}$ is at least k, as proven below.

Lemma 41. *For any $k \in \mathbb{N}$ and any finite interpretation $\mathcal{I}$, the girth of $\mathcal{I}^{[k]}$ is at least k.*

Proof. We will prove inductively over immediate structures $\mathcal{J}_0 = \mathcal{I}^{\rightarrow}, \mathcal{J}_1, \ldots, \mathcal{J}_n = \mathcal{I}^{[k]}$ produced in Definition 35 that each of them have girth greater than k. For $i = 0$ it is clear that $\mathcal{J}_0$ has girth at least k (actually its girth is ∞). Assume that for all $i < m$ the girth of each $\mathcal{J}_i$ for $i < m$ is at least k. We will show that the girth of $\mathcal{I}_m$ is at least k.

For contradiction assume that the girth of $\mathcal{J}_m$ is smaller than k. We recall that v_m is the m–th minimally k–blocked elements from $\mathsf{Bl}^{[k]}_{\mathcal{I}^{\rightarrow}}$ and v'_m is the element k–blocking v_m. Since $\mathcal{J}_m$ was obtained from $\mathcal{J}_{m-1}$ and the girth of $\mathcal{J}_{m-1}$ is at least k then the only possibility of a anonymous cycle of length at least k to be present in $\mathcal{J}_m$ is to contain a freshly added edge between predecessors w of v_m and v'_m, namely (w, v'_m) for some $r \in N_R$ as a replacement for an original edge (w, v_m).

Let ρ be an arbitrary shortest anonymous cycle in $\mathcal{J}_{m-1}$. As we already discussed it contains an edge (w, v_m) between some domain element w. Hence ρ is in the form $(w, v'_m)\rho'$ where ρ' is some path from v'_m to w. But note that due the definition of k–blocked element the distance between v_m and v'_m is at least k. Hence ρ' is of length at least k. Thus ρ is not shorter than k, which contradict our initial assumption. Hence the girth of $\mathcal{J}_m$ is at least k, which allows us to conclude that the girth of $\mathcal{J}_n = \mathcal{I}^{[k]}$ is also at least k. □

Once k is greater than the number of atoms in q (denoted with $|q|$), the k–loosening of a model is still "locally acyclic enough" so the query matches only in a "forest-shaped" manner. We will exploit this property when designing an algorithm for deciding conjunctive query entailment in Section 7.2.

Lemma 42. *For every conjunctive query q, a positive integer $k > |q|$ and a finite interpretation $\mathcal{I}$, the following equivalence $\mathcal{I}^{\rightarrow} \models q \Leftrightarrow \mathcal{I}^{[k]} \models q$ holds.*

Proof. Let $\mathsf{suff}_s(w)$ be a function which for an input word $w \in (\Delta^{\mathcal{I}})^+$ returns w if $|w| \leq s$ or its suffix of length s otherwise. Moreover let $\mathcal{I}_k^{\rightarrow}$ be the substructure of $\mathcal{I}^{\rightarrow}$ with domain restricted to sequences of length at most k only. Note that $\mathfrak{h}(w) = \mathsf{suff}_k(w)$ is a homomorphism from $\mathcal{I}^{\rightarrow}$ to $\mathcal{I}^{\rightarrow}$ (since $w \equiv_{\mathsf{fb}} \mathsf{suff}_k(w)$, see the proof of Lemma 33). Hence if there is a match π of q in $\mathcal{I}^{\rightarrow}$, there is also a match π' of q in $\mathcal{I}_k^{\rightarrow}$. Since $\mathcal{I}_k^{\rightarrow}$ is a substructure of $\mathcal{I}^{[k]}$ (due to the definition of minimally k–blocked elements and Definition 35), hence π' is also a match in $\mathcal{I}^{[k]}$.

For the opposite way, that i.e., $\mathcal{I}^{[k]} \models q$ implies $\mathcal{I}^{\rightarrow} \models q$, it is sufficient to show (since $k > |q|$) that there is a homomorphism from any substructure of the size k of $\mathcal{I}^{[k]}$ to $\mathcal{I}^{\rightarrow}$. Take an arbitrary element $w \in \Delta^{\mathcal{I}^{[k]}}$ and take a interpretation $\mathcal{I}_w^{[k]}$ be an interpretation obtained by restricting the domain to elements reachable from w in at most k steps. More formally we define the sets $R_i(w)$ of those elements reachable from w in at most i steps, i.e. $R_0(w) = \{w\}$, and $R_i(w) = R_{i-1}(w) \cup \{v \in \Delta^{\mathcal{I}^{[k]}} \mid \exists r \in N_R \ (u, v) \in r^{\mathcal{I}^{[k]}} \wedge u \in R_{i-1}(w)\}$ for all $i > 0$. We set $\Delta_w^{\mathcal{I}^{[k]}} = R_k(w)$. First see that $\mathcal{I}_w^{[k]}$ is a tree-shaped. Indeed if it would contain an anonymous cycle of length at most k it would contradict the fact that the girth of $\mathcal{I}^{[k]}$ is at least k (by Lemma 41). Hence we take a homomorphism $\mathfrak{h} : \mathcal{I}_w^{[k]} \to \mathcal{I}^{\rightarrow}$ defined as $\mathfrak{h}(x) = \mathsf{suff}_k(x)$ and see that if there is a match π of q in $\mathcal{I}^{[k]}$, then $\pi' = (\mathfrak{h} \circ \pi)$ would also be a match of q in $\mathcal{I}^{\rightarrow}$. □

7.1.3 Making ERCBoxes be satisfied again

We next consider how to adjust a k-loosening such that it again satisfies the initial ERCBox. Since role inverses are not expressible in $\mathcal{ALCSCC}$, creating multiple copies of a single element and forward-linking them to other elements precisely in the same way as the original element, can be done without any harm to modelhood nor query-non-entailment. We formalize this intuition below.

