TECHNISCHE UNIVERSITÄT DRESDEN

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

MASTER THESIS

BRANCHING TEMPORAL DESCRIPTION LOGICS:
REASONING ABOUT $CTL_{\mathcal{ALC}}$ AND $CTL_{\mathcal{EL}}$ CONCEPTS

VICTOR GUTIERREZ-BASULTO

OVERSEEING PROFESSOR
PROF. DR. FRANZ BAADER
SUPERVISOR
PROF. DR. CARSTEN LUTZ

Dresden, Germany January, 2009

TECHNISCHE UNIVERSITÄT DRESDEN

Author:	**Víctor Didier Gutiérrez Basulto**
Matrikel-Nr:	**3361795**
Title:	**Branching Temporal Description Logics: Reasoning about $\mathbf{CTL}_{\mathcal{ALC}}$ and $\mathbf{CTL}_{\mathcal{EL}}$ concepts**
Degree:	**Master of Science**
Date of submision:	**19.01.09**

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.

Víctor Didier Gutiérrez Basulto

Abstract

In many applications of description logics (DLs) it is no longer enough to describe the *static* aspects of the application domain. In particular, there is a need to formalize the temporal evolution of an application domain. This is the case, for example, if we want to use DLs as conceptual modeling languages for temporal databases. Another example are medical ontologies, where the representation of concepts often requires reference to temporal patterns. However, description logics have been designed and used as a formalism for knowledge representation and reasoning only in *static* application domains. Therefore, DLs are not able to express temporal aspects of knowledge. The previous observations have resulted in diverse proposals of temporal description logics (TDLs). In particular, one approach is to combine standard description logics, such as $\mathcal{ALC}$ and $\mathcal{EL}$, with standard temporal logics, such as LTL, CTL and CTL*.

In this thesis, we follow the mentioned approach. More precisely, we use the description logics $\mathcal{ALC}$ and $\mathcal{EL}$ in the DL component and the temporal logic computation tree logic (CTL) in the temporal component. These combinations result in two TDLs, namely CTL$_{\mathcal{ALC}}$ from the combination of $\mathcal{ALC}$ and CTL, and CTL$_{\mathcal{EL}}$ from the combination of CTL and $\mathcal{EL}$. In CTL$_{\mathcal{ALC}}$ and CTL$_{\mathcal{EL}}$, we focus on temporal reasoning about concepts, i.e., we apply temporal operators only to concepts. After introducing CTL$_{\mathcal{ALC}}$ and CTL$_{\mathcal{EL}}$, we determine the computational complexity for reasoning problems in the mentioned logics. More precisely, we show that satisfiability w.r.t. TBoxes with expanding domains in CTL$_{\mathcal{ALC}}$ is EXPTIME-complete. We show also that subsumption w.r.t. TBoxes with expanding domains in CTL$_{\mathcal{EL}}$ is intractable, in particular, it is EXPTIME-complete.

Acknowledgments

First of all, I would like to thank my supervisor, Carsten Lutz for his infinite patience, his friendly guidance and his excellent comments and ideas. Without his supervision this thesis had not been successfully finished. Moreover, I would like to thank Prof. Franz Baader for introducing me to the fascinating topic of description logics. Finally, I want to thank Prof. Stephen Hoelldobler for his advice along my stay in the computational logic program.

Many thanks goes to my colleagues in the CL program. Specially, I appreciate all the trips we did together, the cooking and drinking time and all the hanging around. Special thanks to my Latinamerican, German, Pakistanis and Indian friends.

I appreciate the financial support given me by the Mexican National Council for Science and Technology and the German Service of Academic Exchange that made possible my studies in the CL programm.

I would like to thank my beloved wife Angelica for all her love, patience and scientific ideas. Nothing of this would be real without her. Last but not least, I thank my parents and brother for being who and how they are.

Contents

Chapter 1

Introduction

In this chapter, we give an introduction to the topics that we treat throughout this thesis. In the first section, we present the main ideas behind description logics. Then, in the second section, we introduce temporal description logics. Thereafter, in the third section, we introduce the temporal logic computation tree logic. Finally, in the fourth section, we set the objective and describe the structure of this work.

1.1 Description logics

Description logics (DLs) are a well known family of logic-based knowledge representation formalisms that allow to represent and reason about conceptual knowledge in a structured and well-understood way. An important characteristic of DLs is that they provide a formal way to construct a knowledge representation. We use description logics in knowledge-based systems that offer reasoning services. These reasoning services allow to effectively extract implicit consequences from the explicitly represented knowledge.

In the 1970's –prior to description logics– two approaches to represent knowledge arose, namely *semantic networks* (Quillian, 1968) and *frames* (Minsky, 1974). These formalisms, semantic networks and frames, use simple graphs and structured objects to represent knowledge, respectively. The intuition behind semantic networks and frames was that, by means of the mentioned structures, representation could be simpler than in powerful logic-based approaches and thus reasoning would be more efficient. The main deficit of semantic networks and frames was the lack of semantics, and as a result, the problem of ambiguities.

