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

LTCS–Report

Reasoning with Temporal Properties over Axioms of *DL-Lite*

Stefan Borgwardt Marcel Lippmann Veronika Thost

LTCS-Report 14-06

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

Reasoning with Temporal Properties over Axioms of *DL-Lite*

Stefan Borgwardt Marcel Lippmann
Veronika Thost

Institute of Theoretical Computer Science
Technische Universität Dresden, Germany
{stefborg,lippmann,thost}@tcs.inf.tu-dresden.de

Abstract

Recently, a lot of research has combined description logics (DLs) of the *DL-Lite* family with temporal formalisms. Such logics are proposed to be used for situation recognition and temporalized ontology-based data access. In this report, we consider *DL-Lite*-LTL, in which axioms formulated in a member of the *DL-Lite* family are combined using the operators of propositional linear-time temporal logic (LTL). We consider the satisfiability problem of this logic in the presence of so-called rigid symbols whose interpretation does not change over time. In contrast to more expressive temporalized DLs, the computational complexity of this problem is the same as for LTL, even w.r.t. rigid symbols.

Contents

1 Introduction

Description logics (DLs) [BCM+07] are a well-investigated family of logic-based knowledge representation formalisms. The combination of DLs with temporal formalisms provides expressive power to represent dynamical aspects of the application domain, e.g. the specification of a system that evolves over time. Various temporalized DLs have been proposed in the literature (see [LWZ08] for a survey). Recently, a lot of research has focused on combining members of the *DL-Lite* family with temporal logics [AKL+07, AKRZ09, AKRZ10, AKWZ13, BLT13]. Logics of the *DL-Lite* family are tailored towards conceptual modeling and ontology-based data access [CDL+09, CDL+05]. Thus, such temporalized DLs and temporal query languages are proposed to be used in context-aware applications and for temporalized ontology-based data access.

In this report, we consider *DL-Lite*-LTL, which is a combination of *DL-Lite* with propositional linear-time temporal logic (LTL) [Pnu77]. Instead of allowing temporal operators to occur within the *DL-Lite* axioms, as it is done in various other temporal extensions of *DL-Lite*, we follow the approach of [BGL12]. The latter paper introduces the temporalized DL $\mathcal{ALC}$-LTL, whose formulae combine axioms of the expressive DL $\mathcal{ALC}$ using the operators of LTL.

As an example of a *DL-Lite*-LTL formula, consider

$$\mathsf{Process}(p_1) \wedge \Diamond(\exists\mathsf{sendSignal}^-(p_1) \wedge \mathsf{Terminated}(p_1)),$$

which expresses that p_1 is a process that at some point receives a signal although it has already been terminated. In the form of *concept inclusions*, we can also incorporate terminological knowledge into our formulae. For example,

$$\Box(\exists\mathsf{sendSignal} \sqsubseteq \mathsf{Process} \wedge \exists\mathsf{sendSignal}^- \sqsubseteq \mathsf{Process}) \wedge \ldots$$

expresses the restriction that only processes can send and receive signals. We are interested here in the *satisfiability* of such formulae, i.e. in deciding whether the described situation can actually happen.

Temporal languages are often augmented with so-called *rigid* symbols, which are symbols whose interpretation does not change over time. For instance, in our example above, it makes sense to designate $\mathsf{Process}$ as rigid, but $\mathsf{sendSignal}$ as not rigid (*flexible*) to allow a process to send different signals at different points in time.

In [BGL12], it is shown that the complexity of satisfiability in $\mathcal{ALC}$-LTL increases if rigid symbols are allowed. More precisely, it jumps from ExpTime-complete (without rigid symbols) to NExpTime-complete in the presence of rigid concepts. When rigid roles are considered in addition, the satisfiability problem is even 2-ExpTime-complete.

We show in this report that for *DL-Lite*-LTL this does not apply: satisfiability in *DL-Lite*-LTL (even with rigid roles) is PSPACE-complete, and thus has the same complexity as satisfiability in LTL. To show that, we roughly follow the approach from [BGL12], where the satisfiability problem is split into two independent problems—one for the temporal dimension and one for the DL dimension. However, here we cannot treat both problems independently in order to obtain a tight complexity result. Instead, we have to integrate the *DL-Lite* satisfiability test into the PSPACE decision procedure for LTL satisfiability [SC85].

2 Preliminaries

In this section, we define the syntax and semantics of *DL-Lite*-LTL. For the DL part, we focus on *DL-Lite*$_{core}$ [ACKZ09], which is the core language of the *DL-Lite* family. Throughout this report, let $\mathsf{N_C}$, $\mathsf{N_R}$, and $\mathsf{N_I}$ be non-empty, pairwise disjoint sets of *concept*, *role*, and *individual names*, respectively. We additionally denote by $\mathsf{N_R^-}$ the set of all *roles* of the form R or R^- with $R \in \mathsf{N_R}$.

Definition 2.1. Basic concepts *B and* (general) concepts *C are built from concept names $A \in \mathsf{N_C}$ and roles $R \in \mathsf{N_R^-}$ according to the following syntax rules:*

$$B ::= A \mid \exists R \qquad\qquad C ::= B \mid \neg B$$

A concept inclusion (CI) *is of the form $B \sqsubseteq C$, where B is a basic concept and C is a general concept. An* assertion *is of the form $B(a)$ or $R(a,b)$, where B is a basic concept, $R \in \mathsf{N_R}$, and $a, b \in \mathsf{N_I}$. A* TBox *is a finite set of concept inclusions, and an* ABox *is a finite set of assertions and* negated *assertions of the form $\neg B(a)$ or $\neg R(a,b)$. Together, a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$ form an* ontology *$\mathcal{O} = (\mathcal{T}, \mathcal{A})$.*

We call both concept inclusions and assertions *axioms*. The notion of an ontology as defined above extends the common definition of a *DL-Lite*$_{core}$-ontology by negated assertions. However, the additional expressivity does not affect the complexity of reasoning in this logic [ACKZ09] (see also Section 3). For the sake of brevity, we refer to this kind of ontology as *DL-Lite*-ontology.

We sometimes use the abbreviations $R^-(a,b) := R(b,a)$, $(R^-)^- := R$, and $\neg(\neg B(a)) := B(a)$ for $R \in \mathsf{N_R}$, $a, b \in \mathsf{N_I}$, and a basic concept B.

We now define the semantics of *DL-Lite*$_{core}$.

Definition 2.2. *An* interpretation *is a pair $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where $\Delta^\mathcal{I}$ is a non-empty set (called* domain*) and $\cdot^\mathcal{I}$ is a function that assigns to every $A \in \mathsf{N_C}$ a set $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, to every $R \in \mathsf{N_R}$ a binary relation $R^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$, and to every $a \in \mathsf{N_I}$*

an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. This function is extended as follows:

$$(R^-)^{\mathcal{I}} := \{(y,x) \mid (x,y) \in R^{\mathcal{I}}\};$$
$$(\exists R)^{\mathcal{I}} := \{x \mid \text{there is an } y \in \Delta^{\mathcal{I}} \text{ such that } (x,y) \in R^{\mathcal{I}}\}; \text{ and}$$
$$(\neg C)^{\mathcal{I}} := \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}.$$

$\mathcal{I}$ is a model *of a CI $B \sqsubseteq C$ if $B^{\mathcal{I}} \subseteq C^{\mathcal{I}}$, of a concept assertion $A(a)$ if $a^{\mathcal{I}} \in A^{\mathcal{I}}$, and of a role assertion $R(a,b)$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$. It is a* model *of a negated assertion $\neg\alpha$ iff it is not a model of α. Furthermore, $\mathcal{I}$ is a* model *of a set of axioms or an ontology if it is a model of all axioms contained in it. For an axiom, set of axioms, or ontology α, we write $\mathcal{I} \models \alpha$ if $\mathcal{I}$ is a model of α. For a set of axioms or an ontology $\mathcal{O}$, we further write $\mathcal{O} \models \alpha$ if every model of $\mathcal{O}$ is also a model of α. An ontology is* consistent *if it has a model.*

We assume that all models $\mathcal{I}$ of an axiom, a set of axioms, or an ontology α satisfy the *unique name assumption* (UNA); that is, for all distinct individual names a, b occurring in α, we have $a^{\mathcal{I}} \neq b^{\mathcal{I}}$.

In a temporal setting, it may be desirable that the interpretation of certain concepts and roles does not change over time. Thus, in the following we consider a set $\mathsf{N_{RC}} \subseteq \mathsf{N_C}$ of *rigid concept names* and a set $\mathsf{N_{RR}} \subseteq \mathsf{N_R}$ of *rigid role names*. Correspondingly, we use $\mathsf{N_{RR}^-}$ to denote all *rigid roles*, i.e. roles built from rigid role names. We call basic concepts, general concepts, and CIs *rigid* if they contain only rigid symbols. Entities that are not rigid are called *flexible*.

We now define the syntax and semantics of *DL-Lite*-LTL. It differs from LTL in that the propositional variables are replaced by *DL-Lite$_{core}$* axioms.

