TECHNISCHE UNIVERSITÄT DRESDEN

Technische Universität Dresden
Institute for Theoretical Computer Science
Chair for Automata Theory

LTCS-Report

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

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

LTCS-Report 17-03

Postal Address:
Lehrstuhl für Automatentheorie
Institut für Theoretische Informatik
TU Dresden
01062 Dresden

http://lat.inf.tu-dresden.de

Visiting Address:
Nöthnitzer Str. 46
Dresden

Contents

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

Description Logics [7] are a well-investigated family of logic-based knowledge representation languages, which provide the formal basis for the Web Ontology Language OWL.[1] As a consequence, 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[2]). For example, the concept of a patient with a concussion can formally be expressed in DLs as $\mathsf{Patient} \sqcap \exists \mathsf{finding}.\mathsf{Concussion}$, which is built from the concept names (i.e., unary predicates) Patient and Concussion and the role name (i.e., binary predicate) finding using the concept constructors conjunction ($\sqcap$) and existential restriction ($\exists r.C$). Concepts and roles can then be used within terminological and assertional axioms to state facts about the application domain, such as that concussion is a disease ($\mathsf{Concussion} \sqsubseteq \mathsf{Disease}$) and that patient Bob has a concussion ($\mathsf{Patient}(\mathsf{BOB}), \mathsf{finding}(\mathsf{BOB}, \mathsf{F1}), \mathsf{Concussion}(\mathsf{F1})$).

This example, taken from [8], can also 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" [18]. However, the temporal pattern inherent in this concept (after the concussion, the patient remained conscious until the examination) cannot be modelled in the DL used for SNOMED CT.

To overcome the problem that pure DLs are not able to express such temporal patterns, a great variety of temporal extensions of DLs have been investigated in the literature.[3] In the present paper, we concentrate on the DL $\mathcal{ALC}$ and combine it with linear temporal logic (LTL), a point-based temporal logic whose semantics assumes 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 [8] or also use them within concepts [14, 19]. With the latter, one can then formalize "concussion 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 [8], 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 hasFather are clearly rigid, whereas Conscious and finding are flexible, i.e., not rigid. If temporal operators can be used within concepts, rigid concepts can be expressed using terminological axioms, but rigid roles cannot. In fact, they usually render the combined logic undecidable [14, Proposition 3.34]. In contrast, in the setting considered in [8], rigid roles do not cause undecidability, but adding rigidity leads to an increase in complexity.

In this paper, we address a shortcoming of the purely qualitative temporal description logics

[1] `https://www.w3.org/TR/2009/WD-owl2-overview-20090327/`

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

[3] We refer the reader to [14, 16] for an overview of the field of temporal DLs.

mentioned until now. The qualitative 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 in case the examination took place in temporal proximity to the concussion. Otherwise, an intermediate 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 [11] like the following: 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. On the temporal logic side, extensions of LTL by such intervals have been investigated in detail [1, 2, 15]. 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 [15] 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 [12], 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 terminological axioms. In Section 2, we basically recall some of the results obtained in [12], but show that they also hold if additionally temporalized assertional axioms 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. Basically, this allows one to state that elements belonging to a concept need to 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,[4] 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 consider the effect of adding interval-rigid concepts and roles as well as metric temporal operators to the logic $\mathcal{ALC}$-LTL of [8], where temporal operators can only be applied to axioms. 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). Finally, in Section 5 we investigate the complexity of this logic without interval-rigid names, which extends the analysis from [8] to quantitative temporal operators (see Table 3). Detailed proofs of all results can be found in the appendix.

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 is quite different from ours, since it 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, an interesting metric temporal extension of Datalog over the reals was proposed, which however cannot express interval-rigid names nor existential restrictions [10].

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

We first introduce the description logic $\mathcal{ALC}$ and its metric temporal extension $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ [12], 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

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

axioms, and show that this does not change the complexity of reasoning compared to the results of [12].

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 $c_1, c_2 \in \mathbb{N}$ 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:

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

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 [7, 14].

Relation to $\mathrm{LTL}_{\mathcal{ALC}}$. The notation $\cdot^{\mathsf{bin}}$ refers to the fact that the endpoints of the intervals are given in binary. However, this does not increase the expressivity compared to $\mathrm{LTL}_{\mathcal{ALC}}$ [16], where only the qualitative $\mathcal{U}$ operator is allowed. In fact, one can expand any formula $\phi\,\mathcal{U}_{[c_1,c_2]}\psi$ to $\bigvee_{c_1 \le i \le c_2}(\bigcirc^i\psi \wedge \bigwedge_{0 \le j < i} \bigcirc^j\phi)$, where $\bigcirc^i$ denotes i nested $\bigcirc$ operators, and similarly for concepts. Likewise, $\phi\,\mathcal{U}_{[c_1,\infty)}\psi$ is equivalent to $\big(\bigwedge_{0 \le i < c_1} \bigcirc^i\phi\big) \wedge \bigcirc^{c_1}\phi\,\mathcal{U}\psi$. If this transformation is recursively applied to subformulae, then the size of the resulting formula is exponential: ignoring the nested $\bigcirc$ operators, its syntax tree has polynomial depth and an exponential

branching factor; and the $\bigcirc^i$ formulae have exponential depth, but introduce no branching. This blowup cannot be avoided in general [1, 12].

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 [8]: 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 [12, 16] 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 [12, 16] can be extended to our setting by incorporating *named types* into their quasimodel construction to deal with assertions (see also [19], our Section 3, and [14, 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 [19]; we only state it here for completeness. In [12], 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 [15], 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, so-called *rigid* names are considered, whose interpretation is not allowed to change over time. To formally define this notion, we fix a finite set $\mathsf{N}_{\mathsf{Rig}} \subseteq \mathsf{N}_{\mathsf{C}} \cup \mathsf{N}_{\mathsf{R}}$ 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 $X^{\mathcal{I}_i} = X^{\mathcal{I}_j}$ should hold for all $X \in \mathsf{N}_{\mathsf{Rig}}$ 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 $\mathrm{LTL}_{\mathcal{ALC}}$ [14, Theorem 11.1]. Hence, it is not fruitful to consider rigid names in $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ (they will become meaningful later, when we look at other logics).

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 $\mathsf{N}_{\mathsf{IRig}} \subseteq (\mathsf{N}_{\mathsf{C}} \cup \mathsf{N}_{\mathsf{R}}) \setminus \mathsf{N}_{\mathsf{Rig}}$ of *interval-rigid names*, and a function $\mathsf{iRig}\colon \mathsf{N}_{\mathsf{IRig}} \to \mathbb{N}_{\geq 2}$. 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 $X \in \mathsf{N}_{\mathsf{IRig}}$ with $\mathsf{iRig}(X) = k$, and $i \in \mathbb{N}$:

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).

	$\mathsf{N_{IRig}} \subseteq \mathsf{N_C} \cup \mathsf{N_R}$	$\mathsf{N_{IRig}} \subseteq \mathsf{N_C}$
$\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$	2-ExpSpace $\leq$ [Th. 4]	2-ExpSpace $\geq$ [12]
$\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$, global GCIs	2-ExpTime-hard (*)	ExpSpace $\geq$ [2], $\leq$ [Th. 1]
$\mathrm{LTL}_{\mathcal{ALC}}$	2-ExpTime-hard	ExpSpace $\geq$ [14], $\leq$ [19]
$\mathrm{LTL}_{\mathcal{ALC}}$, global GCIs	2-ExpTime-hard [Th. 8]	ExpTime $\geq$ [17], $\leq$ [Th. 1]

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 $\ell \in [j, j+k)$.

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 $(\mathsf{N_C} \cup \mathsf{N_R}) \setminus (\mathsf{N_{Rig}} \cup \mathsf{N_{IRig}})$ 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 $\mathsf{N_{IRig}} \subseteq \mathsf{N_C}$ / $\mathsf{N_{Rig}} \subseteq \mathsf{N_C}$), 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 $\exists r.\top$. Therefore, it is not necessary to consider the case where only role names can be (interval-)rigid. The fact that $\mathsf{N_{Rig}}$ and $\mathsf{N_{IRig}}$ 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 conjuncts of the form $\big(A \sqsubseteq \Box_{[0,k)} A\big) \wedge \Box\big(\neg A \sqsubseteq \bigcirc(\neg A \sqcup \Box_{[0,k)} A)\big)$, Theorem 1 directly yields the complexity results in the right column (again, 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 Section 4, by an easy adaption of a reduction from [8]. 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 $\mathsf{N_{IRig}} \subseteq \mathsf{N_R}$ since interval-rigid concepts can be simulated. Moreover, for this section we assume that $\mathsf{N_{Rig}}$ 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 [12]. In [12], 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 [12] 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 $\sigma = \sigma(0) \dots \sigma(\ell_\varphi)$ 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 $r = r(0)r(1)\dots$ 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 $\sigma(i)\dots\sigma(j)$.

Since 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 actually be

composed into a coherent model. In particular, we have to take care of (interval-rigid) role connections between elements. A *role constraint* for φ is a tuple (σ, σ', s, k), where σ, σ' are concept run segments, $s \in \mathsf{rol}(\varphi)$, and $k \in [1, \mathsf{iRig}(s)]$, 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 $\sigma \overset{s}{k} \sigma'$ as a shorthand for the role constraint (σ, σ', s, k). Intuitively, $\sigma \overset{s}{k} \sigma'$ 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 $\sigma_a \overset{s}{k} \sigma_b \in \mathcal{C}_Q$ 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 $\sigma \overset{s}{k} \sigma' \in \mathcal{C}_Q$ 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 $\sigma_1 \overset{s}{k} \sigma_2 \in Q$ with $k < \mathsf{iRig}(s)$, there is $\sigma_1' \overset{s}{k+1} \sigma_2' \in Q'$ with $(\sigma_2, \sigma_2') \in \pi$; and

C6 for all $(\sigma_1, \sigma_1') \in \pi$ and $\sigma_1' \overset{s}{k+1} \sigma_2' \in Q'$ with $k > 1$, there is $\sigma_1 \overset{s}{k} \sigma_2 \in Q$ 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 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 1, 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

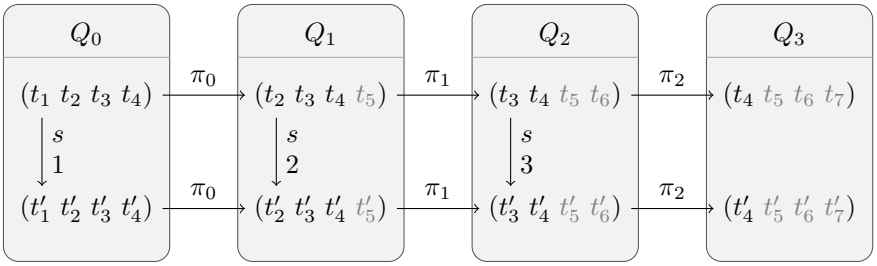

Figure 1: Illustration of role constraints and compatibility relations.

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 $S(0)S(1)\dots$ and $\mathfrak{R}$ is a non-empty set of runs, such that

M1 the runs in $\mathfrak{R}$ are of the form $\sigma_0(0)\sigma_1(0)\sigma_2(0)\dots$ 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 $\sigma_0(0)\sigma_1(0)\sigma_2(0)\dots$ always contain the whole run segments $\sigma_0, \sigma_1, \sigma_2, \dots$, 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*

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

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 $S(0), \dots, S(n+m)$ one after the other. To show that this sequence corresponds to a quasimodel as in Lemma 3, only a constant number of quasistates has to be kept in memory, and the size of each quasistate is double exponentially bounded in the size of the input. 2-ExpSpace-hardness holds already for the case without interval-rigid names or assertions [12].

