Metric Temporal Description Logics with Interval-Rigid Names (Extended Abstract)*

Franz Baader, Stefan Borgwardt, Patrick Koopmann, Ana Ozaki, Veronika Thost

Institute of Theoretical Computer Science and cfaed, TU Dresden, Germany
firstname.lastname@tu-dresden.de

Abstract In contrast to qualitative linear temporal logics, which can be used to state that some property will eventually be satisfied, metric temporal logics allow to formulate constraints on how long it may take until the property is satisfied. While most of the work on combining Description Logics (DLs) with temporal logics has concentrated on qualitative temporal logics, there has recently been a growing interest in extending this work to the quantitative case. In this paper, we complement existing results on the combination of DLs with metric temporal logics over the natural numbers by introducing interval-rigid names. This allows to state that elements in the extension of certain names stay in this extension for at least some specified amount of time.

1 Introduction

DL-based ontologies are employed in many application areas, but they are particularly successful in the medical domain (see, e.g., the medical ontologies Galen and SNOMED CT[1]). For example, the concept of a patient with a concussion can be expressed as $\mathsf{Patient} \sqcap \exists \mathsf{finding}.\mathsf{Concussion}$. This example, taken from [10], can be used to illustrate a shortcoming of pure DLs. For a doctor, it is important to know whether the concussed patient has lost consciousness, which is the reason why SNOMED CT contains a concept for “concussion with no loss of consciousness” [19]. However, the temporal pattern inherent in this concept (after the concussion, the patient remained conscious until the examination) cannot be modeled in the DL used for SNOMED CT.

To overcome this problem, a great variety of temporal extensions of DLs have been investigated.[2] In the present paper, we concentrate on $\mathcal{ALC}$ and combine it with a metric variant of linear temporal logic (LTL), a point-based temporal logic over a linear flow of time. But even if these two logics are fixed, there are several other design decisions to be made. One can either apply temporal operators only to axioms [10] (i.e., general concept inclusions (GCIs) and assertions) or also use them within concepts [15, 20]. With the latter, one can formalize “concussion

* Supported by DFG in the CRC 912 (HAEC), the project BA 1122/19-1 (GoAsQ) and the Cluster of Excellence “Center for Advancing Electronics Dresden” (cfaed).

[1] see http://www.opengalen.org/ and http://www.snomed.org/

[2] We refer the reader to [15, 17] for an overview of the field of temporal DLs.

with no loss of consciousness" by the (temporal) concept $\exists\mathsf{finding}.\mathsf{Concussion} \sqcap (\mathsf{Conscious}\ \mathcal{U}\ \exists\mathsf{procedure}.\mathsf{Examination})$, where $\mathcal{U}$ is the *until*-operator of LTL. With the logic of [10], one cannot formulate temporal concepts, but could express that a particular patient, e.g., Bob, had a concussion and did not lose consciousness until he was examined. Another decision to be made is whether to allow for *rigid concepts and roles*, whose interpretation does not vary over time. For example, concepts like Human and roles like $\mathsf{hasFather}$ are clearly rigid, whereas $\mathsf{Conscious}$ and $\mathsf{finding}$ are flexible, i.e., not rigid. If temporal operators can be used within concepts, rigid concepts can be expressed using GCIs, but rigid roles cannot. In fact, they usually render the combined logic undecidable [15, Proposition 3.34]. In contrast, in the setting considered in [10], rigid roles do not cause undecidability, but adding rigidity leads to an increase in complexity.

In this paper, we address a shortcoming of the qualitative temporal description logics mentioned until now. The until-operator in our example does not say anything about how long after the concussion that examination happened. However, the above definition of "concussion with no loss of consciousness" is only sensible if the examination took place shortly after the concussion. Otherwise, a loss of consciousness could also have been due to other causes. As another example, when formulating eligibility criteria for clinical trials, one needs to express quantitative temporal patterns [13] like 'patients that had a treatment causing a reaction between 45 and 180 days after the treatment, and had no additional treatment before the reaction': $\mathsf{Treatment} \sqcap \bigcirc\big((\neg\mathsf{Treatment})\,\mathcal{U}_{[45,180]}\mathsf{Reaction}\big)$, where $\bigcirc$ is the *next*-operator. Extensions of LTL by such intervals have been investigated in detail [1, 2, 16]. Using the next-operator of LTL as well as disjunction, their effect can actually be simulated within qualitative LTL, but if the interval boundaries are encoded in binary, this leads to an exponential blowup. The complexity results in [1] imply that this blowup can in general not be avoided, but in [16] it is shown that using intervals of a restricted form (where the lower bound is 0) does not increase the complexity compared to the qualitative case. In [14], the combination of the DL $\mathcal{ALC}$ with a metric extension of LTL is investigated. The paper considers both the case where temporal operators are applied only within concepts and the case where they are applied both within concepts and outside of GCIs. In Section 2, we basically recall some of the results obtained in [14], but show that they also hold if additionally temporalized assertions are available.

In Section 3, we extend the logic $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ of Section 2 with *interval-rigid* names, a means of expressiveness that has not been considered before. It allows one to state that elements belonging to a concept belong to that concept for at least k consecutive time points, and similarly for roles. For example, according to the WHO, patients with paucibacillary leprosy should receive MDT as treatment for 6 consecutive months,[3] which can be expressed by making the role $\mathsf{getMDTagainstPB}$ rigid for 6 time points (assuming that each time point represents one month). In Section 4, we briefly discuss results for extensions of the logic $\mathcal{ALC}$-LTL of [10] with interval-rigid concepts and roles as well as metric temporal operators, where temporal operators can only be applied to axioms.

[3] see `http://www.who.int/lep/mdt/duration/en/`.

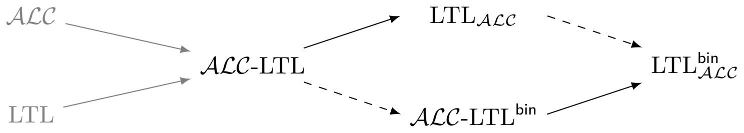
$\mathcal{ALC}$
LTL
$\mathcal{ALC}$-LTL
$\mathrm{LTL}_{\mathcal{ALC}}$
$\mathcal{ALC}$-$\mathrm{LTL}^{\mathsf{bin}}$
$\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$

Figure 1. Language inclusions, with languages investigated in this paper highlighted. Dashed arrows indicate same expressivity.