Definition 2.3. *DL-Lite-LTL formulae are defined by induction:*

- *Every axiom is a DL-Lite-LTL formula.*
- *If ϕ_1 and ϕ_2 are DL-Lite-LTL formulae, then so are $\phi_1 \wedge \phi_2$, $\neg\phi_1$, $\bigcirc\phi_1$ ("next"), and $\phi_1 \,\mathsf{U}\, \phi_2$ ("until").*

As usual in LTL, we define $\phi_1 \vee \phi_2 := \neg(\neg\phi_1 \wedge \neg\phi_2)$, $\mathsf{true} := A(a) \vee \neg A(a)$ for some $a \in \mathsf{N_I}$ and $A \in \mathsf{N_C}$, $\Diamond\phi_1 := \mathsf{true}\,\mathsf{U}\,\phi_1$ ("eventually in the future"), and $\Box\phi_1 := \neg\Diamond\neg\phi_1$ ("always in the future").

We denote by $\mathsf{N_I}(\phi)$ the set of all individual names that occur in a *DL-Lite*-LTL formula ϕ, and similarly for $\mathsf{N_C}$, $\mathsf{N_{RC}}$, $\mathsf{N_R}$, $\mathsf{N_{RR}}$, $\mathsf{N_R^-}$, and $\mathsf{N_{RR}^-}$. Further, we use the notation $\mathsf{BC}(\phi)$ for the set of all basic concepts that can be built from $\mathsf{N_C}(\phi)$ and $\mathsf{N_R^-}(\phi)$, $\mathsf{BC}^\neg(\phi)$ for the set $\mathsf{BC}(\phi)$ extended by negation, $\mathsf{BC_R^\neg}(\phi)$ for the restriction of $\mathsf{BC}^\neg(\phi)$ to rigid concepts, and $\mathsf{BC_R}(\phi) := \mathsf{BC_R^\neg}(\phi) \cap \mathsf{BC}(\phi)$. We define the sets $\mathsf{N_I}(\mathcal{O})$, $\mathsf{BC}^\neg(\mathcal{O})$, etc. for a *DL-Lite*-ontology $\mathcal{O}$ in the same way.

The semantics of *DL-Lite*-LTL is based on *DL-Lite*-LTL structures, which are sequences of *DL-Lite*-interpretations over the same non-empty domain Δ, i.e. we make the constant domain assumption, which is also made in [BGL12].

Definition 2.4. *A DL-Lite*-LTL structure *is a sequence* $\mathfrak{I} = (\mathcal{I}_i)_{i \geq 0}$ *of interpretations* $\mathcal{I}_i = (\Delta, \cdot^{\mathcal{I}_i})$ *that* respect rigid names*, i.e. we have* $x^{\mathcal{I}_i} = x^{\mathcal{I}_j}$ *for all* $i, j \geq 0$ *and* $x \in \mathsf{N_{RC}} \cup \mathsf{N_{RR}} \cup \mathsf{N_I}$. *A DL-Lite-LTL formula* ϕ *is* valid *in a DL-Lite-LTL structure* $\mathfrak{I} = (\mathcal{I}_i)_{i \geq 0}$ *at a time point* $i \geq 0$ *(written* $\mathfrak{I}, i \models \phi$*) if the following recursive conditions are satisfied:*

$$
\begin{array}{lll}
\mathfrak{I}, i \models \alpha & \textit{iff} & \mathcal{I}_i \models \alpha \textit{ for an axiom } \alpha \\
\mathfrak{I}, i \models \phi_1 \wedge \phi_2 & \textit{iff} & \mathfrak{I}, i \models \phi_1 \textit{ and } \mathfrak{I}, i \models \phi_2 \\
\mathfrak{I}, i \models \neg \phi_1 & \textit{iff} & \textit{not } \mathfrak{I}, i \models \phi_1 \\
\mathfrak{I}, i \models \bigcirc \phi_1 & \textit{iff} & \mathfrak{I}, i+1 \models \phi_1 \\
\mathfrak{I}, i \models \phi_1 \,\mathsf{U}\, \phi_2 & \textit{iff} & \textit{there is some } k \geq i \textit{ such that } \mathfrak{I}, k \models \phi_2 \\
 & & \textit{and } \mathfrak{I}, j \models \phi_1 \textit{ for all } j,\ i \leq j < k
\end{array}
$$

As before, we additionally assume that every interpretation $\mathcal{I}_i$ *in* $\mathfrak{I}$ *satisfies the UNA w.r.t.* $\mathsf{N_I}(\phi)$. *A DL-Lite-LTL formula* ϕ *is* satisfiable *iff there is such a DL-Lite-LTL structure* $\mathfrak{I}$ *with* $\mathfrak{I}, 0 \models \phi$*, and it is* valid *if* $\mathcal{I}, 0 \models \phi$ *holds for all DL-Lite-LTL structures* $\mathfrak{I}$.

In addition to the UNA, we assume that the interpretation of individual names is rigid. This is a standard assumption in temporalized DLs [BGL12, GKWZ03, WZ00].

In propositional LTL, satisfiability is defined in the same way, with the exception that *LTL-structures* are sequences $\mathfrak{I} = (w_i)_{i \geq 0}$ of *worlds* w_i that are sets of propositional variables, and we have $\mathfrak{I}, i \models p$ for a propositional variable p iff $p \in w_i$.

In this report, we show that the satisfiability problem for *DL-Lite*-LTL formulae is PSpace-complete, and thus the same holds for validity. In our formalism, we can also incorporate background knowledge in the form of *temporal knowledge bases* $\mathcal{K} = (\mathcal{T}, (\mathcal{A}_i)_{0 \leq i \leq n})$ that consist of a global TBox and a finite sequence of ABoxes, as introduced in [BBL13, BLT13]. A *DL-Lite*-LTL structure $\mathfrak{I} = (\mathcal{I}_i)_{i \geq 0}$ is a *model* of $\mathcal{K}$ if $\mathcal{I}_i \models \mathcal{A}_i$ for all i, $0 \leq i \leq n$, and $\mathcal{I}_i \models \mathcal{T}$ for all $i \geq 0$. It is easy to see that $\mathcal{K}$ can be encoded into a *DL-Lite*-LTL formula $\phi_{\mathcal{K}}$ of size quadratic in the size of $\mathcal{K}$ that is valid in exactly the models of $\mathcal{K}$. Thus, the satisfiability of a *DL-Lite*-LTL formula ϕ in a model of $\mathcal{K}$ is equivalent to the satisfiability of $\phi \wedge \phi_{\mathcal{K}}$, and similarly, ϕ is valid in all models of $\mathcal{K}$ iff $\neg \phi_{\mathcal{K}} \vee \phi$ is valid.

3 Canonical Models Revisited

As a preliminary step to solving the satisfiability problem for arbitrary *DL-Lite*-LTL formulae, we consider the special case of deciding the consistency of a *DL-*

Lite-ontology. For this, it suffices to check whether the *canonical interpretation* (similar to the one constructed in [KLT$^+$10]) is a model of a given ontology. We show that this is true even in the presence of negated assertions. The canonical interpretation for a *DL-Lite*-ontology is constructed by introducing individuals c_R, $R \in \mathsf{N}_\mathsf{R}^-$, to witness relevant existential restrictions from the ontology.

Definition 3.1. *Let $\mathcal{O} = (\mathcal{T}, \mathcal{A})$ be a DL-Lite-ontology, $R, R' \in \mathsf{N}_\mathsf{R}^-(\mathcal{O})$, and $a \in \mathsf{N}_\mathsf{I}(\mathcal{A})$. We denote by $a \rightsquigarrow c_R$ the fact that $\mathcal{O} \models \exists R(a)$. We further write $c_R \rightsquigarrow c_{R'}$ if $\mathcal{T} \models \exists R^- \sqsubseteq \exists R'$ and $R^- \neq R'$. The role R is called* generating *in $\mathcal{O}$ if there exist $b \in \mathsf{N}_\mathsf{I}(\mathcal{A})$ and $R_1, ..., R_n = R$, $R_i \in \mathsf{N}_\mathsf{R}^-(\mathcal{O})$, such that $b \rightsquigarrow c_{R_1} \rightsquigarrow \cdots \rightsquigarrow c_{R_n}$.*

The canonical interpretation $\mathcal{I}_\mathcal{O}$ for $\mathcal{O}$ is defined as follows, for all $a \in \mathsf{N}_\mathsf{I}(\mathcal{A})$, $A \in \mathsf{N}_\mathsf{C}$, and $R \in \mathsf{N}_\mathsf{R}$:

$$\begin{aligned}
\Delta^{\mathcal{I}_\mathcal{O}} &:= \mathsf{N}_\mathsf{I}(\mathcal{A}) \cup \{c_R \mid R \in \mathsf{N}_\mathsf{R}^- \text{ is generating in } \mathcal{O}\}, \\
a^{\mathcal{I}_\mathcal{O}} &:= a, \\
A^{\mathcal{I}_\mathcal{O}} &:= \{a \in \mathsf{N}_\mathsf{I}(\mathcal{A}) \mid \mathcal{O} \models A(a)\} \cup \{c_R \in \Delta^{\mathcal{I}_\mathcal{O}} \mid \mathcal{T} \models \exists R^- \sqsubseteq A\}, \text{ and} \\
R^{\mathcal{I}_\mathcal{O}} &:= \{(a,b) \mid R(a,b) \in \mathcal{A}\} \cup \{(x, c_R) \mid x \in \Delta^{\mathcal{I}_\mathcal{O}},\ x \rightsquigarrow c_R\} \cup \\
&\qquad \{(c_{R^-}, x) \mid x \in \Delta^{\mathcal{I}_\mathcal{O}},\ x \rightsquigarrow c_{R^-}\}.
\end{aligned}$$

The above definition differs from that in [KLT$^+$10] in that the latter restricts the definition of $a \rightsquigarrow c_R$ to those cases where the ABox $\mathcal{A}$ does not already contain an assertion of the form $R(a, b)$.

