TECHNISCHE UNIVERSITÄT DRESDEN

International Masters Programme in Computational Logic
Institute for Theoretical Computer Science
Computer Science

Master Thesis

Integrate Action Formalisms into Linear Temporal Description Logics

Anees ul Mehdi

Overseeing Professor
Prof. Dr. Franz Baader

Supervisor
Hongkai Liu

Dresden, Germany August, 2009

TECHNISCHE UNIVERSITÄT DRESDEN

Author:	**Anees ul Mehdi**
Matrikel-Nr:	**3280409**
Title:	**Integrate Action Formalisms into Linear Temporal Description Logics**
Degree:	**Master of Science**
Date of submision:	06.08.09

Declaration

Hereby I certify that the thesis has been written by me. Any help that I have received in my research work has been acknowledged. Additionally, I certify that I have not used any auxiliary sources and literature except those I cited in the thesis.

Anees ul Mehdi

Abstract

Description logics (DLs) provide expressiveness much beyond the expressiveness of propositional logic while still maintaining decidability of reasoning. This makes DLs a natural choice for formalizing actions. Besides DLs are also used in several application domains. However representing dynamic aspects of such application domains is not out of question. As a result, temporal extensions of DLs have been investigated in literature. In formalizing actions, sometimes we come across a situation, where we want to be sure of a property to hold at a certain time. Thus a suitable approach is of using temporalized DLs in describing such properties meanwhile formalizing actions in DLs. In this thesis, we present the integration of action formalisms in a temporalized DL.

We consider the *satisfiability problem* of an $\mathcal{ALCO}$-LTL formula with respect to an acyclic TBox, an ABox and actions i.e., we check if there is a sequence of world states (interpretations) such that the formula is satisfied in this sequence whereas the semantics of the actions is also respected. We consider two different cases; a simple case in which we consider *unconditional actions* where all the changes imposed by an action hold trivially after the application of the action and a general case in which we consider *conditional actions*. A conditional action requires certain conditions to hold in order to impose such changes. In the former case, we reduce the problem to the ABox consistency problem, whereas in the later case, we reduced it to the emptiness problem of a Büchi automaton and the ABox consistency problem.

Contents

Chapter 1

Introduction

This chapter provides an introduction to the topics that we deal with in this work. In Section 1.1, we present a brief introduction to description logics (DLs). In the second section of the chapter, we give an introductory overview of temporal description logics and briefly discuss the need for such extensions of DLs. Later in Section 1.3, we introduce action formalisms based on DLs. Finally, we set the objectives of our work in Section 1.4 and outline the structure of the work in Section 1.5.

1.1 Description Logics

The main focus in knowledge representation is usually not only on providing a high-level description of the domain of discourse but also some techniques (reasoning services) to extract implicit consequences from the explicit knowledge represented. Among different knowledge representation formalisms, logic based formalisms are the most popular one. In such formalisms some logic is used to provide a formal syntax as well as a formal semantics. Description logics are well-known logic based formalisms with the characteristic that they allow us to represent and reason about conceptual knowledge in a structured and well-understood way.

In 1970's different approaches to knowledge representation were developed. The logical based formalisms were usually based on predicate logic, and hence resulting in different decidability/computability issues (Franz Baader *et al.*, 2003). In non-logical based formalisms, because of the lack of semantics, the interpretations of knowledge represented were quite subjective. The *frame system* (Minsky, 1974) and *semantic networks* (Quillian, 1968) were the most famous non-logical based formalism for knowledge representation.

Unlike logic based formalisms, these formalisms were often based on the use of graphical interfaces by representing knowledge by means of some data structures. Different procedures were used to manipulate these structures for accomplishing the reasoning task. The main problem in such formalisms was the lack of semantics thus causing ambiguities.

On the one hand, we have predicate logic which is very expressive but mean while undecidable. On the other hand, propositional logic allows decidability, nevertheless expressiveness is compromised. This tradeoff between the expressiveness of a representation formalism and the difficulty of reasoning over the representations built using this formalism, was first observed by Brachman and Leavesque in (Brachman & Levesque, 1984), where they also provided the language $\mathcal{FL}^-$ (Frame Language) as an example. Soon the focus of research turned towards finding a formalism which would be as a compromise between expressibility of the formalism and complexity of the reasoning services provided by it. This led to the birth of description logics. $\mathcal{ALC}$ is one of the most basic DLs that was introduced by Schaußand Smolka (Schmidt-Schauß& Smolka, 1991). It has been extended to many other more expressive DLs by allowing additional constructors. However, this addition of constructors possibly increases the complexity of reasoning services. One can say that DLs are a family of knowledge representation formalisms which have the expressive power between propositional logic and predicate logic.

The basic notions in description logics are of *concept names* (unary predicates), *role names* (binary predicates) and individual names (constants). Different constructors are used in different DLs to build *concept descriptions* to model a domain. DLs are characterized by the set of constructors they provide (e.g., $\mathcal{ALC}$ is a DL with conjunction, disjunction, negation, existential restriction and value restriction). For example the following concept description expresses "a female student who has studied only medical courses":

$$\mathsf{Student} \sqcap \mathsf{Female} \sqcap \forall \mathsf{has\text{-}studied}.\mathsf{MedicalCourses}$$

In modeling an application domain in DLs, concept descriptions are used to built statements in a DL knowledge base (KB). Such DL KBs can be divided into two parts: a terminological one and an assertional one. In the first part, relevant notions of the domain can be described by stating properties of concepts and roles, and relationships between them. This part is called the TBox. A TBox introduces abbreviations (concept names) for a complex concept description. For example the concept name Father can be used as an abbreviation for the following concept description:

$$\mathsf{Father} \equiv \mathsf{Male} \sqcap \mathsf{Parent}$$

Such a statement is called a *concept definition.* A TBox contains finitely many concept definitions. The assertional part of the knowledge base is used to to describe the factual knowledge of the domain by stating properties of individuals. For example to describe that "John is a Student", "John is a son of Joseph" and "Marry is not a daughter of John", one can use the assertions $\mathsf{Student(JOHN)}$, $\mathsf{sonOf(JOHN, JOSEPH)}$ and $\neg\mathsf{daughterOf(Marry,JOHN)}$ respectively. The first assertion is a concept assertion whereas the second and third assertions are role assertions. Here "JOHN", "MARRY" and "JOSEPH" are individual names that are used to represent the individuals John, Marry and Joseph respectively. A finite set such assertions is called an *ABox.*

The semantics for a DL is given in terms of interpretations. An *interpretation* is composed of a *domain* which is a non-empty set of individuals and an *interpretation function* that assigns a subset of the domain to each concept name, a binary relation on the domain to each role name and an element of the domain to each individual name. The interpretation is extended to concept descriptions by interpreting each constructor according to the semantics provided by the DL, e.g., conjunction of two concepts is interpreted as intersection of their interpretations. *Satisfiability* of a concept description requires that it is interpreted by a non-empty subset of the domain. In such a case, the interpretation is said to be a *model* of the concept description. We say that an interpretation is a *model* of a TBox $\mathcal{T}$ if and only if the left-hand side and the right-hand side of every concept definition in $\mathcal{T}$ are interpreted identically. An assertion $C(a)$ for a concept name C and individual a is *satisfied* by an interpretation if a is interpreted by an element of the C's interpretation. Similarly for an assertion $r(a, b)$ to be satisfied in an interpretation, the interpretation of a and b must be in the relation defined by the interpretation of r. And in case of $\neg r(a, b)$ the interpretation of a and b must not be in the relation defined by the interpretation of r. We call an interpretation a *model* of an ABox $\mathcal{A}$ if it satisfies all the assertion in $\mathcal{A}$. An ABox $\mathcal{A}$ is said to be *consistent w.r.t.* a TBox $\mathcal{T}$ if there is a model of the $\mathcal{T}$ that is also a model of $\mathcal{A}$.

In most of the DLs, the basic constructors are conjunction ($\sqcap$), disjunction ($\sqcup$), negation ($\neg$), existential restriction ($\exists$) and value restriction ($\forall$). The DL $\mathcal{ALC}$ allows only these constructors. In some DLs, an additional constructor called *nominal* is allowed, which can be used to describe a concept containing only one individual e.g., $\{\mathsf{JOHN}\}$ is a concept description.

Such a concept description is interpreted by the singleton set. The presence of this additional constructor is indicated by appending the letter $\mathcal{O}$ with the name of the DL. For example, the DL which extends $\mathcal{ALC}$ with nominals is named as $\mathcal{ALCO}$. We will use $\mathcal{ALCO}$ in this work. Later on we will explain the reason for choosing $\mathcal{ALCO}$. In the following section, we describe the extensions of DLs with temporal operators and the need for such extensions.

1.2 Temporal Extensions of DLs

In many applications it is important to describe temporal patterns. For example, using DLs to represent conceptual models of temporal databases (Artale *et al.*, 2002). As another example, suppose that we want to represent the fact that "If John is a student and has good grades, he will be a scholarship holder sometime in the future". In order to describe such a temporal knowledge conveniently, temporal extensions of DLs have been investigated in literature. Schild was first to consider the combination of DL $\mathcal{ALC}$ with linear temporal logic (LTL) (Schild, 1993), which since then have experienced constant development in the sense that the DL and the temporal component have varied. An important issue in temporalizing DLs is deciding to which pieces of syntax temporal operator can be applied. For example, temporal operator can occur within a concept descriptions or in front of an assertion. We use the description logics $\mathcal{ALCO}$ in the DL component and the linear temporal logic (LTL) (Krger & Merz, 2008) in the temporal component, which we call $\mathcal{ALCO}$-LTL. We allow the application of a temporal operator to ABox assertions only (Baader *et al.*, 2008b). For example the above concept can be described as follows:

$$\big((\mathsf{Student} \sqcap \exists \mathsf{has.GoodGrades})(\mathsf{JOHN}) \rightarrow \Diamond(hold.Scholarship(\mathsf{JOHN}))\big)$$

Where $\Diamond$ is the temporal modality which is usually read as "eventually" (Baier & Katoen, 2008).

Note that in temporalized DLs, we have two parts, a temporal part and a DL part. The semantics of a temporalized DL needs to consider both of the parts. Therefore interpretations of concept names and role names do not depend only on a DL interpretation but also on a time point. Hence semantics for a temporalized DL is given in terms of an infinite sequence of DL interpretations. Such a sequence is usually called a *structure*. In case of $\mathcal{ALCO}$-LTL we call such a sequence an *$\mathcal{ALCO}$-LTL structure*. For some concept and role names it is not desirable that their interpretation changes over time (Baader *et al.*, 2008b). Such concept names or role names are called *rigid concept names* and *rigid role names* respectively.

1.3 Description Logic and Action Formalisms

In literature there are several action theories to model dynamic application domains e.g., Situation Calculus (Reiter, 2001), Fluent Calculus (Thielscher, 2005). Nevertheless these formalisms are usually formulated in predicate or higher-order logic and hence do not permit decidable reasoning (Milicic, 2008). As a solution, one can go for propositional logic but has to compromise on expressiveness. Description Logics (DLs) are a well-known family of knowledge representation formalisms and can be viewed as fragments of predicate logic. On the one hand DLs are considerably more expressive than propositional logic, on the other hand, they are still decidable unlike predicate logic (Franz Baader *et al.*, 2003). DLs are also the basis of the WSC-recommended Web ontology language OWL and hence the availability of actions in Semantic Web raised the question of formalizing such actions in DLs.

In (Baader *et al.*, 2005a), an approach of formalizing actions in DLs has been presented. In this approach, ABox assertions are used to describe world states and actions, while concept definitions are used as domain constraints which have to be satisfied no matter how the actions change the world. For example the action "If John is an applicant for a scholarship and has good grades then allot him a scholarship" can be expressed by

$$(\mathsf{Applicant} \sqcap \exists \mathsf{has}.\mathsf{GoodGrades})(\mathsf{JOHN})/\exists \mathsf{alloted}.\mathsf{Scholarship}(\mathsf{JOHN})$$

An interpretation corresponds to a state of world which is changed by actions. Therefore if $(\mathsf{Applicant} \sqcap \exists \mathsf{has}.\mathsf{GoodGrades})(\mathsf{JOHN})$ holds in an interpretation then $\exists \mathsf{alloted}.\mathsf{Scholarship}(\mathsf{JOHN})$ holds in the next interpretation when the above action is applied.

Note that in temporalized DLs the semantics depends on an infinite sequence of interpretations. In action theories, we have sequence of states of the world. In DL based action formalisms interpretations correspond to states of the world, hence we have sequence of interpretations. The difference between such a sequence in temporalized DL and a DL based action formalism is that in temporalized DL, except for the rigid concept and role names, the interpretations of the other concept names and role names can vary arbitrarily at different time point, whereas in a DL based action formalism, the only changes are due to the application of an action, i.e., the interpretations of concept and role names stay the same unless an action changes them.

One of the basic reasoning problems in action formalisms is the projection problem. The *projection problem* is to check if a given assertion always holds in every state reached from the initial state through the application of a given sequence of actions. In (Baader *et al.*, 2005a), it shown that the use of nominals for solving the projection problem is unavoidable. It is also shown that this problem is PSpace-Complete. We will see that the reasoning problem considered in this work is at least as hard as the projection problem. Hence, we choose $\mathcal{ALCO}$. We could have started with $\mathcal{ALC}$, but the complexity results we get with $\mathcal{ALC}$ and with $\mathcal{ALCO}$ coincide.

1.4 Objective

The beauty of description logics is that they are more expressive than propositional logic as well as they provide decidable reasoning services. In some application domains one needs to describe temporal patterns. To model such application domains conveniently, DLs are extended with temporal component (Artale *et al.*, 2007; Artale & Franconi, 2000; Wolter & Zakharyaschev, 2000; Artale *et al.*, 2002; Baader *et al.*, 2008a). In action theory community, DLs are good choice to model actions: reason being again their expressiveness and the availability of decidable reasoning services. In many applications we want to be sure about some properties to hold at a certain time before some actions to be taken. As an example, consider a robot which can perform post delivery to each office in an organization. Lets call this robot "Postboy'". The set of actions of Postboy can be formulated in a DL. Suppose we want to make sure that it is never the case that the robot is out of power before it reaches to its recharge panel. We can describe this condition as following:

$$\Box((\neg\mathsf{outOfBattery} \sqcup \mathsf{hasAccess.RechargePanel})(\mathsf{POSTBOY}))$$

We can model such problems by using temporalized DL to describe the properties and also use DLs to describe actions. In this thesis, we consider $\mathcal{ALCO}$ for formalizing actions and $\mathcal{ALCO}$-LTL for specifying such properties. Our objective is to provide a method to check whether there exists a sequence of world states (interpretations) such that the property is satisfied in a certain state and the semantics of the actions is respected, i.e., a state changes to another state only via the application of actions. We call this *the satisfiability problem*[1]. We consider two cases. In the first case, we re-

[1]In this work, by satisfiability problem we always mean this problem until specified.

strict ourselves to those actions (*unconditional*) where the effects of actions trivially holds when the actions are applied. In the second case, we consider actions in general sense where certain properties needs to be satisfied for the effect of the actions to hold after the application. We call such actions *conditional*. In each case, we check some properties described in $\mathcal{ACLO}$-LTL given some actions and a knowledge base formulated in $\mathcal{ALCO}$. We also analyze the computational complexity of solving the satisfiability problem.

1.5 Structure

In Chapter 2, We introduce notions and theoretical background that we will depend on throughout the work. In Section 2.1, we present an introduction to the description logic $\mathcal{ALCO}$. We will also discuss some reasoning problems in $\mathcal{ALCO}$. In Section 2.2, we introduce the temporalized description logic $\mathcal{ALCO}$-LTL by discussing its syntax and semantics. Action formalisms based on DLs are discussed in Section 2.3. Some inference problems like the satisfiability problem and the validity problem are introduced in Section 2.4. Finally in Section 2.5, we present a brief introduction to linear temporal logic(LTL) and Büchi Automata. We also discuss the reduction of the satisfiability problem in LTL to the emptiness problem in a Büchi automaton.

In Chapter 3, we consider conditional actions and the the satisfiability problem introduced in Chapter 2. We will also discuss complexity issues of this very particular case by reducing the problem into ABox consistency w.r.t. to an acyclic TBox.

In Chapter 4, we consider the problem in general by allowing conditional action, we reduce the satisfiability problem to emptiness problem of a Büchi automaton and ABox consistency w.r.t. an acyclic TBox.

Finally in Chapter 5 of the work, we will present some concluding remarks and possible extensions of the work.

Chapter 2

Preliminaries

In this chapter we provide the theoretical background needed for the thesis.

2.1 Introducing Description Logic $\mathcal{ALCO}$

In this section we will present the description logic $\mathcal{ALCO}$ by providing its syntax and semantics. We will also present some reasoning problems in $\mathcal{ALCO}$. We will mainly follow the notations as presented in (Baader *et al.*, 2005b).

2.1.1 Syntax and Semantics

In following, we present the syntax and semantics for the description logic $\mathcal{ALCO}$.

Definition 2.1. let N_C, N_R and N_I be disjoint and countably infinite sets of *concept names*, *role names* and *individual names* respectively. The set of $\mathcal{ALCO}$ *concept descriptions* (or *concepts* in short) is the smallest set satisfying the following properties:

- every concept name $A \in N_C$ is a concept description;
- $\top$ (top concept) and $\bot$ (bottom concept) are concept descriptions;
- if C and D are $\mathcal{ALCO}$ concept descriptions, r is a role name and a is an individual name then the following are $\mathcal{ALCO}$ concept descriptions:

$\neg C$	(negation)
$C \sqcap D$	(conjunction)
$C \sqcup D$	(disjunction)
$\exists r.C$	(existential restriction)
$\forall r.C$	(value restriction)
$\{a\}$	(nominal)

Every concept description other than a concept name is called a *complex concept*.