Definition 43. *For any interpretation $\mathcal{I}$ and any sets $S \subseteq (\Delta^{\mathcal{I}} \times \mathbb{N}_+)$ we define the S–*duplication *of $\mathcal{I}$ as the interpretation $\mathcal{I}_{+S} = (\Delta^{\mathcal{I}}_{+S}, \cdot^{\mathcal{I}_{+S}})$ with:*

- $\Delta^{\mathcal{I}}_{+S} = \Delta^{\mathcal{I}} \cup \bigcup_{(v,n) \in S} \{v_{cpy}^{(i)} \mid 1 \leq i \leq n\}$,
- *$a^{\mathcal{I}_{+S}} = a^{\mathcal{I}}$ for each individual name $a \in \mathsf{Ind}_{\mathcal{A}}$,*
- *For concept names $A \in N_C$ and role names $r \in N_R$ we set:*
  - $A^{\mathcal{I}_{+S}} = A^{\mathcal{I}} \cup \bigcup_{(v,n) \in S} \left\{ v_{cpy}^{(i)} \mid 1 \leq i \leq n \wedge v \in A^{\mathcal{I}} \right\}$, *and,*
  - $r^{\mathcal{I}_{+S}} = r^{\mathcal{I}} \cup \bigcup_{(v,n) \in S} \left\{ (v_{cpy}^{(i)}, w) \mid 1 \leq i \leq n \wedge (v, w) \in r^{\mathcal{I}} \right\}$.

As in the case of previous constructions, one can show that the S–duplication of $\mathcal{I}$ preserves satisfaction of ABoxes and TBoxes.

Lemma 44. *For any finite $\mathcal{I}$ and normalized ABox $\mathcal{A}$ and normalized TBox $\mathcal{T}$, if $\mathcal{I} \models (\mathcal{A}, \mathcal{T})$, then for any $S \subseteq (\Delta^{\mathcal{I}} \times \mathbb{N}_+)$, the S–duplication $\mathcal{I}_{+S}$ of $\mathcal{I}$ is also a model of $(\mathcal{A}, \mathcal{T})$.*

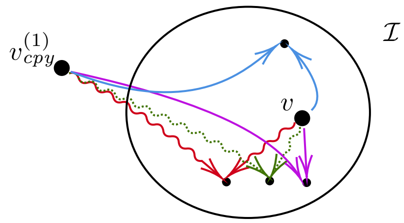


Figure 2: The interpretation $\mathcal{I}_{+\{(v,1)\}}$ obtained from $\mathcal{I}$ by duplicating a node v.

Proof. Since $\mathcal{I}$ is a submodel of $\mathcal{I}_{+S}$ we conclude that $\mathcal{I}_{+S} \models \mathcal{A}$. To see that S–duplication does not violate the TBox $\mathcal{T}$, it is sufficient to see that for any $i \in \mathbb{N}_+$ and $v \in \Delta^{\mathcal{I}}$ an element $v_{cpy}^{(i)}$ is bisimilar to v (which follows immediately from Definition 43). Hence $\mathcal{I}_{+S} \models (\mathcal{A}, \mathcal{T})$. □

Moreover a conjunctive query q has a match in $\mathcal{I}$ if and only if it has a match in $\mathcal{I}_{+S}$.

Lemma 45. *For any conjunctive query q and any $S \subseteq (\Delta^{\mathcal{I}} \times \mathbb{N}_+)$ and any interpretation $\mathcal{I}$, the equivalence $\mathcal{I} \models q \Leftrightarrow \mathcal{I}_{+S} \models q$ holds.*

Proof. Without loss of generality we assume all concepts appearing in q are atomic. If $\mathcal{I}$ has a match π of q, then trivially π is also a match in $\mathcal{I}_{+S}$ (due to the fact that $\mathcal{I}$ is a submodel of $\mathcal{I}_{+S}$). For the second direction, assume that there is a query match π of q in $\mathcal{I}_{+S}$. Let us define $\mathfrak{h} : \mathcal{I}_{+S} \to \mathcal{I}$ as $\mathfrak{h}\big(v_{cpy}^{(i)}\big) = v$ for freshly copied elements and as $\mathfrak{h}(v) = v$ otherwise. It is easy to see that $\mathfrak{h}$ is a homomorphism, and hence $\mathfrak{h} \circ \pi$ is a match of q in $\mathcal{I}$. Thus the equivalence $\mathcal{I} \models q \Leftrightarrow \mathcal{I}_{+S} \models q$ holds. □

From Lemma 45 and Lemma 42 we can immediately conclude:

Lemma 46. *For any conjunctive query q, any positive integer $k > |q|$ and any finite interpretation $\mathcal{I}$ the following equivalence holds: $\mathcal{I}^{\rightarrow} \models q \Leftrightarrow \mathcal{I}_{+S}^{[k]} \models q$.*