If $\mathcal{O}$ is inconsistent, then it is obvious that $\mathcal{I}_\mathcal{O}$ cannot be a model of $\mathcal{O}$. The converse of this statement is a little harder to show. Lemma 3.2 states an even more general result that is needed later in the report.

Lemma 3.2. *Let $\mathcal{O} = (\mathcal{T}, \mathcal{A})$ be a DL-Lite-ontology and $\mathcal{A}^\neg$ a finite set of negated assertions formulated over $\mathsf{N}_\mathsf{C}(\mathcal{O})$, $\mathsf{N}_\mathsf{R}^-(\mathcal{O})$, and $\mathsf{N}_\mathsf{I}(\mathcal{O})$. If $(\mathcal{T}, \mathcal{A} \cup \mathcal{A}^\neg)$ is consistent, then $\mathcal{I}_\mathcal{O}$ is a model of this ontology.*

Proof. By the definition of $\mathcal{I}_\mathcal{O}$, the latter is clearly a model of all positive assertions in $\mathcal{A}$. For the remaining axioms, we first prove the following claim:

$$\text{For all } B \in \mathsf{BC}(\mathcal{O}) \text{ and } a \in \mathsf{N}_\mathsf{I}(\mathcal{O}), \text{ we have } \mathcal{O} \models B(a) \text{ iff } a \in B^{\mathcal{I}_\mathcal{O}}. \tag{1}$$

If B is a concept name, the definition of $\mathcal{I}_\mathcal{O}$ directly yields the claim. Otherwise, B is of the form $\exists R$ for some $R \in \mathsf{N}_\mathsf{R}^-(\mathcal{O})$. By the definition of $\mathcal{I}_\mathcal{O}$, we then have $a \in (\exists R)^{\mathcal{I}_\mathcal{O}}$ iff either (i) $R(a, b) \in \mathcal{A}$ for some $b \in \mathsf{N}_\mathsf{I}(\mathcal{A})$, or (ii) $a \rightsquigarrow c_R$. But (i) implies (ii) since if $R(a, b) \in \mathcal{A}$, then a always has an R-successor. We conclude that $a \in (\exists R)^{\mathcal{I}_\mathcal{O}}$ iff $\mathcal{O} \models \exists R(a)$, which completes the proof of (1).

Consider now any negated concept assertion $\neg B(a) \in \mathcal{A} \cup \mathcal{A}^\neg$. Since the ontology $(\mathcal{T}, \mathcal{A} \cup \mathcal{A}^\neg)$ is consistent, it has a model $\mathcal{I}$ that in particular satisfies $a^\mathcal{I} \notin B^\mathcal{I}$.

This implies that $\mathcal{O} \not\models B(a)$, and thus $a \notin B^{\mathcal{I}_\mathcal{O}}$ by (1), which shows $\mathcal{I}_\mathcal{O} \models \neg B(a)$. Similarly, for every negated role assertion $\neg R(a,b) \in \mathcal{A} \cup \mathcal{A}^\neg$ we know that $R(a,b) \notin \mathcal{A}$ by our assumption that $(\mathcal{T}, \mathcal{A} \cup \mathcal{A}^\neg)$ is consistent. The definition of $\mathcal{I}_\mathcal{O}$ thus yields that $(a,b) \notin R^{\mathcal{I}_\mathcal{O}}$.

It remains to show that $\mathcal{I}_\mathcal{O}$ is also a model of all CIs $B \sqsubseteq C$ in $\mathcal{T}$. For any $a \in B^{\mathcal{I}_\mathcal{O}} \cap \mathsf{N_I}(\mathcal{A})$, we obtain $\mathcal{O} \models B(a)$ from (1). Since $\mathcal{O} \models B \sqsubseteq C$, this implies that $\mathcal{O} \models C(a)$, and thus $a \in C^{\mathcal{I}_\mathcal{O}}$, again by (1).

Consider now any unnamed domain element $c_R \in \Delta^{\mathcal{I}_\mathcal{O}}$ where R is a generating role in $\mathcal{O}$. We first show the following:

$$\text{For all } B' \in \mathsf{BC}(\mathcal{O}) \text{ with } c_R \in B'^{\mathcal{I}_\mathcal{O}}, \text{ we have } \mathcal{T} \models \exists R^- \sqsubseteq B'. \tag{2}$$

If B' is a concept name, then this follows directly from the definition of $\mathcal{I}_\mathcal{O}$. Otherwise, B' is of the form $\exists R'$ for some $R' \in \mathsf{N}_\mathsf{R}^-(\mathcal{O})$. From $c_R \in (\exists R')^{\mathcal{I}_\mathcal{O}}$, it follows that either (i) $c_R \rightsquigarrow c_{R'}$, or (ii) $R = R'^-$ and $x \rightsquigarrow c_R$ for some $x \in \Delta^{\mathcal{I}_\mathcal{O}}$. In case (i), we have $\mathcal{T} \models \exists R^- \sqsubseteq \exists R'$ by definition. In case (ii), this inclusion is trivial since we then have $\exists R^- = \exists R'$. This completes the proof of (2).

If $c_R \in B^{\mathcal{I}_\mathcal{O}}$, then from (2) and $B \sqsubseteq C \in \mathcal{T}$ it follows that $\mathcal{T} \models \exists R^- \sqsubseteq C$. To conclude the proof, we show that this implies $c_R \in C^{\mathcal{I}_\mathcal{O}}$.

- If C is a concept name, this follows directly from the definition of $\mathcal{I}_\mathcal{O}$.
- If C is of the form $\exists R'$ for a role $R' \in \mathsf{N}_\mathsf{R}^-(\mathcal{O})$, we know from $\mathcal{T} \models \exists R^- \sqsubseteq \exists R'$ and the fact that R is generating in $\mathcal{O}$ that R' is also generating in $\mathcal{O}$ and $c_R \rightsquigarrow c_{R'}$. Hence, we get $(c_R, c_{R'}) \in R'^{\mathcal{I}_\mathcal{O}}$ and thus $c_R \in (\exists R')^{\mathcal{I}_\mathcal{O}}$.
- If $C = \neg B'$ for a basic concept B', assume by contradiction that $c_R \in B'^{\mathcal{I}_\mathcal{O}}$. By (2), we obtain $\mathcal{T} \models \exists R^- \sqsubseteq B'$. But since we also have $\mathcal{T} \models \exists R^- \sqsubseteq \neg B'$, we conclude $\mathcal{T} \models \exists R^- \sqsubseteq \neg \exists R^-$, i.e. $\exists R^-$ must be empty. But this contradicts the fact that $c_R \in (\exists R^-)^{\mathcal{I}_\mathcal{O}}$ since R is generating in $\mathcal{O}$.

□

Thus, negated assertions are irrelevant for the construction of the canonical model, as long as they do not cause the ontology to become inconsistent.

4 Satisfiability in *DL-Lite*-LTL

We first show PSPACE-hardness, which is a straightforward consequence of the complexity of the satisfiability problem in propositional LTL.

Lemma 4.1. *Satisfiability in DL-Lite-LTL is* PSPACE*-hard even if no rigid names are available.*

Proof. We reduce the satisfiability problem of propositional LTL formulae, which is PSpace-complete [SC85]. Let ψ be a propositional LTL formula over the propositional variables $p_1, \ldots, p_n$. The formula ϕ is obtained from ψ by replacing every propositional variable p_i with $A_i(a)$ for i, $1 \leq i \leq n$, where a is an individual name and $A_1, \ldots, A_n$ are n distinct concept names. Obviously, ϕ is a *DL-Lite*-LTL formula. It is easy to see that every propositional LTL structure satisfying ψ induces a *DL-Lite*-LTL structure satisfying ϕ, and vice versa. $\square$

The proof of the upper bound follows the basic approach from [BGL12], but additionally utilizes the characteristics of *DL-Lite*. In the following, let ϕ be a *DL-Lite*-LTL formula to be tested for satisfiability. The *propositional abstraction* ϕ^{p} of ϕ is created by replacing each axiom by a propositional variable such that there is a 1–1 relationship between the axioms $\alpha_1, \ldots, \alpha_n$ occurring in ϕ and the propositional variables $p_1, \ldots, p_n$ used for the abstraction. In what follows, we assume that p_i was used to replace α_i for all i, $1 \leq i \leq n$. For a subset $X \subseteq \{p_1, \ldots, p_n\}$, we denote by $\overline{X}$ its complement $\{p_1, \ldots, p_n\} \setminus X$.