As an example, the concept $\mathsf{Male} \sqcap \exists \mathsf{hasChild}.\{\mathsf{John}\}$ describes the concept "father of John". We abbreviate the concept description $\neg C \sqcup D$ by $C \rightarrow D$ for concept descriptions C and D.

For the semantics of $\mathcal{ALCO}$ we introduce the notion of interpretations.

Definition 2.2. An $\mathcal{ALCO}$ *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 function that assigns

- to each concept name A, a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$;
- to each role name r, a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$;
- to each individual name a, an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ such that for all $a, b \in N_I$, if $a \neq b$ then $a^{\mathcal{I}} \neq b^{\mathcal{I}}$.

The interpretation of complex concepts is then defined as follows:

$$
\begin{aligned}
(\{a\})^{\mathcal{I}} &= \{a^{\mathcal{I}}\} \\
(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\exists r.C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} \text{with } (x, y) \in r^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \\
(\forall r.C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \forall y \in \Delta^{\mathcal{I}} \text{ if } (x, y) \in r^{\mathcal{I}}, \text{ then } y \in C^{\mathcal{I}}\}
\end{aligned}
$$

$\mathcal{I}$ is called a *model* of a concept C if $C^{\mathcal{I}} \neq \emptyset$.

The restriction to interprate different individual names by different elements of the domain enforces the interpretation $\mathcal{I}$ to satisfy the *unique name assumption* (UNA). The UNA is commonly adopted by action formalisms community.

As mentioned in Chapter 1, a DL knowledge base is composed of two parts: a terminological part (called the TBox) and an assertional part(called the ABox). In the following, we talk about the terminological part.

Definition 2.3. A *concept definition* is of the form $A \equiv C$ where A is a concept name and C a concept. A *TBox* is a finite set of concept definitions such that there is no concept name which occurs in the left-hand side of two different concept definitions. We say that a concept name A *directly uses* a concept name B w.r.t. to a TBox $\mathcal{T}$ if there is a concept definition $A \equiv C \in \mathcal{T}$ with B occurring in C. Let *uses* be the transitive closure of directly uses. Then a TBox $\mathcal{T}$ is *acyclic* if no concept name uses itself w.r.t. $\mathcal{T}$.

An interpretation $\mathcal{I}$ *satisfies* a concept definition $A \equiv C$ (written as $\mathcal{I} \models A \equiv C$) if $A^{\mathcal{I}} = C^{\mathcal{I}}$. $\mathcal{I}$ is called a *model* of a TBox $\mathcal{T}$, written $\mathcal{I} \models \mathcal{T}$, if it satisfies all concept definitions in $\mathcal{T}$.

A concept name A is *defined w.r.t* to a TBox $\mathcal{T}$ if A occurs on the left-hand side of a concept definition in $\mathcal{T}$, and *primitive* w.r.t. $\mathcal{T}$ otherwise.

To describe a concrete situation by stating properties of individuals we use the assertional part of the knowledge base which is defined as follows:

Definition 2.4. An *assertion (or ABox assertion)* is of the form $C(a)$, $r(a, b)$ or $\neg r(a, b)$, where $a, b \in N_I$, C is a concept, and r is a role name. An *ABox* is a finite set of assertions. An interpretation $\mathcal{I}$ *satisfies* an assertion

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

If φ is an assertion, then we write $\mathcal{I} \models \varphi$ iff $\mathcal{I}$ satisfies φ. An interpretation $\mathcal{I}$ is called a *model* of an ABox $\mathcal{A}$, written $\mathcal{I} \models \mathcal{A}$, if $\mathcal{I}$ satisfies all assertions in $\mathcal{A}$.

Definition 2.5. A knowledge base (KB) is a pair $(\mathcal{T}, \mathcal{A})$, where $\mathcal{T}$ is a TBox and $\mathcal{A}$ is an ABox. An interpretation $\mathcal{I}$ is a model of a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ (written as $\mathcal{I} \models \mathcal{K}$) if $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$.

2.1.2 DL Reasoning Problems

Usually inference problems in DLs are defined with respect to a KB consisting of a TBox and an ABox. But in some special cases we might consider such problems with the TBox or/and ABox being empty.

Definition 2.6. Given a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ with $\mathcal{T}$ a TBox and $\mathcal{A}$ an ABox, $\mathcal{K}$ is *consistent* iff it has a model, i.e., there is an interpretation $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$. An assertion φ is *satisfiable* w.r.t. $\mathcal{T}$ if there is a model $\mathcal{I}$ of $\mathcal{T}$ such that $\mathcal{I} \models \varphi$. The ABox $\mathcal{A}$ is *consistent* w.r.t. the TBox $\mathcal{T}$ iff there is a model $\mathcal{I}$ of $\mathcal{T}$ such that $\mathcal{I} \models \mathcal{A}$. *Consistency problem (or consistency in short)* of an ABox $\mathcal{A}$ w.r.t. a TBox $\mathcal{T}$ is to check whether $\mathcal{A}$ is consistent w.r.t. $\mathcal{T}$.

2.2 Temporalized Description Logic $\mathcal{ALCO}$-LTL

In practice, we come across situations where we need to represent dynamic aspects of the application domain. To model such dynamic applications in a convenient way we need a combination of the standard DLs with standard temporal logic, where one dimension is for time and the other for the DL domain (Wolter & Zakharyaschev, 1998). Here we present such a temporalized DL called *$\mathcal{ALCO}$-LTL* with $\mathcal{ALCO}$ in the DL component and linear temporal logic (LTL) in the temporal component (Gabbay *et al.*, 2003). We will follow notions as presented in (Baader *et al.*, 2008b).

Definition 2.7. $\mathcal{ALCO}$-LTL formulas are defined by induction:

- if α is an assertion then α is an $\mathcal{ALCO}$-LTL formula;
- if ϕ, ψ are $\mathcal{ALCO}$-LTL formulas, then so are $\phi \wedge \psi$, $\phi \vee \psi$, $\phi \mathsf{U} \psi$, and $\mathsf{X}\psi$.

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

In linear temporal logic we define a temporal (or Kripke) structure for the semantics (Krger & Merz, 2008). Such a structure is infinite sequence $x_0, x_1, \ldots$ with $x_i \in 2^{\mathsf{PL}}$ with $i \geq 1$ and PL being the set of propositional variables. By 2^{PL} we mean the set of all the subsets of PL. But in case of $\mathcal{ALCO}$-LTL formulas, temporal components are attached with ABox assertions. The usual temporal structures, therefore, can not be used to interpret such assertions at a certain time point. Hence we have to use the notion of structures which are infinite sequence of $\mathcal{ALCO}$ interpretations.

Definition 2.8. An $\mathcal{ALCO}$-*LTL* structure is a *sequence* $\mathfrak{I} = (\mathcal{I}_i)_{i=0,1,\ldots}$ of $\mathcal{ALCO}$ interpretations $\mathcal{I}_i = (\Delta, \cdot^{\mathcal{I}_i})$ such that $a_i^{\mathcal{I}} = a_j^{\mathcal{I}}$ for all individual names a and $i, j \in \{0, 1, 2, \ldots\}$.

Given an $\mathcal{ALCO}$-LTL formula ϕ, an $\mathcal{ALC}$-LTL structure $\mathfrak{I} = (\mathcal{I}_i)_{i=0,1,\ldots,}$ and a time point $i \in \{0, 1, 2, \ldots\}$, satisfaction of ϕ in $\mathfrak{I}$ at time i (written $\mathfrak{I}, i \models \phi$) is defined inductively:

- $\mathfrak{I}, i \models \phi$ iff $\mathcal{I}_i \models \phi$ for an ABox assertion ϕ
- $\mathfrak{I}, i \models \phi \wedge \psi$ iff $\mathfrak{I}, i \models \phi$ and $\mathfrak{I}, i \models \psi$
- $\mathfrak{I}, i \models \phi \vee \psi$ iff $\mathfrak{I}, i \models \phi$ or $\mathfrak{I}, i \models \psi$
- $\mathfrak{I}, i \models \neg\phi$ iff $\mathfrak{I}, i \not\models \phi$
- $\mathfrak{I}, i \models \mathsf{X}\phi$ iff $\mathfrak{I}, i+1 \models \phi$
- $\mathfrak{I}, i \models \phi \mathsf{U} \psi$ iff there is $k \geq i$ such that $\mathfrak{I}, k \models \psi$ and $\mathfrak{I}, j \models \phi$ for all j, $i \leq j < k$

2.3 DL-based Action Formalisms

In (Baader *et al.*, 2005b; Baader *et al.*, 2005a), a frame work for formalizing actions in DLs is proposed. We use their notions of actions.

Definition 2.9. Let $\mathcal{T}$ be an acyclic TBox. An *action* is a finite set of *post-conditions* of the form φ/ψ, where φ is an ABox assertion and ψ is a *primitive literal for* $\mathcal{T}$, i.e., an ABox assertion of the form $A(a)$, $\neg A(a)$, $r(a, b)$, or $\neg r(a, b)$ with A a primitive concept name in $\mathcal{T}$, r a role name, and a and b individual names.

We will further classify the post-conditions into *unconditional* and *conditional*. By unconditional post-conditions we mean that all the post-conditions are of the form $\top(a)/\psi$ for an arbitrary individual name a, i.e., the condition $\top(a)$ trivially holds in all interpretations. All the other post-conditions will be referred as conditional post-conditions. We call an action α *unconditional* if it contains only unconditional post-conditions other we call it *conditional*. In unconditional actions, we simply write ψ instead of $\top(a)/\psi$.

Given a set of actions $\mathfrak{A}$, an infinite sequence of actions from $\mathfrak{A}$ is a function $w : \mathbb{N} \rightarrow \mathfrak{A}$ such that $w(i)$ is the i-th action in the sequence. We will be using infinite sequence of actions of the form $w = \alpha_1 \dots \alpha_p(\beta_1 \dots \beta_q)^\omega$ by which we mean the following:

$$w(i) := \begin{cases} \alpha_{i+1} & 0 \leq i < p, \\ \beta_{((i-p) \text{ mod } q)+1} & i \geq p. \end{cases}$$

where mod is the modulus function.

Intuitively by the post-condition φ/ψ we mean that, if φ is true before executing the action, then ψ should hold afterwards.

Definition 2.10. Let $\mathcal{T}$ be an acyclic TBox, an action α for $\mathcal{T}$, and $\mathcal{I}$, $\mathcal{I}'$ models of $\mathcal{T}$ sharing the same domain and agreeing on the interpretation of all individual names. We say that $\mathcal{I}'$ is *the result of updating* $\mathcal{I}$ with α, written $\mathcal{I} \Rightarrow^{\mathcal{T}}_{\alpha} \mathcal{I}'$, if for each primitive concept name A and role name r, we have

$$\begin{aligned} A^{\mathcal{I}'} &= (A^{\mathcal{I}} \cup \{a^{\mathcal{I}} \mid \varphi/A(a) \in \alpha \text{ and } \mathcal{I} \models \varphi\}) \setminus \{a^{\mathcal{I}} \mid \varphi/\neg A(a) \in \alpha \text{ and } \mathcal{I} \models \varphi\} \\ r^{\mathcal{I}'} &= (r^{\mathcal{I}} \cup \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid \varphi/r(a,b) \in \alpha \text{ and } \mathcal{I} \models \varphi\}) \\ &\qquad \setminus \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid \varphi/\neg r(a,b) \in \alpha \text{ and } \mathcal{I} \models \varphi\} \end{aligned}$$

For a sequence of actions $\alpha_1, \dots, \alpha_k$, $\mathcal{I}'$ is *the result of updating* $\mathcal{I}$ by this sequence($\mathcal{I} \Rightarrow^{\mathcal{T}}_{\alpha_1,\dots,\alpha_k} \mathcal{I}'$) if there are models $\mathcal{I}_0, \dots, \mathcal{I}_k$ of $\mathcal{T}$ with $\mathcal{I} = \mathcal{I}_0$, $\mathcal{I}' = \mathcal{I}_k$, and $\mathcal{I}_{i-1} \Rightarrow^{\mathcal{T}}_{\alpha_i} \mathcal{I}_i$ for $1 \leq i \leq k$.

Note that only those facts change by the application of an action that are forced to change by post-conditions of the action, i.e., if $\mathcal{I}$ and $\mathcal{I}'$ are interpretations, and α is an action for an acyclic TBox $\mathcal{T}$ with $\psi/\phi \in \alpha$ such that $\mathcal{I} \models \psi$ and $\mathcal{I} \Rightarrow^{\mathcal{T}}_{\alpha} \mathcal{I}'$, then we have that $\mathcal{I}' \models \phi$. Besides these changes, the semantics of action also requires that nothing else changes in the interpretation of any primitive concept name or role name. This we will refer to as the *minimization of change*. Further note that for acyclic TBoxes, the actions are deterministic, i.e., for $\mathcal{T}$ and α if $\mathcal{I} \Rightarrow^{\mathcal{T}}_{\alpha} \mathcal{I}'$ and $\mathcal{I} \Rightarrow^{\mathcal{T}}_{\alpha} \mathcal{I}''$ then $\mathcal{I}' = \mathcal{I}''$. This follows from Definition 2.10 and the fact that acyclic TBoxes are *definitorial*: the interpretation of defined concepts is uniquely determined by the interpretation of the primitive concepts and role names (Baader & Lutz, n.d.).

We assume that all the given actions are consistent in the sense that for a given action α with post-conditions φ_1/ψ,$\varphi_2/\neg\psi \in \alpha$, it is never the case that both φ_1 and φ_2 are satisfied in an interpretation $\mathcal{I}$, i.e., the ABox $\{\varphi_1, \varphi_2\}$ is inconsistent.

Note that given an acyclic TBox $\mathcal{T}$, an unconditional action α and two models $\mathcal{I}$ and $\mathcal{I}'$ of $\mathcal{T}$ with $\mathcal{I} \Rightarrow_{\alpha}^{\mathcal{T}} \mathcal{I}'$, all the post-condition in α always hold in $\mathcal{I}'$ independent of $\mathcal{I}$. Based on this observation we have the following property:

Lemma 2.11. *Let $\mathcal{T}$ be an acyclic TBox, $\alpha_1, \alpha_2, ..., \alpha_n$ be a finite sequence of unconditional action for $\mathcal{T}$. For any interpretation $\mathcal{I}$, $\mathcal{I}'$ and $\mathcal{I}''$ we have that: if $\mathcal{I} \Rightarrow_{\alpha_1,\alpha_2,....,\alpha_n}^{\mathcal{T}} \mathcal{I}' \Rightarrow_{\alpha_1,\alpha_2,....,\alpha_n}^{\mathcal{T}} \mathcal{I}''$ then $\mathcal{I}' = \mathcal{I}''$.*

Proof: By Definition 2.10 there are interpretations $\mathcal{I} = \mathcal{I}_0, \mathcal{I}_1, ..., \mathcal{I}_n = \mathcal{I}', \ldots, \mathcal{I}_{2n} = \mathcal{I}''$ such that $\mathcal{I}_{i-1} \Rightarrow_{\alpha_i}^{\mathcal{T}} \mathcal{I}_i$ and $\mathcal{I}_{n+i-1} \Rightarrow_{\alpha_i}^{\mathcal{T}} \mathcal{I}_{n+i}$ for $1 \leq i \leq n$. Further $\Delta^{\mathcal{I}} = \Delta^{\mathcal{I}_0} = ... = \Delta^{\mathcal{I}_n} = ... = \Delta^{\mathcal{I}_{2n}}$ and each individual is interpreted by the same domain element under any of these interpretations. Since $\mathcal{T}$ is acyclic, it suffices to show for each primitive concept B and role name r, $A^{\mathcal{I}'} = A^{\mathcal{I}''}$ and $r^{\mathcal{I}'} = r^{\mathcal{I}''}$. We prove by contradiction. Assume there is a primitive concept A such that $A^{\mathcal{I}'} \neq A^{\mathcal{I}''}$ or a role name r such that $r^{\mathcal{I}'} \neq r^{\mathcal{I}''}$. We consider the case of the concept A only, the case of role name r can be treated analogously.

Since $A^{\mathcal{I}'} \neq A^{\mathcal{I}''}$, at least one of the following must hold;

1. $A^{\mathcal{I}_n} \not\subseteq A^{\mathcal{I}_{2n}}$
   which implies that there is an $x \in \Delta^{\mathcal{I}}$ (here each of the interpretation shares the common domain) such that $x \in A^{\mathcal{I}_n} \setminus A^{\mathcal{I}_{2n}}$. But since $\mathcal{I}_n \Rightarrow_{\alpha_1}^{\mathcal{T}} \mathcal{I}_{n+1} \Rightarrow_{\alpha_2}^{\mathcal{T}}, \ldots, \Rightarrow_{\alpha_n}^{\mathcal{T}} \mathcal{I}_{2n}$, there is $i \in \{1, \ldots, n\}$ such that

   - $\neg A(a) \in \alpha_i$
   - $a^{\mathcal{I}} = x$ (again each individual name is interpreted by the same domain element under each interpretation).
   - there is no $j \in \{1, \ldots, n\}$ such that $j > i$ and $A(a) \in \alpha_j$.

   But $\mathcal{I} = \mathcal{I}_0 \Rightarrow_{\alpha_1}^{\mathcal{T}} \mathcal{I}_1 \Rightarrow_{\alpha_2}^{\mathcal{T}}, \ldots, \Rightarrow_{\alpha_n}^{\mathcal{T}} \mathcal{I}_n$ implies that $x \notin A^{\mathcal{I}_i}$ as $\neg A(a) \in \alpha_i$, and since there is no $j \in \{1, \ldots, n\}$ with $j > i$ and $A(a) \in \alpha_j$, hence $x \notin A^{\mathcal{I}_n}$ which is a contradiction.

