

Temporal Description Logics: A Survey

Carsten Lutz
Inst. of Theoretical Computer Science
TU Dresden, Germany
lutz@tcs.inf.tu-dresden.de

Frank Wolter
Department of Computer Science
University of Liverpool, U.K.
wolter@liverpool.ac.uk

Michael Zakharyashev
School of Computer Science and Information Systems
Birkbeck College London, U.K.
michael@dcs.bbk.ac.uk

Abstract

We survey temporal description logics that are based on standard temporal logics such as LTL and CTL. In particular, we concentrate on the computational complexity of the satisfiability problem and algorithms for deciding it.

1. Introduction

Description Logics (DLs) are a well-known family of logic-based knowledge representation formalisms with a number of relevant applications. For example, DLs enjoy significant popularity as ontology languages, which has resulted in the standardization of a description logic as part of W3C’s ontology language OWL. Other relevant applications of DLs include the representation of and reasoning about conceptual database models. Notably, extended entity-relation (EER) models and UML class diagrams can be embedded into DLs, and DL reasoners can be used to verify their consistency and to derive implicit consequences of the model [16].

In many applications of DLs, temporal aspects play an important role. For example, the description of a concept in an ontology often involves reference to temporal patterns—consider, e.g., the definition of Malaria in a bio-medical ontology. Likewise, time plays a crucial role when using DLs to represent conceptual models of *temporal* databases. These observations have resulted in a rather diverse literature on *temporal description logic* (TDLs). Proposals for TDLs include combinations of DLs with Halpern and Shoham’s logic of time intervals [28, 40], formalisms inspired by action logics [2], and the treatment of time points and intervals as a datatype [35]. For more information, see the previous surveys [3, 4].

In contrast to the existing surveys, we focus on combinations of standard DLs such as \mathcal{ALC} with standard temporal logics (TLs) such as LTL (linear time temporal logic) and CTL (computational tree logic). Such combinations are based on a two-dimensional semantics, where one dimension is for time and the other for the DL domain. They were first proposed by Schild [38] in 1993, and since then have experienced constant development. Notably, TDLs of this kind are well-suited for capturing the temporal aspects of concepts in ontologies, and for encoding temporal EER models [5].

We mainly discuss complexity results and algorithms for satisfiability checking, which is the most important reasoning problem for DLs and TDLs. In particular, satisfiability can be used to decide consistency of (encoded) temporal EER models. For all TDLs considered in this paper, satisfiability is inter-reducible with subsumption in polynomial time. Thus, the covered results also apply to the latter problem, which is highly relevant for reasoning about ontologies.

When constructing a two-dimensional TDL, a number of design decisions have to be made. For example, one has to choose a concrete DL and a TL to be combined. In this paper, we mainly use \mathcal{ALC} for the DL part and LTL for the TL part. Another important decision is whether to apply the TL operators to DL concepts, roles, TBoxes, or ABoxes. The structure of this survey roughly reflects the available choices regarding this second issue. After some preliminaries in Section 2, in Section 3 we analyze the application of TL operators to concepts and roles. In Sections 4 and 5, we consider the application of these operators to TBoxes and ABoxes, respectively. Section 6 is devoted to TDLs whose DL component is the lightweight description logic DL-Lite, and Section 7 is concerned with TDLs whose TL component is CTL. Throughout the paper, we also discuss

whether or not the choice of a different DL has an impact on the presented results.

2. Preliminaries

We introduce the basics of DLs and define a temporal semantics for the TDLs considered in this survey.

2.1 Description Logic

Traditionally, DLs focus on representing the terminological knowledge of an application domain. The central ingredient to such a representation are *concepts*, which are built from a countably infinite set N_C of *concept names* and a countably infinite set N_R of *role names*, by applying the available concept constructors. In the basic propositionally closed description logic \mathcal{ALC} , these constructors are

$$A, \top, \neg C, C \sqcap D, \exists r.C,$$

where A ranges over N_C , C and D range over concepts, and r ranges over N_R . As usual, we use \perp as an abbreviation for $\neg\top$, $C \sqcup D$ for $\neg(\neg C \sqcap \neg D)$, and $\forall r.C$ for $\neg\exists r.\neg C$.

The main use of concepts is in *TBoxes*, which define and interrelate terminological notions. Formally, TBoxes are finite sets of *concept inclusions* (CIs) $C \sqsubseteq D$. Complementing TBoxes, which represent knowledge at the conceptual level, *ABoxes* represent knowledge at the instance level. Formally, an ABox is a finite set of *assertions* $C(a)$ and $r(a, b)$, where a and b are individuals from a countably infinite set N_I of *individual names*. The semantics of DLs is based on the notion of an *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a nonempty *domain* and $\cdot^{\mathcal{I}}$ is a function that maps every concept name $A \in N_C$ to a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, every role name $r \in N_R$ to a relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and every individual name $a \in N_I$ to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. The interpretation $\cdot^{\mathcal{I}}$ can be lifted to composite concepts as follows:

$$\begin{aligned} \top^{\mathcal{I}} &= \Delta^{\mathcal{I}}, \\ (\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}, \\ (C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}}, \\ (\exists r.C)^{\mathcal{I}} &= \{d \in \Delta^{\mathcal{I}} \mid \exists d' \in \Delta^{\mathcal{I}} (d, d') \in r^{\mathcal{I}} \}. \end{aligned}$$

An interpretation \mathcal{I} is a *model* of a concept C if $C^{\mathcal{I}} \neq \emptyset$; it is a *model* of a TBox \mathcal{T} if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all $C \sqsubseteq D$ in \mathcal{T} , and a *model* of an ABox \mathcal{A} if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ for all $C(a) \in \mathcal{A}$ and $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$ for all $r(a, b) \in \mathcal{A}$. A concept C is *satisfiable* w.r.t. a TBox \mathcal{T} if there is a common model of C and \mathcal{T} .

2.2 Temporal Interpretations

There are many ways of using standard temporal logic operators such as \circ (at the next moment), \diamond (eventually),

\square (always in the future), and \mathcal{U} (until) in order to add a temporal dimension to DLs. For example, we can introduce these operators as extra concept constructors that allow us to describe the temporal behaviour of concept membership. Alternatively or additionally, the temporal operators can be applied to roles, TBoxes, and ABoxes. The resulting TDLs differ in many aspects, in particular modelling capabilities and computational properties. What unifies all of them, however, is a common semantics.

A *temporal interpretation* $\mathfrak{J} = (\Delta^{\mathfrak{J}}, \cdot^{\mathfrak{J}})$ consists of a nonempty domain $\Delta^{\mathfrak{J}}$ and a function $\cdot^{\mathfrak{J}}$ that maps every concept name $A \in N_C$ to a subset $A^{\mathfrak{J}} \subseteq \mathbb{N} \times \Delta^{\mathfrak{J}}$, every role name $r \in N_R$ to a subset $r^{\mathfrak{J}} \subseteq \mathbb{N} \times \Delta^{\mathfrak{J}} \times \Delta^{\mathfrak{J}}$, and every individual name $a \in N_I$ to an element $a^{\mathfrak{J}} \in \Delta^{\mathfrak{J}}$. Intuitively, the elements of \mathbb{N} represent time points, ordered by $<$. Thus, $(n, d) \in A^{\mathfrak{J}}$ means that, in the interpretation \mathfrak{J} , d is an instance of A at time point n , and similarly for role names. Individual names do not have the temporal index because they are assumed to be *rigid*, i.e., interpreted in the same way at every point of time. Temporal interpretations as introduced above are obviously a special case of temporal first-order structures (without function symbols and equality, and with at most binary predicates); see [24].

Equivalently, a temporal interpretation \mathfrak{J} can be defined as an infinite sequence $\mathfrak{J}(0), \mathfrak{J}(1), \dots$ of (non-temporal) interpretations that share the same domain $\Delta^{\mathfrak{J}}$ and interpret individual names in the same way. We will use this alternative definition whenever more suitable. The restriction that all $\mathfrak{J}(i)$ share the same domain is known as the *constant domain assumption*, i.e., objects are never destroyed or created over time. Using the alternative definition, it is also easy to define temporal interpretations with expanding domains ($\Delta^{\mathfrak{J}(0)} \subseteq \Delta^{\mathfrak{J}(1)} \subseteq \dots$), decreasing domains ($\Delta^{\mathfrak{J}(0)} \supseteq \Delta^{\mathfrak{J}(1)} \supseteq \dots$), and varying domains (no restrictions imposed).

The temporal interpretations defined above assume discrete linear time that is bounded to the past and unbounded to the future. Of course, there are many other models of time: to obtain Dedekind-complete time, one can replace \mathbb{N} with the real numbers; to give up infinite future, one can use finite (but unbounded) initial parts of \mathbb{N} , etc. To keep things simple, however, we mainly use $(\mathbb{N}, <)$ as our model of time and only deviate from it when necessary.

3. Temporal Concepts

In this section, we discuss TDLs that are suited for reasoning about the temporal evolution of concepts.

3.1 Introducing $LTL_{\mathcal{ALC}}$

We define the temporal description logic $LTL_{\mathcal{ALC}}$ whose concepts are formed using the concept constructors of

\mathcal{ALC} enriched with the temporal constructors $\circ C$ and CUD , as known from propositional *linear time temporal logic* (LTL); see, e.g., [23]. As usual, we use $\diamond C$ as an abbreviation for $\top UC$ and $\square C$ for $\neg \diamond \neg C$. TBoxes are defined in the same way as in the case of \mathcal{ALC} , but now using $LTL_{\mathcal{ALC}}$ concepts instead of \mathcal{ALC} concepts. For simplicity, we work without ABoxes for now. For example, the following CI states that any non-EU country has to be first an EU member candidate before it can become an EU member:

$$\neg \text{EU_member} \sqcap \diamond \text{EU_member} \sqsubseteq \\ \diamond (\text{EU_candidate} \mathcal{U} \text{EU_member}).$$

Like in propositional LTL, we can further enrich $LTL_{\mathcal{ALC}}$ by adding additional temporal operators, e.g., past operators for ‘at the previous moment’ and ‘since’ [46, 5] or fixed-point operators [22]. For the purposes of this paper, we will concentrate on \circ and \mathcal{U} , and only mention additional temporal operators when necessary.