We now consider sets of the form $\mathcal{S} \subseteq 2^{\{p_1,\ldots,p_n\}}$ that constrain the types of interpretations allowed to occur in the model of ϕ. Every such set induces the following LTL formula:

$$\phi^{\mathsf{p}}_{\mathcal{S}} = \phi^{\mathsf{p}} \wedge \Box \left(\bigvee_{X \in \mathcal{S}} \left(\bigwedge_{p \in X} p \wedge \bigwedge_{p \in \overline{X}} \neg p \right) \right)$$

If ϕ is satisfiable in a *DL-Lite*-LTL structure $\mathfrak{I} = (\mathcal{I}_i)_{i \geq 0}$, then there is a set $\mathcal{S} \subseteq 2^{\{p_1,\ldots,p_n\}}$ such that $\phi^{\mathsf{p}}_{\mathcal{S}}$ is satisfiable in a propositional LTL structure. To see this, for each $i \geq 0$ we define

$$X_i := \{p_j \mid 1 \leq j \leq n,\ \mathcal{I}_i \models \alpha_j\}$$

and set $\mathcal{S} := \{X_i \mid i \geq 0\}$. We say that $\mathcal{S}$ is *induced* by the *DL-Lite*-LTL structure $\mathfrak{I}$. The fact that $\mathfrak{I}$ satisfies ϕ implies that its *propositional abstraction* satisfies $\phi^{\mathsf{p}}_{\mathcal{S}}$, where the propositional abstraction $\mathfrak{I}^{\mathsf{p}} = (w_i)_{i \geq 0}$ of $\mathfrak{I}$ is defined such that w_i contains p_j iff $\mathcal{I}_i$ satisfies α_j.

However, guessing such a set $\mathcal{S}$ and then testing whether the induced formula $\phi^{\mathsf{p}}_{\mathcal{S}}$ is satisfiable is not sufficient for checking satisfiability of ϕ. It must also be checked whether $\mathcal{S}$ can indeed be induced by some *DL-Lite*-LTL structure that also respects the rigid concept and role names.

Assume for now that a set $\mathcal{S} = \{X_1, \ldots, X_k\} \subseteq 2^{\{p_1,\ldots,p_n\}}$ is given. We introduce the set $\mathsf{N}^{\mathsf{n}}_{\mathsf{I}} := \{a^i_j \mid p_j \in \overline{X_i},\ \alpha_j \text{ is a CI}\}$ of auxiliary individual names that do not occur in ϕ, and define the ontologies $(\mathcal{T}_i, \mathcal{A}_i)$, $1 \leq i \leq k$, where

$$\begin{aligned}
\mathcal{T}_i &:= \{\alpha_j \mid p_j \in X_i,\ \alpha_j \text{ is a CI}\} \text{ and} \\
\mathcal{A}_i &:= \{\alpha_j \mid p_j \in X_i,\ \alpha_j \text{ is an assertion}\} \cup \\
&\quad\ \{\neg\alpha_j \mid p_j \in \overline{X_i},\ \alpha_j \text{ is an assertion}\} \cup \\
&\quad\ \{B(a_j^i), \neg C(a_j^i) \mid p_j \in \overline{X_i},\ \alpha_j = B \sqsubseteq C\}.
\end{aligned}$$

Definition 4.2. *A set $\mathcal{S} = \{X_1, \dots, X_k\} \subseteq 2^{\{p_1,\dots,p_n\}}$ is* r-satisfiable *if there are interpretations $\mathcal{J}_1, \dots, \mathcal{J}_k$ such that*

- *each $\mathcal{J}_i$, $1 \leq i \leq k$, is a model of $(\mathcal{T}_i, \mathcal{A}_i)$; and*
- *they share the same domain and respect rigid names (cf. Definition 2.4).*

The following basic reduction is similar to the one used in [BGL12, Lemma 4.3], except for the introduction of the explicit counterexamples a_j^i for the CIs that should not be satisfied.

Lemma 4.3. *The DL-Lite-LTL formula ϕ is satisfiable iff there is an r-satisfiable set $\mathcal{S} = \{X_1, \dots, X_k\} \subseteq 2^{\{p_1,\dots,p_n\}}$ such that the propositional LTL formula $\phi_{\mathcal{S}}^{\mathsf{p}}$ is satisfiable.*

Proof. As described above, every *DL-Lite*-LTL structure $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$ satisfying ϕ induces a set $\mathcal{S} = \{X_1, \dots, X_k\}$ such that $\phi_{\mathcal{S}}^{\mathsf{p}}$ is satisfiable. In this construction, one can find a mapping $\iota\colon \{1, \dots, k\} \to \mathbb{N}$ such that the equality $X_i = \{p_j \mid 1 \leq j \leq n,\ \mathcal{I}_{\iota(i)} \models \alpha_j\}$ holds for all i between 1 and k. By assumption, the interpretations $\mathcal{J}_i := \mathcal{I}_{\iota(i)}$, $1 \leq i \leq k$, have a common domain Δ, respect rigid names, and satisfy the UNA w.r.t. $\mathsf{N_I}(\phi)$. Furthermore, by construction each $\mathcal{J}_i$ already satisfies $\mathcal{T}_i$ and those elements of $\mathcal{A}_i$ induced by *assertions* α_j. To satisfy the remaining assertions in $\mathcal{A}_i$ (i.e. those involving the new names from $\mathsf{N_I^n}$), we extend $\mathcal{J}_1, \dots, \mathcal{J}_k$ simultaneously to new interpretations $\mathcal{J}_1', \dots, \mathcal{J}_k'$ over a larger domain.

Observe that for all $a_j^i \in \mathsf{N_I^n}$ we have $p_j \in \overline{X_i}$, and hence $\mathcal{J}_i = \mathcal{I}_{\iota(i)} \not\models \alpha_j$ by the construction of ι above. Thus, there is a mapping $f\colon \mathsf{N_I^n} \to \Delta$ such that for all $a_j^i \in \mathsf{N_I^n}$ with $\alpha_j = B \sqsubseteq C$ we have $f(a_j^i) \in B^{\mathcal{J}_i} \setminus C^{\mathcal{J}_i}$. In other words, each $\mathcal{J}_i$ already contains domain elements $f(a_j^i)$ required to refute the CIs α_j for which $p_j \in \overline{X_i}$. To ensure that the UNA remains satisfied for the new individual names a_j^i, we have to copy these domain elements. For convenience, we extend the mapping f to Δ by setting $f(x) := x$ for all $x \in \Delta$.

We define $\Delta' := \Delta \cup \mathsf{N_I^n}$ as the common domain of the new interpretations $\mathcal{J}_1', \dots, \mathcal{J}_k'$, where we assume that the new domain elements a_j^i do not occur in Δ. We define $(a_j^{i'})^{\mathcal{J}_i'} := a_j^{i'}$ for all $a_j^{i'} \in \mathsf{N_I^n}$, and set $a^{\mathcal{J}_i'} := a^{\mathcal{J}_i}$ for all other individual names a. For all concept names A and role names r, we set

$$A^{\mathcal{J}_i'} := \{x \in \Delta' \mid f(x) \in A^{\mathcal{J}_i}\} \text{ and}$$
$$r^{\mathcal{J}_i'} := \{(x,y) \in \Delta' \times \Delta' \mid (f(x), f(y)) \in r^{\mathcal{J}_i}\}.$$

It is easy to see that for all basic concepts $B \in \mathsf{BC}(\phi)$ we have $x \in B^{\mathcal{J}_i'}$ iff $f(x) \in B^{\mathcal{J}_i}$. Thus, each $\mathcal{J}_i'$ still satisfies the assertions of $\mathcal{A}_i$ not involving the new names. Furthermore, all CIs in $\mathcal{T}_i$ remain satisfied since any counterexample in $\mathcal{J}_i'$ would immediately yield a counterexample in $\mathcal{J}_i$ (observe that f is surjective). Additionally, this construction ensures that the assertions involving $\mathsf{N}_\mathsf{I}^\mathsf{n}$ are now satisfied. We thus obtain models $\mathcal{J}_i'$ of $(\mathcal{T}_i, \mathcal{A}_i)$ that all share the same domain Δ', respect rigid names, and satisfy the UNA w.r.t. $\mathsf{N}_\mathsf{I}(\phi) \cup \mathsf{N}_\mathsf{I}^\mathsf{n} \supseteq \mathsf{N}_\mathsf{I}(\mathcal{A}_i)$.

Conversely, assume that there are a set $\mathcal{S} = \{X_1, \dots, X_k\} \subseteq 2^{\{p_1,\dots,p_n\}}$ and interpretations $\mathcal{J}_1, \dots, \mathcal{J}_k$ sharing the same domain and respecting rigid names such that each $\mathcal{J}_i$ is a model of $(\mathcal{T}_i, \mathcal{A}_i)$ and $\phi^\mathsf{p}_\mathcal{S}$ is satisfiable. Then, there must be an LTL structure $\mathfrak{J} = (w_i)_{i\geq 0}$ with $\mathfrak{J}, 0 \models \phi^\mathsf{p}_\mathcal{S}$. By construction, there is a mapping $\iota \colon \mathbb{N} \to \{1, \dots, k\}$ such that $w_i = X_{\iota(i)}$ holds for all $i \geq 0$. We construct the *DL-Lite*-LTL structure $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$ by setting $\mathcal{I}_i := \mathcal{J}_{\iota(i)}$ for each $i \geq 0$. These interpretations have a common domain and respect rigid names. By construction, each $\mathcal{I}_i$ satisfies the axioms specified by the propositional variables in $X_{\iota(i)} = w_i$ and refutes the axioms corresponding to $\{p_1, \dots, p_n\} \setminus w_i$. Since $\mathfrak{J}, 0 \models \phi^\mathsf{p}$, this means that $\mathfrak{I}, 0 \models \phi$ (see Definition 2.4). $\square$

Obviously, we can guess a single set $X_i \subseteq \{p_1, \dots, p_n\}$ within PSpace. However, the propositional LTL formula $\phi^\mathsf{p}_\mathcal{S}$ is of size exponential in the size of ϕ. Thus, a direct application of the PSpace decision procedure for satisfiability in propositional LTL would only yield an ExpSpace upper bound. Also, keeping $\mathcal{S}$ in memory already requires exponential space. The latter problem is addressed in the following by guessing polynomially many additional axioms that allow us to separate the r-satisfiability test for $\mathcal{S}$ into independent consistency tests for each $(\mathcal{T}_i, \mathcal{A}_i)$.

