TECHNISCHE
UNIVERSITÄT
DRESDEN

Dresden University of Technology
Institute for Theoretical Computer Science
Chair for Automata Theory

LTCS–Report

Integrate Action Formalisms into Linear Temporal Description Logics

Franz Baader, Hongkai Liu, Anees ul Mehdi

LTCS-Report 09 - 03

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

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

Visiting Address:
Nöthnitzer Str. 46
Dresden

Integrate Action Formalisms into Linear Temporal Description Logics

Franz Baader, Hongkai Liu, Anees ul Mehdi

Instituts für Theoretische Informatik
Technische Universität Dresden
Germany

{baader,liu}@tcs.inf.tu-dresden.de

aneesulmehdi@gmail.com

Abstract

The verification problem for action logic programs with non-terminating behaviour is in general undecidable. In this paper, we consider a restricted setting in which the problem becomes decidable. On the one hand, we abstract from the actual execution sequences of a non-terminating program by considering infinite sequences of actions defined by a Büchi automaton. On the other hand, we assume that the logic underlying our action formalism is a decidable description logic rather than full first-order predicate logic.

1 Introduction

Action programming languages like Golog [8] and Flux [14], which are respectively based on the situation calculus and the fluent calculus, can be used to control the behaviour of autonomous agents and mobile robots. Often, programs written in these languages are non-terminating since the robots are supposed to perform open ended tasks, like delivering coffee as long as there are requests. To ensure that the execution of such a program leads to the desired behaviour of the robot, one needs to specify the required properties in a formal way, and then verify that these requirements are met by any (infinite) execution of the program. In the coffee delivery example, one might, e.g., want to show that anyone requesting coffee will eventually get it delivered. When trying to automate this verification task, one has to deal with two sources of undecidability: (i) the expressiveness of the programming constructs (while loops, recursion) and (ii) the expressiveness of situation/fluent calculus, which encompasses full first-order predicate logic.

Verification for non-terminating Golog programs has first been addressed by De Giacomo, Ternovskaia, and Reiter [7], who express both the semantics of the programs and the properties to be verified using an appropriate fixpoint logic. To verify a property of a program, one first needs to compute a fixpoint, which is expressed in second-order logic. In general, this computation need not terminate (this corresponds to the first source of undecidability). Even if the fixpoint computation does terminate, verifying that the desired property holds requires a manual, meta-theoretic proof. Attempts to automate this approach are usually restricted to propositional logic [11]. Claßen and Lakemeyer [6] aim at the fully automated verification of non-terminating Golog programs. They specify properties in an extension of the situation calculus by constructs of the first-order temporal logic CTL*. Verification then basically boils down to the computation of a fixpoint, where again this computation need not terminate. If the fixpoint computation terminates, then the proof that the desired property holds is a deduction in the underlying logic (i.e., no meta-theoretic reasoning is required). However, due to the second source of undecidability mentioned above, this deduction problem is in general not decidable.

In the present paper, we introduce a restricted setting, where both sources of undecidability are avoided. Regarding the first source, instead of examining the actual execution sequences of a given Golog or Flux program, we consider infinite sequences of actions that are accepted by a given Büchi automaton $\mathcal{B}$. If $\mathcal{B}$ is an abstraction of the program, i.e. all possible execution sequences of the program are accepted by $\mathcal{B}$, then any property that holds in all the sequences accepted by $\mathcal{B}$ is also a property that is satisfied by any execution of the program. For example, assume that, among other actions, researcher John can perform the action "review paper," which makes him tired, and that robot Robin can perform the actions "deliver paper" and "deliver coffee," where the latter one results in John no longer being tired, whereas the former one results in John having to review yet another paper. The property ϕ_{tired} we want to ensure is that John does not stay tired indefinitely, i.e., whenever he is tired at some time point, then there is a later time point at which he is not tired. Assume that there is a complex program controlling Robin's behaviour, but we can show that Robin will infinitely often deliver coffee. Thus, the Büchi automaton $\mathcal{B}_{deliver}$ that accepts all action sequences that contain the action "deliver coffee" infinitely often is an abstraction of this program, and it is easy to see that any infinite sequence of actions accepted by this automaton satisfies ϕ_{tired}.

To avoid the second source of undecidability, we restrict the underlying logic to a decidable description logic. Description Logics (DLs) [2] are a well-known family of knowledge representation formalisms that may be viewed as fragments of first-order logic (FO). The main strength of DLs is that they offer considerable expressive power going far beyond propositional logic, while reasoning is still decidable. An action formalism based on DLs was first introduced in [4], and it was shown that important reasoning problems such as the projection problem,

which are undecidable in the full situation/fluent calculus, are decidable in this restricted formalism.

In this paper, we show that these positive results can be extended to the verification problem. As logic for specifying properties of infinite sequences of DL actions, we use the temporalized DL $\mathcal{ALC}$-LTL recently introduced in [3], which extends the well-known propositional linear temporal logic (LTL) [12] by allowing for the use of axioms (i.e., TBox and ABox statements) of the basic DL $\mathcal{ALC}$ in place of propositional letters.[1] Note that the property ϕ_{tired} that we have used in the above coffee delivery example can easily be expressed in LTL.

In the next section, we first recall the basic definitions for DLs, action formalisms based on DLs, temporalized DLs, and Büchi automata, and then introduce the verification problem and its dual, the satisfiability problem, which asks whether there is an infinite sequence of actions accepted by the given Büchi automaton $\mathcal{B}$ that satisfies the property. Since these problems are interreducible in polynomial time, we then concentrate on solving the satisfiability problem. In Section 3, we consider a restricted version of the general problem, where the Büchi automaton accepts exactly one infinite sequence of unconditional actions without occlusions. The general problem is then investigated in Section 4.

2 Preliminaries

The integration of actions formalisms can be applied to any DL which has well-defined semantics of actions [4]. In this paper, we investigate the computational complexity of the inference problems at hand for DLs between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$, hence we give the syntax and semantics of those DLs [2] in this section.

In DLs, concepts are inductively defined starting with a set $\mathsf{N_C}$ of *concept names*, a set $\mathsf{N_R}$ of *role names*, and (possibly) a set $\mathsf{N_I}$ of *individual names*. The expressiveness of a DL is determined by a set of *constructors*. The relevant constructors to the DLs considered in this paper are shown in Table 1, where we use C, D to denote concepts, A to denote a concept name, r to denote roles, and a, b to denote individual names. As usual, we use $\top$ as an abbreviation for $A \sqcup \neg A$.

The DL that allows only for negation, conjunction, disjunction, existential restriction, and value restriction is called $\mathcal{ALC}$. Different extensions of $\mathcal{ALC}$ allow additionally for different constructors, indicated by the name of the DL. For example, the name $\mathcal{ALCQIO}$ stands for the DL which extends $\mathcal{ALC}$ with $\mathcal{Q}$ualified number restriction, $\mathcal{I}$nverse role, and n$\mathcal{O}$minals. If a DL allows for inverse roles, a *role* is r or r^{-1} for some $r \in \mathsf{N_R}$. A role is a role name otherwise.

An *interpretation* $\mathcal{I}$ is a pair $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$ is a non-empty set and $\cdot^{\mathcal{I}}$ is a

[1] More precisely, we will consider the extension of $\mathcal{ALC}$-LTL to DLs between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$, but disallow TBox statements.

Name	Syntax	Semantics
negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
existential restriction	$\exists r.C$	$\{x \mid \exists y.((x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}})\}$
value restriction	$\forall r.C$	$\{x \mid \forall y.((x,y) \in r^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}})\}$
qualified number restriction	$(\leqslant n\ r\ C)$ $(\geqslant n\ r\ C)$	$\{d \mid \sharp\{e \in C^{\mathcal{I}} \mid (d,e) \in r^{\mathcal{I}}\} \leq n\}$ $\{d \mid \sharp\{e \in C^{\mathcal{I}} \mid (d,e) \in r^{\mathcal{I}}\} \geq n\}$
inverse role	r^-	$\{(y,x) \mid (x,y) \in r^{\mathcal{I}}\}$
nominal	$\{a\}$	$\{a^{\mathcal{I}}\}$

Table 1: Syntax and semantics of concepts and roles.

mapping that assigns a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$ to each concept name $A \in \mathsf{N_C}$, a binary relation $r^{\mathcal{I}}$ on $\Delta^{\mathcal{I}}$ to each role name $r \in \mathsf{N_R}$, and an element $a^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$ to each individual name $a \in \mathsf{N_I}$. Moreover, for all $a, b \in \mathsf{N_I}$, $a \neq b$ implies $a^{\mathcal{I}} \neq b^{\mathcal{I}}$. The interpretation of inverse role and concept descriptions is shown in the third column of Table 1, where $\#S$ is the cardinality of a set S. We call an $x \in \Delta^{\mathcal{I}}$ is a *named* object in $\mathcal{I}$ iff there exists an $a \in \mathsf{N_I}$ with $a^{\mathcal{I}} = x$. Otherwise, x is an *anonymous* object in $\mathcal{I}$.

An *acyclic TBox* $\mathcal{T}$ is a finite set of *concept definitions* of the form $A \equiv C$ such that there is no concept name A occurring twice on the left-hand side of concept definitions or using directly or indirectly itself in its definition [2]. The concept names occurring on the left-hand side of concept definitions of $\mathcal{T}$ are called *defined concept names* in $\mathcal{T}$, whereas all other concept names are called *primitive concept names* in $\mathcal{T}$. An interpretation $\mathcal{I}$ is a *model* of $\mathcal{T}$ (denoted by $\mathcal{I} \models \mathcal{T}$) iff for all $A \equiv C \in \mathcal{T}$, we have $A^{\mathcal{I}} = C^{\mathcal{I}}$. Note that we restrict our attention to acyclic TBoxes since, for more general TBox formalisms involving general concept inclusion axioms (GCIs), it is not clear how to define an appropriate semantics for DL actions.

An *ABox* $\mathcal{A}$ is a finite set of *assertions* of the form $C(a)$, $r(a,b)$, or $\neg r(a,b)$, where C is a concept, r is a role, a, b are individual names. We call an assertion with the first form a *concept assertion*, a *role assertion* otherwise. An interpretation $\mathcal{I}$ is a *model* of assertion

$$C(a) \text{ iff } a^{\mathcal{I}} \in C^{\mathcal{I}};$$
$$r(a,b) \text{ iff } (a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}};$$
$$\neg r(a,b) \text{ iff } (a^{\mathcal{I}}, b^{\mathcal{I}}) \notin r^{\mathcal{I}}.$$

$\mathcal{I}$ is a *model* of $\mathcal{A}$ (denoted by $\mathcal{I} \models \mathcal{A}$) iff for all $\varphi \in \mathcal{A}$, $\mathcal{I}$ is a model of φ.

Given an assertion γ, its negation $\neg\gamma$ is again an assertion: $\neg(C(a)) := (\neg C)(a)$, $\neg(r(a,b)) := \neg r(a,b)$, and $\neg(\neg r(a,b)) := r(a,b)$.

We say that An ABox $\mathcal{A}$ is *consistent* w.r.t. an acyclic TBox $\mathcal{T}$ if $\mathcal{A}$ and $\mathcal{T}$ have a common model. Consistency of an ABox w.r.t. an acyclic TBox is one of the standard inference problems in DLs, which is to decide existence of a model for a given knowledge base. Its complexity in DLs between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$ has been thoroughly studied. We will see later on that the inference problems considered in this paper can be decided with the help of consistency.

The temporalized DLs are obtained from propositional linear temporal logic (LTL) [12] by allowing for the use of assertions in place of propositional letters.

Definition 1. DL-LTL formulae are defined by induction:

- if β is an assertion, then β is an DL-LTL formula;
- if ϕ, ψ are DL-LTL formulae, then so are $\phi \wedge \psi$, $\phi \vee \psi$, $\neg\phi$, $\phi \mathsf{U} \psi$, and $\mathsf{X}\phi$.

$\triangle$

We use true as an abbreviation for $\top(a)$, $\phi \rightarrow \psi$ for $\neg\phi \vee \psi$, $\Diamond\phi$ for $\mathsf{true}\mathsf{U}\phi$ (*diamond*, which should be read as "sometime in the future"), and $\Box\phi$ for $\neg(\mathsf{true}\mathsf{U}\neg\phi)$ (*box*, which should be read as "always in the future").

The difference to the logic $\mathcal{ALC}$-LTL introduced in [3] is, on the one hand, that assertions in DLs between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$ rather than just $\mathcal{ALC}$-assertions can be used. On the other hand, an $\mathcal{ALC}$-LTL formula may also contain GCIs, whereas in DL-LTL we do not allow the use of terminological axioms. Instead, we use a global acyclic TBox, whose concept definitions must hold at every time point. The semantics of DL-LTL is based on DL-LTL structures, which are infinite sequences of interpretations over the same non-empty domain Δ (constant domain assumption) in which every individual name stands for a unique element of Δ (rigid individual names).