Description logics appeared as a sort of compromise between, on the one hand, the features of semantic networks and frames, and on the other, logic-based formalisms. The earliest DL system is KL-ONE (Brachman & Schmolze, 1985), introduced in 1985. Later on, Schimdt-Schauß showed that KL-ONE is undecidable (Schmidt-Schauß, 1989). In 1991, Schimdt-Schauß and Smolka introduced the decidable language $\mathcal{ALC}$ (Schmidt-Schauß& Smolka, 1991).

1.1.1 Syntax

The basic notions in description logics are *concept names* (unary predicates) and *roles* (binary relations). A specific DL is mainly characterized by the set of *constructors* it provides to build more complex concepts and roles out of atomic ones. Particular description logics have individual names, e.g., $\mathcal{ALC}$, $\mathcal{EL}$ and $\mathcal{ALCN}$.

In description logics, *concept descriptions* are the basis for expressing knowledge. To construct concept descriptions, we use concept names, roles and constructors. Concept names denote classes of objects in a certain domain, e.g., Mother, Human, Number, etc. Roles are binary relations between objects of the domain, e.g., has, loves.

The description logic $\mathcal{ALC}$ is the "smallest" DL that is propositionally closed, i.e., that provides for all Boolean connectives. More precisely, $\mathcal{ALC}$ provides the constructors: negation ($\neg$), conjunction ($\sqcup$), disjunction ($\sqcap$), and existential ($\exists$) and universal ($\forall$) restriction. The following is a concept description in $\mathcal{ALC}$.

$$\mathsf{Human} \sqcap \mathsf{Male} \sqcap \exists \mathsf{has\_child}.\top \tag{1.1}$$

Here, Human and Male are concept names, has_child is a role name and $\top, \sqcap$ are constructors. $\top$ is an abbreviation for some fixed propositional tautology such as $A \sqcup \neg A$. The concept description (1.1) defines the notion of "father".

1.1.2 Knowledge bases

Description logic *knowledge bases* usually consist of two components, namely a *TBox* and an *ABox*. A TBox stores terminological knowledge and background knowledge about an application domain. An ABox stores assertional knowledge about the individuals, i.e., knowledge about the state of affairs in a particular "world".

There are several kinds of TBoxes. The most common TBoxes are *acyclic TBoxes* and *general TBoxes*. Acyclic TBoxes consist of concept definitions of the form $A \doteq C$, defining the concept name A as a complex concept C. We call a TBox acyclic if and only if the definition of no concept refers directly or indirectly to itself and the left-hand sides of all concept definitions are pair-wise distinct. General TBoxes allow for *general concept inclusions (GCIs)*. A GCI is of the form $C \sqsubseteq D$, where C and D are (possibly) complex concepts, and states that C implies D. We use general TBoxes to formulate general constraints and acyclic TBoxes to define concepts, i.e., acyclic TBoxes assign concept names to complex concepts thus they define abbreviations.

The following $\mathcal{ALC}$-TBox defines the concepts of Parent, Mother and Father, and requires that every Human has only human children.

$$\begin{array}{rcl} \mathsf{Parent} & \doteq & \mathsf{Human} \sqcap \exists \mathsf{has\_child}.\top \\ \mathsf{Mother} & \doteq & \mathsf{Parent} \sqcap \mathsf{Female} \\ \mathsf{Father} & \doteq & \mathsf{Parent} \sqcap \mathsf{Male} \\ \mathsf{Human} & \sqsubseteq & \forall \mathsf{has\_child}.\mathsf{Human} \end{array}$$

Moreover, the following ABox states that yazmin is a female human with a human child carmen.

$$A := \{(\mathsf{Human} \sqcap \mathsf{Female})\ (\mathsf{yazmin}),\ \mathsf{has\_child}(\mathsf{yazmin},\mathsf{carmen}),\ \mathsf{Human}(\mathsf{carmen})\}.$$

Due to the objective of this thesis, in the sequel we do not consider ABoxes.

1.1.3 Semantics and inferences

The semantics of concept descriptions is given in terms of an *interpretation*. An interpretation consists of a non-empty set of individuals, the *interpretation domain*, and an *interpretation function*. The latter assigns concept names to sets of elements of the interpretation domain, and role names to a binary relation on the interpretation domain. The interpretation function is inductively extended to arbitrary concept descriptions. Thus, we interpret concept descriptions as subsets of the interpretation domain. The following interpretation gives the semantic of the concept description (1.1), where $\Delta^{\mathcal{I}}$ is the interpretation domain and $\cdot^{\mathcal{I}}$ is the interpretation function.

$\Delta^{\mathcal{I}} = \{\mathsf{PETER, JOHN, KATE, ROSE , HILLARY}\}$,
$\mathsf{Human}^{\mathcal{I}} = \Delta^{\mathcal{I}}$, $\mathsf{Male}^{\mathcal{I}} = \{\mathsf{PETER, JOHN}\}$,
$\mathsf{has\_child}^{\mathcal{I}} = \{(\mathsf{PETER,KATE}), (\mathsf{JOHN,HILLARY}), (\mathsf{JOHN,PETER})\}$.