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 can show a tight (2-ExpTime) complexity bound only if we consider a modified temporal semantics that uses $\mathbb{Z}$ instead of $\mathbb{N}$. With a semantics

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 if the semantics is only defined for $\mathbb{N}$, since then a quasistate at time point 1 can have role constraints $\sigma \overset{s}{k} \sigma'$ 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 $\text{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ formula with global GCIs is an $\text{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ formula of the form $\Box\mathcal{T} \wedge \phi$, where $\mathcal{T}$ is a conjunction of GCIs and ϕ is an $\text{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ 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 $\ldots r(-1)r(0)r(1)\ldots$ and sequences $\ldots S(-1)S(0)S(1)\ldots$ 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 complexity 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 [16, Theorem 3] and [12, Theorem 2]), with the difference that we eliminate run segments and role constraints instead of types, which gives us a 2-ExpTime upper bound. The matching lower bound can be shown similarly to Theorem 8 in Section 4.

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

4 $\mathcal{ALC}$-LTL$^{\mathsf{bin}}$ with Interval-Rigid Names

After the very expressive DL $\text{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$, we now focus on its sublogic $\mathcal{ALC}$-$\text{LTL}^{\mathsf{bin}}$, which does not allow temporal operators within concepts (cf. [8]). That is, an $\mathcal{ALC}$-$\text{LTL}^{\mathsf{bin}}$ *formula* is an $\text{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ formula in which all concepts are $\mathcal{ALC}$ concepts. Recall that $\mathcal{ALC}$-LTL, which has been investigated in [8] (though not with interval-rigid names), restricts $\mathcal{ALC}$-$\text{LTL}^{\mathsf{bin}}$ to intervals of the form $[0, \infty)$. In this section, we show several complexity lower bounds that already hold for $\mathcal{ALC}$-LTL with interval-rigid names. As done in [8], for brevity, we distinguish here the variants with global GCIs by the subscript $\cdot_{|gGCI}$. In contrast to $\text{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$, in $\mathcal{ALC}$-LTL rigid concepts cannot be simulated by GCIs and rigid roles do not lead to undecidability [8]. Hence, we investigate here also the settings with rigid concepts and/or roles.

The results of this section are summarized in Table 2. Central to our hardness proofs is the insight that interval-rigid concepts can express the operator $\bigcirc$ on the concept level. In particular, we show that the combination of rigid roles with interval-rigid concepts already leads to undecidability, by a reduction from a tiling problem. If rigid names are disallowed, but we have interval-rigid names, we can only show 2-ExpTime-hardness. If only interval-rigid concepts are allowed, then satisfiability is ExpSpace-hard. All of these hardness results already hold for $\mathcal{ALC}$-LTL, and some of them even with global GCIs.

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

	$\mathsf{N_{IRig}} \subseteq \mathsf{N_C}$, $\mathsf{N_{Rig}} \subseteq \mathsf{N_C} \cup \mathsf{N_R}$	$\mathsf{N_{IRig}} \subseteq \mathsf{N_C} \cup \mathsf{N_R}$, $\mathsf{N_{Rig}} \subseteq \mathsf{N_C}$ or $\mathsf{N_{Rig}} = \emptyset$	$\mathsf{N_{IRig}} \subseteq \mathsf{N_C}$, $\mathsf{N_{Rig}} \subseteq \mathsf{N_C}$ or $\mathsf{N_{Rig}} = \emptyset$
$\mathcal{ALC}$-LTL$^{\mathsf{bin}}$	undec.	2-ExpTime-hard	ExpSpace $\leq$ [Th. 1]
$\mathcal{ALC}$-LTL$^{\mathsf{bin}}_{\lvert gGCI}$	undec.	2-ExpTime-hard	ExpTime-hard
$\mathcal{ALC}$-LTL	undec.	2-ExpTime-hard	ExpSpace $\geq$ [Th. 9]
$\mathcal{ALC}$-LTL$_{\lvert gGCI}$	undec. [Th. 6]	2-ExpTime-hard [Th. 8]	ExpTime $\geq$ [17], $\leq$ [Th. 1]

4.1 Rigid Roles and Interval-Rigid Concepts

We show that satisfiability of $\mathcal{ALC}$-LTL with rigid roles and interval-rigid concepts is undecidable, even if we only allow global GCIs. Our proof is by a reduction from the following tiling problem.

> 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 $\mathcal{ALC}$-LTL$_{\lvert gGCI}$ formula ϕ_T that expresses this property. In our encoding, we use the following names:

- a rigid role name r to encode the vertical dimension of the $\mathbb{N} \times \mathbb{N}$ grid;
- flexible concept names A^0, A^1, A^2 to encode the progression along the horizontal (temporal) dimension; for convenience, we consider all superscripts modulo 3, i.e., we have $A^3 = A^0$ and $A^{-1} = A^2$;
- flexible concept names P_t, $t \in T$, to denote the current tile type;
- 2-rigid concept names N^0_t, N^1_t, N^2_t, for the horizontally adjacent tile type;
- an individual name a denotes the first row of the grid.

We define ϕ_T as the conjunction of the following $\mathcal{ALC}$-LTL$_{\lvert gGCI}$ formulae. First, every domain element must have exactly one tile type:

$$\Box\Big(\top \sqsubseteq \bigsqcup_{t \in T} \Big(P_t \sqcap \bigsqcap_{t' \in T,\, t \neq t'} \neg P_{t'}\Big)\Big)$$

For the vertical dimension, we enforce an infinite rigid r-chain starting from a, and restrict adjacent tile types to be compatible:

$$\Box(\top \sqsubseteq \exists r.\top), \quad \Box\Big(P_t \sqsubseteq \bigsqcup_{(t,t') \in V} \forall r.P_{t'}\Big)$$

For each time point i, we mark all individuals along the r-chain with the concept name $A^{(i \bmod 3)}$, by using the following formulae, for $0 \leq i \leq 2$:

$$A^0(a), \quad \Box\big(A^i(a) \to \bigcirc A^{i+1}(a)\big), \quad \Box(A^i \sqsubseteq \neg A^{i+1} \sqcap \forall r.A^i)$$

To encode the compatibility of horizontally adjacent tiles, we add the following formulae, for $0 \leq i \leq 2$ and $t \in T$:

$$\Box\Big(P_t \sqcap A^i \sqsubseteq \bigsqcup_{(t,t') \in H} N^i_{t'}\Big), \quad \Box(N^i_t \sqcap A^{i+1} \sqsubseteq P_t), \quad \Box(A^{i-1} \sqsubseteq \neg N^i_t)$$

These express that any domain element with tile type t (expressed by P_t) at a time point marked with A^i must have a compatible type t' at the next time point (expressed by $N^i_{t'}$). Since all $N^i_{t'}$ are false at the previous time point (designated by A^{i-1}) and $\mathsf{iRig}(N^i_{t'}) = 2$, any $N^i_{t'}$ that holds at the current time point is still active at the next time point (described by A^{i+1}), where it then implies $P_{t'}$.

Finally, we express the condition on t_0 via the formula $\Box\Diamond P_{t_0}(a)$. We now obtain the claimed undecidability from known results about the tiling problem [13].

Theorem 6. *Satisfiability in $\mathcal{ALC}$-LTL$_{|gGCI}$ w.r.t. rigid roles and interval-rigid concepts is Σ^1_1-hard, and thus not even recursively enumerable.*

4.2 Interval-Rigid Roles

Since rigid roles make the logic undecidable, we consider the case where instead only interval-rigid roles (and concepts) are allowed. While interval-rigid concepts can be expressed using $\Box_{[0,k)}$ in GCIs (see Section 3), this does not work for interval-rigid roles, since this would require (at least) inclusion axioms on roles. However, the presence of such axioms (denoted by replacing $\mathcal{ALC}$ with $\mathcal{ALCH}$ [7]) immediately causes undecidability in conjunction with interval-rigid roles.

Theorem 7. *Satisfiability in $\mathcal{ALCH}$-LTL w.r.t. interval-rigid names is Σ^1_1-hard.*

Without role inclusions, we obtain 2-ExpTime-hardness by an easy adaptation of the proof of 2-ExpTime-hardness for $\mathcal{ALC}$-LTL$_{|gGCI}$ with rigid roles from [8].

Theorem 8. *Satisfiability in $\mathcal{ALC}$-LTL$_{|gGCI}$ with respect to interval-rigid names is* 2-ExpTime*-hard.*

4.3 Rigid and Interval-Rigid Concepts

As the last setting, we consider the case where only concept names can be rigid or interval-rigid, for which we show ExpSpace-completeness. For the upper bound, recall from Section 3 that rigid concepts and interval-rigid concepts are expressible in $\mathrm{LTL}^{0,\infty}_{\mathcal{ALC}}$ via global GCIs, so that we can apply Theorem 1. The same observation yields an ExpTime upper bound for satisfiability in $\mathcal{ALC}$-LTL w.r.t. global GCIs, which is tight since satisfiability in ordinary $\mathcal{ALC}$ is already ExpTime-hard [17].

We show the ExpSpace lower bound by a reduction from satisfiability of $\mathcal{ALC}$-LTL$^{\bigcirc}$, the extension of $\mathcal{ALC}$-LTL in which $\bigcirc$ can be applied to concepts, to satisfiability of $\mathcal{ALC}$-LTL w.r.t. interval-rigid concepts. It is shown in [14, Theorem 11.33] that satisfiability in (a syntactic variant of) $\mathcal{ALC}$-LTL$^{\bigcirc}$ is ExpSpace-hard. To simulate $\bigcirc$ using interval-rigid concept names, we use a similar construction as in Section 4.1, where we mark all individuals at time point i with $A^{(i \bmod 3)}$, and use 2-rigid concept names to transfer information between time points. More precisely, we first define an $\mathcal{ALC}$-LTL formula ψ as the conjunction of the following formulae, where $0 \leq i \leq 2$:

$$(\top \sqsubseteq A^0), \quad \Box((\top \sqsubseteq A^i) \to \bigcirc(\top \sqsubseteq A^{i+1})), \quad \Box(A^i \sqsubseteq \neg A^{i+1})$$

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 $\leq$ [Th. 11]	ExpSpace $\leq$ [Th. 11]	ExpSpace
$\mathcal{ALC}$-LTL$^{\mathsf{bin}}_{\lvert gGCI}$	2-ExpTime	ExpSpace	ExpSpace $\geq$ [1]
$\mathcal{ALC}$-LTL	2-ExpTime	NExpTime [8]	ExpTime $\leq$ [8]
$\mathcal{ALC}$-LTL$_{\lvert gGCI}$	2-ExpTime $\geq$ [8]	ExpTime $\leq$ [8]	ExpTime $\geq$ [17]

We now simulate concepts of the form $\bigcirc C$ via fresh, 2-rigid concept names $A^i_{\bigcirc C}$, $0 \leq i \leq 2$. Given any $\mathcal{ALC}$-LTL$^{\bigcirc}$ formula or concept α, we denote by $\alpha^{\bigcirc}$ the result of replacing each outermost concept of the form $\bigcirc C$ in α by

$$\bigsqcup_{0 \leq i \leq 2} (A^i_{\bigcirc C} \sqcap A^i).$$

To express the semantics of $\bigcirc C$, we use the conjunction $\psi_{\bigcirc C}$ of the following formulae:

$$\Box(A^i_{\bigcirc C} \sqcap A^{i+1} \sqsubseteq C^{\bigcirc}), \quad \Box(C^{\bigcirc} \sqcap A^{i+1} \sqsubseteq A^i_{\bigcirc C}), \quad \Box(A^{i-1} \sqsubseteq \neg A^i_{\bigcirc C})$$

As in Section 4.1, $A^i_{\bigcirc C}$ must either be satisfied at both time points designated by A^i and A^{i+1}, or at neither of them. Furthermore, an individual satisfies $\bigcirc C$ iff it satisfies $A^i_{\bigcirc C} \sqcap A^i$ for some i, $0 \leq i \leq 2$.

One can show that an $\mathcal{ALC}$-LTL$^{\bigcirc}$ formula ϕ is satisfiable iff the $\mathcal{ALC}$-LTL formula $\phi^{\bigcirc} \wedge \psi \wedge \bigwedge_{\bigcirc C \in \mathsf{csub}(\phi)} \psi_{\bigcirc C}$ is satisfiable.