The semantics of $LTL_{\mathcal{ALC}}$ is based on temporal interpretations. To extend $\cdot^{\mathcal{J}}$ to composite concepts, we use the same clauses as in Section 2.1 for the Booleans, plus the following ones:

$$\begin{aligned} (\exists r.C)^{\mathcal{J}} &= \{(n, d) \mid \exists (n, d') \in C^{\mathcal{J}} (n, d, d') \in r^{\mathcal{J}}\}, \\ (\circ C)^{\mathcal{J}} &= \{(n, d) \mid (n+1, d) \in C^{\mathcal{J}}\}, \\ (CUD)^{\mathcal{J}} &= \{(n, d) \mid \exists m \geq n ((m, d) \in D^{\mathcal{J}} \wedge \\ &\quad (k, d) \in C^{\mathcal{J}} \text{ for } n \leq k < m)\}. \end{aligned}$$

A temporal interpretation \mathcal{J} is a *model* of a concept C if C is satisfied at time point 0, i.e., $(d, 0) \in C^{\mathcal{J}}$ for some $d \in \Delta^{\mathcal{J}}$. It is a *model* of a TBox \mathcal{T} iff $C^{\mathcal{J}} \subseteq D^{\mathcal{J}}$ for all $C \sqsubseteq D$ in \mathcal{T} . Thus, the CIs of a TBox are regarded as temporally *global* constraints in the sense that they should hold at every moment of time. An $LTL_{\mathcal{ALC}}$ concept C is *satisfiable* w.r.t. a TBox \mathcal{T} if there is a common model of C and \mathcal{T} . It is not hard to see that this is equivalent to saying that there is a model \mathcal{J} of \mathcal{T} with $C^{\mathcal{J}} \neq \emptyset$. The problem of *concept satisfiability w.r.t. TBoxes* in $LTL_{\mathcal{ALC}}$ is to decide, given a concept C and a TBox \mathcal{T} , whether C is satisfiable w.r.t. \mathcal{T} .

Intuitively, $LTL_{\mathcal{ALC}}$ can be viewed as a two-dimensional combination of \mathcal{ALC} with LTL [24]. If we treat concept names as propositional variables of LTL, then every temporal interpretation \mathcal{J} and every element $d \in \Delta^{\mathcal{J}}$ give rise to the LTL structure \mathfrak{M}_d defined by

$$\mathfrak{M}_d(n) = \{A \in \mathbf{N}_C \mid (n, d) \in A^{\mathcal{J}}\}, \quad \text{for all } n \geq 0.$$

In this sense, the roles in $LTL_{\mathcal{ALC}}$ are a second dimension: they allow us to switch between different LTL structures \mathfrak{M}_d and $\mathfrak{M}_{d'}$ without changing the current time point. The different expressive and computational properties of $LTL_{\mathcal{ALC}}$ and other TDLs mainly depend on the degree of interaction between the temporal component and the DL

component. One effect of this interaction in $LTL_{\mathcal{ALC}}$ is the loss of the *finite model property* (FMP). In pure \mathcal{ALC} , satisfiability of a concept C w.r.t. a TBox \mathcal{T} implies satisfiability in a model with a finite domain. In contrast, the following $LTL_{\mathcal{ALC}}$ TBox \mathcal{T} is such that the concept A is satisfiable w.r.t. \mathcal{T} , but in every model \mathcal{J} of A and \mathcal{T} , the domain $\Delta^{\mathcal{J}}$ must be infinite:

$$A \sqsubseteq \circ \square (\neg A \sqcap \exists r.A).$$

In spite of this effect, the interaction between LTL and \mathcal{ALC} in $LTL_{\mathcal{ALC}}$ is actually rather weak, as illustrated in the next section.

3.2 $LTL_{\mathcal{ALC}}$ with Expanding Domains

In this section and the subsequent one, we analyze the complexity of concept satisfiability in $LTL_{\mathcal{ALC}}$. We start by considering models with expanding domains because this case is particularly simple. Note first that constant domains and expanding domains give rise to different versions of concept satisfiability. For example, the following TBox has a (temporal) model with expanding domains, but no model with constant domains:

$$\top \sqsubseteq \circ (A \sqcap \exists r.\neg A). \quad (*)$$

A key observation regarding $LTL_{\mathcal{ALC}}$ with expanding domains is that this logic is closely connected to the *fusion* of LTL and \mathcal{ALC} , a general combination method for modal logics studied, e.g., in [11, 23]. To make this connection explicit, we introduce an alternative interpretation for $LTL_{\mathcal{ALC}}$.

Let $\text{succ} \notin \mathbf{N}_R$ be a special role name that is not allowed in $LTL_{\mathcal{ALC}}$ concepts. A (non-temporal) interpretation \mathcal{I} is called a *fusion interpretation* iff $\text{succ}^{\mathcal{I}}$ is a total function. For $d, e \in \Delta^{\mathcal{I}}$, we write $d \rightsquigarrow e$ to indicate that e is reachable from d by applying $\text{succ}^{\mathcal{I}}$ to d zero or more times. To interpret an $LTL_{\mathcal{ALC}}$ concept in \mathcal{I} , we use the clauses from Section 2.1 together with the following ones:

$$\begin{aligned} (\circ C)^{\mathcal{I}} &= \{d \mid \text{succ}^{\mathcal{I}}(d) \in C^{\mathcal{I}}\}, \\ (CUD)^{\mathcal{I}} &= \{d \mid \exists e \in D^{\mathcal{I}} (d \rightsquigarrow e \wedge \forall e' \neq e \\ &\quad (d \rightsquigarrow e' \rightsquigarrow e \Rightarrow e' \in C^{\mathcal{I}}))\}. \end{aligned}$$

The following lemma due to Schild [38] shows that temporal interpretations with expanding domains and fusion interpretations are equivalent for $LTL_{\mathcal{ALC}}$ concepts and TBoxes.

Lemma 1. *Let C be an $LTL_{\mathcal{ALC}}$ concept and \mathcal{T} a TBox. Then there is a temporal model of C and \mathcal{T} with expanding domains iff there is a fusion model of C and \mathcal{T} .*

We give a proof sketch. For the ‘if’ direction, let \mathcal{I} be a fusion model of C and \mathcal{T} . W.l.o.g. we may assume that \mathcal{I} is tree-shaped, i.e., the directed graph $G_{\mathcal{I}} = (V, E)$ with $V = \Delta^{\mathcal{I}}$ and $E = \text{succ}^{\mathcal{I}} \cup \bigcup_{r \in \mathbb{N}_{\mathbb{R}}} r^{\mathcal{I}}$ is a tree. Let $d_r \in \Delta^{\mathcal{I}}$ be the root of the tree. We can rename the elements of $\Delta^{\mathcal{I}}$ so that

$$\begin{aligned} \Delta^{\mathcal{I}} &\subseteq \mathbb{N} \times \Delta, \text{ for some set } \Delta, \\ d_r &\text{ is renamed to an element of } \{0\} \times \Delta, \\ r^{\mathcal{I}} &\subseteq \{(n, d), (n, e) \mid n \geq 0 \wedge d, e \in \Delta\}, \\ \text{succ}^{\mathcal{I}} &\subseteq \{(n, d), (n+1, d) \mid n \geq 0 \wedge d \in \Delta\}, \end{aligned}$$

for all $r \in \mathbb{N}_{\mathbb{R}}$. It is now straightforward to convert \mathcal{I} into a temporal interpretation $\mathfrak{J} = \mathfrak{J}(0), \mathfrak{J}(1), \dots$ with expanding domains that is a model of C and \mathcal{T} . For the ‘only if’ direction, it is straightforward to convert a temporal model $\mathfrak{J} = \mathfrak{J}(0), \mathfrak{J}(1), \dots$ for C and \mathcal{T} into a fusion interpretation \mathcal{I} with $\Delta^{\mathcal{I}} \subseteq \{(n, d) \mid d \in \Delta^{\mathfrak{J}(n)}\}$.

Lemma 1 shows that concept satisfiability in $LTL_{\mathcal{ALC}}$ w.r.t. TBoxes and with expanding domains can be reduced to concept satisfiability w.r.t. TBoxes in any extension of \mathcal{ALC} that provides for functional roles and allows us to express CUD under the fusion semantics given above (note that $\circ C$ can be expressed as $\forall \text{succ}.C$ under this semantics). An example of such an extension is \mathcal{ALC} with functional roles and the role operators known from propositional dynamic logic (PDL): composition, union, reflexive-transitive closure, and test [20, 8]. As shown in [38, 21], satisfiability w.r.t. TBoxes is EXPTIME-complete in this DL, and PSPACE-complete if transitive closure can only be applied to functional roles and no TBoxes are admitted. These upper bounds thus transfer to $LTL_{\mathcal{ALC}}$ with expanding domains, and corresponding lower bounds carry over from \mathcal{ALC} .

Theorem 2 ([38]). *Concept satisfiability in $LTL_{\mathcal{ALC}}$ with expanding domains is PSPACE-complete without TBoxes and EXPTIME-complete with TBoxes.*

Recall that satisfiability in LTL is PSPACE-complete [41] and satisfiability in \mathcal{ALC} is PSPACE-complete without TBoxes [39] and EXPTIME-complete with TBoxes [37]. Thus, $LTL_{\mathcal{ALC}}$ is computationally rather well-behaved: concept satisfiability is not harder than in the component logics. Additionally, the characterization in terms of fusion models opens a way to practical reasoning systems that are not too different from standard \mathcal{ALC} reasoners.

3.3 $LTL_{\mathcal{ALC}}$ with Constant Domains

In [38], Lemma 1 is stated for the constant domain case (modulo some neglectable differences in the model of time). Alas, this was not correct. For example, the TBox (*) above has a fusion model, but no temporal model with constant domains. If \mathbb{Z} is used instead of \mathbb{N} as the set of time points

in temporal interpretations, then an analogue of Lemma 1 for constant domains can be obtained by requiring that fusion models interpret succ as a total and *surjective* function. Satisfiability in $LTL_{\mathcal{ALC}}$ can then be decided by reduction to the extension of \mathcal{ALC} described in the previous section and additionally enriched with inverse roles. In our case, where \mathbb{N} is used as the set of time points, such a simple fix does not seem to work. In the following, we give a dedicated algorithm for deciding satisfiability in $LTL_{\mathcal{ALC}}$ with constant domains.

