Temporally Attributed Description Logics

Ana Ozaki[1][0000−0002−3889−6207], Markus Krötzsch[2][0000−0002−9172−2601], and Sebastian Rudolph[2][0000−0002−1609−2080]

[1] KRDB Research Centre, Free University of Bozen-Bolzano, Italy
[2] TU Dresden, Dresden, Germany

Abstract. Knowledge graphs are based on graph models enriched with (sets of) attribute-value pairs, called annotations, attached to vertices and edges. Many application scenarios of knowledge graphs crucially rely on the frequent use of annotations related to *time*. Building upon attributed logics, we design description logics enriched with temporal annotations whose values are interpreted over discrete time. Investigating the complexity of reasoning in this new formalism, it turns out that reasoning in our temporally attributed description logic $\mathcal{ALCH}^{\mathbb{T}}_{@}$ is highly undecidable; thus we establish restrictions where it becomes decidable, and even tractable.

1 Introduction

Graph-based data formats play an essential role in modern information management, since they offer schematic flexibility, ease information re-use, and simplify data integration. Ontological knowledge representation has been shown to offer many benefits to such data-intensive applications, e.g., by supporting integration, querying, error detection, or repair. However, practical *knowledge graphs*, such as Wikidata [38] or YAGO2 [21], are based on *enriched* graphs where edges are augmented with additional annotations.

Example 1. Figure 1 shows an excerpt of the information that Wikidata provides about Franz Baader. Binary relations, such as memberOf(FranzBaader, AcademiaEuropaea), are the main modelling primitive for encoding knowledge. They correspond to labelled directed edges in the graph. However, many of these edges are annotated with additional information, specifying validity times, references (collapsed in the figure), auxiliary details, and other pieces of information that pertain to this binary relationship.

A similar approach to knowledge modelling is followed in the popular *Property Graph* data model [34], and supported by modern graph stores such as Amazon Azure, BlazeGraph, and Neo4j. Other data models allowing attribute-value pairs to be associated with relations are UML, entity-relation and object-role modelling (see, e.g., [10,37,2] for works drawing the connection between these data models and DLs). Predicate logic does not have a corresponding notion of enriched relationships, and established ontology languages that are based on traditional logic are therefore not readily applicable to enriched graphs [22]. To provide better modelling support, *attributed logics* have been proposed as a way of integrating annotations with logical reasoning [32]. This approach has been applied to description logics (DLs) [7] to obtain *attributed DLs* [23,24,12].

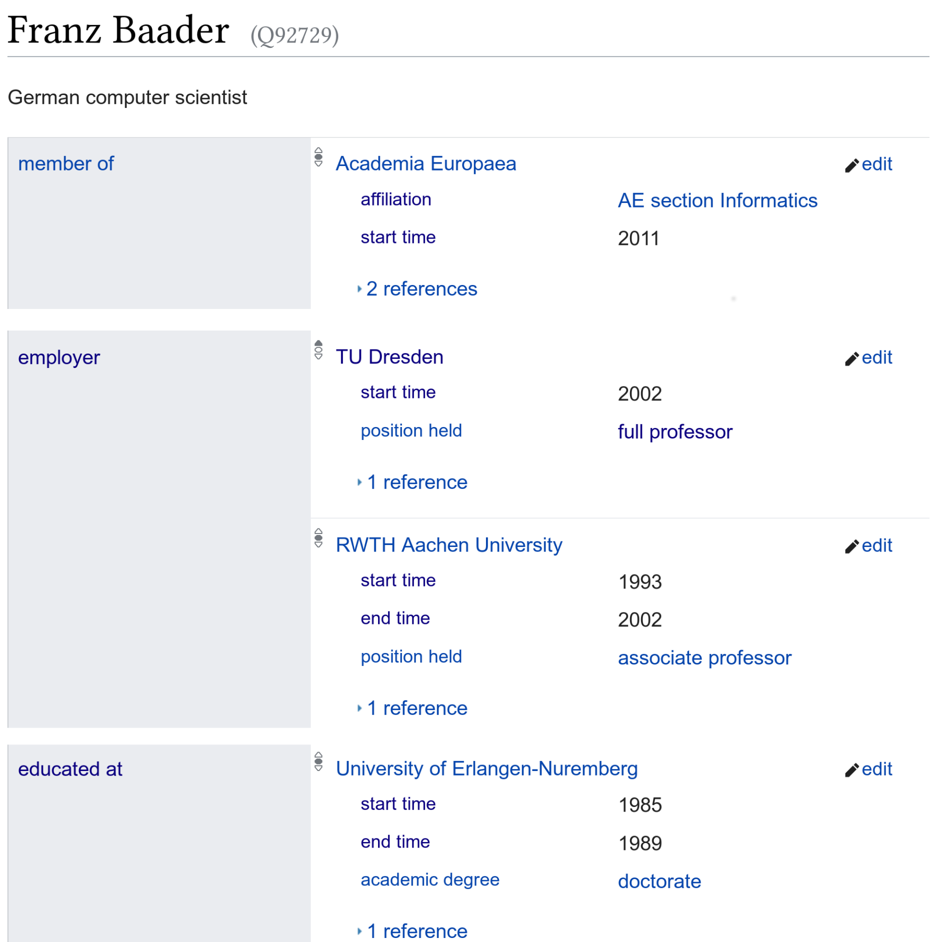

Fig. 1. Excerpt of the Wikidata page of Franz Baader; `https://wikidata.org/wiki/Q92729`

Annotations in practical knowledge graphs have many purposes, such as recording provenance, specifying context, or encoding n-ary relations. One of their most important uses, however, is to encode *temporal validity* of statements. In Wikidata, e.g., *start*/*end time* and *point in time* are among the most frequent annotations, used in 6.7 million statements overall.[3] YAGO2 introduced the SPOTL data format that enriches *subject-property-object* triples (known from RDF) with information on *time* and *location* [21].

Reasoning with time clearly requires an adequate semantics, and many approaches were proposed. Validity time points and intervals are a classical topic in data management [17,18], and similar models of time have also been studied in ontologies [4,26]. However,

[3] As of March 2019, the only more common annotations are *reference* (provenance) and *determination method* (context); see `https://tools.wmflabs.org/sqid/#/browse?type=properties&sortpropertyqualifiers=fa-sort-desc`

researchers in ontologies have most commonly focussed on abstract models of time as used in temporal logics [31,39,8]. Temporal reasoning in $\mathcal{ALC}$ with concrete domains was proposed by Lutz et. al [29]. It is known that satisfiability of $\mathcal{ALC}$ with a concrete domain consisting of a dense domain and containing the predicates = and < is ExpTime-complete [28]. In the same setting but for *discrete time*, the complexity of the satisfiability problem is open, a criterion which only guarantees decidability has been proposed by Carapelle and Turhan [15]. None of these approaches has been considered for attributed logics yet, and indeed support for temporal reasoning for knowledge graphs, such as Wikidata and YAGO2, is still missing today. In this paper, we address this shortcoming by endowing attributed description logics with a temporal semantics for annotations. Indeed, annotations are already well-suited for representing time-related data.

Example 2. We introduce temporally attributed DLs that use special temporal annotation attributes, which can refer to individual time points or to intervals of time. For example, information about Franz Baader's current employment can be expressed by an annotated DL fact as follows:

$$\mathsf{employer}(\mathsf{FranzBaader}, \mathsf{TUD})@[\mathsf{since}: 2002, \mathsf{position}: \mathsf{fullProfessor}] \tag{1}$$

Here, the special temporal attribute since is used alongside the regular attribute position. Likewise, we can express intervals, as in the following axiom[4]

$$\mathsf{educatedAt}(\mathsf{FranzBaader}, \mathsf{FAU})@[\mathsf{during} : [1985, 1989], \mathsf{degree}: \mathsf{doctorate}] \tag{2}$$

Some facts might also be associated with a specific time rather than with a duration. For example, we could encode some of the knowledge in Wikidata with the fact:

$$\mathsf{bornIn}(\mathsf{FranzBaader}, \mathsf{Spalt})@[\mathsf{time}: 1959] \tag{3}$$

Not all people are as thoroughly documented on Wikidata, but attributed DLs also provide ways of leaving some information unspecified, as in the following fact about one of Baader's former doctoral students:

$$\big(\exists \mathsf{bornIn}@[\,\mathsf{between} : [1950, 2000]\,].\top\big)(\mathsf{Carsten}), \tag{4}$$

which merely states that Carsten Lutz was born *somewhere* within the second half of the 20th century.

To deal with such temporally annotated data in a semantically adequate way and to specify temporal background knowledge, we propose the temporally attributed description logic $\mathcal{ALCH}^{\mathbb{T}}_{@}$ that enables reasoning and querying with such information. In addition to the basic support for representing information with attributes, our logic includes a special semantics for temporal attributes, and the support for (safe) variables in DL axioms. Beyond defining syntax and semantics of $\mathcal{ALCH}^{\mathbb{T}}_{@}$, our contributions are the following:

- We show that the full formalism is highly undecidable using an encoding of a recurring tiling problem.

[4] FAU is the official abbreviation for the *Friedrich-Alexander University* in Erlangen/Nuremberg.

$$\exists \mathsf{employer}@\lfloor \mathsf{position}\colon \mathsf{fullProfessor}\rfloor.\top \sqsubseteq \mathsf{Professor} \quad (5)$$

$$X : \lfloor \mathsf{position}\colon \mathsf{fullProfessor}\rfloor \quad (\exists \mathsf{employer}@X.\top \sqsubseteq \mathsf{Professor}) \quad (6)$$

$$X : \lfloor \mathsf{position}\colon \mathsf{fullProfessor}\rfloor \quad (\exists \mathsf{employer}@X.\top \sqsubseteq \mathsf{Professor}@[\mathsf{time}\colon X.\mathsf{time}]) \quad (7)$$

$$\exists \mathsf{employer}@[\mathsf{position}\colon \mathsf{fullProfessor}, \mathsf{time}\colon x].\top \sqsubseteq \mathsf{Professor}@[\mathsf{time}\colon x] \quad (8)$$

Fig. 2. Examples for axioms in attributed description logics

- We present three ways (of increasing reasoning complexity) for regaining decidability: disallowing variables altogether (ExpTime), disallowing the use of variables only for temporal attributes (2ExpTime), or disallowing the use of temporal attributes referencing time points in the future (3ExpTime).
- Finally we single out a lightweight case based on the description logic $\mathcal{EL}$ which features PTime reasoning.

2 Temporally Attributed DLs

We first present the syntax and underlying intuition of temporally attributed description logics. In DL, a true fact corresponds to the membership of an element in a class, or of a pair of elements in a binary relation. Attributed DLs further allow each true fact to carry a finite set of annotations [23], given as attribute-value pairs. As suggested in Example 2, the same relationship may be true with several different annotation sets, e.g., to capture that Baader has been educated at FAU Erlangen-Nuremberg during two intervals: once for his PhD and once for his *Diplom* (not shown in Fig. 1).

Example 3. To guide the reader in following the formal definitions, we first illustrate the main features of attributed DL by means of some example axioms, shown in Fig. 2. We already use time as an example annotation, but do not yet rely on any specific semantic interpretation for this attribute.

The (non-temporal) attributed DL axiom (5) states that people employed as full professors are professors. The *open specifier* $\lfloor \mathsf{position}\colon \mathsf{fullProfessor}\rfloor$ requires that the given attribute is among the annotations, but allows other annotations to be there as well (denoted by the half-open brackets). Axiom (6) is equivalent to (5), but assigns the annotation set to a *set variable* X.

If the employer relation specifies a validity time, the same time would apply to Professor. This is accomplished by axiom (7), which uses the expression $\mathsf{time}\colon X.\mathsf{time}$ to declare that *all* (zero or more) time values of X should be copied. The closed brackets in the conclusion specify that no further attribute-value pairs may occur in the annotation of the conclusion.

A subtly different meaning is captured by (8), which uses an *object variable* x as a placeholder for a single attribute value. In contrast to (8), axiom (7) (i) requires that at least one time annotation is present (rather than allowing zero or more), (ii) requires that the annotation set in the premise has exactly two attribute-value pairs (rather than being open for more), and (iii) infers distinct Professor assertions for each time x (rather

$$\exists \mathsf{educatedAt}@\lfloor \mathsf{before}\colon x \rfloor.\top \sqsubseteq \neg \exists \mathsf{bornIn}@\lfloor \mathsf{time}\colon x \rfloor.\top \tag{9}$$

$$\exists \mathsf{educatedAt}@\lfloor \mathsf{time}\colon x \rfloor.\top \sqsubseteq \neg \exists \mathsf{bornIn}@\lfloor \mathsf{after}\colon x \rfloor.\top \tag{10}$$

$$\exists \mathsf{educatedAt}@\lfloor \mathsf{time}\colon x \rfloor.\top \sqsubseteq \exists \mathsf{educatedAt}@\lfloor \mathsf{during} : [x, x] \rfloor.\top \tag{11}$$

$$\exists \mathsf{bornIn}@\lfloor \mathsf{between} : [x, y] \rfloor.\top \sqsubseteq \neg \exists \mathsf{educatedAt}@\lfloor \mathsf{before}\colon x \rfloor.\top \tag{12}$$

Fig. 3. Examples for axioms in temporally attributed description logics

than one assertion that copies all time points). Item (iii) deserves some reflection. As argued above, it is meaningful that the same fact holds true with different annotation sets, and this does not imply that it is also true with the union of these annotations. However, in the case of time, our intuition is that something is true at several times individually exactly if it is true at all of these times together. Our formal semantics will ensure that this equivalence holds.

We define our description logic $\mathcal{ALCH}^{\mathbb{T}}_{@}$ as a multi-sorted version of the attributed DL $\mathcal{ALCH}_{@}$, thereby introducing datatypes for time points and intervals. Elements of the different types are represented by members of mutually disjoint sets of *(abstract) individual names* $\mathsf{N_I}$, *time points* $\mathsf{N_T}$, and *time intervals* $\mathsf{N_T^2}$. We represent time points by natural numbers, and assume that elements of $\mathsf{N_T}$ ($\mathsf{N_T^2}$) are (pairs of) numbers in *binary* encoding. We write $[k, \ell]$ for a pair of numbers k, ℓ in $\mathsf{N_T^2}$. Moreover, we require that there are the following seven special individual names, called *temporal attributes*: $\mathsf{time}, \mathsf{before}, \mathsf{after}, \mathsf{until}, \mathsf{since}, \mathsf{during}, \mathsf{between} \in \mathsf{N_I}$.

The intuitive meaning of temporal attributes is as one might expect: time describes individual times at which a statement is true, while the others describe (half-open) intervals. The meaning of before, after, and between is existential in that they require the statement to hold only at some time in the interval, while until, since, and during are universal and require something to be true throughout an interval.

Example 4. The examples in Fig. 3 illustrate the special semantics of temporal attributes. Axiom (9) states that nobody can be educated before being born. Axiom (10) is equivalent. In particular, our semantics ensures that temporal attributes like time, before, and after will be inferred even when not stated explicitly. For example, (11) is a tautology. Longer intervals of during can be inferred for any span of consecutive time points (our time model is discrete). Finally, we also allow using object variables in time intervals, as illustrated in (12), which is actually *equivalent* to (9) as well.