Interestingly, in the presence of rigid roles, interval-rigid concepts actually cause undecidability. Without rigid roles, the addition of interval-rigid concepts and roles leaves the logic decidable, but in some cases increases the complexity (see Table 2). An overview of the logics considered and their relations is shown in Figure 1. Detailed proofs of all results can be found in [7, 8].

Related Work. Apart from the above references, we want to point out work on combining DLs with Halpern and Shoham's interval logic [3, 4]. This setting uses intervals (rather than time points) as the basic time units. In [6], the authors combine $\mathcal{ALC}$ concepts with the (qualitative) operators $\Diamond$ ('at some time point') and $\Box$ ('at all time points') on roles, but do not consider quantitative variants. Recently, a metric temporal extension of Datalog over the reals was proposed, which however cannot express interval-rigid names nor existential restrictions [12].

2 The Temporal Description Logic $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$

We first introduce the description logic $\mathcal{ALC}$ and its metric temporal extension $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ [14], which augments $\mathcal{ALC}$ by allowing metric temporal logic operators [1] both within $\mathcal{ALC}$ axioms and to combine these axioms. We actually consider a slight extension of $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ by assertional axioms.

Syntax. Let $\mathsf{N_C}$, $\mathsf{N_R}$ and $\mathsf{N_I}$ be countably infinite sets of *concept names*, *role names*, and *individual names*, respectively. An $\mathcal{ALC}$ *concept* is an expression given by $C, D ::= A \mid \top \mid \neg C \mid C \sqcap D \mid \exists r.C$, where $A \in \mathsf{N_C}$ and $r \in \mathsf{N_R}$. $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ *concepts* extend $\mathcal{ALC}$ concepts with the constructors $\bigcirc C$ and $C\,\mathcal{U}_I D$, where I is an interval of the form $[c_1, c_2]$ or $[c_1, \infty)$ with $c_1, c_2 \in \mathbb{N}$, $c_1 \leq c_2$, given in *binary*. We may use $[c_1, c_2)$ to abbreviate $[c_1, c_2 - 1]$, and similarly for the left endpoint. For example, $A\,\mathcal{U}_{[2,5)} B \sqcap \exists r.\bigcirc A$ is an $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ concept.

An $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ *axiom* is either a *general concept inclusion (GCI)* of the form $C \sqsubseteq D$, or an *assertion* of the form $C(a)$ or $r(a, b)$, where C, D are $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ concepts, $r \in \mathsf{N_R}$, and $a, b \in \mathsf{N_I}$. $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ *formulae* are expressions of the form $\phi, \psi ::= \alpha \mid \top \mid \neg\phi \mid \phi \wedge \psi \mid \bigcirc\phi \mid \phi\,\mathcal{U}_I \psi$, where α is an $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ axiom.

Semantics. A DL *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ over a non-empty set $\Delta^{\mathcal{I}}$, called the *domain*, defines an *interpretation function* $\cdot^{\mathcal{I}}$ that maps each concept name $A \in \mathsf{N_C}$ to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, each role name $r \in \mathsf{N_R}$ to a binary relation $r^{\mathcal{I}}$ on $\Delta^{\mathcal{I}}$ and each individual name $a \in \mathsf{N_I}$ to an element $a^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, such that

$a^{\mathcal{I}_i} \neq b^{\mathcal{I}_i}$ whenever $a \neq b$, $a, b \in \mathsf{N_I}$ (*unique name assumption*). As usual, we extend the mapping $\cdot^{\mathcal{I}}$ from concept names to $\mathcal{ALC}$ concepts as follows:

$$\top^{\mathcal{I}_i} := \Delta^{\mathfrak{I}}, \qquad (\neg C)^{\mathcal{I}_i} := \Delta^{\mathfrak{I}} \setminus C^{\mathcal{I}_i}, \qquad (C \sqcap D)^{\mathcal{I}_i} := C^{\mathcal{I}_i} \cap D^{\mathcal{I}_i},$$
$$(\exists r.C)^{\mathcal{I}_i} := \{d \in \Delta^{\mathfrak{I}} \mid \exists e \in C^{\mathcal{I}_i} : (d, e) \in r^{\mathcal{I}_i}\}.$$

A *(temporal DL) interpretation* is a structure $\mathfrak{I} = (\Delta^{\mathfrak{I}}, (\mathcal{I}_i)_{i \in \mathbb{N}})$, where each $\mathcal{I}_i = (\Delta^{\mathfrak{I}}, \cdot^{\mathcal{I}_i})$, $i \in \mathbb{N}$, is a DL interpretation over $\Delta^{\mathfrak{I}}$ (*constant domain assumption*) and $a^{\mathcal{I}_i} = a^{\mathcal{I}_j}$ for all $a \in \mathsf{N_I}$ and $i, j \in \mathbb{N}$, i.e., the interpretation of individual names is fixed. The mappings $\cdot^{\mathcal{I}_i}$ are extended to $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ concepts as follows:

$$(\bigcirc C)^{\mathcal{I}_i} := \{d \in \Delta^{\mathfrak{I}} \mid d \in C^{\mathcal{I}_{i+1}}\},$$
$$(C\,\mathcal{U}_I D)^{\mathcal{I}_i} := \{d \in \Delta^{\mathfrak{I}} \mid \exists k\colon k - i \in I,\ d \in D^{\mathcal{I}_k},\ \text{and } \forall j \in [i, k)\colon d \in C^{\mathcal{I}_j}\}.$$

The concept $C\,\mathcal{U}_I D$ requires D to be satisfied at some point in the interval I, and C to hold at all time points before that.