Note that for any finite $\mathcal{I}$ being a model of a normalized $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ it could be the case that $\mathcal{I}^{[k]}$ does not satisfy the ERCBox $\mathcal{R}$ anymore. However, the inequalities from $\mathcal{R}$ have the convenient property that if a vector $\vec{x}$ containing the cardinalities of all atomic concepts' extensions is a solution to $\mathcal{R}$, then also a vector $c \cdot \vec{x}$, i.e., the vector obtained by multiplying each entry of $\vec{x}$ by a constant c, is a solution to $\mathcal{R}$. Thus there is also a solution to $\mathcal{R}$ in the shape $(1 + |\Delta^{\mathcal{I}^{[k]}}|) \cdot \vec{x_{\mathcal{I}}}$, where $\vec{x_{\mathcal{I}}}$ is the solution to $\mathcal{R}$ describing the atomic concept extensions' cardinalities in $\mathcal{I}$. Since $\mathcal{I}^{[k]}$ preserves (non-)emptiness of all concepts from $\mathcal{I}$, we can simply duplicate an appropriate number of elements from $\mathcal{I}^{[k]}$, until the ERCBox $\mathcal{R}$ will be satisfied again. The whole procedure is described in the forthcoming lemma.

Lemma 47. *For any consistent normalized $\mathcal{ALCSCC}$ knowledge base $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ and for any of its finite models $\mathcal{I}$ there exists a finite $S \subseteq (\Delta^{\mathcal{I}} \times \mathbb{N}_+)$ such that $\mathcal{I}_{+S}^{[k]} \models (\mathcal{A}, \mathcal{T}, \mathcal{R})$ holds.*

Proof. Let $\mathbb{C}$ be the set of all atomic concepts appearing in normalized $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$. In this proof, a *type* means a conjunction of (possibly negated) concepts from $\mathbb{C}$. With $\mathbb{T}_{\mathbb{C}}$ we denote the set of all possible types.

The ERCBox $\mathcal{R}'$ is obtained from $\mathcal{R}$ by replacing each inequality ε from $\mathcal{R}$ of the form:

$$\varepsilon = N_1|C_1| + \ldots + N_k|C_k| + B \leq N_{k+1}|C_{k+1}| + \ldots + N_{k+\ell}|C_{k+\ell}|$$

with the corresponding inequality ε':

$$\varepsilon = \Sigma_{i=1}^{k} N_k\big(\Sigma_{C \in \mathbb{C}, C \models C_i}|C|\big) + B \leq \Sigma_{i=k+1}^{k+\ell} N_k\big(\Sigma_{C \in \mathbb{C}, C \models C_i}|C|\big).$$

Note that any model $\mathcal{I} \models (\mathcal{A}, \mathcal{T}, \mathcal{R})$ is also a model of $(\mathcal{A}, \mathcal{T}, \mathcal{R}')$ and vice versa.

Let $\vec{x_{\mathcal{I}}}$ be the solution to $\mathcal{R}'$ describing the types' cardinalities in $\mathcal{I}$ (such solution exists since $\mathcal{I} \models \mathcal{R}'$). As we have already mentioned before, the inequalities from $\mathcal{R}$ have the convenient property that if a vector $\vec{x}$ is a solution to $\mathcal{R}'$, then also a vector $c \times \vec{x}$, i.e., the vector obtained by multiplying each entry of $\vec{x}$ by a constant c, is a solution to $\mathcal{R}'$. Thus there is also a solution $\vec{y}$ to $\mathcal{R}'$ in the shape $\vec{y} = (1 + |\Delta^{\mathcal{I}^{[k]}}|) \cdot \vec{x_{\mathcal{I}}}$.

The desired set $S \subseteq \mathbb{N} \times \Delta^{\mathcal{I}^{[k]}}$ is defined as follows. It is composed of all pairs $(c - |t^{\mathcal{I}^{[k]}}|, w_t)$ for each type $t \in \mathbb{T}_{\mathbb{C}}$ having a non-zero entry c in $\vec{y}$ (where w_t is an arbitrary fixed domain element from $\mathcal{I}^{[k]}$ having a type t). Note that such an element w_t exists since the k–loosening and forward-unravelings preserve types (see e.g proofs of Lemma 39 and Lemma 33).

It remains to argue that $\mathcal{I}^{[k]}_{+S} \models (\mathcal{A}, \mathcal{T}, \mathcal{R})$ holds. To see that $\mathcal{I}^{[k]}_{+S} \models \mathcal{R}$ it is enough to see that $\mathcal{I}^{[k]}_{+S} \models \mathcal{R}$ holds due to the fact that the vector describing the types' cardinalities in $\mathcal{I}^{[k]}_{+S}$ is equal to $\vec{y}$ (and $\vec{y}$ was obtained by multiplying each entry of the initial solution $\vec{x_{\mathcal{I}}}$). Moreover we conclude $\mathcal{I}^{[k]}_{+S} \models (\mathcal{A}, \mathcal{T})$ holds from Lemma 44. Hence $\mathcal{I}^{[k]}_{+S} \models (\mathcal{A}, \mathcal{T}, \mathcal{R})$. □

This concludes our construction, the core result of which can be informally stated as follows: *For any $\mathcal{ALCSCC}$ knowledge base $\mathcal{K}$ and every CQ q holds: if $\mathcal{K} \models q$ then there is a forest-shaped query match of q into every model of $\mathcal{K}$.* This follows from the fact that the any model of $\mathcal{K}$ not admitting such a match would allow us to construct a model without any query matches, contradicting the assumption. We make this statement more formal by introducing the forthcoming notion of n–acyclic models.