With these examples in mind, we continue to define the syntax of our temporal DLs formally. Axioms of $\mathcal{ALCH}^{\mathbb{T}}_{@}$ are further based on sets of *concept names* $\mathsf{N_C}$, and *role names* $\mathsf{N_R}$. Attributes are represented by individual names, and we associate a *value type* $\mathsf{valtype}(a)$ with each individual $a \in \mathsf{N_I}$ for this purpose: during and between have value type $\mathsf{N_T^2}$, all other temporal attributes have value type $\mathsf{N_T}$, and all other individuals have value type $\mathsf{N_I}$. An *attribute-value pair* is an expression $a\colon v$ where $a \in \mathsf{N_I}$ and $v \in \mathsf{valtype}(a)$. Now concept and role assertions of $\mathcal{ALCH}^{\mathbb{T}}_{@}$ have the following form, respectively:

$$C(a)@[a_1 : v_1, \ldots, a_n : v_n] \tag{13}$$
$$r(a,b)@[a_1 : v_1, \ldots, a_n : v_n] \tag{14}$$

where $C \in \mathsf{N_C}$, $r \in \mathsf{N_R}$, $a, b \in \mathsf{N_I}$, and $a_i : v_i$ are attribute-value pairs. Note that (4) in Example 2 is not a concept assertion in the sense of (13), since it uses a complex concept expression. As usual in DLs, our language will allow us to encode such complex assertions by giving them a new name in a terminological axiom.

Role and concept inclusion axioms of $\mathcal{ALCH}^{\mathbb{T}}_{@}$ introduce additional expressive power to refer to partially specified and variable annotation sets. Attribute values may now also contain *object variables* taken from pairwise disjoint sets $\mathsf{Var}(\mathsf{N_I})$, $\mathsf{Var}(\mathsf{N_T})$, and $\mathsf{Var}(\mathsf{N_T^2})$. Moreover, whole annotation sets might be represented by *set variables* from a set $\mathsf{N_V}$.

Definition 1. *An* (annotation set) specifier *can be a set variable $X \in \mathsf{N_V}$, a* closed specifier *of the form $[a_1 : v_1, \ldots, a_n : v_n]$, or an* open specifier *of the form $\lfloor a_1 : v_1, \ldots, a_n : v_n \rfloor$, where $n \geq 0$, $a_i \in \mathsf{N_I}$ and each v_i is an expression that is compatible with the value type of its attribute in the sense that it has one of the following forms:*

- $v_i \in \mathsf{valtype}(a_i) \cup \mathsf{Var}(\mathsf{valtype}(a_i))$, *or*
- $v_i = [v, w]$ *with* $\mathsf{valtype}(a_i) = \mathsf{N_T^2}$ *and* v, w *in* $\mathsf{N_T} \cup \mathsf{Var}(\mathsf{N_T})$, *or*
- $v_i = X.b$ *with* $X \in \mathsf{N_V}$, $b \in \mathsf{N_I}$, *and* $\mathsf{valtype}(a_i) = \mathsf{valtype}(b)$.

The set of all specifiers is denoted **S**. *A specifier is* ground *if it does not contain variables.*

Intuitively, closed specifiers define specific annotation sets whereas open specifiers provide lower bounds [23]. Object variables are used to copy values from one attribute to another, as long as the attributes have the same value type (in the same annotation set or in a new one); the expression $X.b$ is used to copy all of the zero or more b-values of annotation set X. We also allow specifiers to be empty. That is, we allow $\lfloor\rfloor$ (meaning "any annotation set") and $[]$ (meaning "the empty annotation set"). To simplify notation, we may omit $@\lfloor\rfloor$ and $@[]$ in role or concept expressions (and $@[]$ in assertions).

Definition 2. $\mathcal{ALCH}^{\mathbb{T}}_{@}$ role expressions *have the form $r@S$ with $r \in \mathsf{N_R}$ and $S \in \mathbf{S}$.* $\mathcal{ALCH}^{\mathbb{T}}_{@}$ concept expressions *C, D are defined recursively:*

$$C, D ::= \top \mid A@S \mid \neg C \mid (C \sqcap D) \mid \exists R.C \tag{15}$$

with $A \in \mathsf{N_C}$, $S \in \mathbf{S}$ and R an $\mathcal{ALCH}^{\mathbb{T}}_{@}$ role expression.

We use abbreviations $(C \sqcup D)$, $\bot$, and $\forall R.C$ for $\neg(\neg C \sqcap \neg D)$, $\neg\top$, and $\neg(\exists R.\neg C)$, respectively. $\mathcal{ALCH}^{\mathbb{T}}_{@}$ axioms are essentially just (role/concept) inclusions between $\mathcal{ALCH}^{\mathbb{T}}_{@}$ role and concept expressions, which may, however, share variables.

Example 5. Object variables can be used to create new intervals of time using the temporal information present on the annotations. In the following example, we illustrate a concept inclusion that allows for inferring the (minimum) period in which a person typically is a PhD student:

$$\begin{aligned}\exists \mathsf{obtainedMSc}@\lfloor \mathsf{between} : [x, x'] \rfloor.\top \sqcap \exists \mathsf{obtainedPhD}@\lfloor \mathsf{between} : [y, y'] \rfloor.\top \\ \sqsubseteq \mathsf{PhDStudent}@\lfloor \mathsf{during} : [x', y] \rfloor\end{aligned} \tag{16}$$

It is sometimes useful to represent annotations by variables while also specifying some further constraints on their possible values. This can be accommodated by adding such constraints as (optional) prefixes to axioms.

Definition 3. *An* $\mathcal{ALCH}^{\mathbb{T}}_{@}$ concept inclusion *is an expression of the form*

$$X_1 : S_1, \ldots, X_n : S_n \quad (C \sqsubseteq D), \tag{17}$$

where C, D *are* $\mathcal{ALCH}^{\mathbb{T}}_{@}$ *concept expressions,* $S_1, \ldots, S_n \in \mathbf{S}$ *are closed or open specifiers, and* $X_1, \ldots, X_n \in \mathsf{N_V}$ *are set variables occurring in* C, D *or in* $S_1, \ldots, S_n$*. We require that all variables are* safe *in the following sense:*

(1) every set variable in the axiom also occurs in the left concept C*,*[5] *and*
(2) every object variable in the axiom also occurs either in the left concept C *or in a specifier* S_i *in a prefix* $X_i : S_i$*.*

$\mathcal{ALCH}^{\mathbb{T}}_{@}$ role inclusions *are defined analogously, but with role expressions instead of the concept expressions. An* $\mathcal{ALCH}^{\mathbb{T}}_{@}$ ontology *is a set of* $\mathcal{ALCH}^{\mathbb{T}}_{@}$ *assertions, and role and concept inclusions.*

Note that any $\mathcal{ALCH}$ axiom is also an $\mathcal{ALCH}^{\mathbb{T}}_{@}$ axiom in the sense that the absence of explicit annotations can be considered to mean "$@\lfloor\rfloor$."

3 Semantics of Temporally Attributed DLs

We first recall the general semantics of attributed DLs without temporal attributes. The semantics of $\mathcal{ALCH}^{\mathbb{T}}_{@}$ can then be obtained as a multi-sorted extension that imposes additional restrictions on the interpretation of time.

An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of attributed logic consists of a non-empty domain $\Delta^{\mathcal{I}}$ and a function $\cdot^{\mathcal{I}}$. Individual names $a \in \mathsf{N_I}$ are interpreted as elements $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. To interpret annotation sets, we use the set $\Phi^{\mathcal{I}}$ of all finite binary relations over $\Delta^{\mathcal{I}}$. Each concept name $C \in \mathsf{N_C}$ is interpreted as a set $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Phi^{\mathcal{I}}$ of elements with annotations, and each role name $r \in \mathsf{N_R}$ is interpreted as a set $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \times \Phi^{\mathcal{I}}$ of pairs of elements with annotations. Each element (pair of elements) may appear with multiple annotations [23].

Note that attributes are represented by domain elements in this semantics. This has no actual impact on reasoning in the context of this paper, and could be changed to use a separate *sort* for attributes or to consider them as a kind of predicate that is part of a fixed schema. While this detail is immaterial to our proofs, it is worth noting that attributes are also treated as special kinds of domain objects in important practical knowledge graphs. Both RDF-based models and Wikidata use (technically different) notions of *property* that are part of the domain and can therefore be described by facts. This ability is frequently used in practice to store annotations, to declare constraints, or to establish

[5] This is a simplification from previous works [24] where set variables were allowed to occur in the specifier prefix only under some circumstances. It is not hard to see that our simplification does not relinquish relevant expressivity if we permit some normalisation.

mappings to external vocabularies. We believe that in particular constraint information and mappings in datasets should be accessible to ontological reasoning. In contrast, the Property Graph data model represents attributes as *property keys* (plain strings) that cannot be used as objects (vertices) in the graph [33]. However, in this model, attribute values (*property values*) cannot refer to objects in the graph either. We do not consider it desirable to impose those restrictions, since our more general model can capture more real-world graphs, and is useful for expressing many natural statements (e.g., values like *full professor* in Fig. 1 refer to domain objects, which Property Graph would not allow).

3.1 Time-Sorted Interpretations

To deal with time, we define interpretation that include temporal sorts in addition to the usual abstract domain.

Definition 4. *A* time-sorted interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ *is an interpretation with a domain* $\Delta^{\mathcal{I}}$ *that is a disjoint union of* $\Delta^{\mathcal{I}}_I \cup \Delta^{\mathcal{I}}_T \cup \Delta^{\mathcal{I}}_{2T}$*, where* $\Delta^{\mathcal{I}}_I$ *is the* abstract domain*,* $\Delta^{\mathcal{I}}_T$ *is a finite or infinite interval,*[6] *called* temporal domain*, and* $\Delta^{\mathcal{I}}_{2T} = \Delta^{\mathcal{I}}_T \times \Delta^{\mathcal{I}}_T$*.*

We interpret individual names $a \in \mathsf{N_I}$ *as elements* $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}_I$*; time points* $t \in \mathsf{N_T}$ *as* $t^{\mathcal{I}} \in \Delta^{\mathcal{I}}_T$*; and intervals* $[t, t'] \in \mathsf{N_T^2}$ *as* $[t, t']^{\mathcal{I}} = (t^{\mathcal{I}}, t'^{\mathcal{I}}) \in \Delta^{\mathcal{I}}_{2T}$*. A pair* $(\delta, \epsilon) \in \Delta^{\mathcal{I}}_I \times \Delta^{\mathcal{I}}$ *is* well-typed*, if one of the following holds:*

(a) $\delta = a^{\mathcal{I}}$ *for an attribute* a *of value type* $\mathsf{N_T}$ *and* $\epsilon \in \Delta^{\mathcal{I}}_T$*; or*

(b) $\delta = a^{\mathcal{I}}$ *for an attribute* a *of value type* $\mathsf{N_T^2}$ *and* $\epsilon \in \Delta^{\mathcal{I}}_{2T}$*; or*

(c) $\delta = a^{\mathcal{I}}$ *for an attribute* a *of value type* $\mathsf{N_I}$ *and* $\epsilon \in \Delta^{\mathcal{I}}_I$*.*

Let $\Phi^{\mathcal{I}}$ *be the set of all finite sets of well-typed pairs. The function* $\cdot^{\mathcal{I}}$ *maps concept names* $C \in \mathsf{N_C}$ *to* $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Phi^{\mathcal{I}}$ *and role names* $r \in \mathsf{N_R}$ *to* $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \times \Phi^{\mathcal{I}}$*.*

Note that $\Delta^{\mathcal{I}}_T$ can be finite if $\mathsf{N_T}$ and $\mathsf{N_T^2}$ are (which is always admissible, since any ontology mentions only finitely many time points). $\mathcal{I}$ *satisfies* a concept assertion $C(a)@[a_1\colon v_1, \ldots, a_n\colon v_n]$ if $(a^{\mathcal{I}}, \{(a_1^{\mathcal{I}}, v_1^{\mathcal{I}}), \ldots, (a_n^{\mathcal{I}}, v_n^{\mathcal{I}})\}) \in C^{\mathcal{I}}$, and likewise for role assertions. For interpreting expressions with (object or set) variables, we need a notion of variable assignment.

Definition 5 (semantics of terms). *A* variable assignment *for a time-sorted interpretation* $\mathcal{I}$ *is a function* $\mathcal{Z}$ *that maps set variables* $X \in \mathsf{N_V}$ *to finite binary relations* $\mathcal{Z}(X) \in \Phi^{\mathcal{I}}$*, and object variables* $x \in \mathsf{Var}(\mathsf{N_I}) \cup \mathsf{Var}(\mathsf{N_T}) \cup \mathsf{Var}(\mathsf{N_T^2})$ *to elements* $\mathcal{Z}(x) \in \Delta^{\mathcal{I}}_I \cup \Delta^{\mathcal{I}}_T \cup \Delta^{\mathcal{I}}_{2T}$ *(respecting their types). For (set or object) variables* x*, let* $x^{\mathcal{I},\mathcal{Z}} := \mathcal{Z}(x)$*, and for abstract individuals, time points, or time intervals* a*, let* $a^{\mathcal{I},\mathcal{Z}} := a^{\mathcal{I}}$*.*

Intuitively, each specifiers defines a set of annotation sets. For closed specifiers, there is just one such set (corresponding exactly to the specified attribute-value pairs), whereas for open specifiers, we obtain many sets (namely all supersets of the set that was specified). The following definition is making this formal, and also defines the semantics for all types of expressions that may occur in the value position of attributes within specifiers.

[6] As usual for the natural numbers, a finite interval $[k, \ell]$ is $\{n \in \mathbb{N} \mid k \leq n \leq \ell\}$ and an infinite interval $[k, \infty)$ is $\{n \in \mathbb{N} \mid k \leq n\}$.

Definition 6 (semantics of specifiers). *A specifier $S \in \mathbf{S}$ is interpreted as a set $S^{\mathcal{I},\mathcal{Z}} \subseteq \Phi^{\mathcal{I}}$ of matching annotation sets. We set $X^{\mathcal{I},\mathcal{Z}} := \{\mathcal{Z}(X)\}$ for variables $X \in \mathsf{N_V}$. The semantics of closed specifiers is defined as follows:*

(i) $[a\colon v]^{\mathcal{I},\mathcal{Z}} := \{\{(a^{\mathcal{I}}, v^{\mathcal{I},\mathcal{Z}})\}\}$*, with* $v \in \mathsf{valtype}(a) \cup \mathsf{Var}(\mathsf{valtype}(a))$*;*

(ii) $[a\colon [v,w]]^{\mathcal{I},\mathcal{Z}} := \{\{(a^{\mathcal{I}}, (v^{\mathcal{I},\mathcal{Z}}, w^{\mathcal{I},\mathcal{Z}}))\}\}$*, with* $\mathsf{valtype}(a) = \mathsf{N_T^2}$*, and* $v, w \in \mathsf{N_T} \cup \mathsf{Var}(\mathsf{N_T})$*;*

(iii) $[a\colon X.b]^{\mathcal{I},\mathcal{Z}} := \{\{(a^{\mathcal{I}}, \delta) \mid (b^{\mathcal{I}}, \delta) \in \mathcal{Z}(X)\}\}$*;*

(iv) $[a_1\colon v_1, \ldots, a_n\colon v_n]^{\mathcal{I},\mathcal{Z}} := \{\bigcup_{i=1}^n F_i\}$ *with* $\{F_i\} = [a_i\colon v_i]^{\mathcal{I},\mathcal{Z}}$ *for all* $i \in \{1, \ldots, n\}$*.*

$S^{\mathcal{I},\mathcal{Z}}$ therefore is a singleton set for variables and closed specifiers. For open specifiers, however, we define $\lfloor a_1\colon v_1, \ldots, a_n\colon v_n \rfloor^{\mathcal{I},\mathcal{Z}}$ to be the set

$$\{F \in \Phi^{\mathcal{I}} \mid F \supseteq G \text{ for } \{G\} = [a_1\colon v_1, \ldots, a_n\colon v_n]^{\mathcal{I},\mathcal{Z}}\}.$$

With the above definitions in place, we can now define the semantics of concepts and roles in the expected way, simply adding the appropriate condition for the additional annotation sets.