Theorem 9. *Satisfiability in $\mathcal{ALC}$-LTL with respect to interval-rigid concepts is* ExpSpace*-hard.*

5 $\mathcal{ALC}$-LTL$^{\mathsf{bin}}$ without Interval-Rigid Names

To conclude our investigation of metric temporal DLs, we consider the setting of $\mathcal{ALC}$-LTL$^{\mathsf{bin}}$ without interval-rigid names. Table 3 summarizes the results of this section, where we also include the known results about $\mathcal{ALC}$-LTL for comparison [8]. Observe that all lower bounds follow from known results. In particular, ExpSpace-hardness for $\mathcal{ALC}$-LTL$^{\mathsf{bin}}_{\lvert gGCI}$ is inherited from LTL$^{\mathsf{bin}}$ [1,2], while rigid role names increase the complexity to 2-ExpTime in $\mathcal{ALC}$-LTL$_{\lvert gGCI}$ [8].

The upper bounds can be shown using a unified approach that was first proposed in [8]. The idea is to split the satisfiability test into two parts: one for the temporal and one for the DL dimension. In what follows, let ϕ be an $\mathcal{ALC}$-LTL$^{\mathsf{bin}}$ formula. The *propositional abstraction* ϕ^{p} is the propositional LTL$^{\mathsf{bin}}$ formula obtained from ϕ by replacing every $\mathcal{ALC}$ axiom by a propositional variable in such a way that there is a 1:1 relationship between the $\mathcal{ALC}$ axioms $\alpha_1, \ldots, \alpha_m$ occurring in ϕ and the propositional variables $p_1, \ldots, p_m$ in ϕ^{p}.

The goal is to try to find a model of ϕ^{p} and then use it to construct a model of ϕ (if such a model exists). While satisfiability of ϕ implies that ϕ^{p} is also satisfiable, the converse is not true. For example, the propositional abstraction $p \wedge q \wedge \neg r$ of $\phi = A \sqsubseteq B \wedge A(a) \wedge \neg B(a)$ is satisfiable, while ϕ is not. To rule out such cases, we collect the propositional worlds occurring in a model of ϕ^{p} into a (non-empty) set $\mathcal{W} \subseteq 2^{\{p_1, \ldots, p_m\}}$, which is then used to check the satisfiability of the original formula (w.r.t. rigid names). This is captured by the LTL$^{\mathsf{bin}}$ formula $\phi^{\mathrm{p}}_{\mathcal{W}} := \phi^{\mathrm{p}} \wedge \phi_{\mathcal{W}}$,

where $\phi_{\mathcal{W}}$ is the (exponential) LTL formula

$$\Box \bigvee_{W \in \mathcal{W}} \left(\bigwedge_{p \in W} p \wedge \bigwedge_{p \in \overline{W}} \neg p \right)$$

in which $\overline{W} := \{p_1, \dots, p_m\} \setminus W$ denotes the complement of W. The formula $\phi^{\mathrm{p}}_{\mathcal{W}}$ states that, when looking for a propositional model of ϕ^{p}, we are only allowed to use worlds from $\mathcal{W}$.

Since satisfiability of ϕ implies satisfiability of $\phi^{\mathrm{p}}_{\mathcal{W}}$ for some $\mathcal{W}$, we can proceed as follows: choose a set of worlds $\mathcal{W}$, test whether $\phi^{\mathrm{p}}_{\mathcal{W}}$ is satisfiable, and then check whether a model with worlds from $\mathcal{W}$ can indeed be lifted to a temporal DL interpretation (respecting rigid names). To check the latter, we consider the conjunction $\bigwedge_{p_j \in W} \alpha_j \wedge \bigwedge_{p_j \in \overline{W}} \neg \alpha_j$ for every $W \in \mathcal{W}$. However, the rigid names require that all these conjunctions are *simultaneously* checked for satisfiability. To tell apart the *flexible* names X occurring in different elements of $\mathcal{W} = \{W_1, \dots, W_k\}$, we introduce copies $X^{(i)}$ for all $i \in [1, k]$. The axioms $\alpha_j^{(i)}$ are obtained from α_j by replacing every flexible name X by $X^{(i)}$, which yields the following conjunction of exponential size:

$$\chi_{\mathcal{W}} := \bigwedge_{i=1}^{k} \Big(\bigwedge_{p_j \in W_i} \alpha_j^{(i)} \wedge \bigwedge_{p_j \in \overline{W_i}} \neg \alpha_j^{(i)} \Big).$$

The following characterization from [8] can be easily adapted to our setting:

Lemma 10 (Adaptation of [8])**.** *An $\mathcal{ALC}$-LTL$^{\mathsf{bin}}$ formula ϕ is satisfiable w.r.t. rigid names iff a set $\mathcal{W} \subseteq 2^{\{p_1, \dots, p_m\}}$ exists so that $\phi^{\mathrm{p}}_{\mathcal{W}}$ and $\chi_{\mathcal{W}}$ are both satisfiable.*

To obtain the upper bounds in Table 3, recall from Section 2 that there is an exponentially larger LTL formula $\phi^{\mathrm{p}\prime}$ that is equivalent to the LTL$^{\mathsf{bin}}$ formula ϕ^{p}. Since $\phi_{\mathcal{W}}$ is also an LTL formula of exponential size, satisfiability of the conjunction $\phi^{\mathrm{p}\prime} \wedge \phi_{\mathcal{W}}$ can be checked in ExpSpace. Since the complexity of the satisfiability problem for $\chi_{\mathcal{W}}$ remains the same as in the case of $\mathcal{ALC}$-LTL, we obtain the claimed upper bounds from the techniques in [8]. This means that, in most cases, the complexity of the DL part is dominated by the ExpSpace complexity of the temporal part. The only exception is the 2-ExpTime-bound for $\mathcal{ALC}$-LTL$^{\mathsf{bin}}$ with rigid names.

Theorem 11. *Satisfiability in $\mathcal{ALC}$-LTL$^{\mathsf{bin}}$ is in 2-ExpTime w.r.t. rigid names, and in ExpSpace w.r.t. rigid concepts.*

6 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.

In principle, the arguments for $\mathcal{ALC}$-LTL$^{\mathsf{bin}}$ in Section 5 are also applicable if we replace $\mathcal{ALC}$ by the light-weight DLs *DL-Lite* or $\mathcal{EL}$, yielding tight complexity bounds based on the known results from [5,9]. It would be interesting to investigate temporal DLs based on *DL-Lite* and $\mathcal{EL}$ with interval-rigid roles and metric operators.

References

[1] Rajeev Alur and Thomas A. Henzinger. Real-time logics: Complexity and expressiveness. *Inf. Comput.*, 104(1):35–77, 1993.

[2] Rajeev Alur and Thomas A. Henzinger. A really temporal logic. *J. ACM*, 41(1):181–204, 1994.

[3] A. Artale, D. Bresolin, A. Montanari, G. Sciavicco, and V. Ryzhikov. DL-Lite and interval temporal logics: A marriage proposal. In *Proc. of the 21st Eur. Conf. on Artificial Intelligence (ECAI'14)*, pages 957–958. IOS Press, 2014.

[4] A. Artale, R. Kontchakov, V. Ryzhikov, and M. Zakharyaschev. Tractable interval temporal propositional and description logics. In *Proc. of the 29th AAAI Conf. on Artificial Intelligence (AAAI'15)*, pages 1417–1423. AAAI Press, 2015.

[5] Alessandro Artale, Roman Kontchakov, Carsten Lutz, Frank Wolter, and Michael Zakharyaschev. Temporalising tractable description logics. In *Proc. of the 14th Int. Symp. on Temporal Representation and Reasoning (TIME'07)*, pages 11–22. IEEE Press, 2007.

[6] Alessandro Artale, Carsten Lutz, and David Toman. A description logic of change. In *Proc. of the 20th Int. Joint Conf. on Artificial Intelligence (IJCAI'07)*, pages 218–223, 2007.

[7] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2nd edition, 2007.

[8] Franz Baader, Silvio Ghilardi, and Carsten Lutz. LTL over description logic axioms. *ACM Trans. Comput. Log.*, 13(3):21:1–21:32, 2012.

[9] Stefan Borgwardt and Veronika Thost. Temporal query answering in the description logic $\mathcal{EL}$. In *Proc. of the 24th Int. Joint Conf. on Artificial Intelligence (IJCAI'15)*, pages 2819–2825. AAAI Press, 2015.

[10] Sebastian Brandt, Elem Güzel Kalaycı, Roman Kontchakov, Vladislav Ryzhikov, Guohui Xiao, and Michael Zakharyaschev. Ontology-based data access with a Horn fragment of metric temporal logic. In *Proc. of the 31st AAAI Conf. on Artificial Intelligence (AAAI'17)*, pages 1070–1076. AAAI Press, 2017.

[11] Christopher L. Crowe and Cui Tao. 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.

[12] 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.

[13] David Harel. Effective transformations on infinite trees, with applications to high undecidability, dominoes, and fairness. *J. ACM*, 33(1):224–248, 1986.

[14] Agi Kurucz, Frank Wolter, Michael Zakharyaschev, and Dov M Gabbay. *Many-dimensional modal logics: Theory and applications*. Gulf Professional Publishing, 2003.

[15] Carsten Lutz, Dirk Walther, and Frank Wolter. Quantitative temporal logics over the reals: PSPACE and below. *Inf. Comput.*, 205(1):99–123, 2007.

[16] Carsten Lutz, Frank Wolter, and Michael Zakharyaschev. Temporal description logics: A survey. In *Proc. of the 15th Int. Symp. on Temporal Representation and Reasoning (TIME'08)*, pages 3–14. IEEE Press, 2008.

[17] Klaus Schild. A correspondence theory for terminological logics: Preliminary report. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI'91)*, pages 466–471. Morgan Kaufmann, 1991.

[18] Stefan Schulz, Kornél Markó, and Bontawee Suntisrivaraporn. 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.

[19] Frank Wolter and Michael Zakharyaschev. Temporalizing description logics. In *Frontiers of Combining Systems 2*, pages 379–402. Research Studies Press/Wiley, 2000.

A Proofs for Section 2

We now discuss Theorem 1, which for convenience we restate below.

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.*

The lower bounds all follow from the results in [12,16]. For the 2-ExpSpace upper bound, we refer to Theorem 4. As we already mentioned, ExpSpace-completeness of $\mathrm{LTL}_{\mathcal{ALC}}$ has already been shown in [19]. The remaining results are provided in Lemmata 13 and 14 in the following.

The proofs in [12,16] for the remaining cases of $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ and $\mathrm{LTL}_{\mathcal{ALC}}$ w.r.t. global GCIs (but without assertions) rely on the fact that, if a formula in these logics is satisfiable, then one can always find a quasimodel $(S, \mathfrak{R})$ that is *monotone*, in the sense that, for all $i \in \mathbb{N}$,

$$S(i) \supseteq S(i+1).$$

With assertions of the form $C(a)$ or $r(a,b)$, each quasistate must have exactly one named type per individual, so the quasimodels cannot be monotone in the sense described above. In what follows, we point out that if a formula in these logics has a quasimodel, then it has a quasimodel that is monotone in its 'unnamed part' and sketch the necessary changes (using this modified notion of monotonicity) to extend the results obtained in [12,16] for $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ and $\mathrm{LTL}_{\mathcal{ALC}}$ w.r.t. global GCIs to our formulas, which contain assertions.