2. $A^{\mathcal{I}_{2n}} \not\subseteq A^{\mathcal{I}_n}$
   we can reach a contradiction in a similar way as in case 1.

Hence $\mathcal{I}_n = \mathcal{I}_{2n}$, i.e. $\mathcal{I}' = \mathcal{I}''$. □

One of the most important reasoning problem in actions formalisms is the projection problem which is defined as follows:

Definition 2.12. For a given acyclic TBox $\mathcal{T}$, $\alpha_1, \ldots, \alpha_q$ actions for $\mathcal{T}$ and an ABox A, an assertion ϑ is a *consequence of applying* $\alpha_1, \ldots, \alpha_q$ in $\mathcal{A}$ w.r.t. $\mathcal{T}$ iff, for all models $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$, and all $\mathcal{I}'$ with $\mathcal{T} \Rightarrow^{\mathcal{T}}_{\alpha_1,\ldots,\alpha_q} \mathcal{I}'$, we have $\mathcal{I}' \models \vartheta$. The *projection problem* is to check if ϑ is a *consequence of applying* $\alpha_1, \ldots, \alpha_q$ in $\mathcal{A}$ w.r.t. $\mathcal{T}$.

It is shown (Baader *et al.*, 2005b) that the projection problem is PSpace-complete for $\mathcal{ALCO}$.

2.4 Inference Problem

In this section we introduce the satisfiability and validity problems of an $\mathcal{ALCO}$-LTL formula with respect to a given acyclic TBox, an ABox and an infinite sequence of actions.

Definition 2.13. Let φ be an $\mathcal{ALCO}$-LTL formula. Further suppose that we are given a TBox $\mathcal{T}$, an ABox $\mathcal{A}$ and an infinite sequence of actions w of the form $w = \alpha_1, \ldots, \alpha_p(\beta_1, \ldots, \beta_q)^\omega$ all formulated in $\mathcal{ALCO}$, we say that φ is *satisfiable with respect to* $\mathcal{T}$, $\mathcal{A}$ and w iff there is an $\mathcal{ALCO}$-LTL structure $\mathfrak{I} = (\mathcal{I}_i)_{i=0,1,\ldots}$ such that

- $\mathcal{I}_0 \models \mathcal{A}$
- $\mathcal{I}_i \Rightarrow^{\mathcal{T}}_{w(i)} \mathcal{I}_{i+1}$ for $i \geq 0$.
- $\mathfrak{I}, 0 \models \varphi$

We say that φ is *valid with respect to* $\mathcal{T}$, $\mathcal{A}$ and w iff for any $\mathcal{ALCO}$-LTL structure $\mathfrak{I} = (\mathcal{I}_i)_{i=0,1,\ldots}$ with $\mathcal{I}_0 \models \mathcal{A}$ and $\mathcal{I}_i \Rightarrow^{\mathcal{T}}_{w(i)} \mathcal{I}_{i+1}$ for $i \geq 0$, we have that $\mathfrak{I}, 0 \models \varphi$.

The validity problem can be polynomially reduced to the satisfiability problem: φ is valid w.r.t $\mathcal{A}$, $\mathcal{T}$ and w iff $\neg\varphi$ is unsatisfiable w.r.t $\mathcal{A}$, $\mathcal{T}$ and w.

We consider two cases. In the first case, the infinite sequence w of actions contain unconditional actions only, whereas in the second case we consider conditional actions . In the following chapters we deal both cases by reducing the satisfiability problem to the ABox consistency w.r.t. an acyclic TBox in the first case and to the emptiness problem in a Büchi automaton and ABox consistency w.r.t. an acyclic TBox in the second case. In both cases we discuss different complexity issues concerning the reduction. Since the validity problem can be reduced to the satisfiability problem, we concentrate on satisfiability problem.

In Chapter 4, we focus on satisfiability problem for conditional case. As mentioned, besides ABox consistency problem w.r.t. an acyclic TBox, we reduce the problem to the emptiness problem of a Büchi automaton. The idea is that we for a given $\mathcal{ALCO}$-LTL formula φ we construct its propositional abstraction by replacing each assertion in φ with a propositional variable. This abstraction formula is hence an LTL formula. We check the satisfiability of this constructed formula using automata-based approach i.e, we construct a Büchi automaton to decide the its satisfiability. In the following section we discuss this approach.

2.5 Linear Temporal Logic

In this section we provide a brief introduction to linear temporal logic (LTL), which extends propositional logic by temporal operators. The model of time followed in LTL is linear in the sense that at each moment in time there is a single successor moment. We refer to (Baier & Katoen, 2008) for further detail on LTL. We first present the syntax and semantics of LTL and then define the satisfiability problem in LTL. Later on we provide a brief introduction to Büchi automata. At the end we discuss the emptiness problem in a Büchi automaton and the reduction of the satisfiability problem in LTL to this problem.

2.5.1 Syntax and Semantics

As LTL is propositional logic extended with temporal operators, we suppose that PL is the set of propositional variable. By 2^{PL} we mean the set of all the subsets of PL. The basic temporal operators we use are the X-operator and the U-operator.

Definition 2.14. The set of *LTL formulas* is the smallest set such that

- each propositional variable $p \in \mathsf{PL}$ is a formula;
- if φ is a formula, then so are $\neg\varphi$ and $\mathsf{X}\varphi$;
- if φ and ψ are formulas, then so are $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$ and $(\varphi \mathsf{U} \psi)$.

The temporal modalities $\Box$ ("always") and $\Diamond$ ("eventually") are derived from U-operator as follows:

$$\Diamond\varphi := \mathsf{true}\mathsf{U}\varphi \quad \Box\varphi := \neg\Diamond\neg\varphi$$

where true is abbreviation for $(p \vee \neg p)$ for a propositional variable p.

In propositional logic, interpretations are subsets of PL (the set of propositional variable). The satisfiability of a formula in propositional logic depends on the (non)membership of the propositional variable (occurring in the formula) in the interpretation. In LTL we have to consider time as well, i.e., for the semantics of a formula in LTL we have to consider propositional interpretation and a certain time point. Hence LTL interpretations are a sequence of propositional interpretations.

Definition 2.15. An *LTL structure* M is an infinite sequence of $x_0x_1\ldots$ with $x_i \in 2^{\mathsf{PL}}$ for $i \geq 0$. We define satisfaction of LTL formulas in M at time point $n \in \mathbb{N}$ as follows:

$$\begin{array}{rll} M,n \models p & \text{iff} & p \in x_n, \text{ for all } p \in \mathsf{PL} \\ M,n \models \neg\varphi & \text{iff} & M,n \not\models \varphi \\ M,n \models (\varphi \wedge \psi) & \text{iff} & M,n \models \varphi \text{ and } M,n \models \psi \\ M,n \models (\varphi \vee \psi) & \text{iff} & M,n \models \varphi \text{ or } M,n \models \psi \\ M,n \models \mathsf{X}\varphi & \text{iff} & M,n+1 \models \varphi \\ M,n \models (\varphi\mathsf{U}\psi) & \text{iff} & \exists m \geq n : M,m \models \psi \text{ and } \forall k \text{ with } n \leq k < m : M,k \models \varphi \end{array}$$

An LTL formula φ is *satisfiable* iff there is an LTL structure M such that $M, 0 \models \varphi$. The *satisfiability problem* of an LTL formula φ is to check whether a given LTL formula φ is satisfiable.

2.5.2 Büchi Automata and LTL

Note that one can think of LTL structures as an infinite word over 2^{PL}. Vardi and Wolper were the first to present an automata-based approach for checking satisfiability of an LTL formula (Vardi & Wolper, 1986). The basic idea is that we can construct a Büchi automaton from an LTL formula such that the formula is satisfiable iff the language of the automaton is nonempty.

Definition 2.16. A (non-deterministic) Büchi automaton is a tuple $\mathcal{A} = (Q, \Sigma, I, \Delta, F_1, \ldots, F_n)$, with $n \geq 0$, with

- Q a finite set of *states*;
- Σ a finite *alphabet*;
- $I \subseteq Q$ a set of *initial states*;
- $\Delta \subseteq Q \times Q$ a *transition relation*;
- $F_i \subseteq Q$ a set of *accepting states*, for $1 \leq i \leq n$.

Let $w = a_0 a_1 \cdots \in \Sigma^\omega$. A *run* of $\mathcal{A}$ on w is a word $q_0 q_1 \cdots \in Q^\omega$ such that

- $q_0 \in I$;
- $(q_i, w_i, q_{i+1}) \in \Delta$ for all $i \geq 0$.

A run $q_0, q_1, \ldots$ is *accepting* if the the set $\{i | q_i \in F_j\}$ is infinite for all j with $1 \leq j \leq n$. The ω-language accepted by $\mathcal{A}$ is defined as

$$L(\mathcal{A}) := \{w \in \Sigma^\omega \mid \text{there is an accepting run of } \mathcal{A} \text{ on } w\}$$

.

Note that there are more than one set of accepting states in $\mathcal{A}$ but one can construct a Büchi automaton $\mathcal{A}'$ with only one set of accepting states such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$ (Baier & Katoen, 2008).

The *emptiness problem* for Büchi automata asks, given an automaton $\mathcal{A}$ whether $L(\mathcal{A}) = \emptyset$. In (Vardi, 1996), this problem is shown to be decidable in linear time.

Now we show the reduction of satisfiability problem in LTL to emptiness problem in a Büchi automaton. This construction assumes φ does not contain disjunctions. However this restriction is not problematic as any LTL formula can be polynomially converted to an equivalent LTL formula with no disjunctions.

Definition 2.17.

The *closure* $\mathsf{cl}(\phi)$ of an LTL formula ϕ is the smallest set of formulas closed under taking subformulas of φ and closed under single negations, i.e.,

- $\mathsf{cl}(p) := \{p, \neg p\}$;
- $\mathsf{cl}(\neg\varphi) := \mathsf{cl}(\varphi)$;
- if $\vartheta \in \{\varphi \wedge \psi, \varphi \mathsf{U} \psi\}$, then $\mathsf{cl}(\vartheta) := \{\vartheta, \neg\vartheta\} \cup \mathsf{cl}(\varphi) \cup \mathsf{cl}(\psi)$

- $\mathsf{cl}(\mathsf{X}\varphi) := \{\mathsf{X}\varphi, \neg\mathsf{X}\varphi\} \cup \mathsf{cl}(\varphi)$.

For a given LTL formula φ and an LTL structure $M = x_0 x_1 \dots$ one get a sequence $T_0 T_1 \dots$ of subsets of $\mathsf{cl}(\varphi)$ by setting

$$T_i := \{\vartheta \in \mathsf{cl}(\varphi) \mid M, i \models \vartheta\}$$

The sequence $T_0 T_1 \dots$ is said to be *realized* by M. It follows from the semantics that the formulas occurring in any T_i for each $i \geq 0$ satisfy the conditions specified in the definition of types (that is given below). Hence each T_i in the sequence $T_0 T_1 \dots$ realized by an LTL structure is a type.

Definition 2.18. A *type* for an LTL formula φ is a subset $T \subseteq \mathsf{cl}(\varphi)$ such that:

- $\psi \in T$ iff $\neg\psi \notin T$, for all $\neg\psi \in \mathsf{cl}(\varphi)$;
- $\{\psi, \vartheta\} \subseteq T$ iff $(\psi \wedge \vartheta) \in T$, for all $(\psi \wedge \vartheta) \in \mathsf{cl}(\varphi)$

We denote the set of types for φ by $\mathsf{TP}(\varphi)$. For $T, T' \in \mathsf{TP}(\varphi)$, by $T \rightarrow_{\mathsf{X}} T'$ we mean

- for all $\mathsf{X}\psi \in \mathsf{cl}(\varphi), \mathsf{X}\psi \in T$ iff $\psi \in T'$;
- for all $(\psi \mathsf{U} \vartheta) \in \mathsf{cl}(\varphi)$, we have $(\psi \mathsf{U} \vartheta) \in T$ iff
  - $\vartheta \in T$ or
  - $\psi \in T$ and $(\psi \mathsf{U} \vartheta) \in T'$

Let φ is an LTL formula with n until (U) formulas occurring in it. Further suppose that these until formulas are linearly ordered. We construct a Büchi automaton $\mathcal{A}$ such that φ is satisfiable iff $\mathcal{L}(\mathcal{A}) \neq \emptyset$. The set of states of this automaton is the set of types for φ. Now $\mathcal{A}$ is defined as follows:

Definition 2.19. The automaton $\mathcal{A}_\varphi$ is defined as $(Q, \Sigma, I, \Delta, F_1, \dots, F_n)$:

- $Q = \mathsf{TP}(\varphi)$;
- $\Sigma = 2^{\mathsf{PL}}$;
- $I := \{T \in Q \mid \varphi \in T\}$;
- $\Delta := \{(T, x, T') \mid x \cap \mathsf{cl}(\varphi) = T \cap \mathsf{PL} \text{ and } T \rightarrow_{\mathsf{X}} T'\}$;
- $F_i := \{T \in Q \mid (\psi \mathsf{U} \vartheta) \notin T \text{ or } \vartheta \in T\}$ (for $1 \leq i \leq n$) if the i-th until formula in φ is $(\psi \mathsf{U} \vartheta)$.

The following lemma shows that the satisfiability of φ can be decided by deciding $\mathcal{L}(\varphi)$.

Lemma 2.20. *Let $M = x_0 x_1 \ldots$ be an LTL structure, then we have the following:*

- $M \in \mathcal{L}(\mathcal{A}_\varphi)$ *iff* $M, 0 \models \varphi$.
- *For all* $\vartheta \in \mathsf{cl}(\varphi)$ *we have that* $\vartheta \in T_i$ *iff* $M, i \models \vartheta$

For the proof of the lemma we refer to (Vardi & Wolper, 1994).

Chapter 3

Unconditional-Post Condition

Suppose we are given an acyclic TBox $\mathcal{T}$, an ABox $\mathcal{A}$ and an infinite sequence $w = \alpha_1\alpha_2 \ldots \alpha_p(\beta_1\beta_2 \ldots \beta_q)^\omega$ of unconditional actions for $\mathcal{T}$, all formulated in $\mathcal{ALCO}$. Let φ be an $\mathcal{ALCO}$-LTL formula. Then the satisfiability problem is to check whether φ is satisfiable with respect to $\mathcal{T}$, $\mathcal{A}$ and w. In this chapter we show that this problem is PSpace-complete. The upper bound is shown by the reduction of this problem to ABox consistency w.r.t. an acyclic TBox and the lower bound is shown by reducing projection to this problem. We construct a TBox $\mathcal{T}_{\mathsf{red}}$ and ABoxes $\mathcal{A}_{\mathsf{red}}$ and $\mathcal{A}_\varphi$ from $\mathcal{T}$, $\mathcal{A}$, w and φ in a way that we have the following property:

φ is satisfiable with respect to $\mathcal{A}$, $\mathcal{T}$ and w iff $\mathcal{A}_{\mathsf{red}} \cup \mathcal{A}_\varphi$ is consistent w.r.t. $\mathcal{T}_{\mathsf{red}}$.

Before presenting the reduction we introduce some notations. In the following we call $\mathcal{A}$, $\mathcal{T}$, w and φ the *input*. An $\mathcal{ALCO}$-LTL formula is said to be in *Negation Normal Form* (NNF in short) if it contains LTL negation $\neg$ only in the front of $\mathcal{ALCO}$ assertions. Any formula can be transformed to an equivalent formula in NNF. For this first we introduce the dual operator R of U (Baier & Katoen, 2008). For an $\mathcal{ALCO}$-LTL structure $\mathfrak{I} = (\mathcal{I}_i)_{i=0,1,\ldots}$ we have that $\mathfrak{I}, i \models \phi \mathsf{R} \psi$ iff either for all $k \geq i$ we have that $\mathfrak{I}, k \models \psi$ or there exists $k \geq i$ such that $\mathfrak{I}, k \models \phi$ and for all j with $i \leq j \leq k$, $\mathfrak{I}, j \models \psi$. An $\mathcal{ALCO}$-LTL formula can be transformed into an equivalent formula in NNF by applying the following equivalence preserving transformation rules to it:

$$\begin{array}{rcl}
\neg\neg\phi & \longrightarrow & \phi \\
\neg(\phi \wedge \psi) & \longrightarrow & (\neg\phi \vee \neg\psi) \\
\neg(\phi \vee \psi) & \longrightarrow & (\neg\phi \wedge \neg\psi) \\
\neg\mathsf{X}\phi & \longrightarrow & \mathsf{X}\neg\phi \\
\neg(\phi\mathsf{U}\psi) & \longrightarrow & (\neg\phi\mathsf{R}\neg\psi) \\
\neg(\phi\mathsf{R}\psi) & \longrightarrow & (\neg\phi\mathsf{U}\neg\psi)
\end{array}$$

Given $\mathcal{ALCO}$-LTL formula ϕ in NNF, by definition, the LTL negation occurs only in front of assertions. We can further push the LTL negation into assertions by using the following set of rules:

$$\begin{array}{rcl}
\neg(C(a)) & \longrightarrow & \neg C(a) \\
\neg(r(a,b)) & \longrightarrow & \neg r(a,b) \\
\neg(\neg r(a,b)) & \longrightarrow & r(a,b)
\end{array}$$

From now on we assume that all the formulas are in NNF with the LTL negations pushed to the assertions. We take the approach as presented in (Baader *et al.*, 2005b) to construct $\mathcal{T}_{\mathsf{red}}$ and $\mathcal{A}_{\mathrm{red}}$ from $\mathcal{T}$ and $\mathcal{A}$, and present tableau rules for constructing $\mathcal{A}_\varphi$ from φ. The idea of the reduction is to construct $\mathcal{A}_\varphi$, $\mathcal{A}_{\mathrm{red}}$ and $\mathcal{T}_{\mathsf{red}}$ in a way such that each single model $\mathcal{I}$ (we call it a *reduction model*) of them encodes a sequence of $\mathcal{ALCO}$ interpretations $\mathcal{I}_0, \mathcal{I}_1, \ldots, \mathcal{I}_n$ for $n = p+2q-1$. Through out this chapter we fix $n = p+2q-1$. Based on this sequence we define an $\mathcal{ALCO}$-LTL structure $\mathfrak{I}$ such that the reduction model $\mathcal{I}$ is a model for $\mathcal{T}_{\mathsf{red}}$, $\mathcal{A}_{\mathsf{red}}$, and $\mathcal{A}_\varphi$ if and only if $\mathfrak{I}$ is a witness of the satisfiability of φ w.r.t. $\mathcal{A}$, $\mathcal{T}$ and w. Note that for the case unconditional actions Lemma 2.11 holds: If $\mathcal{J}$ is an $\mathcal{ALCO}$ interpretation and $\mathcal{J}'$ is a result of updating $\mathcal{J}$ by a sequence $\gamma_1, \ldots, \gamma_m$ of unconditional actions then $\mathcal{J}'$ is a result of updating itself by $\gamma_1, \ldots, \gamma_m$. This allows us to construct $\mathfrak{I}$ based on $\mathcal{I}_0, \ldots, \mathcal{I}_n$ as follows:

$$\mathfrak{I}, i := \begin{cases} \mathcal{I}_i & 0 \leq i \leq n \\ \mathcal{I}_{p+q+((i-p) \bmod q)} & n < i \end{cases}$$

In other words $\mathfrak{I}$ is of the form

$$(\mathcal{I}_0, \ldots, \mathcal{I}_{p+q}, \ldots, \mathcal{I}_{p+2q-1}, \mathcal{I}_{p+q}, \ldots, \mathcal{I}_{p+2q-1}, \mathcal{I}_{p+q}, \ldots)$$

We want to make sure that the semantics of the actions is respected, i.e., changes are only due to the application of some actions and nothing else changes in interpretations of primitive concept names and role names. We follow the approach as in (Baader *et al.*, 2005b) by distinguishing two kinds of elements in interpretations. We call an element $d \in \Delta^{\mathcal{I}}$ *named* if $a^{\mathcal{I}} = d$ for some individual a used in the input and *unnamed* otherwise. We make this distinction because the changes caused by an action in concept (non)membership and role (non)membership involving (at least) one unnamed domain element never occur.