7.1.4 The notion of n–acyclic models

Given a finite interpretation $\mathcal{J}$ we say that it is *k–acyclic*, if there exists a finite interpretation $\mathcal{I}$ such that $\mathcal{J} = \mathcal{I}^{[k]}_{+S}$ holds for some finite set $S \subseteq (\Delta^{\mathcal{I}} \times \mathbb{N}_+)$.

The next lemma states that to falsify conjunctive query we do not need to look for arbitrary finite counter-models but it is enough to consider the class of $(|q| + 1)$–acyclic models. Indeed:

Lemma 48. *For any normalized $\mathcal{ALCSCC}$ knowledge base $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ and any conjunctive query q, if there is a finite interpretation such that $\mathcal{I} \models \mathcal{K}$ but $\mathcal{I} \not\models q$, then there is a $(|q| + 1)$–acyclic model $\mathcal{I}'$ such that $\mathcal{I}' \models \mathcal{K}$ and $\mathcal{I}' \not\models q$.*

Proof. It is enough to take $\mathcal{J} = \mathcal{I}^{[(|q|+1)]}_{+S}$ for S given in 47. The modelhood preservation follows from Lemma 39 and Lemma 44. Query non entailment is due to Lemma 40 and Lemma 42. □

Moreover conjunctive query entailment over $(|q| + 1)$–acyclic models is equivalent to entailment over their forward-unravelings. This fact follows directly from Lemma 46.

Lemma 49. *For any interpretation $\mathcal{I}$ being a $(|q| + 1)$–acyclic model of an $\mathcal{ALCSCC}$ knowledge base $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ composed of a normalized ABox $\mathcal{A}$, TBox $\mathcal{T}$ and ERCBox $\mathcal{R}$ the equivalence $\mathcal{I} \models q \Leftrightarrow \mathcal{I}^{\rightarrow} \models q$ holds.*

Due to Lemma 49 we can restrict our attention to query matches over the unfolding of $(|q| + 1)$–acyclic models only. it allow us to use a machinery of spoilers, splittings and fork rewrittings from [18], developed for deciding unrestricted CQ entailment, to the case of finite query entailment with only some minor modifications.

7.2 Deciding query entailment in exponential time

Now we are ready to employ the announced exponential time method for deciding conjunctive query entailment from [18]. For a given $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ and a query q, we enumerate a set of $\mathcal{ALCH}^{\cap}$ knowledge bases $\mathcal{K}_s = (\mathcal{A}', \mathcal{T}')$ called *spoilers* and check whether $\mathcal{K} \cup \mathcal{K}_s$ is consistent. Spoilers are modeled to prevent forest-shaped query matches. They are constructed by, on the one hand, rolling-up tree-shaped partial query matches into concepts and forbidding existence of such concept in a model and, on the other hand, forbidding certain behaviour of the Abox part of a model. Lutz [18] shows that one can restrict ones attention to exponentially many spoilers and that the size of each such spoiler is only polynomial in $|\mathcal{K}|$ and $|q|$. The algorithm for CQ entailment is then obtained by simply replacing Lutz's satisfiability algorithm for $\mathcal{ALCH}^{\cap}$ knowledge bases[9] by our finite satisfiability algorithm for $\mathcal{ALCSCC}$ knowledge bases from the previous sections. We derive correctness of the procedure as follows: $\mathcal{K} \cup \mathcal{K}_s$ is satisfiable for some spoiler $\mathcal{K}_s$ exactly if there is a model of $\mathcal{K}$ without forest-shaped matches of q and hence – thanks to our above argument – there is a model without any match of q.

Let q be a conjunctive query and let $\mathrm{Var}(q)$ be the set of variables appearing in q. Through this Section we always assume that q contains only atomic concepts and no answer variables. Note that q can be seen as a directed graph $G_q = (V_q, E_q)$, where vertices from V_q are simply variables from $\mathrm{Var}(q)$ and for any two nodes x, y there exists an edge $(x, y) \in E_q$ between them if and only ifs $r(x, y) \in q$ for some $r \in N_r$. We say that q is *tree-shaped* if G_q is a directed tree.

We start by introducing a notion of *forks* and *splittings* from [18].

Forks. For a conjunctive query q we say that a conjunctive query q' *is obtained from* q *by fork elimination*, if q' is obtained from q by selecting two atoms $r(y, x)$ and $s(x, z)$ and identifying variables y and z. A query q_{fr} is a *fork rewriting* of q if q_{fr} is obtained from q by applying fork elimination (possibly multiple times). A *maximal fork rewriting* fork rewriting of q is a query q_{mfr} obtained by exhaustively application of fork elimination. It is known from [18] that maximal fork rewriting is unique (up to variable renaming), thus we speak about *the* maximal fork rewriting.

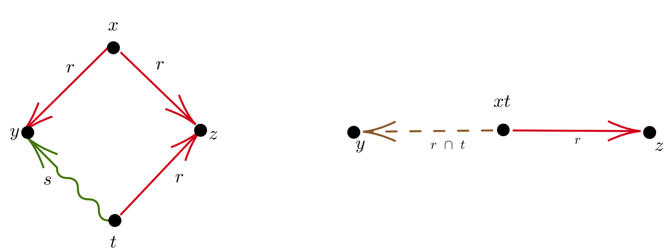


Figure 3: A query $q = r(x, y) \wedge r(x, z) \wedge r(t, z) \wedge s(t, y)$ (left) and its fork-rewriting (right) obtained by identifying variables x and t.