Our algorithm performs type elimination, as first used by Pratt in the context of PDL [30]. Let C be an $LTL_{\mathcal{ALC}}$ concept and \mathcal{T} a TBox. W.l.o.g. we can assume that \mathcal{T} is of the form $\{\top \sqsubseteq C_{\mathcal{T}}\}$. Let $\text{sub}(C, \mathcal{T})$ be the set of all subconcepts of C and $C_{\mathcal{T}}$, and $\text{cl}(C, \mathcal{T})$ the closure under single negations of the set

$$\text{sub}(C, \mathcal{T}) \cup \{\circ(CUD) \mid CUD \in \text{sub}(C, \mathcal{T})\}.$$

A *type* for C and \mathcal{T} is a subset $t \subseteq \text{cl}(C, \mathcal{T})$ such that the following conditions are satisfied:

- $A \in t$ iff $\neg A \notin t$, for all $A \in \text{cl}(C, \mathcal{T})$,
- $C \sqcap D \in t$ iff $\{C, D\} \subseteq t$, for all $C \sqcap D \in \text{cl}(C, \mathcal{T})$,
- $CUD \in t$ iff $D \in t$ or $\{C, \circ(CUD)\} \subseteq t$, for all $CUD \in \text{cl}(C, \mathcal{T})$,
- $C_{\mathcal{T}} \in t$.

Let $\Pi(C, \mathcal{T})$ denote the set of all types for C and \mathcal{T} . For $t, t' \in \Pi(C, \mathcal{T})$, we say that t and t' are *compatible* if $\circ C \in t$ implies $C \in t'$, for all $\circ C \in \text{cl}(C, \mathcal{T})$. A *temporal type* for C and \mathcal{T} has the form (t, i) , with $t \in \Pi(C, \mathcal{T})$ and $i \in \mathbb{N}$. We say that (t, i) is *realizable w.r.t. \mathcal{T}* if there is a temporal model \mathfrak{J} of \mathcal{T} and a $d \in \Delta^{\mathfrak{J}}$ such that for all $D \in \text{cl}(C, \mathcal{T})$, we have $(i, d) \in D^{\mathfrak{J}}$ iff $D \in t$. Clearly, C is satisfiable w.r.t. \mathcal{T} if there is a type $t \in \Pi(C, \mathcal{T})$ such that $C \in t$ and $(t, 0)$ is realizable w.r.t. \mathcal{T} .

The following algorithm generates all temporal types that are realizable w.r.t. \mathcal{T} . For every $n \geq 0$, let $\delta(n) = \min\{n, |\Pi(C, \mathcal{T})|\}$. The algorithm starts with the set of temporal types

$$M_0 := \Pi(C, \mathcal{T}) \times \{0, \dots, |\Pi(C, \mathcal{T})|\}$$

and then generates a sequence of sets $M_0 \supseteq M_1 \supseteq M_2 \dots$, where M_{j+1} is obtained from M_j by eliminating all temporal types (t, i) that violate one of the following conditions:

1. for all $\exists r.D \in t$, there is a temporal type $(t', i) \in M_j$ such that $\{D\} \cup \{\neg E \mid \neg \exists r.E \in t\} \subseteq t'$;
2. there is a temporal type $(t', \delta(i+1)) \in M_j$ such that t and t' are compatible;
3. if $i > 0$, there is a temporal type $(t', i-1) \in M_j$ such that t' and t are compatible;

4. for all $CUD \in t$, there is a sequence of temporal types

$$(t_1, i_1), (t_2, i_2), \dots, (t_k, i_k) \in M_j$$

such that (i) $(t_1, i_1) = (t, i)$, (ii) $i_{\ell+1} = \delta(i_\ell + 1)$ for $1 \leq \ell \leq k$, (iii) $D \in t_k$, (iv) t_ℓ and $t_{\ell+1}$ are compatible and $C \in t_\ell$ for $1 \leq \ell < k$.

The algorithm stops when $M_j = M_{j+1}$. In this case, we call M_j the *final set* computed by the algorithm. Intuitively, the algorithm repeatedly eliminates temporal types that are not realizable. Thus, we return ‘satisfiable’ iff the final set contains a temporal type $(t, 0)$ with $C \in t$, and ‘unsatisfiable’ otherwise. The correctness of the algorithm is proved in the appendix. It follows from the proof that a temporal type (t, i) with $i \geq |\Pi(C, \mathcal{T})|$ is realizable iff the temporal type $(t, |\Pi(C, \mathcal{T})|)$ is realizable. This explains the use of the function δ . It is not hard to see that the algorithm runs in exponential time. Thus, we obtain the following result.

Theorem 3. *Concept satisfiability in $LTL_{\mathcal{ALCC}}$ w.r.t. TBoxes and with constant domains is EXPTIME-complete.*

Satisfiability with expanding, decreasing, and varying domains can be polynomially reduced to satisfiability with constant domains. Thus, the result from Theorem 2 applies to those cases as well. Here, we exemplarily consider the case of decreasing domains. Let C be a concept and \mathcal{T} a TBox. W.l.o.g. we may assume that \mathcal{T} is of the form $\{\top \sqsubseteq C_{\mathcal{T}}\}$. To reduce satisfiability with decreasing domains to satisfiability with constant domains, we introduce a fresh concept name E that denotes the *existence* of a domain element in an interpretation. Let C^* be obtained from C (and $C_{\mathcal{T}}^*$ from $C_{\mathcal{T}}$) by replacing every subconcept $\exists r.D$ with $\exists r.(E \sqcap D)$, every subconcept $\circ D$ with $\circ(D \sqcap E)$, and every subconcept DUD' with $(D \sqcap E)\mathcal{U}(D' \sqcap E)$. Then, C and \mathcal{T} have a model with decreasing domains iff $C^* \sqcap E$ and $\{E \sqsubseteq C_{\mathcal{T}}^*, \circ E \sqsubseteq E\}$ have a model with constant domains. Such reductions work in all TDLs considered in this paper. So from now on we concentrate on constant domains.

3.4 Rigid Roles

In DL, additional constraints for roles have become an important means to increase the expressive power. A typical example is the introduction of transitive roles or role inclusions. In TDL, the most important constraint of this type is *rigid roles*. A role r has a *rigid interpretation* in \mathcal{I} if

$$(n, d) \in r^{\mathcal{I}} \quad \text{iff} \quad (m, d) \in r^{\mathcal{I}}, \quad \text{for } n, m \in \mathbb{N}, d \in \Delta^{\mathcal{I}}.$$

The role `geographical_part_of` is a typical rigid role, whereas `member_of` will mostly be non-rigid. We remark that, in contrast to roles, no additional constraints are required to express that a *concept* is rigid: the CIs $C \sqsubseteq \Box C$ and $\neg C \sqsubseteq \Box \neg C$ imply that $(n, d) \in C^{\mathcal{I}}$ iff $(m, d) \in C^{\mathcal{I}}$, for all $n, m \in \mathbb{N}$.

Theorem 4. *Concept satisfiability in $LTL_{\mathcal{ALCC}}$ w.r.t. TBoxes and with a single rigid role is Σ_1^1 -hard, thus undecidable and not even recursively enumerable.*

This result can be proved by reduction of a well-known Σ_1^1 -complete problem, namely the *recurrent tiling problem* [29]: given a set $S = \{t_0, \dots, t_n\}$ of tile types, decide whether it is possible to cover the $\mathbb{N} \times \mathbb{N}$ -grid with tiles of these types such that t_0 appears infinitely often in the first row. Recall that each t_i is a 4-tuple of colours $\langle \text{left}(t_i), \text{right}(t_i), \text{up}(t_i), \text{down}(t_i) \rangle$, and a tile of type t_j can be a right (up) neighbour of a t_i -tile only if $\text{right}(t_i) = \text{left}(t_j)$ ($\text{up}(t_i) = \text{down}(t_j)$). We encode this problem in $LTL_{\mathcal{ALCC}}$ with a single rigid role r . Let A_0, \dots, A_n be concept names representing the given tile types, and let the TBox \mathcal{T} contain the following CIs:

1. $\top \sqsubseteq \exists r.\top$, $\top \sqsubseteq \bigsqcup_{i \leq n} A_i$, $A_i \sqcap A_j \sqsubseteq \perp$, for $i \neq j$,
2. $A_i \sqsubseteq \forall r. \bigsqcup_{\text{up}(t_i) = \text{down}(t_j)} A_j$, $i \leq n$,
3. $A_i \sqsubseteq \circ \bigsqcup_{\text{right}(t_i) = \text{left}(t_j)} A_j$, $i \leq n$.

Then the concept $\Box \diamond A_0$ is satisfiable w.r.t. \mathcal{T} iff S can tile the $\mathbb{N} \times \mathbb{N}$ -grid with tiles of type t_0 appearing infinitely often in the first row. We sketch the proof of the ‘only if’ direction. Let $(0, d_0)$ be an instance of $\Box \diamond A_0$ in a model \mathcal{I} of \mathcal{T} . By the first CI in \mathcal{T} , there is an infinite sequence $(0, d_0)r^{\mathcal{I}}(0, d_1)r^{\mathcal{I}}(0, d_2)r^{\mathcal{I}} \dots$ at time point 0. We can use it as the first column of the grid. Since r is a rigid role, this sequence is found also at any other time point, which gives us the rest of the grid. By the second and third CIs, every point of the grid is covered by a unique tile, and the last two CIs guarantee that the colours of the tiles match.

In the reduction above, one dimension of the grid is implicit in the flow of time. The second dimension is generated by the infinite chain enforced by the first CI in \mathcal{T} . Without TBoxes or with only acyclic ones (see [10] for a definition), such a chain cannot be enforced. In these cases, concept satisfiability becomes decidable, though it is still very expensive.

Theorem 5 ([24]). *In $LTL_{\mathcal{ALCC}}$ with rigid roles, concept satisfiability w.r.t. acyclic TBoxes is decidable. The problem is hard for non-elementary time already for concepts containing only one rigid role and no non-rigid roles.*

The intuition behind the proof of this result is as follows. Define the *role depth* $\text{rd}(C)$ of an $LTL_{\mathcal{ALCC}}$ concepts C as the number of nestings of existential restrictions $\exists r$ in C . For example, $\text{rd}(A \sqcap \exists r.B) = 1$ and $\text{rd}(\exists r_1.(B \sqcap \exists r_2.A)) = 2$. In the same way as for \mathcal{ALC} , one can show that any satisfiable $LTL_{\mathcal{ALCC}}$ concept C is satisfiable in a temporal interpretation \mathcal{I} in which no

path $(n, d_0)r^{\exists}(n, d_1)r^{\exists} \dots r^{\exists}(n, d_k)$ has length larger than $\text{rd}(C)$. Thus, it is not possible to generate the infinite second dimension of an $\mathbb{N} \times \mathbb{N}$ -grid. However, the complexity of deciding satisfiability of a concept C still grows super-exponentially with $\text{rd}(C)$, and in [24] it is shown that the satisfiability problem for concepts C with $\text{rd}(C) \leq n$ is $n\text{EXPSPACE}$ -complete, for all $n \geq 1$. The lower bound is proved using the yardstick technique of Stockmeyer [42] and the upper bound uses, in addition to the bound on the length of paths, the fact that each $\mathcal{J}(n)$ can be assumed to be tree-shaped.