First we extend the quasimodel construction from [12,16] to deal with assertions. We use the same notions of concept/formula/named types from Section 3.1 for an $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$/$\mathrm{LTL}_{\mathcal{ALC}}$ formula $\varphi = \Box\mathcal{T} \wedge \phi$ with global GCIs. To encode the fact that GCIs hold on the whole time line, we require (in addition to **T1'** and **T2'**) that all GCIs in $\mathcal{T}$ occur in all formula types. Here, a quasistate Q for φ is a set of types such that:

- Q contains exactly one formula type t_Q;
- Q contains exactly one named type t_a for each $a \in \mathsf{ind}(\varphi)$;
- if $t \in Q$ and $\exists s.D \in t$, then there is a $t' \in Q$ with $\{D\} \cup \{\neg E \mid \neg\exists s.E \in t\} \subseteq t'$.
- for all $C \sqsubseteq D \in \mathsf{cl}^{\mathsf{f}}(\varphi)$, we have $C \sqsubseteq D \in t_Q$ iff $C \in t$ implies $D \in t$ for all concept types $t \in Q$;
- for all $C(a) \in \mathsf{cl}^{\mathsf{f}}(\varphi)$, we have $C(a) \in t_Q$ iff $C \in t_a$; and
- for all $s(a,b) \in \mathsf{cl}^{\mathsf{f}}(\varphi)$, we have $s(a,b) \in t_Q$ implies $\{\neg E \mid \neg\exists s.E \in t_a\} \subseteq t_b$.

We also adapt the notion of a (concept/formula) run, which here is a sequence $r = r(0)r(1)\dots$ of types such that, for all $n \geq 0$:

- for all $a \in \mathsf{ind}(\varphi)$, $a \in r(0)$ iff $a \in r(n)$;
- for all $\bigcirc\alpha \in \mathsf{cl}^*(\varphi)$, $\bigcirc\alpha \in r(n)$ iff $\alpha \in r(n+1)$; and
- for all $\alpha\,\mathcal{U}_I\beta \in \mathsf{cl}^*(\varphi)$, $\alpha\,\mathcal{U}_I\beta \in r(n)$ iff there is $j \in I$ such that $\beta \in r(n+j)$ and $\alpha \in r(n+i)$, for all $i \in [0, j)$.

where cl^* is cl^{c} for concept runs and cl^{f} for formula runs, and the first condition does not apply to formula runs. Note that this definition of runs is equivalent to the definition provided in Section 3.1, only that we do not refer to run segments explicitly this time.

Finally, here a quasimodel is a pair $(S, \mathfrak{R})$, where S is an infinite sequence of quasistates and $\mathfrak{R}$ is a set of runs such that:

- $\varphi \in t_{S(0)}$;
- $r(n) \in S(n)$, for all $n \geq 0$ and $r \in \mathfrak{R}$; and
- for all $t \in S(n)$, $n \geq 0$, there is a run $r \in \mathfrak{R}$ such that $r(n) = t$.

Given a quasistate Q, we denote by Q^{u} the *unnamed* part of Q, that is, the maximal subset of Q, that does not contain named types or formula types. We say that a quasimodel $(S, \mathfrak{R})$ is u-*monotone* if, for all $i \in \mathbb{N}$,

$$S^{\mathsf{u}}(i) \supseteq S^{\mathsf{u}}(i+1).$$

Lemma 12 (Direct adaptation from [12,16])**.** *An* $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$/$\mathrm{LTL}_{\mathcal{ALC}}$ *formula is satisfiable w.r.t. global GCIs iff it has a* u*-monotone quasimodel.*

We now argue that we can extend the upper bound for $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ w.r.t. global GCIs in [12] to our formulae with assertions. Note that the two logics we consider here do not have (interval-)rigid names and we use ℓ_φ to denote the largest number occurring in φ. We use $\natural_\varphi$ to denote the size of the (exponential) set of all possible types for φ. The main idea in the argument provided in [12] is that if there is a quasimodel, then we can assume w.l.o.g. that there is a quasimodel $(S, \mathfrak{R})$ where S is of the form

$$Q_0^{n_0} \dots Q_{m-1}^{n_{m-1}} Q_m^{\omega}$$

for quasistates $Q_0, \dots, Q_m$ with $Q_i \supsetneq Q_{i+1}$, $0 \leq i < m$, and numbers $n_0, \dots, n_{m-1}$ double exponentially bounded in $|\varphi|$. Since these numbers can be encoded in binary using exponential space, this regularity allows the authors to devise a non-deterministic ExpSpace algorithm to decide satisfiability of $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ w.r.t. global GCIs. Using our modified notion of monotonicity (Lemma 12), we can assume w.l.o.g. that if there is a quasimodel then there is a quasimodel $(S, \mathfrak{R})$ where the unnamed part of S is of the form

$$(Q_0^{\mathsf{u}})^{n_0} \dots (Q_{m-1}^{\mathsf{u}})^{n_{m-1}} (Q_m^{\mathsf{u}})^{\omega}$$

with $Q_i^{\mathsf{u}} \supsetneq Q_{i+1}^{\mathsf{u}}$, $0 \leq i < m$, and numbers $n_0, \dots, n_{m-1}$ double exponentially bounded in $|\varphi|$.

We cannot obtain the same kind of monotonicity if we take into account the named part of quasimodels, but we can find a different regularity that is sufficient to decide the existence of quasimodels in exponential space. Note that a regular quasimodel as above takes $n_a = \sum_{i<m} n_i + 1$ time points until the unnamed part stabilizes, where n_a is still double exponentially bounded. That is, for all $i > n_a$, we have $S^{\mathsf{u}}(i) = S^{\mathsf{u}}(n_a)$, and therefore, in this part of the

quasimodel, the only variation happens due to the formula run and the linear number of named runs. By [12, Lemma 4], for every type $t \in S(i)$, we can find a run that goes through t in $S(i)$ which is of the following regular form:

$$r(0) \dots r(n)(r(n+1) \dots r(n+m))^\omega,$$

where n and m are double exponentially bounded in the size of the input. Using a similar idea as in the proof for this lemma, and the fact that there is only a linear number of named and formula runs in each quasimodel, one can show that any quasimodel can be transformed into a quasimodel for the same formula which is of the following form:

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

where $n > n_a$, and both n and m are double exponentially bounded. This allows one to devise a non-deterministic ExpSpace-algorithm for satisfiability as follows: first construct the unnamed part of the quasimodel, then guess the numbers n and m and each quasistate one by one, where we keep a constant number of unnamed quasistates in memory to decide the satisfaction of until-formulae. (See also Appendix B for a similar algorithm.) We thus have the following result.

Lemma 13. *Satisfiability in* $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ *w.r.t. global GCIs is* ExpSpace*-complete.*

Following the argument provided in [16], one can further show membership in ExpTime of satisfiability in $\mathrm{LTL}_{\mathcal{ALC}}$ w.r.t. global GCIs, by using a type elimination procedure to construct sets $S_0, \dots, S_n$ with $S_i \supseteq S_{i+1}$ and n exponentially bounded in $|\varphi|$. If we apply a similar procedure to our formula with assertions, these sets are not yet quasistates, since they do not ensure the conditions related to named and formula types. To check the existence of a quasimodel, one can first use a similar type elimination procedure to find sets $S_0, \dots, S_n$ with $S_i^{\mathsf{u}} \supseteq S_{i+1}^{\mathsf{u}}$ (the unnamed part of the quasimodel) and then consider the graph (V_S, E_S), where the set V_S of vertices is the set of all quasistates S'_i such that $S_i'^{\mathsf{u}} = S_i^{\mathsf{u}}$. It follows from Lemma 12 that V_S has at most $(\natural_\varphi)^{|\mathsf{ind}(\varphi)|+2}$ many quasistates.

To define the edges of our graph, we use a notion of compatibility between types. More specifically, we say that *a pair* (t, t') *of types is compatible* if there is a run r where $r(n) = t$ and $r(n+1) = t'$, for some $n \geq 0$. Then, E_S contains all pairs $(S'_i, S'_j) \in V_S \times V_S$ such that there is a compatibility relation between the types with all types in S'_i in the domain and all types in S'_j in the range of this relation. We can then decide the existence of a regular quasimodel by performing a series of reachability tests in this graph (see Appendix C for a similar and more detailed argument on how to construct the named part of a quasimodel using a graph with quasistates as vertices).

Lemma 14. *Satisfiability in* $\mathrm{LTL}_{\mathcal{ALC}}$ *w.r.t. global GCIs is* ExpTime*-complete.*

We now move to the proof of Theorem 2.

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

Proof. We first describe the reduction of temporal concepts. The idea from [15] is to use a counter to determine the temporal distance to the *nearest* time point that satisfies a concept of the form $C\,\mathcal{U}_{[0,c]}\,D$ (i.e., it satisfies D, and C is satisfied at all time points in between). This counter counts (backwards in time) from 0 up to $c+1$, and then stays at $c+1$; the concept $C\,\mathcal{U}_{[0,c]}\,D$ is satisfied iff the counter value is $\leq c$. The counter is represented by fresh concept names, one for each bit in the binary representation, of which there are polynomially many.

This suffices, since for each individual and each time point there is a unique nearest time point satisfying $C\,\mathcal{U}_{[0,c]}D$, which uniquely determines the counter value. For a concept of the form $C\,\mathcal{U}_{[c,\infty)}D$, we can use a similar counter, which however determines the distances to the *furthest* time point that satisfies the concept. Then, $C\,\mathcal{U}_{[c,\infty)}D$ is satisfied iff the counter value is $\geq c$.

More formally, let φ be an $\mathrm{LTL}^{0,\infty}_{\mathcal{ALC}}$ formula. For each subconcept of the form $F = C\,\mathcal{U}_I D$ occurring in φ, where I is either $[0,c]$ or $[c,\infty)$, we introduce fresh concept names $A^F_0, \dots, A^F_{\ell_F}$, where $\ell_F = \lceil \log(c+1) \rceil - 1$. We now define a recursive transformation $\cdot^*$ on $\mathrm{LTL}^{0,\infty}_{\mathcal{ALC}}$ concepts as follows:

$$\begin{aligned}
\top^* &:= \top \\
A^* &:= A \quad \text{for all } A \in \mathsf{N_C} \\
(\neg C)^* &:= \neg C^* \\
(C \sqcap D)^* &:= C^* \sqcap D^* \\
(\bigcirc C)^* &:= \bigcirc C^* \\
(C\,\mathcal{U}_{[0,c]}D)^* &:= (A^{C\,\mathcal{U}_{[0,c]}\,D} \leq c) \\
(C\,\mathcal{U}_{[c,\infty)}D)^* &:= (A^{C\,\mathcal{U}_{[c,\infty)}\,D} \geq c),
\end{aligned}$$

where $(A^F \leq c)$ is the concept

$$(C^*\,\mathcal{U}\,D^*) \sqcap \left(\prod_{i=0}^{\ell_F} (A^F_i = c_i) \sqcup \bigsqcup_{i=0}^{\ell_F} \left((A^F_i < c_i) \sqcap \prod_{j=i+1}^{\ell_F} (A^F_i = c_i) \right) \right).$$

Here, c_i denotes the i-th bit of c, and $(A^F_i = c_i)$ is defined as A^F_i if $c_i = 1$, and $\neg A^F_i$ otherwise. Similarly, $(A^F_i < c_i)$ is $\neg A^F_i$ if $c_i = 1$, and $\bot$ otherwise. The concept $(A^F \leq c)$ simply compares the counter value represented by the concept names A^F_i to the value of c. The concept $C\,\mathcal{U}\,D$ ensures that the counter value is well-defined, i.e., that there actually exists a future time point that satisfies D such that C remains satisfied in the meantime. We similarly define $(A^F \geq c)$, and finally obtain φ^* from φ by replacing every concept C by C^*.

To describe the behaviour of the counter, we use the global GCI $\top \sqsubseteq B^F_0 \sqcap B^F_1 \sqcap B^F_2$, where for $F = C\,\mathcal{U}_{[0,c]}D$ we define

$$\begin{aligned}
B^F_0 &:= D^* \leftrightarrow (A^F = 0), \\
B^F_1 &:= (C^* \sqcap \neg D^* \sqcap \bigcirc(A^F \leq c)) \rightarrow \prod_{i=0}^{\ell_F} \left(\left(\prod_{j=0}^{i-1} \bigcirc A^F_j \right) \leftrightarrow \left(\left(\bigcirc A^F_i\right) \leftrightarrow \left(\neg A^F_i\right) \right) \right), \\
B^F_2 &:= (C^* \sqcap \neg D^* \sqcap \bigcirc(A^F = c+1)) \rightarrow (A^F = c+1).
\end{aligned}$$

These three concepts express that the counter A^F is reset to 0 whenever we encounter D, it is increased (backwards in time) up to $c+1$ as long as C remains satisfied, and it remains at $c+1$ afterwards (until we encounter the next D or C is not satisfied any more). The big conjunction in B^F_1 expresses that a bit i of the counter is flipped (from 0 to 1 or from 1 to 0) iff all lower bits are 1 at the next time point, because then a carry bit has to be added. These concepts formalize the intuition described in the beginning of this proof, that the counter A^F represents the time until the nearest occurrence of D that satisfies the until-concept. Then, instead of F, we can equivalently use the concept $F^* = (A^F \leq c)$ to check whether the counter value is at most c.