Before presenting the reduction, we present the notations we will use. By Sub we mean the set of all concepts appearing in the input and this is closed under taking subconcept. For each $C \in \mathsf{Sub}$ and every $i \leq n$, we introduce a concept name $T_C^{(i)}$ to represent the interpretation of concept C in the i-interpretation encoded by the reduction model. Similarly for every primitive concept A, to represent its interpretation in i-interpretation encoded by the reduction model for all $i \leq n$, we introduce a concept name $A^{(i)}$. This interpretation of A involves named elements only. For the unnamed part of the interpretation of A, since the concept membership of unnamed elements never changes, we can get it in $A^{(0)}$. Similarly we introduce role name $r^{(i)}$ for every role name r in the input and every $i \leq n$. By $r^{(i)}$ we denote the interpretation of r in the i-th interpretation encoded by the reduction model but records role relationships where both involved domain elements are named. We denote the set of exactly name elements in the interpretations by concept name N. We denote the set of individual names occurring in the input with Obj.

Now we discuss the components of $\mathcal{T}_{\mathsf{red}}$. The first component states that N denotes exactly the named domain elements.

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

The second component is $\mathcal{T}_{\mathsf{Sub}}$. It enforces the restriction of changes in unnamed elements. Further it also contains one concept definition for every $i \leq n$ and every $C \in \mathsf{Sub}$ that is not defined concept name in $\mathcal{T}$. These concept definitions ensure that $T_C^{(i)}$ stands for the interpretation of C in

i-interpretation encoded by the reduction model:

$$
\begin{array}{rcl}
T^{(i)}_A & \equiv & (N \sqcap A^{(i)}) \sqcup (\neg N \sqcap A^{(0)}) \quad \text{if } A \text{ is primitive in } \mathcal{T} \\
T^{(i)}_{\neg C} & \equiv & \neg T^{(i)}_C \\
T^{(i)}_{C \sqcap D} & \equiv & T^{(i)}_C \sqcap T^{(i)}_D \\
T^{(i)}_{C \sqcup D} & \equiv & T^{(i)}_C \sqcup T^{(i)}_D \\
T^{(i)}_{\exists r C} & \equiv & \big(N \sqcap (\exists r^{(0)}.(\neg N \sqcap T^{(i)}_C) \sqcup \exists r^{(i)}.(N \sqcap T^{(i))}_C)) \sqcup (\neg N \sqcap \exists r^{(0)}.T^{(i)}_C)\big) \\
T^{(i)}_{\forall r C} & \equiv & \big(N \rightarrow (\forall r^{(0)}.(N \sqcup T^{(i)}_C) \sqcap \forall r^{(i)}.(N \rightarrow T^{(i))}_C)) \sqcap (\neg N \rightarrow \forall r^{(0)}.T^{(i)}_C\big)
\end{array}
$$

The TBox $\mathcal{T}_{\mathsf{red}}$ is now given as follows:

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

The last component of $\mathcal{T}_{\mathsf{red}}$ ensures that all definitions from the input TBox $\mathcal{T}$ are satisfied by all interpretations $\mathcal{I}_0, \mathcal{I}_1, \ldots, \mathcal{I}_n$ encoded by the reduction model.

Now we discuss the components of the ABox $\mathcal{A}_{\text{red}}$. For any ABox assertion ϕ we define $\phi^{(i)}$ as follows:

$$\phi^{(i)} := \begin{cases} T^{(i)}_C(a) & \text{if } \phi = C(a) \\ r^{(i)}(a,b) & \text{if } \phi = r(a,b) \\ \neg r^{(i)}(a,b) & \text{if } \phi = \neg r(a,b) \end{cases} \qquad (*)$$

The first component $\mathcal{A}^{(i)}_{\text{post}}$ (for $1 \leq i \leq n$) of $\mathcal{A}_{\text{red}}$ formalizes satisfaction of the post-conditions by $\mathcal{I}_i$ in the reduction model $\mathcal{I}$.

$$\mathcal{A}^{(i)}_{\text{post}} = \{\psi^{(i)} | \psi \in w(i-1)\}$$

For any given action in w the minimization of change is enforced by the ABox $\mathcal{A}^{(i)}_{\text{min}}$. For $1 \leq i \leq n$, $\mathcal{A}^{(i)}_{\text{min}}$ contains the following assertion:

- for all $a \in \mathsf{Obj}$ and primitive concept name A in the input with $\neg A(a) \notin w(i-1)$:

  $$a : (A^{(i-1)} \rightarrow A^{(i)})$$

  for all $a \in \mathsf{Obj}$ and $A \in \mathsf{Prim}$ with $A(a) \notin w(i-1)$:

  $$a : (\neg A^{(i-1)} \rightarrow \neg A^{(i)})$$

- for all $a, b \in \mathsf{Obj}$ and role name r in the input with $\neg r(a,b) \notin w(i-1)$.

$$a : (\exists r^{(i-1)}.\{b\} \rightarrow \exists r^{(i)}.\{b\})$$

for all $a, b \in \mathsf{Obj}$ and role name r with $r(a,b) \notin w(i-1)$.

$$a : (\forall r^{(i-1)}.\neg\{b\} \rightarrow \forall r^{(i)}.\neg\{b\})$$

The last component of $\mathcal{A}_{\text{red}}$ is $\mathcal{A}_{\text{ini}}$ ensuring that the first interpretation of the encoded sequence is a model of the input ABox $\mathcal{A}$.

$$\mathcal{A}_{\text{ini}} := \{\phi^{(0)} \mid \phi \in \mathcal{A}\}$$

Now we define $\mathcal{A}_{\text{red}}$ as

$$\mathcal{A}_{\text{red}} := \mathcal{A}_{\text{ini}} \cup \bigcup_{1 \leq i \leq n} \mathcal{A}^{(i)}_{\text{post}} \cup \bigcup_{1 \leq i \leq n} \mathcal{A}^{(i)}_{\text{min}}$$

The construction of $\mathcal{T}_{\mathsf{red}}$ and $\mathcal{A}_{\mathsf{red}}$ is inspired by (Baader *et al.*, 2005b) and we use the proof of Lemma 15 in their work to formulate the following lemma.

Lemma 3.1. *Given an acyclic TBox $\mathcal{T}$, an ABox $\mathcal{A}$ and a sequence of action $w = \alpha_1 \ldots \alpha_p(\beta_1 \ldots \beta_q)^\omega$, and $\mathcal{T}_{\mathsf{red}}$ and $\mathcal{A}_{\mathsf{red}}$ constructed as above, we have:*

- *for a sequence of $\mathcal{ALCO}$ interpretations $\mathcal{I}_0, \ldots, \mathcal{I}_n$ with $\mathcal{I}_0 \models \mathcal{A}$ and $\mathcal{I}_i \Rightarrow^{\mathcal{T}}_{w(i)} \mathcal{I}_{i+1}$ for $0 \leq i < n$, there exists an $\mathcal{ALCO}$ interpretation $\mathcal{J}$ such that $\mathcal{J} \models \mathcal{T}_{\mathsf{red}}$, $\mathcal{J} \models \mathcal{A}_{\mathsf{red}}$ and for any assertion ϕ and $0 \leq i < n$, $\mathcal{I}_i \models \phi$ iff $\mathcal{J} \models \phi^{(i)}$.*

- *for an $\mathcal{ALCO}$ interpretation $\mathcal{J}$ such that $\mathcal{J} \models \mathcal{T}_{\mathsf{red}}$ and $\mathcal{J} \models \mathcal{A}_{\mathsf{red}}$, there exists $\mathcal{ALCO}$ interpretations $\mathcal{I}_0, \ldots, \mathcal{I}_n$ such that $\mathcal{I}_0 \models \mathcal{A}$ and $\mathcal{I}_i \Rightarrow^{\mathcal{T}}_{w(i)} \mathcal{I}_{i+1}$ for $0 \leq i < n$, and for any assertion ϕ and $0 \leq i < n$, $\mathcal{I}_i \models \phi$ iff $\mathcal{J} \models \phi^{(i)}$.*

Finally we describe how to construct the ABox $\mathcal{A}_\varphi$ from the given $\mathcal{ALCO}$-LTL formula φ. First introduce formulas of the form $\varphi^{(i)}$ with φ a $\mathcal{ALCO}$-LTL formula and $i \in \mathbb{N}$. We call such formulas *labeled $\mathcal{ALCO}$-LTL formulas.* For a given $\mathcal{ALCO}$-LTL structure $\mathfrak{I} = (\mathcal{I}_i)_{i=0,1,\ldots}$ the superscript i in $\varphi^{(i)}$ indicates the interpretation of φ under $\mathcal{I}_i$. As mentioned, we reduce the problem of satisfiability to ABox consistency w.r.t an acyclic TBox. We construct $\mathcal{A}_\varphi$ from φ such that φ is satisfiable with respect to $\mathcal{T}_{\mathsf{red}}$, $\mathcal{A}_{\text{red}}$ and w if and only the ABox $\mathcal{A}_\varphi \cup \mathcal{A}_{\text{red}}$ is consistent with respect to $\mathcal{T}_{\mathsf{red}}$. For constructing $\mathcal{A}_\varphi$ from φ we use tableau rules that are presented in Figure 3.1 where we have the following:

- in $\wedge$-rule:
  $A' := A \setminus \{(\phi_1 \wedge \phi_2)^{(i)}\} \ \cup \ \{\phi_1^{(i)}, \phi_2^{(i)}\}$
- in $\vee$-rule:
  $A' := A \setminus \{(\phi_1 \wedge \phi_2)^{(i)}\} \ \cup \ \{\phi_1^{(i)}\}$
  $A'' := A \setminus \{(\phi_1 \wedge \phi_2)^{(i)}\} \ \cup \ \{\phi_2^{(i)}\}$
- in X-rule I:
  $A' := A \setminus \{(\mathsf{X}\phi)^{(i)}\} \ \cup \ \{\phi^{(i+1)}\}$
- in X-rule II:
  $A' := A \setminus \{(\mathsf{X}\phi)^{(i)}\} \ \cup \ \{\phi^{(p+q)}\}$
- in U-rule I:
  $A_k := A\setminus\{(\phi_1\wedge\phi_2)^{(i)}\} \cup \{\phi_1^{(i)}, \ldots, \phi_1^{(k-1)}, \phi_2^{(k)}\}$ for all k with $i \leq k \leq n$.
- in U-rule II:
  $A_k := A \setminus \{(\phi_1 \wedge \phi_2)^{(i)}\} \ \cup \ \{\phi_1^{(i)}, \ldots, \phi_1^{(n)}, \phi_1^{(p+q)}, \ldots, \phi_1^{(k-1)}, \phi_2^{(k)}\}$ for all k with $p + q \leq k < i$.
- in R-rule I:
  $A_k := A \setminus \{(\phi_1 \wedge \phi_2)^{(i)}\} \ \cup \ \{\phi_2^{(i)}, \ldots, \phi_2^{(k)}, \phi_1^{(k)}\}$ for all k with $i \leq k \leq n$.
  $A' := \{\phi_2^{(i)}, \phi_2^{(i+1)}, \ldots, \phi_2^{(p+2q-1)}\}$.
- in R-rule II:
  $A_k := A \setminus \{(\phi_1 \wedge \phi_2)^{(i)}\} \ \cup \ \{\phi_2^{(i)}, \ldots, \phi_2^{(n)}, \phi_2^{(p+q)}, \ldots, \phi_2^{(k)}, \phi_1^{(k)}\}$ for all k with $p + q \leq k < i$.
  $A' := \{\phi_2^{(i)}, \ldots, \phi_2^{(n)}, \phi_2^{(p+q)}, \phi_2^{(p+q+1)}, \ldots, \phi_2^{(i-1)}\}$.

We start with the set $\mathcal{S}_0 = \big\{\{\varphi^{(0)}\}\big\}$, and apply the tableau rules to the set $\mathcal{S}_0$ producing a sequence of sets $\mathcal{S}_0 \to \mathcal{S}_1 \to \mathcal{S}_2 \to \ldots$ until no more rules are applicable. Note that $\mathcal{S}_0$ is a set with the only element $\{\varphi^{(0)}\}$. The tableau rules are such that each set in the sequence contains sets of the labeled $\mathcal{ALCO}$-LTL formulas. Given a set $A \in \mathcal{S}_i$ for some $i \geq 0$ we call A *complete* if no more rules apply to it.

Lemma 3.2. *Let $\mathcal{S}$, $\mathcal{S}'$ be any two sets in the process of tableau rule application such that $\mathcal{S}'$ is obtained from $\mathcal{S}$ by applying a tableau rule to $A \in \mathcal{S}$. Then for any $\mathcal{ALCO}$-LTL structure $\mathfrak{I} = (\mathcal{I}_i)_{i=0,1,\ldots}$ with $\mathcal{I}_i \Rightarrow^{\mathcal{T}}_{w(i)} \mathcal{I}_{i+1}$ for $i \geq 0$, the following statements are equivalent:*

- *$\mathfrak{I}, i \models \phi$ for all $\phi^{(i)} \in A$.*
- *there exists a set $A' \in \mathcal{S}' \setminus \mathcal{S}$ such that $\mathfrak{I}, i \models \phi$ for all $\phi^{(i)} \in A'$.*

$$\frac{A \in \mathcal{S} \;\wedge\; (\phi_1 \wedge \phi_2)^{(i)} \in A}{\mathcal{S}' := \mathcal{S} \setminus \{A\} \;\cup\; \{A'\}} \;\wedge\text{-rule}$$

$$\frac{A \in \mathcal{S} \;\wedge\; (\phi_1 \vee \phi_2)^{(i)} \in A}{\mathcal{S}' := A \setminus \{(\phi_1 \vee \phi_2)^{(i)}\} \;\cup\; \{A', A''\}} \;\vee\text{-rule}$$

$$\frac{A \in \mathcal{S} \;\wedge\; (\mathsf{X}\phi)^{(i)} \in A \;\wedge\; i < n}{\mathcal{S}' := \mathcal{S} \setminus \{A\} \;\cup\; \{A'\}} \;\mathsf{X}\text{-rule I}$$

$$\frac{A \in \mathcal{S} \;\wedge\; (\mathsf{X}\phi)^{(i)} \in A \;\wedge\; i = n}{\mathcal{S}' := \mathcal{S} \setminus \{A\} \;\cup\; \{A'\}} \;\mathsf{X}\text{-rule II}$$

$$\frac{A \in \mathcal{S} \;\wedge\; (\phi_1 \mathsf{U} \phi_2)^{(i)} \in A \;\wedge\; i \leq p+q}{\mathcal{S}' := \mathcal{S} \setminus \{A\} \;\cup\; \{A_i, \dots, A_n\}} \;\mathsf{U}\text{-rule I}$$

$$\frac{A \in \mathcal{S} \;\wedge\; (\phi_1 \mathsf{U} \phi_2)^{(i)} \in A \;\wedge\; p+q < i}{\mathcal{S}' := \mathcal{S} \setminus \{A\} \;\cup\; \{A_{p+q}, \dots, A_n\}} \;\mathsf{U}\text{-rule II}$$

$$\frac{A \in \mathcal{S} \;\wedge\; (\phi_1 \mathsf{R} \phi_2)^{(i)} \in A \;\wedge\; i \leq p+q}{\mathcal{S}' := \mathcal{S} \setminus \{A\} \;\cup\; \{A_i, \dots, A_n, A'\}} \;\mathsf{R}\text{-rule I}$$

$$\frac{A \in \mathcal{S} \;\wedge\; (\phi_1 \mathsf{R} \phi_2)^{(i)} \in A \;\wedge\; i \leq p+q}{\mathcal{S}' := \mathcal{S} \setminus \{A\} \;\cup\; \{A_{p+q}, \dots, A_n, A'\}} \;\mathsf{R}\text{-rule II}$$

Figure 3.1: Tableau rules.