The *validity* of an $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ formula ϕ in $\mathfrak{I}$ at time point $i \in \mathbb{N}$ (written $\mathfrak{I}, i \models \phi$) is inductively defined as follows:

$$\begin{array}{llll}
\mathfrak{I}, i \models C \sqsubseteq D & \text{iff } C^{\mathcal{I}_i} \subseteq D^{\mathcal{I}_i} & \mathfrak{I}, i \models \phi \wedge \psi & \text{iff } \mathfrak{I}, i \models \phi \text{ and } \mathfrak{I}, i \models \psi \\
\mathfrak{I}, i \models C(a) & \text{iff } a^{\mathcal{I}_i} \in C^{\mathcal{I}_i} & \mathfrak{I}, i \models \bigcirc\phi & \text{iff } \mathfrak{I}, i+1 \models \phi \\
\mathfrak{I}, i \models r(a, b) & \text{iff } (a^{\mathcal{I}_i}, b^{\mathcal{I}_i}) \in r^{\mathcal{I}_i} & \mathfrak{I}, i \models \phi\,\mathcal{U}_I\psi & \text{iff } \exists k\colon k - i \in I,\ \mathfrak{I}, k \models \psi, \\
\mathfrak{I}, i \models \neg\phi & \text{iff not } \mathfrak{I}, i \models \phi & & \quad\text{and } \forall j \in [i, k)\colon \mathfrak{I}, j \models \phi.
\end{array}$$

As usual, we define $\bot := \neg\top$, $C \sqcup D := \neg(\neg C \sqcap \neg D)$, $\forall r.C := \neg(\exists r.\neg C)$, $\phi \vee \psi := \neg(\neg\phi \wedge \neg\psi)$, $\alpha\mathcal{U}\beta := \alpha\mathcal{U}_{[0,\infty)}\beta$, $\Diamond_I\alpha := \top\mathcal{U}_I\alpha$, $\Box_I\alpha := \neg\Diamond_I\neg\alpha$, $\Diamond\alpha := \top\mathcal{U}\alpha$, and $\Box\alpha := \neg\Diamond\neg\alpha$, where α, β are either concepts or formulae [9,15]. Note that, given the semantics of $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$, $\bigcirc\alpha$ is equivalent to $\Diamond_{[1,1]}\alpha$.

Relation to $\mathbf{LTL}_{\mathcal{ALC}}$. The superscript $\cdot^{\mathsf{bin}}$ denotes that the endpoints of the intervals are given in binary. This does not increase the expressivity compared to $\mathrm{LTL}_{\mathcal{ALC}}$ [17], which allows only the qualitative $\mathcal{U}$. In fact, one can expand a formula $\phi\mathcal{U}_{[c_1,c_2]}\psi$ to $\bigvee_{c_1 \leq i \leq c_2}(\bigcirc^i\psi \wedge \bigwedge_{0 \leq j < i} \bigcirc^j\phi)$, where $\bigcirc^i$ denotes i nested $\bigcirc$ operators. Likewise, $\phi\mathcal{U}_{[c_1,\infty)}\psi$ is equivalent to $\big(\bigwedge_{0 \leq i < c_1} \bigcirc^i\phi\big) \wedge \bigcirc^{c_1}\phi\mathcal{U}\psi$. If this transformation is recursively applied to subformulae, then the size of the result is exponential: ignoring the nested $\bigcirc$ operators, its syntax tree has polynomial depth and an exponential branching factor; the $\bigcirc^i$ formulae have exponential depth, but introduce no branching. This blowup cannot be avoided in general [1,14].

Reasoning. We are interested in the complexity of the *satisfiability* problem in $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$, i.e., deciding whether there exists an interpretation $\mathfrak{I}$ such that $\mathfrak{I}, 0 \models \phi$ holds for a given $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ formula ϕ. We also consider a syntactic restriction from [10]: we say that ϕ is an $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ *formula with global GCIs* if it is of the form $\Box\mathcal{T} \wedge \varphi$, where $\mathcal{T}$ is a conjunction of GCIs and φ is an $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ formula that does not contain GCIs. By *satisfiability w.r.t. global GCIs* we refer to the satisfiability problem restricted to such formulae.

First results. The papers [14, 17] consider the reasoning problems of concept satisfiability in $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ w.r.t. *TBoxes* (corresponding to formulae with global GCIs and without assertions) and satisfiability of $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ *temporal TBoxes* (formulae without assertions). However, these results from [14,17] can be extended to our setting by incorporating *named types* into their quasimodel construction to deal with assertions (see also [20], our Section 3, and [15, Theorem 2.27]).

Theorem 1. *Satisfiability in* $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ *is* 2-ExpSpace-*complete, and* ExpSpace-*complete w.r.t. global GCIs. In* $\mathrm{LTL}_{\mathcal{ALC}}$*, this problem is* ExpSpace-*complete, and* ExpTime-*complete w.r.t. global GCIs.*

Note that ExpSpace-completeness for $\mathrm{LTL}_{\mathcal{ALC}}$ with assertions has already been shown in [20]; we only state it here for completeness. In [14], also the intermediate logic $\mathrm{LTL}^{0,\infty}_{\mathcal{ALC}}$ was investigated, where only intervals of the form $[0, c]$ and $[c, \infty)$ are allowed. However, in [16], it was shown for a branching temporal logic that $\mathcal{U}_{[0,c]}$ can be simulated by the classical $\mathcal{U}$ operator, while only increasing the size of the formula by a polynomial factor. We extend this result to intervals of the form $[c, \infty)$, and apply it to $\mathrm{LTL}^{0,\infty}_{\mathcal{ALC}}$.

Theorem 2. *Any* $\mathrm{LTL}^{0,\infty}_{\mathcal{ALC}}$ *formula can be translated in polynomial time into an equisatisfiable* $\mathrm{LTL}_{\mathcal{ALC}}$ *formula.*

This reduction is quite modular; for example, if the formula has only global GCIs, then this is still the case after the reduction. In fact, the reduction applies to all sublogics of $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ that we consider in this paper. Hence, in the following we do not explicitly consider logics with the superscript $\cdot^{0,\infty}$, knowing that they have the same complexity as the corresponding temporal DLs using only $\mathcal{U}$.

3 $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ with Interval-Rigid Names

In many temporal DLs, *rigid* names are considered, whose interpretation does not change over time. Formally, we fix a finite set $\mathsf{N_{Rig}} \subseteq \mathsf{N_C} \cup \mathsf{N_R}$ of *rigid* concept and role names, and require interpretations $\mathfrak{I} = (\Delta^{\mathfrak{I}}, (\mathcal{I}_i)_{i \in \mathbb{N}})$ to *respect* these names, in the sense that $X^{\mathcal{I}_i} = X^{\mathcal{I}_j}$ holds for all $X \in \mathsf{N_{Rig}}$ and $i, j \in \mathbb{N}$. It turns out that $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ can already express rigid concepts via the (global) GCIs $C \sqsubseteq \bigcirc C$ and $\neg C \sqsubseteq \bigcirc \neg C$. The same does not hold for rigid roles, which lead to undecidability even in $\mathrm{LTL}_{\mathcal{ALC}}$ [15, Theorem 11.1]. Hence, it is not fruitful to consider rigid names in $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ (but they are meaningful for the logics of Section 4).