For $F = C\,\mathcal{U}_{[c,\infty)}\,D$, we similarly define

$$B_0^F := (D^* \sqcap \bigcirc\neg(C\,\mathcal{U}D)) \leftrightarrow (A^F = 0),$$
$$B_1^F := (C^* \sqcap \bigcirc(A^F \leq c)) \rightarrow \prod_{i=0}^{\ell_F}\left(\left(\prod_{j=0}^{i-1}\bigcirc A_j^F\right) \leftrightarrow \left(\left(\bigcirc A_i^F\right) \leftrightarrow \left(\neg A_i^F\right)\right)\right),$$
$$B_2^F := (C^* \sqcap \bigcirc(A^F = c+1)) \rightarrow (A^F = c+1).$$

The differences to the previous case are that the counter is only reset to 0 at the last possible D, i.e., whenever $C\,\mathcal{U}D$ does not hold afterwards, and that the counter is increased regardless of D. This again reflects the intuition described above, that A^F marks the distance to the last possible occurrence of D that satisfies F. Intermediate occurrences of D can simply be ignored, as long as C remains satisfied.

It is easy to check that φ is satisfiable iff $\varphi^* \wedge \bigwedge_F \Box(\top \sqsubseteq B_0^F \sqcap B_1^F \sqcap B_2^F)$ is satisfiable, where F ranges over all until-concepts in φ that are not already of the form $C\,\mathcal{U}D$.

To simulate until-*formulae* ψ in φ, we use exactly the same construction, where we replace concepts by formulae, i.e.,

- instead of A_i^F we use the GCI $A_i^\psi(a)$, where a is a fresh individual name,
- instead of $\sqcap$ we use $\wedge$,
- we define the translation $\cdot^*$ similarly on formulae,
- instead of $\top \sqsubseteq B_0^F \sqcap B_1^F \sqcap B_2^F$ we use $B_0^\psi \wedge B_1^\psi \wedge B_2^\psi$.

Overall, this yields an $\mathrm{LTL}_{\mathcal{ALC}}$ formula of polynomial size that is satisfiable iff φ is satisfiable.

To apply this construction to sublogics of $\mathrm{LTL}_{\mathcal{ALC}}^{0,\infty}$, observe that we only introduce global GCIs, and hence satisfiability in $\mathrm{LTL}_{\mathcal{ALC}}^{0,\infty}$ w.r.t. global GCIs can be polynomially reduced to satisfiability in $\mathrm{LTL}_{\mathcal{ALC}}$ w.r.t. global GCIs. Moreover, the reduction is not affected by the (interval-)rigidity of the used names, and temporal operators on the concept level are only used if they were already present in φ. □

B Proofs for Section 3.1

We first prove that the existence of quasimodels completely captures satisfiability of $\mathrm{LTL}_{\mathcal{ALC}}^{\mathsf{bin}}$-formulae.

Lemma 15. *The* $\mathrm{LTL}_{\mathcal{ALC}}^{\mathsf{bin}}$ *formula* φ *is satisfiable w.r.t. interval-rigid names iff there is a quasimodel for* φ*.*

Proof. ($\Rightarrow$) Assume there is a model $\mathfrak{I} = (\Delta^\mathfrak{I}, (\mathcal{I}_i)_{i\in\mathbb{N}})$ of φ that respects interval-rigid names. We can assume without loss of generality that no unnamed domain element has a named element as a role successor. If this is not the case, we can easily generate such a model by unraveling [7], thereby creating unnamed copies of the named elements whenever they appear as role successors. We now associate a concept run r_d to every domain element $d \in \Delta^\mathfrak{I}$ by setting

$$r_d(i) = \{C \in \mathsf{cl}^{\mathsf{c}}(\varphi) \mid d \in C^{\mathcal{I}_i}\} \cup \{a \in \mathsf{ind}(\varphi) \mid d = a^{\mathcal{I}_i}\}.$$

Moreover, we define the formula run $r_\mathfrak{I}$ by $r_\mathfrak{I}(i) = \{\psi \in \mathsf{cl}^{\mathsf{f}}(\varphi) \mid \mathfrak{I}, i \models \psi\}$. We now set $\mathfrak{R} = \{r_d \mid d \in \Delta^\mathfrak{I}\} \cup \{r_\mathfrak{I}\}$, and define the infinite sequence of quasistates $S(i)$, $i \in \mathbb{N}$, as follows.

For $i \in \mathbb{N}$, $\mathcal{R}_{S(i)}$ contains all run segments $r^{[i,i+\ell_\varphi]}$ with $r \in \mathfrak{R}$. Furthermore, $\mathcal{C}_{S(i)}$ is the set of role constraints $\sigma \overset{s}{k} \sigma'$, such that $\sigma = r_d^{[i,i+\ell_\varphi]}$, $\sigma' = r_{d'}^{[i,i+\ell_\varphi]}$, $k \in [1, \mathsf{iRig}(s)]$, $(d,d') \in s^{\mathcal{I}_\ell}$ for all $\ell \in (n-k, n-k+\mathsf{iRig}(s)]$, and $(d,d') \notin s^{\mathcal{I}_{n-k}}$. It is straightforward to show that $(S, \mathfrak{R})$ is a quasimodel. In particular, the maximal compatibility relations π_i, $i \in \mathbb{N}$, must contain the pairs $(r^{[i,i+\ell_\varphi]}, r^{[i+1,i+1+\ell_\varphi]})$ with $r \in \mathfrak{R}$.

($\Leftarrow$) Assume there is a quasimodel $(S, \mathfrak{R})$ for φ. We define the $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ interpretation $\mathfrak{I} = (\Delta^{\mathfrak{I}}, (\mathcal{I}_i)_{i \in \mathbb{N}})$ as follows:

$$
\begin{aligned}
\Delta^{\mathfrak{I}} &= \{d_r \mid r \in \mathfrak{R} \text{ is a concept run}\} \\
a^{\mathcal{I}_i} &= d_r \text{ for the unique } r \in \mathfrak{R} \text{ with } a \in r(0) \\
A^{\mathcal{I}_i} &= \{d_r \mid A \in r(i),\ r \in \mathfrak{R}\} \\
s^{\mathcal{I}_i} &= \{(d_r, d_{r'}) \mid r^{[j,j+\ell_\varphi]} \overset{s}{1} r'^{[j,j+\ell_\varphi]} \in S(j),\ i-j \in [0, \mathsf{iRig}(s)),\ r, r' \in \mathfrak{R}\}
\end{aligned}
$$

To see that $\mathfrak{I}$ respects interval-rigid names, consider any $(d_r, d_{r'}) \in s^{\mathcal{I}_i}$. There must be a j with $i - j \in [0, \mathsf{iRig}(s))$ and $r^{[j,j+\ell_\varphi]} \overset{s}{1} r'^{[j,j+\ell_\varphi]} \in S(j)$. But then we have $(d_r, d_{r'}) \in s^{\mathcal{I}_\ell}$ for all $\ell \in [j, j+\mathsf{iRig}(s))$, and this interval includes i.

To show that $\mathfrak{I}$ is also a model of φ, we prove the following claim.

Claim. For all concept runs $r \in \mathfrak{R}$, $C \in \mathsf{cl}^{\mathsf{c}}(\varphi)$ and $i \in \mathbb{N}$, we have

$$C \in r(i) \quad \text{iff} \quad d_r \in C^{\mathcal{I}_i}.$$

Proof of the claim. We argue by structural induction. It should be clear that it holds whenever C is a concept name. Moreover, the cases $C = \neg D$ and $C = D \sqcap E$ are straightforward. It remains to consider $\exists$, $\bigcirc$, and $\mathcal{U}_I$.

- Assume $C = \exists s.D$: if $\exists s.D \in r(i)$, then by **M1** and **Q6** there is a run segment $\sigma \in \mathcal{R}_{S(i)}$ such that $r^{[i,i+\ell_\varphi]} \overset{s}{k} \sigma \in \mathcal{C}_{S(i)}$, $D \in \sigma(0)$, and $k \in [1, \mathsf{iRig}(s)]$. By repeated application of **M1**, **C4**, and **C6**, we can find another run segment $\sigma' \in \mathcal{R}_{S(i-(k-1))}$ such that $r^{[i-(k-1),i-(k-1)+\ell_\varphi]} \overset{s}{1} \sigma' \in \mathcal{C}_{S(i-(k-1))}$ and $\sigma'^{\geq k-1} = \sigma^{\leq \ell_\varphi-(k-1)}$. By **M2**, there must be a run $r' \in \mathfrak{R}$ such that $r'^{[i-(k-1),i-(k-1)+\ell_\varphi]} = \sigma'$ and $D \in r'(i)$. By the induction hypothesis and the definition of $s^{\mathcal{I}_i}$, we obtain $d_{r'} \in D^{\mathcal{I}_i}$ and $(d_r, d_{r'}) \in s^{\mathcal{I}_i}$, and hence $d_r \in (\exists s.D)^{\mathcal{I}_i}$.

  Conversely, if $d_r \in (\exists s.D)^{\mathcal{I}_i}$, then there is $r' \in \mathfrak{R}$ such that $(d_r, d_{r'}) \in s^{\mathcal{I}_i}$ and $d_{r'} \in D^{\mathcal{I}_i}$. By the induction hypothesis and the definition of $s^{\mathcal{I}_i}$, we have $D \in r'(i)$ and $r^{[j,j+\ell_\varphi]} \overset{s}{1} r'^{[j,j+\ell_\varphi]} \in \mathcal{C}_{S(j)}$ for some j with $i - j \in [0, \mathsf{iRig}(s))$. By repeated application of **C4** and **C5**, we obtain a role constraint $r^{[i,i+\ell_\varphi]} \overset{s}{i-j+1} \sigma'$, where $\sigma' \in \mathcal{R}_{S(i)}$ is such that $\sigma'^{\leq j+\ell_\varphi-i} = r'^{[i,j+\ell_\varphi]}$. Hence, we have $D \in r'(i) = \sigma'(0)$, and thus $\exists s.D \in r(i)$ by **C1**.

- Assume $C = \bigcirc D$: we have that $\bigcirc D \in r(i)$ iff $D \in r(i+1)$ (by **R1**) iff $d_r \in D^{\mathcal{I}_{i+1}}$ (by induction) iff $d_r \in (\bigcirc D)^{\mathcal{I}_i}$.

- Assume $C = D \mathcal{U}_I E$: we have that $D \mathcal{U}_I E \in r(i)$ iff there is j such that $j - i \in I$, $E \in r(j)$ and $D \in r(\ell)$ for all $\ell \in [i, j)$ (by **R3**–**R4**); by induction, this happens iff $d_r \in E^{\mathcal{I}_j}$ and $d_r \in D^{\mathcal{I}_i}$ for all $\ell \in [i, j)$, which is equivalent to $d_r \in (D \mathcal{U}_I E)^{\mathcal{I}_i}$.

Using **Q3**–**Q5** and similar arguments as above, we can now show that, for all $\psi \in \mathsf{cl}^{\mathsf{f}}(\varphi)$, it holds that $\mathfrak{I}, i \models \psi$ iff $\psi \in r(i)$, where r is the unique formula run in $\mathfrak{R}$. Hence, by **M3** we obtain that $\mathfrak{I}, 0 \models \varphi$. □

Before we show Lemma 3, we prove the following two auxiliary results.