Definition 2. A *DL-LTL structure* is a sequence $\mathfrak{I} = (\mathcal{I}_i)_{i=0,1,\ldots}$ of interpretations $\mathcal{I}_i = (\Delta, \cdot^{\mathcal{I}_i})$ such that $a^{\mathcal{I}_i} = a^{\mathcal{I}_j}$ for all individual names a and all $i, j \in \{0, 1, 2, \ldots\}$. Given a DL-LTL formula ϕ, a DL-LTL structure $\mathfrak{I} = (\mathcal{I}_i)_{i=0,1,\ldots}$, and a time point $i \in \{0, 1, 2, \ldots\}$, validity of ϕ in $\mathfrak{I}$ at time i (written $\mathfrak{I}, i \models \phi$) is defined inductively:

$$
\begin{array}{lll}
\mathfrak{I}, i \models \beta & \text{iff} & \mathcal{I}_i \text{ satisfies the ABox assertion } \beta \\
\mathfrak{I}, i \models \phi \wedge \psi & \text{iff} & \mathfrak{I}, i \models \phi \text{ and } \mathfrak{I}, i \models \psi \\
\mathfrak{I}, i \models \phi \vee \psi & \text{iff} & \mathfrak{I}, i \models \phi \text{ or } \mathfrak{I}, i \models \psi \\
\mathfrak{I}, i \models \neg\phi & \text{iff} & \text{not } \mathfrak{I}, i \models \phi \\
\mathfrak{I}, i \models \mathsf{X}\phi & \text{iff} & \mathfrak{I}, i+1 \models \phi \\
\mathfrak{I}, i \models \phi \mathsf{U} \psi & \text{iff} & \text{there is } k \geq i \text{ such that } \mathfrak{I}, k \models \psi \\
 & & \text{and } \mathfrak{I}, j \models \phi \text{ for all } j, i \leq j < k
\end{array}
$$

△

In this paper, we assume that the transition from $\mathcal{I}_i$ to $\mathcal{I}_{i+1}$ in a DL-LTL structure is caused by the application of an action. We recall the pertinent definitions for DL actions from [4].

Definition 3 (Action)**.** Let $\mathcal{T}$ be an acyclic TBox. An *action* α for $\mathcal{T}$ is a triple $(\mathsf{pre}, \mathsf{occ}, \mathsf{post})$ which consists of

- a finite set pre of ABox assertions, the *pre-conditions*;
- a finite set occ of *occlusions* of the form $A(a)$ or $r(a, b)$, with A a primitive concept name in $\mathcal{T}$, r a role name, and $a, b \in \mathsf{N_I}$;
- a finite set post of *conditional post-conditions* of the form β/γ, where γ is an ABox assertion and ψ is a *primitive literal for* $\mathcal{T}$, i.e., an ABox assertion $A(a)$, $\neg A(a)$, $r(a, b)$, or $\neg r(a, b)$ with A a primitive concept name in $\mathcal{T}$, r a role name, and a, b individual names.

If every $\beta/\gamma \in \mathsf{post}$ is of the form $\top(a)/\gamma$, then we call α an *unconditional action*, and in this case we write γ instead of $\top(a)/\gamma$. Otherwise, it is a *conditional action*. We say that an action α is *without occlusions* if $\mathsf{occ} = \emptyset$. Otherwise, α is *with occlusions*. △

Basically, such an action is applicable in an interpretation if its pre-conditions are satisfied. The conditional post-condition β/γ requires that γ must hold after the application of the action if β was satisfied before the application. In addition, nothing should change that is not required to change by some post-condition. Occlusions specify the parts where the concept names and the role names can change freely.

Definition 4. Let $\mathcal{T}$ be an acyclic TBox, $\alpha = (\mathsf{pre}, \mathsf{occ}, \mathsf{post})$ an action for $\mathcal{T}$, and $\mathcal{I}, \mathcal{I}'$ interpretations sharing the same domain and interpretations of all individual names. We say that α *may transform* $\mathcal{I}$ *to* $\mathcal{I}'$ *w.r.t.* $\mathcal{T}$ ($\mathcal{I} \Rightarrow_\alpha^{\mathcal{T}} \mathcal{I}'$) iff, $\mathcal{I}$ and $\mathcal{I}'$ are models of $\mathcal{T}$ and for each primitive concept name A in $\mathcal{T}$ and each role name r, we have

$$\begin{aligned} A^{\mathcal{I}'} \cap I_A &= ((A^{\mathcal{I}} \cup A^+) \setminus A^-) \cap I_A, \text{ and} \\ r^{\mathcal{I}'} \cap I_r &= ((r^{\mathcal{I}} \cup r^+) \setminus r^-) \cap I_r, \end{aligned}$$

where

$$\begin{aligned} A^+ &= \{b^{\mathcal{I}} \mid \varphi/A(b) \in \mathsf{post} \wedge \mathcal{I} \models \varphi\}, \\ A^- &= \{b^{\mathcal{I}} \mid \varphi/\neg A(b) \in \mathsf{post} \wedge \mathcal{I} \models \varphi\}, \\ I_A &= (\Delta^{\mathcal{I}} \setminus \{b^{\mathcal{I}} \mid A(b) \in \mathsf{occ}\}) \cup (A^+ \cup A^-), \\ r^+ &= \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid \varphi/r(a, b) \in \mathsf{post} \wedge \mathcal{I} \models \varphi\}, \\ r^- &= \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid \varphi/\neg r(a, b) \in \mathsf{post} \wedge \mathcal{I} \models \varphi\}, \text{ and} \\ I_r &= ((\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}) \setminus \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid r(a, b) \in \mathsf{occ}\}) \cup (r^+ \cup r^-). \end{aligned}$$

We say that α is *executable* in $\mathcal{I}$ if $\mathcal{I}$ is a model of pre. $\triangle$

Let $\alpha = \alpha_1 \cdots \alpha_m$ be a finite sequence of actions. We use $\mathcal{I} \Rightarrow_{\alpha}^{\mathcal{T}} \mathcal{I}'$ to abbreviate $\mathcal{I} \Rightarrow_{\alpha_1}^{\mathcal{T}} \cdots \Rightarrow_{\alpha_m}^{\mathcal{T}} \mathcal{I}'$. We say that an action α is *consistent* with $\mathcal{T}$ iff for all $\beta_1/\gamma, \beta_2/\neg\gamma$ in the post-conditions of α, $\{\beta_1, \beta_2\} \cup \mathcal{T}$ is inconsistent. In this paper, we consider only consistent actions. The requirement of consistent actions is to avoid an unintuitive result of applying actions. For example, the set of post-conditions of an *inconsistent* action α is $\{\beta/A(a), \beta/\neg A(a)\}$. Thus, for all models $\mathcal{I}$ of $\mathcal{T}$ with $\mathcal{I} \models \varphi$, and for all $\mathcal{I}'$ with $\mathcal{I} \Rightarrow_{\alpha}^{\mathcal{T}} \mathcal{I}'$, according to Definition 4, $\mathcal{I}' \not\models A(a)$. This means that $\beta/A(a)$ is not satisfied by the application of α.

It follows directly from Definition 4 that for all interpretation $\mathcal{I}$, DL actions can only change the interpretations of named objects in $\mathcal{I}$. Note that for all acyclic TBoxes $\mathcal{T}$ and for all actions α for $\mathcal{T}$, if α is without occlusions, then for all models $\mathcal{I}$ of $\mathcal{T}$, there *exists* a *unique* $\mathcal{I}'$ such that $\mathcal{I} \Rightarrow_{\alpha}^{\mathcal{T}} \mathcal{I}'$ [4].

In this paper, we are interested in deciding whether the executions of *infinite* sequences of actions satisfy a (temporal) property expressed in DL-LTL. Let Σ be a finite set of actions for $\mathcal{T}$. An *infinite sequence* of such actions can be viewed as an infinite word over the alphabet Σ, i.e., a mapping $w : \mathbb{N} \to \Sigma$, where $\mathbb{N}$ denotes the set of non-negative integers.

Definition 5. Let $\mathcal{T}$ be an acyclic TBox, $\mathcal{A}$ be an ABox, and w an infinite sequence of actions for $\mathcal{T}$. The DL-LTL structure $\mathfrak{I} = (\mathcal{I}_i)_{i=0,1,\ldots}$ w.r.t. $\mathcal{T}$ and w is *generated by w from $\mathcal{A}$ w.r.t. $\mathcal{T}$* if $\mathcal{I}_0$ is a model of $\mathcal{A}$ and, for all $i \geq 0$, we have $\mathcal{I}_i \Rightarrow_{w(i)}^{\mathcal{T}} \mathcal{I}_{i+1}$ and $w(i)$ is executable in $\mathcal{I}_i$. $\triangle$

For the verification problem, we consider infinite sequences of actions accepted by a Büchi automaton. *Büchi automata* are finite automata accepting infinite words [15]. A Büchi automaton $\mathcal{B}$ basically looks and works like a "normal" finite automaton, but it receives infinite words w as inputs, and thus generates infinite runs. An infinite run of $\mathcal{B}$ on w is an infinite word $r : \mathbb{N} \to Q$ over the alphabet Q of states of $\mathcal{B}$ such that $r(0)$ is an initial state and, for every $i \geq 0$, there is a transition of $\mathcal{B}$ from the state $r(i)$ with letter $w(i)$ to the state $r(i+1)$. This run is accepting if it infinitely often reaches a final state. The *language $L_\omega(\mathcal{B})$ of infinite words accepted by $\mathcal{B}$* consists of all infinite words w over Σ such that $\mathcal{B}$ has an accepting run on w.

We are now ready to give a formal definition of the *verification problem*, which was informally introduced in Section 1, as the problem of deciding validity of a DL-LTL formula w.r.t. an acyclic TBox, an ABox, and a Büchi automaton.

Definition 6. Let $\mathcal{T}$ be an acyclic TBox, $\mathcal{A}$ an ABox, Σ a finite set of actions for $\mathcal{T}$, $\mathcal{B}$ a Büchi automaton for the alphabet Σ, and ϕ a DL-LTL formula.

- ϕ is *valid w.r.t. $\mathcal{T}$, $\mathcal{A}$, and $\mathcal{B}$* if $\mathfrak{I}, 0 \models \phi$ holds for all $w \in L_\omega(\mathcal{B})$ and all DL-LTL structures $\mathfrak{I}$ generated by w from $\mathcal{A}$ w.r.t. $\mathcal{T}$.

- ϕ is *satisfiable w.r.t.* $\mathcal{T}$*,* $\mathcal{A}$*, and* $\mathcal{B}$ if there is $w \in L_\omega(\mathcal{B})$ and a DL-LTL structures $\mathfrak{I}$ generated by w from $\mathcal{A}$ w.r.t. $\mathcal{T}$ such that $\mathfrak{I}, 0 \models \phi$.

$\triangle$

Obviously, ϕ is valid w.r.t. $\mathcal{T}$, $\mathcal{A}$ and $\mathcal{B}$ iff $\neg\phi$ is unsatisfiable w.r.t. $\mathcal{T}$, $\mathcal{A}$ and $\mathcal{B}$. For this reason, we concentrate in the following on solving the satisfiability problem.

Let us give the formal description of our example in the previous section. We use the following initial ABox

$$\begin{array}{rcl} \mathcal{A} & = & \{\exists \mathsf{hasSubmittedPaper}.\{\mathsf{paper}_i\}(\mathsf{ecai2010}) \mid 1 \leq i \leq 500\} \cup \\ & & \{\neg \mathsf{Reviewed}(\mathsf{paper}_i) \mid 1 \leq i \leq 500\} \end{array}$$

to state that there are 500 papers submitted to the conference ecai2010 and that none of them has been reviewed yet. Robot Robin is in charge of delivering papers to the reviewers and keeping them vigorous by serving them coffee. John is one of the reviewers. We define the following actions:

$$\begin{array}{rcl} \mathsf{reviewPaper}_i & = & (\{\neg \mathsf{Tired}(\mathsf{john}), \mathsf{Assigned}(\mathsf{paper}_i)\}, \emptyset, \\ & & \{\neg \mathsf{Reviewed}(\mathsf{paper}_i)/\mathsf{Reviewed}(\mathsf{paper}_i), \\ & & \neg \mathsf{Reviewed}(\mathsf{paper}_i)/\mathsf{Tired}(\mathsf{john})\}), \\ \mathsf{deliverPaper}_i & = & (\{\exists \mathsf{hasSubmittedPaper}.\{\mathsf{paper}_i\}(\mathsf{ecai2010})\}, \emptyset, \{\mathsf{Assigned}(\mathsf{paper}_i)\}), \\ \mathsf{deliverCoffee} & = & (\emptyset, \emptyset, \{\neg \mathsf{Tired}(\mathsf{john})\}), \end{array}$$

where i is with $1 \leq i \leq 500$. The property ϕ_{tired} is captured by the following DL-LTL formula:

$$\phi_{tired} = \Box(\mathsf{Tired}(\mathsf{john}) \rightarrow \Diamond \neg \mathsf{Tired}(\mathsf{john})).$$

The Büchi automaton $\mathcal{B}_{deliver}$ is depicted in Figure 1. The state q_0 is the initial state and q_1 is the final state. The alphabet of $\mathcal{B}_{deliver}$ is Σ which consists of the actions defined above. The actions $\mathsf{reviewPaper}_i$, $\mathsf{deliverPaper}_i$, and deliverCoffee are respectively abbreviated with rP_i, dP_i, and dC. It is easy to check that for every $w \in \Sigma^\omega$, $w \in L_\omega(\mathcal{B}_{deliver})$ iff the action deliverCoffee occurs infinitely often in w.