Definition 7 (semantics of concepts and roles). *For $A \in \mathsf{N_C}$, $r \in \mathsf{N_R}$, and $S \in \mathbf{S}$, let:*

$$(A@S)^{\mathcal{I},\mathcal{Z}} := \{\delta \mid (\delta, F) \in A^{\mathcal{I}} \text{ for some } F \in S^{\mathcal{I},\mathcal{Z}}\}, \tag{18}$$

$$(r@S)^{\mathcal{I},\mathcal{Z}} := \{(\delta, \epsilon) \mid (\delta, \epsilon, F) \in r^{\mathcal{I}} \text{ for some } F \in S^{\mathcal{I},\mathcal{Z}}\}. \tag{19}$$

The semantics of further concept expressions is defined as usual: $\top^{\mathcal{I},\mathcal{Z}} = \Delta^{\mathcal{I}}$, $\neg C^{\mathcal{I},\mathcal{Z}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I},\mathcal{Z}}$, $(C \sqcap D)^{\mathcal{I},\mathcal{Z}} = C^{\mathcal{I},\mathcal{Z}} \cap D^{\mathcal{I},\mathcal{Z}}$, and $(\exists R.C)^{\mathcal{I},\mathcal{Z}} = \{\delta \mid$ there is $(\delta, \epsilon) \in R^{\mathcal{I},\mathcal{Z}}$ with $\epsilon \in C^{\mathcal{I},\mathcal{Z}}\}$.

$\mathcal{I}$ *satisfies* a concept inclusion of the form (17) if, for all variable assignments $\mathcal{Z}$ that satisfy $\mathcal{Z}(X_i) \in S_i^{\mathcal{I},\mathcal{Z}}$ for all $1 \leq i \leq n$, we have $C^{\mathcal{I},\mathcal{Z}} \subseteq D^{\mathcal{I},\mathcal{Z}}$. Satisfaction of role inclusions is defined analogously. $\mathcal{I}$ satisfies an ontology if it satisfies all of its axioms. As usual, $\models$ denotes both satisfaction and the induced logical entailment relation.

3.2 Semantics of Time

Time-sorted interpretations can be used to interpret $\mathcal{ALCH}_@^{\mathbb{T}}$ ontologies, but they do not take the intended semantics of time into account yet. For example, we might find that $A(c)@[\mathsf{after}\colon 1993]$ holds whereas $A(c)@[\mathsf{time}\colon t]$ does not hold for any time $t \in \mathsf{N_T}$ with $t^{\mathcal{I}} > 1993$. To ensure consistency, we would like to view an interpretation with temporal domain $\Delta_T^{\mathcal{I}}$ as a sequence $(\mathcal{I}_i)_{i \in \Delta_T^{\mathcal{I}}}$ of regular (unsorted) interpretations that define the state of the world at each point in time. Such a sequence represents a *local* view of time as a sequence of events, whereas the time-sorted interpretation represents a *global* view that can explicitly refer to time points. Axioms of $\mathcal{ALCH}_@^{\mathbb{T}}$ refer to this global view, but it should be based on an actual sequence of events. To simplify the relationship between local and global views, we assume that the underlying abstract domain $\Delta_I^{\mathcal{I}}$ and interpretation of constants remains the same over time.

Definition 8. *Consider a temporal domain* $\Delta_T^{\mathcal{I}}$ *and an abstract domain* $\Delta_I^{\mathcal{I}}$*, and let* $(\mathcal{I}_i)_{i\in\Delta_T^{\mathcal{I}}}$ *be a sequence of (unsorted) interpretations with domain* $\Delta_I^{\mathcal{I}}$*, such that, for all* $a \in \mathsf{N_I}$*, we have* $a^{\mathcal{I}_i} = a^{\mathcal{I}_j}$ *for all* $i, j \in \Delta_T^{\mathcal{I}}$*.*

We define a global interpretation *for* $(\mathcal{I}_i)_{i\in\Delta_T^{\mathcal{I}}}$ *as a multi-sorted interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ *as follows. Let* $a^{\mathcal{I}} = a^{\mathcal{I}_i}$ *for all* $a \in \mathsf{N_I}$*. For any finite set* $F \in \Phi^{\mathcal{I}}$*, let* $F_I \coloneqq F \cap (\Delta_I^{\mathcal{I}} \times \Delta_I^{\mathcal{I}})$ *denote its abstract part without any temporal attributes. For any* $A \in \mathsf{N_C}$*,* $\delta \in \Delta^{\mathcal{I}}$*, and* $F \in \Phi^{\mathcal{I}}$ *with* $F \setminus F_I \neq \emptyset$*, we have* $(\delta, F) \in A^{\mathcal{I}}$ *if and only if*[7] $(\delta, F_I) \in A^{\mathcal{I}_i}$ *for some* $i \in \Delta_T^{\mathcal{I}}$*, and the following conditions hold for all* $(a^{\mathcal{I}}, x) \in F$*:*

- *if* $a = \mathsf{time}$*, then* $(\delta, F_I) \in A^{\mathcal{I}_x}$*,*
- *if* $a = \mathsf{before}$*, then* $(\delta, F_I) \in A^{\mathcal{I}_j}$ *for some* $j < x$*,*
- *if* $a = \mathsf{after}$*, then* $(\delta, F_I) \in A^{\mathcal{I}_j}$ *for some* $j > x$*,*
- *if* $a = \mathsf{until}$*, then* $(\delta, F_I) \in A^{\mathcal{I}_j}$ *for all* $j \leq x$*,*
- *if* $a = \mathsf{since}$*, then* $(\delta, F_I) \in A^{\mathcal{I}_j}$ *for all* $j \geq x$*,*
- *if* $a = \mathsf{between}$*, then* $(\delta, F_I) \in A^{\mathcal{I}_j}$ *for some* $j \in [x]$*,*
- *if* $a = \mathsf{during}$*, then* $(\delta, F_I) \in A^{\mathcal{I}_j}$ *for all* $j \in [x]$*,*

where $[x]$ *for an element* $x \in \Delta_{2T}^{\mathcal{I}}$ *denotes the finite interval represented by the pair of numbers x, and* $j \in \Delta_T^{\mathcal{I}}$*. For roles* $r \in \mathsf{N_R}$*, we define* $(\delta, \epsilon, F) \in r^{\mathcal{I}}$ *analogously.*

In words: in a global interpretation all tuples are consistent with the given sequence of local interpretations. One can see a global interpretation as a snapshot of a local interpretation, with timestamps encoding the information of the temporal sequence. If a global interpretation does not contain temporal attributes the characterization of Definition 8 holds vacuously for any temporal sequence, meaning that without temporal attributes the semantics is essentially the same as for $\mathcal{ALCH}_@$.

Definition 9. *An* interpretation *of* $\mathcal{ALCH}_@^{\mathbb{T}}$ *is a time-sorted interpretation* $\mathcal{I}$ *that is a global interpretation of an interpretation sequence* $(\mathcal{I}_i)_{i\in\Delta_T^{\mathcal{I}}}$ *as in Definition 8.*

A model *of an* $\mathcal{ALCH}_@^{\mathbb{T}}$ *ontology* $\mathcal{O}$ *is an* $\mathcal{ALCH}_@^{\mathbb{T}}$ *interpretation that satisfies* $\mathcal{O}$*, and* $\mathcal{O}$ entails *an axiom* α*, written* $\mathcal{O} \models \alpha$*, if* α *is satisfied by all models of* $\mathcal{O}$*.*

By virtue of the syntax and semantics of $\mathcal{ALCH}_@^{\mathbb{T}}$ we can express background knowledge that helps to maintain integrity of the annotated knowledge and allows us to derive new information from it.

Example 6. Recall the imprecise assertion (4). Even without investigating further into the life of Carsten Lutz, we do know that he has published papers as early as 1997 [30], hence we can assume that he was educated before that:

$$\big(\exists \mathsf{educatedAt}@\lfloor \mathsf{before}\colon 1997\rfloor.\top\big)(\mathsf{Carsten}) \tag{20}$$

where we again simplify presentation by allowing a complex concept expression in an assertion. Now together with axiom (9) (or, equivalently, (10) or (12)), we can infer

$$\big(\exists \mathsf{bornIn}@\lfloor \mathsf{between} : [1950, 1996]\rfloor.\top\big)(\mathsf{Carsten}) \tag{21}$$

which, though hardly more precise, serves to illustrate entailments in $\mathcal{ALCH}_@^{\mathbb{T}}$.[8]

[7] 'for some $i \in \Delta_T^{\mathcal{I}}$' is useful for attributes which universally quantify time points (e.g., until).
[8] Readers who long for greater precision may consult the literature [27].

Some temporal attributes are closely related. Clearly, time can be captured by using during or between with singleton intervals. Conversely, during can be expressed by specifying all time points in the respective interval explicitly using time, but this incurs an exponential blow-up over the binary encoding of time intervals. Similarly, between could be expressed as a disjunction of statements with specific times. Since time can be infinite, since and after cannot be captured using finite intervals. It may seem as if until and before correspond to during and between using intervals starting at 0. However, it is not certain that 0 is the first element in the temporal domain of an interpretation, and the next example shows that this cannot be assumed in general.

Example 7. The ontology with the two axioms $C(a)@[\mathsf{until}\colon 10]$ and $C@[\mathsf{before}\colon 5] \sqsubseteq \bot$ is satisfiable in $\mathcal{ALCH}^{\mathbb{T}}_{@}$, but it does not have models that have times before 5. Replacing $\mathsf{until}\colon 10$ with $\mathsf{during}\colon [0, 10]$ would therefore lead to an inconsistent ontology.

4 Reasoning in $\mathcal{ALCH}^{\mathbb{T}}_{@}$

In our investigations, we focus on the decidability and complexity of the satisfiability problem as the central reasoning task. As usual, entailment of assertions is reducible to satisfiability. Also, our definition of assertions could be easily extended to complex concept expressions, since such assertions can be encoded using concept inclusions. Thus, all of our decidability and complexity results hold for the problem of answering instance queries, defined as the class of the assertions allowing complex concept expressions, such as that of Example 2 (Equation 4).

In this section, we study the expressivity and decidability in $\mathcal{ALCH}^{\mathbb{T}}_{@}$. Our first result, Theorem 1, shows that reasoning is on the first level of the analytical hierarchy and therefore highly undecidable.

Theorem 1. *Satisfiability of $\mathcal{ALCH}^{\mathbb{T}}_{@}$ ontologies is Σ^1_1-hard, and thus not recursively enumerable. Moreover, the problem is Σ^1_1-hard even with at most one set variable per inclusion and with only the temporal attributes* time *and* after.

Proof. We reduce from the following tiling problem, known to be Σ^1_1-hard [20]: given a finite set of tile types T with horizontal and vertical compatibility relations H and V, respectively, and $t_0 \in T$, decide whether one can tile $\mathbb{N} \times \mathbb{N}$ with t_0 appearing infinitely often in the first row. We define an $\mathcal{ALCH}^{\mathbb{T}}_{@}$ ontology $\mathcal{O}_{T,t_0}$ that expresses this property. In our encoding, we use the following symbols:

- a concept name A, to mark individuals representing a grid position with a time point;
- a concept name P to keep time points associated with previous columns in the grid;
- concept names A_t, for each $t \in T$, to mark individuals with tile types;
- an individual name a, to be connected with the first row of the grid;
- an auxiliary concept name I, to mark the individual a, and a concept name B, used to create the vertical axis;
- role names r, s, to connect horizontally and vertically the elements of the grid, respectively.

We define $\mathcal{O}_{T,t_0}$ as the set of the following $\mathcal{ALCH}^{\mathbb{T}}_{@}$ assertion and concept inclusions. We start encoding the first row of the grid with an assertion $I(a)$ and the concept inclusions:

$$I \sqsubseteq \exists r.A@\lfloor \mathsf{time}\colon 0 \rfloor \text{ and } \exists r.A@X \sqsubseteq \exists r.A@\lfloor \mathsf{after}\colon X.\mathsf{time} \rfloor.$$

Every element in A must be marked in at most one time point (in fact, exactly one):

$$A@X \sqsubseteq \neg A@\lfloor \mathsf{after}\colon X.\mathsf{time} \rfloor \tag{22}$$

Every element representing a grid position can be associated with exactly one tile type at the same time point:

$$A@X \sqsubseteq \bigsqcup_{t \in T} A_t@\lfloor \mathsf{time}\colon X.\mathsf{time} \rfloor,$$
$$\exists r.A_t@X \sqsubseteq \neg \exists r.A_{t'}@\lfloor \mathsf{time}\colon X.\mathsf{time} \rfloor, \text{ for } t \neq t' \in T.$$

We also have:

$$A_t@X \sqsubseteq A@\lfloor \mathsf{time}\colon X.\mathsf{time} \rfloor, \text{ for each } t \in T$$

to ensure that elements are in A_t and A at the same time point (exactly one one, see Eq. 22). The condition that t_0 appears infinitely often in the first row is expressed with:

$$I \sqsubseteq \exists r.(A_{t_0}@\lfloor \mathsf{time}\colon 0 \rfloor \sqcup A_{t_0}@\lfloor \mathsf{after}\colon 0 \rfloor),$$
$$I \sqcap \exists r.A_{t_0}@X \sqsubseteq \exists r.A_{t_0}@\lfloor \mathsf{after}\colon X.\mathsf{time} \rfloor.$$

To vertically connect subsequent rows of the grid, we have:

$$I \sqsubseteq B \text{ and } B \sqsubseteq \exists s.B.$$

We add, for each $t \in T$, the following inclusion to ensure compatibility between vertically adjacent tile types:

$$\exists r.A_t@X \sqsubseteq \forall s.\exists r.(\bigsqcup_{(t,t') \in V} A_{t'}@\lfloor \mathsf{time}\colon X.\mathsf{time} \rfloor)$$

We also have:

$$\exists s.\exists r.A@X \sqsubseteq \exists r.A@\lfloor \mathsf{time}\colon X.\mathsf{time} \rfloor$$

to ensure that the set of time points in each row is the same. We now encode compatibility between horizontally adjacent tile types. We first state that, given a node associated with a time point p, for every sibling node d, if d is associated with a time point after p then we mark d with P and p:

$$\exists r.A@X \sqsubseteq \forall r.(\neg A@\lfloor \mathsf{after}\colon X.\mathsf{time} \rfloor \sqcup P@\lfloor \mathsf{time}\colon X.\mathsf{time} \rfloor).$$

For each node, P keeps the time points associated with previous columns in the grid (finitely many). We also have:

$$\exists r.P@X \sqsubseteq \exists r.A@\lfloor \mathsf{time}\colon X.\mathsf{time} \rfloor \text{ and } P@X \sqsubseteq A@\lfloor \mathsf{after}\colon X.\mathsf{time} \rfloor$$

to ensure that P keeps only those previous time points. Finally, for each $t \in T$, we add to $\mathcal{O}_{T,t_0}$ the inclusion:

$$\exists r.A_t @X \sqsubseteq \forall r.(\neg A@\lfloor \mathsf{after}\colon X.\mathsf{time}\rfloor \sqcup P@\lfloor \mathsf{after}\colon X.\mathsf{time}\rfloor \sqcup \bigsqcup_{(t,t')\in H} A_{t'}).$$

Intuitively, as P keeps the time points associated with previous columns in the grid, only the node representing the horizontally adjacent grid position of a node associated with a time point p will not be marked with P after p. □

Theorem 2 shows that even if after is only allowed in assertions reasoning is undecidable, though, in the arithmetical hierarchy [35]. For this statement, recall that Σ_1^0 is the class of recursively enumerable problems.

Theorem 2. *Satisfiability of $\mathcal{ALCH}^{\mathbb{T}}_{@}$ ontologies with the temporal attributes* time, after *and* before *but* after *only in assertions is Σ_1^0-complete. The problem is Σ_1^0-hard even with at most one set variable per inclusion.*

The detailed proof of this result can be found in the appendix.

5 Decidable Temporally Attributed DLs

To recover decidability, we need to restrict $\mathcal{ALCH}^{\mathbb{T}}_{@}$ in some way. In this section, we do so by restricting the use of variables or of temporal attributes, leading to a range of different reasoning complexities.