Splittings. The next definition speaks about the abstract way how a conjunctive query can match a model, without making reference to a concrete model nor a concrete match.

[9] Note that $\mathcal{ALCH}^{\cap}$ is a sub-logic of $\mathcal{ALCSCC}$.

Let $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ be a normalized $\mathcal{ALCSCC}$ knowledge base composed of an Abox $\mathcal{A}$, Tbox $\mathcal{T}$ and an ERCBox $\mathcal{R}$. A *splitting* of a conjunctive query q w.r.t $\mathcal{K}$ is a tuple

$$\Pi = (R, T, S_1, S_2, \ldots, S_n, \mu, \nu),$$

where the sets R, T, S_i induce a partition of the set $\mathrm{Var}(q)$, the function $\mu : \{1, 2, \ldots, n\} \to R$ assigns to each set S_i a variable $\mu(i) \in R$, and the function $\nu : R \to \mathsf{Ind}_{\mathcal{A}}$ assigns to each variable from R a named individual from $\mathcal{A}$. A splitting Π has to satisfy the following conditions:[10]

- the query $q\big|_T$ is a variable disjoint union of tree-shaped queries,
- queries $q\big|_{S_i}$ for all $i \in \{1, 2, \ldots, n\}$ are tree-shaped,
- for any atom $r(x, y) \in q$ the variables x, y either belong to the same set $R, T, S_1, S_2, \ldots, S_n$ or $x \in R, y \in S_i$ with x being the root of a tree $q\big|_{S_i}$, and
- for any $i \in \{1, 2, \ldots, n\}$ there is an atom $r(\mu(i), x_0) \in q$ with x_0 the root of $q\big|_{S_i}$.

It might be easier to think that a splitting Π actually consists of "roots" R (corresponding to the Abox part of the model) named by the function ν), together with their "subtrees" S_i and of some arbitrary trees T somewhere far in a model.

Rolling up concepts. We employ a known technique [11, 18, 12] of *rolling-up a tree-shaped query into a concept.* For a given conjunctive query q we define an $\mathcal{ALCH}^{\cap}$ concept $C_{q,x}$ (for each variable $x \in \mathrm{Var}(q)$) as follows. If x is a leaf in G_q then

$$C_{q,x} = \bigsqcap_{C(x) \in q} C.$$

Otherwise we set

$$C_{q,x} = \bigsqcap_{C(x) \in q} C \sqcap \bigsqcap_{(x,y) \in E_q} \exists (\bigcap_{s(x,y) \in q} s).C_{q,y}.$$

The forthcoming lemma links together all presented notions.

Definition 50. *Let q be a conjunctive query and let $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ be a (consistent) normalized $\mathcal{ALCSCC}$ knowledge base with a model $\mathcal{I}$. We say that a pair (q_{fr}, Π), composed of a fork rewriting q_{fr} of q and a splitting $\Pi = (R, T, S_1, S_2, \ldots, S_n, \mu, \nu)$ w.r.t $\mathcal{K}$,* is compatible with *$\mathcal{I}$, if:*

- *for each disconnected component $\widehat{q}$ of T, there is an element $d \in \Delta^{\mathcal{I}}$ with $d \in (C_{\widehat{q}})^{\mathcal{I}}$,*
- *if $C(x) \in q_{\mathsf{fr}}$ with $x \in R$, then $\nu(x)^{\mathcal{I}} \in C^{\mathcal{I}}$,*
- *if $r(x, y) \in q_{\mathsf{fr}}$ with $x, y \in R$, then $(\nu(x)^{\mathcal{I}}, \nu(y)^{\mathcal{I}}) \in r^{\mathcal{I}}$, and*
- *for all $1 \leq i \leq n$ we have (for x_0 being the root of $q_{\mathsf{fr}}\big|_{S_i}$):*

$$\nu(\mu(i))^{\mathcal{I}} \in \left(\exists \left(\bigcap_{s(\mu(i), x_0) \in q_{\mathsf{fr}}} s \right) . C_{q_{\mathsf{fr}}\big|_{S_i}, x_0} \right)^{\mathcal{I}}$$

[10] With $q\big|_X$ we denote the restriction of a query to the set of variables X

Lemma 51. *Take q and $\mathcal{K}$ as stated in Definition 50 and let $\mathcal{I}$ be any $(|q|+1)$–acyclic model of $\mathcal{K}$. Then $\mathcal{I} \models q$ if and only if there exists a pair (q_{fr}, Π) of a fork rewriting and splitting such that (q_{fr}, Π) is compatible with $\mathcal{I}$.*

Proof. Let $\mathcal{I}^{\rightarrow}$ be the forward-unraveling of $\mathcal{I}$. A similar lemma was proven in [18] and its proof without any changes at all can be seen as a proof that $\mathcal{I}^{\rightarrow} \models q$ iff $\mathcal{I}^{\rightarrow}$ is compatible with some (q_{fr}, Π).