We say that an interpretation $\mathcal{I}$ 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 $\mathcal{T}$ is interpreted identically, and the extension of C is contained in the extension of D for every GCI $C \sqsubseteq D$ in $\mathcal{T}$.

The standard reasoning problems in description logics are *satisfiability* and *subsumption*. A concept C is *satisfiable* if there exists an interpretation $\mathcal{I}$ such that $C^{\mathcal{I}} \neq \emptyset$. We say that $\mathcal{I}$ is a *model* of C. A concept D *subsumes* a concept C (written $C \sqsubseteq D$) if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all interpretations $\mathcal{I}$. Satisfiability and subsumption can take into account TBoxes. A concept C is *satisfiable w.r.t. a TBox $\mathcal{T}$* if and only if there exists a model of $\mathcal{T}$ such that C is interpreted as a non-empty set. C is *subsumed by a concept D w.r.t. $\mathcal{T}$* if and only if C is more specific than D in the sense that, w.r.t. every model of $\mathcal{T}$, the interpretation of C is a subset of that of D.

1.1.4 Complexity of inference problems

The computational complexity of satisfiability and subsumption depends on the expressivity of the underlying description logic.

$\mathcal{ALC}$-concept satisfiability is PSPACE-complete (Schmidt-Schauß& Smolka, 1991). The complexity increases if we take into account TBoxes. More precisely, $\mathcal{ALC}$-concept satisfiability w.r.t. general TBoxes is EXPTIME-complete (Schild, 1991; Schild, 1994). There are also DLs with polynomial inference problems, such as $\mathcal{EL}$. The description logic $\mathcal{EL}$ provides only existential quantification, conjunction, and the top concept. $\mathcal{EL}$-concept subsumption w.r.t. TBoxes is decidable in polynomial time (Brandt, 2004).

1.1.5 DLs and other logics

There exists a connection between description logics and various other logics. This connection can be used to transfer complexity and (un)decidability results between the DLs and other logics. In particular, there exists a close connection between modal logic (Blackburn *et al.*, 2006) and description logics.

Schild (Schild, 1991) observed that $\mathcal{ALC}$ is a syntactic variant of the multi-modal logic K_ω. Kripke structures and description logics interpretations can be translated into one another. Hence, DL concept satisfiability is modal logic formula satisfiability. Since formula satisfiability in K_ω is PSPACE-complete, we can obtain an alternative proof of the PSPACE-completeness of $\mathcal{ALC}$-concept satisfiability by using the correspondence between $\mathcal{ALC}$ and K_ω. Description logics is not only related with modal logics. In particular, there exists also a connection between description logics and decidable fragments of first order logics (Lutz *et al.*, 2001).

1.2 Temporal description logics

In many applications of description logics (Baader *et al.*, 2003) is necessary to describe the temporal aspect of the application domain. This is the case, when we use DLs to represent conceptual models of temporal databases (Artale *et al.*, 2002). Another example is the use of DLs as ontology languages or conceptual modeling languages, where the description of a concept may involve reference to temporal patterns. As an example, consider the concept Mortal. A faithful representation of Mortal should say that a mortal is a living being who is alive until he dies.

The expressiveness of pure description logics is not sufficient to describe temporal aspects of knowledge. Hence, DLs cannot describe concepts that refer to temporal patterns, such as Mortal. Due to this observation, a diverse literature on *temporal description logics (TDLs)* has emerged. Proposals for TDLs include the combination of description logics with Halpern and Shoham's logic of time intervals (Schmiedel, 1990), formalisms inspired by action logics (Artale & Franconi, 1998), the treatment of time points and intervals as a datatype (Lutz, 2004) and the combination of standard description logics with standard (propositional) temporal logics into logics with a two-dimensional semantics, where one dimension is for time and the other for the DL domain (Schild, 1993; Wolter & Zakharyaschev, 2000; Gabbay *et al.*, 2003). In 1993, Schild proposed the latter combinations (Schild, 1993), which since then have experienced constant development in the sense that the DL and the temporal component have varied. For more information about the proposals presented here, see the surveys (Artale & Franconi, 2000; Artale & Franconi, 2005; Lutz *et al.*, 2008).

In this thesis, we follow the last approach, where we use the description logics $\mathcal{ALC}$ and $\mathcal{EL}$ in the DL component and the temporal logic computation tree logic (CTL) (Clarke & Emerson, 1982) in the temporal component. After we fix the description logic and the temporal logic that we combine, there remain several degrees of freedom when we define the resulting temporal description logic. An important decision is whether to apply the temporal operators to DL concepts, roles, TBoxes or ABoxes. As we could expect, the resulting TDLs differ in many aspects, in particular, expressiveness and computational properties. In this thesis, we use temporal operators only as concept constructors.