A straightforward approach for recovering decidability is to restrict to *ground* $\mathcal{ALCH}^{\mathbb{T}}_{@}$, where we disallow set and object variables altogether. It is clear from the known complexity of $\mathcal{ALCH}$ that reasoning is still ExpTime-hard. We establish a matching membership result by providing a satisfiability-preserving polynomial time translation to $\mathcal{ALCH}$ extended with role conjunctions and disjunctions (denoted $\mathcal{ALCH}b$), where satisfiability is known to be in ExpTime [36].

Theorem 3. *Satisfiability of ground $\mathcal{ALCH}^{\mathbb{T}}_{@}$ ontologies is* ExpTime*-complete.*

Proof. Consider a ground $\mathcal{ALCH}^{\mathbb{T}}_{@}$ ontology $\mathcal{O}$, and let $k_0 < \ldots < k_n$ be the ascending sequence of all numbers mentioned (in binary encoding) in time points or in time intervals in $\mathcal{O}$. We define $\mathbb{N}_{\mathcal{O}} := \{k_i \mid 0 \leq i \leq n\} \cup \{k_i + 1 \mid 0 \leq i < n\}$, and let $k_{\mathsf{min}} := \min(\mathbb{N}_{\mathcal{O}})$ and $k_{\mathsf{max}} = \max(\mathbb{N}_{\mathcal{O}})$, where we assume $k_{\mathsf{min}} = k_{\mathsf{max}} = 0$ if $\mathbb{N}_{\mathcal{O}} = \emptyset$. For a finite interval $v \subseteq \mathbb{N}$, let $\mathbb{N}^v_{\mathcal{O}}$ be the set of all finite, non-empty intervals $u \subseteq v$ with end points in $\mathbb{N}_{\mathcal{O}}$. The number of intervals in $\mathbb{N}^v_{\mathcal{O}}$ then is polynomial in the size of $\mathcal{O}$.

We translate $\mathcal{O}$ into an $\mathcal{ALCH}b$ ontology $\mathcal{O}^{\dagger}$ as follows. First, $\mathcal{O}^{\dagger}$ contains every axiom from $\mathcal{O}$, with each annotated concept name $A@S$ and each annotated role name $r@S$ replaced by a fresh concept name A_S and a fresh role name r_S, respectively.

Second, given a ground specifier S, we denote by $S(a\colon b)$ the result of removing all temporal attributes from S and adding the pair $a\colon b$. Moreover, let $S_{\mathbb{T}}$ be the set of temporal attribute-value pairs in S. Then, for each A_S and r_S with $S_{\mathbb{T}} \neq \emptyset$, $\mathcal{O}^\dagger$ contains the equivalences (as usual, $\equiv$ refers to bidirectional $\sqsubseteq$ here):

$$A_S \equiv \bigsqcap_{(a:b)\in S_{\mathbb{T}}} (A_{S(a:b)})^\sharp \quad \text{and} \quad r_S \equiv \bigsqcap_{(a:b)\in S_{\mathbb{T}}} (r_{S(a:b)})^\sharp \tag{23}$$

where the concept/role expressions $(H_{S(a:b)})^\sharp$ for $H \in \{A, r\}$ are defined as follows:

- $(H_{S(\mathsf{during}:v)})^\sharp = \bigsqcap_{u\in\mathbb{N}^v_{\mathcal{O}}} H_{S(\mathsf{during}:u)}$
- $(H_{S(\mathsf{between}:v)})^\sharp = \bigsqcup_{k\in(v\cap\mathbb{N}_{\mathcal{O}})} H_{S(\mathsf{during}:[k,k])}$
- $(H_{S(\mathsf{time}:k)})^\sharp = (H_{S(\mathsf{during}:[k,k])})^\sharp$
- $(H_{S(\mathsf{since}:k)})^\sharp = (H_{S(\mathsf{during}:[k,k_{\mathsf{max}}])})^\sharp \sqcap H_{S(\mathsf{since}:k_{\mathsf{max}})}$
- $(H_{S(\mathsf{until}:k)})^\sharp = (H_{S(\mathsf{during}:[k_{\mathsf{min}},k])})^\sharp \sqcap H_{S(\mathsf{until}:k_{\mathsf{min}})}$
- $(H_{S(\mathsf{after}:k)})^\sharp = (H_{S(\mathsf{between}:[k+1,k_{\mathsf{max}}])})^\sharp \sqcup H_{S(\mathsf{after}:k_{\mathsf{max}})}$
- $(H_{S(\mathsf{before}:k)})^\sharp = (H_{S(\mathsf{between}:[k_{\mathsf{min}},k-1])})^\sharp \sqcup H_{S(\mathsf{before}:k_{\mathsf{min}})}$

where $k \neq k_{\mathsf{min}}$ and $k \neq k_{\mathsf{max}}$. If $k \in \{k_{\mathsf{min}}, k_{\mathsf{max}}\}$ then we set $(H_{S(a:k)})^\sharp = H_{S(a:k)}$. Only polynomially many inclusions in the size of $\mathcal{O}$ are introduced by (23) in $\mathcal{O}^\dagger$.

Finally, given attribute-value pairs $a\colon b$ and $c\colon d$ for temporal attributes a and b, we say that $a\colon b$ *implies* $c\colon d$ if $A(e)@[a\colon b] \models A(e)@[c\colon d]$ for some arbitrary $A \in \mathsf{N_C}$ and $e \in \mathsf{N_I}$. Based on a given $\mathsf{N_I}$, this implication relationship is computable in polynomial time. We then extend $\mathcal{O}^\dagger$ with all inclusions $A_S \sqsubseteq A_T$ and $r_S \sqsubseteq r_T$, where A_S, A_T and r_S, r_T are concept and role names occurring in $\mathcal{O}^\dagger$, including those introduced in (23), such that for each temporal attribute-value pair $c\colon d$ in T there is a temporal attribute-value pair $a\colon b$ in S such that $a\colon b$ implies $c\colon d$ and:

- T is an open specifier and the set of non-temporal attribute-value pairs in S is a superset of the set of non-temporal attribute-value pairs in T; or
- S, T are closed specifiers and the set of non-temporal attribute-value pairs in S is equal to the set of non-temporal attribute-value pairs in T.

This finishes the construction of $\mathcal{O}^\dagger$. As shown in the appendix, $\mathcal{O}$ is satisfiable iff $\mathcal{O}^\dagger$ is satisfiable. □

While ground $\mathcal{ALCH}^{\mathbb{T}}_{@}$ can already be used for some interesting conclusions, it is still rather limited. However, satisfiability of (non-ground) $\mathcal{ALCH}_{@}$ ontologies is also decidable [23], and indeed we can regain decidability in $\mathcal{ALCH}^{\mathbb{T}}_{@}$ by restricting the use of variables to non-temporal attributes. Using a similar reasoning as in the case of $\mathcal{ALCH}_{@}$, we obtain a 2ExpTime upper bound by constructing an equisatisfiable (exponentially larger) ground $\mathcal{ALCH}^{\mathbb{T}}_{@}$ ontology. The details of this proof are given in the appendix.

Theorem 4. *Satisfiability in $\mathcal{ALCH}^{\mathbb{T}}_{@}$ is* 2ExpTime*-complete for ontologies without expressions of the form $X.a$; $a\colon x$ with x in* $\mathsf{Var}(\mathsf{N_T})$*; and $a\colon[t, t']$ with one of t, t' in* $\mathsf{Var}(\mathsf{N_T})$*, where a is a temporal attribute.*

Another way for regaining decidability is by limiting the temporal attributes that make reference to time points in the future:

Theorem 5. *Satisfiability of $\mathcal{ALCH}^{\mathbb{T}}_{@}$ ontologies with only the temporal attributes* $\mathsf{during}, \mathsf{time}, \mathsf{before}$ *and* until *is in* 3ExpTime.

The proof of this result is found in the appendix. It is based on translating the $\mathcal{ALCH}^{\mathbb{T}}_{@}$ ontology into a ground $\mathcal{ALCH}^{\mathbb{T}}_{@}$ ontology, which, however, is double-exponential in size if we assume that time points in the temporal domain have been encoded in binary. The claimed 3ExpTime upper bound then follows from Theorem 3.

Our result in our next Theorem 6 below is that this upper bound is tight. The proof is by reduction from the word problem for double-exponentially space-bounded alternating Turing machines (ATMs) [16] to the entailment problem for $\mathcal{ALCH}^{\mathbb{T}}_{@}$ ontologies. The main challenge in this reduction is that we need a mechanism that allows us to transfer the information of a double-exponentially space bounded tape, so that each configuration following a given configuration is actually a successor configuration (i.e., tape cells are changed according to the transition relation). We encode our tape using time: we can have exponentially many time points in an interval with end points encoded in binary. So considering each time point as a bit position, we construct a counter with *exponentially many bits*, encoding the position of double-exponentially many tape cells.

Theorem 6. *Satisfiability of $\mathcal{ALCH}^{\mathbb{T}}_{@}$ ontologies with only* time *and* before *is* 3ExpTime-*hard.*

Our main theorem of this section completes and summarises our results regarding decidability and complexity for different combinations of temporal attributes:

Theorem 7. *In $\mathcal{ALCH}^{\mathbb{T}}_{@}$, any combination of temporal attributes containing* $\{\mathsf{time}, \mathsf{after}\}$ *is undecidable. Moreover, the combination* $\{\mathsf{time}, \mathsf{before}\}$ *is* 3ExpTime-*complete, and the combination* $\{\mathsf{time}, \mathsf{during}, \mathsf{since}, \mathsf{until}\}$ *and every subset of it are* 2ExpTime-*complete.*

The cases of undecidability and 3ExpTime-completeness follow from (the proofs of) Theorems 1, 5, and 6. Hardness for 2ExpTime is inherited from $\mathcal{ALCH}_{@}$ [23], so our proof in the appendix mainly needs to establish the membership for this case.

Certain combinations referring to time points in the future, e.g., time and since, are harmless while others are highly undecidable, e.g., time and after (by Theorem 1). Essentially, what causes undecidability in $\mathcal{ALCH}^{\mathbb{T}}_{@}$ is a combination with the ability to refer to arbitrarily many intervals of time points in the future.

6 Lightweight Temporal Attributed DLs

The complexities of the previous section are still rather high, whereas modern description logics research has often aimed at identifying tractable DLs [9]. In this section, we therefore seek to obtain a tractable temporally attributed DL that is based on the popular $\mathcal{EL}$-family of DLs [6]. We investigate $\mathcal{ELH}^{\mathbb{T}}_{@}$, the fragment of $\mathcal{ALCH}^{\mathbb{T}}_{@}$ which uses only $\exists$, $\sqcap$, $\top$ and $\bot$ in concept expressions. It is clear that variables lead to intractable reasoning complexities, but it turns out that ground $\mathcal{ELH}^{\mathbb{T}}_{@}$ still remains intractable:

Theorem 8. *Satisfiability of ground $\mathcal{ELH}^{\mathbb{T}}_{@}$ ontologies is* ExpTime-*complete.*

Proof. The upper bound follows from Theorem 3. For the lower bound, we show how one can encode disjunctions (i.e., inclusions of the form $\top \sqsubseteq B \sqcup C$), which allow us to reduce satisfiability of ground $\mathcal{ALCH}^{\mathbb{T}}_{@}$ to satisfiability of ground $\mathcal{ELH}^{\mathbb{T}}_{@}$ ontologies. In fact, several combinations of the temporal attributes time, between, before and after suffice to encode $\top \sqsubseteq B \sqcup C$. For example, see the inclusions using the temporal attributes time and between: $\top \sqsubseteq A@\lfloor \mathsf{between} : [1,2] \rfloor$, $A@\lfloor \mathsf{time} : 1 \rfloor \sqsubseteq B$, $A@\lfloor \mathsf{time} : 2 \rfloor \sqsubseteq C$. □

It is known that the entailment problem for $\mathcal{EL}$ ontologies with concept and role names annotated with time intervals over finite models is in PTime [26]. Indeed, our temporal attribute during can be seen as a syntactic variant of the time intervals in the mentioned work and, if we restrict to the temporal attributes time, during, since and until, the complexity of the satisfiability problem for ground $\mathcal{ELH}^{\mathbb{T}}_{@}$ ontologies is in PTime. Our proof here (for ground $\mathcal{ELH}^{\mathbb{T}}_{@}$ over $\mathbb{N}$ or over a finite interval in $\mathbb{N}$) is based on a polynomial translation to $\mathcal{ELH}$ extended with role conjunction, where satisfiability is PTime-complete [36].

Theorem 9. *Satisfiability of ground $\mathcal{ELH}^{\mathbb{T}}_{@}$ ontologies without the temporal attributes* between*,* before *and* after *is* PTime*-complete.*

Proof. Hardness follows from the PTime-hardness of $\mathcal{EL}$ [6]. For membership, note that the translation in Theorem 3 for the temporal attributes during, since and until does not introduce disjunctions or negations. So the result of translating a ground $\mathcal{ELH}^{\mathbb{T}}_{@}$ ontology belongs to $\mathcal{ELH}$ extended with role conjunction. □

7 Related Work

In this section, we discuss the main differences and similarities between our logic and other related formalisms. Potentially related works include classical first-order and second-order logic, temporal extensions of description logics, and temporal extensions of other logics. When setting out to compare our approach to other logics, it is important to understand that there is no immediate formal basis for doing so. Our approach differs both in syntax (structure of formulae) and in semantics (model theory) from existing logics, so that an immediate comparison is not possible. There are three distinct perspectives one might take for discussing comparisons:

(1) Translate models of temporally attributed logics to models of another logic, and investigate which classes of models can be characterised by theories of either type.
(2) Look for polynomial reductions of common inference tasks, i.e., for syntactic translations between formulae that preserve the answer to some decision problem.
(3) Compare intuitive modelling capabilities on an informal level, looking at intended usage and application scenarios.

Approach (1) can lead to the closest relationships between two distinct logical formalisms. Unfortunately, it is not obvious how to relate our temporalised model theory to classical logical formalisms. It is clear that one could capture the semantic conditions of temporally attributed DLs in second-order logic, which would lead to models that explicitly define (axiomatically) the temporal domain and that associate temporal validity with every tuple. This is close in spirit to the way in which *weak second-order logic* was related to (non-temporal) attributed logics by Marx et al. [32], although their work did in fact show a mere reduction of satisfiability in the sense of (2). Our undecidability results of Theorem 1 imply that, for any faithful translation of temporally attributed models into classical relational structures, $\mathcal{ALCH}^{\mathbb{T}}_{@}$ can capture classes of models that are not expressible in first-order logic.

Besides the translation to models of classical logic, it might also be promising to seek direct translations to model theories of temporal logics, especially to metric temporal logics (MTL) [19,5]. So far, the combination of MTL with DLs has only been investigated considering discrete time domains. Recent works on DatalogMTL consider dense (real or rational) time domains [13,14], into which our integer time could be embedded. Note that the containment of integers in rationals and reals does not mean that there is any corresponding relationship between the expressivity of the logics (indeed, decision procedures for DatalogMTL are also based on restricting attention to a suitably defined set of discrete, non-dense time points). However, choosing a discrete domain does not mean that the complexity of the satisfiability problem is lower, neither it means that the technical results are simpler (as we have already pointed out in the introduction, the complexity of satisfiability of $\mathcal{ALC}$ with a concrete discrete domain using the predicates = and < is open). A detailed semantic comparison requires a thorough investigation of the semantic assumptions in either logic, which has to be left to future research.