Given three sets $X_I \subseteq \mathsf{N}_\mathsf{I}$, $X_C \subseteq \mathsf{N}_\mathsf{C}$, and $X_R \subseteq \mathsf{N}_\mathsf{R}^-$, an *ABox type* $\approx$ for X_I, X_C, and X_R is a subset of the closure under negation of

$$\{A(a), R(a,b), (\exists R)(a) \mid a, b \in X_I,\ A \in X_C,\ R \in X_R\},$$

with the property that for each of these assertions α we have $\alpha \in \approx$ iff $\neg\alpha \notin \approx$. The additional information we guess is divided in two parts:

- A binary relation $\mathsf{Sub}_\mathsf{R} \subseteq \mathsf{BC}_\mathsf{R}(\phi) \times \mathsf{BC}_\mathsf{R}^\neg(\phi)$ that specifies which rigid CIs hold in the models of $(\mathcal{T}_i, \mathcal{A}_i)$.

  We denote by $\mathsf{N}_\mathsf{I}^\mathsf{R} := \{a_{B,C} \mid B \in \mathsf{BC}_\mathsf{R}(\phi),\ C \in \mathsf{BC}_\mathsf{R}^\neg(\phi),\ (B,C) \notin \mathsf{Sub}_\mathsf{R}\}$ a set of fresh individual names that will be used to ensure that certain rigid CIs do *not* hold.

- An ABox type for $\mathsf{N_I}(\phi) \cup \mathsf{N_I^R}$, $\mathsf{N_{RC}}(\phi)$, and $\mathsf{N_{RR}^-}(\phi)$ that completely describes the behavior of all named individuals on the relevant concepts and roles.

This is formalized in the following definition.

Definition 4.4. *An ontology* $\mathcal{O}_\mathsf{R} = (\mathcal{T}_\mathsf{R}, \mathcal{A}_\mathsf{R})$ *is called* r-complete *for* ϕ *if there are a binary relation* $\mathsf{Sub_R} \subseteq \mathsf{BC_R}(\phi) \times \mathsf{BC_R^-}(\phi)$ *and an ABox type* $\approx_\mathsf{R}$ *for* $\mathsf{N_I}(\phi) \cup \mathsf{N_I^R}$, $\mathsf{N_{RC}}(\phi)$, *and* $\mathsf{N_{RR}^-}(\phi)$ *such that*

- $\mathcal{T}_\mathsf{R} := \{B \sqsubseteq C \mid (B, C) \in \mathsf{Sub_R}\}$ *and*
- $\mathcal{A}_\mathsf{R}$ *is the union of* $\{B(a_{B,C}), \neg C(a_{B,C}) \mid a_{B,C} \in \mathsf{N_I^R}\}$ *and* $\approx_\mathsf{R}$.

The idea is that the additional information in $\mathcal{O}_\mathsf{R}$ is enough to test r-satisfiability of $\mathcal{S}$ using independent consistency tests for $\mathcal{O}_\mathsf{R}^i := (\mathcal{T}_i \cup \mathcal{T}_\mathsf{R}, \mathcal{A}_i \cup \mathcal{A}_\mathsf{R})$, $1 \le i \le k$.

Lemma 4.5. *If* $\mathcal{S}$ *is r-satisfiable, then there is an r-complete ontology* $\mathcal{O}_\mathsf{R}$ *for* ϕ *such that all* $\mathcal{O}_\mathsf{R}^i$, $1 \le i \le k$, *are consistent.*

Proof. Let $\mathcal{J}_1, \dots, \mathcal{J}_k$ be the interpretations that exist by the r-satisfiability of $\mathcal{S}$. We construct the r-complete ontology $\mathcal{O}_\mathsf{R}$ by defining

$$\mathsf{Sub_R} := \{(B, C) \in \mathsf{BC_R}(\phi) \times \mathsf{BC_R^-}(\phi) \mid \mathcal{J}_1 \models B \sqsubseteq C\} \text{ and}$$
$$\approx_\mathsf{R} := \{\alpha \mid \alpha \in \mathcal{A}_\phi,\ \mathcal{J}_1 \models \alpha\} \cup \{\neg\alpha \mid \alpha \in \mathcal{A}_\phi,\ \mathcal{J}_1 \not\models \alpha\},$$

where $\mathcal{A}_\phi$ denotes the set of all assertions over $\mathsf{N_I}(\phi) \cup \mathsf{N_I^R}$, $\mathsf{N_{RC}}(\phi)$, and $\mathsf{N_{RR}^-}(\phi)$.

Note that every rigid axiom is satisfied by $\mathcal{J}_1$ iff it is satisfied by $\mathcal{J}_2, \dots, \mathcal{J}_k$ since they agree on the interpretation of the rigid names. Using the same technique as in the proof of Lemma 4.3 to copy the counterexamples for the rigid CIs that do not hold, we can thus extend the interpretations $\mathcal{J}_i$ to models of the induced ontologies $\mathcal{O}_\mathsf{R}^i$ for all i, $1 \le i \le k$. □

In the remainder of this section, we prove the converse of this lemma. Let $\mathcal{O}_\mathsf{R} = (\mathcal{T}_\mathsf{R}, \mathcal{A}_\mathsf{R})$ be an r-complete ontology with a relation $\mathsf{Sub_R}$ and an ABox type $\approx_\mathsf{R}$ and $\mathcal{I}_i$ models of $\mathcal{O}_\mathsf{R}^i$ for all i, $1 \le i \le k$. By Lemma 3.2, we can assume these to be the canonical models $\mathcal{I}_{\mathcal{O}_\mathsf{R}^i}$. To distinguish the unnamed elements, we write $c_{R,i}$ for the element c_R in the domain of $\mathcal{I}_{\mathcal{O}_\mathsf{R}^i}$.

Thus, the domain of each $\mathcal{I}_i = \mathcal{I}_{\mathcal{O}_\mathsf{R}^i}$ is $\Delta^{\mathcal{I}_i} = \mathsf{N_I}(\phi) \cup \mathsf{N_I^R} \cup \Delta_\mathsf{u}^{\mathcal{I}_i}$, where

$$\Delta_\mathsf{u}^{\mathcal{I}_i} := \{a_j^i \in \mathsf{N_I^n}\} \cup \{c_{R,i} \mid R \in \mathsf{N_R^-},\ R \text{ is generating in } \mathcal{O}_\mathsf{R}^i\}$$

contains the domain elements unique to this interpretation. Apart from the unnamed domain elements, the domains also differ in the individual names a_j^i in $\mathsf{N_I^n}$ that were introduced to provide counterexamples for CIs α_j with $p_j \in \overline{X_i}$.

The following is a first easy observation about the rigid CIs that hold in these interpretations.

Lemma 4.6. *For all $B \in \mathsf{BC_R}(\phi)$, $C \in \mathsf{BC_R^{\neg}}(\phi)$, and $1 \le i \le k$, we have $\mathcal{I}_i \models B \sqsubseteq C$ iff $(B, C) \in \mathsf{Sub_R}$.*

Proof. If $(B, C) \in \mathsf{Sub_R}$, then $B \sqsubseteq C \in \mathcal{T}_\mathsf{R}$, and thus our assumption that $\mathcal{I}_i \models \mathcal{T}_\mathsf{R}$ yields $\mathcal{I}_i \models B \sqsubseteq C$. Conversely, if $(B, C) \notin \mathsf{Sub_R}$, then by $\mathcal{I}_i \models \mathcal{A}_\mathsf{R}$ we get $a_{B,C} \in B^{\mathcal{I}_i} \setminus C^{\mathcal{I}_i}$, and thus $a_{B,C}$ is a counterexample to the rigid CI $B \sqsubseteq C$ in $\mathcal{I}_i$, which means that $\mathcal{I}_i \not\models B \sqsubseteq C$. □

Consequently, a rigid CI either holds in all $\mathcal{I}_i$, $1 \le i \le k$, or in none of them.

