Happy Ever After: Temporally Attributed Description Logics

Ana Ozaki [a], Markus Krötzsch [b], Sebastian Rudolph [c]

[a] KRDB Research Centre, Free University of Bozen-Bolzano, Italy
[b] Center for Advancing Electronics Dresden (cfaed), TU Dresden, Germany
[c] Computational Logic Group, TU 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*. Based on recently proposed 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 [24] or YAGO2 [13], are based on *enriched* graphs where edges are augmented with additional annotations. To model these enriched graphs, *attributed logics* have been proposed as a way of integrating annotations with logical reasoning [20,14,15]. Other formalisms for reasoning over annotated relations have been studied in the context of data modelling [6].

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 5.4 million statements overall.[1] In YAGO2, time and space are the main types of annotations considered.

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 [9,10], and similar models of time have also been studied in ontologies [2,16]. However, researchers in ontologies have most commonly focussed on abstract models of time as

[1] As of October 2018, 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`

used in temporal logics [19,25,5]. Temporal reasoning in $\mathcal{ALC}$ with concrete domains was proposed by Lutz et. al [18]. 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 [17]. 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 [7]. 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 1. The fact that Johannes Gutenberg died in Mainz in 1468 could be encoded in attributed DLs as:

$$\mathsf{diedIn}(\mathsf{Gutenberg}, \mathsf{Mainz})@[\mathsf{time}\colon 1468]$$

Not all annotations are temporal, and we can also annotate concept assertions, e.g., to state that he lived in Strasbourg:

$$\mathsf{Lived}(\mathsf{Gutenberg})@[\mathsf{loc}\colon \mathsf{Strasbourg}]$$

Gutenberg's early life is less certain, and we only know that he was born between 1394 and 1404 in Mainz. Such uncertainty about precise dates is very common in practice. Nevertheless, we would like to record the information available, which could be expressed as follows:

$$\mathsf{bornIn}(\mathsf{Gutenberg}, \mathsf{Mainz})@[\mathsf{between} : [1394, 1404]]$$

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 $\mathcal{ALCH}^{\mathbb{T}}_{@}$, enabling reasoning and querying support for such information. Beyond defining syntax and semantics of $\mathcal{ALCH}^{\mathbb{T}}_{@}$, this paper's contributions are the following:

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

The paper is self-contained. Details on some long proofs can be found in the appendix of the extended online version [21].

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 [14] , given as attribute-value pairs. As suggested in Example 1, the same relationship may be true with several different annotation sets, e.g., in case Gutenberg also lived elsewhere.

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.

Axioms of $\mathcal{ALCH}^{\mathbb{T}}_{@}$ are further based on sets of *concept names* $\mathsf{N_C}$, *role names* $\mathsf{N_R}$, and *(set) variables* $\mathsf{N_V}$. Attributes are represented by individual names, and we associate a *value type* $\mathsf{vt}(a)$ 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 $v \in \mathsf{vt}(a)$. Now, concept and role assertions of $\mathcal{ALCH}^{\mathbb{T}}_{@}$ have the following form:

$$C(a)@[a_1\colon v_1, \ldots, a_n\colon v_n] \qquad r(a, b)@[a_1\colon v_1, \ldots, a_n\colon v_n]$$

where $C \in \mathsf{N_C}$, $r \in \mathsf{N_R}$, $a, b \in \mathsf{N_I}$, and $a_i\colon v_i$ are attribute-value pairs.