1.3 A temporal logic: CTL

Temporal logic is a type of *modal logic* (Gabbay *et al.*, 1994). A temporal logic allows for the specification of the relative order of events. Some examples are "the car stops once the driver pushes the brake", or "the message is received after it has been sent". However, a temporal logic does not support any means to refer to the precise timing of events. One might thus say that the modalities in temporal logic are *time-abstract*. Due to these characteristics, temporal logics have been adopted as a powerful tool for specifying and verifying concurrent programs (Pnueli, 1977).

Temporal logics are often classified according to whether time is assumed to have a linear or a branching structure (Lamport, 1980). In linear temporal logics, each moment in time has a unique possible future while in branching temporal logics each moment in time may split time into several possible futures. In this thesis, we focus on *branching time temporal logics* (Gabbay *et al.*, 2000), in particular, *computation tree logic (CTL)* (Clarke & Emerson, 1982).

Computation tree logic provides temporal branching connectives that are composed of a *path quantifier* immediately followed by a single *linear temporal connective* (Emerson, 1990). The path quantifiers are A ("for all paths") and E ("for some path"). The linear time connectives are $\bigcirc$ ("next") and $\mathcal{U}$ ("until"). For example, the formula $\mathsf{E}(p\mathcal{U}\,q)$ says that there is a computation starting at the current time point along which p holds until q holds.

Computation tree logic enables us to make powerful assertions about the behavior of a program. For example, $\mathsf{E}(true\mathcal{U}\,q)$ says that there is a computation starting at the current time point along which q eventually holds. We abbreviate this by $\mathsf{E}\Diamond q$. Another example, consider the formula $\mathsf{A}(true\mathcal{U}\,\neg q)$. The last formula holds in a state s if along every path starting at s eventually $\neg q$ holds, abbreviated by ($\mathbf{A}\Diamond\neg q$). It may seem that all temporal connectives talk about finite computations. However, we can combine temporal and Boolean connectives to form assertions about infinite computations. For example, $\neg\mathbf{A}\Diamond\neg q$ says that there is an infinite path starting at the current time point along which q always holds, abbreviated by $\mathbf{E}\Box q$.

The semantics of computation tree logic is defined in terms of an infinite, directed *tree* of states or time instants. A *direct edge* from node s to node t means that it is possible to pass from s to t, or that t is a possible future from s. Each transversal of a tree starting in its root represents a single path. A tree itself thus represents all possible paths. A tree rooted at state s represents all possible infinite computations that start in s.

1.4 Objective and structure

As we have stated above, in many applications of description logics, being able to represent temporal aspects of the domain is quite useful. To allow DLs to express temporal aspects, we can choose among several approaches. In this thesis, we focus on the temporal extensions of description logics that emerge from the combination of standard DLs with standard temporal logics. In particular, we treat combinations that use computation tree logic (CTL) in the temporal component. We have decided to consider CTL due to two important reasons. First, as argued in (Lutz *et al.*, 2008), linear temporal logics, such as *linear temporal logic (LTL)*; see e.g., (Gabbay *et al.*, 1994), are not able to distinguish between possible, actual, and necessary future developments. Suppose, for example, that we want to describe countries that can join the EU in the future. The GCI

$$\mathsf{EU\_candidate} \sqsubseteq \Diamond\mathsf{EU\_member},$$

expresses that, sooner or later, every EU candidate will join the EU. However, the last statement seems too strong. What we actually mean is that, under certain circumstances, an EU candidate may join the EU in the future. CTL allows to formalize statements of the previous sort. If $\mathbf{E}$ is understood as "it is possible that" and $\mathbf{A}$ as "it is necessary that", then

$$\mathsf{EU\_candidate} \sqsubseteq \mathbf{E}\Diamond\mathsf{EU\_member}$$

means that each EU candidate has the possibility to join the EU.

The second reason to choose CTL is that, most of the research on combinations of DLs and temporal logics concentrates on the case where LTL is used in the temporal dimension (Schild, 1993; Artale *et al.*, 2007; Baader *et al.*, 2008; Lutz *et al.*, 2008). In particular, in branching time TDLs –in contrast to linear time TDLs– two problems remain open:

1. No tight complexity results are known.
2. Simpler reasoning problems than satisfiability of temporal $\mathrm{CTL}_{\mathcal{ALC}}$ TBoxes have not yet been investigated, where $\mathrm{CTL}_{\mathcal{ALC}}$ is the temporal description logic that emerges from the combination of $\mathcal{ALC}$ and CTL.

In this thesis, we introduce two temporal description logics, namely $\mathrm{CTL}_{\mathcal{ALC}}$ and $\mathrm{CTL}_{\mathcal{EL}}$. These TDLs, $\mathrm{CTL}_{\mathcal{ALC}}$ and $\mathrm{CTL}_{\mathcal{EL}}$, result from the combination of CTL with $\mathcal{ALC}$ and $\mathcal{EL}$, respectively. However, even after we fix the DL and the temporal logic to be combined, we have to make another decision, namely to which pieces of syntax we apply temporal operators. In this thesis, we treat temporal reasoning about concepts, i.e., we only apply temporal operators to concepts.