Proof: Suppose that $\mathcal{S}' = \mathcal{S} \setminus \{A\} \cup \{A_1, \dots, A_n\}$ and by the tableau rules we know that $n \leq p + 2q$. Further let $\phi^{(i)} \in A$ be the formula to which the tableau rule is applied. According to the tableau rule, each $A' \in \{B_1, \dots, B_n\}$ is obtained from A by removing $\phi^{(i)}$ from A and adding finitely many strict subformulas of $\phi^{(i)}$ to A. It, therefore, suffices to show that $\mathfrak{I}, i \models \phi^{(i)}$ if and only if $\mathfrak{I}$ models all the newly added sub formulas in A' for some $A' \in \{A_1, \dots, A_n\}$. Note that by Lemma 2.11, $\mathfrak{I}$ is of form:

$$(\mathcal{I}_0, \dots, \mathcal{I}_p, \dots, \mathcal{I}_{p+q}, \dots, \mathcal{I}_n, \mathcal{I}_{p+q} \dots, \mathcal{I}_n, \mathcal{I}_{p+q}, \dots)$$

- The lemma holds trivially for $\wedge$-rule and $\vee$-rule, and for X-rule it follows from the form of $\mathfrak{I}$.
- U-rule I: Let $(\phi_1 \mathsf{U} \phi_2)^{(i)}$ be the formula in A removed as the consequence of the rule application. Then by the conditions of U-rule I, we know

that for all $i \leq p+q$ we have that:
$\mathfrak{I}, i \models (\phi_1 \mathsf{U} \phi_2)$
iff (by Semantics)
$\exists k \geq i : \mathfrak{I}, k \models \phi_2 \wedge \forall j (i \leq j < k) : \mathfrak{I}, j \models \phi_1$
iff (by Lemma 2.11)
$\exists k$ with $i \leq k \leq n : \mathfrak{I}, k \models \phi_2 \wedge \forall j (i \leq j < k) : \mathfrak{I}, j \models \phi_1$.
iff
$\exists k$ with $i \leq k \leq n$ such that $\mathfrak{I}, k \models \phi_2$, $\mathfrak{I}, i \models \phi_1$,..., $\mathfrak{I}, (k-1) \models \phi_1$
iff (by U-rule I)
$\mathfrak{I}, i \models \psi$ for all newly added subformulas $\psi^{(i)}$ in A_k.

- U-rule II: Let $(\phi_1 \mathsf{U} \phi_2)^{(i)}$ with $i \leq p+q$ be the formula in A removed as the consequence of the rule application. By U-rule II we have that:
  $\mathfrak{I}, i \models (\phi_1 \mathsf{U} \phi_2)$
  iff (by Semantics)
  $\exists k \geq i : \mathfrak{I}, k \models \phi_2 \wedge \forall j (i \leq j < k) : \mathfrak{I}, j \models \phi_1$
  iff (by Lemma 2.11)

  - $\exists k$ with $i \leq k \leq n$ such that $\mathfrak{I}, k \models \phi_2$ and $\mathfrak{I}, j \models \phi_1$ for j with $i \leq j < k$ or

  - $\exists k$ with $p+q \leq k < i$ such that $\mathfrak{I}, k \models \phi_2$ and $\mathfrak{I}, j \models \phi_1$ for all j with $i \leq j$ and for all j with $p+q \leq j < k$.

  iff

  - $\exists k$ with $i \leq k \leq n$ such that $\mathfrak{I}, k \models \phi_2, \mathfrak{I}, i \models \phi_1, \ldots, \mathfrak{I}, k-1 \models \phi_1$ or

  - $\exists k$ with $p+q \leq k < i$ such that $\mathfrak{I}, k \models \phi_2, \mathfrak{I}, i \models \phi_1, \ldots, \mathfrak{I}, n \models \phi_1$ and $\mathfrak{I}, p+q \models \phi_1, \ldots, \mathfrak{I}, k \models \phi_1$.

  It is equivalent to $\mathfrak{I}, i \models \psi$ for all newly added subformulas $\psi^{(i)}$ in A_k.

  Similarly, the case for R-rule I and R-rule II can be proved from the semantics of these operators and Lemma 2.11. □

For constructing $\mathcal{A}_\varphi$ from φ we start with the set $\{\{\varphi^{(0)}\}\}$ and apply tableau rules until we get a set $\mathcal{S}'$ of all complete sets. By applying the rules we introduced for pushing LTL negation into assertions, it follows from

(*) that completes sets can be viewed as ABoxes. One of these complete sets serves as $\mathcal{A}_\varphi$. With non-determinism we know which of the complete set is $\mathcal{A}_\varphi$. In the following lemma we prove that the construction of $\mathcal{A}_\varphi$ is in PSpace and hence the termination of the tableau rules application is immediately implied.

Lemma 3.3. *Given an $\mathcal{ALCO}$-LTL formula φ, the construction of φ_{red} from φ, using the tableau rules presented, is in PSpace.*

Proof: Let $\mathcal{S}$ be a set in some status of the tableau algorithm starting with $\{\{\varphi^{(0)}\}\}$ and let $\mathcal{S}'$ be the set obtained by an application of one of the tableau rules to a set $A \in \mathcal{S}$. According to the tableau rules, $\mathcal{S}'$ is obtained by removing A from $\mathcal{S}$ and adding new sets say $A_1, \ldots, A_k$ (we call them *successors* of A) with some $k \leq p + 2q$. Now we have following:

- Each set can have at most polynomially many successors, i.e., $k \leq p + 2q$, and hence the number of successors of a set is polynomial in the size of the input.
- According to the tableau rules, each successor A_j of a set A for some j with $1 \leq j \leq k$ is obtained by removing a labeled formula, say $\phi^{(i)}$, from A and adding polynomially many labeled strict subformulas $\psi^{(i')}$ of $\phi^{(i)}$ with $i' \leq n$.
- A rule is applicable if and only if there is a set in $\mathcal{S}$ containing a labeled formula with at least one LTL formula occurring in it.

Starting from $\{\{\varphi^{(0)}\}\}$, we non-deterministically generate a complete set $\mathcal{A}_\varphi$. Hence overall this construction is in NPSpace. And by Savitch's theorem (Papadimitriou, 1993), the construction is in PSpace. □

In the following we show that the consistency of $\mathcal{A}_{\mathsf{red}} \cup \mathcal{A}_\varphi$ w.r.t. $\mathcal{T}_{\mathsf{red}}$ indeed decides the satisfiability of φ w.r.t $\mathcal{T}$, $\mathcal{A}$ and w.

Lemma 3.4. *φ is satisfiable with respect to $\mathcal{A}$, $\mathcal{T}$ and w if and only if $\mathcal{A}_\varphi \cup \mathcal{A}_{red}$ is consistent with respect to $\mathcal{T}_{\mathsf{red}}$.*

Proof: "$\Longleftarrow$"
Suppose that $\mathcal{A}_\varphi \cup \mathcal{A}_{\text{red}}$ is consistent with respect to $\mathcal{T}_{\mathsf{red}}$. Then there is a model $\mathcal{J}$ of $\mathcal{T}_{\mathsf{red}}$ such that $\mathcal{J} \models \mathcal{A}_\varphi \cup \mathcal{A}_{\text{red}}$. We have to to show that φ is satisfiable with respect to $\mathcal{A}$, $\mathcal{T}$ and w. By Lemma 3.1, there are $\mathcal{ALCO}$ interpretations $\mathcal{I}_0, \ldots, \mathcal{I}_{p+2q-1}$ such that $\mathcal{I}_0 \models \mathcal{A}$ and $\mathcal{I}_{i-1} \Rightarrow^{\mathcal{T}}_{w(i-1)} \mathcal{I}_i$ for $1 \leq i \leq p + 2q - 1$. Now let $\mathfrak{I}$ be the following $\mathcal{ALCO}$-LTL structure:

$$(\mathcal{I}_0, \ldots, \mathcal{I}_{p+q}, \ldots, \mathcal{I}_{p+2q-1}, \mathcal{I}_{p+q}, \ldots, \mathcal{I}_{p+2q-1}, \mathcal{I}_{p+q}, \ldots)$$

By Lemma 2.11 we have that $\mathcal{I}_i \Rightarrow^{\mathcal{T}}_{w(i)} \mathcal{I}_{i+1}$ for all $i \geq 0$. Further since $\mathcal{J} \models \mathcal{A}_\varphi$ we have that $\mathcal{J} \models \phi^{(i)}$ for all $\phi^{(i)} \in \mathcal{A}_\varphi$. By Lemma 3.1 we get that $\mathcal{I}_i \models \phi$ and therefore $\mathfrak{I}, i \models \phi$ for all $\phi^{(i)} \in \mathcal{A}_\varphi$. By Lemma 3.2, we get that $\mathfrak{I}, 0 \models \varphi$. Hence φ is satisfiable w.r.t. $\mathcal{T}$, $\mathcal{A}$ and w.
"$\Rightarrow$"
Let us suppose that φ is satisfiable with respect to $\mathcal{T}$, $\mathcal{A}$ and w. Therefore there is an $\mathcal{ALCO}$-LTL structure $\mathfrak{I} = (\mathcal{I}_i)_{i=0,1,\ldots}$ satisfying the following:

- $\mathcal{I}_0 \models \mathcal{A}$
- $\mathcal{I}_i \Rightarrow^{\mathcal{T}}_{w(i)} \mathcal{I}_{i+1}$ for $i \geq 0$.
- $\mathfrak{I}, 0 \models \varphi$

Now consider the sequence of $\mathcal{ALCO}$-interpretations $\mathcal{I}_0, \ldots, \mathcal{I}_{p+2q-1}$. By Lemma 3.1 there is a model $\mathcal{J}$ of $\mathcal{T}_{\mathsf{red}}$ such that $\mathcal{J} \models \mathcal{A}_{\text{red}}$. We have to show that $\mathcal{J} \models \mathcal{A}_\varphi$. But since $\mathfrak{I}, 0 \models \varphi$ and $\mathcal{A}_\varphi$ is constructed from φ by the application of tableau rules, it follows from Lemma 3.2 that $\mathcal{I}_i \models \phi$ for all $\phi^{(i)} \in \mathcal{A}_\varphi$. Therefore by Lemma 3.1, we get that $\mathcal{J} \models \phi^{(i)}$ for all $\phi^{(i)} \in \mathcal{A}_\varphi$ and hence $\mathcal{J} \models \mathcal{A}_\varphi$. Therefore $\mathcal{A}_{\mathsf{red}} \cup \mathcal{A}_\varphi$ is consistent w.r.t. $\mathcal{T}_{\mathsf{red}}$. □

Lemma 3.5. *Given an acyclic TBox $\mathcal{T}$, an ABox $\mathcal{A}$ and an infinite sequence of unconditional actions $w = \alpha_1, \ldots, \alpha_p(\beta_1, \ldots, \beta_q)^\omega$ all formulated in $\mathcal{ALCO}$, the satisfiability of an $\mathcal{ALCO}$-LTL formula φ w.r.t. $\mathcal{T}$, $\mathcal{A}$ and w is in PSpace.*

Proof: We have seen that the construction $\mathcal{A}_\varphi$ of from φ is in PSpace. We also know that the ABox consistency problem w.r.t. to an acyclic TBox is PSpace-complete for $\mathcal{ALCO}$ (Schaerf, 1994). Further since the size of $\mathcal{A}_{\text{red}}$ and $\mathcal{T}_{\mathsf{red}}$ are polynomial in the size of the input (Baader *et al.*, 2005b), Lemma 3.4 immediately yields Lemma 3.5. □

Theorem 3.6. *The satisfiability and validity problems are PSpace-complete.*

Proof: We have seen that we can polynomially reduce the validity problem to the satisfiability problem. Further we have seen in Lemma 3.5 that the satisfiability problem is in PSpace. Hence validity problem is also in PSpace. In (Baader *et al.*, 2005b), the projection problem for $\mathcal{ALCO}$ is shown to be PSpace-complete. We reduce the projection problem to the validity problem as follows: Let φ be an assertion, $\mathcal{T}$ an acyclic TBox, $\mathcal{A}$ an ABox and $\alpha_1, \ldots, \alpha_p$ be unconditional actions for $\mathcal{T}$ all formulated in $\mathcal{ALCO}$. We set $w := \alpha_1, \ldots, \alpha_p(\emptyset)^\omega$. Further we set $\bar{\varphi} := \mathsf{X}^p\varphi$. It is easy to see that φ is a consequence of applying $\alpha_1, \ldots, \alpha_p$ in $\mathcal{A}$ w.r.t. $\mathcal{T}$ iff $\bar{\varphi}$ is valid w.r.t. $\mathcal{T}$, $\mathcal{A}$ and w. Hence the validity problem is PSpace-complete and therefore, so is the satisfiability problem. □

Note that given an an acyclic TBox $\mathcal{T}$, and $\mathcal{I}$ and $\mathcal{I}'$ models of $\mathcal{T}$, for an unconditional-action α for $\mathcal{T}$ with $\mathcal{I} \Rightarrow_{\alpha}^{\mathcal{T}} \mathcal{I}'$ we have that all of the post-conditions of α always holds in $\mathcal{I}'$ regardless of $\mathcal{I}$. Hence for unconditional actions we can show that if an interpretation $\mathcal{I}'$ is a result of updating an interpretation $\mathcal{I}$ by a sequence of unconditional actions, then we get again $\mathcal{I}'$ as a result of updating $\mathcal{I}'$ by the sequence, i.e., a sequence of interpretations runs into cycle after updating it by a sequence of actions twice (Lemma 2.11). We have seen that the definition of the witness $\mathcal{ALCO}$-LTL structure in the reduction depends on this property. In the next chapter we will consider conditional actions where we will see that in case of conditional actions, Lemma 2.11 would not hold anymore.

Chapter 4

Conditional Actions

In the previous chapter we dealt with the satisfiability and validity problem of an $\mathcal{ALCO}$-LTL formula φ with respect to a given acyclic TBox $\mathcal{T}$, an ABox $\mathcal{A}$ and an infinite sequence w of unconditional actions for $\mathcal{T}$. We constructed TBox a $\mathcal{T}_{\mathsf{red}}$, and ABoxes $\mathcal{A}_{\mathsf{red}}$ and $\mathcal{A}_\varphi$ such that we had the following:

$$\varphi \text{ is satisfiable w.r.t. } \mathcal{T}, \mathcal{A} \text{ and } w \text{ iff } \mathcal{A}_{\mathsf{red}} \cup \mathcal{A}_\varphi \text{ is consistent w.r.t. } \mathcal{T}_{\mathsf{red}}.$$

For unconditional actions we have the property that if an interpretation $\mathcal{I}$ is a result of updating another interpretation by a sequence of unconditional action, then updating $\mathcal{I}$ by this sequence yields $\mathcal{I}$. This we have shown in Lemma 2.11. This property allowed us to consider finitely many $\mathcal{ALCO}$ interpretations in dealing with the satisfiability problem. Although an $\mathcal{ALCO}$-LTL is an infinite sequence of $\mathcal{ALCO}$ interpretations, we defined such a structure considering only finitely many interpretations. In other words, an $\mathcal{ALCO}$-LTL structure for the satisfiability problem is infinite sequence of finitely many different interpretations. In this chapter we consider the general case where we deal with conditional actions. We will see that Lemma 2.11 does not hold any more when considering conditional actions. Hence the reduction we used in the previous chapter, does not work. Nevertheless we will see that we still need to consider finitely many different interpretations in this case and can define a witness $\mathcal{ALCO}$-LTL structure based on these interpretations for the satisfiability of φ w.r.t. $\mathcal{T}$, $\mathcal{A}$ and w.

Let us first provide a simple counter example to Lemma 2.11 for the case of conditional actions. Let $\mathcal{T}$ be an acyclic TBox and $\alpha = \{\varphi/\neg\phi, \neg\phi/\psi\}$ an action for $\mathcal{T}$. Let $\mathcal{I}$, $\mathcal{I}'$ and $\mathcal{I}''$ be interpretations such that $\mathcal{I} \Rightarrow_\alpha^\mathcal{T} \mathcal{I}' \Rightarrow_\alpha^\mathcal{T} \mathcal{I}''$. Further suppose that for the interpretation $\mathcal{I}$ we have that $\mathcal{I} \models \varphi$, $\mathcal{I} \models \neg\psi$ and $\mathcal{I} \not\models \neg\phi$. By Definition 2.10 $\mathcal{I}' \models \neg\phi$ and $\mathcal{I}' \models \neg\psi$ since the only changes are due to the post-conditions. Also that $\mathcal{I}'' \models \psi$. Hence we get that $\mathcal{I}' \neq \mathcal{I}''$.