Lemma 16. *Let $(S, \mathfrak{R})$ be a quasimodel for φ such that $S(n) = S(m)$ for $n < m$. Then $(S', \mathfrak{R}')$ with $S' = S^{\leq n} \cdot S^{>m}$ and*

$$\mathfrak{R}' = \{r_1^{\leq n} \cdot r_2^{>m} \mid r_1, r_2 \in \mathfrak{R},\ r_1^{[n,n+\ell_\varphi]} = r_2^{[m,m+\ell_\varphi]}\}$$

is also a quasimodel for φ.

Proof. We want to show that $(S', \mathfrak{R}')$ satisfies conditions **M1**–**M4**. First note that if $(S, \mathfrak{R})$ is a quasimodel for φ then $(S', \mathfrak{R}')$ satisfies **M3** and **M4**. Also, if $r_1, r_2 \in \mathfrak{R}$ and $r_1^{[n,n+\ell_\varphi]} = r_2^{[m,m+\ell_\varphi]}$ then $r_1^{\leq n} \cdot r_2^{>m}$ is a run and so, $\mathfrak{R}'$ is a set of runs. Since $S(n) = S(m)$, for every $r_1 \in \mathfrak{R}$ there is $r_2 \in \mathfrak{R}$ that satisfies $r_1^{[n,n+\ell_\varphi]} = r_2^{[m,m+\ell_\varphi]}$ and vice versa (by swapping n and m). It follows that $(S', \mathfrak{R}')$ satisfies **M2**. Finally, as $S(n) = S(m)$, the pair $(S(n), S(m+1))$ is compatible and since we require $r_1^{[n,n+\ell_\varphi]} = r_2^{[m,m+\ell_\varphi]}$, we also have that **M1** holds for $(S', \mathfrak{R}')$. □

In the following, let $\sharp_\varphi$ denote the size of the (double exponential) set of all possible run segments and role constraints for φ. We further say that a (finite or infinite) sequence r of types starting at $r(0)$ *realizes* an until-expression $\alpha \mathcal{U}_I \beta \in r(0)$ if there is an $i \in I$ with $\beta \in r(i)$ and $\alpha \in r(j)$ for all $j \in [0, i)$.

Lemma 17. *φ has a quasimodel iff there is a sequence of quasistates*

$$S(0) \dots S(n) S(n+1) \dots S(n+m)$$

such that

- *$n, m \leq (|\varphi| \cdot \sharp_\varphi^2 + 1) \cdot 2^{\sharp_\varphi}$;*
- *for all $i \in [0, n+m)$, the pair $((S(i), S(i+1))$ is compatible, with π_i being the maximal compatibility relation;*
- *$S(n) = S(n+m)$;*
- *$S(0)$ satisfies **M3** and **M4**;*
- *for every $\sigma_1 \in \mathcal{R}_{S(n)}$, there is a sequence $r = \sigma_1(0)\sigma_2(0) \dots \sigma_m(0)$ that realizes all until-expressions in $\sigma_1(0)$ such that $(\sigma_i, \sigma_{i+1}) \in \pi_{n+i}$ for all $i \in [0, m)$.*

Proof. ($\Rightarrow$) Let $(S, \mathfrak{R})$ be a quasimodel for φ and observe that the number of possible quasistates is bounded by $2^{\sharp_\varphi}$. We can assume w.l.o.g. that there is some $n \leq 2^{\sharp_\varphi}$ such that all quasistates in $S^{\geq n}$ occur infinitely often. If this is not the case, we can simply take n as the maximal number such that $S(m) \neq S(n)$ holds for all $m > n$, and use Lemma 16 to remove repeating quasistates in $S^{[0,n]}$.

Consider now an arbitrary $\alpha \mathcal{U}_I \beta \in \sigma(0)$ for some $\sigma \in \mathcal{R}_{S(n)}$, and take any $r \in \mathfrak{R}$ with $r^{[n,n+\ell_\varphi]} = \sigma$, which must exist by **M2**. Let $m' \geq n$ be the minimal number such that $r^{[n,n+m']}$ realizes $\alpha \mathcal{U}_I \beta$. Assume now that there are i, j such that $n < i < j < m'$, $r^{[i,i+\ell_\varphi]} = r^{[j,j+\ell_\varphi]}$, and $S(i) = S(j)$. By Lemma 16, there is a quasimodel $(S', \mathfrak{R}')$ for φ such that $S' = S^{\leq i} \cdot S^{>j}$ and $r^{\leq i} \cdot r^{>j}$ is a run in $\mathfrak{R}'$. It follows that we can construct a quasimodel $(S_1, \mathfrak{R}_1)$ for φ with $S_1^{\leq n} = S^{\leq n}$ and a run $r_1 \in \mathfrak{R}_1$ such that $r_1^{[n,n+\ell_\varphi]} = \sigma \in \mathcal{R}_{S_1(n)} = \mathcal{R}_{S(n)}$ and $\alpha \mathcal{U}_I \beta$ is realized by the subsequence $r_1^{[n,n+m_1]}$, for some $m_1 \leq \sharp_\varphi \cdot 2^{\sharp_\varphi}$ ($2^{\sharp_\varphi}$ is the number of possibilities for $S(i)$, and $\sharp_\varphi$ is the number of possibilities for $r^{[i,i+\ell_\varphi]}$).

Then we consider the next expression of the form $\alpha' \mathcal{U}_{I'} \beta' \in \sigma(0)$ and assume that $r_1^{[n,n+m'']}$ realizes it for some minimal $m'' \geq m_1$. Using the same construction as above, we obtain a

quasimodel $(S_2, \mathfrak{R}_2)$ for φ with $S_2^{\leq n+m_1} = S_1^{\leq n+m_1}$ and a run $r_2 \in \mathfrak{R}_2$ such that $r_2^{[n,n+\ell_\varphi]} = \sigma \in \mathcal{R}_{S(n)}$ and $r_2^{[n,n+m_2]}$ realizes both $\alpha \mathcal{U}_I \beta$ and $\alpha' \mathcal{U}_{I'} \beta'$, for some $m_2 \leq 2 \cdot \sharp_\varphi \cdot 2^{\sharp\varphi}$. We can proceed in this way and construct a quasimodel containing a run through σ that realizes all until-expressions in $\sigma(0)$ after at most $|\varphi| \cdot \sharp_\varphi \cdot 2^{\sharp\varphi}$ steps.

After that, we consider in the same manner another run segment $\sigma' \in \mathcal{R}_{S(n)}$. To realize all until-expressions in some run through σ', we need at most $|\varphi| \cdot \sharp_\varphi \cdot 2^{\sharp\varphi}$ additional steps. Since there are at most $\sharp_\varphi$ run segments in $\mathcal{R}_{S(n)}$, we require at most $|\varphi| \cdot \sharp_\varphi^2 \cdot 2^{\sharp\varphi}$ steps to realize all until-expressions in all run segments in $\mathcal{R}_{S(n)}$. We now take another $2^{\sharp\varphi}$ steps to find a time point $m \leq (|\varphi| \cdot \sharp_\varphi^2 + 1) \cdot 2^{\sharp\varphi}$ such that we have $S^*(n) = S^*(n+m)$ in the resulting quasimodel $(S^*, \mathfrak{R}^*)$ (since we assumed that $S(n)$ occurs infinitely often). By our construction, we have that all until-expressions in $S^*(n)$ can be realized by runs through $S^{*[n,n+m]}$, and hence the sequence $S^{*\leq n+m}$ satisfies all conditions required in the lemma.

($\Leftarrow$) Let now $S^*(0) \dots S^*(n+m)$ be a sequence with the given properties. We construct a quasimodel $(S, \mathfrak{R})$ for φ, where S is defined by

$$S = S^*(0) \dots S^*(n)(S^*(n+1) \dots S^*(n+m))^\omega,$$

and $\mathfrak{R}$ contains all sequences of types of the form

$$\begin{aligned} \sigma_0^0(0) \dots \sigma_n^0(0) \cdot \sigma_{n+1}^0(0) \dots \sigma_{n+m}^0(0) \cdot \\ \sigma_{n+1}^1(0) \dots \sigma_{n+m}^1(0) \cdot \\ \sigma_{n+1}^2(0) \dots \sigma_{n+m}^2(0) \cdot \dots, \end{aligned}$$

where each σ_i^j is an element of $S^*(i)$, each pair of adjacent run segments in this sequence is contained in the corresponding compatibility relation (for σ_{n+m}^j and σ_{n+1}^{j+1} we consider π_n), and there are infinitely many $j \geq 1$ for which $\sigma_{n+m}^{j-1}(0)\sigma_{n+1}^j(0) \dots \sigma_{n+m}^j(0)$ realizes all until-expressions in $\sigma_{n+m}^{j-1}(0)$.

Due to **C4**, for all $r \in \mathfrak{R}$, each subsequence of length $\ell_\varphi + 1$ is a run segment from one of the sets $S^*(i)$. To show that r is a run, we verify Condition **R4**. But this follows from the local Condition **R3**, which states that each until-expression is either satisfied by the next ℓ_φ types in r, or it is deferred. In the latter case, our construction ensures that the until-expression is not deferred indefinitely.

It remains to show that $(S, \mathfrak{R})$ satisfies Conditions **M1**–**M4**. Condition **M1** is immediately satisfied by our construction of $\mathfrak{R}$. Conditions **M3** and **M4** only concern $S(0)$, and are also satisfied since $S(0) = S^*(0)$ by construction. For Condition **M2**, for every $i \geq 0$ and $\sigma \in S(i)$, we need to find a run $r \in \mathfrak{R}$ with $r^{[i,i+\ell_\varphi]} = \sigma$. By **C3** and **C4**, we can extend σ to the left until we reach $S(0)$. The same argument holds for the other direction, but there we additionally make sure that, whenever we reach a $\sigma' \in S^*(n+m)$, we continue it in such a way that all until-expressions in σ' are realized by the next m types. This is always possible by our assumptions on the sequence $S^*(0) \dots S^*(n+m)$. □

One side result of the constructions in the above proof is that, for a satisfiable formula, we can always find a regular quasimodel.

Lemma 18. *If φ has a quasimodel, then it has a quasimodel $(S, \mathfrak{R})$ in which S is of the form*

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

such that n and m are bounded triple exponentially in the size of φ and iRig.

Proof. From any quasimodel for φ, we obtain the desired regular quasimodel by first applying one direction of the proof of Lemma 17, and then the other. □

From Lemmas 15 and 18, Lemma 3 directly follows. Based on this result, we can show ExpSpace-completeness of satisfiability of $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$-formulae.

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

Proof. Since the lower bound follows from [12], we only need to prove the upper bound. By Lemmas 15 and 17, it suffices to decide the existence of a sequence of quasistates with certain properties. We show how the latter can be decided non-deterministically in double exponential space by guessing such a sequence, quasistate after quasistate. The basic observation is that storing a single quasistate requires at most double exponential space, and that we only require a constant number of quasistates in memory at any point of the computation.