After setting the temporal description logics to investigate, we show that satisfiability of $\mathrm{CTL}_{\mathcal{ALC}}$ temporal concepts w.r.t. TBoxes with expanding domains is EXPTIME-complete. Since satisfiability in $\mathcal{ALC}$ with TBoxes is EXPTIME-complete and satisfiability in CTL is EXPTIME-complete as well, we can consider $\mathrm{CTL}_{\mathcal{ALC}}$ as rather well-behaved, i.e., concept satisfiability is not harder than in the component logics.

Thereafter, we prove that subsumption of temporal concepts w.r.t. TBoxes with expanding domains in $\mathrm{CTL}_{\mathcal{EL}}$ is EXPTIME-complete. Thus, reasoning in $\mathrm{CTL}_{\mathcal{EL}}$ does not remain tractable as in pure $\mathcal{EL}$.

Recall that CTL –in contrast to LTL– can distinguish between possible, actual, and necessary future developments. The results obtained in this thesis show that the last fact does not increase the computational complexity for reasoning problems in $\mathrm{CTL}_{\mathcal{ALC}}$ ($\mathrm{CTL}_{\mathcal{EL}}$) with respect to the computational complexity for those in $\mathrm{LTL}_{\mathcal{ALC}}$ ($\mathrm{LTL}_{\mathcal{EL}}$). More precisely, satisfiability of $\mathrm{LTL}_{\mathcal{ALC}}$ concepts w.r.t. TBoxes with expanding domains is EXPTIME-complete and subsumption of temporal concepts w.r.t. TBoxes with expanding domains in $\mathrm{LTL}_{\mathcal{EL}}$ is EXPTIME-complete.

Regarding branching temporal description logics, the results presented in this thesis are the first tight complexity results known.

We structure this thesis as follows:

- In Chapter 2, we introduce the theoretical background. We introduce the basics of description logics: their syntax, semantics and standard reasoning problems. Next, we present the formal preliminaries of the temporal logic computation tree logic.

- In Chapter 3, we introduce the temporal description logic $\mathrm{CTL}_{\mathcal{ALC}}$: its syntax and semantics. Next, we give a characterization of the semantics of $\mathrm{CTL}_{\mathcal{ALC}}$ in terms of *fusion models*. Moreover, this characterization allows to establish a connection between $\mathrm{CTL}_{\mathcal{ALC}}$ and the *standard μ-calculus*. Thereafter, using this connection, we determine the complexity of reasoning in $\mathrm{CTL}_{\mathcal{ALC}}$.

- In Chapter 4, we introduce the temporal description logic $\mathrm{CTL}_{\mathcal{EL}}$. Thereafter, we determine the complexity of reasoning in the mentioned logic. To prove the lower bound, we reduce satisfiability w.r.t. TBoxes in $\mathcal{ALC}$ to subsumption w.r.t. TBoxes in $\mathrm{CTL}_{\mathcal{EL}}$.

- In Chapter 5, we summarize the results of this thesis and briefly discuss future prospects.

Chapter 2

Preliminaries

In this chapter, we introduce the basic notions that we use throughout the following chapters. In the first two sections, we present the basics of the description logics $\mathcal{ALC}$ and $\mathcal{EL}$. In the last section, we give a formal definition of the temporal logic CTL.

2.1 Introducing $\mathcal{ALC}$

The description logic $\mathcal{ALC}$, was first introduced by Schmidt-Schauß and Smolka (Schmidt-Schauß& Smolka, 1991). The name $\mathcal{ALC}$ stands for *Attribute Language with Complements*. $\mathcal{ALC}$ is the "smallest" description logic that is propositionally closed, i.e., that provides for all Boolean connectives. More precisely, $\mathcal{ALC}$ concepts are built from the Boolean connectives and so-called existential and universal value restrictions.

Definition 2.1 ($\mathcal{ALC}$ syntax). Let N_C and N_R be disjoint sets of *concept names* and *role names*, respectively. The set of $\mathcal{ALC}$-*concept descriptions* is defined inductively as follows:

1. Each concept name $A \in N_C$ is an $\mathcal{ALC}$-concept description.
2. $\top$ and $\bot$ are $\mathcal{ALC}$-concept descriptions.
3. If C, D are $\mathcal{ALC}$-concept descriptions, and $r \in N_R$, then the following are also $\mathcal{ALC}$-concept descriptions:
   - $C \sqcap D$, $C \sqcup D$, $\neg C$,

- $\exists r.C$, $\forall r.C$.

Example 2.2.

$$\mathsf{Animal} \sqcap \exists\, \mathsf{eats}.\mathsf{Meat}$$

This concept description describes the class of carnivores.