Recall that given an $\mathcal{ALCO}$-LTL formula φ, an acyclic TBox $\mathcal{T}$, an ABox $\mathcal{A}$ and an infinite sequence $w = \alpha_1 \ldots \alpha_p(\beta_1 \ldots \beta_q)^\omega$ of actions for $\mathcal{T}$ we say that φ is satisfiable with respect to $\mathcal{T}$, $\mathcal{A}$ and w iff there is an $\mathcal{ALCO}$-LTL structure $\mathfrak{I} = (\mathcal{I}_i)_{i=0,1,\ldots}$ such that

- $\mathcal{I}_0 \models \mathcal{A}$
- $\mathcal{I}_i \Rightarrow^{\mathcal{T}}_{w(i)} \mathcal{I}_{i+1}$ for $i \geq 0$
- $\mathfrak{I}, 0 \models \varphi$

The validity of φ w.r.t. $\mathcal{T}$, $\mathcal{A}$, and w requires that for each $\mathcal{ALCO}$-LTL structure $\mathfrak{I} = (\mathcal{I}_i)_{i=0,1,\ldots}$ with $\mathcal{I}_0 \models \mathcal{A}$ and $\mathcal{I}_i \Rightarrow^{\mathcal{T}}_{w(i)} \mathcal{I}_{i+1}$ for $i \geq 0$, we have $\mathfrak{I}, 0 \models \varphi$. Note that we are considering an infinite sequence of conditional actions. Given a Büchi automaton $B_{\mathsf{act}} = (Q_{\mathsf{act}}, \Sigma_{\mathsf{act}}, \Delta_{\mathsf{act}}, I_{\mathsf{act}}, F_{\mathsf{act}})$, we extend the definition of satisfiability and validity problem in this chapter as follows:

Definition 4.1. We say φ is satisfiable with respect to $\mathcal{T}$, $\mathcal{A}$ and B_{act} iff there is a $w \in \mathcal{L}(B_{\mathsf{act}})$ such that φ is satisfiable with respect to $\mathcal{T}$, $\mathcal{A}$ and w. φ is valid w.r.t. $\mathcal{T}$, $\mathcal{A}$ and B_{act} iff for all $w \in \mathcal{L}(B_{\mathsf{act}})$ we have that φ is valid with respect to $\mathcal{T}$, $\mathcal{A}$ and w.

The satisfiability (validity) problem asks whether φ is satisfiable (valid) with respect to $\mathcal{T}$, $\mathcal{A}$ and B_{act}. We can reduce the validity problem polynomially to the satisfiability problem: φ is valid w.r.t. $\mathcal{T}$, $\mathcal{A}$ and B_{act} iff $\neg\varphi$ is not satisfiable w.r.t. $\mathcal{T}$, $\mathcal{A}$ and B_{act}. From now on, we concentrate on satisfiability problem. Note that this case is more general than the case of unconditional actions in the sense that the satisfiability problem for the case of unconditional actions can be decided using the method for deciding the satisfiability problem for conditional actions. Since an infinite sequence of actions is an infinite word over a set of action, given an infinite sequence w of actions, we can build a Büchi automata $\mathcal{B}_w$ such that $\mathcal{L}(\mathcal{B}_w) = \{w\}$. Now an $\mathcal{ALCO}$-LTL formula φ is satisfiable with respect to an acyclic TBox $\mathcal{T}$, an ABox $\mathcal{A}$ and w iff it is satisfiable w.r.t. $\mathcal{T}$, $\mathcal{A}$ and $\mathcal{B}_w$.

To decide the satisfiability problem, we first introduce some notations. We call the given $\mathcal{T}$, $\mathcal{A}$, w and φ the *input*. Since any $\mathcal{ALCO}$-LTL formula ϕ can be transformed into an equivalent formula ϕ' containing no LTL operator other than $\neg$, $\wedge$ and U, we will assume that φ contains only $\neg$, $\wedge$ and U as the LTL operators. We make this assumption because we will construct a Büchi automaton from an LTL formula $\bar{\varphi}$ that we will obtain from φ and

$\mathcal{A}$. In Section 2.5.2 we have seen that such a construction assumes LTL formula without disjunction. We denote the set of assertions occurring in the input and assertions of the form $A(a)$ or $r(a, b)$ for each concept name A, role name r and individual names a, b in the input, by Assert. We introduce a propositional variable p_ϑ for each $\vartheta \in \mathsf{Assert}$, such that there is a $1 - 1$ relationship between each ϑ and the corresponding propositional variables p_ϑ introduced. We denote the set of these propositional variables by PL. For an $\mathcal{ALCO}$-LTL formula ϕ occurring in the input, by $\hat{\phi}$ we mean its *propositional abstraction* obtained by replacing each assertion in ϕ with its corresponding propositional variable. Note that for an assertion $\vartheta \in \mathsf{Assert}$ and an $\mathcal{ALCO}$ interpretation $\mathcal{I}$, we have either $\mathcal{I} \models \vartheta$ or $\mathcal{I} \not\models \vartheta$ but not both. As PL is finite, for an $\mathcal{ALCO}$-LTL structure $\mathfrak{I} = (\mathcal{I}_i)_{i=0,1,\ldots}$, there cannot be infinitely many different sets X_i defined as following:

$$X_i := \{p_\vartheta \in \mathsf{PL} \mid \mathcal{I}_i \models \vartheta \text{ and } \vartheta \in \mathsf{Assert}\}$$

In other words, for any $\mathcal{ALCO}$-LTL structure, $\mathfrak{I} = (\mathcal{I}_i)_{i=0,1,\ldots}$ there is a set $\{X_1, \ldots, X_k\} \subseteq 2^{\mathsf{PL}}$ (we call it the *set induced* by $\mathfrak{I}$) such that for each $i \geq 0$ there is a $\iota_i \in \{1, \ldots, k\}$ with $X_i = X_{\iota_i}$. We will use this characteristic of such an $\mathcal{ALCO}$-LTL structure in deciding the satisfiability problem. We reduce the satisfiability problem to emptiness problem of a Büchi automaton and knowledge base consistency problem in $\mathcal{ALCO}$. Given an $\mathcal{ALCO}$-LTL formula φ, an acyclic TBox $\mathcal{T}$, an ABox $\mathcal{A}$ and a Büchi automaton B_{act} (with set of actions as its alphabet), we guess a set $\mathcal{S} = \{X_1, \ldots, X_k\} \subseteq 2^{\mathsf{PL}}$ and construct a Büchi automaton $B_\mathcal{S}$, an acyclic TBox $\mathcal{T}_{\mathsf{red}}$ and an ABox $\mathcal{A}_\mathcal{S}$ such that

$$\varphi \text{ is satisfiable w.r.t. } \mathcal{T}, \mathcal{A} \text{ and } B_{\mathsf{act}} \text{ iff } \mathcal{L}(B_\mathcal{S}) \neq \emptyset \text{ and } \mathcal{A}_\mathcal{S} \text{ is consistent w.r.t. } \mathcal{T}_{\mathsf{red}}.$$

In the case of unconditional actions, the reduction model was an encoding of a finite sequence of $\mathcal{ALCO}$ interpretations. The witness $\mathcal{ALCO}$-LTL structure was defined based on these interpretations. In this chapter, we define the witness $\mathcal{ALCO}$-LTL structure based on a model of $\mathcal{T}_{\mathsf{red}} \cup \mathcal{A}_\mathcal{S}$ and elements of set $\mathcal{S}$. Since $\mathcal{S}$ is finite, we have to consider finitely many interpretations in defining the witness $\mathcal{ALCO}$-LTL structure.

To respect the semantics of actions, we again distinguish between the two kinds of elements in interpretations; *named* and *unnamed* elements. The minimization of changes on named elements are achieved by proper encoding of the states and transition relation of $B_\mathcal{S}$ whereas that on unnamed elements are enforced by the TBox $\mathcal{T}_{\mathsf{red}}$. In the following we discuss the idea behind the reduction.

First we construct the Büchi automaton $B_{\mathcal{S}}$. We set

$$\bar{\varphi} := \hat{\varphi} \wedge \bigwedge_{\vartheta \in \mathcal{A}} p_\vartheta$$

Recall that $\hat{\varphi}$ is the propositional abstraction of φ. With out loss of generality we assume that each variable in PL occurs in $\bar{\varphi}$. If not we can conjunct the propositional tautology $p \vee \neg p$ to the abstraction of φ for each propositional variable p in PL that does not occur in the abstraction. Note that $\bar{\varphi}$ is a propositional LTL formula. The closure of the formula $\bar{\varphi}$ is defined in the same way as in Definition 2.18 and is denote by $\mathsf{cl}(\bar{\varphi})$. As discussed Section 2.5.2 we can construct a Büchi automaton to check the satisfiability of $\bar{\varphi}$ by checking the emptiness problem of the automaton . Let $B_{\mathsf{LTL}} = (\mathsf{TP}(\bar{\varphi}), 2^{\mathsf{PL}}, \Delta_{\mathsf{LTL}}, I_{\mathsf{LTL}}, F_1, \ldots, F_n)$ be the corresponding automaton for $\bar{\varphi}$.

We define $B_{\mathcal{S}} := (Q, \Sigma, \Delta, I, F'_1, \ldots, F'_{n+1})$ as following:

1. $Q := \mathsf{TP}(\bar{\varphi}) \times Q_{\mathsf{act}}$
2. $\Sigma := 2^{\mathsf{PL}} \times \Sigma_{\mathsf{act}}$
3. $I := \{(T, q) \mid \bar{\varphi} \in T,\ q \in I\}$
4. $\langle (T, q), (x, \alpha), (T', q') \in \Delta$ iff we have the following:
   - $(T, x, T') \in \Delta_{\mathsf{LTL}}$
   - $(q, \alpha, q') \in \Delta_{\mathsf{act}}$
   - $x \in \mathcal{S}$
   - Let $p_{A(a)} \in \mathsf{PL}$ for a primitive concept name A and an individual name a occurring in the input:
     - if $p_{A(a)} \in T$ and there is no $\phi/\neg A(a) \in \alpha$ such that $p_\phi \in T$ then $p_{A(a)} \in T'$.
     - if $\neg p_{A(a)} \in T$ and there is no $\phi/A(a) \in \alpha$ such that $p_\phi \in T$ then $\neg p_{A(a)} \in T'$.
   - Let $p_{r(a,b)} \in \mathsf{PL}$ for a role name r and individual names a, b occurring in the input:
     - if $p_{r(a,b)} \in T$ and there is no $\phi/\neg r(a, b) \in \alpha$ such that $p_\phi \in T$ then $p_{r(a,b)} \in T'$.
     - if $\neg p_{r(a,b)} \in T$ and there is no $\phi/r(a, b) \in \alpha$ such that $p_\phi \in T$ then $\neg p_{r(a,b)} \in T'$.

- For each $\phi/\vartheta \in \alpha$, if $p_\phi \in T$ then $p_\vartheta \in T'$.

5. $F'_i := \{(T, q) \mid T \in F_i\}$ for $1 \leq i \leq n$ and $F'_{n+1} := \{(T, q) \mid q \in F_{\mathsf{act}}\}$

Later on we will define an $\mathcal{ALCO}$-LTL structure based on an accepting run of $B_\mathcal{S}$. Note that the last three points in the definition of Δ enforces the transitions of $B_\mathcal{S}$ to respect the semantics of actions on named elements. Nevertheless this enforcement is up to propositional level, i.e., up to propositional abstraction of actions.

Now we switch towards the construction of $\mathcal{A}_\mathcal{S}$ and $\mathcal{T}_{\mathsf{red}}$. As mentioned previously, we will define an $\mathcal{ALCO}$-LTL structure based on an accepting run of $B_\mathcal{S}$. Nevertheless this is not sufficient. As an example consider that we have assertions $\forall r.A(a)$, $r(a, b)$ and $\neg A(b)$ in $\mathcal{A}$. Let p, q and r be their corresponding propositional variables. It might be the case that $M, 0 \models \{p, q, r\}$ for an LTL structure M but there is no $\mathcal{ALCO}$ interpretation $\mathcal{I}$ such that $\mathcal{I} \models \{\forall r.A(a), r(a, b), \neg A(b)\}$. We construct $\mathcal{A}_\mathcal{S}$ in such a way that such problems are avoided.

$$\mathcal{A}^{(i)} := \{\psi^{(i)} \mid p_\psi \in X_i\} \cup \{\neg\psi^{(i)} \mid p_\psi \notin X_i\} \text{ for } 1 \leq i \leq k$$

$$\mathcal{A}_\mathcal{S} := \bigcup_{1 \leq i \leq k} A^{(i)}$$

where $\psi^{(i)}$ is defined in the same way as in Chapter 3 and the negation $\neg$ in $\neg\psi^{(i)}$ can be push into ψ by the rules we introduced there.

Besides the above problem, the interpretation $\mathfrak{I}$ needs to enforce the domain constraints imposed by $\mathcal{T}$ and the restriction of changes on unnamed elements. This we will achieved by $\mathcal{T}_{\mathsf{red}}$. The definition of $\mathcal{T}_{\mathsf{red}}$ is quite similar as in the previous case of unconditional actions. The only difference is that we consider i with $1 \leq i \leq k$ in the definition.

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

Lemma 4.2. *φ is satisfiable w.r.t. $\mathcal{T}$, $\mathcal{A}$ and B_{act} iff there is a set $\mathcal{S} \subseteq 2^{PL}$ such that $\mathcal{L}(B_\mathcal{S}) \neq \emptyset$ and $\mathcal{A}_\mathcal{S}$ is consistent w.r.t. $\mathcal{T}_{red}$.*

Proof: "$\Rightarrow$"
Suppose that φ is satisfiable w.r.t. $\mathcal{T}$, $\mathcal{A}$ and B_{act}. Then there is an infinite sequence of actions $w \in \mathcal{L}(B_{\mathsf{act}})$ such that φ is satisfiable w.r.t. $\mathcal{T}$, $\mathcal{A}$ and w. Let $q_0, q_1, \ldots$ be the accepting run of B_{act} on w. Moreover there is an $\mathcal{ALCO}$-LTL structure $\mathfrak{I} = (\mathcal{I}_i)_{i=0,1,\ldots}$ such that

- $\mathcal{I}_0 \models \mathcal{A}$;

- $\mathcal{I}_i \Rightarrow^{\mathcal{T}}_{w(i)} \mathcal{I}_{i+1}$ for $i \geq 0$;
- $\mathfrak{I}, 0 \models \varphi$.

Let $\mathcal{S} = \{X_1, \ldots, X_k\}$ be the set induced by $\mathfrak{I}$. We define $x_i := \{p_\vartheta \in \mathsf{PL} \mid \mathcal{I}_i \models \vartheta \text{ and } \vartheta \in \mathsf{Assert}\}$ for $i \geq 0$. Note that for each $i \geq 0$ there is a $\iota_i \in \{1, \ldots, k\}$ such that $x_i = X_{\iota_i}$. We set

$$T_i := \{\hat{\phi} \in \mathsf{cl}(\bar{\varphi}) \mid \mathfrak{I}, i \models \phi\} \tag{4.1}$$

Recall that $\hat{\phi}$ is the propositional abstraction of ϕ. It is easy to see that each T_i satisfies the following:

- $\psi \in T_i$ iff $\neg\psi \notin T_i$, for all $\neg\psi \in \mathsf{cl}(\bar{\varphi})$;
- $\{\psi, \vartheta\} \subseteq T_i$ iff $\psi \wedge \vartheta \in T_i$, for all $\psi \wedge \vartheta \in \mathsf{cl}(\bar{\varphi})$.

Hence each T_i is a type for $\bar{\varphi}$. Now we show that $T_0, T_1, \ldots$ is an accepting run of B_{LTL} on $x_0 x_1 \ldots$. By definition

$$\bar{\varphi} = \hat{\varphi} \wedge \bigwedge_{\vartheta \in \mathcal{A}} p_\vartheta$$

Since $\mathfrak{I}, 0 \models \varphi$ and $\mathcal{I}_0 \models \mathcal{A}$, we get that $\hat{\varphi} \in T_0$ and $\{p_\vartheta \mid \vartheta \in \mathcal{A}\} \subseteq T_0$. By definition of types we get that $\bar{\varphi} \in T_0$. Hence $T_0 \in I_{\mathsf{LTL}}$. We now show that $(T_i, x_i, T_{i+1}) \in \Delta_{\mathsf{LTL}}$ for $i \geq 0$ as follows:

- it is easy to see that $x_i \cap \mathsf{cl}(\bar{\varphi}) = T_i \cap \mathsf{PL}$.
- $T_i \rightarrow_{\mathsf{X}} T_{i+1}$ for $i \geq 0$:
  - Let $\mathsf{X}\hat{\phi} \in \mathsf{cl}(\bar{\varphi})$. $\mathsf{X}\hat{\phi} \in T_i$ iff (by (4.1)) $\mathfrak{I}, i \models \mathsf{X}\phi$ (semantics of X-operator)iff $\mathfrak{I}, i+1 \models \phi$ iff (by (4.1)) $\hat{\phi} \in T_{i+1}$.
  - Let $(\hat{\phi_1} \mathsf{U} \hat{\phi_2}) \in \mathsf{cl}(\bar{\varphi})$. $(\hat{\phi_1} \mathsf{U} \hat{\phi_2}) \in T_i$ iff (by (4.1)) $\mathfrak{I}, i \models (\phi_1 \mathsf{U} \phi_2)$ iff (by semantics of U-operator) $\exists k \geq i$ such that $\mathfrak{I}, k \models \phi_2$ and $\forall i \leq j < k$ we have that $\mathfrak{I}, j \models \phi_1$ iff $\mathfrak{I}, i \models \phi_2$ or $\exists k > i$ such that $\mathfrak{I}, k \models \phi_2$ and $\mathfrak{I}, j \models \phi_1$ for all j with $i \leq j < k$ iff (by (4.1)) $\hat{\phi_2} \in T_i$ or $\hat{\phi_1} \in T_i$ and $(\hat{\phi_1} \mathsf{U} \hat{\phi_2}) \in T_{i+1}$.