Approach (2), the syntactic reduction of inference tasks, is the heart of our complexity results. Our upper complexity bounds are obtained by either grounding the ontology and then translating it to an ontology in a classical DL; or directly translating it into a classical DL. Most DLs, including $\mathcal{ALC}$ and $\mathcal{EL}$, are syntactic variants of fragments of first-order logic [7], and thus our decidable fragments can be translated into first-order logic. The difference in the complexity results for $\mathcal{ALC}$ is due to the ability of expressing certain statements in a more succinct way. For $\mathcal{EL}$, we have shown that some temporal attributes increase expressivity, allowing disjunctions (and negations) to be encoded in the logic. A similar interplay between temporal logic and $\mathcal{EL}$ has also been observed in other studies on temporal DLs [3]. Nevertheless the resulting logic is still expressible in $\mathcal{ALC}$ and, thus also in first-order logic.

Approach (3), the comparison of intuitive semantics and modelling applications, brings many further logics into the scope of investigation (not surprisingly, the motivation of modelling time has inspired many technically diverse formalisms). Some of the statements used in our examples can also be naturally expressed in temporal DLs. For instance, axiom (10) in Fig. 3 is expressible in $\mathcal{ALC}$ extended with Linear Temporal Logic [31,39] with:

$$\exists \mathsf{educatedAt}.\top \sqsubseteq \neg\Diamond\exists \mathsf{bornIn}.\top.$$

Other authors have also considered extending $\mathcal{ALC}$ with Metric Temporal Logic (MTL) [19,5], where axiom (4) of Example 2 can be expressed with:

$$\Diamond_{[1950,2000]}\exists \mathsf{bornIn}.\top\,(\mathsf{Carsten}).$$

However, axiom (16) from Example 5 cannot be naturally expressed by temporal DLs. The complexity results can also be very different, for instance, the complexity of propositional

MTL is already undecidable over the reals and ExpSpace-complete over the naturals [1], whereas in Theorem 3 of this paper we show that we can enhance $\mathcal{ALC}$ with many types of time related annotations with time points encoded in *binary* while keeping the same ExpTime complexity of $\mathcal{ALC}$. Regarding temporal $\mathcal{EL}$, it is known that, if temporal operators are allowed in concept expressions then satisfiability is not easier than satisfiability for temporal $\mathcal{ALC}$ [3]; and it decreases to PSpace if temporal operators can only be applied over the axioms [11]. Our lightweight fragment based on $\mathcal{EL}$ features PTime complexity but allows only ground specifiers using particular types of temporal attributes. Syntactic restrictions on the specifiers, similar to those used for attributed $\mathcal{EL}$ [23,24], could also be applied to have a more interesting PTime fragment of temporally attributed $\mathcal{EL}$.

8 Conclusion

We investigated decidability and complexities of attributed description logics enriched with special attributes whose values are interpreted over a temporal dimension. We discussed several ways of restricting the general, undecidable setting in order to regain decidability. Our complexity results range from PTime to 3ExpTime.

As future work, we plan to study forms of generalising our logic to capture the semantics of other standard types of annotations in knowledge graphs, such as provenance [12] and spatial information. Another direction is to study our logic over other temporal domains such as the real numbers (see [13,14] for a combination of Datalog with MTL over the reals). It would also be interesting to investigate query answering.

Acknowledgements This work is partly supported by the German Research Foundation (DFG) in CRC 248 (Perspicuous Systems), CRC 912 (HAEC), and Emmy Noether grant KR 4381/1-1; and by the European Research Council (ERC) Consolidator Grant 771779 (DeciGUT).

References

1. R. Alur and T.A. Henzinger. Real-time logics: Complexity and expressiveness. *Information and Computation*, 104(1):35 – 77, 1993.
2. Alessandro Artale, Enrico Franconi, Rafael Peñaloza, and Francesco Sportelli. A decidable very expressive description logic for databases. In *The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part I*, pages 37–52, 2017.
3. Alessandro Artale, Roman Kontchakov, Carsten Lutz, Frank Wolter, and Michael Zakharyaschev. Temporalising tractable description logics. In *14th International Symposium on Temporal Representation and Reasoning (TIME 2007), 28-30 June 2007, Alicante, Spain*, pages 11–22, 2007.
4. Alessandro Artale, Roman Kontchakov, Frank Wolter, and Michael Zakharyaschev. Temporal description logic for ontology-based data access. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 711–717, 2013.
5. Franz Baader, Stefan Borgwardt, Patrick Koopmann, Ana Ozaki, and Veronika Thost. Metric temporal description logics with interval-rigid names. In *Frontiers of Combining Systems - 11th International Symposium, FroCoS*, pages 60–76, 2017.

6. Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *Proc. 19th Int. Joint Conf. on Artificial Intelligence (IJCAI'05)*, pages 364–369. Professional Book Center, 2005.
7. Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, second edition, 2007.
8. Franz Baader, Silvio Ghilardi, and Carsten Lutz. LTL over description logic axioms. *ACM Trans. Comput. Log.*, 13(3), 2012.
9. Franz Baader, Carsten Lutz, and Anni-Yasmin Turhan. Small is again beautiful in description logics. *KI*, 24(1):25–33, 2010.
10. Daniela Berardi, Diego Calvanese, and Giuseppe De Giacomo. Reasoning on UML class diagrams. *Artif. Intell.*, 168(1-2):70–118, 2005.
11. Stefan Borgwardt and Veronika Thost. Temporal query answering in the description logic EL. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 2819–2825, 2015.
12. Camille Bourgaux and Ana Ozaki. Querying attributed DL-Lite ontologies using provenance semirings. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, 2019.
13. Sebastian Brandt, Elem Güzel Kalayci, Roman Kontchakov, Vladislav Ryzhikov, Guohui Xiao, and Michael Zakharyaschev. Ontology-based data access with a horn fragment of metric temporal logic. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 1070–1076, 2017.
14. Sebastian Brandt, Elem Güzel Kalayci, Vladislav Ryzhikov, Guohui Xiao, and Michael Zakharyaschev. Querying log data with metric temporal logic. *J. Artif. Intell. Res.*, 62:829–877, 2018.
15. Claudia Carapelle and Anni-Yasmin Turhan. Description logics reasoning w.r.t. general Tboxes is decidable for concrete domains with the EHD-property. In *ECAI 2016 - 22nd European Conference on Artificial Intelligence*, pages 1440–1448, 2016.
16. Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. Alternation. *J. of the ACM*, 28(1):114–133, 1981.
17. Jan Chomicki. Temporal query languages: A survey. In Dov M. Gabbay and Hans Jürgen Ohlbach, editors, *Proc. 1st Int. Conf. on Database Theory (ICDT'94)*, volume 827 of *LNCS*, pages 506–534. Springer, 1994.
18. Michael David Fisher, Dov M. Gabbay, and Lluis Vila, editors. *Handbook of Temporal Reasoning in Artificial Intelligence*. Elsevier, 2005.
19. Víctor Gutiérrez-Basulto, Jean Christoph Jung, and Ana Ozaki. On metric temporal description logics. In *Proc. of the 22nd Eur. Conf. on Artificial Intelligence (ECAI'16)*, pages 837–845. IOS Press, 2016.
20. David Harel. Effective transformations on infinite trees, with applications to high undecidability, dominoes, and fairness. *J. ACM*, 33(1):224–248, 1986.
21. Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *J. of Artif. Intell.*, 194:28–61, 2013.
22. Markus Krötzsch. Ontologies for knowledge graphs? In Alessandro Artale, Birte Glimm, and Roman Kontchakov, editors, *Proc. 30th Int. Workshop on Description Logics (DL'17)*, volume 1879 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2017.
23. Markus Krötzsch, Maximilian Marx, Ana Ozaki, and Veronika Thost. Attributed description logics: Ontologies for knowledge graphs. In *The Semantic Web - ISWC - 16th International Semantic Web Conference*, pages 418–435, 2017.
24. Markus Krötzsch, Maximilian Marx, Ana Ozaki, and Veronika Thost. Attributed description logics: Reasoning on knowledge graphs. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI*, pages 5309–5313, 2018.

25. Markus Krötzsch, Sebastian Rudolph, and Pascal Hitzler. Complexities of Horn description logics. *ACM Trans. Comput. Logic*, 14(1):2:1–2:36, 2013.
26. Jared Leo, Ulrike Sattler, and Bijan Parsia. Temporalising EL concepts with time intervals. In *Proc. 27th Int. Workshop on Description Logics (DL'14)*, volume 1193 of *CEUR Workshop Proceedings*, pages 620–632. CEUR-WS.org, 2014.
27. Carsten Lutz. *The complexity of description logics with concrete domains*. PhD thesis, RWTH Aachen University, Germany, 2002.
28. Carsten Lutz. Combining interval-based temporal reasoning with general tboxes. *Artificial Intelligence*, 152(2):235 – 274, 2004.
29. Carsten Lutz, Volker Haarslev, and Ralf Möller. A concept language with role-forming predicate restrictions. Technical report, University of Hamburg, 1997.
30. Carsten Lutz and Ralf Möller. Defined topological relations in description logics. In Ronald J. Brachman, Francesco M. Donini, Enrico Franconi, Ian Horrocks, Alon Y. Levy, and Marie-Christine Rousset, editors, *Proc. 1997 Int. Workshop on Description Logics (DL'97)*, volume 410 of *URA-CNRS*, 1997.
31. Carsten Lutz, Frank Wolter, and Michael Zakharyaschev. Temporal description logics: A survey. In *15th International Symposium on Temporal Representation and Reasoning, TIME*, pages 3–14, 2008.
32. Maximilian Marx, Markus Krötzsch, and Veronika Thost. Logic on MARS: Ontologies for generalised property graphs. In Carles Sierra, editor, *Proc. 26th Int. Joint Conf. on Artificial Intelligence (IJCAI'17)*, pages 1188–1194. IJCAI, 2017.
33. openCypher community. Cypher Query Language Reference, Version 9. available at `http://www.opencypher.org/resources`, 2019.
34. Marko A. Rodriguez and Peter Neubauer. Constructions from dots and lines. *Bulletin of the American Society for Information Science and Technology*, 36(6):35–41, 2010.
35. Hartley Rogers, Jr. *Theory of Recursive Functions and Effective Computability*. MIT Press, paperback edition edition, 1987.
36. Sebastian Rudolph, Markus Krötzsch, and Pascal Hitzler. Cheap Boolean role constructors for description logics. In Steffen Hölldobler, Carsten Lutz, and Heinrich Wansing, editors, *Proc. 11th European Conf. on Logics in Artificial Intelligence (JELIA'08)*, volume 5293 of *LNAI*, pages 362–374. Springer, 2008.
37. Francesco Sportelli and Enrico Franconi. Formalisation of ORM derivation rules and their mapping into OWL. In *On the Move to Meaningful Internet Systems: OTM 2016 Conferences - Confederated International Conferences: CoopIS, C&TC, and ODBASE 2016, Rhodes, Greece, October 24-28, 2016, Proceedings*, pages 827–843, 2016.
38. Denny Vrandečić and Markus Krötzsch. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10), 2014.
39. Frank Wolter and Michael Zakharyaschev. Temporalizing description logics. *Frontiers of Combining Systems*, 2:379–402, 1999.

A Proofs for Section 4

Theorem 2. *Satisfiability of $\mathcal{ALCH}^{\mathbb{T}}_{@}$ ontologies with the temporal attributes* time, after *and* before *but* after *only in assertions is Σ^0_1-complete. The problem is Σ^0_1-hard even with at most one set variable per inclusion.*

Proof. We first show hardness. We reduce the word problem for deterministic Turing machines (DTM) to satisfiability of $\mathcal{ALCH}^{\mathbb{T}}_{@}$ ontologies with the temporal attribute after occurring only in assertions. A DTM is a tuple $(Q, \Sigma, \Theta, q_0, q_{\mathsf{f}})$, where:

- Q is a finite set of states,
- Σ is a finite alphabet containing the *blank symbol* $\sqcup$,
- $\{q_0, q_{\mathsf{f}}\} \subseteq Q$ are the *initial* and the *final* states, resp., and
- $\Theta : Q \times \Sigma \to Q \times \Sigma \times \{l, r\}$ is the *transition function*.

A *configuration* of $\mathcal{M}$ is a word wqw' with $w, w' \in \Sigma^*$ and q in Q. The meaning is that the (one-sided infinite) tape contains the word ww' with only blanks behind it, the machine is in state q and the head is on the left-most symbol of w'. The notion of a *successive configuration* is defined in the usual way, in terms of the transition relation Θ. A *computation* of $\mathcal{M}$ on a word w is a sequence of successive configurations $\alpha_0, \alpha_1, \ldots$, where $\alpha_0 = q_0 w$ is the *initial configuration* for the input w. Let $\mathcal{M}$ be a DTM and $w = \sigma_1 \sigma_2 \cdots \sigma_n$ an input word. Assume w.l.o.g. that $\mathcal{M}$ never attempts to move to the left when its head is in the left-most tape position and that q_0 occurs only in the domain of Θ (but not in the range).

We construct an $\mathcal{ALCH}^{\mathbb{T}}_{@}$ ontology $\mathcal{O}_{\mathcal{M},w}$ with after occurring only in assertions that is satisfiable iff $\mathcal{M}$ accepts w. Models of $\mathcal{O}_{\mathcal{M},w}$ have a similar structure as in the proof of Theorem 1. We create a vertical chain with:

$$I(a), \quad I \sqsubseteq B \quad \text{and} \quad B \sqsubseteq \exists s.B$$

and ensure that horizontally the set of time points is the same:

$$\exists r.A@X \sqsubseteq \forall s.\exists r.A@\lfloor \mathsf{time}\colon X.\mathsf{time} \rfloor, \tag{24}$$

$$\exists s.\exists r.A@X \sqsubseteq \exists r.A@\lfloor \mathsf{time}\colon X.\mathsf{time} \rfloor. \tag{25}$$

Every element representing a tape cell is marked with A in at most one time point (in fact, it will be exactly one):

$$A@X \sqsubseteq \neg A@\lfloor \mathsf{before}\colon X.\mathsf{time} \rfloor$$

The main difference is that horizontally we do not have infinitely many sibling nodes. That is, over the naturals, adding the inclusion $\exists r.A@X \sqsubseteq \exists r.A@\lfloor \mathsf{before}\colon X.\mathsf{time} \rfloor$ would make $\mathcal{O}_{\mathcal{M},w}$ unsatisfiable and here we cannot use after in inclusions. Instead, for each $q \neq q_{\mathsf{f}}$ in Q, we add to $\mathcal{O}_{\mathcal{M},w}$ the inclusions:

$$S_q \sqcap A@X \sqsubseteq S_q@\lfloor \mathsf{time}\colon X.\mathsf{time} \rfloor, \tag{26}$$

$$\exists r.S_q@X \sqsubseteq \exists r.A@\lfloor \mathsf{before}\colon X.\mathsf{time} \rfloor \tag{27}$$

where S_q is a concept name representing a state. Intuitively, each vertically aligned set of elements (w.r.t. time) represents a configuration and a sequence of configurations going backwards in time represents a computation of $\mathcal{M}$ with input w. The goal is to ensure that $\mathcal{O}_{\mathcal{M},w}$ is satisfiable iff we reach the final state, that is, w is accepted by $\mathcal{M}$.

We now add to $\mathcal{O}_{\mathcal{M},w}$ assertions to trigger the inclusions in Equations 24, 25, 26 and 27:

$$r(a,b), \quad S_{q_0}(b), \quad A(b)@[\mathsf{after}\colon 0].$$

We also use in our encoding concepts C_σ for each symbol $\sigma \in \Sigma$. To encode the input word $w = \sigma_1\sigma_2\cdots\sigma_n$, we add:

$$C_\sigma \sqcap A@X \sqsubseteq C_\sigma@\lfloor\mathsf{time}\colon X.\mathsf{time}\rfloor \text{ for each } \sigma \in \Sigma,$$

$$C_{\sigma_1}(b), \quad \exists r.S_{q_0}@X \sqsubseteq \forall s^i.\exists r.C_{\sigma_{i+1}}@\lfloor\mathsf{time}\colon X.\mathsf{time}\rfloor$$

for $1 \leq i < n$. It is straightforward to add inclusions encoding that (i) the rest of the tape in the initial configuration is filled with the blank symbol, (ii) each node representing a tape cell in a configuration is associated with only one C_σ with $\sigma \in \Sigma$ and (iii) at most one S_q with $q \in Q$ (exactly the node representing the head position). Also, for each element, the time point associated with A is the same for the concepts of the form C_σ and S_q (if true in the node).