3 The Case of a Single Cyclic Sequence of Unconditional Actions without Occlusions

We say that the infinite word w is *cyclic* if it starts with an initial word $\alpha_1 \cdots \alpha_m$ and then repeats a non-empty word $\beta_1 \cdots \beta_n$ infinitely often. We denote such a cyclic word by $w = \alpha_1 \cdots \alpha_m (\beta_1 \cdots \beta_n)^\omega$. The following facts are well-known [15]

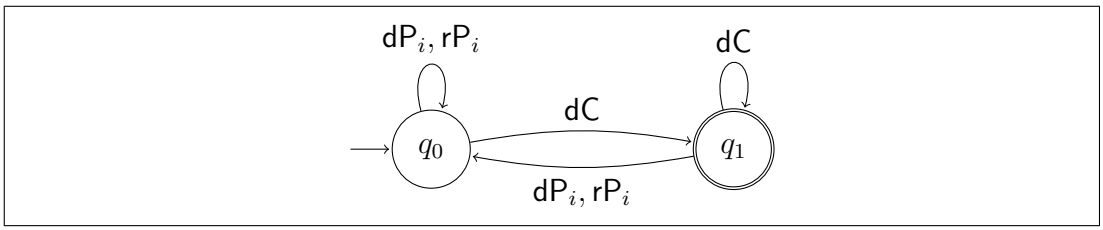

Figure 1: $\mathcal{B}_{deliver}$

(and easy to see): if $\mathcal{B}$ is a Büchi automaton that accepts a singleton language $\{w\}$, then w is a cyclic word of the form $w = \alpha_1 \cdots \alpha_m(\beta_1 \cdots \beta_n)^\omega$ where m, n are bounded by the cardinality of the set of states of $\mathcal{B}$; conversely any singleton language $\{w\}$ consisting of a cyclic word $w = \alpha_1 \cdots \alpha_m(\beta_1 \cdots \beta_n)^\omega$ is accepted by a corresponding Büchi automaton $\mathcal{B}_w$ such that the cardinality of the set of states of $\mathcal{B}$ is linear in $m + n$.

In this section, we consider only Büchi automata accepting singleton languages. In addition, we restrict the attention to unconditional actions without occlusions. Thus, *for the remainder of this section, we assume that* $\mathcal{T}$ *is an acyclic TBox,* $\mathcal{A}$ *an ABox,* Σ *a finite set of* unconditional *actions for* $\mathcal{T}$ *(without occlusions),* $\mathcal{B}_w$ *a Büchi automaton for the alphabet* Σ *accepting the singleton language* $\{w\}$ *for* $w = \alpha_1 \cdots \alpha_m(\beta_1 \cdots \beta_n)^\omega$*, and* ϕ *an DL-LTL formula.* Such a cyclic sequence of actions represents a program that, after an initialization phase, runs in a non-terminating loop.

Lemma 7. *Let* $\mathcal{I}$ *and* $\mathcal{I}'$ *be two interpretations and* $\beta = \beta_1 \cdots \beta_n$ *be a sequence of actions for a TBox* $\mathcal{T}$*. If* $\mathcal{I} \Rightarrow^{\mathcal{T}}_{\beta} \mathcal{I}'$*, then* $\mathcal{I}' \Rightarrow^{\mathcal{T}}_{\beta} \mathcal{I}'$*.*

Proof. Suppose $\mathcal{I}' \Rightarrow^{\mathcal{T}}_{\beta} \mathcal{J}'$ for some interpretation $\mathcal{J}'$ (such a $\mathcal{J}'$ always exists since for every model $\mathcal{I}$ of $\mathcal{T}$ and every action α for $\mathcal{T}$, if α is without occlusions, then there exists an $\mathcal{I}'$ with $\mathcal{I} \Rightarrow^{\mathcal{T}}_{\alpha} \mathcal{I}'$ [4]). Thus, it is enough to show that $\mathcal{I}' = \mathcal{J}'$. Since both interpretations $\mathcal{I}'$ and $\mathcal{J}'$ share the domain (suppose it is denoted by Δ) and interpretations of all individual names, it remains to show for all primary concept name A w.r.t. $\mathcal{T}$ and all $r \in \mathsf{N_R}$, we have $A^{\mathcal{I}'} = A^{\mathcal{J}'}$ and $r^{\mathcal{I}'} = r^{\mathcal{J}'}$. Here we show only the former and the latter can be proved analogously.

"$\subseteq$": Assume that $A^{\mathcal{I}'} \not\subseteq A^{\mathcal{J}'}$. Then there is a $d \in \Delta$ such that $d \in A^{\mathcal{I}'}$ and $d \notin A^{\mathcal{J}'}$. Since $\mathcal{I}' \Rightarrow^{\mathcal{T}}_{\beta} \mathcal{J}'$, there is a $\beta_j \in \{\beta_1, \ldots, \beta_n\}$ such that $\neg A(a) \in \beta_j$ for some $a \in \mathsf{N_I}$ with $a^{\mathcal{I}} = d$ and for all i with $j < i \leq n$, we have $A(a) \notin \beta_i$. (Intuitively, it means that d is removed from A by β_j and never added afterwards.) However, together with $\mathcal{I} \Rightarrow^{\mathcal{T}}_{\beta} \mathcal{I}'$, such a β_j in β implies $d \notin A^{\mathcal{I}'}$, which contradicts the assumption.

"$\supseteq$": Assume that $A^{\mathcal{J}'} \not\subseteq A^{\mathcal{I}'}$. Then there is a $d \in \Delta$ such that $d \in A^{\mathcal{J}'}$ and $d \notin A^{\mathcal{I}'}$. Since $\mathcal{I}' \Rightarrow^{\mathcal{T}}_{\beta} \mathcal{J}'$, there is a $\beta_j \in \{\beta_1, \ldots, \beta_n\}$ such that $A(a) \in \beta_j$ for some $a \in \mathsf{N_I}$ with $a^{\mathcal{I}} = d$ and for all i with $j < i \leq n$, we have $\neg A(a) \notin \beta_i$.

(Intuitively, it means that d is added into A by β_j and never removed afterwards.) However, together with $\mathcal{I} \Rightarrow_\beta^{\mathcal{T}} \mathcal{I}'$, such a β_j in β implies $d \in A^{\mathcal{I}'}$, which contradicts the assumption. ❑

The main observation that allows us to solve the satisfiability problem for ϕ w.r.t. $\mathcal{T}$, $\mathcal{A}$ and $\mathcal{B}_w$ is that each DL-LTL structure generated by w from $\mathcal{A}$ w.r.t. $\mathcal{T}$ "runs into a cycle" after the first $m + 2n$ interpretations. This is a direct consequence of Lemma 7.

Lemma 8. *Let $\mathfrak{I} = (\mathcal{I}_i)_{i=0,1,\ldots}$ be a DL-LTL structure generated by $w = \alpha_1 \cdots \alpha_m(\beta_1 \cdots \beta_n)^\omega$ from $\mathcal{A}$ w.r.t. $\mathcal{T}$. Then $\mathcal{I}_{m+kn+i} = \mathcal{I}_{m+n+i}$ for all $k \geq 2$ and $0 \leq i < n$.*

Based on this observation, we can solve the satisfiability problem by the reduction from the satisfiability problem of DL-LTL formulas to the consistency problem:

- Construct an acyclic TBox $\mathcal{T}_{\mathsf{red}}$ and an ABox $\mathcal{A}_{\mathsf{red}}$ from $\mathcal{A}$, $\mathcal{T}$, w, and ϕ,
- Construct an ABox $\mathcal{A}_{\mathsf{pre}}$ from w, and
- Compute an ABox $\mathcal{A}_\phi$ from ϕ by a tableau algorithm.

We show that ϕ is satisfiable w.r.t. $\mathcal{T}$, $\mathcal{A}$, and $\mathcal{B}_w$ iff $\mathcal{A}_{\mathsf{red}} \cup \mathcal{A}_{\mathsf{pre}} \cup \mathcal{A}_\phi$ is consistent w.r.t. $\mathcal{T}_{\mathsf{red}}$.

Without loss of generality, we can assume that there are no LTL negation signs in ϕ. First, we transform ϕ into negation normal form (NNF), i.e., LTL negation signs occur only in front of ABox assertions. To this end, we need to introduce the dual operator R of U. $\varphi \mathsf{R} \psi = \neg(\neg\varphi \mathsf{U} \neg\psi)$, i.e., $\mathfrak{I}, i \models \varphi \mathsf{R} \psi$ iff for all $m \geq i$, $\mathfrak{I}, m \models \psi$ or there exists a k such that $\mathfrak{I}, k \models \varphi$ and $\mathfrak{I}, j \models \psi$ for all j with $i \leq j \leq k$.

$$\begin{array}{rclcrcl} \neg(\varphi \wedge \psi) & \rightsquigarrow & \neg\varphi \vee \neg\psi; & & \neg(\varphi \vee \psi) & \rightsquigarrow & \neg\varphi \wedge \neg\psi; \\ \neg(\varphi \mathsf{U} \psi) & \rightsquigarrow & \neg\varphi \mathsf{R} \neg\psi; & & \neg(\varphi \mathsf{R} \psi) & \rightsquigarrow & \neg\varphi \mathsf{U} \neg\psi; \\ \neg\mathsf{X}\varphi & \rightsquigarrow & \mathsf{X}\neg\varphi. & & & & \end{array}$$

By exhaustively applying the above rules, every DL-LTL formula ϕ can be transformed to an equivalent one in NNF. What is more, replace respectively $\neg(C(a))$ with $(\neg C)(a)$, $\neg(r(a,b))$ with $\neg r(a,b)$, and $\neg((\neg r)(a,b))$ with $r(a,b)$ after NNF of φ is obtained. It is clear that those replacements are satisfiability preserving and can be done in polynomial time of the size of ϕ. The size of obtained formula is polynomial in the size of ϕ [17].

Basically, we apply the approach for solving the projection problem from [4] to the finite sequence of actions $\alpha_1 \cdots \alpha_m \ \beta_1 \cdots \beta_n \beta_1 \cdots \beta_{n-1}$. In this approach, time-stamped copies of all concept and role names occurring in the input (i.e., in

$w, \mathcal{T}, \mathcal{A}, \phi)$ are generated, together with a number of additional auxiliary concept names. Using this extended vocabulary, one builds, for every assertion γ occurring in the input, time-stamped variants $\gamma^{(i)}$ for all $i, 0 \leq i \leq m+2n-1$. The extended vocabulary is also used to construct an acyclic TBox $\mathcal{T}_{\mathsf{red}}$ and an ABox $\mathcal{A}_{\mathsf{red}}$. As we will see in the construction, for each concept name A and each role name r in the input, we introduce labeled names $A^{(i)}$, $T_A^{(i)}$, and $r^{(i)}$ to describe the interpretation of those names after the sequence of action $w(0) \cdots w(i-1)$.[2]

Let Obj be the set of all the individual names in the input, i.e, in $\mathcal{A}$, $\mathcal{T}$, ϕ or $\mathcal{B}_w$.

$$\mathcal{T}_N = \{N \equiv \bigsqcup_{a \in \mathsf{Obj}} \{a\}\}.$$

Let Sub be the set of the subconcepts in the input. For every $C \in \mathsf{Sub}$, if $C \in \mathsf{Sub}$ is not a defined concept name of $\mathcal{T}$, then there is a concept definition of $T_C^{(i)}$ in $\mathcal{T}_{\mathsf{Sub}}^{(i)}$. Moreover, $\mathcal{T}_{\mathsf{Sub}}^{(i)}$ contains only those concept definitions. The concept definition of $T_C^{(i)}$ is defined inductively on the structure of C as described in Figure 2. We are now ready to assemble $\mathcal{T}_{\mathsf{red}}$:

$$\mathcal{T}_{\mathsf{red}} = \mathcal{T}_N \cup (\bigcup_{i=0}^{m+2n-1} \mathcal{T}_{\mathsf{Sub}}^{(i)}) \cup \{T_A^{(i)} \equiv T_E^{(i)} \mid A \equiv E \in \mathcal{T}, i \leq m+2n-1\}.$$

The TBoxes $\mathcal{T}_N$ and $\mathcal{T}_{\mathsf{sub}}^{(i)}$ can ensure that the interpretations of concept and role names remain unchanged by actions on the anonymous objects and the last part of $\mathcal{T}_{\mathsf{red}}$ is to make sure that $\mathcal{T}$ is satisfied no matter how actions change an interpretation. The changes by actions on the named objects will be guaranteed by $\mathcal{A}_{\mathsf{red}}$. For every ABox assertion φ we define $\varphi^{(i)}$ as

$$\varphi^{(i)} = \begin{cases} T_C^{(i)}(a) & \text{if } \varphi = C(a) \\ r^{(i)}(a,b) & \text{if } \varphi = r(a,b) \\ \neg r^{(i)}(a,b) & \text{if } \varphi = \neg r(a,b) \end{cases} \quad (1)$$

For $1 \leq i \leq m+2n-1$, we define

$$\mathcal{A}_{\mathsf{post}}^{(i)} = \{\gamma^{(i)} \mid \gamma \in \mathsf{post}_{i-1}\}$$

For $1 \leq i \leq m+2n-1$ the ABox $\mathcal{A}_{\mathsf{min}}^{(i)}$ only contains

1. the following assertions for every $a \in \mathsf{Obj}$ and every primitive concept name A in $\mathcal{T}$ in the input:

$$\begin{aligned} a &: (A^{(i-1)} \rightarrow A^{(i)}) \text{ if } \neg A(a) \notin \mathsf{post}_{i-1} \\ a &: (\neg A^{(i-1)} \rightarrow \neg A^{(i)}) \text{ if } A(a) \notin \mathsf{post}_{i-1} \end{aligned}$$

[2]There are other names introduced in the reduction. The intuition of those names is given in detail in [4]. Note that the auxiliary individual name a_{help} and ABox $\mathcal{A}_{\mathsf{aux}}$ and role names r_a used in [4] are not necessary here since we consider only unconditional actions.