Hence $T_0 T_1 \ldots$ is a run of B_{LTL} on $x_0 x_1 \ldots$. We have to show that it is an accepting run. By our assumption B_{LTL} has n sets of final states, namely $F_1, \ldots, F_n$. Suppose that $T_0 T_1 \ldots$ is not accepting. It means that for some $j \in \{1, \ldots, n\}$, the set $\{i \in \mathbb{N} \mid T_i \in F_j\}$ is finite and, therefore, there exists an $i_0 \in \mathbb{N}$ such that $T_i \notin F_j$ for all $i \geq i_0$. Let $(\hat{\phi_1} \mathsf{U} \hat{\phi_2})$ be the j-th until subformula in $\bar{\varphi}$. By definition of set of the final states of B_{LTL} we get that $(\hat{\phi_1}\mathsf{U}\hat{\phi_2}) \in T_i$ and $\hat{\phi_2} \notin T_i$, for all $i \geq i_0$. Hence, by (4.1), $\mathcal{I}_{i_0} \models (\phi_1 \mathsf{U} \phi_2)$ and $\mathcal{I}_i \not\models \phi_2$ for all $i \geq i_0$, which is a contradiction to the semantics of U.

To show that $\mathcal{L}(B_{\mathsf{LTL}}) \neq \emptyset$ we construct a word $u = u_1 u_2 \dots$ such that $u \in \mathcal{L}(B_{\mathcal{S}})$ by setting $u_i := (x_i, w(i))$. Note that by definition of Σ we have that $u_0 u_1 \dots \in \Sigma^*$. We set $r_i := (T_i, q_i)$ for $i \geq 0$ and show that $r_0 r_1 \dots$ is an accepting run of $B_{\mathcal{S}}$ on $u_0 u_1 \dots$ as follows:

- $(T_0, q_0) \in I$: as $T_0 \in I_{\mathsf{LTL}}$ and $q_0 \in I_{\mathsf{act}}$.
- $\langle (T_i, q_i), (x_i, w(i)), (T_{i+1}, q_{i+1}) \rangle \in \Delta$ for all $i \geq 0$:
  - $(T_i, x_i, T_{i+1}) \in \Delta_{\mathsf{LTL}}$.
  - $(q_i, w(i), q_{i+1}) \in \Delta_{\mathsf{act}}$.
  - $x_i \in \mathcal{S}$ follows from the definition of x_i
  - Let $p_{A(a)} \in \mathsf{PL}$ for a primitive concept name A and individual name a occurring in the input.
    * Suppose that $p_{A(a)} \in T_i$ and there is no $\phi/\neg A(a) \in w(i)$ such that $p_\phi \in T_i$. We have to show that $p_{A(a)} \in T_{i+1}$. Suppose, to contrary, that $p_{A(a)} \notin T_{i+1}$. By definition of types, therefore, $\neg p_{A(a)} \in T_{i+1}$. By (4.1) $\mathfrak{I}, i{+}1 \not\models A(a)$, i.e., $\mathcal{I}_{i+1} \not\models A(a)$. But as $\mathcal{I}_i \Rightarrow^{\mathcal{T}}_{w(i)} \mathcal{I}_{i+1}$ and there is no $\phi/\neg A(a) \in w(i)$, therefore, $\mathcal{I}_{i+1} \models A(a)$ which is a contradiction.
    * Similarly one can show that if $\neg p_{A(a)} \in T_i$ and there is no $\phi/A(a) \in w(i)$ such that $p_\phi \in T_i$, then $p_{A(a)} \in T_{i+1}$.
  - With similar arguments as in the above case, we can show that for $p_{r(a,b)} \in T_i$ with r a role name, and a and b individual names, occurring in the input:
    * if $p_{r(a,b)} \in T_i$ and there is not $\phi/\neg r(a,b) \in w(i)$ such that $p_\phi \in T_i$ then $p_{r(a,b)} \in T_{i+1}$.
    * if $\neg p_{r(a,b)} \in T_i$ and there is not $\phi/r(a,b) \in w(i)$ such that $p_\phi \in T_i$ then $\neg p_{r(a,b)} \in T_{i+1}$.
  - Let $\phi/\vartheta \in w(i)$ with $p_\phi \in T_i$. By 4.1 we have that $\mathfrak{I}, i \models \phi$ and hence $\mathcal{I}_i \models \phi$. It follows from Definition 2.10 that $\mathcal{I}_{i+1} \models \vartheta$ and hence $\mathfrak{I}, i+1 \models \vartheta$. Again by (4.1) we get that $p_\vartheta \in T_{i+1}$.
- Since $T_0 T_1 \dots$ is an accepting run of B_{LTL} on $x_0 x_1 \dots$, the set $\{i \mid T_i \in F_j\}$ is infinite for $1 \leq j \leq n$. Hence we get that the set $\{i \mid r_i =$

$(T_i, q_i) \in F'_j\}$ for $1 \leq j \leq n$ is infinite as $F'_j = \{(T, q) \mid T \in F_j\}$. Similarly as $q_0 q_1 \ldots$ is an accepting run of B_{act} on w, the set $\{i \mid T_i \in F_{\mathsf{act}}\}$ is infinite as well which implies that the set $\{i \mid r_i = (T_i, q_i) \in F'_{n+1}\}$ is infinite as $F'_{n+1} := \{(T, q) \mid q \in F_{\mathsf{act}}\}$. Hence $r = r_0 r1 \ldots$ is an accepting run of $B_{\mathcal{S}}$ on $u = u_0 u_1 \ldots$.

Therefore we get that $u_0 u_1 \cdots \in \mathcal{L}(B_{\mathcal{S}})$.

Finally we show that the knowledge base $\mathcal{A}_{\mathcal{S}}$ is consistent w.r.t. $\mathcal{T}_{\mathsf{red}}$. For this we define an $\mathcal{ALCO}$ interpretation $\mathcal{J}$ as follows:

- $\Delta^{\mathcal{J}} := \Delta^{\mathcal{I}_0} (= \Delta^{\mathcal{I}_1} = \Delta^{\mathcal{I}_2} = \ldots)$
- $a^{\mathcal{J}} := a^{\mathcal{I}_0} (= a^{\mathcal{I}_1} = a^{\mathcal{I}_2} = \ldots)$ for an individual name a in the input.
- $N^{\mathcal{J}} := \{a^{\mathcal{J}} \mid a \in \mathsf{Obj}\}$
- for each i with $1 \leq i \leq k$ there is an $\iota_i \geq 0$ (not necessarily unique) such that $\{p_\vartheta \mid \mathfrak{I}, \iota_i \models \vartheta\} = X_i$. Now for each concept name A, role name r and $1 \leq i \leq k$:
  - $(A^{(i)})^{\mathcal{J}} := A^{\mathcal{I}_{\iota_i}}$
  - $(r^{(i)})^{\mathcal{J}} := r^{\mathcal{I}_{\iota_i}}$
  - $(T_C^{(i)})^{\mathcal{J}} := C^{\mathcal{I}_{\iota_i}}$
- whereas $(A^{(0)})^{\mathcal{J}}$ and $(r^{(0)})^{\mathcal{J}}$ are defined as follows:
  - $(A^{(0)})^{\mathcal{J}} := A^{\mathcal{I}_0}$
  - $(r^{(0)})^{\mathcal{J}} := r^{\mathcal{I}_0}$

Note that the definition of $\mathcal{J}$ implies the following:
Claim 1. For an assertion $\vartheta \in \mathsf{Assert}$ and $1 \leq i \leq k$:

$$\mathcal{I}_{\iota_i} \models \vartheta \text{ iff } \mathcal{J} \models \vartheta^{(i)}$$

Depending on ϑ we have to make the following case distinctions.

- $\vartheta = C(a)$ for a concept name C and individual name a occurring in the input:
  By definition, $\vartheta^{(i)} = T_C^{(i)}(a)$. Now $\mathcal{J} \models \vartheta^{(i)}$ iff $\mathcal{J} \models T_C^{(i)}(a)$ iff (by semantics) $a^{\mathcal{J}} \in (T_C^{(i)})^{\mathcal{J}}$ iff (by definition of $\mathcal{J}$) $a^{\mathcal{I}_{\iota_i}} \in (C)^{\mathcal{I}_{\iota_i}}$ iff $\mathcal{I}_{\iota_i} \models \vartheta$.

- $\vartheta = r(a,b)$ for a role name r and individual names a and b occurring in the input:
  By definition, $\vartheta^{(i)} = r^{(i)}(a,b)$. Now $\mathcal{J} \models \vartheta^{(i)}$ iff $\mathcal{J} \models r^{(i)}(a,b)$ iff (by semantics) $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in (r^{(i)})^{\mathcal{J}}$ iff (by definition of $\mathcal{J}$) $(a^{\mathcal{I}_{\iota_i}}, b^{\mathcal{I}_{\iota_i}}) \in r^{\mathcal{I}_{\iota_i}}$ iff $\mathcal{I}_{\iota_i} \models r(a,b) = \vartheta$.

- The case of $\vartheta = \neg r(a,b)$ can be proved analogously.

This finishes the proof of Claim 1.

Now we show that $\mathcal{J} \models \mathcal{T}_{\mathsf{red}}$. First note that $\mathcal{J} \models \mathcal{T}_N$ follows from the definition of $N^{\mathcal{J}}$. Since each $\mathcal{I}_i$ for $i \geq 0$ is a model of $\mathcal{T}$ it follows from the definition of $\mathcal{J}$ that $\mathcal{J} \models \{T_A^{(i)} \equiv T_E^{(i)} \mid A \equiv E \in \mathcal{T}, 1 \leq i \leq k\}$. To show that $\mathcal{J} \models \mathcal{T}_{\mathsf{Sub}}$ it suffices to show that for any concept occurring in the input $\mathcal{J}$ satisfies the every concept definition in $\mathcal{T}_{\mathsf{Sub}}$ with $T_E^{(i)}$ on the left-hand side. This we can show by induction on structure of E. For the proof we refer to (Baader *et al.*, 2005b). However in their work $\mathcal{J}$ is defined based on a sequence of interpretation that satisfy the semantics of the actions. This additional property of the interpretation is not required in showing $\mathcal{J} \models \mathcal{J}_{\mathsf{Sub}}$. At the end we show that $\mathcal{J} \models \mathcal{A}_{\mathcal{S}}$. Consider the set $X_i \in \mathcal{S}$ for $i \in \{1, \ldots, k\}$. For each $p_\vartheta \in X_i$ and $p_\psi \notin X_i$ it follows from definition of X_i that $\mathcal{I}_{\iota_i} \models \vartheta$ and $\mathcal{I}_{\iota_i} \not\models \psi$ respectively. Hence it follows from Claim 1 that $\mathcal{J} \models \vartheta^{(i)}$ and $\mathcal{J} \not\models \psi^{(i)}$ respectively. Therefore we get that $\mathcal{J} \models \mathcal{A}^{(i)}$ and hence $\mathcal{J} \models \mathcal{A}_{\mathcal{S}}$.

"$\Leftarrow$"
Let $\mathcal{S} = \{X_1, \ldots, X_k\}$ with $X_i \subseteq \mathsf{PL}$ for $1 \leq i \leq k$ such that $\mathcal{L}(B_{\mathcal{S}}) \neq \emptyset$ and $\mathcal{A}_{\mathcal{S}}$ is consistent w.r.t. $\mathcal{T}_{\mathsf{red}}$. Therefore there is a word $u_0 u_1 \cdots \in \mathcal{L}(B_{\mathcal{S}})$ and a model $\mathcal{J}$ of $\mathcal{T}_{\mathsf{red}}$ such that $\mathcal{J} \models \mathcal{A}_{\mathcal{S}}$. As $\Sigma = 2^{\mathsf{PL}} \times \Sigma_{\mathsf{act}}$, we suppose that $u_i = (x_i, w(i))$ with $x_i \in 2^{\mathsf{PL}}$ and $w(i) \in \Sigma_{\mathsf{act}}$ for $i \geq 0$. Further let $r_0 r_1 \ldots$, with $r_i = (T_i, q_i)$ for all $i \geq 0$, be an accepting run of $B_{\mathcal{S}}$ on $x_0 x_1 \ldots$. It follows from the construction of $B_{\mathcal{S}}$ that each T_i for $i \geq 0$ is a type for $\bar{\varphi}$. Now we show that $q_0 q_1 \ldots$ is an accepting run of B_{act} on $w(0)w(i) \ldots$. Note that

- $q_0 \in I_{\mathsf{act}}$ as $r_0 = (T_0, q_0) \in I)$.

- $(q_i, w(i), q_{i+1}) \in \Delta_{\mathsf{act}}$ is implied by Δ.

Hence $q_0 q_1 \ldots$ is an accepting run of B_{act} on $w(0)w(i) \ldots$ as $(T_0, q_0)(T_1, q_1) \ldots$ is accepting $B_{\mathcal{S}}$. Similarly we can show that $T_0 T_1 \ldots$ is an accepting run of B_{LTL} on $x_0 x_1 \ldots$.

Note that the definition of Δ implies for each $i \geq 0$ we have that $x_i \in \mathcal{S}$ and as each $x_i \in 2^{\mathsf{PL}}$, therefore for each $i \geq 0$ there is $\iota_i \in \{1, \ldots, k\}$ such

that $x_i = X_{\iota_i}$. Now we define an $\mathcal{ALCO}$-LTL structure $\mathfrak{I} = (\mathcal{I}_i)_{i=0,1,\dots}$ as follows:
For each $i \geq 0$ and individual name a, concept name A and role name r occurring in the input:

- $\Delta^{\mathcal{I}_i} := \Delta^{\mathcal{J}}$
- $a^{\mathcal{I}_i} := a^{\mathcal{J}}$
- $A^{\mathcal{I}_i} := (T_A^{(\iota_i)})^{\mathcal{J}}$
- $r^{\mathcal{I}_i} := (r^{(\iota_i)})^{\mathcal{J}} \cap (N^{\mathcal{J}} \times N^{\mathcal{J}}) \;\cup\; (r^{(0)})^{\mathcal{J}} \cap (\Delta^{\mathcal{J}} \times (\neg N)^{\mathcal{J}} \cup (\neg N)^{\mathcal{J}} \times \Delta^{\mathcal{J}})$

It follows from the definition of $\mathfrak{I}$ that the following holds:

- For individual name a and b and role name r occurring in the input we have
$$(a^{\mathcal{I}_i}, b^{\mathcal{I}_i}) \in r^{\mathcal{I}_i} \text{ iff } (a^{\mathcal{J}}, b^{\mathcal{J}}) \in (r^{(\iota_i)})^{\mathcal{J}}$$
For elements x and y of $\Delta^{\mathcal{J}}$, such that either x is not occurring in the input or y, and each role name r occurring in the input we have
$$(x, y) \in r^{\mathcal{I}_i} \text{ iff } (x, y) \in (r^{(0)})^{\mathcal{J}}$$

One can also show by structural induction that $E^{\mathcal{I}_i} = (T_E^{(\iota_i)})^{\mathcal{J}}$ for every concept description E in the input or subconcept of E (for proof we refer to (Baader *et al.*, 2005b)). Based on these properties of $\mathfrak{I}$, it is easy to see that for each assertion ϑ in the input we have:

$$\mathcal{J} \models \vartheta^{(\iota_i)} \text{ iff } \mathcal{I}_i \models \vartheta \quad (^{**})$$

The proof of $(**)$ is similar to the proof of Claim 1.
Claim 2. For each $\hat{\psi} \in \mathsf{cl}(\bar{\varphi})$ we have the following $\hat{\psi} \in T_i$ iff $\mathfrak{I}, i \models \psi$.
By Lemma 2.20 it suffices to show that

$$M, i \models \hat{\psi} \text{ iff } \mathfrak{I}, i \models \psi$$

where M is the LTL structure defined as $M = x_0 x_1 \dots$. We prove by induction on structure of $\hat{\psi}$.

- If $\hat{\psi} = p_\psi \in \mathsf{PL}$
$M, i \models p_\psi$ iff (by Lemma 2.20) $p_\psi \in T_i$ iff (by definition of Δ: $T_i \cap \mathsf{PL} = x_i \cap \mathsf{cl}(\bar{\varphi})$) $p_\psi \in x_i$ iff $p_\psi \in X_{\iota_i}$ iff (as $\mathcal{J} \models \mathcal{A}_{\mathcal{S}}$) $\mathcal{J} \models \psi^{(\iota_i)}$ iff $(^{**})$ $\mathcal{I}_i \models \psi$ and hence $\mathfrak{I}, i \models \psi$.

- $\hat{\psi} = (\hat{\psi}_1 \mathsf{U} \hat{\psi}_2)$:
  $M, i \models (\hat{\psi}_1 \mathsf{U} \hat{\psi}_2)$ iff (by semantics of U) $\exists k \geq i$ such that $M, k \models \hat{\psi}_2$ and for all j with $i \leq j < k$ we have that $M, j \models \hat{\psi}_1$ iff (induction hypothesis) $\exists k \geq i$ such that $\mathfrak{I}, k \models \psi_2$ and for all j with $i \leq j < k$ we have that $\mathfrak{I}, j \models \psi_1$ iff (by semantics of U) iff $\mathfrak{I}, i \models (\psi_1 \mathsf{U} \psi_2)$.

- Similarly one can prove the case for $\hat{\psi} = \neg\hat{\phi}$ and $\hat{\psi} = (\hat{\psi}_1 \wedge \hat{\psi}_2)$.

This finishes the proof of Claim 2. Now we show that $\mathfrak{I}$ satisfies the following:

- $\mathfrak{I}, 0 \models \varphi$ and $\mathcal{I}_0 \models \mathcal{A}$:
  Since $T_0 T_1 \ldots$ is an accepting run of B_{LTL} on $x_0 x_1 \ldots$, therefore $\bar{\varphi} \in T_0$. As T_0 is a type hence $\hat{\varphi} \in T_0$ ($\hat{\varphi}$ is the propositional abstraction of φ) and $p_\vartheta \in T_0$ for each $\vartheta \in \mathcal{A}$. By Claim 2, $\mathfrak{I}, 0 \models \varphi$ and also that $\mathfrak{I}, 0 \models \vartheta$ for each $\vartheta \in \mathcal{A}$. Therefore $\mathcal{I}_0 \models \mathcal{A}$.