We now construct the models $\mathcal{J}_i$ of $(\mathcal{T}_i, \mathcal{A}_i)$ required for the r-satisfiability of $\mathcal{S}$ by joining the domains of the interpretations $\mathcal{I}_i$ and ensuring that they interpret all rigid names in the same way. We now use the common domain $\Delta := \mathsf{N_I}(\phi) \cup \mathsf{N_I^R} \cup \bigcup_{i=1}^{k} \Delta_\mathsf{u}^{\mathcal{I}_i}$ and define, for all i, $1 \le i \le k$, the interpretations $\mathcal{J}_i$ as follows:

- For all $a \in \mathsf{N_I}(\phi) \cup \mathsf{N_I^R} \cup \mathsf{N_I^n}$, we set $a^{\mathcal{J}_i} := a$.
- For all rigid concept names A, we define $A^{\mathcal{J}_i} := \bigcup_{j=1}^{k} A^{\mathcal{I}_j}$.
- For all flexible concept names A, we define

$$A^{\mathcal{J}_i} := A^{\mathcal{I}_i} \cup \bigcup_{j=1}^{k} \bigcup_{\substack{B \in \mathsf{BC_R}(\phi) \\ \mathcal{I}_i \models B \sqsubseteq A}} B^{\mathcal{I}_j}.$$

- For all rigid role names R, we define $R^{\mathcal{J}_i} := \bigcup_{j=1}^{k} R^{\mathcal{I}_j}$.
- For all flexible role names R, we define

$$R^{\mathcal{J}_i} := R^{\mathcal{I}_i} \cup \bigcup_{j=1}^{k} \left(\bigcup_{\substack{B \in \mathsf{BC_R}(\phi) \\ \mathcal{I}_i \models B \sqsubseteq \exists R}} B^{\mathcal{I}_j} \times \{c_{R,i}\} \cup \bigcup_{\substack{B \in \mathsf{BC_R}(\phi) \\ \mathcal{I}_i \models B \sqsubseteq \exists R^-}} \{c_{R^-,i}\} \times B^{\mathcal{I}_j} \right).$$

To prove that this last definition is well-defined, we have to verify that $c_{R,i}$ is actually an element of $\Delta_\mathsf{u}^{\mathcal{I}_i}$ whenever $B^{\mathcal{I}_j}$ is non-empty for some $B \in \mathsf{BC_R}(\phi)$ with $\mathcal{I}_i \models B \sqsubseteq \exists R$. In this case, we have $\mathcal{I}_j \not\models B \sqsubseteq \neg B$, and thus $(B, \neg B) \notin \mathsf{Sub_R}$ by Lemma 4.6, which implies that $B(a_{B,\neg B}) \in \mathcal{A}_\mathsf{R}$. Since $\mathcal{I}_i \models B \sqsubseteq \exists R$ and $\mathcal{I}_i \models \mathcal{A}_\mathsf{R}$, this implies that $\mathcal{I}_i = \mathcal{I}_{\mathcal{O}_\mathsf{R}^i}$ cannot be a model of $\neg\exists R(a_{B,\neg B})$. By Lemma 3.2, $\mathcal{O}_\mathsf{R}^i$ together with $\neg\exists R(a_{B,\neg B})$ is inconsistent, and thus $\mathcal{O}_\mathsf{R}^i \models \exists R(a_{B,\neg B})$. This shows that R is generating in $\mathcal{O}_\mathsf{R}^i$, and hence $c_{R,i} \in \Delta_\mathsf{u}^{\mathcal{I}_i}$.

We have thus constructed interpretations $\mathcal{J}_1, \ldots, \mathcal{J}_k$ that have the same domain, respect rigid names, and satisfy the UNA for all relevant individual names. It remains to show that each $\mathcal{J}_i$ is still a model of $(\mathcal{T}_i, \mathcal{A}_i)$. We first prove a basic connection between the interpretations $\mathcal{J}_i$ and $\mathcal{I}_i$.

Lemma 4.7. *For all i, $1 \leq i \leq k$ and $B \in \mathsf{BC}(\phi)$, the following hold:*

a) *for every $a \in \mathsf{N_I}(\phi) \cup \mathsf{N_I^R}$, we have $a \in B^{\mathcal{J}_i}$ iff $a \in B^{\mathcal{I}_i}$;*

b) *if B is rigid, then for every $x \in \Delta_{\mathsf{u}}^{\mathcal{I}_j}$, $1 \leq j \leq k$, we have $x \in B^{\mathcal{J}_i}$ iff $x \in B^{\mathcal{I}_j}$; and*

c) *if B is flexible, then for every $x \in \Delta_{\mathsf{u}}^{\mathcal{I}_j}$, $1 \leq j \leq k$, we have $x \in B^{\mathcal{J}_i}$ iff*

   - *$i = j$ and $x \in B^{\mathcal{I}_i}$, or*
   - *there is a $B' \in \mathsf{BC_R}(\phi)$ with $x \in (B')^{\mathcal{I}_j}$ and $\mathcal{I}_i \models B' \sqsubseteq B$.*

Proof. For a), consider first the case that $B \in \mathsf{N_{RC}}(\phi)$. Recall that all $\mathcal{I}_j$, $1 \leq j \leq k$, agree on the interpretation of all rigid concept names on $\mathsf{N_I}(\phi) \cup \mathsf{N_I^R}$ since they satisfy $\approx_{\mathsf{R}}$. Thus, we have have $B^{\mathcal{J}_i} \cap (\mathsf{N_I}(\phi) \cup \mathsf{N_I^R}) = B^{\mathcal{I}_i} \cap (\mathsf{N_I}(\phi) \cup \mathsf{N_I^R})$.

If B is of the form $\exists R$ for a rigid role $R \in \mathsf{N_{RR}^-}(\phi)$, then $a \in B^{\mathcal{I}_i}$ clearly implies $a \in B^{\mathcal{J}_i}$ since $R^{\mathcal{I}_i}$ is contained in $R^{\mathcal{J}_i}$. On the other hand, if $a \in B^{\mathcal{J}_i}$, then by the definition of $\mathcal{J}_i$, a must have an R-successor in at least one $\mathcal{I}_j$, $1 \leq j \leq k$. Since $\mathcal{I}_j \models \approx_{\mathsf{R}}$, we cannot have $\neg\exists R(a) \in \approx_{\mathsf{R}}$, and thus we must have $\exists R(a) \in \approx_{\mathsf{R}}$ since $\approx_{\mathsf{R}}$ is an ABox type. Since $\mathcal{I}_i \models \approx_{\mathsf{R}}$, this implies that $a \in (\exists R)^{\mathcal{I}_i} = B^{\mathcal{I}_i}$.

Consider now any flexible basic concept B. By the definition of $\mathcal{J}_i$ on the flexible names, we have $a \in B^{\mathcal{J}_i}$ iff either (i) $a \in B^{\mathcal{I}_i}$, or (ii) $a \in (B')^{\mathcal{I}_j}$ for some j, $1 \leq j \leq k$, and $B' \in \mathsf{BC_R}(\phi)$ with $\mathcal{I}_i \models B' \sqsubseteq B$. But (ii) implies (i) since $a \in (B')^{\mathcal{I}_j}$ yields $B'(a) \in \approx_{\mathsf{R}}$, and thus $a \in (B')^{\mathcal{I}_i} \subseteq (\exists R)^{\mathcal{I}_i}$, as above. We conclude that $a \in B^{\mathcal{J}_i}$ iff $a \in B^{\mathcal{I}_i}$, as desired.

For b), let B be rigid and $x \in \Delta_{\mathsf{u}}^{\mathcal{I}_j}$. Since x does not belong to any $\Delta_{\mathsf{u}}^{\mathcal{I}_{j'}}$ with $j' \neq j$, by the definition of $\mathcal{J}_i$ on the rigid concept and role names, we immediately get $x \in B^{\mathcal{J}_i}$ iff $x \in B^{\mathcal{I}_j}$.

For c), we consider first the case that $B \in \mathsf{N_C}(\phi)$ is a flexible concept name. Then the definition of $\mathcal{J}_i$ directly yields $x \in B^{\mathcal{J}_i}$ iff $i = j$ and $x \in B^{\mathcal{I}_i}$ or $x \in (B')^{\mathcal{I}_j}$ for some $B' \in \mathsf{BC_R}(\phi)$ with $\mathcal{I}_i \models B' \sqsubseteq B$. If B is of the form $\exists R$ for a flexible role $R \in \mathsf{N_R^-}(\phi)$, then by the definition of $\mathcal{J}_i$ we have $x \in B^{\mathcal{J}_i}$ iff one of the following alternatives is satisfied:

- $i = j$ and $x \in B^{\mathcal{I}_i}$,
- there is a $B' \in \mathsf{BC_R}(\phi)$ with $x \in (B')^{\mathcal{I}_j}$ and $\mathcal{I}_i \models B' \sqsubseteq B$, or
- x is of the form $c_{R^-,i} \in \Delta_{\mathsf{u}}^{\mathcal{I}_i}$ and there is a $B' \in \mathsf{BC_R}(\phi)$ such that $(B')^{\mathcal{I}_j}$ is not empty and $\mathcal{I}_i \models B' \sqsubseteq \exists R^-$.

But the last case is included in the first one, because then we also have $i = j$ and $x = c_{R^-,i} \in (\exists R)^{\mathcal{I}_i} = B^{\mathcal{I}_i}$ since R^- is generating in $\mathcal{O}_{\mathsf{R}}^i$ (see Definition 3.1). $\square$

In particular, this implies that for all i, $1 \leq i \leq k$, $B \in \mathsf{BC}(\phi)$, and $x \in \Delta^{\mathcal{I}_i}$ we have $x \in B^{\mathcal{J}_i}$ iff $x \in B^{\mathcal{I}_i}$. This means that on the original domain $\Delta^{\mathcal{I}_i}$ the interpretation of the basic concepts does not change.

The next lemmas show that $\mathcal{J}_i$ is in fact as intended. To show that $\mathcal{J}_i$ is a model of $(\mathcal{T}_i, \mathcal{A}_i)$, we first consider the assertions.