$$
\begin{array}{rcl}
T_A^{(i)} & \equiv & (N \sqcap A^{(i)}) \sqcup (\neg N \sqcap A^{(0)}) \text{ for a primary concept name } A \text{ w.r.t. } \mathcal{T} \\
T_{\{a\}}^{(i)} & \equiv & \{a\} \\
T_{\neg C}^{(i)} & \equiv & \neg T_C^{(i)} \\
T_{C \sqcap D}^{(i)} & \equiv & T_C^{(i)} \sqcap T_D^{(i)} \\
T_{C \sqcup D}^{(i)} & \equiv & T_C^{(i)} \sqcup T_D^{(i)} \\
T_{\exists r.C}^{(i)} & \equiv & \Big(N \sqcap \big((\exists r^{(0)}.(\neg N \sqcap T_C^{(i)})) \sqcup (\exists r^{(i)}.(N \sqcap T_C^{(i)}))\big)\Big) \\
 & & \sqcup (\neg N \sqcap \exists r^{(0)}.T_C^{(i)})) \\
T_{\forall r.C}^{(i)} & \equiv & \Big(N \rightarrow \big((\forall r^{(0)}.(\neg N \rightarrow T_C^{(i)})) \sqcap (\forall r^{(i)}.(N \rightarrow T_C^{(i)}))\big)\Big) \\
 & & \sqcap (\neg N \rightarrow \forall r^{(0)}.T_C^{(i)}) \\
T_{(\geqslant n\ r\ C)}^{(i)} & \equiv & \Big(N \sqcap \bigsqcup_{0 \leq j \leq \min\{n, \#\mathsf{Obj}\}} \big((\geqslant j\ r^{(i)}\ (N \sqcap T_C^{(i)})) \sqcap \\
 & & \qquad (\geqslant (n-j)\ r^{(0)}\ (\neg N \sqcap T_C^{(i)}))\big)\Big) \\
 & & \sqcup (\neg N \sqcap (\geqslant n\ r^{(0)}\ T_C^{(i)})) \\
T_{(\leqslant n\ r\ C)}^{(i)} & \equiv & \Big(N \rightarrow \bigsqcap_{0 \leq j \leq \min\{n+1, \#\mathsf{Obj}\}} \big(\neg(\geqslant j\ r^{(i)}\ (N \sqcap T_C^{(i)})) \sqcup \\
 & & \qquad \neg(\geqslant (n-j)\ r^{(0)}\ (\neg N \sqcap T_C^{(i)}))\big)\Big) \\
 & & \sqcap (\neg N \rightarrow (\leqslant n\ r^{(0)}\ T_C^{(i)}))
\end{array}
$$

Figure 2: Concept definitions in $\mathcal{T}_{\mathsf{Sub}}^{(i)}$.

2. the following assertions for all $a, b \in \mathsf{Obj}$ and every role name r in the input:

$$
\begin{array}{rcl}
a & : & (\exists r^{(i-1)}.\{b\} \rightarrow \exists r^{(i)}.\{b\}) \text{ if } \neg r(a,b) \notin \mathsf{post}_{i-1} \\
a & : & (\forall r^{(i-1)}.\neg\{b\} \rightarrow \forall r^{(i)}.\neg\{b\}) \text{ if } r(a,b) \notin \mathsf{post}_{i-1}.
\end{array}
$$

The ABox $\mathcal{A}_{\mathsf{ini}}$ is defined as follows:

$$\mathcal{A}_{\mathsf{ini}} = \{\varphi^{(0)} \mid \varphi \in \mathcal{A}\}.$$

Then, we construct $\mathcal{A}_{\mathsf{red}}$:

$$\mathcal{A}_{\mathsf{red}} = \mathcal{A}_{\mathsf{ini}} \cup \bigcup_{i=1}^{m+2n-1} \mathcal{A}_{\mathsf{post}}^{(i)} \cup \bigcup_{i=1}^{m+2n-1} \mathcal{A}_{\mathsf{min}}^{(i)}.$$

As revealed in [4], from every model of $\mathcal{T}_{\mathsf{red}}$ and $\mathcal{A}_{\mathsf{red}}$ we can construct the crucial part of a DL-LTL structure generated by w from $\mathcal{A}$ w.r.t. $\mathcal{T}$ and vice versa.

Lemma 9. *Let $(\mathcal{T}, \mathcal{A}, \mathcal{B}_w, \phi)$ be an input of the satisfiability problem. Let $\mathcal{A}_{\mathsf{red}}$ and $\mathcal{T}_{\mathsf{red}}$ be respectively the ABox and the TBox obtained according to the above construction using $w = \alpha_1 \dots \alpha_m \beta_1 \dots \beta_n$ is the only word accepted by $\mathcal{B}_w$. Then, we have*

- *for every sequence* $\mathcal{I}_0, \ldots, \mathcal{I}_{m+2n-1}$ *of models of* $\mathcal{T}$ *such that* $\mathcal{I}_0 \models \mathcal{A}$ *and* $\mathcal{I}_i \Rightarrow^{\mathcal{T}}_{w(i)} \mathcal{I}_{i+1}$ *for every* i *with* $0 \leq i < m + 2n - 1$*, there exists an interpretation* $\mathcal{J}$ *such that* $\mathcal{J} \models \mathcal{A}_{\mathsf{red}}$*,* $\mathcal{J} \models \mathcal{T}_{\mathsf{red}}$*, and for all* $i \in \{0, \ldots, m+2n-1\}$ *and for all assertions* γ *in the input,* $\mathcal{I}_i \models \gamma$ *iff* $\mathcal{J} \models \gamma^{(i)}$*.*
- *for every interpretation* $\mathcal{J}$ *such that* $\mathcal{J} \models \mathcal{A}_{\mathsf{red}}$*,* $\mathcal{J} \models \mathcal{T}_{\mathsf{red}}$*, there exists a sequence* $\mathcal{I}_0, \ldots, \mathcal{I}_{m+2n-1}$ *of models of* $\mathcal{T}$ *such that* $\mathcal{I}_0 \models \mathcal{A}$*, and for every* i *with* $0 \leq i < m + 2n - 1$*, we have* $\mathcal{I}_i \Rightarrow^{\mathcal{T}}_{w(i)} \mathcal{I}_{i+1}$ *and for all assertions* γ *in the input,* $\mathcal{I}_i \models \gamma$ *iff* $\mathcal{J} \models \gamma^{(i)}$*.*

Employing this property of $\mathcal{A}_{\mathsf{red}}$ and $\mathcal{T}_{\mathsf{red}}$, we can also check executability of w. Define $\mathcal{A}_{\mathsf{pre}}$ as follows:

$$\mathcal{A}_{\mathsf{pre}} = \bigcup_{0 \leq j < m+2n-1} \{\gamma^{(j)} \mid \gamma \in \mathsf{pre}_j\}.$$

The tableau rules displayed in Figure 3 try to satisfy the semantics of LTL operators in the DL-LTL formula ϕ, where in $\vee$-rule we have:

$$\begin{aligned} \mathcal{B}' &= (\mathcal{A} \setminus \{(\varphi_1 \vee \varphi_2)^{(i)}\}) \cup \{\varphi_1^{(i)}\}, \text{ and} \\ \mathcal{B}'' &= (\mathcal{A} \setminus \{(\varphi_1 \vee \varphi_2)^{(i)}\}) \cup \{\varphi_2^{(i)}\}; \end{aligned}$$

in U-rule$_1$ and U-rule$_2$, we have:

$$\begin{aligned} \mathcal{B}_k &= (\mathcal{A} \setminus \{(\varphi_1 \mathsf{U} \varphi_2)^{(i)}\}) \cup \{\varphi_1^{(i)}, \ldots, \varphi_1^{(k-1)}, \varphi_2^{(k)}\} \\ &\quad \text{for all } k \text{ with } i \leq k < m + 2n, \text{ and} \\ \mathcal{B}_k &= (\mathcal{A} \setminus \{(\varphi_1 \mathsf{U} \varphi_2)^{(i)}\}) \cup \{\varphi_1^{(i)}, \ldots, \varphi_1^{(m+2n-1)}, \varphi_1^{(m+n)}, \ldots, \varphi_1^{(k-1)}, \varphi_2^{(k)}\}, \\ &\quad \text{for all } k \text{ with } m + n \leq k < i; \end{aligned}$$

and in R-rule$_1$ and R-rule$_2$ we have:

$$\begin{aligned} \mathcal{B}_k &= (\mathcal{A} \setminus \{(\varphi_1 \mathsf{R} \varphi_2)^{(i)}\}) \cup \{\varphi_2^{(i)}, \ldots, \varphi_2^{(k)}, \varphi_1^{(k)}\}, \\ &\quad \text{for all } k \text{ with } i \leq k < m + 2n, \\ \mathcal{B}_1^{\infty} &= \{\varphi_2^{(i)}, \ldots, \varphi_2^{(m+2n-1)}\}, \\ \mathcal{B}_k &= (\mathcal{A} \setminus \{(\varphi_1 \mathsf{U} \varphi_2)^{(i)}\}) \cup \{\varphi_2^{(i)}, \ldots, \varphi_2^{(m+2n-1)}, \varphi_2^{(m+n)}, \ldots, \varphi_2^{(k)}, \varphi_1^{(k)}\}, \\ &\quad \text{for all } k \text{ with } n + m \leq k < i, \text{ and} \\ \mathcal{B}_2^{\infty} &= \{\varphi_2^{(m+n)}, \ldots, \varphi_2^{(m+2n-1)}\}. \end{aligned}$$

As we can see, the tableau rules work on a set of sets of DL-LTL formulas. Each formula is labeled with (i). Intuitively, the label stands for the time point, e.g., $\psi^{(i)}$ can be read as the formula ψ holds at time point i. We apply exhaustively the tableau rules to $\mathfrak{S} = \{\{\phi^{(0)}\}\}$. The following lemma tells us that every application of a tableau rule preserves satisfiability of the formula the rule applies to:

$$\frac{\mathcal{A} \in \mathfrak{S} \wedge (\varphi_1 \wedge \varphi_2)^{(i)} \in \mathcal{A}}{\mathcal{A} := (\mathcal{A} \setminus \{(\varphi_1 \wedge \varphi_2)^{(i)}\}) \cup \{\varphi_1^{(i)}, \varphi_2^{(i)}\}} \ \wedge\text{-rule}$$

$$\frac{\mathcal{A} \in \mathfrak{S} \wedge (\varphi_1 \vee \varphi_2)^{(i)} \in \mathcal{A}}{\mathfrak{S} := (\mathfrak{S} \setminus \{\mathcal{A}\}) \cup \{\mathcal{B}', \mathcal{B}''\}} \ \vee\text{-rule}$$

$$\frac{\mathcal{A} \in \mathfrak{S} \wedge (\mathsf{X}\varphi)^{(i)} \in \mathcal{A} \wedge i < m + 2n - 1}{\mathcal{A} := \mathcal{A} \setminus \{(\mathsf{X}\varphi)^{(i)}\} \cup \{\varphi^{(i+1)}\}} \ \mathsf{X}\text{-rule}_1$$

$$\frac{\mathcal{A} \in \mathfrak{S} \wedge (\mathsf{X}\varphi)^{(i)} \in \mathcal{A} \wedge i = m + 2n - 1}{\mathcal{A} := \mathcal{A} \setminus \{(\mathsf{X}\varphi)^{(i)}\} \cup \{\varphi^{(n+m)}\}} \ \mathsf{X}\text{-rule}_2$$

$$\frac{\mathcal{A} \in \mathfrak{S} \wedge (\varphi_1 \mathsf{U} \varphi_2)^{(i)} \in \mathcal{A} \wedge i \leq m + n}{\mathfrak{S} := \mathfrak{S} \setminus \{\mathcal{A}\} \cup \{\mathcal{B}_i, \ldots, \mathcal{B}_{m+2n-1}\}} \ \mathsf{U}\text{-rule}_1$$

$$\frac{\mathcal{A} \in \mathfrak{S} \wedge (\varphi_1 \mathsf{U} \varphi_2)^{(i)} \in \mathcal{A} \wedge i > m + n}{\mathfrak{S} := \mathfrak{S} \setminus \{\mathcal{A}\} \cup \{\mathcal{B}_{m+n}, \ldots, \mathcal{B}_{m+2n-1}\}} \ \mathsf{U}\text{-rule}_2$$

$$\frac{\mathcal{A} \in \mathfrak{S} \wedge (\varphi_1 \mathsf{R} \varphi_2)^{(i)} \in \mathcal{A} \wedge i \leq m + n}{\mathfrak{S} := \mathfrak{S} \setminus \{\mathcal{A}\} \cup \{\mathcal{B}_i, \ldots, \mathcal{B}_{m+2n-1}, \mathcal{B}_1^\infty\}} \ \mathsf{R}\text{-rule}_1$$

$$\frac{\mathcal{A} \in \mathfrak{S} \wedge (\varphi_1 \mathsf{R} \varphi_2)^{(i)} \in \mathcal{A} \wedge i > m + n}{\mathfrak{S} := \mathfrak{S} \setminus \{\mathcal{A}\} \cup \{\mathcal{B}_{m+n}, \ldots, \mathcal{B}_{m+2n-1}, \mathcal{B}_2^\infty\}} \ \mathsf{R}\text{-rule}_2$$

Figure 3: Tableau rules.