To access the 'next' configuration, we use an auxiliary concept F that keeps time points in the future. Recall that since a computation here goes backwards in time, these time points are associated with previous configurations:

$$\exists r.A@X \sqsubseteq \forall r.(\neg A@\lfloor\mathsf{before}\colon X.\mathsf{time}\rfloor \sqcup F@\lfloor\mathsf{time}\colon X.\mathsf{time}\rfloor).$$

We now ensure that tape contents are transferred to the 'next' configuration, except for the tape cell at the head position:

$$\exists r.(C_\sigma@X \sqcap S_{\overline{q}}) \sqsubseteq \forall r.(F@\lfloor\mathsf{before}\colon X.\mathsf{time}\rfloor \sqcup \neg A@\lfloor\mathsf{before}\colon X.\mathsf{time}\rfloor \sqcup C_\sigma)$$

for each $\sigma \in \Sigma$, where $S_{\overline{q}}$ is a shorthand for $\neg \bigsqcup_{q\in Q} S_q$. Finally we encode the transition function. We explain for $\Theta(q,\sigma) = (q',\tau,D)$ with $D = r$ (the case with $D = l$ can be handled analogously). We encode that the 'next' state is q':

$$\exists r.(S_q@X \sqcap C_\sigma) \sqsubseteq \forall s.\forall r.(F@\lfloor\mathsf{before}\colon X.\mathsf{time}\rfloor \sqcup \neg A@\lfloor\mathsf{before}\colon X.\mathsf{time}\rfloor \sqcup S_{q'}) \tag{28}$$

and change to τ the tape cell at the (previous) head position:

$$\exists r.(S_q@X \sqcap C_\sigma) \sqsubseteq \forall r.(F@\lfloor\mathsf{before}\colon X.\mathsf{time}\rfloor \sqcup \neg A@\lfloor\mathsf{before}\colon X.\mathsf{time}\rfloor \sqcup C_\tau).$$

Equation 28 also increments the head position.

This finishes our reduction.

For the upper bound, we point out that if an $\mathcal{ALCH}^{\mathbb{T}}_{@}$ ontology $\mathcal{O}$ with after only in assertions is satisfiable then there is a satisfiable ontology $\mathcal{O}'$ that is the result of replacing each occurrence of $\mathsf{after} : k$ in $\mathcal{O}$ by some $\mathsf{time} : l$ with $k < l \in \mathbb{N}$. By Theorem 5, one can decide satisfiability of $\mathcal{O}'$ (that is, satisfiability of ontologies with only the temporal attributes time and before). As the replacements of $\mathsf{after} : k$ by $\mathsf{time} : l$ in assertions can be enumerated, it follows that satisfiability of $\mathcal{ALCH}^{\mathbb{T}}_{@}$ ontologies is in Σ^0_1. □

B Proofs for Section 5

Theorem 3. *Satisfiability of ground $\mathcal{ALCH}^{\mathbb{T}}_{@}$ ontologies is* ExpTime*-complete.*

Proof. The construction of an ontology $\mathcal{O}^{\dagger}$ was already given in the main text. It remains to show that $\mathcal{O}$ is satisfiable iff $\mathcal{O}^{\dagger}$ is satisfiable. Given a model $\mathcal{I}$ of $\mathcal{O}$, we directly obtain an $\mathcal{ALCH}b$ interpretation $\mathcal{J}$ over $\Delta^{\mathcal{I}}$ by undoing the renaming and applying $\mathcal{I}$, i.e., by mapping $A_S \in \mathsf{N_C}$ to $A@S^{\mathcal{I}}$, $r_S \in \mathsf{N_R}$ to $r@S^{\mathcal{I}}$, and $a \in \mathsf{N_I}$ to $a^{\mathcal{I}}$. By the semantics of $\mathcal{ALCH}^{\mathbb{T}}_{@}$, $\mathcal{J} \models \mathcal{O}^{\dagger}$. Conversely, given an $\mathcal{ALCH}b$ model $\mathcal{J}$ of $\mathcal{O}^{\dagger}$, we construct an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of $\mathcal{ALCH}^{\mathbb{T}}_{@}$ with $\Delta^{\mathcal{I}}_{T} = [\max(0, k_{\mathsf{min}} - 2), k_{\mathsf{max}} + 2]$ and $\Delta^{\mathcal{I}}_{I} = \Delta^{\mathcal{J}} \cup \{\star\} \cup \mathbb{T}$, where $\mathbb{T}$ is the set of temporal attributes and $\star$ is a fresh individual name. We define $a^{\mathcal{I}} \coloneqq a^{\mathcal{J}}$ for all $a \in \mathsf{N_I} \cup \mathsf{N_T} \cup \mathsf{N}^2_{\mathsf{T}}$.

For a ground closed specifier S with $a_1 : b_1, \ldots, a_n : b_n$ as non-temporal attributes, we define:

$$F_S \coloneqq \{(a_1^{\mathcal{I}}, b_1^{\mathcal{I}}), \ldots, (a_n^{\mathcal{I}}, b_n^{\mathcal{I}})\}.$$

Similarly, for a ground open specifier S with $a_1 : b_1, \ldots, a_n : b_n$ as non-temporal attribute-value pairs, we define:

$$F_S \coloneqq \{(a_1^{\mathcal{I}}, b_1^{\mathcal{I}}), \ldots, (a_n^{\mathcal{I}}, b_n^{\mathcal{I}}), (\star, \star)\}.$$

To simplify the presentation, we write $a : b \in S$ if $a : b$ occurs in S. Furthermore, let $A^{\mathcal{I}_i}$ be the set of all tuples (δ, F_S) such that one of the following holds:

- $\delta \in A_S^{\mathcal{J}}, \mathsf{during}: v \in S$ and $i \in v$;
- $\delta \in A_S^{\mathcal{J}}, \mathsf{after}: k_{\mathsf{max}} \in S$ and $i = k_{\mathsf{max}} + 1$;
- $\delta \in A_S^{\mathcal{J}}, \mathsf{since}: k_{\mathsf{max}} \in S$ and $k_{\mathsf{max}} + 1 \leq i \leq k_{\mathsf{max}} + 2$;
- $\delta \in A_S^{\mathcal{J}}, \mathsf{before}: k_{\mathsf{min}} \in S, i = k_{\mathsf{min}} - 1$ and $k_{\mathsf{min}} > 0$;
- $\delta \in A_S^{\mathcal{J}}, \mathsf{until}: k_{\mathsf{min}} \in S, \max(k_{\mathsf{min}} - 2, 0) \leq i \leq k_{\mathsf{min}} - 1$ and $k_{\mathsf{min}} > 0$.

We define $r^{\mathcal{I}_i}$ analogously. Given the definitions of $A^{\mathcal{I}_i}$ and $r^{\mathcal{I}_i}$, for all $i \in \mathbb{N}$, $A \in \mathsf{N_C}$ and $r \in \mathsf{N_R}$, we define $\cdot^{\mathcal{I}}$ as in Definition 8.

Claim. For all A_S, r_S occurring in $\mathcal{O}^{\dagger}$: (1) $A_S^{\mathcal{J}} = A@S^{\mathcal{I}}$ and (2) $r_S^{\mathcal{J}} = r@S^{\mathcal{I}}$.

Proof of the Claim. If no temporal attribute occurs in S then by definition of $\mathcal{I}$ (in particular, F_S), we clearly have that $\delta \in A_S^{\mathcal{J}}$ iff $\delta \in A@S^{\mathcal{I}}$. Also, by semantics of $\mathcal{ALCH}^{\mathbb{T}}_{@}$, for a ground specifier S with a non-empty set $S_{\mathbb{T}}$ of temporal attributes the following holds for any $\mathcal{I}$ and concept $A@S$:

$$A@S^{\mathcal{I}} = \bigcap_{a : b \in S_{\mathbb{T}}} A@S(a : b)^{\mathcal{I}}$$

So we can consider $A@S$ with S containing only one temporal attribute. We argue for during and $\mathsf{between}$ (one can give a similar argument for the other temporal attributes):

- if the temporal attribute-value pair $\mathsf{during}: v$ is in S then, by definition of $\mathcal{I}$ (and F_S), $\delta \in A_S^{\mathcal{J}}$ iff $\delta \in A@S^{\mathcal{I}}$;

– if the temporal attribute-value pair between: v is in S then, by Equation 23, $\delta \in A_S^{\mathcal{J}}$ iff $\delta \in \bigcup_{k \in v \cap \mathbb{N}_{\mathcal{O}}} A_{S(\mathsf{during}\colon [k,k])}^{\mathcal{J}}$. By definition of $\mathcal{I}$, $\delta \in A_{S(\mathsf{during}\colon [k,k])}^{\mathcal{J}}$ iff $\delta \in A@S(\mathsf{during}\colon [k,k])^{\mathcal{I}}$, for $k \in v \cap \mathbb{N}_{\mathcal{O}}$. Then,

$$\delta \in A_S^{\mathcal{J}} \text{ iff } \delta \in \bigcup_{k \in v \cap \mathbb{N}_{\mathcal{O}}} A@S(\mathsf{during}\colon [k,k])^{\mathcal{I}};$$

so $\delta \in A@S^{\mathcal{I}}$.

In the definition of $\mathbb{N}_{\mathcal{O}}$, we add $k_i + 1$ for each k_i occurring in $\mathcal{O}$, to ensure that axioms such as $\top \sqsubseteq A@\lfloor \mathsf{between}\colon [k,l] \rfloor \sqcap \neg A@\lfloor \mathsf{time}\colon k \rfloor \sqcap \neg A@\lfloor \mathsf{time}\colon l \rfloor$ with $l - k \geq 2$ remain satisfiable. Also, in the definition of $\mathcal{I}$ we use the interval $\Delta_T^{\mathcal{I}} = [\max(0, k_{\mathsf{min}} - 2), k_{\mathsf{max}} + 2]$, and so, we give a margin of two 'additional' points in each side of the interval $[k_{\mathsf{min}}, k_{\mathsf{max}}]$ used in the translation. This is to ensure that axioms such as $\top \sqsubseteq A@\lfloor \mathsf{before}\colon k_{\mathsf{min}} \rfloor \sqcap \neg A@\lfloor \mathsf{until}\colon k_{\mathsf{min}} \rfloor$ with $k_{\mathsf{min}} \geq 2$ remain satisfiable. Point (2) can be proven with an easy adaptation of Point (1).

The Claim directly implies that $\mathcal{I} \models \mathcal{O}$. Note that $\star$ ensures that axioms such as $\top \sqsubseteq A@\lfloor a\colon b \rfloor \sqcap \neg A@[a\colon b]$ remain satisfiable. □

Theorem 4. *Satisfiability in $\mathcal{ALCH}_{@}^{\mathbb{T}}$ is* 2ExpTime*-complete for ontologies without expressions of the form $X.a$; $a\colon x$ with x in* $\mathsf{Var}(\mathsf{N_T})$*; and $a\colon [t, t']$ with one of t, t' in* $\mathsf{Var}(\mathsf{N_T})$*, where a is a temporal attribute.*

Proof. The 2ExpTime lower bound follows from the fact that satisfiability of $\mathcal{ALCH}_{@}$ (so without temporal attributes) is already 2ExpTime-hard [23]. Our proof strategy for the upper bound consists on defining an ontology with grounded versions of inclusion axioms. Let $\mathcal{O}$ be an $\mathcal{ALCH}_{@}^{\mathbb{T}}$ ontology and let $\mathsf{N} \coloneqq \mathsf{N_I^{\mathcal{O}}} \cup \mathsf{N_T^{\mathcal{O}}} \cup \mathsf{N_T^{2\mathcal{O}}}$ be the union of the sets of individual names, time points, and intervals, occurring in $\mathcal{O}$, respectively. Let $\mathcal{I}$ be an interpretation of $\mathcal{ALCH}_{@}^{\mathbb{T}}$ over the domain $\Delta^{\mathcal{I}} = \mathsf{N} \cup \{x\}$, where x is a fresh individual name, satisfying $a^{\mathcal{I}} = a$ for all $a \in \mathsf{N}$. Let $\mathcal{Z} : \mathsf{N_V} \to \Phi_{\mathcal{O}}^{\mathcal{I}}$ be a variable assignment, where $\Phi_{\mathcal{O}}^{\mathcal{I}} \coloneqq \mathcal{P}_{\mathsf{fin}}\left(\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}\right)$. Consider a concept inclusion α of the form $X_1 : S_1, \ldots, X_n : S_n\ (C \sqsubseteq D)$. We say that $\mathcal{Z}$ is *compatible with* α if $\mathcal{Z}(X_i) \in S_i^{\mathcal{I},\mathcal{Z}}$ for all $1 \leq i \leq n$. In this case, the *$\mathcal{Z}$-instance $\alpha_{\mathcal{Z}}$* of α is the concept inclusion $C' \sqsubseteq D'$ obtained by

– replacing each X_i by $[a\colon b \mid (a,b) \in \mathcal{Z}(X_i)]$;
– replacing every $a\colon X_i.b$ occurring in some specifier (with a, b *non-temporal* attributes) by all $a\colon c$ such that $(b,c) \in \mathcal{Z}(X_i)$; and
– replacing each object variable x by $\mathcal{Z}(x)$.

Then, the grounding $\mathcal{O}_g$ of $\mathcal{O}$ contains all $\mathcal{Z}$-instances $\alpha_{\mathcal{Z}}$ for all concept inclusions α in $\mathcal{O}$ and all compatible variable assignments $\mathcal{Z}$; and analogous axioms for role inclusions.

There may be (at most) exponentially many different instances for each terminological axiom in $\mathcal{O}$, thus $\mathcal{O}_g$ is of exponential size. We show that $\mathcal{O}$ is satisfiable iff $\mathcal{O}_g$ is satisfiable. By construction, we have $\mathcal{O} \models \mathcal{O}_g$, i.e., any model of $\mathcal{O}$ is also a model of $\mathcal{O}_g$. Conversely, let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a model of $\mathcal{O}_g$. W.l.o.g., assume that there is $x \in \Delta^{\mathcal{I}}$ such that $x \neq a^{\mathcal{I}}$ for all $a \in \mathsf{N_I^{\mathcal{O}}} \setminus \{x\}$. For an annotation set $F \in \mathcal{P}_{\mathsf{fin}}\left(\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}\right)$, we define

$\mathrm{rep}_x(F)$ to be the annotation set obtained from F by replacing any individual $\delta \notin \mathcal{I}(\mathsf{N}_\mathsf{I}^\mathcal{O})$ in F by x.