To augment the expressivity of temporal DLs while avoiding undecidability, we propose *interval-rigid* names. In contrast to rigid names, interval-rigid names only need to remain rigid for a limited period of time. Formally, we take a finite set $\mathsf{N_{IRig}} \subseteq (\mathsf{N_C} \cup \mathsf{N_R}) \setminus \mathsf{N_{Rig}}$ of *interval-rigid names*, and a function $\mathsf{iRig} \colon \mathsf{N_{IRig}} \to \mathbb{N}_{\geq 2}$. An interpretation $\mathfrak{I} = (\Delta^{\mathfrak{I}}, (\mathcal{I}_i)_{i \in \mathbb{N}})$ *respects* the interval-rigid names if the following holds for all $X \in \mathsf{N_{IRig}}$ with $\mathsf{iRig}(X) = k$, and $i \in \mathbb{N}$:

> For each $d \in X^{\mathcal{I}_i}$, there is a time point $j \in \mathbb{N}$ such that $i \in [j, j + k)$ and $d \in X^{\mathcal{I}_\ell}$ for all $\ell \in [j, j + k)$.

Table 1. Complexity of satisfiability in $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ w.r.t. interval-rigid names. For (*), we have 2-ExpTime-completeness for the temporal semantics based on $\mathbb{Z}$ (Th. 5).

	$\mathsf{N}_{\mathsf{IRig}} \subseteq \mathsf{N}_{\mathsf{C}} \cup \mathsf{N}_{\mathsf{R}}$	$\mathsf{N}_{\mathsf{IRig}} \subseteq \mathsf{N}_{\mathsf{C}}$
$\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$	2-ExpSpace $\leq$ [Th. 4]	2-ExpSpace $\geq$ [14]
$\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$, global GCIs	2-ExpTime-hard (*)	ExpSpace $\geq$ [2], $\leq$ [Th. 1]
$\mathrm{LTL}_{\mathcal{ALC}}$	2-ExpTime-hard	ExpSpace $\geq$ [15], $\leq$ [20]
$\mathrm{LTL}_{\mathcal{ALC}}$, global GCIs	2-ExpTime-hard [8]	ExpTime $\geq$ [18], $\leq$ [Th. 1]

Intuitively, any element (or pair of elements) in the interpretation of an interval-rigid name must be in that interpretation for at least k consecutive time points. We call such a name *k-rigid.* The names in $(\mathsf{N}_{\mathsf{C}} \cup \mathsf{N}_{\mathsf{R}}) \setminus (\mathsf{N}_{\mathsf{Rig}} \cup \mathsf{N}_{\mathsf{IRig}})$ are called *flexible.* For simplicity, we assume that iRig assigns 1 to all flexible names.

We investigate the complexity of *satisfiability w.r.t. (interval-)rigid names* (or *(interval-)rigid concepts* if $\mathsf{N}_{\mathsf{IRig}} \subseteq \mathsf{N}_{\mathsf{C}}$ / $\mathsf{N}_{\mathsf{Rig}} \subseteq \mathsf{N}_{\mathsf{C}}$), which is defined as before, but considers only interpretations that respect (interval-)rigid names. Note that (interval-)rigid roles can be used to simulate (interval-)rigid concepts via existential restrictions $\exists r.\top$ (e.g., see [10]). Hence, it is not necessary to consider the case where only role names can be (interval-)rigid. The fact that $\mathsf{N}_{\mathsf{Rig}}$ and $\mathsf{N}_{\mathsf{IRig}}$ are finite is not a restriction, as formulae can only use finitely many names. We assume that the values of iRig are given in binary. Table 1 summarizes our results for $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$. Since interval-rigid concepts A can be simulated by formulae $\big(A \sqsubseteq \Box_{[0,k)}A\big) \wedge \Box\big(\neg A \sqsubseteq \bigcirc(\neg A \sqcup \Box_{[0,k)}A)\big)$, Theorem 1 yields the complexity results in the right column (for sublogics of $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ this is not always so easy). The GCI $A \sqsubseteq \Box_{[0,k)}A$ that applies only to the first time point does not affect the complexity results, even if we restrict all other GCIs to be global.

The complexity of $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ with interval-rigid roles is harder to establish. We first show in Section 3.1 that the general upper bound of 2-ExpSpace still holds, by a novel quasimodel construction. For global GCIs, we show 2-ExpTime-hardness in [8], by an easy adaption of a reduction from [10]. We show 2-ExpTime-completeness if we modify the temporal semantics to be infinite in both directions, i.e., replace $\mathbb{N}$ by $\mathbb{Z}$ in the definition of interpretations (see Section 3.2). We leave the case for the semantics based on $\mathbb{N}$ as future work. To simplify the proofs of the upper bounds, we usually assume that $\mathsf{N}_{\mathsf{IRig}} \subseteq \mathsf{N}_{\mathsf{R}}$ since interval-rigid concepts can be simulated. Moreover, for this section we assume that $\mathsf{N}_{\mathsf{Rig}}$ is empty, as rigid concepts do not affect the complexity of $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$, and rigid roles make satisfiability undecidable.

3.1 Satisfiability is in 2-ExpSpace

For the 2-ExpSpace upper bound, we extend the notion of *quasimodels* from [14]. In [14], quasimodels are abstractions of interpretations in which each time point is represented by a *quasistate*, which contains *types.* Each type describes the interpretation for a single domain element, while a quasistate collects the information about all domain elements at a single time point. Central for the complexity results in [14] is that every satisfiable formula has a quasimodel of a certain regular form, which can be guessed and checked in double exponential space. To handle interval-rigid roles, we extend this approach so that each quasistate additionally provides information about the temporal evolution of domain elements over a window of fixed width, and show that under this extended notion, satisfiability is still captured by the existence of regular quasimodels.