Lemma 4.8. *For all i, $1 \leq i \leq k$, $\mathcal{J}_i$ is a model of $\mathcal{A}_i$.*

Proof. Let $C(a)$ be a (negated) basic concept assertion in $\mathcal{A}_i$. If $a \in \mathsf{N_I}(\phi)$, then Lemma 4.7a) yields that $\mathcal{J}_i \models C(a)$ since we have $\mathcal{I}_i \models C(a)$ by assumption. For every (negated) role assertion $R(a, b)$ $(\neg R(a, b))$ in $\mathcal{A}_i$, we know by construction that $a, b \in \mathsf{N_I}(\phi)$. Since all $\mathcal{I}_j$, $1 \leq j \leq k$, satisfy $\approx_\mathsf{R}$, we have $R^{\mathcal{J}_i} \cap (\mathsf{N_I}(\phi) \times \mathsf{N_I}(\phi)) = R^{\mathcal{I}_i} \cap (\mathsf{N_I}(\phi) \times \mathsf{N_I}(\phi))$ by the definition of $\mathcal{J}_i$, regardless of whether R is rigid or not. The claim now follows from the fact that $\mathcal{I}_i$ satisfies $R(a, b)$ $(\neg R(a, b))$.

It remains to consider those concept assertions $C(a)$ in $\mathcal{A}_i$ where $a \in \mathsf{N_I^n} \cap \Delta_\mathsf{u}^{\mathcal{I}_i}$ and $C \in \mathsf{BC}^\neg(\phi)$. But then Lemma 4.7b) and c) yield $\mathcal{J}_i \models C(a)$ since we know that $\mathcal{I}_i \models C(a)$. To see this, note that $i = j$ and a only occurs in the domain of $\mathcal{I}_i$, and thus the second condition of c) is subsumed by the first condition. $\square$

It remains to show that all CIs in $\mathcal{T}_i$ are satisfied by $\mathcal{J}_i$.

Lemma 4.9. *For all i, $1 \leq i \leq k$, $\mathcal{J}_i$ is a model of $\mathcal{T}_i$.*

Proof. Let $B \sqsubseteq C$ be a CI in $\mathcal{T}_i$. By assumption, we know that

$$\mathcal{I}_i \models B \sqsubseteq C. \tag{3}$$

To show that $B^{\mathcal{J}_i} \subseteq C^{\mathcal{J}_i}$, take any $x \in B^{\mathcal{J}_i}$. If $x \in \mathsf{N_I}(\phi) \cup \mathsf{N_I^R}$, then by Lemma 4.7a) we get $x \in B^{\mathcal{I}_i}$. By (3), this implies $x \in C^{\mathcal{I}_i}$, which yields $x \in C^{\mathcal{J}_i}$, again by Lemma 4.7a).

Otherwise, x must be an element of $\Delta_\mathsf{u}^{\mathcal{I}_j}$, for some j, $1 \leq j \leq k$.

In the special case that B is flexible and the first condition of Lemma 4.7c) applies, we have $i = j$ and $x \in B^{\mathcal{I}_i}$. We then obtain from (3) that $x \in C^{\mathcal{I}_i}$. By Lemma 4.7b) and c), on $\Delta^{\mathcal{I}_i}$ the interpretation of all basic concepts is the same under $\mathcal{I}_i$ and $\mathcal{J}_i$, and thus we get $x \in C^{\mathcal{J}_i}$.

In the remaining two cases, namely that (i) B is rigid, or (ii) B is flexible and the second condition of Lemma 4.7c) applies, we first show that there is a $B_1 \in \mathsf{BC_R}(\phi)$ satisfying the following two conditions:

$$x \in B_1^{\mathcal{I}_j} \tag{4}$$

$$\mathcal{I}_i \models B_1 \sqsubseteq C \tag{5}$$

In case (i), we can simply choose $B_1 := B$, which already satisfies both requirements by our assumption that $x \in B^{\mathcal{I}_j}$ and (3). In case (ii), from Lemma 4.7c) we get a $B' \in \mathsf{BC_R}(\phi)$ with $x \in (B')^{\mathcal{I}_j}$ and $\mathcal{I}_i \models B' \sqsubseteq B$. But then (3) implies that also $\mathcal{I}_i \models B' \sqsubseteq C$ holds, and thus we can set $B_1 := B'$.

Given a rigid basic concept B_1 satisfying (4) and (5), we now make a case distinction on the shape of C to show that $x \in C^{\mathcal{J}_i}$.

- If C is also rigid, then Lemma 4.6 and (5) yield that $\mathcal{I}_j \models B_1 \sqsubseteq C$. From (4), we thus get $x \in C^{\mathcal{I}_j}$, and hence $x \in C^{\mathcal{J}_i}$ by Lemma 4.7b).
- If $C \in \mathsf{BC}(\phi)$ is a flexible basic concept, then (4) and (5) yield $x \in C^{\mathcal{J}_i}$ by Lemma 4.7c).
- If C is of the form $\neg B_2$ for a flexible basic concept $B_2 \in \mathsf{BC}(\phi)$, we have to show that $x \notin B_2^{\mathcal{J}_i}$. Assume to the contrary that $x \in B_2^{\mathcal{J}_i}$. Then one of the alternatives of Lemma 4.7c) must hold.

  If $i = j$ and $x \in B_2^{\mathcal{I}_i}$, then (5) and $C = \neg B_2$ yield that $x \notin B_1^{\mathcal{I}_i} = B_1^{\mathcal{I}_j}$, in contradiction to (4).

  Otherwise, there is a $B' \in \mathsf{BC_R}(\phi)$ with $x \in (B')^{\mathcal{I}_j}$ and $\mathcal{I}_i \models B' \sqsubseteq B_2$. Together with (5) and $C = \neg B_2$, we obtain $\mathcal{I}_i \models B' \sqsubseteq \neg B_1$, and thus Lemma 4.6 yields $\mathcal{I}_j \models B' \sqsubseteq \neg B_1$. This implies that $x \notin B_1^{\mathcal{I}_j}$, which again contradicts (4). □

We have thus shown the converse of Lemma 4.5.

Lemma 4.10. *If there is an r-complete ontology $\mathcal{O}_\mathsf{R}$ for ϕ such that all $\mathcal{O}_\mathsf{R}^i$, $1 \leq i \leq k$, are consistent, then $\mathcal{S}$ is r-satisfiable.*

It remains to show how to combine the reductions described in this section in order to obtain a PSpace-satisfiability test for *DL-Lite*-LTL formulae.

This procedure is based on the original polynomial space-bounded Turing machines for LTL-satisfiability constructed in [SC85]. Given a propositional LTL-formula ϕ^p, the machine $\mathsf{A}_{\phi^\mathsf{p}}$ iteratively guesses complete sets of (negated) subformulae of ϕ^p specifying which subformulae are satisfied at each point in time. Every such set induces a unique $X_i \subseteq \{p_1, \ldots, p_n\}$ containing the propositional variables that are true.

In [SC85, Theorem 4.7], it is shown that if ϕ^p is satisfiable, there must be a *periodic* model of ϕ^p with a period that is exponential in the size of ϕ^p. Hence, $\mathsf{A}_{\phi^\mathsf{p}}$ first guesses two polynomial-sized indices specifying the beginning and end of the first period. Then, it continuously increments a (polynomial-sized) counter and in each step guesses a complete set of (negated) subformulae of ϕ^p. It then checks Boolean consistency of this set and consistency with the set of the previous time point according to the temporal operators. For example, if the previous set contains the formula $p_1 \mathsf{U} p_2$, then either it must also contain p_2 or it contains p_1 and the current set contains $p_1 \mathsf{U} p_2$. In this way, the satisfaction of the U-formula is deferred to the next time point.

In each step, the oldest set is discarded and replaced by the next one. When the counter reaches the beginning of the period, it stores the current set and continues until it reaches the end of the period. At that point, instead of guessing the next set of subformulae, the stored set is used and checked for consistency with the previous set as described above. $\mathsf{A}_{\phi^\mathsf{p}}$ additionally has to ensure that all U-subformulae are satisfied within the period. Thus, the Turing machine never has to remember more than three sets of polynomial size.

We now modify this procedure for our purposes.

Theorem 4.11. *Satisfiability in DL-Lite-LTL is* PSpace*-complete.*

Proof. By Lemmata 4.3, 4.5, and 4.10, the satisfiability of a *DL-Lite*-LTL formula ϕ is equivalent to the existence of a set $\mathcal{S} = \{X_1, \dots, X_k\} \subseteq 2^{\{p_1,\dots,p_n\}}$ and an r-complete ontology $\mathcal{O}_\mathsf{R}$ for ϕ such that $\phi^\mathsf{p}_\mathcal{S}$ is satisfiable and all $\mathcal{O}^i_\mathsf{R}$, $1 \le i \le n$, are consistent. Note that the only difference between ϕ^p and $\phi^\mathsf{p}_\mathcal{S}$ is the requirement that all worlds in a model of $\phi^\mathsf{p}_\mathcal{S}$ should be included in $\mathcal{S}$. Additionally, $\mathcal{O}_\mathsf{R}$ is independent of $\mathcal{S}$ and the additional individual names a^j_i used in each $\mathcal{A}_i$ are not shared among the ontologies $\mathcal{O}^i_\mathsf{R}$. It is thus not necessary to actually construct the whole set $\mathcal{S}$—it is enough to show that each set X_i we encounter when checking ϕ^p (*not* $\phi^\mathsf{p}_\mathcal{S}$) for satisfiability induces a consistent ontology $\mathcal{O}^i_\mathsf{R}$.