We first guess the two numbers n, m from Lemma 17, which can be stored using double exponentially many bits. We then guess the first n quasistates $S(i)$ one after another, such that Conditions **M3** and **M4** are satisfied for $S(0)$, and each pair $(S(i), S(i+1))$ of consecutive quasistates is compatible. For this, we only need to keep two consecutive quasistates in memory at any point. We now store the quasistate $S(n)$, and for each $\sigma \in \mathcal{R}_{S(n)}$ we keep a list of until-expressions that need to be satisfied, which initially contains all $\alpha \mathcal{U} \beta \in \sigma(0)$. We then guess the next m quasistates as before, where we additionally test whether there is a run starting in σ that satisfies all required until-expressions. For this, it suffices to guess which run segment of the next quasistate corresponds to this run; this of course has to be compatible with the run segment guessed for the previous quasistate. We remove any until-expression from our list that is satisfied by this guessed run segment, and require that all until-expressions must have been removed when we reach $S(n+m)$. This only requires double exponential space. Finally, after we have guessed $S(n+m)$, we accept if $S(n+m) = S(n)$. By Lemmas 15 and 17, our algorithm is sound and complete. Since at each step we only require double exponential space, we thus establish that satisfiability of $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ formulae w.r.t. interval-rigid names can be decided in 2-ExpSpace. □

C Proofs for Section 3.2

To prove Theorem 5, we show that in order to check satisfiability of an $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ formula φ with global GCIs and interval-rigid roles it suffices to consider quasimodels $(S, \mathfrak{R})$ with S of the form:

$$^{\omega}(\Xi_0) \cdot \Xi_1 \cdot (\Xi_2)^{\omega},$$

where each Ξ_i is a sequence of quasistates of at most double exponential length. Besides being regular on both sides, we can assume that the quasistates in S have the same 'unnamed part'. Recall that given a quasistate $Q = (\mathcal{R}_Q, \mathcal{C}_Q)$, we denote by $Q^{\mathsf{u}} = (\mathcal{R}^{\mathsf{u}}_Q, \mathcal{C}^{\mathsf{u}}_Q)$ the *unnamed* part of Q, that is, $\mathcal{R}^{\mathsf{u}}_Q$ and $\mathcal{C}^{\mathsf{u}}_Q$ are the maximal subsets of $\mathcal{R}_Q$ and $\mathcal{C}_Q$, respectively, that do not contain named types or formula types. We say that a role constraint is named if it contains a named run segment. Also, we say that a quasimodel $(S, \mathfrak{R})$ is u-*constant* if

- $S^{\mathsf{u}}(i) = S^{\mathsf{u}}(j)$, for all $i, j \in \mathbb{Z}$.

In the following, we show that, if φ is satisfiable, then one can always find a quasimodel that is u-constant (Lemma 20) and regular (Lemma 24). We then show a 2-ExpTime procedure that checks the existence of such quasimodel.

First, the following adaptation of Lemma 15 can be shown exactly as before.

Lemma 19. *An* $\mathrm{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ *formula* φ *with global GCIs is satisfiable over* $\mathbb{Z}$ *w.r.t. interval-rigid names iff there is a quasimodel for* φ*.*

We now proceed to show that we can restrict our search to quasimodels of the special form described above.

Lemma 20. *If φ has a quasimodel, then it has a* u*-constant quasimodel.*

Proof. Assume that $(S, \mathfrak{R})$ is a quasimodel for φ. We modify $S(n)$ by adding $\bigcup_{i\in\mathbb{Z}} S^{\mathsf{u}}(i)$. Recall that, by **C2**, our quasimodels do not have role constraints of the form $\sigma \overset{s}{k} \sigma_a$, with σ unnamed and σ_a a named run segment. We then have that each modified $S(n)$ satisfies **Q6**. As all GCIs are global, we also have that **Q3** holds. Since we do not change formula run segments, named run segments and named role constraints, Conditions **Q1**, **Q2**, **Q4**, and **Q5** still hold. Thus, our modified S is a sequence of quasistates.

We now add to $\mathfrak{R}$ all runs in

$$\{r^{\leftarrow i}, r^{\rightarrow i} \mid r \in \mathfrak{R} \text{ an unnamed concept run}, i \geq 0\},$$

where $r^{\leftarrow i}(n) := r(n-i)$ and $r^{\rightarrow i}(n) := r(n+i)$ for all $r \in \mathfrak{R}$ and $n \in \mathbb{Z}$. We then have that $(S, \mathfrak{R})$ is still a quasimodel and satisfies $S^{\mathsf{u}}(i) = S^{\mathsf{u}}(j)$ for all $i, j \in \mathbb{Z}$, as required. □

To show regularity as in Lemma 18, the last preparation we need is to establish a bound on the number of different quasistates that can occur in a quasimodel. This is done in the following lemma, where we use $\flat_\varphi$ to denote $2 \cdot |\mathsf{csub}(\varphi)| \cdot |\mathsf{ind}(\varphi)|^2 \cdot (\ell_\varphi + 1)$.

Lemma 21. *If φ has a quasimodel, then it has a quasimodel $(S, \mathfrak{R})$ with at most $(\sharp_\varphi)^{\flat_\varphi}$ many different quasistates.*

Proof. By Lemma 20, we can assume w.l.o.g. that if φ has a quasimodel $(S, \mathfrak{R})$ then $S^{\mathsf{u}}(i) = S^{\mathsf{u}}(j)$, for all $i, j \in \mathbb{Z}$. Thus, to prove this lemma it suffices to show the following claim.

Claim. If φ has a quasimodel, then it has a quasimodel with at most $\flat_\varphi$ named role constraints in each quasistate.

Let $(S, \mathfrak{R})$ be a quasimodel for φ. For all $n \in \mathbb{Z}$, $a \in \mathsf{ind}(\varphi)$, and $\exists s.D \in \sigma_a(0)$, we

$$\text{select one } \sigma_a \overset{s}{k} \sigma \in \mathcal{C}_{S(n)} \text{ with } D \in \sigma(0). \tag{$\dagger$}$$

These role constraints must exist by **Q6**. Let S' be the result of removing from S all non-selected named role constraints of the form $\sigma_a \overset{s}{k} \sigma$ with σ an unnamed run segment. Then S' is a sequence of quasistates that may not be compatible.

To regain compatibility, we proceed as follows. As $(S, \mathfrak{R})$ is a quasimodel, there is a maximal compatibility relation π_n for each pair $(S(n), S(n+1))$, $n \in \mathbb{Z}$. Then, for each selected $\sigma_a \overset{s}{k} \sigma \in \mathcal{C}_{S(n)}$ with $k < \mathsf{iRig}(s)$, there exists a role constraint

$$\sigma'_a \overset{s}{k+1} \sigma' \in \mathcal{C}_{S(n+1)} \text{ such that } \{(\sigma_a, \sigma'_a), (\sigma, \sigma')\} \subseteq \pi_n.$$

We select one such role constraint and add it to $\mathcal{C}_{S'(n+1)}$. We proceed in this way until we add a role constraint $\sigma''_a \overset{s}{\mathsf{iRig}(s)} \sigma'' \in \mathcal{C}_{S(n+\mathsf{iRig}(s)-k)}$ to $\mathcal{C}_{S'(n+\mathsf{iRig}(s)-k)}$. Similarly, if $k > 1$ and we go backwards, there exists a

$$\sigma'_a \overset{s}{k-1} \sigma' \in \mathcal{C}_{S(n-1)} \text{ such that } \{(\sigma'_a, \sigma_a), (\sigma', \sigma)\} \subseteq \pi_{n-1}.$$

We select one such role constraint, add it to $\mathcal{C}_{S'(n-1)}$, and proceed in this way until we add a role constraint $\sigma''_a \overset{s}{1} \sigma'' \in \mathcal{C}_{S(n-k+1)}$ to $\mathcal{C}_{S'(n-k+1)}$. With this construction, for each selected role constraint there is another role constraint in the next (or the previous) quasistate to satisfy **C5** (or **C6**) for π_n. Since each quasistate contains exactly one named run segment, it is easy to see

that each π_n is still a compatibility relation for $(S'(n), S'(n+1))$. This suffices to show that $(S', \mathfrak{R})$ is still a quasimodel of φ.

Regarding the number of named role constraints in each quasistate, we argue as follows. For each quasistate in S, the number of role constraints selected in (†) is bounded by $|\mathsf{csub}(\varphi)| \cdot |\mathsf{ind}(\varphi)|$. For each of those role constraints selected in (†), we add at most ℓ_φ additional role constraints (forward and backwards). So, in each quasistate, the number of role constraints of the form $\sigma_a \overset{s}{k} \sigma$ with an unnamed run segment σ is bounded by $2 \cdot |\mathsf{csub}(\varphi)| \cdot |\mathsf{ind}(\varphi)| \cdot (\ell_\varphi + 1)$. The quadratic term in $\flat_\varphi$ is for role constraints of the form $\sigma_a \overset{s}{k} \sigma_b$ (which we did not change and, so, are the same as in S). □

The proof of the following is a straightforward adaptation of the proof of Lemma 16.

Lemma 22. *Let $(S, \mathfrak{R})$ be a quasimodel for φ such that $S(n) = S(m)$ for $n < m$. Then $(S', \mathfrak{R}')$ with $S' = S^{\leq n} \cdot S^{>m}$ and*

$$\mathfrak{R}' = \{r_1^{\leq n} \cdot r_2^{>m} \mid r_1, r_2 \in \mathfrak{R},\ r_1^{[n,n+\ell_\varphi]} = r_2^{[m,m+\ell_\varphi]}\}$$

is also a quasimodel for φ.

Lemma 23. *φ has a quasimodel iff there is a sequence $S = \Xi_0 \cdot \Xi_1 \cdot \Xi_2$ of quasistates such that*

- *the length ℓ_i of each Ξ_i is bounded by $(|\varphi| \cdot \sharp_\varphi^2 + 1) \cdot (\sharp_\varphi)^{\flat_\varphi}$ (denote by $\ell = \ell_0 + \ell_1 + \ell_2$ the length of S);*
- *for all $i \in [0, \ell)$, the pair $((S(i), S(i+1))$ is compatible, with π_i being the maximal compatibility relation;*
- *$S^{\mathsf{u}}(i) = S^{\mathsf{u}}(j)$, for all $i, j \in [0, \ell)$;*
- *$\Xi_0(0) = \Xi_1(0)$ and $\Xi_1(\ell_1 - 1) = \Xi_2(\ell_2 - 1)$;*
- *$\Xi_1(n)$ satisfies* **M4***, for some $n \in [0, \ell_1)$;*
- *for all $\sigma_1 \in \mathcal{R}_{\Xi_1(\ell_1-1)}$, there is a sequence $\sigma_1(0)\sigma_2(0)\dots\sigma_k(0)$ that realizes all until-expressions in $\sigma_1(0)$ such that $(\sigma_i, \sigma_{i+1}) \in \pi_{\ell_0+\ell_1-1+i}$ for all $i \in [0, k)$.*

Proof. The proof proceeds in the same way as for Lemma 17, the only exception being that the number of possible quasistates is now bounded by $(\sharp_\varphi)^{\flat_\varphi}$ instead of $2^{\sharp_\varphi}$. Note that applying Lemma 22 for cutting out parts of the quasimodel does not affect the properties given by Lemmas 20–21. One can show regularity on the left side of the quasimodel by simpler arguments since there we do not have to satisfy any until-expressions, but only modify the quasimodel so that we can find a repeating quasistate at most $2 \cdot (\sharp_\varphi)^{\flat_\varphi}$ steps before $\Xi_1(n)$. □

As before, this immediately implies regularity.

Lemma 24. *If φ has a quasimodel, then it has a quasimodel $(S, \mathfrak{R})$ with S of the form*

$$^{\omega}(\Xi_0) \cdot \Xi_1 \cdot (\Xi_2)^{\omega},$$

where each Ξ_i is a sequence of quasistates of length at most $(|\varphi| \cdot \sharp_\varphi^2 + 1) \cdot (\sharp_\varphi)^{\flat_\varphi}$.

We now describe the algorithm for checking the existence of a sequence as in Lemma 23. The first step is to find the unnamed part that is shared by all quasistates. For this, we first compute in 2-ExpTime the set $\mathsf{rs}(\varphi)$ of run segments for φ and the set $\mathsf{rc}(\varphi)$ of role constraints over $\mathsf{rs}(\varphi)$. We then initialize $Q = (\mathcal{R}_Q, \mathcal{C}_Q)$ with $\mathcal{R}_Q = \mathsf{rs}(\varphi)$ and $\mathcal{C}_Q = \mathsf{rc}(\varphi)$. Note that this is not

a quasistate since it contains several formula run segments and several named run segments for each $a \in \mathsf{ind}(\varphi)$.