We now formalize this intuition. Let φ be an $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ formula. Denote by $\mathsf{csub}(\varphi)/\mathsf{fsub}(\varphi)/\mathsf{ind}(\varphi)/\mathsf{rol}(\varphi)$ the set of all concepts/formulae/individuals/roles occurring in φ, by $\mathsf{cl}^{\mathsf{c}}(\varphi)$ the closure of $\mathsf{csub}(\varphi) \cup \{C\mathcal{U}D \mid C\mathcal{U}_{[c,\infty)}D \in \mathsf{csub}(\varphi)\}$ under single negations, and likewise for $\mathsf{cl}^{\mathsf{f}}(\varphi)$ and $\mathsf{fsub}(\phi)$. A *concept type for* φ is any subset t of $\mathsf{cl}^{\mathsf{c}}(\varphi) \cup \mathsf{ind}(\varphi)$ such that

T1 $\neg C \in t$ iff $C \notin t$, for all $\neg C \in \mathsf{cl}^{\mathsf{c}}(\varphi)$;
T2 $C \sqcap D \in t$ iff $C, D \in t$, for all $C \sqcap D \in \mathsf{cl}^{\mathsf{c}}(\varphi)$; and
T3 t contains at most one individual name.

Similarly, we define *formula types* $t \subseteq \mathsf{cl}^{\mathsf{f}}(\varphi)$ by the following conditions:

T1' $\neg\alpha \in t$ iff $\alpha \notin t$, for all $\neg\alpha \in \mathsf{cl}^{\mathsf{f}}(\varphi)$; and
T2' $\alpha \wedge \beta \in t$ iff $\alpha, \beta \in t$, for all $\alpha \wedge \beta \in \mathsf{cl}^{\mathsf{f}}(\varphi)$.

Intuitively, a concept type describes one domain element at a single time point, while a formula type expresses constraints on all domain elements. If $a \in t \cap \mathsf{ind}(\varphi)$, then t describes an named element, and we call it a *named type.*

To put an upper bound on the time window we have to consider, we consider the largest number occurring in φ and iRig, and denote it by ℓ_φ. Then, a *(concept/formula) run segment* for φ is a sequence $\sigma = \sigma(0)\ldots\sigma(\ell_\varphi)$ composed exclusively of concept or formula types, respectively, such that

R1 $\bigcirc\alpha \in \sigma(0)$ iff $\alpha \in \sigma(1)$, for all $\bigcirc\alpha \in \mathsf{cl}^*(\varphi)$;
R2 for all $a \in \mathsf{ind}(\varphi)$ an $n \in (0, \ell_\varphi]$, we have $a \in \sigma(0)$ iff $a \in \sigma(n)$;
R3 for all $\alpha\mathcal{U}_I\beta \in \mathsf{cl}^*(\varphi)$, we have $\alpha\mathcal{U}_I\beta \in \sigma(0)$ iff (a) there is $j \in I \cap [0, \ell_\varphi]$ such that $\beta \in \sigma(j)$ and $\alpha \in \sigma(i)$ for all $i \in [0, j)$, or (b) I is of the form $[c, \infty)$ and $\alpha, \alpha\mathcal{U}\beta \in \sigma(i)$ for all $i \in [0, \ell_\varphi]$,

where cl^* is either cl^{c} or cl^{f} (as appropriate), and **R2** does not apply to formula run segments. A concept run segment captures the evolution of a domain element over a sequence of $\ell_\varphi + 1$ time points, and a formula run segment describes general constraints on the interpretation over a sequence of $\ell_\varphi + 1$ time points.

The evolution over the complete time line is captured by *(concept/formula) runs* for φ, which are infinite sequences $r = r(0)r(1)\ldots$ such that each subsequence of length $\ell_\varphi + 1$ is a (concept/formula) run segment, and additionally

R4 $\alpha\mathcal{U}_{[c,\infty)}\beta \in r(n)$ implies that there is $j \geq n + c$ such that $\beta \in r(j)$ and $\alpha \in r(i)$ for all $i \in [n, j)$.

A concept run (segment) is *named* if it contains only (equivalently, any) named types. We may write r_a (σ_a) to denote a run (segment) that contains an individual name a. For a run (segment) σ, we write $\sigma^{>i}$ for the subsequence of σ starting at $i+1$, $\sigma^{<i}$ for the one stopping at $i-1$, and $\sigma^{[i,j]}$ for $\sigma(i)\dots\sigma(j)$.

As we cannot explicitly represent infinite runs, we use run segments to construct them step-by-step. For this, it is important that a set of concept runs (segments) can be composed into a coherent model. In particular, we have to take care of (interval-rigid) role connections. A *role constraint* for φ is a tuple (σ, σ', s, k), with concept run segments σ, σ', $s \in \mathsf{rol}(\varphi)$, $k \in [1, \mathsf{iRig}(s)]$, such that

C1 $\{\neg C \mid \neg\exists s.C \in \sigma(0)\} \subseteq \sigma'(0)$; and
C2 if σ' is named, then σ is also named.

We write $\sigma \;{}^{s}_{k}\; \sigma'$ as a shorthand for the role constraint (σ, σ', s, k). Intuitively, $\sigma \;{}^{s}_{k}\; \sigma'$ means that the domain elements described by $\sigma(0), \sigma'(0)$ are connected by the role s at the current time point, and also at the $k-1$ previous time points. In this case, we need to ensure that these elements stay connected for at least the following $\mathsf{iRig}(s) - k$ time points. Condition **C1** ensures that, if $\sigma(0)$ cannot have any s-successors that satisfy C, then $\sigma'(0)$ does not satisfy C.