Let $\sim$ be the equivalence relation induced by $\mathrm{rep}_x(F) = \mathrm{rep}_x(G)$ and define an interpretation $\mathcal{J}$ of $\mathcal{ALCH}_@^\mathbb{T}$ over the domain $\Delta^\mathcal{J} := \Delta^\mathcal{I}$, where $A^\mathcal{J} := \{(\delta, F) \mid (\delta, G) \in A^\mathcal{I}$ and $F \sim G\}$ for all $A \in \mathsf{N_C}$, $r^\mathcal{J} := \{(\delta, \epsilon, F) \mid (\delta, \epsilon, G) \in r^\mathcal{I}$ and $F \sim G\}$ for all $r \in \mathsf{N_R}$, and $a^\mathcal{J} := a^\mathcal{I}$ for all $a \in \mathsf{N_I} \cup \mathsf{N_T} \cup \mathsf{N_T^2}$. It remains to show that $\mathcal{J}$ is indeed a model of $\mathcal{O}$. Suppose for a contradiction that there is a concept inclusion α in $\mathcal{O}$ that is not satisfied by $\mathcal{J}$ (the case for role inclusions is analogous). Then we have some compatible variable assignment $\mathcal{Z}$ that leaves α unsatisfied. Let $\mathcal{Z}_x$ be the variable assignment $X \mapsto \mathrm{rep}_x(\mathcal{Z}(X))$ for all $X \in \mathsf{N_V}$. Clearly, as expressions of the form $a\colon X_i.b$, $a\colon x$, and $a\colon [t, t']$ with at least one of t, t' an object variable, are not allowed for a, b being temporal attributes, $\mathcal{Z}_x$ is also compatible with α. But now we have $C^{\mathcal{J},\mathcal{Z}} = C^{\mathcal{I},\mathcal{Z}_x}$ for all $\mathcal{ALCH}_@^\mathbb{T}$ concepts C, yielding the contradiction $\mathcal{I} \not\models \alpha_{\mathcal{Z}_x}$. Thus, $\mathcal{O}$ is satisfiable iff $\mathcal{O}_g$ is satisfiable. The result then follows from Theorem 3. ◻

Theorem 5. *Satisfiability of $\mathcal{ALCH}_@^\mathbb{T}$ ontologies with only the temporal attributes* during, time, before *and* until *is in* 3ExpTime.

Proof. The difference w.r.t. the proof of Theorem 4 is that here expressions of the form $a\colon X_i.b$, $a\colon x$, and $a\colon [t, t']$ with at least one of t, t' an object variable, may occur in front of the temporal attributes during, before, time and until and the other temporal attributes are not allowed (not even in assertions). Let v be the internal $[0, k]$, where k is the largest number occurring in $\mathcal{O}$ (or 0 if no number occurs). To define our ground translation, we consider variable assignments $\mathcal{Z} : \mathsf{N_V} \to \Phi^\mathcal{I}_{\mathcal{O},v}$, where $\Phi^\mathcal{I}_{\mathcal{O},v} := \mathcal{P}_\mathsf{fin}\left(\Delta^\mathcal{I} \times \Delta^\mathcal{I}\right)$ and $\Delta^\mathcal{I}$ is the set of all individual names in $\mathcal{O}$ plus a fresh individual name x, all time points in v and all intervals contained in v. This gives us a ground ontology $\mathcal{O}_g$ with size double-exponential in the size of $\mathcal{O}$. Clearly, $\mathcal{O}$ is satisfiable iff $\mathcal{O}_g$ is satisfiable. ◻

Theorem 6. *Satisfiability of $\mathcal{ALCH}_@^\mathbb{T}$ ontologies with only* time *and* before *is* 3ExpTime-*hard.*

Proof. We reduce the word problem for double-exponentially space-bounded alternating Turing machines (ATMs) to the entailment problem for $\mathcal{ALCH}_@^\mathbb{T}$ ontologies. We consider w.l.o.g. ATMs with only finite computations on any input. As usual, an ATM is a tuple $\mathcal{M} = (Q, \Sigma, \Theta, q_0)$, where:

- $Q = Q_\exists \uplus Q_\forall$ is a finite set of states, partitioned into *existential states* $Q_\exists$ and *universal states* $Q_\forall$,
- Σ is a finite alphabet containing the *blank symbol* $\sqcup$,
- $q_0 \in Q$ is the *initial state*, and
- $\Theta \subseteq Q \times \Sigma \times Q \times \Sigma \times \{l, r\}$ is the *transition relation*.

We use the same notions of configuration, computation and initial configuration given in the proof of Theorem 2. We recall the acceptance condition of an ATM. A configuration $\alpha = wqw'$ is *accepting* iff

- α is a universal configuration and all its successor configurations are accepting, or
- α is an existential configuration and at least one of its successor configurations is accepting.

Note that, by the definition above, universal configurations without any successors are accepting. We assume w.l.o.g. that all configurations wqw of a computation of $\mathcal{M}$ satisfy $|ww'| \leq 2^{2^n}$. $\mathcal{M}$ *accepts* a word in $(\Sigma \setminus \{\sqcup\})^*$ (in space double-exponential in the size of the input) iff the initial configuration is accepting.

There exists a double-exponentially space bounded ATM $\mathcal{M} = (Q, \Sigma, q_0, \Theta)$ whose word problem is 3ExpTime-hard [16]. Let $\mathcal{M}$ be such a double-exponentially space bounded ATM and $w = \sigma_1\sigma_2 \cdots \sigma_n$ an input word. W.l.o.g., we assume that $\mathcal{M}$ never attempts to move to the left (right) when the head is on the left-most (right-most) tape cell.

We construct an $\mathcal{ALCH}^{\mathbb{T}}_{@}$ ontology $\mathcal{O}_{\mathcal{M},w}$ that entails $A(a)$ iff $\mathcal{M}$ accepts w. We represent configurations using individuals in $\mathcal{O}_{\mathcal{M},w}$, which are connected to the corresponding successor configurations by roles encoding the transition. W.l.o.g., we assume that these individuals form a tree, which we call the *configuration tree*. Furthermore, each node of this tree, i.e., each configuration, is connected to 2^{2^n} individuals representing the tape cells. The main ingredients of our construction are:

- an individual a denoting the root of the configuration tree;
- an attribute bit, with values in $\{0, 1\}$, used to encode double-exponentially many tape positions;
- an attribute flip which has value 1 at a (unique) time point where bit has value 0 and bit has value 1 in all subsequent time points;
- a concept A marking accepting configurations;
- a concept H marking the head position;
- a concept T marking tape cells;
- a concept I marking the initial configuration;
- concepts S_q for each state $q \in Q$;
- concepts C_σ for each symbol $\sigma \in \Sigma$;
- roles r_θ for all transitions $\theta \in \Theta$;
- a role tape connecting configurations to tape cells; and
- attributes $a_0, \ldots, a_n$ to encode the binary representation of time values.

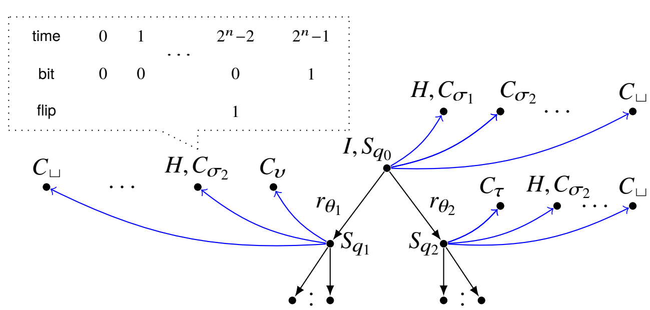

Fig. 4. A model of $\mathcal{O}_{\mathcal{M},w}$ encoding the computation tree of an ATM; blue edges (potentially grey) represent the tape role (we omit for brevity T in nodes representing tape cells)

To encode the binary representation of time values we first state that for $\mathsf{time}\colon 2^n - 1$ we have all a_i set to 1:

$$T \sqsubseteq T@\lfloor \mathsf{time}\colon 2^n - 1, a_n\colon 1, \ldots, a_0\colon 1 \rfloor.$$

We now use the following intuition: if the a_i attributes represent a pattern $s \cdot 1000$, where s is a binary sequence and $\cdot$ means concatenation, then $s \cdot 0111$ should occur *before* that pattern in the time line. To ensure this, we add concept inclusions of the form, for all $0 \leq i \leq n$:

$$X\mathbf{:}\,S\ (T@X \sqsubseteq T@\lfloor \mathsf{before}\colon X.\mathsf{time}, P^X_{a_i} \rfloor)$$

where S is $\lfloor a_i\colon 1, a_{i-1}\colon 0, \ldots, a_0\colon 0 \rfloor$ and $P^X_{a_i}$ abbreviates

$$a_n\colon X.a_n, \ldots, a_{i+1}\colon X.a_{i+1}, a_i\colon 0, a_{i-1}\colon 1, \ldots, a_0\colon 1.$$

By further adding a concept inclusion encoding that a_i can only be one of $1, 0$ at the same time point we have that, in any model, the a_i attributes encode the binary representation of the corresponding time value, for time points in $[0, 2^n - 1]$. This means that, for time points in $[0, 2^n - 1]$, we can simulate the temporal attribute after by using variables and specifiers of the form $X\mathbf{:}\,\lfloor a_i\colon 1 \rfloor$ and $\lfloor a_n\colon X.a_n, \ldots, a_{i-1}\colon X.a_{i-1} a_i\colon 0 \rfloor$, for all $0 \leq i \leq n$.

Remark 1. To simplify the presentation, in the following, we use the temporal attributes after and during (the latter is used to encode the initial configuration). Given the construction above it is straightforward to replace the inclusions using after and during with inclusions using the attributes a_i.

We encode the meaning of the attribute flip (i.e., it has value 1 at the time point from which bits should be flipped to increment a tape position) with the following concept inclusions:

$$T@\lfloor \mathsf{bit}\colon 0 \rfloor \sqsubseteq T@\lfloor \mathsf{flip}\colon 1 \rfloor \tag{29}$$

$$X\mathbf{:}\,\lfloor \mathsf{flip}\colon 1 \rfloor (T@X \sqsubseteq \neg T@\lfloor \mathsf{bit}\colon 0, \mathsf{after}\colon X.\mathsf{time} \rfloor) \tag{30}$$

$$X\mathbf{:}\,\lfloor \mathsf{flip}\colon 1 \rfloor (T@X \sqsubseteq T@\lfloor \mathsf{bit}\colon 0, \mathsf{time}\colon X.\mathsf{time} \rfloor) \tag{31}$$

Intuitively, in Equation 29 we say that if there is a time point where we have bit with value 0 then there is a time point where we should flip some bit to increment the tape position, i.e., where flip is 1. In Equation 30 we state that there is no bit with value 0 after a time point marked with flip set to 1. Finally, in Equation 31, we state that bit has value 0 where flip has value 1. Thus, Equations 30 and 31 ensure that there is at most one time point where flip has value 1.

Let $\mathbf{\Omega}$ be a sequence with the following variables X_i^j, with $1 \leq i \leq n$ and $1 \leq j \leq 5$, and their respective specifiers:

- $X_i^1\mathbf{:}\,\lfloor \mathsf{flip}\colon 1 \rfloor$, we look at our auxiliary attribute that indicates from which time point we should flip our bits to obtain the next tape position (this will be a time point with bit value 0);

– we also define $X_i^2 : \lfloor \mathsf{before}: X_i^1.\mathsf{time}, \mathsf{bit}: 0 \rfloor$ and $X_i^3 : \lfloor \mathsf{before}: X_i^1.\mathsf{time}, \mathsf{bit}: 1 \rfloor$, to filter time points with bit values 0 and 1, respectively, before the time point with $\mathsf{flip} : 1$, related to X_i^1;
– we use $X_i^4 : \lfloor \mathsf{bit} : 0 \rfloor$ and $X_i^5 : \lfloor \mathsf{bit} : 1 \rfloor$ to filter time points bit values 0 and 1, respectively.

Basically, the first three variables are related to specifiers that filter the information needed to increment the tape position encoded with the bit attribute. The last two variables X_i^j are related to specifiers that filter the information needed to copy the tape position. We now define specifiers S_i^j, S_i, for $1 \leq i \leq n$ and $1 \leq j \leq 5$. Intuitively, the next four specifiers are used to increment the tape position, using the information given by the X_i^j variables. The last two specifiers copy the tape position, again using the information given by the X_i^j variables:

– the negation of a concept expression associated with $S_i = \lfloor \mathsf{after} : X_i^1.\mathsf{time}, \mathsf{bit} : 1 \rfloor$ ensures that we have $\mathsf{bit} : 0$ in all time points after the time point marked with $\mathsf{flip} : 1$ in the previous position;
– we use $S_i^1 = \lfloor \mathsf{time} : X_i^1.\mathsf{time}, \mathsf{bit} : 1 \rfloor$ to flip to 1 the bit marked with $\mathsf{flip} : 1$ in the previous position;
– in addition, we define $S_i^2 = \lfloor \mathsf{time} : X_i^2.\mathsf{time}, \mathsf{bit} : 0 \rfloor$ and $S_i^3 = \lfloor \mathsf{time} : X_i^3.\mathsf{time}, \mathsf{bit} : 1 \rfloor$ to transfer to the next tape position bit values which should not be flipped (i.e., those that are before the time point with $\mathsf{flip} : 1$);
– finally, we define $S_i^4 = \lfloor \mathsf{time} : X_i^4.\mathsf{time}, \mathsf{bit} : 0 \rfloor$ and $S_i^5 = \lfloor \mathsf{time} : X_i^5.\mathsf{time}, \mathsf{bit} : 1 \rfloor$, to receive a copy of the bit values.

To simplify the presentation, we define the abbreviations $P_i, P_i^+, P_i^=$ for the following concepts, respectively, to be used in concept inclusions with $\mathbf{\Omega}$:

– $\bigsqcap_{1 \leq j \leq 5} T@X_i^j$, we filter the bits encoding a tape position and the information of which bits should be flipped in order to increment it;
– $\bigsqcap_{1 \leq j \leq 3} T@S_i^j \sqcap \neg T@S_i$, we increment the tape position,
– $\bigsqcap_{4 \leq j \leq 5} T@S_i^j$, we copy the tape position.

We may also write $P, P^+, P^=$ if $i = 1$.

Encoding the initial configuration We add assertions to $\mathcal{O}_{\mathcal{M},w}$ that encode the initial configuration of $\mathcal{M}$. We mark the root of the configuration tree with the initial state by adding $S_{q_0}(a)$ and initialise the tape cells with the input word by adding $I(a)$ and the concept inclusions:

$$\mathbf{\Omega}\ \big(I \sqsubseteq \exists \mathsf{tape}.(H \sqcap C_{\sigma_1} \sqcap T@\lfloor \mathsf{during} : [0, 2^n - 1], \mathsf{bit}: 0 \rfloor)\big)$$
$$\mathbf{\Omega}\ \big(I \sqcap \exists \mathsf{tape}.P_i \sqsubseteq \exists \mathsf{tape}.\big(C_{\sigma_{i+1}} \sqcap P_i^+\big)\big) \text{ for } 1 \leq i < n$$
$$\mathbf{\Omega}\ \big(I \sqcap \exists \mathsf{tape}.P_n \sqsubseteq \exists \mathsf{tape}.\big(C_{\sqcup} \sqcap P_n^+\big)\big)$$

The intuition is as follows. In the first inclusion, we place the head, represented by the concept H, in the first position of the tape and fill the tape cell with the first symbol of the input word, represented by the concept C_{σ_1}. We then add the remaining symbols of the input word in their corresponding tape positions. In the last inclusion we add a blank symbol after the input word. We now add the following concept inclusion fill the remaining tape cells with blank in the initial configuration marked with the concept I:

$$\mathbf{\Omega}\ \big(I \sqcap \exists \mathsf{tape}.(C_{\sqcup} \sqcap P) \sqsubseteq \exists \mathsf{tape}.(C_{\sqcup} \sqcap P^{+})\big)$$

Synchronising configurations For each transition $\theta \in \Theta$, we make sure that tape contents are transferred to successor configurations, except for the tape cell at the head position:

$$\mathbf{\Omega}\ (\exists \mathsf{tape}.(P \sqcap \neg H \sqcap C_{\sigma}) \sqsubseteq \forall r_{\theta}.\exists \mathsf{tape}.(P^{=} \sqcap C_{\sigma}))$$

We now encode our transitions $\theta = (q, \sigma, q', \tau, D) \in \Theta$ with concept inclusions of the form (we explain for $D = r$, the case $D = l$ is analogous):

$$\mathbf{\Omega}\Big(S_q \sqcap \exists \mathsf{tape}.\big(H \sqcap P \sqcap C_{\sigma}\big) \sqcap \exists \mathsf{tape}.\big(P^{+} \sqcap C_{\upsilon}\big) \sqsubseteq$$
$$\exists r_{\theta}.(S_{q'} \sqcap \exists \mathsf{tape}.\big(H \sqcap P^{+} \sqcap C_{\upsilon}\big) \sqcap \exists \mathsf{tape}.(P^{=} \sqcap C_{\tau}))\Big)$$

Essentially, if the head is at position P then, to move it to the right, we increment the head position using P^{+} in the successor configuration. We use the specifiers in $\mathbf{\Omega}$ to modify the tape cell with C_{σ} in the head position to C_{τ} in the successor configuration.