We give the formal semantics of $\mathcal{ALC}$ by interpretations. We can see an interpretation as a mapping from concept descriptions to an specific domain. In particular, we interpret concepts as unary predicates over the domain and roles as binary relations over the domain.

Definition 2.3 ($\mathcal{ALC}$ semantics). An *interpretation* $\mathcal{I}$ consists of a non-empty interpretation domain $\Delta^{\mathcal{I}}$ and an interpretation function $\cdot^{\mathcal{I}}$ that

- assigns to each $A \in N_C$ a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$,
- assigns to each $r \in N_R$ a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

The interpretation function is then inductively extended to the rest of $\mathcal{ALC}$-concept descriptions as follows:

- $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$, $\bot^{\mathcal{I}} = \emptyset$,
- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$,
- $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$,
- $(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$,
- $(\exists r.C)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} \mid \text{there is an } e \in \Delta^{\mathcal{I}} \text{ with } (d,e) \in r^{\mathcal{I}} \text{ and } e \in C^{\mathcal{I}}\}$,
- $(\forall r.C)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} \mid \text{for all } e \in \Delta^{\mathcal{I}}, (d,e) \in r^{\mathcal{I}} \text{ implies } e \in C^{\mathcal{I}}\}$.

Next, we introduce a normal form for $\mathcal{ALC}$-concepts.

Definition 2.4 ($\mathcal{ALC}$ NNF). An $\mathcal{ALC}$-concept C is in *negation normal form (NNF)* if negation occurs only in front of concept names. Every $\mathcal{ALC}$-concept can be converted into an equivalent one in NNF by exhaustively applying the following rules:

$$\begin{array}{rclcrcl} \neg\neg C & \equiv & C & & & & \\ \neg(C \sqcap D) & \equiv & \neg C \sqcup \neg D & \quad & \neg(\exists r.C) & \equiv & \forall r.\neg C \\ \neg(C \sqcup D) & \equiv & \neg C \sqcap \neg D & & \neg(\forall r.C) & \equiv & \exists r.\neg C. \end{array}$$

In description logics, we use TBoxes to capture the background knowledge about the world. In this thesis, we use *general TBoxes* which introduce constraints of the form "for all the domain elements where C holds, D holds as well".

Definition 2.5 (GCI, general TBox). If C and D are $\mathcal{ALC}$-concept descriptions, then the expression $C \sqsubseteq D$ is called a *generalized concept inclusion axiom (GCI)*. A finite set $\mathcal{T}$ of GCIs is called a *general TBox or TBox*.

An interpretation $\mathcal{I}$ is a *model* of a general TBox $\mathcal{T}$ if for every GCI $C \sqsubseteq D \in \mathcal{T}$, it holds that $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

Various reasoning problems are considered for DLs. For the purpose of this thesis, it is sufficient to introduce only two of them: concept satisfiability and concept subsumption.

Definition 2.6 (DL reasoning problems). Let C be an $\mathcal{ALC}$-concept description, and $\mathcal{T}$ a general TBox. Then,

- *C is satisfiable w.r.t. $\mathcal{T}$* if there is a model $\mathcal{I}$ of $\mathcal{T}$ such that $C^{\mathcal{I}} \neq \emptyset$.
- *C is subsumed by D w.r.t. $\mathcal{T}$* (written $C \sqsubseteq_{\mathcal{T}} D$) if and only if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{T}$.

Note that, in a description logic providing the Boolean connectives, subsumption can be reduced to (un)satisfiability since $C \sqsubseteq D$ iff $C \sqcap \neg D$ is unsatisfiable. The converse also holds since C is unsatisfiable iff C is subsumed by $\bot$. The previous observations imply that, when establishing lower and upper complexity bounds we may restrict ourselves to satisfiability since all the obtained results are also valid for subsumption.

The next definitions introduce a special kind of model.

Definition 2.7. Let C be a concept and $\mathcal{T}$ a TBox. C has a *tree-model w.r.t. $\mathcal{T}$* if the following holds, if C is satisfiable w.r.t. $\mathcal{T}$, then C is satisfiable in a model of $\mathcal{T}$ that is tree-shaped and whose root belongs to C.

Definition 2.8. A description logic $\mathcal{L}$ has the *tree-model property* iff every concept that is satisfiable w.r.t. a TBox has a tree-model w.r.t. this TBox.

Proposition 2.9. *$\mathcal{ALC}$ has the tree-model property.*

2.2 Introducing $\mathcal{EL}$

The description logic $\mathcal{EL}$ is one of the most basic description logics. The name $\mathcal{EL}$ stands for *Existential Language*. $\mathcal{EL}$ is less expressive than $\mathcal{ALC}$. More precisely, $\mathcal{EL}$ allows only for existential quantification ($\exists$), conjunction ($\sqcap$), and the top concept ($\top$).