Role and concept inclusion axioms of $\mathcal{ALCH}^{\mathbb{T}}_{@}$ introduce additional expressive power to refer to partially specified and variable annotation sets. An *(annotation set) specifier* can be a set variable $X \in \mathsf{N_V}$, a *closed specifier* $[a_1\colon v_1, \ldots, a_n\colon v_n]$, or an *open specifier* $\lfloor a_1\colon v_1, \ldots, a_n\colon v_n \rfloor$, where $a_i \in \mathsf{N_I}$ and either $v_i \in \mathsf{vt}(a_i)$ or $v_i = X.b$ with $X \in \mathsf{N_V}$, $b \in \mathsf{N_I}$, and $\mathsf{vt}(a_i) = \mathsf{vt}(b)$. Intuitively speaking, closed specifiers define specific annotation sets whereas open specifiers merely provide lower bounds. The notation $X.b$ is used to copy all of the zero or more b-values of annotation set X to a new annotation set. The set of all specifiers is denoted $\mathbf{S}$. A specifier is *ground* if it does not contain variables. $\mathcal{ALCH}^{\mathbb{T}}_{@}$ *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:

$$C, D ::= \top \mid A@S \mid \neg C \mid (C \sqcap D) \mid \exists R.C \tag{1}$$

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 DL inclusions between $\mathcal{ALCH}^{\mathbb{T}}_{@}$ role and concept expressions, which may, however, share variables.

Example 2. In an ontology containing biographical information, we might want to make sure that children cannot be born before their parents. This can be expressed by the axiom

$$\exists \mathsf{bornIn}@X.\top \sqsubseteq \neg\exists \mathsf{hasChild}@\lfloor\,\rfloor.\exists \mathsf{bornIn}@\lfloor \mathsf{before}\colon X.\mathsf{time} \rfloor.\top.$$

Similar axioms can be used, e.g., to state that nobody has more than one birthday ("is born before being born").

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. Hence we define an $\mathcal{ALCH}^{\mathbb{T}}_{@}$ *concept inclusion* as an expression of the form

$$X_1 : S_1, \ldots, X_n : S_n \quad (C \sqsubseteq D), \tag{2}$$

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 $S_1, \ldots, S_n$. $\mathcal{ALCH}^{\mathbb{T}}_{@}$ *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. To simplify notation, we sometimes omit the specifier $\lfloor\,\rfloor$ (meaning "any annotation set") in role or concept expressions. In this sense, any $\mathcal{ALCH}$ axiom is also an $\mathcal{ALCH}^{\mathbb{T}}_{@}$ axiom.

3 Formal Semantics

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 $\Phi^{\mathcal{I}} := \mathcal{P}_{\mathsf{fin}}\left(\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}\right)$ of all finite binary relations over $\Delta^{\mathcal{I}}$. Now 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. Note that each element (pair of elements) may appear with multiple different annotations. $\mathcal{I}$ *satisfies* a concept assertion $C(a)@[a_1 : v_1, \ldots, a_n : v_n]$ 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. Expressions with free set variables are interpreted using variable assignments $\mathcal{Z} : \mathsf{N_V} \to \Phi^{\mathcal{I}}$. 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) $[a : b]^{\mathcal{I},\mathcal{Z}} := \{\{(a^{\mathcal{I}}, b^{\mathcal{I}})\}\}$
(ii) $[a : X.b]^{\mathcal{I},\mathcal{Z}} := \{\{(a^{\mathcal{I}}, \delta) \mid (b^{\mathcal{I}}, \delta) \in \mathcal{Z}(X)\}\}$
(iii) $[a_1 : v_1, \ldots, a_n : v_n]^{\mathcal{I},\mathcal{Z}} := \{\bigcup_{i=1}^{n} F_i\}$ where $\{F_i\} = [a_i : v_i]^{\mathcal{I},\mathcal{Z}}$ 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 $\lfloor a_1 : v_1, \ldots, a_n : v_n \rfloor^{\mathcal{I},\mathcal{Z}}$ to be the set

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

Now given $A \in \mathsf{N_C}$, $r \in \mathsf{N_R}$, and $S \in \mathbf{S}$, we define:

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

Further DL expressions are 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 $(\exists R.C)^{\mathcal{I},\mathcal{Z}} = \{\delta \mid \text{there is } (\delta,\epsilon) \in R^{\mathcal{I},\mathcal{Z}} \text{ with } \epsilon \in C^{\mathcal{I},\mathcal{Z}}\}$.

$\mathcal{I}$ *satisfies* a concept inclusion of the form (2) 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.