Hence if $\mathcal{I}^{\rightarrow}$ is compatible with some (q_{fr}, Π) we can infer that $\mathcal{I}^{\rightarrow} \models q$ holds and by Corollary 49 we conclude that $\mathcal{I} \models q$. For the opposite way, assume that $\mathcal{I}^{\rightarrow} \models q$ holds. Thus $\mathcal{I}^{\rightarrow}$ is compatible with some (q_{fr}, Π). The construction of forward-unravelings is concept preserving (see e.g. the proof of Lemma 33), thus the first and the last item of Definition 50 are satisfied by $\mathcal{I}$. To conclude the satisfaction of the second and the third items of Definition 50 it is enough to see that forward-unravelings preserve Aboxes (namely Lemma 30). Hence $\mathcal{I}$ is compatible with (q_{fr}, Π). ☐

Spoilers and super-spoilers. Let $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ be normalized $\mathcal{ALCSCC}$ knowledge base, let q be a conjunctive query and let $\Pi = (R, T, S_1, S_2, \ldots, S_n, \mu, \nu)$ be a splitting of q w.r.t $\mathcal{K}$. Moreover, let $q_1, \ldots, q_n$ be the tree-shaped disconnected components of $q\big|_T$ with roots $x_1, \ldots, x_n$.

We say that $\mathcal{ALCH}^{\cap}$ knowledge base $\mathcal{K}_s = (\mathcal{A}_s, \mathcal{T}_s)$ is a *spoiler* for q, $\mathcal{K}$ and Π if one of the following conditions hold:

- $\left(\top \sqsubseteq \neg C_{q_i, x_i}\right) \in \mathcal{T}_s$, for some $1 \leq i \leq k$,
- there is an atom $C(x) \in q$ with $q \in R$ but $\neg C(\nu(x)) \in \mathcal{A}_s$
- there is an atom $r(x, y) \in q$ with $x, y \in R$ but $\neg r(\nu(x), \nu(y)) \in \mathcal{A}_s$
- $\neg D(\nu(\mu(i))) \in \mathcal{A}_s$ for some $1 \leq i \leq n$, where (for x_0 being the root of $q\big|_{S_i}$):

$$D = \left(\exists \left(\bigcap_{(\mu(i), x_0) \in q} s\right) . C_{q\big|_{S_i}}\right)^{\mathcal{I}}$$

A *super-spoiler* for q and $\mathcal{K}$ is a *minimal* $\mathcal{ALCH}^{\cap}$ knowledge base $\mathcal{K}_s = (\mathcal{A}_s, \mathcal{T}_s)$ such that for any splitting Π of q w.r.t $\mathcal{K}$, the knowledge base $\mathcal{K}_s$ is a spoiler for q, $\mathcal{K}$ and Π.

The following lemma describes the purpose of spoilers:

Lemma 52. *Let $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ be a normalized $\mathcal{ALCSCC}$ knowledge base and let q be a conjunctive query. The query $\mathcal{K} \not\models q$ if and only if there exists a super-spoiler $\mathcal{K}_s = (\mathcal{A}_s, \mathcal{T}_s)$ such that the knowledge base $(\mathcal{A} \cup \mathcal{A}_s, \mathcal{T} \cup \mathcal{T}_s, \mathcal{R})$ is consistent.*

Proof. Note that a similar Lemma was proven in [18] for infinite tree-shaped models. Its proof can be read without any changes as a proof of the following statement: for all unravelings $\mathcal{I}^{\rightarrow}$ the condition $\mathcal{I}^{\rightarrow} \not\models q$ holds iff $(\mathcal{A} \cup \mathcal{A}_s, \mathcal{T} \cup \mathcal{T}_s, \mathcal{R})$ is consistent for some super-spoiler $\mathcal{K}_s = (\mathcal{A}_s, \mathcal{T}_s)$.

If $\mathcal{K} \not\models q$ then (from Lemma 48) there exists a $(|q|+1)$–acyclic counter-model $\mathcal{I}$ for q, i.e., a model $\mathcal{I}$ satisfying $\mathcal{I} \not\models q$. Then also $\mathcal{I}^{\rightarrow} \not\models q$ (follows from Corollary 49). From [18] we infer that there exists a super-spoiler $\mathcal{K}_s = (\mathcal{A}_s, \mathcal{T}_s)$ for $\mathcal{I}^{\rightarrow}$. Since $\mathcal{I}^{\rightarrow}$ and $\mathcal{I}$ satisfy the same $\mathcal{ALCSCC}$ formulae, we conclude that $(\mathcal{A} \cup \mathcal{A}_s, \mathcal{T} \cup \mathcal{T}_s, \mathcal{R})$ is consistent.

For the opposite way assume that there exists a super-spoiler $\mathcal{K}_s = (\mathcal{A}_s, \mathcal{T}_s)$ such that $\mathcal{K}' = (\mathcal{A} \cup \mathcal{A}_s, \mathcal{T} \cup \mathcal{T}_s, \mathcal{R})$ is consistent. Then there is a $(|q|+1)$–acyclic model $\mathcal{I}$ of $\mathcal{K}'$. Aiming for contradiction assume that $\mathcal{K} \models q$. Hence there is a query match in $\mathcal{I}$ and from Corollary 49 we also know that $\mathcal{I}^{\rightarrow} \models q$. But it contradicts the Lutz's Lemma [18] for infinite tree-shaped models. Hence, $\mathcal{I}^{\rightarrow} \not\models q$. Thus $\mathcal{I} \not\models q$ which clearly implies that $\mathcal{K} \not\models q$. ☐

The last ingredient for designing an exponential time algorithm for deciding query entailment is to estimate the number of super-spoilers as well as their size. By showing that one can restrict attention only to trees being subtrees of a maximal fork rewriting, Lutz [18] have shown that (independently of the underlying DL formalism) the following lemma holds:

Lemma 53 ([18])**.** *Let $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ be a normalized $\mathcal{ALCSCC}$ knowledge base and let q be a conjunctive query. Then the total number of super-spoilers for $\mathcal{K}$ and q is only exponential in $(|q| + |\mathcal{K}|)$ and the size of each super-spoiler is only polynomial in $(|q| + |\mathcal{K}|)$. Moreover the set of super-spoilers can be enumerated in exponential time.*

Proof. Immediate conclusion from Lemma 4, Lemma 5 and Lemma 6 from [18]. □

The algorithm for deciding conjunctive query entailment for $\mathcal{ALCSCC}$ knowledge bases $\mathcal{K} = (\mathcal{A}, \mathcal{T})$ w.r.t Aboxes, Tboxes and ERCBoxes is quite simple. We enumerate all super-spoilers $\mathcal{K}_s = (\mathcal{A}_s, \mathcal{T}_s)$ (from Lemma 53 we know that there are only exponentially many of them and the enumeration process can be done in exponential time) and run a satisfiability test for $\mathcal{K}' = (\mathcal{A} \cup \mathcal{A}_s, \mathcal{T} \cup \mathcal{T}_s, \mathcal{R})$ by employing an algorithm described in Theorem 20. Since the size of $\mathcal{K}_s$ is only polynomial in $(|q| + |\mathcal{K}|)$ then the size of $\mathcal{K}'$ is also only polynomial in $(|q| + |\mathcal{K}|)$. Hence the satisfiability check can be done in *ExpTime* (by Theorem 20 again). We return the answer that q is not entailed by $\mathcal{K}$ if $\mathcal{K}'$ is satisfiable for some super-spoiler and that the query is entailed otherwise. Correctness of the procedure is guaranteed by Lemma 52. Hence we obtain:

Theorem 54. *Conjunctive query entailment from $\mathcal{ALCSCC}$ ERCBoxes wrt. $\mathcal{ALCSCC}$ ABoxes is ExpTime-complete.*

Moreover, since $\mathcal{ALCHQ}$ is a sublogic of $\mathcal{ALCSCC}$ (in a sense that for every $\mathcal{ALCHQ}$ concept we find an equisatisfiable $\mathcal{ALCSCC}$ concept), as a corollary we obtain the first known exponential time algorithm for deciding finite query entailment over $\mathcal{ALCHQ}$ knowledge bases.

Corollary 55. *Conjunctive query entailment from $\mathcal{ALCHQ}$ TBoxes wrt. $\mathcal{ALCHQ}$ ABoxes is ExpTime-complete.*

The *ExpTime* lower bounds comes already from $\mathcal{ALC}$ concept satisfiability w.r.t TBoxes.

8 Conclusion

We have introduced the DL $\mathcal{ALCSCC}^{++}$, which allows for mixing local and global cardinality constraints. Though being considerably more expressive than previously investigated DLs with cardinality constraints, reasoning in $\mathcal{ALCSCC}^{++}$ has turned out to be not harder that reasoning in $\mathcal{ALC}$ with very simple cardinality restrictions. However, extending $\mathcal{ALCSCC}^{++}$ with inverse roles causes undecidability for the standard inference satisfiability, as does considering the non-standard inference of query entailment in $\mathcal{ALCSCC}^{++}$. We were able to show that decidability of query entailment can be regained by considering restricted cardinality constraints (ERCBoxes) in the sub-logic $\mathcal{ALCSCC}$ of $\mathcal{ALCSCC}^{++}$. The ExpTime upper bound proved for this task depends on the ExpTime upper bound for ABox consistency in $\mathcal{ALCSCC}$ w.r.t. ERCBoxes shown for the first time in the present paper.

Some of the results presented here have already been sketched in a paper at the DL workshop [4]. However, there the positive result for query entailment was restricted to a setting without ABox since we did not yet have the result for ABox consistency, and only a 2ExpTime upper bound for the complexity was shown. In addition, the undecidability result for $\mathcal{ALCISCC}^{++}$ is also not contained in [4].

Regarding future work, it would be interesting to investigate the impact that adding inverse roles has on reasoning in $\mathcal{ALCSCC}$ w.r.t. different kinds of terminological boxes (TBox, ERCBox, ECBox), though this will probably be a very hard task. From an application point of view, as a first step towards a more practical query answering algorithm, we intend to investigate the ABox consistency problem in $\mathcal{ALCSCC}$ w.r.t. ERCBoxes. Since type elimination algorithms are not only worst-case, but also best-case exponential, we will try to devise a tableau-based algorithm for this problem, which may use numerical algorithms and satisfiability checkers for QFBAPA as sub-procedures.

Acknowledgements

Franz Baader was partially supported by the German Research Foundation (DFG) within the Research Unit 1513 Hybris and grant 389792660 as part of TRR 248. Bartosz Bednarczyk was supported by the European Research Council (ERC) through the Consolidator Grant 771779 (DeciGUT) and the Polish Ministry of Science and Higher Education program "Diamentowy Grant" no. DI2017 006447. Sebastian Rudolph was supported by the European Research Council (ERC) through the Consolidator Grant 771779 (DeciGUT).

References

[1] Franz Baader. A new description logic with set constraints and cardinality constraints on role successors. In Clare Dixon and Marcelo Finger, editors, *Proceedings of the 11th International Symposium on Frontiers of Combining Systems (FroCoS'17)*, volume 10483 of *Lecture Notes in Computer Science*, pages 43–59, Brasília, Brazil, 2017. Springer-Verlag.