Lemma 10. *Let $\mathfrak{S}$ be the set in some status of the tableau algorithm starting with $\{\{\phi^{(0)}\}\}$. $\mathfrak{S}'$ is obtained from $\mathfrak{S}$ by an application of one of the tableau rules to $\mathcal{A}_l \in \mathfrak{S}$. Then for every DL-LTL structure $\mathfrak{I} = (\mathcal{I}_i)_{i=0,1,\ldots}$ generated by w from $\mathcal{A}$ w.r.t. $\mathcal{T}$, the following statements are equivalent:*

- *$\mathfrak{I}, i \models \varphi$ for all $\varphi^{(i)} \in \mathcal{A}_l$.*
- *there exists an new element $\mathcal{B}_k \in \mathfrak{S}'$ such that $\mathfrak{I}, i \models \varphi$ for all $\varphi^{(i)} \in \mathcal{B}_k$.*

Proof. It is obvious for $\wedge$-rule and $\vee$-rule. By Lemma 7, $\mathfrak{I}$ is of the following form:

$$(\mathcal{I}_0, \ldots, \mathcal{I}_{m+n}, \ldots, \mathcal{I}_{m+2n-1}, \mathcal{I}_{m+n}, \ldots, \mathcal{I}_{m+2n-1}, \mathcal{I}_{m+n}, \ldots).$$

Thus, it follows that the above statements are equivalent for $\mathsf{X}\text{-rule}_1$ and $\mathsf{X}\text{-rule}_2$.

$\mathsf{U}\text{-rule}_1$: Consider the removed formula $(\varphi_1 \mathsf{U} \varphi_2)^{(i)} \in \mathcal{A}_l$. Then we know that for all $i \leq m + n$, $\mathfrak{I}, i \models \varphi_1 \mathsf{U} \varphi_2$ iff (by the semantics of U) there exists a $k \geq i$ such

that $\mathfrak{I}, k \models \varphi_2$ and $\mathfrak{I}, j \models \varphi_1$ for all $i \leq j < k$ iff (from the form of $\mathfrak{I}$ we know that k must be smaller than $m + 2n$) there exists a k with $i \leq k \leq m + 2n - 1$ such that $\mathfrak{I}, k \models \varphi_2$ and $\mathfrak{I}, j \models \varphi_1$ for all j with $i \leq j < k$, i.e., there is one new added $\mathcal{B}_k$ such that $\mathfrak{I}, i \models \varphi$ for all $\varphi^{(i)} \in \mathcal{B}_k \setminus \mathcal{A}_l$.

U-rule$_2$: Consider the removed formula $(\varphi_1 \mathsf{U} \varphi_2)^{(i)} \in \mathcal{A}_l$. Then we know that for all $i > m + n$, $\mathfrak{I}, i \models \varphi_1 \mathsf{U} \varphi_2$ iff (by the semantics of U) there exists a $k \geq i$ such that $\mathfrak{I}, k \models \varphi_2$ and $\mathfrak{I}, j \models \varphi_1$ for all $i \leq j < k$ iff (from the form of $\mathfrak{I}$ we know that k must be between $m + n$ and $m + 2n$) there exists a k with $i \leq k \leq m + 2n - 1$ such that $\mathfrak{I}, k \models \varphi_2$ and $\mathfrak{I}, j \models \varphi_1$ for all j with $i \leq j < k$ or there exists a k with $m + n \leq k < i$ such that $\mathfrak{I}, k \models \varphi_2$ and $\mathfrak{I}, j \models \varphi_1$ for all j with $i \leq j \leq m + 2n$ and for all j with $m + n \leq j < k$ iff there is one new added $\mathcal{B}_k$ such that $\mathfrak{I}, i \models \varphi$ for all $\varphi^{(i)} \in \mathcal{B}_k \setminus \mathcal{A}_l$.

Similarly, the form of $\mathfrak{I}$, together with the semantics of R operator, implies that the two statements in the lemma are equivalent if either of R-rule$_1$ and R-rule$_2$ is applied. ❑

After the tableau algorithm terminates with $\mathfrak{S}$, for every $\mathcal{A}$ in $\mathfrak{S}$, every formula in $\mathcal{A}$ is an ABox assertion and the function defined in (1) can applied to those assertions.[3] Thus, every element in $\mathfrak{S}$ can be viewed as an ABox. Then, we use the set $\mathfrak{S}$, together with the constructed $\mathcal{T}_{\mathsf{red}}$, $\mathcal{A}_{\mathsf{red}}$, and $\mathcal{A}_{\mathsf{pre}}$ to decide whether ϕ is satisfiable w.r.t. $\mathcal{T}$, $\mathcal{A}$, and $\mathcal{B}_w$.

Lemma 11. *Let $\mathfrak{S}$ be the set when the tableau algorithm terminates. Then ϕ is satisfiable w.r.t. $\mathcal{T}$, $\mathcal{A}$, and $\mathcal{B}_w$ iff there is an $\mathcal{A}_\phi \in \mathfrak{S}$ such that $\mathcal{A}_{\mathsf{red}} \cup \mathcal{A}_{\mathsf{pre}} \cup \mathcal{A}_\phi$ is consistent w.r.t. $\mathcal{T}_{\mathsf{red}}$.*

Proof. "⇒": If φ is satisfiable w.r.t. $\mathcal{T}$, $\mathcal{A}$, and $\mathcal{B}_w$ then there is a DL-LTL structure $\mathfrak{I} = (\mathcal{I}_i)_{i=0,1,\dots}$ generated by w from $\mathcal{A}$ w.r.t. $\mathcal{T}$ such that $\mathfrak{I}, 0 \models \mathcal{A}$. By Point 1 of Lemma 9, there exists an interpretation $\mathcal{J}$ such that $\mathcal{J} \models \mathcal{A}_{\mathsf{red}}$, $\mathcal{J} \models \mathcal{T}_{\mathsf{red}}$ and for all $i \in \{0, \dots, m + 2n - 1\}$ and for all assertions γ in the input, $\mathcal{I}_i \models \gamma$ iff $\mathcal{J} \models \gamma^{(i)}$. Since $\mathcal{I}_i \models \mathsf{pre}_i$ for all i with $0 \leq i \leq m + 2n - 1$, we obtain that $\mathcal{J} \models \mathcal{A}_{\mathsf{pre}}$. By Lemma 10, $\mathfrak{I}, 0 \models \phi$ implies there exists $\mathcal{A}_\phi \in \mathfrak{S}$ such that $\mathfrak{I}, i \models \varphi$ for all $\varphi^{(i)} \in \mathcal{A}_\phi$. Since for every $\varphi^{(i)} \in \mathcal{A}_\phi$, φ is an assertion, $\mathfrak{I}, i \models \varphi$ yields $\mathcal{I}_i \models \varphi$. Hence, $\mathcal{J} \models \mathcal{A}_\phi$.

"⇐": Let $\mathcal{J}$ be a model of $\mathcal{A}_{\mathsf{red}} \cup \mathcal{A}_{\mathsf{pre}} \cup \mathcal{A}_\phi$ and $\mathcal{T}_{\mathsf{red}}$ with some $\mathcal{A}_\phi \in \mathfrak{S}$. Then it follows from Point 2 of Lemma 9 that there exist $\mathcal{I}_0, \dots, \mathcal{I}_{n+2m-1}$ such that $\mathcal{I}_0 \models \mathcal{A}$ and $\mathcal{I}_i \Rightarrow^{\mathcal{T}}_{w(i)} \mathcal{I}_{i+1}$ for all i with $0 \leq i < m + 2n - 1$ and for all assertions γ in the input and for all i with $0 \leq i < m + 2n - 1$, $\mathcal{I}_i \models \gamma$ iff $\mathcal{J} \models \gamma^{(i)}$. Define $\mathfrak{I}$ as follows:

$$(\mathcal{I}_0, \dots, \mathcal{I}_{m+n}, \dots, \mathcal{I}_{m+2n-1}, \mathcal{I}_{m+n}, \dots, \mathcal{I}_{m+2n-1}, \mathcal{I}_{m+n}, \dots).$$

[3]The termination of the tableau algorithm will be addressed later on when we analyze the complexity.

By the definition of $\mathcal{A}_{\mathsf{pre}}$, we know that $\mathcal{I}_i \models \mathsf{pre}_i$ for all i with $0 \leq i < m+2n-1$, which implies that $\mathcal{I}_i \models \mathsf{pre}_i$ for all $i \geq 0$. By Lemma 7, $\mathfrak{I}$ is a DL-LTL structure generated by w from $\mathcal{A}$ w.r.t. $\mathcal{T}$. Moreover, $\mathcal{J} \models \mathcal{A}_\phi$ implies for all $\psi^{(i)} \in \mathcal{A}_\phi$, $\mathcal{I}_i \models \psi$, i.e., $\mathfrak{I}, i \models \psi$. By Lemma 10, we have $\mathfrak{I}, 0 \models \phi$. ❑

For arbitrary $\mathcal{T}$, $\mathcal{A}$, $\mathcal{B}_w$, and ϕ, the size of $\mathcal{A}_{\mathsf{red}}$ and $\mathcal{T}_{\mathsf{red}}$ is polynomial in the size of input and they can be constructed in polynomial time. This is independent of the codings of numbers in the number restrictions in the input [9]. It is clear that $\mathcal{A}_{\mathsf{pre}}$ has those properties as well. In general, the size of $\mathfrak{S}$ can be exponential in the size of the input. However, we need only one element $\mathcal{A}_\phi$ in $\mathfrak{S}$ such that $\mathcal{A}_{\mathsf{red}} \cup \mathcal{A}_{\mathsf{pre}} \cup \mathcal{A}_\varphi \cup \mathcal{T}_{\mathsf{red}}$ is consistent. For a DL-LTL formula ϕ, $\mathcal{A}_\phi$ can be constructed in NPSPACE since

- each application of a tableau rule generates at most only $(m + 2n)$ (i.e., polynomially many) sets of labeled formulas;
- every labeled formula in generated sets is a strict subformula of the formula that the rule applies to and i in all labels (i) is never over $m + 2n - 1$;
- there is a tableau rule applicable iff there is an LTL operator in $\mathfrak{S}$.

By Savitch's theorem [10], the construction of $\mathcal{A}_\phi$ can be done in PSPACE. Overall, $\mathcal{A}_{\mathsf{red}}$, $\mathcal{A}_{\mathsf{pre}}$, $\mathcal{A}_\phi$ and $\mathcal{T}_{\mathsf{red}}$ can be constructed in PSPACE. Consistency checking of an $\mathcal{ALCQO}$-ABox w.r.t. an $\mathcal{ALCQO}$-TBox is in PSPACE [5] if the numbers in qualified number restriction are coded in unary. For $\mathcal{ALCIO}$, it is in EXPTIME [1]. For $\mathcal{ALCQIO}$, a fragment of C^2, it is in NEXPTIME [16, 13], even if the numbers are in binary coding. Thus, we obtained an upper bound of the satisfiability problem for DLs between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$.

Lemma 12. *The satisfiability problem of DL-LTL formulas w.r.t. acyclic TBoxes, ABoxes, and Büchi automata (with the restriction specified at the beginning of this section) is*

- *in* PSPACE *for* $\mathcal{ALCQO}$ *if the numbers in qualified number restriction are coded in unary;*
- *in* EXPTIME *for* $\mathcal{ALCIO}$*;*
- *in* NEXPTIME *for* $\mathcal{ALCQIO}$*.*

In what follows, we show that those upper bounds are tight by reducing the projection problem to the (un)satisfiability problem.

Definition 13 (The projection problem)**.** Let $\mathcal{T}$ be an acyclic TBox, $\alpha_1 \cdots \alpha_m$ a finite sequence of actions for $\mathcal{T}$, and $\mathcal{A}$ an ABox. An assertion φ is a *consequence of applying* $\alpha_1 \cdots \alpha_m$ *in* $\mathcal{A}$ *w.r.t.* $\mathcal{T}$ iff for all models of $\mathcal{A}$ and $\mathcal{T}$, and all $\mathcal{I}'$ with $\mathcal{I} \Rightarrow^{\mathcal{T}}_{\alpha_1 \cdots \alpha_m} \mathcal{I}'$, we have $\mathcal{I}' \models \varphi$. $\triangle$

It has been shown in [4] that for DLs $\mathcal{L}$ between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$, the projection problem in $\mathcal{L}$ is as hard as the (in)consistency problem in $\mathcal{LO}$ even if every action has empty set of pre-conditions and occlusions and unconditional post-conditions.

We can reduce the projection problem in $\mathcal{L}$ to the validity problem of DL-LTL formulas w.r.t. acyclic TBoxes, ABoxes, and Büchi sequences of actions in $\mathcal{L}$. Let $\mathcal{A}$ be an ABox and $\alpha_i = (\emptyset, \emptyset, \mathsf{post}_i)$ an unconditional action for an TBox $\mathcal{T}$ for all i with $1 \leq i \leq m$. It is easy to see that an assertion φ is a consequence of applying $\alpha_1 \ldots \alpha_m$ in $\mathcal{A}$ w.r.t. $\mathcal{T}$ iff $\mathsf{X}^m\varphi$ is valid w.r.t. $\mathcal{T}$, $\mathcal{A}$, and $\mathcal{B}_w$ with $w = \alpha_1 \ldots \alpha_m \beta_1^\omega$ and $\beta_1 = (\emptyset, \emptyset, \emptyset)$ (in which X^m is the abbreviation of number m of Xs). Thus, the complexity results about the projection problem in [4] imply the following lemma:

Lemma 14. *The validity problem of DL-LTL formulas w.r.t. acyclic TBoxes, ABoxes, and Büchi automata (with the restriction specified at the beginning of this section) is*

- PSpace*-hard for* $\mathcal{ALC}$*;*
- ExpTime*-hard for* $\mathcal{ALCI}$*;*
- *co-*NExpTime*-hard for* $\mathcal{ALCQI}$*.*

The above lemma does not rely on the coding of numbers. Recall that the validity problem can be further reduced to the (un)satisfiability problem. Thus,

Theorem 15. *The satisfiability problem (and the complement of the validity problem) of DL-LTL formulas w.r.t. acyclic TBoxes, ABoxes, and Büchi automata (with the restriction specified at the beginning of this section) for the DL* $\mathcal{L}$ *is*

- PSpace*-complete if* $\mathcal{L}$ *is in* $\{\mathcal{ALC}, \mathcal{ALCO}, \mathcal{ALCQ}, \mathcal{ALCQO}\}$ *and the numbers in qualified number restriction are coded in unary;*
- ExpTime*-complete if* $\mathcal{L}$ *is in* $\{\mathcal{ALCI}, \mathcal{ALCIO}\}$*;*
- NExpTime*-complete if* $\mathcal{L}$ *is in* $\{\mathcal{ALCQI}, \mathcal{ALCQIO}\}$.