Given a concept run segment $\sigma \in \mathcal{R}_Q$, let X_σ be the set of all images of functions mapping each concept of the form $\exists s.D \in \sigma(0)$ to a role constraint $\sigma \overset{s}{k} \sigma' \in \mathcal{C}_Q$ with $D \in \sigma'(0)$. Given $x_\sigma \in X_\sigma$ and $x_{\sigma'} \in X_{\sigma'}$, we say that $(x_\sigma, x_{\sigma'})$ is *suitable* if $\sigma^{>0} = \sigma'^{<\ell_\varphi}$ and

- $\sigma \overset{s}{k} \sigma_1 \in x_\sigma$ with $k < \mathsf{iRig}(s)$ implies that there is $\sigma' \overset{s}{k+1} \sigma_2 \in x_{\sigma'}$ with $\sigma_1^{>0} = \sigma_2^{<\ell_\varphi}$; and
- $\sigma' \overset{s}{k+1} \sigma_2 \in x_{\sigma'}$ with $k > 0$ implies that there is $\sigma \overset{s}{k} \sigma_1 \in x_\sigma$ with $\sigma_1^{>0} = \sigma_2^{<\ell_\varphi}$.

We define a mapping $\pi \subseteq \mathcal{R}_Q \times \mathcal{R}_Q$, where each π is the set of pairs (σ, σ') such that, for some $x_\sigma \in X_\sigma$ and $x_{\sigma'} \in X_{\sigma'}$, the pair $(x_\sigma, x_{\sigma'})$ is suitable. We then exhaustively eliminate run segments σ from some $\mathcal{R}_Q$ if σ violates one of the following conditions:

E1 there is $\sigma' \in \mathcal{R}_Q$ such that $(\sigma, \sigma') \in \pi$;

E2 there is $\sigma' \in \mathcal{R}_Q$ such that $(\sigma', \sigma) \in \pi$;

E3 for all $C\,\mathcal{U}_I D \in \sigma(0)$ (with $\sigma(0)$ an unnamed type), there is $k > 0$ and a sequence

$$\sigma_2, \dots, \sigma_k \in \mathcal{R}_Q$$

such that $\sigma(0)\sigma_2(0)\cdots\sigma_k(0)$ realizes $C\,\mathcal{U}_I D$, $(\sigma, \sigma_2) \in \pi$, and $(\sigma_\ell, \sigma_{\ell+1}) \in \pi$, for all $1 < \ell < k$.

We assume that whenever $\sigma \in \mathcal{R}_Q$ is eliminated we remove all role constraints involving σ from $\mathcal{C}_Q$ and, for all remaining $\sigma' \in \mathcal{R}_Q$, we update $X_{\sigma'}$ accordingly. We also remove (σ_1, σ_2) from π if there is no $x_{\sigma_1} \in X_{\sigma_1}$ and $x_{\sigma_2} \in X_{\sigma_2}$ such that $(x_{\sigma_1}, x_{\sigma_2})$ is suitable.

When this process terminates, we have found the maximal unnamed part of our quasistates, and it remains to check whether we can choose a suitable formula run and named runs for each $a \in \mathsf{ind}(\varphi)$ to obtain a quasimodel. For this purpose, we consider the following graph (V_Q, E_Q). The set V_Q contains all quasistates Q' such that $\mathcal{R}_{Q'} \subseteq \mathcal{R}_Q$, $\mathcal{C}_{Q'} \subseteq \mathcal{C}_Q$, $Q'^{\mathsf{u}} = Q^{\mathsf{u}}$, and Q' contains at most $\flat_\varphi$ named role constraints, which gives us at most $(\sharp_\varphi)^{\flat_\varphi}$ many quasistates (see Lemma 21). Moreover, E_Q contains all pairs $(Q', Q'') \in V_Q \times V_Q$ that are compatible. Note that, in contrast to Q, every element $Q' \in V_Q$ has exactly one formula run segment and one named run segment for each individual name, due to the definition of quasistates.

We try to find a regular quasimodel as in Lemma 24 by a series of reachability tests in this graph. First, we enumerate all pairs of quasistates $Q_1, Q_2 \in V_Q$ such that $\varphi \in \sigma_{Q_1}(0)$ and Q_2 is reachable from Q_1. There are at most $(\sharp_\varphi)^{2\cdot\flat_\varphi}$ many possibilities to choose these two quasistates, where Q_1 represents a quasistate at some position in Ξ_1 that satisfies **M4** and Q_2 represents the last quasistate of Ξ_1 (as in Lemma 23). This reachability test is clearly possible in double exponential time.

To find the repeating sequence Ξ_2, in particular, we need to ensure that all until-formulae and until-concepts in named types are realized within Ξ_2. Hence, we consider the set U that contains all (polynomially many) until-expressions in $\sigma_{Q_2}(0)$ and $\sigma_{a,Q_2}(0)$ for all $a \in \mathsf{ind}(\varphi)$, where $\sigma_{a,Q_2} \in \mathcal{R}_{Q_2}$ is the unique named run segment for a. We enumerate all possible total orders $\psi_1 < \psi_2 < \cdots < \psi_{|U|}$ over U, of which there are exponentially many. For a fixed such order, we enumerate all possible choices of quasistates $Q_2^1, \dots, Q_2^{|U|} \in V_Q$ such that Q_2^1 is reachable from Q_2, each Q_2^{i+1} is reachable from Q_2^i, for all $i \in [1, |U|)$, and the until-expression ψ_i is satisfied on the path to Q_2^i, for all $i \in [1, |U|]$. As before, there are double exponentially many

possibilities for these quasistates (since $|U|$ is polynomial), and the reachability checks can also be done in double exponential time.

Finally, we check reachability of Q_2 from $Q_2^{|U|}$ to close the loop. Similarly, we also check for a repeating sequence Ξ_0 of quasistates for the left part of our quasimodel from which we can reach Q_1 (chosen before, where $\varphi \in \sigma_{Q_1}(0)$). If all of the described checks succeed for one of the double exponentially many possibilities we enumerate, then our algorithm returns 'satisfiable', and otherwise 'unsatisfiable'.

Lemma 25. *The algorithm returns 'satisfiable' iff φ is satisfiable w.r.t. interval-rigid names.*

Proof. By Lemmas 19 and 23, it suffices to check the existence of a sequence of quasistates as in Lemma 23.

($\Rightarrow$) It is straightforward to check that any sequence resulting from a successful series of the checks described above satisfies the conditions of Lemma 23. In particular, any two consecutive quasistates are compatible by Conditions **E1** and **E2** and the definition of E_Q. Moreover, the required sequences of run segments that satisfy the until-expressions exist by Condition **E3** and the reachability checks in (V_Q, E_Q).

($\Leftarrow$) Let $S = \Xi_0 \cdot \Xi_1 \cdot \Xi_2$ be a sequence as in Lemma 23. Then the shared unnamed part of all the $S(i)$ must be included in Q as constructed by the algorithm. The remaining conditions of Lemma 23 ensure that all our reachability checks succeed, albeit with a possibly larger unnamed part in the quasistates. In particular, the order on U is the same as the one in which these until-expressions are satisfied in Ξ_2. □

We provide the proof of the corresponding lower bound after the proof of Theorem 8 below, since both use the same constructions.

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.*

D Proofs for Section 4

Theorem 7. *Satisfiability in $\mathcal{ALCH}$-LTL w.r.t. interval-rigid names is Σ^1_1-hard.*

Proof. $\mathcal{ALCH}$ denotes the extension of $\mathcal{ALC}$ by the most basic role inclusions of the form $r_1 \sqsubseteq r_2$, where $r_1, r_2 \in \mathsf{N_R}$. Using such axioms, we can simulate the rigid role r in the formula ϕ_T from Section 4.1 by three 2-rigid roles r^0, r^1, r^2, together with the following formulae, for $0 \leq i \leq 2$:

$$\Box\Big(A^i(a) \to \big((r^{i-1} \sqsubseteq r^i) \wedge (\top \sqsubseteq \neg\exists r^{i+1}.\top)\big)\Big).$$

We also have to replace $(\top \sqsubseteq \exists r.\top)$ in ϕ_T by $(\top \sqsubseteq \exists r_0.\top)$, $\forall r.A^i$ by $\forall r^i.A^i$, and $\forall r.P_{t'}$ by $\forall r^0.P_{t'} \sqcap \forall r^1.P_{t'} \sqcap \forall r^2.P_{t'}$. □

Theorem 8. *Satisfiability in $\mathcal{ALC}$-LTL$_{|gGCI}$ with respect to interval-rigid names is* 2-ExpTime*-hard.*

Proof. We consider the reduction used in [8, Lemma 4.2], and observe that the rigid concept and role names used in that reduction need to stay rigid only for 2^k time points, where k is polynomial in the size of the original problem. Since 2^k can be written using polynomially many bits, we can instead designate these names to be 2^k-rigid. However, we also need to do this for

all *negations* of the rigid concept names A, to ensure that, if A is implied to hold at d at some point in the interval $[0, 2^k - 1]$, this information is also propagated backwards in time, i.e., d satisfies A also at time point 0. This is not needed for the single rigid role r used in the proof, since all necessary r-connections are already implied at time point 0. □

We can adapt this result to show the lower bound for Theorem 5, i.e., for $\text{LTL}^{\mathsf{bin}}_{\mathcal{ALC}}$ w.r.t. interval-rigid names and global GCIs and the temporal semantics based on $\mathbb{Z}$. The only problem we face there is that the 2^k-rigid names described above may not be rigid for the interval $[0, 2^k - 1]$, but for any interval of this length that includes 0. However, since we are only interested in the domain elements reachable by a certain role r at time point 0 from an individual name a, and all relevant r-connections are present at time point 0, we can use them to propagate a fresh concept name B to all relevant domain elements, and use the global GCI $\bigcirc B \sqsubseteq \neg\exists r.\top$ to enforce that these r-connections do not exist at time point -1, and hence must exist in the whole interval $[0, 2^k - 1]$ since r is 2^k-rigid. Similarly, we can express that the value of a 2^k-rigid concept name A (where $\neg A$ is also 2^k-rigid) must stay constant in $[0, 2^k - 1]$ via the global GCIs $\bigcirc B \sqcap A \sqsubseteq \neg A$ and $\bigcirc B \sqcap \neg A \sqsubseteq A$.

E Proofs for Section 5

Theorem 11. *Satisfiability in $\mathcal{ALC}$-$\text{LTL}^{\mathsf{bin}}$ is in* 2-ExpTime *w.r.t. rigid names, and in* ExpSpace *w.r.t. rigid concepts.*

Proof. By Lemma 10, to check satisfiability of ϕ, we need to do the following:

(1) Find a set $\mathcal{W} \subseteq 2^{\{p_1,\dots,p_m\}}$.

(2) Check satisfiability of $\chi_{\mathcal{W}}$.

(3) Check satisfiability of $\phi^{\text{p}}_{\mathcal{W}} = \phi^{\text{p}} \wedge \phi_{\mathcal{W}}$.

Depending on the targeted complexity class, step (1) can be handled in different ways (see [8] for details):

- If there are no rigid names, it suffices to consider the *maximal* set $\mathcal{W}$ containing all $W \subseteq \{p_1, \dots, p_m\}$ for which $\bigwedge_{p_j \in W} \alpha_j \wedge \bigwedge_{p_j \in \overline{W}} \neg\alpha_j$ is satisfiable. This works because there are no rigid names that could enforce dependencies between different worlds.
- Otherwise, one can guess a set $\mathcal{W}$ in (non-deterministic) exponential time.
- In general, one can also enumerate all sets $\mathcal{W}$ in (deterministic) double exponential time.

Moreover, step (2) is exactly the same as in [8], which means that we can use the complexity upper bounds from that paper for steps (1) and (2). In particular, they are possible in 2-ExpTime in general, in NExpTime if we only allow rigid concepts, and in ExpTime if we have no rigid names, or we have rigid concepts and global GCIs. It only remains to determine the complexity of step (3), and take the union of the obtained complexity classes.

For this, we translate ϕ^{p} into an exponentially larger LTL formula $\phi^{\text{p}\prime}$. Since the exponentially large $\phi_{\mathcal{W}}$ is already an LTL formula, the satisfiability of the conjunction $\phi^{\text{p}\prime} \wedge \phi_{\mathcal{W}}$ can be checked in exponential space. This yields the claimed results since ExpTime $\subseteq$ NExpTime $\subseteq$ (N)ExpSpace $\subseteq$ 2-ExpTime. □