[2] Franz Baader. Expressive cardinality constraints on $\mathcal{ALCSCC}$ concepts. In *Proceedings of the 34th ACM/SIGAPP Symposium On Applied Computing (SAC'19)*. ACM, 2019.

[3] Franz Baader. Expressive cardinality restrictions on concepts in a description logic with expressive number restrictions. *ACM SIGAPP Applied Computing Review*, 19:5–17, 2019.

[4] Franz Baader, Bartosz Bednarczyk, and Sebastian Rudolph. Satisfiability checking and conjunctive query answering in description logics with global and local cardinality constraints. In Mantas Simkus and Grant E. Weddell, editors, *Proc. of the 2019 Description Logic Workshop (DL 2019)*, volume 2373 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019.

[5] 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, *Proc. of the 12th Int. Symp. on Frontiers of Combining Systems (FroCoS 2019)*, volume 11715 of *Lecture Notes in Computer Science*, pages 203–219. Springer, 2019.

[6] Franz Baader, Martin Buchheit, and Bernhard Hollunder. Cardinality restrictions on concepts. *Artificial Intelligence*, 88(1–2):195–213, 1996.

[7] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[8] Franz Baader and Andreas Ecke. Extending the description logic alc with more expressive cardinality constraints on concepts. In *GCAI 2017. 3rd Global Conference on Artificial Intelligence*, volume 50 of *EPiC Series in Computing*, pages 6–19. EasyChair, 2017.

[9] Franz Baader, Ian Horrocks, Carsten Lutz, and Uli Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017.

[10] Alexander Borgida, Ronald J. Brachman, Deborah L. McGuinness, and Lori Alperin Resnick. CLASSIC: A structural data model for objects. In *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, pages 59–67, 1989.

[11] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. On the decidability of query containment under constraints. In *Proc. of the 17th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS'98)*, pages 149–158, 1998.

[12] Birte Glimm, Carsten Lutz, Ian Horrocks, and Ulrike Sattler. Conjunctive query answering for the description logic SHIQ. *J. of Artificial Intelligence Research*, 31:157–204, 2008.

[13] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988.

[14] Robert Hoehndorf, Paul N. Schofield, and Georgios V. Gkoutos. The role of ontologies in biological and biomedical research: A functional perspective. *Brief. Bioinform.*, 16(6):1069–1080, 2015.

[15] Bernhard Hollunder and Franz Baader. Qualifying number restrictions in concept languages. In *Proc. of the 2nd Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'91)*, pages 335–346, 1991.

[16] Bernhard Hollunder, Werner Nutt, and Manfred Schmidt-Schauß. Subsumption algorithms for concept description languages. In *Proc. of the 9th Eur. Conf. on Artificial Intelligence (ECAI'90)*, pages 348–353, London (United Kingdom), 1990. Pitman.

[17] Viktor Kuncak and Martin C. Rinard. Towards efficient satisfiability checking for Boolean algebra with Presburger arithmetic. In Frank Pfenning, editor, *Proc. of the 21st Int. Conf. on Automated Deduction (CADE-07)*, volume 4603 of *Lecture Notes in Computer Science*, pages 215–230. Springer, 2007.

[18] Carsten Lutz. The complexity of conjunctive query answering in expressive description logics. In Alessandro Armando, Peter Baumgartner, and Gilles Dowek, editors, *Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2008)*, Lecture Notes in Artificial Intelligence, pages 179–193. Springer-Verlag, 2008.

[19] Rafael Peñaloza and Nico Potyka. Towards statistical reasoning in description logics over finite domains. In Serafin Moral and Olivier Pivert, editors, *Proc. of the 11th Int. Conf. on Scalable Uncertainty Management (SUM 2017)*, volume 10564 of *Lecture Notes in Computer Science*. Springer-Verlag, 2017.

[20] Ian Pratt-Hartmann. Data-complexity of the two-variable fragment with counting quantifiers. *Inf. Comput.*, 207(8):867–888, 2009.

[21] "Johann" Sebastian Rudolph. Presburger concept cardinality constraints in very expressive description logics – allegro sexagenarioso ma non ritardando. In Carsten Lutz, Uli Sattler, Cesare Tinelli, Anni-Yasmin Turhan, and Frank Wolter, editors, *Description Logic, Theory Combination, and All That – Essays Dedicated to Franz Baader on the Occasion of His 60th Birthday*, volume 11560 of *Lecture Notes in Computer Science*, pages 542–561. Springer, 2019.

[22] Sebastian Rudolph. Foundations of description logics. In Axel Polleres, Claudia d'Amato, Marcelo Arenas, Siegfried Handschuh, Paula Kroner, Sascha Ossowski, and Peter F. Patel-Schneider, editors, *Reasoning Web. Semantic Technologies for the Web of Data – 7th International Summer School 2011*, volume 6848 of *LNCS*, pages 76–136. Springer, 2011.

[23] Stephan Tobies. A PSPACE algorithm for graded modal logic. In Harald Ganzinger, editor, *Proc. of the 16th Int. Conf. on Automated Deduction (CADE'99)*, volume 1632 of *Lecture Notes in Artificial Intelligence*, pages 52–66. Springer-Verlag, 1999.

[24] Stephan Tobies. The complexity of reasoning with cardinality restrictions and nominals in expressive description logics. *J. of Artificial Intelligence Research*, 12:199–217, 2000.

[25] Stephan Tobies. *Complexity Results and Practical Algorithms for Logics in Knowledge Representation*. PhD thesis, LuFG Theoretical Computer Science, RWTH-Aachen, Germany, 2001.