Acceptance Condition Finally, we add concept inclusions that propagate acceptance from the leaf nodes of the configuration tree backwards to the root of the tree. For existential configurations, we add $S_q \sqcap \exists r_{\theta}.A \sqsubseteq A$ for each $q \in Q_{\exists}$, whereas to handle universal configurations, we add, for each $q \in Q_{\forall}$, the concept inclusion

$$S_q \sqcap \exists \mathsf{tape}.\big(C_{\sigma} \sqcap H\big) \sqcap \bigsqcap_{\substack{\theta \in \Theta \\ \theta = (q, \sigma, q', \tau, D)}} \exists r_{\theta}.A \sqsubseteq A$$

where the conjunction may be empty if there are no suitable $\theta \in \Theta$.

With an inductive argument along the recursive definition of acceptance, we show that $\mathcal{O}_{\mathcal{M},w} \models A(a)$ iff $\mathcal{M}$ accepts w.

Given a natural number $i < 2^{2^n}$, we write $i_{\mathbf{b}}[j]$ for the value of the j-th bit of the binary representation of i using 2^n bits, where $i_{\mathbf{b}}[0]$ is the value the most significant bit. In the following, we write B_i as a shorthand for the concept:

$$\bigsqcap_{0 \le y < 2^n} T @ \lfloor \mathsf{bit} : i_{\mathbf{b}}[y], \mathsf{time} : y \rfloor.$$

Following the terminology provided in [25], given an interpretation $\mathcal{I}$ of $\mathcal{ALCH}^{\mathbb{T}}_{@}$, we say that an element $\delta \in \Delta^{\mathcal{I}}_{I}$ *represents* a configuration $\tau_1 \dots \tau_{i-1} q \tau_i \dots \tau_m$ if $(\delta, F) \in S^{\mathcal{I}}_{q}$, for some $F \in \Phi^{\mathcal{I}}$, $\delta \in (\exists \mathsf{tape}.(B_i \sqcap H))^{\mathcal{I}}$ and $\delta \in (\exists \mathsf{tape}.(B_j \sqcap C_{\tau_j}))^{\mathcal{I}}$, for all $0 \le j < 2^{2^n}$. We are now ready to show Claims 1 and 2.

Claim 1 If $\delta \in \Delta^{\mathcal{I}}_{I}$ represents a configuration α and some transition $\theta \in \Theta$ is applicable to α then δ has an r_{θ}-successor that represents the result of applying θ to α.

Proof of Claim 1. Let $\delta \in \Delta_I^{\mathcal{I}}$ be an element representing a configuration α and assume $\theta \in \Theta$ is applicable to α. To synchronise configurations, we added to $\mathcal{O}_{\mathcal{M},w}$ concept inclusions that (1) ensure that tape contents other than the content at the head position are copied to all r_θ-successors of δ; and (2) create an r_θ-successor that represents the correct state, position of the head and corresponding symbols at the previous and current position of the head. Then our concept inclusions ensure that δ has an r_θ-successor that represents the result of applying θ to α.

Claim 2 w is accepted by $\mathcal{M}$ iff $\mathcal{O}_{\mathcal{M},w} \models A(a)$.

Proof of Claim 2. Consider an arbitrary interpretation $\mathcal{I}$ of $\mathcal{ALCH}_{@}^{\mathbb{T}}$ that satisfies $\mathcal{O}_{\mathcal{M},w}$. First we show that if any element $\delta \in \Delta_I^{\mathcal{I}}$ represents an accepting configuration then $(\delta, F) \in A^{\mathcal{I}}$, for some $F \in \Phi^{\mathcal{I}}$. We make a case distinction.

- If α is a universal configuration, then all successor configurations of α must be accepting. By Claim 1, for any θ-successor configuration α' of α there is a corresponding r_θ-successor δ' of δ. By induction hypothesis for α', (δ', F') is in $A^{\mathcal{I}}$, for some $F' \in \Phi^{\mathcal{I}}$. Since this holds for all θ-successor configurations of α, our concept inclusion encoding acceptance of universal configurations implies that $(\delta, F) \in A^{\mathcal{I}}$, for some $F \in \Phi^{\mathcal{I}}$, as required. This argument covers the base case where α has no successors.
- If α is an existential configuration, then there is some accepting θ-successor configuration α' of α. By Claim 1, there is an r_θ-successor δ' of δ that represents α' and, by induction hypothesis, $(\delta', F') \in A^{\mathcal{I}}$, for some $F' \in \Phi^{\mathcal{I}}$. Then, our concept inclusion encoding acceptance of existential configurations applies and so, we conclude that $(\delta, F) \in A^{\mathcal{I}}$, for some $F \in \Phi^{\mathcal{I}}$.

Since elements in $I^{\mathcal{I}}$ represent the initial configuration of $\mathcal{M}$, this shows that $I^{\mathcal{I}} \subseteq A^{\mathcal{I}}$ when the initial configuration is accepting. As $I(a)$ is an assertion in $\mathcal{O}_{\mathcal{M},w}$, we have that $(a^{\mathcal{I}}, G) \in A^{\mathcal{I}}$, for some $G \in \Phi^{\mathcal{I}}$.

We now show that if the initial configuration is not accepting, then there is some interpretation $\mathcal{I}$ of $\mathcal{ALCH}_{@}^{\mathbb{T}}$ such that $I^{\mathcal{I}} \nsubseteq A^{\mathcal{I}}$, in particular, $(a^{\mathcal{I}}, G) \notin A^{\mathcal{I}}$, for all $G \in \Phi^{\mathcal{I}}$. To show this we construct a canonical interpretation $\mathcal{J}$ of $\mathcal{O}_{\mathcal{M},w}$ as follows. Let $\mathsf{Con}_{\mathcal{M}} := \{wqw' \mid |ww'| \leq 2^{2^n}, q \in Q, \{w, w'\} \subseteq \Sigma^*\}$ be the set of all possible $\mathcal{M}$ configurations with size bounded by 2^{2^n}. Also, we define a set $\mathsf{Tp}_{\mathcal{M}} := \{\alpha \cdot c_\sigma^i \mid \alpha \in \mathsf{Con}_{\mathcal{M}}, 0 \leq i < 2^{2^n}, \sigma \in \Sigma\}$, containing individuals that represent tape cells, related to each possible configuration of a computation of $\mathcal{M}$. The domain $\Delta^{\mathcal{J}}$ is a disjoint union of $\Delta_I^{\mathcal{J}} \cup \Delta_T^{\mathcal{J}} \cup \Delta_{2T}^{\mathcal{J}}$, where:

- $\Delta_I^{\mathcal{J}} = \mathsf{Con}_{\mathcal{M}} \cup \mathsf{Tp}_{\mathcal{M}} \cup \mathbb{T}$, where $\mathbb{T} \subseteq \mathsf{N_I}$ is either time or before;
- $\Delta_T^{\mathcal{J}} = \{0^{\mathcal{J}}, \ldots, (2^n - 1)^{\mathcal{J}}\}$; and $\Delta_{2T}^{\mathcal{J}} = \Delta_T^{\mathcal{J}} \times \Delta_T^{\mathcal{J}}$.

The extension of the concepts C_σ, H and B_j in the interpretation is defined as expected so that every element $\alpha \cdot c_\sigma^i \in \mathsf{Tp}_{\mathcal{M}}$ is in C_σ and B_i and no other C_τ or B_j, with $\tau \neq \sigma$ or $i \neq j$. Also, $\alpha \cdot c_\sigma^i$ is in H iff α is of the form wqw' and $|w| = i - 1$. We connect α to $\alpha \cdot c_\sigma^i$ using the role tape iff α has σ at position i. Moreover, α is in S_q iff α is of the form wqw'. We then have that every configuration $\alpha \in \mathsf{Con}_{\mathcal{M}}$ represents itself and no other configuration. $I^{\mathcal{J}}$ is the singleton set containing the initial configuration $a^{\mathcal{J}}$. Given two configurations α and α' and a transition $\theta \in \Theta$, we connect α to α' using the role r_θ iff there is a transition θ from α to α'. Finally, $A^{\mathcal{J}}$ is defined to be the set of tuples (α, F), for some $F \in \Phi^{\mathcal{J}}$, where α is an accepting configuration.

Now, if the initial configuration $a^{\mathcal{J}}$ is not accepting then, by construction, $(a, G) \notin A^{\mathcal{J}}$, for all $G \in \Phi^{\mathcal{J}}$. By checking the concept inclusions in $\mathcal{O}_{\mathcal{M},w}$, we can see that $\mathcal{J}$ satisfies $\mathcal{O}_{\mathcal{M},w}$. Then, $\mathcal{J}$ is a counterexample for $\mathcal{O}_{\mathcal{M},w} \models A(a)$, and so $\mathcal{O}_{\mathcal{M},w} \not\models A(a)$. □

Theorem 7. *In $\mathcal{ALCH}^{\mathbb{T}}_{@}$, any combination of temporal attributes containing* {time, after} *is undecidable. Moreover, the combination* {time, before} *is* 3ExpTime*-complete, and the combination* {time, during, since, until} *and every subset of it are* 2ExpTime*-complete.*

Proof. The proof of Theorem 1 uses only the temporal attributes time and after. Thus, any combination containing these attributes is Σ^1_1-hard. By Theorems 5 and 6 the combination {time, before} is 3ExpTime-complete. It remains to show that the combination {time, during, since, until} is in 2ExpTime (since 2ExpTime-hardness is already known for $\mathcal{ALCH}_@$ [23]).

Our proof strategy consists in showing that, given an $\mathcal{ALCH}^{\mathbb{T}}_{@}$ interpretation and an $\mathcal{ALCH}^{\mathbb{T}}_{@}$ ontology that contains only the temporal attributes in {time, during, since, until}, one can always transform this interpretation so that only time points explicitly mentioned in the ontology are relevant to determine if the interpretation is a model of the ontology. Then one can check satisfiability by grounding the ontology using only those time points explicitly mentioned. We start by providing some notation.

Given an $\mathcal{ALCH}^{\mathbb{T}}_{@}$ ontology $\mathcal{O}$, we define a set $\mathbb{N}_{\mathcal{O}}$ as in Theorem 3, except that we do not need $k_i + 1$ here. To this end, let $k_0 < \ldots < k_n$ be the ascending sequence of all numbers mentioned in time points or in time intervals (as endpoints) in $\mathcal{O}$. We define $\mathbb{N}_{\mathcal{O}}$ as $\{k_i \mid 0 \leq i \leq n\}$, and let $k_{\mathsf{min}} := \min(\mathbb{N}_{\mathcal{O}})$ and $k_{\mathsf{max}} = \max(\mathbb{N}_{\mathcal{O}})$, where we assume $k_{\mathsf{min}} = k_{\mathsf{max}} = 0$ if $\mathbb{N}_{\mathcal{O}} = \emptyset$.

Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be an $\mathcal{ALCH}^{\mathbb{T}}_{@}$ interpretation. By Definition 8, $\mathcal{I}$ is a global interpretation of a sequence $(\mathcal{I}_i)_{i \in \Delta^{\mathcal{I}}_T}$ of $\mathcal{ALCH}_@$ interpretations with domain $\Delta^{\mathcal{I}}_I$. We now define a sequence $(\mathcal{J}_i)_{i \in \Delta^{\mathcal{J}}_T}$ of $\mathcal{ALCH}_@$ interpretations as follows. Let $\Delta^{\mathcal{J}}_I = \Delta^{\mathcal{I}}_I$ and let $\Delta^{\mathcal{J}}_T = \{k^{\mathcal{J}}_{\mathsf{min}}, \ldots, k^{\mathcal{J}}_{\mathsf{max}}\}$. For all $A \in \mathsf{N_C}$, all $F \in \Phi^{\mathcal{I}}$ with $F \setminus F_I \neq \emptyset$ and $k \in [k_{\mathsf{min}}, k_{\mathsf{max}}]$:

$$(\delta, F_I) \in A^{\mathcal{J}_k} \text{ iff } (\delta, F_I) \in A^{\mathcal{I}_k}$$

and either:

(1) $k \in \mathbb{N}_{\mathcal{O}}$; or
(2) there is $k_i < k$ such that $k_i \in \mathbb{N}_{\mathcal{O}}$ and $(\delta, F_I) \in A^{\mathcal{I}_j}$ for all $k_i \leq j \leq k_{\mathsf{max}}$; or
(3) there is $k_i > k$ such that $k_i \in \mathbb{N}_{\mathcal{O}}$ and $(\delta, F_I) \in A^{\mathcal{I}_j}$ for all $k_{\mathsf{min}} \leq j \leq k_i$.

We analogously apply the definition above for all role names $r \in \mathsf{N_R}$. We define $\mathcal{I}_{\mathcal{O}}$ as a global interpretation of the sequence $(\mathcal{J}_i)_{i \in \Delta^{\mathcal{J}}_T}$ and set $(\delta, F) \in A^{\mathcal{I}_{\mathcal{O}}}$ iff $(\delta, F) \in A^{\mathcal{I}}$ for all $A \in \mathsf{N_C}$ with $F = F_I$, and similarly for all role names $r \in \mathsf{N_R}$. Let $\mathcal{O}_{\mathsf{g}}$ be the result of grounding $\mathcal{O}$ in the same way as in the proof of Theorem 4 using time points in $\mathbb{N}_{\mathcal{O}}$ (here $\mathcal{O}$ may have expressions of the form $X.a$, $a\colon x$, or $a\colon[t, t']$, with $a \in$ {time, during, since, until}, and $t, t' \in \mathsf{N_T} \cup \mathsf{Var}(\mathsf{N_T})$).

Claim. For all $A@S, r@S$ occurring in $\mathcal{O}_{\mathsf{g}}$: $A@S^{\mathcal{I}_{\mathcal{O}}} = A@S^{\mathcal{I}}$ and $r@S^{\mathcal{I}_{\mathcal{O}}} = r@S^{\mathcal{I}}$.

Proof of the Claim. This claim follows by definition of $(\mathcal{J}_i)_{i\in\Delta_T^{\mathcal{J}}}$ and the fact that only the temporal attributes $\{\mathsf{time}, \mathsf{during}, \mathsf{since}, \mathsf{until}\}$ are allowed. Correctness for the temporal attributes time and during follows from item (1), whereas correctness for the temporal attributes since and until follows from items (2) and (3), respectively.

By definition of $\mathcal{O}_{\mathsf{g}}$, we know that $\mathcal{O} \models \mathcal{O}_{\mathsf{g}}$. So if $\mathcal{O}$ is satisfiable then $\mathcal{O}_{\mathsf{g}}$ is satisfiable. Conversely, by the Claim, one can show with an inductive argument that $C^{\mathcal{I}_{\mathcal{O}}} = C^{\mathcal{I}}$ for all $\mathcal{ALCH}^{\mathbb{T}}_{@}$ concepts C occurring in $\mathcal{O}_{\mathsf{g}}$. So, if an $\mathcal{ALCH}^{\mathbb{T}}_{@}$ interpretation $\mathcal{I}$ satisfies $\mathcal{O}_{\mathsf{g}}$ then $\mathcal{I}_{\mathcal{O}}$ satisfies $\mathcal{O}$. Since $\mathcal{O}_g$ is at most exponentially larger than $\mathcal{O}$, it follows that satisfiability in this fragment is in 2ExpTime. □