We can now describe the behaviour of a whole interpretation and its elements at a single time point, together with some bounded information about the future (up to ℓ_φ time points). A *quasistate* for φ is a pair $Q = (\mathcal{R}_Q, \mathcal{C}_Q)$, where $\mathcal{R}_Q$ is a set of run segments and $\mathcal{C}_Q$ a set of role constraints over $\mathcal{R}_Q$ such that

Q1 $\mathcal{R}_Q$ contains exactly one formula run segment σ_Q;
Q2 $\mathcal{R}_Q$ contains exactly one named run segment σ_a for each $a \in \mathsf{ind}(\varphi)$;
Q3 for all $C \sqsubseteq D \in \mathsf{cl}^{\mathsf{f}}(\varphi)$, we have $C \sqsubseteq D \in \sigma_Q(0)$ iff $C \in \sigma(0)$ implies $D \in \sigma(0)$ for all concept run segments $\sigma \in \mathcal{R}_Q$;
Q4 for all $C(a) \in \mathsf{cl}^{\mathsf{f}}(\varphi)$, we have $C(a) \in \sigma_Q(0)$ iff $C \in \sigma_a(0)$;
Q5 for all $s(a,b) \in \mathsf{cl}^{\mathsf{f}}(\varphi)$, we have $s(a,b) \in \sigma_Q(0)$ iff $\sigma_a \;{}^{s}_{k}\; \sigma_b \in \mathcal{C}_Q$ for some $k \in [1, \mathsf{iRig}(s)]$; and
Q6 for all $\sigma \in \mathcal{R}_Q$ and $\exists s.D \in \sigma(0)$, there is $\sigma \;{}^{s}_{k}\; \sigma' \in \mathcal{C}_Q$ with $D \in \sigma'(0)$ and $k \in [1, \mathsf{iRig}(s)]$.

We next capture when quasistates can be connected coherently to an infinite sequence. A pair (Q, Q') of quasistates is *compatible* if there is a *compatibility relation* $\pi \subseteq \mathcal{R}_Q \times \mathcal{R}_{Q'}$ such that

C3 every run segment in $\mathcal{R}_Q$ and $\mathcal{R}_{Q'}$ occurs at least once in the domain and range of π, respectively;
C4 each pair $(\sigma, \sigma') \in \pi$ satisfies $\sigma^{>0} = \sigma'^{<\ell_\varphi}$;
C5 for all $(\sigma_1, \sigma_1') \in \pi$ and $\sigma_1 \;{}^{s}_{k}\; \sigma_2 \in Q$ with $k < \mathsf{iRig}(s)$, there is $\sigma_1' \;{}^{\;s}_{k+1}\; \sigma_2' \in Q'$ with $(\sigma_2, \sigma_2') \in \pi$; and
C6 for all $(\sigma_1, \sigma_1') \in \pi$ and $\sigma_1' \;{}^{\;s}_{k+1}\; \sigma_2' \in Q'$ with $k > 1$, there is $\sigma_1 \;{}^{s}_{k}\; \sigma_2 \in Q$ with $(\sigma_2, \sigma_2') \in \pi$.

Such a relation makes sure that we can combine run segments of consecutive quasistates such that the interval-rigid roles are respected. Note that the unique

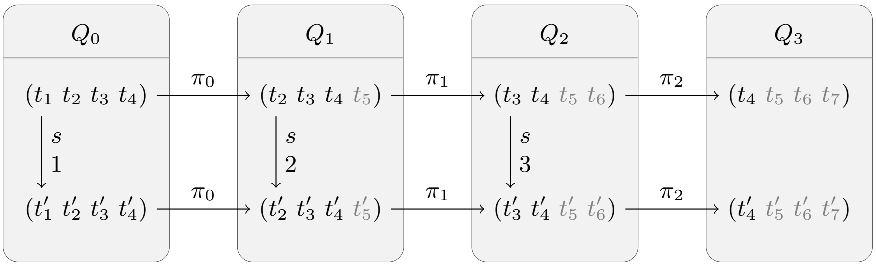

Figure 2. Illustration of role constraints and compatibility relations.

formula run segments must be matched to each other, and likewise for the named run segments. Moreover, the set of all compatibility relations for a pair of quasistates (Q, Q') is closed under union, which means that compatible quasistates always have a unique maximal compatibility relation (w.r.t. set inclusion).

To illustrate this, consider Figure 2, showing a sequence of pairwise compatible quasistates, each containing two run segments. Here, $\ell_\varphi = \mathsf{iRig}(s) = 3$. The relations π_0, π_1, and π_2 satisfy Conditions **C3**–**C6**, which, together with **C1** and **C2**, ensure that a run going through the types t_1, t_2, t_3, and t_4 can be connected to another run via the role s for at least 3 consecutive time points.

Finally, a *quasimodel* for φ is a pair $(S, \mathfrak{R})$, where S is an infinite sequence of compatible quasistates $S(0)S(1)\ldots$ and $\mathfrak{R}$ is a non-empty set of runs, such that

M1 the runs in $\mathfrak{R}$ are of the form $\sigma_0(0)\sigma_1(0)\sigma_2(0)\ldots$ such that, for every $i \in \mathbb{N}$, we have $(\sigma_i, \sigma_{i+1}) \in \pi_i$, where π_i is the maximal compatibility relation for the pair $(S(i), S(i+1))$;
M2 for every $\sigma \in \mathcal{R}_{S(i)}$, there exists a run $r \in \mathfrak{R}$ with $r^{[i,i+\ell_\varphi]} = \sigma$;
M3 every role constraint in $S(0)$ is of the form $\sigma_1 \overset{s}{1} \sigma_2$; and
M4 $\varphi \in \sigma_{S(0)}(0)$.

By **M1**, the runs $\sigma_0(0)\sigma_1(0)\sigma_2(0)\ldots$ always contain the whole run segments $\sigma_0, \sigma_1, \sigma_2, \ldots$, since we have $\sigma_1(0) = \sigma_0(1)$, $\sigma_2(0) = \sigma_0(2)$, and so on. Moreover, $\mathfrak{R}$ always contains exactly one formula run and one named run for each $a \in \mathsf{ind}(\varphi)$.

We can show that every quasimodel describes a satisfying interpretation for φ and, conversely, that every such interpretation can be abstracted to a quasimodel. Moreover, one can always find a quasimodel of a regular shape.

Lemma 3. *An* $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ *formula* φ *is satisfiable w.r.t. interval-rigid names iff* φ *has a quasimodel* $(S, \mathfrak{R})$ *in which* S *is of the form*

$$S(0)\ldots S(n)(S(n+1)\ldots S(n+m))^\omega,$$

where n *and* m *are bounded triple exponentially in the size of* φ *and* iRig*.*

This allows us to devise a non-deterministic 2-ExpSpace algorithm that decides satisfiability of a given $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ formula. Namely, we first guess n and m, and then the quasistates $S(0), \ldots, S(n+m)$ one after the other. To show that this sequence corresponds to a quasimodel as in Lemma 3, note that only three quasistates have to be kept in memory at any time, the sizes of which are double exponentially bounded in the size of the input: the current quasistate, the next quasistate, and the first repeating quasistate $S(n+1)$. 2-ExpSpace-hardness holds already for the case without interval-rigid names or assertions [14].