We will return to the issue of rigid roles in Section 5.

3.5 Temporal Roles

Although already rigid roles are computationally difficult, there are cases where more expressive power is needed for talking about the temporal evolution of roles. In particular, it is natural to apply temporal operators not only to concepts, but also to roles. For example, to define roles ‘will always be a member of’ and ‘will be a member of over and over again,’ one can use the roles $\Box\text{member_of}$ and $\Box\Diamond\text{member_of}$, respectively. Formally, a *temporal role* is a role name prefixed by a finite sequence of \Box and \Diamond operators, and the corresponding semantic clauses are

$$\begin{aligned} (\Diamond r)^{\exists} &= \{(n, d, d') \mid \exists m \geq n \ (m, d, d') \in r^{\exists}\}, \\ (\Box r)^{\exists} &= \{(n, d, d') \mid \forall m \geq n \ (m, d, d') \in r^{\exists}\}. \end{aligned}$$

Unfortunately, it is easy to modify the proof of Theorem 4 to show undecidability of concept satisfiability w.r.t. TBoxes in $\text{LTL}_{\mathcal{ALC}}$ when TBoxes are allowed to contain a single temporal role of the form $\Box r$: simply replace r by $\Box r$ in the reduction of the recurrent tiling problem. Without TBoxes, satisfiability of $\text{LTL}_{\mathcal{ALC}}$ concepts with temporal roles has not yet been investigated. In fact, results have only been obtained for the case where LTL is replaced by standard modal logics such as K or S5 [45, 24, 7]. We briefly discuss the S5 case. It has been argued in [7] that, for the encoding of temporal conceptual database models, it is often sufficient to use the operators \Diamond_u (‘at some time point’) and \Box_u (‘at all time points’) applied to concepts and roles, instead of LTL operators. The semantics of \Diamond_u is defined as

$$\begin{aligned} (\Diamond_u C)^{\exists} &= \{(n, d) \mid \exists m \ (m, d) \in C^{\exists}\}, \\ (\Diamond_u r)^{\exists} &= \{(n, d, d') \mid \exists m \ (m, d, d') \in r^{\exists}\}, \end{aligned}$$

and \Box_u is dual to \Diamond_u . Observe that \Diamond_u and \Box_u do not distinguish between future and past. For this reason, these operators behave exactly like the modal S5 operators, and we denote the resulting language by $\text{S5}_{\mathcal{ALC}}$. The following result is proved in [7]. The technique used for proving the upper bound is an extension of a proof in [45].

Theorem 6 ([7]). *In $\text{S5}_{\mathcal{ALC}}$ with temporal roles, concept satisfiability w.r.t. TBoxes is 2EXPTIME -complete.*

3.6 Varying the DL Component

Theorem 2 is rather robust under extensions of the DL component \mathcal{ALC} . For example, if \mathcal{L} is a DL between \mathcal{ALC} and SHIQ (for a definition see, e.g., [33]) then $\text{LTL}_{\mathcal{L}}$ concept satisfiability w.r.t. TBoxes and with expanding domains has the same complexity as satisfiability in \mathcal{L} w.r.t. TBoxes, and the same holds for the case without TBoxes. Theorem 3 and the constant domain case is equally robust. In particular, concept satisfiability in LTL_{SHIQ} w.r.t. TBoxes is EXPTIME -complete, both with expanding and constant domains.

As a lower bound, Theorem 4 applies to any extension of \mathcal{ALC} , and it is more interesting to look at DL components that are *weaker* than \mathcal{ALC} . One such DL is \mathcal{EL} , which is obtained from \mathcal{ALC} by dropping negation (and thus also \sqcup and \forall). In $\text{LTL}_{\mathcal{EL}}$, every concept is satisfiable w.r.t. every TBox. For this reason, it is more interesting to consider *concept subsumption w.r.t. TBoxes*: given a TBox \mathcal{T} and a concept inclusion $C \sqsubseteq D$, decide whether $C \sqsubseteq D$ follows from \mathcal{T} . In pure \mathcal{EL} , concept subsumption w.r.t. TBoxes is tractable [13]. However, it is shown in [6] that concept satisfiability in $\text{LTL}_{\mathcal{ALC}}$ with TBoxes and rigid roles can be reduced to concept subsumption in $\text{LTL}_{\mathcal{EL}}$ with TBoxes and rigid roles. Thus, Theorem 4 already applies to the case of $\text{LTL}_{\mathcal{EL}}$.

Theorem 5 is less robust than Theorems 2 and 3. It can be shown by a straightforward extension of the proof in [24] that the result still holds for \mathcal{ALCI} , the extension of \mathcal{ALC} with inverse roles. We conjecture that it also holds for \mathcal{ALCQ} , the extension of \mathcal{ALC} with qualified number restrictions. However, concept satisfiability with rigid roles and without TBoxes becomes undecidable if only a single *transitive rigid role* r is added. Intuitively, the reason is that we can then enforce an infinite r -path also without a TBox. Undecidability is proved in [24] using a reduction of Post’s Correspondence Problem. In particular, it follows that concept satisfiability in LTL_{SHIQ} without TBoxes and with rigid roles is undecidable. The proof in [24] even applies to finite (but unbounded) flows of time. However, it then relies on constant domains: a straightforward combination of the arguments from [26, 34] and [24] shows that concept satisfiability in $\text{LTL}_{\mathcal{ALC}}$ with *rigid transitive roles* in expanding domain models with finite flows of time is decidable, though not in time bounded by a primitive recursive function.

In [7], it has been proved that Theorem 6 holds for \mathcal{ALCQI} in place of \mathcal{ALC} .

4. Temporal TBoxes

The TDLs considered in Section 3 allow one to describe the temporal evolution of concepts and roles, but not of con-

cept inclusions. For example, the simple assertion ‘eventually, all European countries will be EU members forever’ cannot be expressed using only temporal concepts and roles, but requires the application of temporal operators to CIs.

4.1 Temporal \mathcal{ALC} TBoxes

To allow the application of temporal operators to concept inclusions, we replace TBoxes with TBox *formulas*. Formally, a *temporal TBox* is built inductively from

$$C \sqsubseteq D, \quad \neg\varphi, \quad \varphi \wedge \psi, \quad \bigcirc\varphi, \quad \mathcal{M}\psi,$$

where, $C \sqsubseteq D$ ranges over concept inclusions and φ, ψ range over temporal TBoxes. We usually make explicit the description logic \mathcal{L} in which concept inclusions are formulated. For example, by mentioning a temporal \mathcal{ALC} TBox we mean that temporal operators may be only applied to its CIs (but not to their concepts), while in a temporal $LTL_{\mathcal{ALC}}$ TBox we allow applications of these operators to both CIs and concepts (but not to roles).

The truth of a temporal TBox φ in a model \mathcal{I} depends on the time point under consideration. We define the *truth relation* $\mathcal{I}, n \models \varphi$ as follows:

$$\begin{aligned} \mathcal{I}, n \models C \sqsubseteq D & \text{ iff } \{d \mid (n, d) \in C^{\mathcal{I}}\} \subseteq \{d \mid (n, d) \in D^{\mathcal{I}}\}, \\ \mathcal{I}, n \models \neg\varphi & \text{ iff } \mathcal{I}, n \not\models \varphi, \\ \mathcal{I}, n \models \varphi \wedge \psi & \text{ iff } \mathcal{I}, n \models \varphi \text{ and } \mathcal{I}, n \models \psi, \\ \mathcal{I}, n \models \bigcirc\varphi & \text{ iff } \mathcal{I}, n+1 \models \varphi, \\ \mathcal{I}, n \models \mathcal{M}\psi & \text{ iff } \exists m \geq n (\mathcal{I}, m \models \psi \text{ and } \\ & \forall n \leq k < m \mathcal{I}, k \models \varphi). \end{aligned}$$

We say that \mathcal{I} is a *model* of a temporal TBox φ if $\mathcal{I}, 0 \models \varphi$. There is thus a fundamental difference between a temporal interpretation \mathcal{I} being a model of a (non-temporal) \mathcal{ALC} TBox \mathcal{T} and of the related *temporal \mathcal{ALC} TBox* $\bigwedge_{C \sqsubseteq D \in \mathcal{T}} C \sqsubseteq D$: in the former case, the CIs in \mathcal{T} are interpreted *globally* and thus have to be satisfied at all time points; in the latter case, they are interpreted locally and only have to be satisfied at time point 0. Indeed, it is easy to see that a temporal interpretation \mathcal{I} is a model of \mathcal{T} iff it is a model of $\bigwedge_{C \sqsubseteq D \in \mathcal{T}} \square(C \sqsubseteq D)$.

The expressive power obtained by applying temporal operators to concepts and concept inclusions is incomparable. In a (non-temporal) $LTL_{\mathcal{ALC}}$ TBox, we can say that the extension of the concept `Independent_country` does not decrease using the CI

$$\text{Independent_country} \sqsubseteq \square \text{Independent_country},$$

but this cannot be expressed by a temporal \mathcal{ALC} TBox. On the other hand, the assertion that eventually all European

countries will be EU members forever can be expressed by the temporal \mathcal{ALC} TBox

$$\diamond \square (\text{European_country} \sqsubseteq \text{EU_member}),$$

but no (non-temporal) $LTL_{\mathcal{ALC}}$ TBox can capture this.

When working with temporal TBoxes, we are interested in *temporal TBox satisfiability*: given a temporal TBox φ , decide whether there is a model \mathcal{I} of φ . It is not necessary to consider *concept* satisfiability w.r.t. a temporal TBox because a concept C is satisfiable w.r.t. a temporal TBox φ if, and only if, the temporal TBox $\neg(\top \sqsubseteq \neg C) \wedge \varphi$ is satisfiable. We start by allowing the application of temporal operators only to CIs.