Adding Time Time points $t \in \mathsf{N_T}$ are encodings of natural numbers, which we denote by $t^{\mathcal{I}}$. Analogously, $v^{\mathcal{I}}$ denotes a pair of numbers for a time interval $v \in \mathsf{N_T^2}$. To represent time, we consider intervals of natural numbers, which can be finite intervals $[k,\ell] = \{n \in \mathbb{N} \mid k \leq n \leq \ell\}$ or infinite intervals $[k,\infty) = \{n \in \mathbb{N} \mid k \leq n\}$. A *temporal domain* $\Delta_T^{\mathcal{I}}$ is a finite or infinite interval such that $t^{\mathcal{I}} \in \Delta_T^{\mathcal{I}}$ for all $t \in \mathsf{N_T}$ and $v^{\mathcal{I}} \in \Delta_T^{\mathcal{I}} \times \Delta_T^{\mathcal{I}}$ for all $v \in \mathsf{N_T^2}$. Note that $\Delta_T^{\mathcal{I}}$ 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).

A *time-sorted interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ has a sorted domain $\Delta^{\mathcal{I}}$ that is a disjoint union $\Delta_I^{\mathcal{I}} \cup \Delta_T^{\mathcal{I}} \cup \Delta_{2T}^{\mathcal{I}}$, where $\Delta_I^{\mathcal{I}}$ is the *abstract domain*, $\Delta_T^{\mathcal{I}}$ is a temporal domain, and $\Delta_{2T}^{\mathcal{I}} = \Delta_T^{\mathcal{I}} \times \Delta_T^{\mathcal{I}}$. We interpret individual names $a \in \mathsf{N_I}$ as elements $a^{\mathcal{I}} \in \Delta_I^{\mathcal{I}}$. A pair $(\delta,\epsilon) \in \Delta_I^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ is *well-typed*, if one of the following holds:

(i) $\delta = a^{\mathcal{I}}$ for a temporal attribute a of value type $\mathsf{N_T}$ and $\epsilon \in \Delta_T^{\mathcal{I}}$;
(ii) $\delta = a^{\mathcal{I}}$ for a temporal attribute a of value type $\mathsf{N_T^2}$ and $\epsilon \in \Delta_{2T}^{\mathcal{I}}$; or
(iii) $\delta \neq a^{\mathcal{I}}$ for all temporal attributes a and $\epsilon \in \Delta_I^{\mathcal{I}}$.

Then $\Phi^{\mathcal{I}}$ is the set of all finite sets of well-typed pairs. The remainder of the interpretation function is defined as in the unsorted case, based on this sorted definition of $\Phi^{\mathcal{I}}$.

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 3. *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 $\mathcal{ALCH}_{@}$ 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 multisorted 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 $F_I := F \cap (\Delta_I^{\mathcal{I}} \times \Delta_I^{\mathcal{I}})$ 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*[2] *$(\delta, F_I) \in A^{\mathcal{I}_i}$ for some $i \in \Delta_T^{\mathcal{I}}$, and the following conditions hold for all $(a^{\mathcal{I}}, x) \in F$:*

[2] 'for some $i \in \Delta_T^{\mathcal{I}}$' is useful for attributes which universally quantify time points (e.g., until).

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

where $[x]$ *for an element* $x \in \Delta^{\mathcal{I}}_{2T}$ *denotes the finite interval represented by the pair of numbers* x*, and* $j \in \Delta^{\mathcal{I}}_{T}$*. 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 3 holds vacuously for any temporal sequence, meaning that without temporal attributes the semantics is essentially the same as for $\mathcal{ALCH}_{@}$.