- $\mathcal{I}_i \models \mathcal{T}$ for $i \geq 0$:
  As $\mathcal{J} \models \mathcal{T}_{\mathsf{red}}$ therefore $\mathcal{J} \models T_A^{(\iota_i)} \equiv T_E^{(\iota_i)}$ for each $A \equiv E \in \mathcal{T}$ and $1 \leq \iota_i \leq k$. Therefore by definition of $\mathfrak{I}$, $\mathcal{I}_i \models A \equiv E$ for all $A \equiv E \in \mathcal{T}$ and $i \geq 0$. Hence $\mathcal{I}_i \models \mathcal{T}$ for all $i \geq 0$.

- $\mathcal{I}_i \Rightarrow_{w(i)}^{\mathcal{T}} \mathcal{I}_{i+1}$ for all $i \geq 0$:
  First we introduce the following notations for a primitive concept name A, role name r and individual name a and b.

  $$
  \begin{aligned}
  A^+ &= \{a^{\mathcal{I}_i} \mid \vartheta/A(a) \in w(i) \wedge \mathcal{I}_i \models \vartheta\} \text{ for some } \vartheta \in \mathsf{Assert} \\
  A^- &= \{a^{\mathcal{I}_i} \mid \vartheta/\neg A(a) \in w(i) \wedge \mathcal{I}_i \models \vartheta\} \text{ for some } \vartheta \in \mathsf{Assert} \\
  r^+ &= \{(a^{\mathcal{I}_i}, b^{\mathcal{I}_i}) \mid \vartheta/r(a,b) \in w(i) \wedge \mathcal{I}_i \models \vartheta\} \text{ for some } \vartheta \in \mathsf{Assert} \\
  r^- &= \{(a^{\mathcal{I}_i}, b^{\mathcal{I}_i}) \mid \vartheta/\neg r(a,b) \in w(i) \wedge \mathcal{I}_i \models \vartheta\} \text{ for some } \vartheta \in \mathsf{Assert}
  \end{aligned}
  $$

  Now by Definition 2.10 it suffices to show the following the following:

  $$
  \begin{aligned}
  A^{\mathcal{I}_{i+1}} &= (A^{\mathcal{I}_i} \cup A^+) \setminus A^- \\
  r^{\mathcal{I}_{i+1}} &= (r^{\mathcal{I}_i} \cup r^+) \setminus r^-
  \end{aligned}
  $$

  First we show that $A^{\mathcal{I}_{i+1}} = (A^{\mathcal{I}_i} \cup A^+) \setminus A^-$.
  "$\subseteq$":
  Let d be an individual name occurring in the input with $d^{\mathcal{I}_{i+1}} \in A^{\mathcal{I}_{i+1}}$ and suppose that $d^{\mathcal{I}_{i+1}} \notin (A^{\mathcal{I}_i} \cup A^+) \setminus A^-$. It means that either $d^{\mathcal{I}_{i+1}} \notin (A^{\mathcal{I}_i} \cup A^+)$ or $d^{\mathcal{I}_{i+1}} \in A^-$

- $d^{\mathcal{I}_{i+1}} \notin (A^{\mathcal{I}_i} \cup A^+)$:
  Therefore $d^{\mathcal{I}_{i+1}} \notin A^{\mathcal{I}_i}$ and $d^{\mathcal{I}_{i+1}} \notin A^+$. But $d^{\mathcal{I}_{i+1}} = d^{\mathcal{I}_i} \notin A^{\mathcal{I}_i}$ implies that $\mathcal{I}_i \not\models A(d)$ and therefore by Claim 2 $p_{A(d)} \notin T_i$. Since T_i is a type therefore $\neg p_{A(d)} \in T_i$. As $d^{\mathcal{I}_i} \notin A^+$, it means that there is no $\vartheta/A(d) \in w(i)$ such that $\mathcal{I}_i \models \vartheta$. Therefore, there is no $\vartheta/A(d) \in w(i)$ such that $p_\vartheta \notin T_i$. But according to the transitions of $B_\mathcal{S}$, $\neg p_{A(d)} \in T_{i+1}$ and hence by Claim 2 $\mathcal{I}_{i+1} \not\models A(d)$, equivalently $d^{\mathcal{I}_{i+1}} \notin A^{\mathcal{I}_{i+1}}$ which is a contradiction.
- $d^{\mathcal{I}_{i+1}} \in A^-$:
  It means that there is $\vartheta/\neg A(d) \in w(i)$ with $\mathcal{I}_i \models \vartheta$. By Claim 2 there is there is $\vartheta/\neg A(d) \in w(i)$ with $p_\vartheta \in T_i$ (as $\mathfrak{I}, i \models \vartheta$). The transition relation of $B_\mathcal{S}$ implies that $\neg p_{A(d)} \in T_{i+1}$. By Claim 2, therefore, $\mathcal{I}_{i+1} \not\models A(d)$ equivalently $d^{\mathcal{I}_{i+1}} \notin A^{\mathcal{I}_{i+1}}$ which again is a contradiction.

Now let $x \in A^{\mathcal{I}_{i+1}}$ such that x is an unnamed element. It follows from the definition of $\mathfrak{I}$ that $x \in (T_A^{(\iota_{i+1})})^\mathcal{J}$ and as $\mathcal{J} \models \mathcal{T}_{\mathsf{Sub}}$ therefore $x \in \big((N \sqcap A^{(\iota_{i+1})} \sqcup (\neg N \sqcap A^{(0)}))\big)^\mathcal{J}$. But as x is an unnamed element, therefore $x \in (\neg N \sqcap A^{(0)})^\mathcal{J}$ and hence $x \in \big((N \sqcap A^{(\iota_i)} \sqcup (\neg N \sqcap A^{(0)}))\big)^\mathcal{J}$. Again definition of $\mathfrak{I}$ and $\mathcal{J} \models \mathcal{T}_{\mathsf{Sub}}$ implies that $x \in A^{\mathcal{I}_i}$. Since $x \notin A^-$, we get that $x \in (A^{\mathcal{I}_i} \cup A^+) \setminus A^-$.

"$\supseteq$":
Let d be an individual name that occurs in the input and $d^{\mathcal{I}_i} \in (A^{\mathcal{I}_i} \cup A^+) \setminus A^-$. It means that $d^{\mathcal{I}_i} \in (A^{\mathcal{I}_i} \cup A^+)$ (i.e., either $d^{\mathcal{I}_i} \in A^{\mathcal{I}_i}$ or $d^{\mathcal{I}_i} \in A^+$) and $d^{\mathcal{I}_i} \notin A^-$.

- $d^{\mathcal{I}_i} \in A^{\mathcal{I}_i}$:
  By Claim 2 we get that $p_{A(d)} \in T_i$ as $\mathcal{I}_i \models A(d)$. $d^{\mathcal{I}_i} \notin A^-$ implies that there is no $\vartheta/\neg A(d) \in w(i)$ with $\mathcal{I}_i \models \vartheta$, and hence there is no $\vartheta/\neg A(d) \in w(i)$ with $p_\vartheta \in T_i$. The transition relation of $B_\mathcal{S}$ enforces that $p_{A(d)} \in T_{i+1}$. By Claim 2 we get that $\mathfrak{I}, i+1 \models A(d)$ and hence $d^{\mathcal{I}_i} \in A^{\mathcal{I}_{i+1}}$.

- $d^{\mathcal{I}_i} \in A^+$:
  $d^{\mathcal{I}_i} \in A^+$ implies that there is $\phi/A(d) \in w(i)$ with $\mathcal{I}_i \models \phi$ and hence, by Claim2, $p_\phi \in T_i$. It follows from the definition of the transition relation of $B_\mathcal{S}$ that $p_{A(d)} \in T_{i+1}$. Therefore by Claim 2, $d^{\mathcal{I}_{i+1}} \in A^{\mathcal{I}_{i+1}}$.

The case of $r^{\mathcal{I}_{i+1}} = (r^{\mathcal{I}_i} \cup r^+) \setminus r^-$ can be shown with similar arguments. Hence $\mathfrak{I}$ is a witness of the satisfiability of φ w.r.t. $\mathcal{T}$, $\mathcal{A}$ and B_{act}. □

Note that guessing $\mathcal{S}$ requires NExpTime in the size of PL as $S \subseteq 2^{\mathsf{PL}}$ and the size of PL is polynomial in the size of the input. It requires ExpSpace in the size of the input to store $\mathcal{S}$. Further the size of B_{LTL} is exponential in the size of $\hat{\phi}$ which is polynomial in the size of input. As the construction of $B_{\mathcal{S}}$ depends on B_{act}, B_{LTL} and $\mathcal{S}$, by definition of $B_{\mathcal{S}}$, it is easy to see that constructing $B_{\mathcal{S}}$ from B_{act}, B_{LTL} and $\mathcal{S}$ requires polynomial time in their size. Hence we get that the overall construction of $B_{\mathcal{S}}$ is in ExpSpace. The emptiness problem of a Büchi automaton is decidable in linear time in the size of the automaton (Vardi, 1996). Hence the emptiness problem of $B_{\mathcal{S}}$ can be decided in ExpTime.

Similarly the construction of both $\mathcal{T}_{\mathsf{red}}$ and $\mathcal{A}_{\mathcal{S}}$ depends on $\mathcal{S}$. The number of concept definitions in $\mathcal{T}_{\mathsf{Sub}}$, the number of concept definitions in $\mathcal{T}_{\mathsf{red}}$ of the form $T_A^{(i)} \equiv T_E^{(i)}$ for $A \equiv E \in \mathcal{T}$, and the number of assertions in $\mathcal{A}_{\mathcal{S}}$ depend on the number of elements in $\mathcal{S}$ which exponential in the size of the input. Hence the construction of both $\mathcal{T}_{\mathsf{red}}$ and $\mathcal{A}_{\mathcal{S}}$ requires ExpSpace. The consistency problem of an ABox w.r.t. an acyclic TBox in $\mathcal{ALCO}$ is PSpace-complete (Schaerf, 1994). Overall the consistency problem of $\mathcal{A}_{\mathcal{S}}$ w.r.t. $\mathcal{T}_{\mathsf{red}}$, therefore, is in ExpSpace.

Theorem 4.3. *Given an acyclic TBox $\mathcal{T}$ and an ABox $\mathcal{A}$ both formulated in $\mathcal{ALCO}$, and a Büchi automaton B_{act} (with a set of conditional actions, formulated in $\mathcal{ALCO}$, as it alphabet), the satisfiability and validity of an $\mathcal{ALCO}$-LTL formula φ w.r.t. $\mathcal{T}$, $\mathcal{A}$ and B_{act} is in ExpSpace.*

In this chapter we have considered the satisfiability problem of an $\mathcal{ALCO}$-LTL formula w.r.t. an acyclic TBox, an ABox and a Büchi automaton (with set of conditional actions as it alphabet). Unlike the reduction in Chapter 3, we reduced the satisfiability problem for the case of conditional action to the emptiness problem of a Büchi automaton and ABox consistency w.r.t. an acyclic TBox in $\mathcal{ALCO}$. We have also shown that the satisfiability problem in case of conditional actions is in ExpSpace. The lower bound of the problem is still open.

Chapter 5

Conclusion

In this thesis, we have consider merging of temporalized DLs and DL-based action formalisms. We introduced inference problems like the *satisfiability* and *validity* problems and later on provided results on their computational complexity. In Chapter 3, we considered the case of unconditional actions and have shown that the satisfiability and validity problem is PSpace-Complete. Similarly in Chapter 4, we dealt with the case where we allow for conditional actions and consider a set of infinite sequence of conditional actions recognized by a Büchi automaton. The reduction we provided in Chapter 4 is more general in the sense that the satisfiability problem in case of unconditional actions can be decided by the method provided for the case of conditional actions. We have seen that allowing conditional actions, the complexity of the satisfiability and validity problem is in ExpSpace. As the reduction depends on a set whose cardinality is exponential in the number of assertions in the input, this causes the exponential space blow up in deciding the satisfiability problem for the case of conditional actions.

The DL we have considered in this work is $\mathcal{ALCO}$ where the ABox consistency with respect to an acyclic TBox problem is already in PSpace complete(Schaerf, 1994). One can study the problems for the fragments of $\mathcal{ALCO}$ and also for more expressive DLs. Yet another open problem is to provide a lower bound for the satisfiability problem for the case of conditional actions.

The notions of DL-based actions formalisms in this work have been taken from (Baader *et al.*, 2005b) where an action α is defined as $\alpha = (\mathsf{pre}, \mathsf{occ}, \mathsf{post})$, where pre is a finite set of ABox assertions, the pre-conditions which specifies under which conditions the action is applicable and occ is a set of *occlusions* of the form $A(a)$ or $r(a, b)$, with A a primitive concept name and r a role name. The set of post-conditions is denoted by post. The role of occlusions is to describe primitive literals that can undergo changes regardless of the post-conditions in the action. In this work, we considered actions with $\mathsf{pre} = \emptyset$ and $\mathsf{occ} = \emptyset$, i.e., we define an action as a set of post-conditions. One can consider cases where actions contain not only post-conditions, but also pre-conditions and occlusions.

References

Artale, A., & Franconi, E. (2000). A survey of temporal extensions of description logics. *Annals of Mathematics and Artificial Intelligence*, **30**(1-4), 171–210.

Artale, A., Franconi, E., Wolter, F., & Zakharyaschev, M. (2002). A temporal description logic for reasoning over conceptual schemas and queries. *Proceedings of the 8th European Conference on Logics in Artificial Intelligence (JELIA-02)*. Lecture Notes in Artificial Intelligence, vol. 2424. Springer-Verlag.

Artale, A., Kontchakov, R., Lutz, C., Wolter, F., & Zakharyaschev, M. (2007). Temporalising tractable description logics. *Proceedings of TIME-07*. IEEE Press.

Baader, F., & Lutz, C. Description logic. Blackburn, Patrick, van Benthem, Johan, & Wolter, Frank (eds), *The handbook of modal logic*. Elsevier.

Baader, F. Lutz, C. Milicic, M. Sattler, U. & Wolter, F.(2005a). Integrating description logics and action formalisms: First results. *Proceedings of the twentieth national conference on artificial intelligence (AAAI-05)*.

Baader, F., Milicic, M., Lutz, C., Sattler, U., & Wolter, F. (2005b). *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. See http://lat.inf.tu-dresden.de/research/reports.html.

Baader, F., Ghilardi, S., & Lutz, C. (2008a). LTL over Description Logic Axioms. *Proceedings of the 21st International Workshop on Description Logics (DL2008)*. CEUR-WS, vol. 353.

Baader, Franz, Ghilardi, Silvio, & Lutz, Carsten. (2008b). *LTL over description logic axioms*. LTCS-Report 08-01. Chair for Automata Theory, Institute for Theoretical Computer Science, Dresden University of Technology, Germany. See http://lat.inf.tu-dresden.de/research/reports.html.

Baier, Christel, & Katoen, Joost-Pieter. (2008). *Principles of model checking*. The MIT Press, Cambridge, Massachusetts London, England.

Brachman, R. J. & Levesque, H. J.(1984). The tractability of subsumption in framebased description languages. *Proceedings of the 4th national conference of the american association for artificial intelligence (aaai-84)*.

Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, & Peter F. Patel-Schneider (eds). (2003). *The description logic handbook: Theory, implementation, and applications*. Cambridge University Press.

Gabbay, D., Kurucz, A., Wolter, F., & Zakharyaschev, M. (2003). *Many–Dimensional Modal Logics: Theory and Applications*. Elsevier.

Krger, Fred, & Merz, Stephan. (2008). *Temporal logic and state systems (texts in theoretical computer science. an eatcs series)*. Springer Publishing Company, Incorporated.

Milicic, M. (2008). *Action, time and space in description logics*. Ph.D. thesis, Chair for Automata Theory, Institute for Theoretical Computer Science, Faculty of Computer Science at TU Dresden.

Minsky, M. (1974). *A framework for representing knowledge*. Tech. rept. Cambridge, MA, USA.

Papadimitriou, Christos H. (1993). *Computational complexity*. Addison Wesley.

Quillian, M. (1968). Semantic memory. *Pages 227–270 of:* Minsky, M. (ed), *Semantic Information Processing*. MIT Press.

Reiter, R. (2001). *Knowledge in action*. MIT Press.

Schaerf, Andrea. (1994). Reasoning with individuals in concept languages. *Data knowledge engineering*, **13**(2), 141–176.

Schild, K. (1993). Combining terminological logics with tense logic. *Pages 105–120 of: Proceedings EPIA 93*. LNCS, vol. 727. Springer.

Schmidt-Schauß, M., & Smolka, G. (1991). Attributive concept descriptions with complements. *Artif. intell.*, **48**(1), 1–26.

Thielscher, M. (2005). *Reasoning robots: The art and science of programming robotic agent*. Number 33 in Applied Logic Series. Kluwer.

Vardi, Moshe Y. (1996). An automata-theoretic approach to linear temporal logic. *Pages 238–266 of: Proceedings of the viii banff higher order workshop conference on logics for concurrency : structure versus automata*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Vardi, Moshe Y., & Wolper, P. (1986). An automata-theoretic approach to automatic program verification. *In proceedings of the first symposium on logic in computer science*.

Vardi, Moshe Y., & Wolper, Pierre. (1994). Reasoning about infinite computations. *Information and computation*, **115**, 1–37.

Wolter, F., & Zakharyaschev, M. (2000). Temporalizing description logics. *Pages 379–401 of: Frontiers of Combining Systems II*. LNCS, vol. 1794. Springer.

Wolter, Frank, & Zakharyaschev, Michael. (1998). Temporalizing description logics. *Pages 104–109 of: In proceedings of frocos'98*.