Theorem 7. *Satisfiability of temporal \mathcal{ALC} TBoxes is EXPTIME-complete.*

Similarly to the case of $LTL_{\mathcal{ALC}}$ without rigid roles, the proof uses the fact that the interaction between \mathcal{ALC} and LTL is rather limited. Indeed, the only interaction between the \mathcal{ALC} interpretations at distinct time points is via the truth of TBox axioms. This setup is similar to the temporalization of logics as studied in [19]. To prove the upper bound, let φ be a temporal \mathcal{ALC} TBox. Denote by \mathcal{C}_φ the set of concept inclusions occurring in φ , and denote by φ^* the LTL-formula that is the result of replacing every CI α in φ with a propositional variable p_α . Then φ is satisfied in some temporal interpretation if, and only if, φ^* is satisfiable in an LTL-model in which, for each n , the set

$$X_n = \{\alpha \mid \alpha \in \mathcal{C}_\varphi, n \models p_\alpha\} \cup \{\neg\alpha \mid \alpha \in \mathcal{C}_\varphi, n \models \neg p_\alpha\}$$

is satisfiable in an \mathcal{ALC} -model. The direction from left to right is clear. Conversely, consider such an LTL-model satisfying φ^* . Take, for each $n \in \mathbb{N}$, an \mathcal{ALC} interpretation \mathcal{I}_n satisfying X_n . We may assume that the domains $\Delta^{\mathcal{I}_n}$ are countably infinite and coincide. Define \mathcal{I} by setting $\mathcal{I}(n) = \mathcal{I}_n$, for $n \geq 0$. It is not hard to see that φ is satisfied in \mathcal{I} . Using this characterization, it is straightforward to prove an EXPTIME upper bound by combining decision procedures for LTL and \mathcal{ALC} . A corresponding lower bound carries over from \mathcal{ALC} .

The proof above also shows that a temporal \mathcal{ALC} TBox is satisfiable in a model with constant domains if, and only if, it is satisfiable in a model with varying domains.

4.2 Temporal $LTL_{\mathcal{ALC}}$ TBoxes

We now consider temporal $LTL_{\mathcal{ALC}}$ TBoxes, which allow the application of temporal operators to both concepts and concept inclusions. We have seen that, when temporal operators are applied only to concepts or only to concept inclusions, the interaction between the TL component and the DL component of a TDL is rather limited. In contrast,

temporal $LTL_{\mathcal{ALC}}$ TBoxes can enforce subtle interactions. For example, the temporal $LTL_{\mathcal{ALC}}$ TBox

$$\bigcirc \neg (\top \sqsubseteq \neg A) \leftrightarrow \neg (\top \sqsubseteq \bigcirc \neg A)$$

expresses the Barcan-formula $\exists x \bigcirc A(x) \leftrightarrow \bigcirc \exists x A(x)$ from modal predicate logic [24]. It is satisfied in all interpretations with constant domains. Due to such interactions, reasoning about temporal $LTL_{\mathcal{ALC}}$ TBoxes cannot be reduced to reasoning in the two components logics in a straightforward way.

Theorem 8 ([46, 24]). *Satisfiability of temporal $LTL_{\mathcal{ALC}}$ TBoxes is EXPSPACE-complete.*

We confine ourselves to a sketch of the upper bound. The crucial observation in most proofs of the PSPACE upper bound for satisfiability in LTL is that every satisfiable LTL-formula is satisfied in a model of the form

$$\mathfrak{M}(0), \dots, \mathfrak{M}(n), (\mathfrak{M}(n+1), \dots, \mathfrak{M}(m))^\omega,$$

where the $\mathfrak{M}(i)$ are propositional valuations and m is at most exponential in the length of φ . When considering temporal interpretations \mathfrak{J} satisfying a temporal $LTL_{\mathcal{ALC}}$ TBox φ , such regular models need not exist. For example, the TBox

$$\varphi = \square (\neg (\top \sqsubseteq \neg A) \wedge (A \sqsubseteq \bigcirc \square \neg A))$$

is not satisfied in any regular model simply because all interpretations $\mathfrak{J}(n)$ have to be distinct. The fundamental idea for proving an EXPSPACE upper bound is that, by abstracting from the domains of interpretations $\mathfrak{J}(n)$ using so-called quasimodels, one can regain a semantics in which regular models always exist.

Quasimodels are abstractions of temporal interpretations $\mathfrak{J} = \mathfrak{J}(0), \mathfrak{J}(1), \dots$ in which the non-temporal interpretations $\mathfrak{J}(i)$ are replaced by sets Ξ_i of *types*, called *quasistates*. Types are defined in the same way as in Section 3.3, but based on the subconcepts of all concepts that occur in the temporal TBox φ for which satisfiability is to be checked. Using the observation that there exist ‘only’ double-exponentially many quasistates, we can apply regularity arguments as used in the LTL case to quasimodels. To satisfy a TBox φ , it suffices to consider regular quasimodels of the form

$$\Xi_0, \dots, \Xi_n, (\Xi_{n+1}, \dots, \Xi_m)^\omega,$$

where m is bounded *double-exponentially* in the length of φ . This explains the increase in complexity from PSPACE to EXPSPACE. Of course, a number of conditions have to be imposed on sequences Ξ_0, Ξ_1, \dots of quasistates so that they can serve as a quasimodel. We will not describe the conditions in detail here, but refer the reader to [24]. The

conditions guarantee that one can re-construct a temporal interpretation from a quasimodel. For example, for each Ξ_i , there has to exist an interpretation \mathcal{I}_i satisfying *exactly* the types in Ξ_i (where concepts starting with a temporal operator are regarded as concept names). This condition ensures that the description logic part of a quasistate is satisfiable. For the temporal part, it is required that for each type $t \in \Xi_n$, there exists a *run* $r : \mathbb{N} \rightarrow \bigcup_{i \geq 0} \Xi_i$ such that

- $r(i) \in \Xi_i$, for $i \geq 0$,
- $r(n) = t$,
- $\bigcirc C \in r(n)$ iff $C \in r(n+1)$,
- $\mathcal{C}UD \in r(n)$ iff there exists $m \geq n$ with $D \in r(m)$ and $C \in r(k)$ for all $n \leq k < m$.

When constructing a temporal interpretation \mathfrak{J} from a quasimodel, the domain $\Delta^{\mathfrak{J}}$ of \mathfrak{J} consists of all such runs. The condition that \mathbb{N} is the domain of a run reflects the fact that we are interested in satisfiability in models with constant domain.

The proof can easily be adapted to expanding domains. In particular, the domains of runs are then upward-closed subsets of \mathbb{N} . This yields an EXPSPACE upper bound also for the case of expanding domains. Alternatively, this bound can be shown using the reduction to constant domains mentioned at the end of Section 3.3.

Tableau-based algorithms for checking satisfiability of temporal $LTL_{\mathcal{ALC}}$ TBoxes that combine a Wolper-style tableau algorithm for LTL [44] with a tableau algorithm for \mathcal{ALC} have been developed in [43] (for expanding domains) and [36] (for constant domains). Both algorithms are based on quasimodels. An implementation for expanding domains is presented in [27].

4.3 Varying the DL Component

The only noteworthy properties of \mathcal{ALC} used in the proof of Theorem 7 are that (i) any satisfiable Boolean combination of concept inclusions is satisfiable in a countably infinite model, and (ii) satisfiability of Boolean combinations of concept inclusions in \mathcal{ALC} is in EXPSPACE (and indeed even in EXPTIME). Thus, Theorem 7 generalizes to any DL with these properties, such as *SHIQ*. In general, if \mathcal{ALC} is replaced by a decidable fragment \mathfrak{F} of first-order logic without equality, and temporal operators are applied to closed formulas only (which is the case for the standard translation of temporal \mathcal{ALC} TBoxes into first-order temporal logic), then the complexity of the resulting fragment of first-order temporal logic will have exactly the same complexity as \mathfrak{F} , if it is at least PSPACE-hard.

Theorem 8 can also be generalized to standard DLs such as *SHIQ*. Notably, a temporal extension of the description logic \mathcal{DLR} , which provides for n -ary relations, was

considered in [5]. The generalization to fragments of first-order temporal logics has been studied systematically, and the resulting fragments are known as *monodic fragments* [31]. In monodic fragments, temporal operators are applied to formulas with at most one free variable. Note that this is the case for the standard translation of temporal $LTL_{\mathcal{ALC}}$ TBoxes into first-order temporal logic. The quasimodel technique has been refined to deal with a variety of monodic fragments [24].

5. Temporal ABoxes

So far, we have neglected ABoxes in TDLs. Similar to TBoxes, there are two approaches to including them: either use a non-temporal ABox that is interpreted in a temporal interpretation, or define temporal ABoxes in which temporal operators can be applied to ABox assertions. We start with the former.

A temporal interpretation \mathcal{I} is a *model* of a (non-temporal) ABox \mathcal{A} if \mathcal{A} is satisfied in \mathcal{I} at time point 0, i.e., $(0, a^{\mathcal{I}}) \in C^{\mathcal{I}}$ for all $C(a) \in \mathcal{A}$ and $(0, a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$ for all $r(a, b) \in \mathcal{A}$. The relevant reasoning problem is *ABox consistency w.r.t. TBoxes*: given a (non-temporal) ABox \mathcal{A} and a (non-temporal) TBox \mathcal{T} , decide whether there is a common temporal model of \mathcal{A} and \mathcal{T} . In all standard TDLs, ABox consistency w.r.t. TBoxes has the same complexity as concept satisfiability w.r.t. TBoxes. In particular, Theorems 2 and 3 easily generalize to ABox consistency.

The case of temporal ABoxes is more interesting. Formally, a *temporal ABox* is a formula built inductively from

$$C(a), \quad r(a, b), \quad \neg\varphi, \quad \varphi \wedge \psi, \quad \bigcirc\varphi, \quad \varphi \mathcal{U} \psi,$$

where $C(a)$ and $r(a, b)$ range over ABox assertions and φ, ψ range over temporal ABoxes. The truth relation for temporal ABoxes is defined in the same way as for temporal TBoxes, with the obvious additional clauses for atoms $C(a)$ and $r(a, b)$. The relevant reasoning problem is *temporal ABox consistency w.r.t. TBoxes*: given a temporal ABox \mathcal{A} and a (non-temporal) TBox \mathcal{T} , decide whether \mathcal{A} and \mathcal{T} have a common model. A common generalization of temporal ABoxes and temporal TBoxes is provided by *temporal knowledge bases* (KBs), which allow atoms of the form $C \sqsubseteq D$, $C(a)$, and $r(a, b)$, and whose syntax and semantics are otherwise defined in the same way as for temporal TBoxes and ABoxes. The relevant reasoning problem is *temporal KB consistency*: given a temporal KB \mathcal{K} , decide whether \mathcal{K} has a model.