4 The General Case

Now, we consider arbitrary Büchi automata and (possibly) conditional actions. In this setting, we cannot use the approach introduced in the previous section. On the one hand, it is easy to see that, for conditional actions, the crucial Lemma 8 need not hold. On the other hand, while any non-empty language accepted by a Büchi automaton contains a cyclic word, it may also contain non-cyclic ones. Thus, it is not a propri clear whether a cyclic word can be taken as the word $w \in L_\omega(\mathcal{B})$ required by the definition of the satisfiability problem.

Our approach for solving satisfiablity of a DL-LTL formula ϕ w.r.t. an acyclic TBox $\mathcal{T}$, an ABox $\mathcal{A}$, and a Büchi automaton $\mathcal{B}$ over an alphabet Σ of (possibly) conditional actions is based on the approach for deciding satisfiablity in $\mathcal{ALC}$-LTL introduced in [3]. Given a DL-LTL formula ϕ to be tested for satisfiability, this approach builds the *propositional abstraction* $\widehat{\phi}$ of ϕ by replacing each assertion[4] γ occurring in ϕ by a corresponding propositional letter p_γ. Let $\mathcal{L}$ be the set of propositional letters used for the abstraction. Consider a set $\mathcal{S} \subseteq \mathcal{P}(\mathcal{L})$, i.e., a set of subsets of $\mathcal{L}$. Such a set induces the following (propositional) LTL formula:

$$\widehat{\phi}_{\mathcal{S}} := \widehat{\phi} \wedge \Box \left(\bigvee_{X \in \mathcal{S}} \left(\bigwedge_{p \in X} p \wedge \bigwedge_{p \notin X} \neg p \right) \right)$$

Intuitively, this formula is satisfiable if there exists a propositional LTL structure satisfying $\widehat{\phi}$ in which, at every time point, the set of propositional letters satisfied at this time point is one of the sets $X \in \mathcal{S}$. To get satisfiability of ϕ from satisfiability of $\widehat{\phi}_{\mathcal{S}}$ for some $\mathcal{S} \subseteq \mathcal{P}(\mathcal{L})$, we must check whether the sets of assertions induced by the sets $X \in \mathcal{S}$ are consistent. To be more precise, assume that a set $\mathcal{S} = \{X_1, \ldots, X_k\} \subseteq \mathcal{P}(\mathcal{L})$ is given. For every $i, 1 \leq i \leq k$, and every concept name A (role name r) occurring in ϕ, we introduce a copy $A^{(i)}$ ($r^{(i)}$). We call $A^{(i)}$ ($r^{(i)}$) the ith copy of A (r). The assertion $\gamma^{(i)}$ is obtained from γ by replacing every occurrence of a concept or role name by its ith copy. The set $\mathcal{S} = \{X_1, \ldots, X_k\}$ induces the following ABox:

$$\mathcal{A}_{\mathcal{S}} := \bigcup_{1 \leq i \leq k} \{\gamma^{(i)} \mid p_\gamma \in X_i\} \cup \{\neg\gamma^{(i)} \mid p_\gamma \notin X_i\}.$$

The following lemma is proved in [3].

Lemma 16. *The DL-LTL formula ϕ is satisfiable iff there is a set $\mathcal{S} \subseteq \mathcal{P}(\mathcal{L})$ such that the propositional LTL formula $\widehat{\phi}_{\mathcal{S}}$ is satisfiable and the ABox $\mathcal{A}_{\mathcal{S}}$ is consistent (w.r.t. the empty TBox).*

[4] In [3], both assertions and GCIs need to be replaced. In the present paper, GCIs are not allowed to occur in LTL formulae, and thus we need to deal only with assertions.

Now, we show how we can use this approach to solve the satisfiability problem introduced in Definition 6, i.e., satisfiability of a DL-LTL formula ϕ w.r.t. an acyclic TBox $\mathcal{T}$, an ABox $\mathcal{A}$, and a Büchi automaton $\mathcal{B}$ over an alphabet Σ of (possibly) conditional actions. First, note that Lemma 16 also holds if we formulate it for DL-LTL formulae, with a DL between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$, rather than $\mathcal{ALC}$-LTL formulae. However, the existence of a set $\mathcal{S} \subseteq \mathcal{P}(\mathcal{L})$ such that $\widehat{\phi}_{\mathcal{S}}$ is satisfiable and the ABox $\mathcal{A}_{\mathcal{S}}$ is consistent is not enough to have satisfiability of ϕ w.r.t. $\mathcal{T}$, $\mathcal{A}$, and $\mathcal{B}$. In fact, the existence of such a set only yields a DL-LTL structure $\mathfrak{I} = (\mathcal{I}_i)_{i=0,1,\ldots}$ satisfying ϕ. We also need to ensure (i) that $\mathcal{I}_0$ is a model of $\mathcal{A}$ and (ii) that there is an infinite word $w \in L_\omega(\mathcal{B})$ such that, for all $i \geq 0$, the transition from $\mathcal{I}_i$ to $\mathcal{I}_{i+1}$ is caused by the action $w(i)$ and $\mathcal{I}_i$ is a model of $\mathcal{T}$.

Ensuring that $\mathcal{I}_0$ is a model of $\mathcal{A}$ is easy since $\mathcal{A}$ can be encoded in the DL-LTL formula by working with the formula $\phi \wedge \bigwedge_{\gamma \in \mathcal{A}} \gamma$ instead of ϕ. For this reason, we will assume in the following (without loss of generality) that *the ABox $\mathcal{A}$ is empty.*

To deal with the second issue, we introduce corresponding propositional letters p_γ not only for the assertions γ occurring in ϕ, but also for (i) the assertions γ occurring in the actions in Σ, and (ii) the assertions γ of the form $A(a)$ and $r(a, b)$ where A, r, a, b occur in ϕ, $\mathcal{T}$, or an action in Σ, A is a concept name that is primitive in $\mathcal{T}$, r is a role name, and a, b are individual names. We call the assertions introduced in (ii) *primitive assertions*. In the following, let $\mathcal{L}$ be the (finite) set of propositional letters obtained this way. Obviously, Lemma 16 still holds if we use this larger set of propositional letters to build the sets $\mathcal{S}$ and the formulae $\widehat{\phi}_{\mathcal{S}}$.

One way of deciding satisfiability of a propositional LTL formula $\widehat{\phi}$ is to construct a Büchi automaton $\mathcal{C}_{\widehat{\phi}}$ that accepts the propositional LTL structures satisfying $\widehat{\phi}$ [18]. To be more precise, let $\Gamma := \mathcal{P}(\mathcal{L})$. A propositional LTL structure $\widehat{\mathfrak{I}} = (w_i)_{i=0,1,\ldots}$ is an infinite sequence of truth assignments to the propositional letters from $\mathcal{L}$. Such a structure can be represented by an infinite word $X = X(0)X(1)\ldots$ over Γ, where $X(i)$ consists of the propositional variables that w_i makes true. The Büchi automaton $\mathcal{C}_{\widehat{\phi}}$ is built such that it accepts exactly those infinite words over Γ that represent propositional LTL structures satisfying $\widehat{\phi}$. Consequently, $\widehat{\phi}$ is satisfiable iff the language accepted by $\mathcal{C}_{\widehat{\phi}}$ is non-empty. The size of $\mathcal{C}_{\widehat{\phi}}$ is exponential in the size of $\widehat{\phi}$, and the emptiness test for Büchi automata is polynomial in the size of the automaton. As sketched in [3], the automaton $\mathcal{C}_{\widehat{\phi}}$ can easily be modified into one accepting exactly the words representing propositional LTL structures satisfying $\widehat{\phi}_{\mathcal{S}}$. In fact, we just need to remove all transitions that use a letter from $\Gamma \setminus \mathcal{S}$. Obviously, this modification can be done in time polynomial in the size of $\mathcal{C}_{\widehat{\phi}}$, and thus in time exponential in the size of $\widehat{\phi}$. We denote the Büchi automaton obtained this way by $\mathcal{C}^{\mathcal{S}}_{\widehat{\phi}}$.

Lemma 17. *Let ϕ be a DL-LTL formula and $\mathcal{L}$ the set of propositional letters constructed as described above. Let $\mathcal{S}$ be a subset of $\mathcal{P}(\mathcal{L})$. We construct $\widehat{\phi}_{\mathcal{S}}$ and $\mathcal{C}^{\mathcal{S}}_{\widehat{\phi}}$ as above. Then for every propositional structure $\widehat{\mathfrak{I}} = (w_i)_{i=0,1,\ldots}$, $\widehat{\mathfrak{I}}, 0 \models \widehat{\phi}_{\mathcal{S}}$ iff the infinite word represented by $\widehat{\mathfrak{I}}$ is accepted by $\mathcal{C}^{\mathcal{S}}_{\widehat{\phi}}$.*

Now, consider the Büchi automaton $\mathcal{B}$ from the input, and assume that it is of the form $\mathcal{B} = (Q, \Sigma, I, \Delta, F)$, where Q is the set of states, $I \subseteq Q$ the set of initial states, $\Delta \subseteq Q \times \Sigma \times Q$ the transition relation, and $F \subseteq Q$ the set of final states. We use $\mathcal{B}$ to construct a Büchi automaton $\mathcal{B}' = (Q', \Gamma, I', \Delta', F')$ that accepts those infinite words $X = X(0)X(1)\ldots$ over the alphabet Γ for which there is an infinite word $w \in L_\omega(\mathcal{B})$ such that the difference between $X(i)$ and $X(i+1)$ is "caused by" the action $w(i)$:

- $\Gamma = \mathcal{P}(\mathcal{L})$;
- $Q' = Q \times \Sigma \times \Gamma$;
- $I' = I \times \Sigma \times \Gamma$;
- $((q, \alpha, X), Y, (q', \alpha', X')) \in \Delta'$ iff the following holds:
  1. $(q, \alpha, q') \in \Delta$;
  2. $X = Y$;
  3. Let $\alpha = (\mathsf{pre}, \mathsf{occ}, \mathsf{post})$.
     - $p_\gamma \in X$ for all $\gamma \in \mathsf{pre}$;
     - if $\beta/\gamma \in \mathsf{post}$ and $p_\beta \in X$ then $p_\gamma \in X'$;
     - for every primitive assertion γ, if $p_\gamma \in X$, $\gamma \notin \mathsf{occ}$, and there is no $\beta/\neg\gamma \in \mathsf{post}$ with $p_\beta \in X$, then $p_\gamma \in X'$;
     - for every primitive assertion γ, if $p_\gamma \notin X$, $\gamma \notin \mathsf{occ}$, and there is no $\beta/\gamma \in \mathsf{post}$ with $p_\beta \in X$, then $p_\gamma \notin X'$;
- $F' = F \times \Sigma \times \Gamma$.

The intersection of the languages $L_\omega(\mathcal{B}')$ and $L_\omega(\mathcal{C}^{\mathcal{S}}_{\widehat{\phi}})$ thus contains those infinite words $X = X(0)X(1)\ldots$ over the alphabet Γ (i) that represent propositional LTL structures satisfying $\widehat{\phi}_{\mathcal{S}}$, and (ii) for which there is an infinite word $w \in L_\omega(\mathcal{B})$ such that the difference between $X(i)$ and $X(i+1)$ is caused by the action $w(i)$, where the formal meaning of "caused by" is given by the conditions in Item 3 of the definition of $\mathcal{B}'$. Since the class of languages of infinite words accepted by Büchi automata is closed under intersection, there is a Büchi automaton $\mathcal{D}(\widehat{\phi}, \mathcal{S}, \mathcal{B})$ accepting this intersection. This automaton can be obtained from $\mathcal{B}'$ and $\mathcal{C}^{\mathcal{S}}_{\widehat{\phi}}$ by a product construction that is a bit more complicated, but not

more complex, than the construction for "normal" finite automata [15]. Thus, like $\mathcal{C}^{\mathcal{S}}_{\widehat{\phi}}$ and $\mathcal{B}'$, the automaton $\mathcal{D}(\widehat{\phi}, \mathcal{S}, \mathcal{B})$ is of size exponential in the size of the input.

Given a word $X = X(0)X(1)\ldots$ accepted by $\mathcal{D}(\widehat{\phi}, \mathcal{S}, \mathcal{B})$, we still cannot be sure that the propositional LTL structure represented by this word can be lifted to a DL-LTL structure generated by a word $w \in L_\omega(\mathcal{B})$ from the empty ABox w.r.t. $\mathcal{T}$. The first problem is that we must ensure that $X = X(0)X(1)\ldots$ can be lifted to a DL-LTL structure $\mathfrak{I} = (\mathcal{I}_i)_{i=0,1,\ldots}$ satisfying ϕ. By Lemma 16, this is the case if the ABox $\mathcal{A}_{\mathcal{S}}$ is consistent (w.r.t. the empty TBox). However, we will see below that we need to adapt the definition of $\mathcal{A}_{\mathcal{S}}$ in order to align it with the approach used to solve the second problem.