To check ϕ for satisfiability, we can thus run a modified version of the Turing machine $\mathsf{A}_{\phi^\mathsf{p}}$ that additionally guesses an r-complete ontology $\mathcal{O}_\mathsf{R}$ in the beginning, and then tests for each guessed set of subformulae whether the induced set $X_i \subseteq \{p_1, \dots, p_n\}$ satisfies the additional requirement that $\mathcal{O}^i_\mathsf{R}$ is consistent. Note that the latter tests can clearly be done in NPSpace since the $\mathcal{O}^i_\mathsf{R}$ are (extended) *DL-Lite$_{core}$*-ontologies of size polynomial in the size of ϕ, and ontology consistency in this logic can be decided in NLogSpace [ACKZ09]. The set $\mathcal{S}$ required for Lemma 4.3 can be obtained by collecting all sets X_i encountered by this machine. However, as described before, this set does not have to be stored explicitly.

Since all this can be done with a nondeterministic Turing machine using only polynomial space (in the size of ϕ), according to [Sav70], satisfiability in *DL-Lite*-LTL can be decided in PSpace. We have PSpace-hardness by Lemma 4.1. □

5 Conclusions

In this report, we have shown that the satisfiability problem for *DL-Lite*-LTL is PSpace-complete and thus has the same complexity as satisfiability in LTL. Interestingly, the complexity stays in PSpace even if rigid names are allowed. In contrast, in the more expressive temporalized DL $\mathcal{ALC}$-LTL, the satisfiability problem is ExpTime-complete if no rigid names are allowed and increases to 2-ExpTime-complete if rigid names are considered [BGL12]. Sometimes, rigid names can even cause undecidability in temporalized DLs [AKL+07, LWZ08].

Although the complexity stays the same, augmenting practical satisfiability tests for propositional LTL such as those described in [BCM+92, CGH97, VW86] to deal with *DL-Lite*-LTL formulae will be more complicated. While all these algorithms can be adapted by executing the consistency test for $\mathcal{O}_{\mathsf{R}}^i$ at each encountered world X_i, one still has to obtain the right r-complete ontology $\mathcal{O}_{\mathsf{R}}$ in the first place; trying all possible $\mathcal{O}_{\mathsf{R}}$ is hardly practical. One possibility to improve this is to try to analyze the structure of ϕ to deduce at least parts of $\mathcal{O}_{\mathsf{R}}$ directly.

Another option is to encode the behavior of $\mathcal{O}_{\mathsf{R}}$ in the *DL-Lite*-LTL formula ϕ as follows. For each rigid axiom α that can occur in an r-complete ontology, we simply add the conjunct $\alpha \leftrightarrow \Box\alpha$ to ϕ, where $\leftrightarrow$ and $\Box$ are the usual abbreviations used in propositional LTL. In this way, one could use the temporal operators to directly enforce the rigidity of all (polynomially many) relevant axioms, and thus "outsource" the problem of guessing $\mathcal{O}_{\mathsf{R}}$ to an optimized propositional LTL solver [BHSV+96, CCGR00, Hol97]. However, it remains to be seen whether this will really increase performance since the LTL solver still does not know anything else about the semantics of these axioms. A solution where the temporal and *DL-Lite* solvers are better integrated is desirable, but more difficult to implement. Analyzing the practical performance of *DL-Lite*-LTL is a topic for future work.

On the theoretical side, we want to investigate the combination of LTL with more expressive members of the *DL-Lite* family. It will be interesting to see how far we can extend the DL part of *DL-Lite*-LTL without increasing the complexity of the satisfiability problem. Moreover, the precise complexity of satisfiability in $\mathcal{EL}$-LTL ($\mathcal{EL}$ is another famous sub-Boolean DL [BBL05]) is still an open problem.

We also want to study whether the decision procedure developed in this report can be adapted to obtain complexity results for temporal query entailment w.r.t. *DL-Lite*-ontologies similar to what was done for $\mathcal{ALC}$ in [BBL13].

References

[ACKZ09] Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev. The *DL-Lite* family and relations. *J. Artif. Intell. Res.*, 36:1–69, 2009.

[AKL+07] Alessandro Artale, Roman Kontchakov, Carsten Lutz, Frank Wolter, and Michael Zakharyaschev. Temporalising tractable description logics. In *Proc. TIME'07*, pages 11–22. 2007.

[AKRZ09] Alessandro Artale, Roman Kontchakov, Vladislav Ryzhikov, and Michael Zakharyaschev. *DL-Lite* with temporalised concepts, rigid axioms and roles. In *Proc. FroCoS'09*, volume 5749 of *LNCS*, pages 133–148, 2009.

[AKRZ10] Alessandro Artale, Roman Kontchakov, Vladislav Ryzhikov, and Michael Zakharyaschev. Temporal conceptual modelling with *DL-Lite*. In *Proc. DL'10*, volume 573 of *CEUR-WS*, 2010.

[AKWZ13] Alessandro Artale, Roman Kontchakov, Frank Wolter, and Michael Zakharyaschev. Temporal description logic for ontology-based data access. In *Proc. IJCAI'13*, 2013.

[BBL05] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proc. IJCAI'05*, pages 364–369, 2005.

[BBL13] Franz Baader, Stefan Borgwardt, and Marcel Lippmann. Temporalizing ontology-based data access. In *Proc. CADE'13*, volume 7898 of *LNCS*, pages 330–344, 2013.

[BCM+92] Jerry R. Burch, Edmund M. Clarke, Kenneth L. McMillan, David L. Dill, and L. J. Hwang. Symbolic model checking: 10^{20} states and beyond. *Inform. Comput.*, 98(2):142–170, 1992.

[BCM+07] 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, 2 edition, 2007.

[BGL12] Franz Baader, Silvio Ghilardi, and Carsten Lutz. LTL over description logic axioms. *ACM T. Comput. Log.*, 13(3):21:1–21:32, 2012.

[BHSV+96] Robert K. Brayton, Gary D. Hachtel, Alberto Sangiovanni-Vincentelli, Fabio Somenzi, Adnan Aziz, Szu-Tsung Cheng, Stephen Edwards, Sunil Khatri, Yuji Kukimoto, Abelardo Pardo, Shaz Qadeer, Rajeev K. Ranjan, Shaker Sarwary, Thomas R. Staple, Gitanjali Swamy, and Tiziano Villa. Vis: A system for verification and synthesis. LNCS, pages 428–432, 1996.

[BLT13] Stefan Borgwardt, Marcel Lippmann, and Veronika Thost. Temporal query answering in the description logic *DL-Lite*. In *Proc. FroCoS'13*, volume 8152 of *LNCS*, pages 165–180, 2013.

[CCGR00] Alessandro Cimatti, Edmund Clarke, Fausto Giunchiglia, and Marco Roveri. NUSMV: A new symbolic model checker. *STTT*, 2(4):410–425, 2000.

[CDL+05] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. *DL-Lite*: Tractable description logics for ontologies. In *Proc. AAAI'05*, pages 602–607, 2005.

[CDL+09] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Mariano Rodriguez-Muro, and Riccardo Rosati. Ontologies and databases: The *DL-Lite* approach. In *5th Int. Summer School Reasoning Web*, volume 5689 of *LNCS*, pages 255–356. 2009.

[CGH97] Edmund M. Clarke, Orna Grumberg, and Kiyoharu Hamaguchi. Another look at LTL model checking. *Formal Methods in System Design*, 10(1):47–71, 1997.

[GKWZ03] Dov M. Gabbay, Agi Kurucz, Frank Wolter, and Michael Zakharyaschev. *Many-Dimensional Modal Logics: Theory and Applications*, volume 148 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 2003.

[Hol97] Gerard J. Holzmann. The model checker SPIN. *IEEE T. Software Eng.*, 23(5):279–295, 1997.

[KLT+10] Roman Kontchakov, Carsten Lutz, David Toman, Frank Wolter, and Michael Zakharyaschev. The combined approach to query answering in *DL-Lite*. In *Proc. KR'10*, pages 247–257, 2010.

[LWZ08] Carsten Lutz, Frank Wolter, and Michael Zakharyaschev. Temporal description logics: A survey. In *Proc. TIME'08*, pages 3–14, 2008.

[Pnu77] Amir Pnueli. The temporal logic of programs. In *Proc. FOCS'77*, pages 46–57, 1977.

[Sav70] Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *J. Comput. Syst. Sci.*, 4(2):177–192, 1970.

[SC85] A. Prasad Sistla and Edmund M. Clarke. The complexity of propositional linear temporal logics. *J. ACM*, 32(3):733–749, 1985.

[VW86] Moshe Y. Vardi and Pierre Wolper. An automata-theoretic approach to automatic program verification. In *Proc. LICS'86*, pages 332–344. IEEE Press, 1986.

[WZ00] Frank Wolter and Michael Zakharyaschev. Temporalizing description logics. In *FroCoS 2*, volume 7 of *Studies in Logic and Computation*, pages 379–402, 2000.