Theorems 7 and 8 can both be extended from temporal TBox satisfiability to temporal KB consistency. To extend Theorem 8, one needs to develop a notion of quasimodels that takes into account ABoxes, as done in [24]. Theorem 7 relies on the fact that temporal \mathcal{ALC} TBoxes can enforce

only a weak interaction between the TL component and the DL component. The interaction is no stronger in the case of temporal KBs, and indeed it is not difficult to extend the sketched proof of Theorem 7 to show that temporal \mathcal{ALC} KB consistency is EXPTIME-complete. Notably, the proof shows that a temporal \mathcal{ALC} KB cannot even be used to define rigid concepts. It is thus natural to increase the expressive power of temporal KBs by adding rigid concepts and/or rigid roles.

Already the addition of rigid concepts increases the interaction between the TL component and the DL component in a non-trivial way and leads to an increase in complexity.

Theorem 9 ([9]). *In \mathcal{ALC} , temporal KB consistency with rigid concepts is NEXPTIME-complete. The lower bound holds even if concepts $\exists r.C$ and $\forall r.C$ are disallowed.*

The lower bound is proved by a reduction of the tiling problem that requires the tiling of a torus of exponential size. The \mathcal{ALC} component is used to ensure that, already at time point 0, each position of the torus is represented by a domain element. Positions are described by a binary encoding in terms of rigid concept names, and the tiling is also represented using rigid concept names. Rigidity of the concept names ensures that the tiling is preserved in all later time points, and it remains for the TL component to check that positions are represented uniquely, and that colours of adjacent tiles are identical.

The upper bound is proved by a reduction to reasoning in the component logics \mathcal{ALC} and LTL. The proof is somewhat similar to the one of Theorem 7, but needs to take into account the increased interaction between the \mathcal{ALC} and LTL components. In particular, it starts by guessing the combinations of rigid concept names that are satisfied in the temporal model. In contrast to temporal KB consistency, temporal ABox consistency with TBoxes and rigid concepts is still EXPTIME-complete [9].

When rigid concepts *and* roles are added to temporal KBs, the complexity increases further. At first sight, one might even think that temporal KB consistency becomes undecidable because we can easily enforce the existence of an $\mathbb{N} \times \mathbb{N}$ -grid; cf. Theorem 4. However, without temporal operators on concepts, we are not able to express that adjacent tiles have the same colour (axiom 3 in the proof of Theorem 4).

Theorem 10 ([9]). *In \mathcal{ALC} , temporal KB consistency with rigid concepts and rigid roles is 2EXPTIME-complete. The lower bound already applies to temporal ABox consistency w.r.t. TBoxes.*

The upper bound is again proved by a reduction to reasoning in the components \mathcal{ALC} and LTL. The increased interaction between the two components due to rigid roles is addressed by replacing the individual \mathcal{ALC} consistency

checks (one for each time point) with a single such check that, intuitively, integrates all time points. If only the temporal operators \diamond and \square are allowed in a temporal \mathcal{ALC} KB, then consistency with rigid concepts and rigid roles is EXPTIME-complete [9].

6. Temporal DL-Lite

In this section, we survey recent results on temporal *DL-Lite*. The *DL-Lite* family of lightweight DLs has been introduced and investigated in [14, 15, 1] with the aim of establishing maximal DLs for which the data complexity of query answering stays within LOGSPACE. In our brief survey, we consider three members of this family: the most expressive *DL-Lite_{bool}* and its Horn and Krom fragments *DL-Lite_{horn}* and *DL-Lite_{krom}*. *DL-Lite_{bool}* has the following concept constructors, for $q \geq 1$:

$$\underbrace{A, \perp, \top, \geq qr, \geq qr^-, \neg C, C \sqcap D,}_{B}$$

Concepts built only from the constructors marked with B are called *basic*. In contrast to \mathcal{ALC} , *DL-Lite_{bool}* has *inverse roles* (r^-) and *number restrictions* ($\geq qr$) which, however, are not *qualified*: $a \in (\geq qr)^{\mathcal{I}}$ means that there are at least q distinct r -arrows starting from a , but the concept membership of their destination cannot be specified (thus $\exists r.C$ is not expressible in *DL-Lite_{bool}*). *DL-Lite_{horn}* is the sublanguage of *DL-Lite_{bool}* with CIs of the form $\sqcap_k B_k \sqsubseteq B$, for basic B, B_k . Finally, *DL-Lite_{krom}* allows only CIs of the form $B_1 \sqsubseteq B_2, B_1 \sqsubseteq \neg B_2, \neg B_1 \sqsubseteq B_2$ with basic B_i . Note that all of these logics can say that a role r is functional: $\geq 2r \sqsubseteq \perp$. Concept satisfiability w.r.t. TBoxes is NP-complete for *DL-Lite_{bool}*, P-complete for *DL-Lite_{horn}*, and NLOGSPACE-complete for *DL-Lite_{krom}*; see [1] and references therein.

Consider now temporal extensions of these DLs. We survey what is currently known about the problem of deciding consistency of temporal $LTL_{DL-Lite}$ KBs with rigid roles, i.e., temporal operators can be applied to concepts, concept inclusions, and ABox assertions, and some roles may have a rigid interpretation. Note that only the application of Boolean operators to concepts (but not to CIs and ABox assertions) is restricted in temporal $LTL_{DL-Lite_{krom}}$ KBs, and likewise for $LTL_{DL-Lite_{horn}}$.

Recall that satisfiability of temporal $LTL_{\mathcal{ALC}}$ TBoxes with rigid roles is highly undecidable by Theorem 4. As in the case of Theorem 10, it may thus look as if consistency of temporal $LTL_{DL-Lite_{bool}}$ KBs with rigid roles is undecidable because we can easily enforce the existence of an $\mathbb{N} \times \mathbb{N}$ -grid. However, despite offering temporal operators on concepts, *DL-Lite_{bool}* is not capable of expressing that adjacent tiles have the same colour. In fact, it turns out that

temporal $LTL_{DL-Lite_{bool}}$ TBoxes can be embedded into the one-variable fragment of first-order temporal logic, which is known to be EXPSpace-complete; see, e.g., [24]. The converse embedding is also possible, and thus we obtain the following result.

Theorem 11 ([6]). *Consistency of temporal $LTL_{DL-Lite_{bool}}$ KBs with rigid roles is EXPSpace-complete.*

The complexity of deciding consistency of temporal $LTL_{DL-Lite_{krom}}$ KBs is an open problem. However, for temporal $LTL_{DL-Lite_{krom}}$ KBs in which only the next-time operator \circ is applied to basic concepts within CIs, consistency is PSPACE-complete. The proof of this result uses the fact that there are polynomially many CIs that are consequences of a set of Krom CIs (they are Krom CIs as well), and the ‘local’ character of the \circ operator applied to concepts. In contrast, for temporal $LTL_{DL-Lite_{horn}}$ KBs, the following rather surprising result can be shown by encoding the EXPSpace-complete $\mathbb{N} \times 2^n$ corridor tiling problem.

Theorem 12 ([6]). *Consistency of temporal $LTL_{DL-Lite_{horn}}$ KBs with rigid roles is EXPSpace-complete.*

One of the main reasons for these ‘positive’ results is as follows: as there is no constructor $\exists r.C$ in *DL-Lite*, we can actually encode rigid roles using temporal constraints on *unary* predicates. Although we obtain unintended models where roles are not rigid, the language is too weak to notice this.

7. Branching Temporal Logic

The linear time temporal logic LTL considered so far is not able to distinguish between possible, actual, and necessary future developments. Suppose, for example, that we want to describe countries that can join the EU in the future. The concept inclusion

$$EU_candidate \sqsubseteq \diamond EU_member,$$

expresses that, sooner or later, every EU candidate will join the EU. However, this statement seems too strong. What we actually mean is that, under certain circumstances, an EU candidate may join the EU in the future—there is still a possibility that it will stay outside the union forever. A natural way of formalising statements of this sort is to switch to branching time and add the CTL path quantifiers A and E that allow quantification over ‘possible (future) histories.’ If E is understood as ‘it is possible that’ and A as ‘it is necessary that,’ then

$$EU_candidate \sqsubseteq E\diamond EU_member$$

means that each EU candidate has the possibility (a possible history) to join the EU. At each moment of time, this

statement is consistent with, say,

$$E(\text{EU_candidate} \sqsubseteq A \rightarrow \diamond \text{EU_member}).$$

A natural branching model of time consists of infinite trees where the root represents the current moment of time (or genesis) and the branches starting from the root represent possible histories.

There are several temporal logics for branching time; see, e.g., [25]. Here we only consider *computational tree logic* CTL* [18] and its fragment CTL [17]. The language of CTL* is the straightforward extension of the language of LTL with the *path (or history) quantifiers* E and A. In CTL, the path quantifiers and temporal operators may occur only in the form $E(CUD)$, $A(C \sqsubseteq D)$, etc. (so $A(C \sqsubseteq \circ D)$ is not a well-formed CTL formula).