This second problem is that we need to ensure that $\mathcal{I}_i \Rightarrow^{\mathcal{T}}_{w(i)} \mathcal{I}_{i+1}$ holds for all $i \geq 0$.[5] Note that Item 3 in the definition of $\mathcal{B}'$ only enforces that the changes to the *named part* of the interpretation (i.e., for the domain elements interpreting individual names) are according to the action $w(i)$. It does not say anything about the *unnamed part* of the interpretation (which, according to the semantics of our actions, should not be modified) and it does not deal with the TBox. Fortunately, this is exactly what the TBox $\mathcal{T}_{\mathsf{red}}$ already used in the previous section is designed for. The idea is that every concept description C occurring in the input (directly or as subdescription) is represented by new concept names $T^{(i)}_C$ for $i = 1, \ldots, k$, where the index i corresponds to the set $X_i \in \mathcal{S}$. Recall that we already have copies $A^{(i)}, r^{(i)}$ $(i = 1, \ldots, k)$ for all concepts and role names occurring in the input. In addition, we now introduce an additional copy $A^{(0)}, r^{(0)}$. Intuitively, for every index i, we want to have an interpretation $\mathcal{I}_i$ that is a model of the ABox

$$\mathcal{A}_i = \{\gamma \mid p_\gamma \in X_i\} \cup \{\neg\gamma \mid p_\gamma \notin X_i\}$$

and of the input TBox $\mathcal{T}$, such that all these interpretations coincide on their unnamed parts. Now, for every concept name A (role name r), the copy $A^{(0)}$ $(r^{(0)})$ corresponds to the extension of A (r) on the unnamed part of $\mathcal{I}_i$ (which is the same for all i), and the copy $A^{(i)}$ $(r^{(i)})$ corresponds to the extension of A (r) on the named part of $\mathcal{I}_i$. For a concept description C, the concept name $T^{(i)}_C$ corresponds to the extension of C in $\mathcal{I}_i$ (both named and unnamed part). Let $\mathcal{S} = \{X_1, \ldots, X_k\}$. We define

$$\mathcal{T}_{\mathsf{red}} = \mathcal{T}_N \cup (\bigcup_{i=1}^{k} \mathcal{T}^{(i)}_{\mathsf{Sub}}) \cup \{T^{(i)}_A \equiv T^{(i)}_E \mid A \equiv E \in \mathcal{T}, 1 \leq i \leq k\},$$

where $\mathcal{T}_N$ and $\mathcal{T}^{(i)}_{\mathsf{Sub}}$ are defined as in the previous section. The TBox $\mathcal{T}_{\mathsf{red}}$ is defined such that, from a model of $\mathcal{T}_{\mathsf{red}}$, one can derive models $\mathcal{I}_i$ of $\mathcal{T}$ coinciding on their unnamed parts. To ensure that $\mathcal{I}_i$ is also a model of $\mathcal{A}_i$, we basically use the

[5] Recall that the definition of $\mathcal{I}_i \Rightarrow^{\mathcal{T}}_{w(i)} \mathcal{I}_{i+1}$ also includes the requirement that $\mathcal{I}_i$ must be a model of $\mathcal{T}$.

ABox $\mathcal{A}_\mathcal{S}$ introduced above with the only difference that $\gamma^{(i)}$ is defined as in (1) in the previous section, instead of the copy $\gamma^{(i)}$ used in [3] (see above). Let $\widehat{\mathcal{A}}_\mathcal{S}$ be the ABox obtained this way:

$$\widehat{\mathcal{A}}_\mathcal{S} = \bigcup_{1 \leq i \leq k} \{\gamma^{(i)} \mid p_\gamma \in X_i\} \cup \{\neg\gamma^{(i)} \mid p_\gamma \notin X_i\}.$$

We are now ready to formlate the main technical result of this section.

Lemma 18. *The DL-LTL formula ϕ is satisfiable w.r.t. $\mathcal{T}$, $\emptyset$, and $\mathcal{B}$ iff there is a set $\mathcal{S} \subseteq \mathcal{P}(\mathcal{L})$ such that $L_\omega(\mathcal{D}(\widehat{\phi}, \mathcal{S}, \mathcal{B})) \neq \emptyset$ and $\widehat{\mathcal{A}}_\mathcal{S}$ is consistent w.r.t. $\mathcal{T}_{\mathsf{red}}$.*

Proof. "⇒": Suppose that ϕ is satisfiable w.r.t. $\mathcal{T}$, $\mathcal{A}$, and $\mathcal{B}$. Then there exist a $w = \alpha_0\alpha_1 \cdots \in L_\omega(\mathcal{B})$ and a DL-LTL structure $\mathfrak{I} = (\mathcal{I}_i)_{i=0,1,\ldots}$ generated by w from $\mathcal{A}$ ($\mathcal{A} = \emptyset$ by our assumption) w.r.t. $\mathcal{T}$ such that $\mathfrak{I}, 0 \models \phi$. We define X_i for all $i \geq 0$ and $\mathcal{S}$ as follows:

$$X_i = \{p_\gamma \in \mathcal{L} \mid \mathcal{I}_i \models \gamma\} \text{ and } \mathcal{S} = \{X_i \mid i \in \mathbb{N}\}.$$

It follows from the definition of X_i that for all $i \geq 0$, for all $p_\gamma \in \mathcal{L}$, $p_\gamma \in X_i$ iff $\mathcal{I}_i \models \gamma$. Let $\widehat{\mathfrak{I}}$ be the propositional LTL structure $(X_i)_{i=0,1,\ldots}$. Then, for all $i \geq 0$,

$$\widehat{\mathfrak{I}}, i \models \bigwedge_{p \in X_i} p \wedge \bigwedge_{p \notin X_i} \neg p,$$

and thus,

$$\widehat{\mathfrak{I}}, i \models \bigvee_{X \in \mathcal{S}} \left(\bigwedge_{p \in X} p \wedge \bigwedge_{p \notin X} \neg p \right).$$

Hence,

$$\widehat{\mathfrak{I}}, 0 \models \Box \left(\bigvee_{X \in \mathcal{S}} \left(\bigwedge_{p \in X} p \wedge \bigwedge_{p \notin X} \neg p \right) \right).$$

Moreover, since $\mathfrak{I}, 0 \models \phi$, $\widehat{\mathfrak{I}}, 0 \models \widehat{\phi}$ (This can be shown by induction on the structure of ϕ). Thus, $\widehat{\mathfrak{I}}, 0 \models \widehat{\phi}_\mathcal{S}$. By Lemma 17, we have that $X_0X_1 \ldots \in L_\omega(\mathcal{C}^\mathcal{S}_{\widehat{\phi}})$.

We now show that $X_0X_1 \ldots$ is accepted by $\mathcal{B}'$. Since $w \in L_\omega(\mathcal{B})$, there exists an accepting run $q_0q_1 \ldots$ of $\mathcal{B}$ on w. We show that $(q_0, \alpha_0, X_0)(q_1, \alpha_1, X_1) \ldots$ is a run of $\mathcal{B}'$ on $X_0X_1 \ldots$: For all $i \geq 0$, we have

1. $(q_i, \alpha_i, q_{i+1}) \in \Delta$;
2. $X_i = X_i$;
3. if $\alpha_i = (\mathsf{pre}_i, \mathsf{occ}_i, \mathsf{post}_i)$ then

- $p_\gamma \in X_i$ for all $\gamma \in \mathsf{pre}_i$: since $\mathfrak{I}$ is a DL-LTL structure generated by w from $\mathcal{A}$ w.r.t. $\mathcal{T}$, $\mathcal{I}_i \models \mathsf{pre}_i$. Thus, for all $\gamma \in \mathsf{pre}_i$, $p_\gamma \in X_i$.
- if $\beta/\gamma \in \mathsf{post}_i$ and $p_\beta \in X_i$ then $p_\gamma \in X_{i+1}$: since $\mathfrak{I}$ is a DL-LTL structure generated by w from $\mathcal{A}$ w.r.t. $\mathcal{T}$, $\mathcal{I}_i \Rightarrow^{\mathcal{T}}_{\alpha_i} \mathcal{I}_{i+1}$. Thus, for every $\beta/\gamma \in \mathsf{post}_i$, we have if $\mathcal{I}_i \models \beta$ then $\mathcal{I}_{i+1} \models \gamma$. By the definition of X_i, $p_\beta \in X_i$ implies that $\mathcal{I}_i \models \beta$. Thus, $\mathcal{I}_{i+1} \models \gamma$, which implies $p_\gamma \in X_{i+1}$.
- for every primitive assertion γ, if $p_\gamma \in X_i$, $\gamma \notin \mathsf{occ}_i$, and there is no $\beta/\neg\gamma \in \mathsf{post}_i$ with $p_\beta \in X_i$, then $p_\gamma \in X_{i+1}$: since $\mathfrak{I}$ is a DL-LTL structure generated by w from $\mathcal{A}$ w.r.t. $\mathcal{T}$, $\mathcal{I}_i \Rightarrow^{\mathcal{T}}_{\alpha_i} \mathcal{I}_{i+1}$. Thus, for every primitive assertion γ, if $\mathcal{I}_i \models \gamma$, $\beta \notin \mathsf{occ}$, and there is no $\beta/\neg\gamma \in \mathsf{post}_i$ with $\mathcal{I}_i \models \beta$, we have $\mathcal{I}_{i+1} \models \gamma$. By the definition of X_i, $p_\gamma \in X_i$ implies that $\mathcal{I}_i \models \gamma$. Moreover, since there is no $\beta/\neg\gamma \in \mathsf{post}_i$ with $p_\beta \in X_i$, there is no $\beta/\neg\gamma \in \mathsf{post}_i$ with $\mathcal{I}_i \models \beta$. Thus, $\mathcal{I}_{i+1} \models \gamma$, which implies that $p_\gamma \in X_{i+1}$.
- for every primitive assertion γ, if $p_\gamma \notin X$ and there is no $\beta/\gamma \in \mathsf{post}$ with $p_\beta \in X$, then $p_\gamma \notin X'$: This can be shown similarly to the previous condition.

It follows from the fact that $q_0 q_1 \dots$ is accepting by $\mathcal{B}$ that the above run is accepting by $\mathcal{B}'$. Thus, $X_0 X_1 \dots$ is accepted by the automaton $\mathcal{D}(\widehat{\phi}, \mathcal{S}, \mathcal{B})$.

It remains to show that $\widehat{\mathcal{A}}_{\mathcal{S}}$ is consistent w.r.t. $\mathcal{T}_{\mathsf{red}}$. Suppose $\mathcal{S} = \{X_1, \dots, X_k\}$. For each $\iota \geq 0$, we know that there is an $i_\iota \in \{1, \dots, k\}$ such that $X_{i_\iota} = \{p_\gamma \in \mathcal{L} \mid \mathcal{I}_\iota \models \gamma\}$. Conversely, for each $i \in \{1, \dots, k\}$, there is an $\iota \geq 0$ such that $i = i_\iota$. Let $\iota_1, \dots, \iota_k \in \{0, 1, \dots\}$ be such that $i_{\iota_1} = 1, \dots, i_{\iota_k} = k$. The interpretation $\mathcal{J}$ is obtained from $\mathcal{I}_{\iota_i}$ by the following construction:[6]

- $\Delta^{\mathcal{J}} := \Delta^{\mathcal{I}_{\iota_1}}$ $(= \Delta^{\mathcal{I}_{\iota_2}} = \dots = \Delta^{\mathcal{I}_{\iota_k}})$,
- $a^{\mathcal{J}} := a^{\mathcal{I}_{\iota_1}}$ $(= a^{\mathcal{I}_{\iota_2}} = \dots = a^{\mathcal{I}_{\iota_k}})$ for all $a \in \mathsf{N_I}$,
- $N^{\mathcal{J}} := \{a^{\mathcal{J}} \mid a \in \mathsf{Obj}\}$,
- $(A^{(i)})^{\mathcal{J}} := A^{\mathcal{I}_{\iota_i}}$ for all concept names A in the input and $1 \leq i \leq k$,
- $(A^{(0)})^{\mathcal{J}} := A^{\mathcal{I}_0}$ for all concept names A in the input,
- $(r^{(i)})^{\mathcal{J}} := r^{\mathcal{I}_{\iota_i}}$ for all role names r in the input and $1 \leq i \leq k$,
- $(r^{(0)})^{\mathcal{J}} := r^{\mathcal{I}_0}$ for all role names r in the input, and
- $(T_C^{(i)})^{\mathcal{J}} := C^{\mathcal{I}_{\iota_i}}$ for all $C \in \mathsf{Sub}$ and $1 \leq i \leq k$.

[6]It is possible that $\mathcal{I}_{\iota_1}, \dots, \mathcal{I}_{\iota_k}$ do not respect the order in $\mathfrak{I}$, but this does not matter for our purpose.

It follows from the definition of $\mathcal{J}$ that $\mathcal{J} \models \mathcal{T}_N$. By induction on the structure of C, it can be shown that for all $C \in \mathsf{Sub}$ and for all i with $1 \leq i \leq k$, $\mathcal{J}$ satisfies the concept definition of $T_C^{(i)}$ (cf. the proof of Lemma 15 in [5] for details). Thus, we get $\mathcal{J} \models \mathcal{T}_{\mathsf{red}}$.

The definition of $\mathcal{J}$ implies that for all i with $1 \leq i \leq k$ and for all $p_\gamma \in \mathcal{L}$, $\mathcal{I}_{\iota_i} \models \gamma$ iff $\mathcal{J} \models \gamma^{(i)}$. The definition of X_i implies that $p_\gamma \in X_i$ iff $\mathcal{I}_{\iota_i} \models \gamma$. Thus, for all i with $1 \leq i \leq k$, $\mathcal{J}$ is a model of the following ABox $\mathcal{A}_i$

$$\{\gamma^{(i)} \mid p_\gamma \in X_i\} \cup \{\neg\gamma^{(i)} \mid p_\gamma \notin X_i\},$$

which implies that $\mathcal{J} \models \widehat{\mathcal{A}}_{\mathcal{S}}$.