Theorem 4. *Satisfiability in* $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ *with respect to interval-rigid names is* 2-ExpSpace-*complete.*

3.2 Global GCIs

For $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ formulae with global GCIs, we show a tight (2-ExpTime) complexity bound only if we consider a modified temporal semantics that uses $\mathbb{Z}$ instead of $\mathbb{N}$. Over $\mathbb{Z}$, every satisfiable formula has a quasimodel in which the unnamed run segments and role constraints are the same for all quasisates. This is not the case for $\mathbb{N}$, since then a quasistate at time point 1 can have role constraints $\sigma \ {}^{s}_{k}\ \sigma'$ with $k > 1$, whereas one at time point 0 cannot (see **M3**).

Hence, interpretations are now of the form $\mathfrak{I} = (\Delta^{\mathfrak{I}}, (\mathcal{I}_i)_{i\in\mathbb{Z}})$, where $\Delta^{\mathfrak{I}}$ is a constant domain and $\mathcal{I}_i$ are classical DL interpretations, as before. Recall that an $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ formula with global GCIs is of the form $\Box\mathcal{T}\wedge\phi$, where $\mathcal{T}$ is a conjunction of GCIs and ϕ is an $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ formula that does not contain GCIs. In order to enforce our GCIs on the whole time line (including the time points before 0), we replace $\Box\mathcal{T}$ with $\Box^-_+$ in that definition, where $\Box^-_+\mathcal{T}$ expresses that in all models $\mathfrak{I}$, $\mathfrak{I}, i \models \mathcal{T}$ for all $i \in \mathbb{Z}$. We furthermore slightly adapt some of the notions introduced in Section 3.1. First, to ensure that GCIs hold on the whole time line, we require (in addition to **T1'** and **T2'**) that all formula types contain all GCIs from $\mathcal{T}$. Additionally, we adapt the notions of runs $\dots r(-1)r(0)r(1)\dots$ and sequences $\dots S(-1)S(0)S(1)\dots$ of quasistates to be infinite in both directions. Hence, we can now drop Condition **M3**, reflecting the fact that, over $\mathbb{Z}$, role connections can exist before time point 0. All other definitions remain unchanged.

The proof follows a similar idea as in the last section. We first show that every formula is satisfiable iff it has a quasimodel of a regular shape, which now is also constant in its unnamed part, in the sense that, if unnamed run segments and role constraints occur in $S(i)$, then they also occur in $S(j)$, for all $i, j \in \mathbb{Z}$. This allows us to devise an elimination procedure (in the spirit of [17, Theorem 3] and [14, Theorem 2]), with the difference that we eliminate run segments and role constraints instead of types, which gives us a 2-ExpTime upper bound.

Theorem 5. *Satisfiability in* $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ *w.r.t. interval-rigid names and global GCIs over* $\mathbb{Z}$ *is* 2-ExpTime-*complete.*

4 Metric Extensions of $\mathcal{ALC}$-LTL

We briefly summarize results that we have obtained for the sublogic $\mathcal{ALC}$-$\mathrm{LTL}^{\mathsf{bin}}$ of $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$, which does not allow temporal operators within concepts (cf. [10]). Due to lack of space, we refer the reader to [8] for all details. An $\mathcal{ALC}$-$\mathrm{LTL}^{\mathsf{bin}}$

Table 2. Complexity of satisfiability in $\mathcal{ALC}$-LTL$^{\mathsf{bin}}$ w.r.t. (interval-)rigid names.

	$\mathsf{N_{IRig}} \subseteq \mathsf{N_C}$,	$\mathsf{N_{IRig}} \subseteq \mathsf{N_C} \cup \mathsf{N_R}$,	$\mathsf{N_{IRig}} \subseteq \mathsf{N_C}$,
	$\mathsf{N_{Rig}} \subseteq \mathsf{N_C} \cup \mathsf{N_R}$	$\mathsf{N_{Rig}} \subseteq \mathsf{N_C}$ or $\mathsf{N_{Rig}} = \emptyset$	$\mathsf{N_{Rig}} \subseteq \mathsf{N_C}$ or $\mathsf{N_{Rig}} = \emptyset$
$\mathcal{ALC}$-LTL$^{\mathsf{bin}}$	undec.	2-ExpTime-hard	ExpSpace $\leq$ [Th. 1]
$\mathcal{ALC}$-LTL$^{\mathsf{bin}}_{\vert gGCI}$	undec.	2-ExpTime-hard	ExpTime-hard
$\mathcal{ALC}$-LTL	undec.	2-ExpTime-hard	ExpSpace $\geq$ [8]
$\mathcal{ALC}$-LTL$_{\vert gGCI}$	undec. [8]	2-ExpTime-hard [8]	ExpTime $\geq$ [18], $\leq$ [Th. 1]

Table 3. Complexity of satisfiability in $\mathcal{ALC}$-LTL$^{\mathsf{bin}}$ without interval-rigid names.

	$\mathsf{N_{Rig}} \subseteq \mathsf{N_C} \cup \mathsf{N_R}$	$\mathsf{N_{Rig}} \subseteq \mathsf{N_C}$	$\mathsf{N_{Rig}} = \emptyset$
$\mathcal{ALC}$-LTL$^{\mathsf{bin}}$	2-ExpTime $\leq$ [8]	ExpSpace $\leq$ [8]	ExpSpace
$\mathcal{ALC}$-LTL$^{\mathsf{bin}}_{\vert gGCI}$	2-ExpTime	ExpSpace	ExpSpace $\geq$ [1]
$\mathcal{ALC}$-LTL	2-ExpTime	NExpTime [10]	ExpTime $\leq$ [10]
$\mathcal{ALC}$-LTL$_{\vert gGCI}$	2-ExpTime $\geq$ [10]	ExpTime $\leq$ [10]	ExpTime $\geq$ [18]

formula is an LTL$^{\mathsf{bin}}_{\mathcal{ALC}}$ formula in which all concepts are $\mathcal{ALC}$ concepts. Recall that $\mathcal{ALC}$-LTL, which has been investigated in [10] (though not with interval-rigid names), restricts $\mathcal{ALC}$-LTL$^{\mathsf{bin}}$ to intervals of the form $[0, \infty)$. As done in [10], for brevity, we distinguish here the variants with global GCIs by the subscript $\cdot_{\vert gGCI}$. In contrast to LTL$^{\mathsf{bin}}_{\mathcal{ALC}}$, in $\mathcal{ALC}$-LTL rigid concepts cannot be simulated by GCIs and rigid roles do not lead to undecidability [10]. Hence, we investigate here also the settings with rigid concepts and/or roles.