A *branching time interpretation* $\mathfrak{I} = (\mathfrak{T}^{\mathfrak{J}}, \Delta^{\mathfrak{J}}, \cdot^{\mathfrak{J}})$ consists of an ω -tree $\mathfrak{T}^{\mathfrak{J}} = (W, <)$, a nonempty domain $\Delta^{\mathfrak{J}}$, and a function $\cdot^{\mathfrak{J}}$ that maps every $A \in \mathbf{N}_C$ to some $A^{\mathfrak{J}} \subseteq W \times \Delta^{\mathfrak{J}}$ and every $r \in \mathbf{N}_R$ to $r^{\mathfrak{J}} \subseteq W \times \Delta^{\mathfrak{J}} \times \Delta^{\mathfrak{J}}$. The elements on each full branch h of $\mathfrak{T}^{\mathfrak{J}}$ are ordered by $<$, and each such branch j is required to be order-isomorphic to $(\mathbb{N}, <)$. Intuitively, branches h represent possible histories, and their elements represent time points in these histories. Thus, the usual temporal operators can be applied to a given history, while E and A quantify over the set of all histories that include the current time point. This means that the interpretation of a complex concept depends not only on the time point w , but also on the history h that this time point belongs to, e.g.,

- $(w, h, d) \in A^{\mathfrak{J}}$ iff $(w, d) \in A^{\mathfrak{J}}$, for a concept name A ,
- $(w, h, d) \in (\diamond C)^{\mathfrak{J}}$ iff there exists $w' \in h$ such that $w < w'$ and $(w', h, d) \in C^{\mathfrak{J}}$,
- $(w, h, d) \in (EC)^{\mathfrak{J}}$ iff there exists a full branch h' in \mathfrak{T} such that $w \in h'$ and $(w, h', d) \in C^{\mathfrak{J}}$.

We consider temporal $\text{CTL}_{\mathcal{ALC}}^*$ and $\text{CTL}_{\mathcal{ALC}}$ TBoxes and KBs. Thus, we allow the application of temporal operators and (in the case of CTL*) path quantifiers to concepts, concept inclusions, and ABox assertions (in the case of KBs), but not to roles. Both problems are defined by extending the definitions for $\text{LTL}_{\mathcal{ALC}}$ in a straightforward way. In particular, the truth-relation $(\mathfrak{I}, w, h) \models \varphi$ for temporal $\text{CTL}_{\mathcal{ALC}}^*$ KBs φ is defined by combining the inductive definitions above with those for \mathcal{ALC} .

The addition of branching to temporal interpretations changes the computational properties of TDLs in a drastic way. We remind the reader that CTL* and CTL are 2EXPTIME- and EXPTIME-complete, respectively. The problem with TDLs based on CTL* and CTL is that, intuitively, they are ‘ $2\frac{1}{2}$ -dimensional’ with branches contributing $\frac{1}{2}$. This is not a good sign as we know from [24] that almost all 3D products of modal-like logics are undecidable.

Theorem 13 ([32]). *Satisfiability of temporal $\text{CTL}_{\mathcal{ALC}}^*$ TBoxes is undecidable.*

The original proof in [32] was given for one-variable first-order CTL* with only one temporal operator \diamond . It is not hard to see, however, that quantification over a single variable can be simulated by means of concept inclusions of the form $\top \sqsubseteq C$ and the Boolean operators on CIs, which are available even in $\text{CTL}_{\text{DL-Lite}_{\text{bool}}}$ (which is, therefore, also undecidable).

Positive decidability results can be obtained by restricting the application of temporal operators and path quantifiers in various ways [12]. An example is the replacement of CTL* with CTL as the temporal component.

Theorem 14 ([32, 12]). *Consistency of temporal $\text{CTL}_{\mathcal{ALC}}$ KBs is decidable.*

The proof extends the quasimodel technique for $\text{LTL}_{\mathcal{ALC}}$ sketched in Section 4 to $\text{CTL}_{\mathcal{ALC}}$. The existence of a quasimodel is decided by an embedding into the monadic second-order theory of trees, and thus only a non-elementary upper bound is known.

Many challenging open problems remain to be investigated in branching time TDLs. In contrast to linear time TDLs, no tight complexity results are known and simpler reasoning problems than satisfiability of temporal $\text{CTL}_{\mathcal{ALC}}$ TBoxes have not yet been investigated.

References

- [1] A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyashev. DL-Lite in the light of first-order logic. In *Proceedings of AAI-07*, pages 361–366. MIT Press, 2007.
- [2] A. Artale and E. Franconi. A temporal description logic for reasoning about actions and plans. *Journal of Artificial Intelligence Research*, 9:463–506, 1998.
- [3] A. Artale and E. Franconi. A survey of temporal extensions of description logics. *Annals of Mathematics in Artificial Intelligence*, 30(1-4):171–210, 2000.
- [4] A. Artale and E. Franconi. Temporal description logics. In *Handbook of Time and Temporal Reasoning in Artificial Intelligence*, pages 375–388. Elsevier, 2005.
- [5] A. Artale, E. Franconi, F. Wolter, and M. Zakharyashev. A temporal description logic for reasoning over conceptual schemas and queries. In *Proceedings of JELIA-02*, volume 2424 of *LNCS*, pages 98–110. Springer, 2002.
- [6] A. Artale, R. Kontchakov, C. Lutz, F. Wolter, and M. Zakharyashev. Temporalising tractable description logics. In *Proceedings of TIME-07*. IEEE Press, 2007.
- [7] A. Artale, C. Lutz, and D. Toman. A description logic of change. In *Proceedings of IJCAI-07*, pages 218–223, 2007.
- [8] F. Baader. Augmenting concept languages by transitive closure of roles: An alternative to terminological cycles. In *Proceedings of IJCAI-91*, pages 446–451. Morgan Kaufmann, 1991.

- [9] F. Baader, S. Ghilardi, and C. Lutz. LTL over description logic axioms. In *Proceedings of DL-08*, CEUR Workshop Proceedings. CEUR-WS.org, 2008.
- [10] F. Baader, R. Küsters, and F. Wolter. Extensions to description logics. In *Description Logic Handbook*, pages 219–261. Cambridge University Press, 2003.
- [11] F. Baader, C. Lutz, H. Sturm, and F. Wolter. Fusions of description logics and abstract description systems. *Journal of Artificial Intelligence Research*, 16:1–58, 2002.
- [12] S. Bauer, I. Hodkinson, F. Wolter, and M. Zakharyashev. On non-local propositional and weak monodic quantified CTL. *Journal of Logic and Computation*, 14(1):3–22, 2004.
- [13] S. Brandt. Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and — what else? In *Proceedings of ECAI-04*, pages 298–302. IOS Press, 2004.
- [14] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. DL-Lite: Tractable description logics for ontologies. In *Proceedings of AAI-05*, pages 602–607. MIT Press, 2005.
- [15] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. In *Proceedings of KR-06*, pages 260–270, 2006.
- [16] D. Calvanese, M. Lenzerini, and D. Nardi. Description logics for conceptual data modeling. In *Logics for Databases and Information Systems*, pages 229–263. Kluwer Academic Publisher, 1998.
- [17] E. Clarke and E. Emerson. Design and synthesis of synchronisation skeletons using branching time temporal logic. In *Logic of Programs*, volume 131 of LNCS, pages 52–71. Springer, 1981.
- [18] E. Emerson and J. Halpern. Sometimes and never revisited: on branching versus linear time temporal logic. *Journal of the ACM*, 33(1):151–178, 1986.
- [19] M. Finger and D. Gabbay. Adding a temporal dimension to a logic system. *Journal of Logic, Language and Information*, 2:203–233, 1992.
- [20] M. Fischer and R. Ladner. Propositional modal logic of programs. In *Proceedings of STOC-77*, pages 286–294. ACM Press, 1977.
- [21] M. Fischer and R. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18:194–211, 1979.
- [22] E. Franconi and D. Toman. Fixpoint extensions of temporal description logics. In *Proceedings of DL-03*, volume 81 of CEUR Workshop Proceedings. CEUR-WS.org, 2003.
- [23] D. Gabbay, I. Hodkinson, and M. Reynolds. *Temporal Logic: Mathematical Foundations and Computational Aspects, Volume 1*. Oxford University Press, 1994.
- [24] D. Gabbay, A. Kurucz, F. Wolter, and M. Zakharyashev. *Many-Dimensional Modal Logics: Theory and Applications*. Elsevier, 2003.
- [25] D. Gabbay, M. Reynolds, and M. Finger. *Temporal Logic: Mathematical Foundations and Computational Aspects, Volume 2*. Oxford University Press, 2000.
- [26] D. Gabelaia, A. Kurucz, F. Wolter, and M. Zakharyashev. Non-primitive recursive decidability of products of modal logics with expanding domains. *Annals of Pure and Applied Logic*, 142(1-3):245–268, 2006.
- [27] C. Günzel. *A tableaux-based reasoner for temporalised description logics*. PhD thesis, University of Liverpool, 2005.
- [28] J. Y. Halpern and Y. Shoham. A propositional modal logic of time intervals. *Journal of the ACM*, 38(4):935–962, 1991.
- [29] D. Harel. Effective transformations on infinite trees, with applications to high undecidability, dominoes, and fairness. *Journal of the ACM*, 33:224–248, 1986.
- [30] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, 2000.
- [31] I. Hodkinson, F. Wolter, and M. Zakharyashev. Decidable fragments of first-order temporal logics. *Annals of Pure and Applied Logic*, 106:85–134, 2000.
- [32] I. Hodkinson, F. Wolter, and M. Zakharyashev. Decidable and undecidable fragments of first-order branching temporal logics. In *Proceedings of LICS-02*, pages 393–402. IEEE Press, 2002.
- [33] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for very expressive description logics. *Logic Journal of the IGPL*, 8(3):239–264, 2000.
- [34] B. Konev, F. Wolter, and M. Zakharyashev. Temporal logics over transitive states. In *Proceedings of CADE-05*, pages 182–203, 2005.
- [35] C. Lutz. Interval-based temporal reasoning with general TBoxes. In *Proceedings of IJCAI-01*, pages 89–94. Morgan Kaufmann, 2001.
- [36] C. Lutz, H. Sturm, F. Wolter, and M. Zakharyashev. Tableaux for temporal description logic with constant domains. In *Proceedings of IJCAR-01*, volume 2083 of LNCS, pages 121–136, 2001.
- [37] K. Schild. A correspondence theory for terminological logics: preliminary report. In *Proceedings of IJCAI-91*, pages 466–471. Morgan Kaufmann, 1991.
- [38] K. Schild. Combining terminological logics with tense logic. In *Proceedings of EPIA-93*, volume 727 of LNCS, pages 105–120. Springer, 1993.
- [39] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48:1–26, 1991.
- [40] A. Schmiedel. Temporal terminological logic. In W. Dietterich, Tom; Swartout, editor, *Proceedings of AAI-90*, pages 640–645. MIT Press, 1990.
- [41] A. Sistla and E. Clarke. The complexity of propositional linear temporal logics. *Journal of the ACM*, 32:733–749, 1985.
- [42] L. Stockmeyer. *The Complexity of Decision Problems in Automata Theory and Logic*. PhD thesis, MIT, 1974.
- [43] H. Sturm and F. Wolter. A tableau calculus for temporal description logic: the expanding domain case. *Journal of Logic and Computation*, 12:809–838, 2002.
- [44] P. Wolper. The tableau method for temporal logic: An overview. *Logique et Analyse*, 28:119–152, 1985.
- [45] F. Wolter and M. Zakharyashev. Modal description logics: modalizing roles. *Fundamenta Informaticae*, 39:411–438, 1999.
- [46] F. Wolter and M. Zakharyashev. Temporalizing description logics. In *Frontiers of Combining Systems II*, volume 1794 of LNCS, pages 379–401. Springer, 2000.