"$\Leftarrow$": Suppose that there is a set $\mathcal{S} \subseteq \mathcal{P}(\mathcal{L})$ such that $L_\omega(\mathcal{D}(\widehat{\phi}, \mathcal{S}, \mathcal{B})) \neq \emptyset$ and $\widehat{\mathcal{A}}_{\mathcal{S}}$ is consistent w.r.t. $\mathcal{T}_{\mathsf{red}}$. Thus, there exists a model $\mathcal{J}$ of $\widehat{\mathcal{A}}_{\mathcal{S}}$ and $\mathcal{T}_{\mathsf{red}}$. For $i \in \{1, \ldots, k\}$, we define $\mathcal{J}_i$ as follows:

- $\Delta^{\mathcal{J}_i} := \Delta^{\mathcal{J}}$,
- $a^{\mathcal{J}_i} := a^{\mathcal{J}}$ for every individual name $a \in \mathsf{N_I}$,
- $A^{\mathcal{J}_i} := (T_A^{(i)})^{\mathcal{J}}$ for every concept name A in the input, and
- $r^{\mathcal{J}_i} := (r^{(i)})^{\mathcal{J}} \cap (N^{\mathcal{J}} \times N^{\mathcal{J}}) \cup (r^{(0)})^{\mathcal{J}} \cap (\Delta^{\mathcal{J}} \times (\neg N)^{\mathcal{J}} \cup (\neg N)^{\mathcal{J}} \times \Delta^{\mathcal{J}})$ for every role name r in the input.

By induction on the structure of C, we can show that for each $C \in \mathsf{Sub}$, $C^{\mathcal{J}_i} = (T_C^{(i)})^{\mathcal{J}}$ (cf. the proof of Lemma 15 in [5] for details). Since $\mathcal{J} \models \mathcal{T}_{\mathsf{red}}$, for all $A \equiv C \in \mathcal{T}$, $\mathcal{J} \models T_A^{(i)} \equiv T_C^{(i)}$ for all i with $1 \leq i \leq k$. Hence, $\mathcal{J}_i \models \mathcal{T}$ for all i with $1 \leq i \leq k$. Moreover, $\mathcal{J} \models \widehat{\mathcal{A}}_{\mathcal{S}}$ implies that for all i with $1 \leq i \leq k$, $\mathcal{J}$ is a model of the following ABox $\mathcal{A}_i$:

$$\{\gamma^{(i)} \mid p_\gamma \in X_i\} \cup \{\neg\gamma^{(i)} \mid p_\gamma \notin X_i\}.$$

Thus, for all $p_\gamma \in \mathcal{L}$, $p_\gamma \in X_i$ iff $\mathcal{J}_i \models \gamma$.

Since $L_\omega(\mathcal{D}(\widehat{\phi}, \mathcal{S}, \mathcal{B})) \neq \emptyset$, then there exists an accepting run $(q_0, \alpha_0, X_0)(q_1, \alpha_1, X_1)\ldots$ of $\mathcal{B}'$ on $X_0X_1\ldots$ such that $X_0X_1\ldots \in L_\omega(\mathcal{C}_{\widehat{\phi}}^{\mathcal{S}})$. By the construction of $\mathcal{B}'$, we get that $w = \alpha_0\alpha_1\cdots \in L_\omega(\mathcal{B})$. By Lemma 17, $X_0X_1\ldots \in L_\omega(\mathcal{C}_{\widehat{\phi}}^{\mathcal{S}})$ implies that the propositional LTL structure $\widehat{\mathcal{J}} = (X_i)_{i=0,1,\ldots}$ has the property that $\widehat{\mathcal{J}}, 0 \models \widehat{\phi}_{\mathcal{S}}$. Thus, $\widehat{\mathcal{J}} \models \widehat{\phi}$ and

$$\widehat{\mathfrak{J}}, 0 \models \Box \left(\bigvee_{X \in \mathcal{S}} \left(\bigwedge_{p \in X} p \wedge \bigwedge_{p \notin X} \neg p \right) \right).$$

The latter implies that for all $\iota \geq 0$,

$$\widehat{\mathfrak{I}}, \iota \models \bigvee_{X \in \mathcal{S}} \left(\bigwedge_{p \in X} p \wedge \bigwedge_{p \notin X} \neg p \right).$$

Hence, for all $\iota \geq 0$, there exists exactly one i_ι such that $1 \leq i_\iota \leq k$, $X_{i_\iota} \in \mathcal{S}$ and $X_\iota = X_{i_\iota}$. Consider the DL-LTL structure $\mathfrak{I} = (\mathcal{I}_\iota)_{\iota=0,1,\ldots}$ with $\mathcal{I}_\iota = \mathcal{J}_{i_\iota}$. Then, for all $\iota \geq 0$, for all $p_\gamma \in \mathcal{L}$, $p_\gamma \in X_\iota$ iff $\mathcal{I}_\iota \models \gamma$. This, together with $\widehat{\mathcal{J}} \models \widehat{\phi}$, yields that $\mathfrak{I}, 0 \models \phi$ (This can be shown by induction on the structure of ϕ).

Now we show that for all $\iota \geq 0$, $\mathcal{I}_\iota \Rightarrow^{\mathcal{T}}_{w(\iota)} \mathcal{I}_{\iota+1}$. By the definition of $\mathcal{I}_\iota$, we know that all of $\mathcal{I}_\iota$ share the domain and interpretation of individuals. Since $\mathcal{J}_i \models \mathcal{T}$ for all i with $1 \leq i \leq k$, by the definition of $\mathcal{I}_\iota$, we have $\mathcal{I}_\iota \models \mathcal{T}$ for all $\iota \geq 0$. It follows from the definitions of $\mathcal{J}_1, \ldots, \mathcal{J}_k$ and the fact $\mathcal{J} \models \mathcal{T}_{\mathsf{red}}$ that for all $x, y \in \Delta^{\mathcal{J}}$, we have for all i with $1 \leq i \leq k$,

- for each primitie concept name A in $\mathcal{T}$, if $x \notin N^{\mathcal{J}}$, then $x \in A^{\mathcal{J}_i}$ iff $x \in (A^{(0)})^{\mathcal{J}}$ and
- for each role name r, if $x \notin N^{\mathcal{J}}$ or $y \notin N^{\mathcal{J}}$, then $(x, y) \in r^{\mathcal{J}_i}$ iff $(x, y) \in (r^{(0)})^{\mathcal{J}}$.

This implies the anonymous objects respect the semantics of actions, which, together with the fact that for all $\iota \geq 0$, for all $p_\gamma \in \mathcal{L}$, $p_\gamma \in X_\iota$ iff $\mathcal{I}_\iota \models \gamma$, and the definition of Δ' (the transition relation of $\mathcal{B}'$) yields that the conditions in Definition 4 are satisfied. Similarly, for all $\iota \geq 0$, $\mathcal{I}_\iota \models \mathsf{pre}_\iota$ since for all $\gamma \in \mathsf{pre}_\iota$, $p_\gamma \in X_\iota$. Thus, α_i is executable in $\mathcal{I}_\iota$. Hence, $\mathfrak{I}$ is a DL-LTL structure generated by w from $\mathcal{A}$ ($\mathcal{A} = \emptyset$ by our assumption) w.r.t. $\mathcal{T}$. ❏

This lemma yields a decision procedure for the satisfiability problem. In fact, the double-exponentially many sets $\mathcal{S} \subseteq \mathcal{P}(\mathcal{L})$ can be enumerate within ExpSpace, and the exponentially large automaton $\mathcal{D}(\widehat{\phi}, \mathcal{S}, \mathcal{B})$ can be tested for emptiness in exponential time. Finally, the ABox $\widehat{\mathcal{A}}_{\mathcal{S}}$ is of exponential size (due to the fact that $\mathcal{S}$ is of exponential size) and the same is true for $\mathcal{T}_{\mathsf{red}}$. Since consistency w.r.t. an acyclic TBox is PSpace-complete in $\mathcal{ALCQO}$ (ExpTime-complete in $\mathcal{ALCIO}$, NExpTime-complete in $\mathcal{ALCQIO}$, respectively), the required consistency test can be performed in ExpSpace (2-ExpTime, 2-NExpTime, respectively).

Theorem 19. *The satisfiability problem (and the complement of the validity problem) of DL-LTL formulas w.r.t. acyclic TBoxes, ABoxes, and Büchi automaton over an alphabet of (possibly) conditional actions (possibly) with occusions is*

- *in* ExpSpace *for* $\mathcal{ALCQO}$ *if the numbers in qualified number restriction are coded in unary;*

- *in 2-*ExpTime *for* $\mathcal{ALCIO}$*;*
- *in 2-*NExpTime *for* $\mathcal{ALCQIO}$*.*

5 Future Work

To sum up, we have shown that the verification problem for non-terminating action logic programs becomes decidable if we abstract from the actual execution sequences of a non-terminating program by considering infinite sequences of actions defined by a Büchi automaton, and assume that the logic employed by the action theory is a decidable description logic.

In this paper, we have assumed that a Büchi automaton abstracting the program in the sense that all possible execution sequences of the program are accepted by this automaton is given (e.g., by the developer of the action program). An important topic for future research is how to generate such an abstraction automatically from a given program. Alternatively, if this is not possible since it yields abstractions that are too coarse (i.e., containing too many infinite sequences of actions that are not execution sequences of the program), it would still be helpful to develop tools that facilitate proving that a given Büchi automaton is an abstraction of a given program. In addition, it will probably be necessary to develop optimized versions of the decisions procedures introduced in this paper before they can be applied to large DL-based action theories.

Acknowledgements We would like to thank Carsten Lutz, Giuseppe de Giacomo, and Gerhard Lakemeyer for helpful discussions.

References

[1] C. Areces, P. Blackburn, and M. Marx. A Road-Map on Complexity for Hybrid Logics. In *CSL: 13th Workshop on Computer Science Logic*. LNCS, Springer-Verlag, 1999.

[2] F. Baader, D. Calvanese, D. L. Mcguinness, D. Nardi, and P. F. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[3] F. Baader, S. Ghilardi, and C. Lutz. LTL over description logic axioms. In G. Brewka and J. Lang, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Eleventh International Conference, KR 2008*, pages 684–694. AAAI Press, 2008.

[4] F. Baader, C. Lutz, M. Miličić, U. Sattler, and F. Wolter. Integrating description logics and action formalisms: First results. In M. Veloso and S. Kambhampati, editors, *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-05)*, Pittsburgh, Pennsylvania (USA), 2005. AAAI Press. A long version of this paper, containing all technical details, was published as LTCS-Report 05-02, Chair for Automata Theory, Institute for Theoretical Computer Science, Dresden University of Technology. See http://lat.inf.tu-dresden.de/research/reports.html.

[5] F. Baader, M. Milicic, C. Lutz, U. Sattler, and F. Wolter. Integrating Description Logics and Action Formalisms for Reasoning about Web Services. LTCS-Report LTCS-05-02, Chair for Automata Theory, Institute for Theoretical Computer Science, Dresden University of Technology, Germany, 2005. See http://lat.inf.tu-dresden.de/research/reports.html.

[6] J. Claßen and G. Lakemeyer. A logic for non-terminating Golog programs. In G. Brewka and J. Lang, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Eleventh International Conference, KR 2008*, pages 589–599. AAAI Press, 2008.

[7] G. D. Giacomo, E. Ternovskaia, and R. Reiter. Non-terminating processes in the situation calculus. In *Proceedings of the AAAI'97 Workshop on Robots, Softbots, Immobots: Theories of Action, Planning and Control*, 1997.

[8] H. J. Levesque, R. Reiter, Y. Lespérance, F. Lin, and R. B. Scherl. GOLOG: A logic programming language for dynamic domains. *J. Log. Program.*, 31(1–3):59–83, 1997.

[9] M. Miličić. *Action, Time and Space in Description Logics*. PhD thesis, Technische Universität Dresden, 2008.

[10] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[11] N. Pelov and E. Ternovska. Reducing inductive definitions to propositional satisfiability. In M. Gabbrielli and G. Gupta, editors, *Logic Programming, Proceedings of the 21st International Conference, ICLP 2005*, volume 3668 of *Lecture Notes in Computer Science*, pages 221–234. Springer, 2005.

[12] A. Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science, FOCS 1997*, pages 46–57. IEEE, 1977.

[13] I. Pratt-Hartmann. Complexity of the Two-Variable Fragment with Counting Quantifiers. *Journal of Logic, Language and Information*, 14(3):369–395, 2005.

[14] M. Thielscher. FLUX: A logic programming method for reasoning agents. *Theory and Practice of Logic Programming*, 5(4–5):533–565, 2005.

[15] W. Thomas. Automata on infinite objects. In *Handbook of Theoretical Computer Science*, volume B, pages 134–189. Elsevier, 1990.

[16] S. Tobies. The Complexity of Reasoning with Cardinality Restrictions and Nominals in Expressive Description Logics. *J. Artif. Intell. Res. (JAIR)*, 12:199–217, 2000.

[17] B. Trakhenbrot and Y. Barzdin. *Finite automata: Behavior and synthesis.* North–Holland, Amsterdam, 1973.

[18] P. Wolper, M. Y. Vardi, and A. P. Sistla. Reasoning about infinite computation paths. In *Proceedings of the 24th Annual Symposium on Foundations of Computer Science, FOCS 1983*, pages 185–194. IEEE, 1983.