Definition 4. *An* interpretation *of* $\mathcal{ALCH}^{\mathbb{T}}_{@}$ *is a time-sorted interpretation* $\mathcal{I}$ *that is a global interpretation of an interpretation sequence* $(\mathcal{I}_i)_{i \in \Delta^{\mathcal{I}}_{T}}$ *as in Definition 3.*

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 5. Along the lines of Example 2, we can state, e.g., that people cannot live after their death:

$$\mathsf{Lived}@X \sqcap \exists \mathsf{diedIn}@\lfloor \mathsf{before}\colon X.\mathsf{time} \rfloor.\top \sqsubseteq \bot \qquad \exists \mathsf{bornIn}@X.\top \sqsubseteq \mathsf{Lived}@X$$

With these background axioms in place, we can infer from the time-annotated facts in Example 1, e.g.,

$$\mathsf{Lived}(\mathsf{Gutenberg})@[\mathsf{between}\colon[1394,1468]]$$

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 6. 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 until: 10 with during: [0, 10] would therefore lead to an inconsistent ontology.

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

In this section, we study the expressivity and decidability in $\mathcal{ALCH}^{\mathbb{T}}_{@}$. Our first result, Theorem 7, shows that reasoning is on the first level of the analytical hierarchy and therefore highly undecidable.

Theorem 7. *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 [12]: 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):

$$A@X \sqsubseteq \neg A@\lfloor \mathsf{after}\colon X.\mathsf{time}\rfloor \tag{3}$$

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. 3). 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: $I \sqsubseteq B$ 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 8 shows that even if after is only allowed in assertions reasoning is undecidable, though, in the arithmetical hierarchy [22].

Theorem 8. *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 (recall Σ^0_1 stands for* RE*). The problem is Σ^0_1-hard even with at most one set variable per inclusion.*

To recover decidability, we need to restrict $\mathcal{ALCH}^{\mathbb{T}}_{@}$ in some way. A simple approach of doing so is to consider ground $\mathcal{ALCH}^{\mathbb{T}}_{@}$ where we disallow set variables altogether. It is clear from the known complexity of $\mathcal{ALCH}$ that reasoning is 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 [23].

Theorem 9. *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 $\mathbb{N}_{\mathcal{O}}^{v}$ 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 $\mathbb{N}_{\mathcal{O}}^{v}$ 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):

$$A_S \equiv \bigsqcap_{(a:b)\in S_{\mathbb{T}}} (A_{S(a:b)})^{\sharp} \quad \text{and} \quad r_S \equiv \bigsqcap_{(a:b)\in S_{\mathbb{T}}} (r_{S(a:b)})^{\sharp} \tag{4}$$

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

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

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 (4) 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 (4), 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 [14], and indeed we can regain decidability in $\mathcal{ALCH}^{\mathbb{T}}_{@}$ by disallowing expressions of the form $X.a$ to be used with temporal attributes a. Indeed, 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.

Theorem 10. *Satisfiability of $\mathcal{ALCH}^{\mathbb{T}}_{@}$ ontologies without expressions of the form $X.a$ for temporal attributes a is* 2ExpTime*-complete.*

Another way for regaining decidability is by limiting the temporal attributes that make reference to time points in the future. Using this assumption, we can translate any $\mathcal{ALCH}^{\mathbb{T}}_{@}$ ontology into a ground $\mathcal{ALCH}^{\mathbb{T}}_{@}$ ontology. In this case, however, the resulting ground ontology is double-exponentially larger if we assume that the size of the temporal domain has been encoded in binary. We therefore obtain a 3ExpTime upper bound (using Theorem 9).

Theorem 11. *Satisfiability of $\mathcal{ALCH}^{\mathbb{T}}_{@}$ ontologies with only the temporal attributes* during, time, before *and* until *is in* 3ExpTime.

Our result in our next Theorem 12 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) [8] 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 12. *Satisfiability of $\mathcal{ALCH}^{\mathbb{T}}_{@}$ ontologies with only* time *and* before *is* 3ExpTime*-hard.*

5 Lightweight Temporal Attributed DLs

In this section we investigate $\mathcal{ELH}^{\mathbb{T}}_{@}$, the fragment of $\mathcal{ALCH}^{\mathbb{T}}_{@}$ which uses only $\exists$, $\sqcap$, $\top$ and $\bot$ in concept expressions. Though, even for ground ontologies, the satisfiability problem for $\mathcal{ELH}^{\mathbb{T}}_{@}$ is not tractable.