Known and new complexity results are compared in Tables 2 and 3 for $\mathcal{ALC}$-LTL$^{\mathsf{bin}}$ with and without interval-rigid names, respectively. In the presence of interval-rigid names, we obtain several hardness results already for $\mathcal{ALC}$-LTL, based on the insight that interval-rigid concepts can express the operator $\bigcirc$ on the concept level. In particular, the combination of rigid roles with interval-rigid concepts already leads to undecidability. If interval-rigid names are disallowed, the complexity of satisfiability in $\mathcal{ALC}$-LTL$^{\mathsf{bin}}$ corresponds to the maximum of the complexities of satisfiability in $\mathcal{ALC}$-LTL and LTL$^{\mathsf{bin}}$.

5 Conclusions

We investigated a series of extensions of LTL$_{\mathcal{ALC}}$ and $\mathcal{ALC}$-LTL with interval-rigid names and metric temporal operators, with complexity results ranging from ExpTime to 2-ExpSpace. Some cases were left open, such as the precise complexity of LTL$^{\mathsf{bin}}_{\mathcal{ALC}}$ with global GCIs, for which we have a partial result for the temporal semantics based on $\mathbb{Z}$. Nevertheless, this paper provides a comprehensive guide to the complexities faced by applications that want to combine ontological reasoning with quantitative temporal logics. For future work, it would be interesting to extend temporal DLs based on light-weight logics such as *DL-Lite* and $\mathcal{EL}$ [5, 11] with interval-rigid roles and metric operators.

References

1. Alur, R., Henzinger, T.A.: Real-time logics: Complexity and expressiveness. Inf. Comput. 104(1), 35–77 (1993)
2. Alur, R., Henzinger, T.A.: A really temporal logic. J. ACM 41(1), 181–204 (1994)
3. Artale, A., Bresolin, D., Montanari, A., Sciavicco, G., Ryzhikov, V.: DL-Lite and interval temporal logics: A marriage proposal. In: Proc. of the 21st Eur. Conf. on Artificial Intelligence (ECAI'14). pp. 957–958. IOS Press (2014)
4. Artale, A., Kontchakov, R., Ryzhikov, V., Zakharyaschev, M.: Tractable interval temporal propositional and description logics. In: Proc. of the 29th AAAI Conf. on Artificial Intelligence (AAAI'15). pp. 1417–1423. AAAI Press (2015)
5. Artale, A., Kontchakov, R., Lutz, C., Wolter, F., Zakharyaschev, M.: Temporalising tractable description logics. In: Proc. of the 14th Int. Symp. on Temporal Representation and Reasoning (TIME'07), pp. 11–22. IEEE Press (2007)
6. Artale, A., Lutz, C., Toman, D.: A description logic of change. In: Proc. of the 20th Int. Joint Conf. on Artificial Intelligence (IJCAI'07). pp. 218–223 (2007)
7. Baader, F., Borgwardt, S., Koopmann, P., Ozaki, A., Thost, V.: Metric temporal description logics with interval-rigid names. In: Proc. of the 11th Int. Symp. on Frontiers of Combining Systems (2017), to appear.
8. Baader, F., Borgwardt, S., Koopmann, P., Ozaki, A., Thost, V.: Metric temporal description logics with interval-rigid names (extended version). LTCS-Report 17-03 (2017), see `https://lat.inf.tu-dresden.de/research/reports.html`
9. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, 2nd edn. (2007)
10. Baader, F., Ghilardi, S., Lutz, C.: LTL over description logic axioms. ACM Trans. Comput. Log. 13(3), 21:1–21:32 (2012)
11. Borgwardt, S., Thost, V.: Temporal query answering in the description logic $\mathcal{EL}$. In: Proc. of the 24th Int. Joint Conf. on Artificial Intelligence (IJCAI'15). pp. 2819–2825. AAAI Press (2015)
12. Brandt, S., Kalaycı, E.G., Kontchakov, R., Ryzhikov, V., Xiao, G., Zakharyaschev, M.: Ontology-based data access with a Horn fragment of metric temporal logic. In: Proc. of the 31st AAAI Conf. on Artificial Intelligence (AAAI'17). pp. 1070–1076. AAAI Press (2017)
13. Crowe, C.L., Tao, C.: Designing ontology-based patterns for the representation of the time-relevant eligibility criteria of clinical protocols. AMIA Summits on Translational Science Proceedings 2015, 173–177 (2015)
14. Gutiérrez-Basulto, V., Jung, J.C., Ozaki, A.: On metric temporal description logics. In: Proc. of the 22nd Eur. Conf. on Artificial Intelligence (ECAI'16). pp. 837–845. IOS Press (2016)
15. Kurucz, A., Wolter, F., Zakharyaschev, M., Gabbay, D.M.: Many-dimensional modal logics: Theory and applications. Gulf Professional Publishing (2003)
16. Lutz, C., Walther, D., Wolter, F.: Quantitative temporal logics over the reals: PSPACE and below. Inf. Comput. 205(1), 99–123 (2007)
17. Lutz, C., Wolter, F., Zakharyaschev, M.: Temporal description logics: A survey. In: Proc. of the 15th Int. Symp. on Temporal Representation and Reasoning (TIME'08). pp. 3–14. IEEE Press (2008)
18. Schild, K.: A correspondence theory for terminological logics: Preliminary report. In: Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI'91). pp. 466–471. Morgan Kaufmann (1991)

19. Schulz, S., Markó, K., Suntisrivaraporn, B.: Formal representation of complex SNOMED CT expressions. BMC Medical Informatics and Decision Making 8(Suppl 1), S9 (2008), selected contributions to the 1st Eur. Conf. on SNOMED CT
20. Wolter, F., Zakharyaschev, M.: Temporalizing description logics. In: Frontiers of Combining Systems 2. pp. 379–402. Research Studies Press/Wiley (2000)