A. Correctness of the Algorithm in Section 3.3

Let M_0, \dots, M_m be the sets computed by the algorithm. Set $n := |\Pi(C, \mathcal{T})|$ and $\mathfrak{T}_i := \{t \mid (t, i) \in M_m\}$ for $i \leq n$.

Lemma 15. *For all $i < n$, we have*

1. $\mathfrak{T}_i \supseteq \mathfrak{T}_{i+1}$;
2. $\mathfrak{T}_i = \mathfrak{T}_{i+1}$ implies $\mathfrak{T}_i = \mathfrak{T}_{i+\ell}$ for all $\ell \leq n - i$.

Proof. (1) Let

$$M = M_m \cup \{(t, j) \mid (t, \ell) \in M_m \text{ for some } \ell \geq j\}.$$

We show that M satisfies conditions 1–4 of the elimination procedure. Based on this, it is trivial to show by induction on i that $M \subseteq M_i$ for all $i \leq m$, which implies (1). We only consider condition 4. Let $(t, j) \in M$ and $CUD \in t$. If $(t, j) \in M_m$, we are done. Otherwise, we have $(t, \ell) \in M_m$ for some $\ell \geq j$. It follows that there are temporal types $(t_1, i_1), \dots, (t_k, i_k) \in M_m$ satisfying (i)–(iv) for $(t, \ell) \in M_m$. Define a sequence $(t'_1, i'_1), \dots, (t'_k, i'_k)$ by setting $(t'_p, i'_p) := (t_p, \delta(i + p - 1))$ for $1 \leq p \leq k$. By definition of M , this sequence exists in M . It clearly satisfies (i)–(iv) for $(t, j) \in M$.

(2) Assume that $\mathfrak{T}_i = \mathfrak{T}_{i+1}$ and let

$$M = M_m \cup \{(t, j) \mid (t, i) \in M_m \text{ and } i \leq j \leq n\}.$$

As before, it suffices to show that elimination conditions 1–4 are satisfied for M . Let $(t, j) \in M \setminus M_m$. We concentrate on conditions 3 and 4. The definition of M yields $(t, i) \in M_m$ and so $(t, i + 1) \in M_m$ since $\mathfrak{T}_i = \mathfrak{T}_{i+1}$. Hence, there is a $(t', i) \in M_m$ such that t' and t are compatible. Again by definition of M , $(t', j - 1) \in M$, and we are done. Now let $CUD \in t$. As $(t, i) \in M_m$, there is a sequence of temporal types $(t_1, i_1), \dots, (t_k, i_k) \in M_m$ satisfying (i)–(iv) of elimination condition 4 for $(t, i) \in M_m$. By definition of M , $(t_\ell, p) \in M$ for $1 \leq \ell \leq k$ and $i \leq p \leq n$. Thus, the sequence $(t'_1, i'_1), \dots, (t'_k, i'_k)$ defined by setting $(t'_\ell, i'_\ell) := (t_\ell, \delta(i + \ell - 1))$ exists in M . It clearly satisfies (i)–(iv) for $(t, j) \in M$. \square

Define an infinite set $M := \{(t, i) \mid (t, \delta(i)) \in M_m\}$ and let the conditions 1'–4' be obtained from the elimination conditions 1–4 by replacing each occurrence of $\delta(w)$ with w and allowing applications of these conditions also to temporal types (t, i) with $i > n$.

Lemma 16. *For all types $(t, i) \in M$, conditions 1'–4' are satisfied.*

Proof. Since $|\mathfrak{T}_0| \leq n$, Lemma 15 implies that $\mathfrak{T}_{n-1} = \mathfrak{T}_n$ or $\mathfrak{T}_n = \emptyset$. In the latter case, condition 2 implies that M_m , and so also M , is empty, which means that we are done. If $\mathfrak{T}_{n-1} = \mathfrak{T}_n$, it is easy to use the fact that M_m satisfies conditions 1–4 to show that M satisfies conditions 1'–4'. \square

A temporal pre-interpretation $\mathfrak{P} = (\Delta^{\mathfrak{P}}, \cdot^{\mathfrak{P}})$ consists of a nonempty domain $\Delta^{\mathfrak{P}}$ and a partial function $\cdot^{\mathfrak{P}}$ from $\mathbb{N} \times \Delta^{\mathfrak{P}}$ to $\Pi(C, \mathcal{T})$.

We are now ready to prove correctness of the algorithm. For soundness, assume that the algorithm returns ‘satisfiable,’ i.e., there is a $(t, 0) \in M_m$ with $C \in t$. Then $(t, 0) \in M$. We construct a sequence of temporal pre-interpretations $\mathfrak{P}_0, \mathfrak{P}_1, \dots$ such that for all $i \geq 0$, $d \in \Delta^{\mathfrak{P}_i}$, and $j \geq 0$ with $\mathfrak{P}_i(j, d)$ defined, we have $j \leq i$, $\mathfrak{P}_i(j', d)$ is defined for all $j' < j$, and

$$(\mathfrak{P}_i(j, d), j) \in M. \quad (\dagger)$$

To start, we set $\Delta^{\mathfrak{P}_0} := \{d_0\}$ and $\mathfrak{P}_0(0, d_0) := t$. To define \mathfrak{P}_{i+1} , we start with $\mathfrak{P}_{i+1} := \mathfrak{P}_i$ and proceed in two steps. For Step 1, choose for each $j \leq i$ a set $T_j \subseteq \Pi(C, \mathcal{T})$ such that

- for each $d \in \Delta^{\mathfrak{P}_{i+1}}$ and $\exists r.D \in \mathfrak{P}_{i+1}(j, d)$ such that $S := \{D\} \cup \{\neg E \mid \neg \exists r.E \in \mathfrak{P}_{i+1}(j, d)\} \not\subseteq \mathfrak{P}_{i+1}(j, d')$ for all $d' \in \Delta^{\mathfrak{P}_{i+1}}$, there is a $t \in T_j$ with $S \subseteq t$;
- $T_j \subseteq \{t \mid (t, j) \in M\}$.

Such T_j exist by (\dagger) and since M satisfies condition 1'. For each $j \leq i$ and $t \in T_j$, add a fresh element $e_{j,t}$ to $\Delta^{\mathfrak{P}_{i+1}}$. By (\dagger) and condition 3', there is a sequence $(t_0, 0), \dots, (t_j, j) \in M$ such that $t_j = t$. For all $j \leq i$, set $\mathfrak{P}_{i+1}(j, e_{j,t}) := t_j$.

For Step 2, consider each $d \in \Delta^{\mathfrak{P}_{i+1}}$ and $j \leq i$ with $\mathfrak{P}_{i+1}(j, d)$ defined and $\mathfrak{P}_{i+1}(j + 1, d)$ undefined. First assume that there is no concept $DUE \in \mathfrak{P}_{i+1}(j, d)$ with $E \notin \mathfrak{P}_{i+1}(j, d)$. By (\dagger) and condition 2', there is a $(t, j + 1) \in M$ such that $\mathfrak{P}_{i+1}(j, d)$ and t are compatible. Set $\mathfrak{P}_{i+1}(j + 1, d) := t$. Now assume that there is a concept $DUE \in \mathfrak{P}_{i+1}(j, d)$ with $E \notin \mathfrak{P}_{i+1}(j, d)$. Assume w.l.o.g. that there is no $D'UE' \in \mathfrak{P}_{i+1}(j, d)$ and $j' \leq j$ with $D'UE', \neg E' \in \mathfrak{P}_{i+1}(j', d)$ for $j' \leq \ell \leq j$, but $DUE, \neg E \notin \mathfrak{P}_{i+1}(j', d)$. Let $(t_1, i), \dots, (t_k, i + k - 1)$ be a minimal sequence satisfying (i)–(iv) of condition 4'. Since $E \notin \mathfrak{P}_{i+1}(j, d)$, the length of this sequence is at least two. Put $\mathfrak{P}_{i+1}(j + 1, d) := t_2$. This finishes the construction of \mathfrak{P}_{i+1} .

Let $\mathfrak{P} := \bigcup_{i \geq 0} \mathfrak{P}_i$. It is easy to verify that \mathfrak{P} is a total function. We convert \mathfrak{P} into a temporal interpretation $\mathfrak{I} = (\Delta^{\mathfrak{P}}, \cdot^{\mathfrak{I}})$ as follows:

- $A^{\mathfrak{I}} := \{(i, d) \mid A \in \mathfrak{P}(i, d)\}$;
- $r^{\mathfrak{I}} := \{(i, d, d') \mid \neg \exists r.D \in \mathfrak{P}(i, d) \Rightarrow \neg D \in \mathfrak{P}(i, d')\}$.

It is routine to verify that $(0, d_0) \in C^{\mathfrak{I}}$ and \mathfrak{I} is a model of \mathcal{T} .

Now for completeness. Assume that C is satisfiable w.r.t. \mathcal{T} and let \mathfrak{I} be a temporal model of C and \mathcal{T} . For

$d \in \Delta^{\mathcal{T}}$ and $i \geq 0$, set $\text{tp}_{\mathcal{J}}(i, d) := \{D \in \text{cl}(C, \mathcal{T}) \mid (i, d) \in D^{\mathcal{J}}\}$. Define

$M := \{(t, \delta(i)) \mid t = \text{tp}_{\mathcal{J}}(j, d) \text{ for some } j \geq i \text{ and } d \in \Delta^{\mathcal{J}}\}$.

It is straightforward to verify that M satisfies conditions 1–4. Thus, it is easily proved by induction on i that $M \subseteq M_i$ for $1 \leq i \leq m$. As \mathcal{I} is a model of C , there is a $(t, 0) \in M$ with $C \in t$. Thus, the algorithm returns ‘satisfiable.’