Theorem 13. *Satisfiability of ground $\mathcal{ELH}^{\mathbb{T}}_{@}$ ontologies is* ExpTime*-complete.*

Proof. The upper bound follows from Theorem 9. 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 $\top \sqsubseteq B \sqcup C$. As an example, see the following inclusions using the temporal attributes time and between: $\top \sqsubseteq A@\lfloor \mathsf{between} : [1,2] \rfloor$, $A@\lfloor \mathsf{time} : 1 \rfloor \sqsubseteq B$, $A@\lfloor \mathsf{time} : 2 \rfloor \sqsubseteq C$. One can also obtain the same type of encoding with before and after: $\top \sqsubseteq A@\lfloor\ \rfloor$, $A@\lfloor \mathsf{before} : 1 \rfloor \sqsubseteq B$, $A@\lfloor \mathsf{after} : 0 \rfloor \sqsubseteq C$.

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 [16]. 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 [23].

Theorem 14. *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 $\mathcal{EL}$ [4]. For membership, note that the translation in Theorem 9 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.

6 Discussion and 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. Some of the statements used in our examples can also be naturally expressed in temporal DLs. For instance, the first statement of Example 5 is expressible by $\mathcal{ALC}$ extended with Linear Temporal Logic [19,25] with:

$$\mathsf{Lived} \sqcap \Diamond\exists\mathsf{bornIn}.\top \sqsubseteq \bot.$$

Other authors have also considered extending $\mathcal{ALC}$ with Metric Temporal Logic (MTL) [11,3], where the last statement of Example 1 can be expressed with:

$$\Diamond_{[1394,1404]}\mathsf{bornIn}(\mathsf{Gutenberg},\mathsf{Mainz}).$$

However, the statement in Example 2 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 9 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 $\mathcal{ALC}$. 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 and spatial information.

Acknowledgements This work was partially supported by the DFG within the cfaed Cluster of Excellence, CRC 912 (HAEC), and Emmy Noether grant KR 4381/1-1, and by the ERC in Consolidator Grant 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, 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.
3. 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.
4. 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.
5. Franz Baader, Silvio Ghilardi, and Carsten Lutz. LTL over description logic axioms. *ACM Trans. Comput. Log.*, 13(3), 2012.
6. Daniela Berardi, Diego Calvanese, and Giuseppe De Giacomo. Reasoning on UML class diagrams. *Artif. Intell.*, 168(1-2):70–118, 2005.
7. 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.
8. Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. Alternation. *J. of the ACM*, 28(1):114–133, 1981.
9. 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.
10. Michael David Fisher, Dov M. Gabbay, and Lluis Vila, editors. *Handbook of Temporal Reasoning in Artificial Intelligence*. Elsevier, 2005.
11. 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.
12. David Harel. Effective transformations on infinite trees, with applications to high undecidability, dominoes, and fairness. *J. ACM*, 33(1):224–248, 1986.
13. 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.
14. 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.
15. 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.
16. 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.
17. Carsten Lutz. Combining interval-based temporal reasoning with general tboxes. *Artificial Intelligence*, 152(2):235 – 274, 2004.
18. Carsten Lutz, Volker Haarslev, and Ralf Möller. A concept language with role-forming predicate restrictions. Technical report, University of Hamburg, 1997.
19. 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.

20. 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.
21. Ana Ozaki, Markus Krötzsch, and Sebastian Rudolph. Happy ever after: Temporally attributed description logics (extended technical report). Available online at `https://iccl.inf.tu-dresden.de/web/Inproceedings3101`, 2018.
22. Hartley Rogers, Jr. *Theory of Recursive Functions and Effective Computability*. MIT Press, paperback edition edition, 1987.
23. 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.
24. Denny Vrandečić and Markus Krötzsch. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10), 2014.
25. Frank Wolter and Michael Zakharyaschev. Temporalizing description logics. *Frontiers of Combining Systems*, 2:379–402, 1999.