Definition 2.10 ($\mathcal{EL}$ syntax). Let N_C and N_R be disjoint sets of concept names and role names, respectively. The set of $\mathcal{EL}$-*concept descriptions* is defined inductively as follows:

1. Each concept name $A \in N_C$ is an $\mathcal{EL}$-concept description.
2. $\top$ is an $\mathcal{EL}$-concept description.
3. If C, D are $\mathcal{EL}$-concept descriptions and $r \in N_R$, then $C \sqcap D$ and $\exists r.C$ are also $\mathcal{EL}$-concept descriptions.

We define the semantics of $\mathcal{EL}$-concept descriptions as we did for $\mathcal{ALC}$. Since $\mathcal{EL}$ does not allow for negation, the satisfiability problem is not interesting (every concept term is satisfiable). However, subsumption is not trivial. A special property of $\mathcal{EL}$ is that checking subsumption w.r.t. TBoxes can be done in polynomial time (Brandt, 2004).

2.3 Introducing CTL

The branching temporal logic CTL, was first introduced by Clarke and Emerson (Clarke & Emerson, 1982). The name CTL stands for *Computation Tree Logic*. CTL is based on propositional logic with a discrete notion of time, and only future modalities. CTL is sufficiently expressive to formulate an important set of so-called *system properties*.

Definition 2.11 (CTL syntax). Let PL be the set of atomic propositions. The class of *computation tree logic formulas* is the smallest set such that

- each propositional letter $p \in PL$ is a formula;
- if ϕ and ψ are formulas, then $\neg\phi$, $\phi \vee \psi$ and $\phi \wedge \psi$ are formulas;
- if ϕ and ψ are formulas, then $\mathbf{A}\bigcirc\phi$, $\mathbf{E}\bigcirc\phi$, $\mathbf{A}(\phi\,\mathcal{U}\,\psi)$ and $\mathbf{E}(\phi\,\mathcal{U}\,\psi)$ are formulas.

The symbols A and E are called *path quantifiers*. Apart from Boolean abbreviations, we use

$$\begin{array}{rcl} \mathsf{A}\Diamond\psi & \text{for} & \mathsf{A}(\top\,\mathcal{U}\,\psi), \\ \mathsf{E}\Diamond\psi & \text{for} & \mathsf{E}(\top\,\mathcal{U}\,\psi), \\ \mathsf{A}\Box\psi & \text{for} & \neg\mathsf{E}\Diamond\neg\psi, \\ \mathsf{E}\Box\psi & \text{for} & \neg\mathsf{A}\Diamond\neg\psi. \end{array}$$

Example 2.12. CTL formulae of the form

$$\mathsf{A}\Box\mathsf{A}\Diamond\psi$$

expresses that ψ is infinitely true in all paths. The CTL formula

$$(\mathsf{A}\Box\mathsf{A}\Diamond\mathsf{crit}_1) \wedge (\mathsf{A}\Box\mathsf{A}\Diamond\mathsf{crit}_2)$$

thus requires each process to have access to the critical section infinitely often. In case of a traffic light, the *safety property* “each red light phase is preceded by a yellow light phase” can be formulated in CTL by

$$\mathsf{A}\Box(\mathsf{yellow} \vee \mathsf{A}\bigcirc\neg\mathsf{red}),$$

intuitively a safety property asserts that “nothing bad happens”. Finally, the *liveness property* “the traffic light is infinitely often green” can be formulated as

$$\mathsf{A}\Box\mathsf{A}\Diamond\mathsf{green},$$

intuitively a liveness property asserts that “something good will happen”.

We define the semantics of CTL with respect to a Kripke structure.

Definition 2.13 (Kripke structure). A *Kripke structure* $\mathcal{M}$ is a triple $\langle S, R, L\rangle$, such that

- S is a set of states,
- $R \subseteq S \times S$ is a total relation, i.e., for all states $s \in S$ there exists a state $s' \in S$ such that $(s, s') \in R$, and
- $L : S \to 2^{LP}$ is a function that labels each state with the set of atomic propositions true in that state.

A *path in* $\mathcal{M}$ is an infinite sequence of states, $\pi = s_0, s_1 \ldots$ such that for every $i \geq 0$, $(s_i, s_{i+1}) \in R$.

Definition 2.14 (CTL semantics). Let $\mathcal{M} = \langle S, R, L \rangle$ be a Kripke structure. We define *satisfaction* of CTL formulas in $\mathcal{M}$ at state $s \in S$ as follows:

$\mathcal{M}, s \models p$	iff	$p \in L(s)$;
$\mathcal{M}, s \models \neg\phi$	iff	$\mathcal{M}, s \not\models \phi$;
$\mathcal{M}, s \models \phi \wedge \psi$	iff	$\mathcal{M}, s \models \phi$ and $\mathcal{M}, s \models \psi$;
$\mathcal{M}, s \models \mathbf{E}\bigcirc\phi$	iff	$\mathcal{M}, t \models \phi$ for some $t \in S$ with $(s, t) \in R$;
$\mathcal{M}, s \models \mathbf{A}\bigcirc\phi$	iff	$\mathcal{M}, t \models \phi$ for all $t \in S$ with $(s, t) \in R$;
$\mathcal{M}, s \models \mathbf{E}(\phi\,\mathcal{U}\,\psi)$	iff	there exists a path $s_0, s_1, \ldots$ in $\mathcal{M}$ with $s_0 = s$ such that there is an $m \geq 0$ with $\mathcal{M}, s_m \models \psi$ and $\mathcal{M}, s_k \models \phi$ for all $k < m$;
$\mathcal{M}, s \models \mathbf{A}(\phi\,\mathcal{U}\,\psi)$	iff	for all paths $s_0, s_1, \ldots$ in $\mathcal{M}$ with $s_0 = s$, there is an $m \geq 0$ such that $\mathcal{M}, s_m \models \psi$ and $\mathcal{M}, s_k \models \phi$ for all $k < m$.

Example 2.15. In the Figure 2.1, we present a visualization of the semantics of the formulas: (1) $\mathbf{A}\Diamond\,\mathsf{Black}$, (2) $\mathbf{E}\Box\,\mathsf{Black}$ and (3) $\mathbf{E}(\mathsf{Gray}\,\mathcal{U}\,\mathsf{Black})$.

Any CTL formula can be transformed into a canonical form, the so called *negation normal form (NNF)*. In order to transform any CTL formula into NNF, for each operator a dual operator needs to be incorporated into the syntax of NNF formulae. To this aim, we introduce the operator $\mathcal{R}$ (called *release*) as the dual of $\mathcal{U}$. We define the release operator as follows:

$$\begin{aligned} \mathbf{E}(\phi\,\mathcal{R}\,\psi) &\equiv \neg\mathbf{A}(\neg\phi\,\mathcal{U}\neg\psi) \\ \mathbf{A}(\phi\,\mathcal{R}\,\psi) &\equiv \neg\mathbf{E}(\neg\phi\,\mathcal{U}\neg\psi). \end{aligned}$$

Definition 2.16 (CTL NNF). A CTL formula is in *negation normal form (NNF)* if negation occurs only in front of atomic propositions. Every CTL formula can be transformed into an equivalent one in NNF by exhaustively applying the following rules:

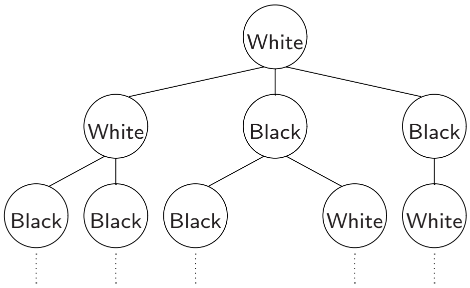
$$\begin{aligned} \neg\neg\phi &\equiv \phi & \neg\mathbf{A}\bigcirc\phi &\equiv \mathbf{E}\bigcirc\neg\phi \\ \neg(\phi \wedge \psi) &\equiv \neg\phi \vee \neg\psi & \neg\mathbf{E}(\phi\,\mathcal{U}\,\psi) &\equiv \mathbf{A}(\neg\phi\,\mathcal{R}\neg\psi) \\ \neg\mathbf{E}\bigcirc\phi &\equiv \mathbf{A}\bigcirc\neg\phi & \neg\mathbf{A}(\phi\,\mathcal{U}\,\psi) &\equiv \mathbf{E}(\neg\phi\,\mathcal{R}\neg\psi). \end{aligned}$$

We present a definition that is useful in the next chapter.

Definition 2.17. A *tree* is a pair $T = (S, R)$ consisting of a set S of *states* and a total relation $R \subseteq S \times S$ such that

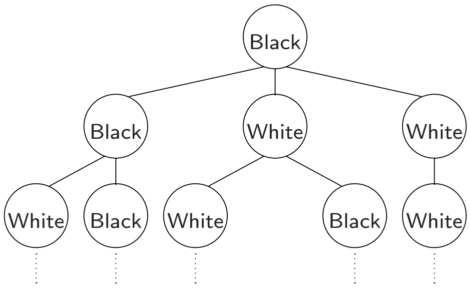
- there is a state s_0 with $\{s \in S \mid (s, s_0) \in R\} = \emptyset$,
- for every $s \in S \setminus \{s_0\}$ there is exactly one $s' \in S$ with $(s', s) \in R$ and, $s_0\, R^* s$ with R^* the transitive closure of R on S.

We call s_0 the *root*.



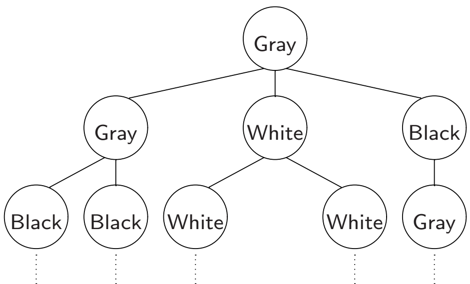
(1)

$\mathbf{A}\Diamond$ Black



(2)

$\mathbf{E}\Box$ Black



(3)

$\mathbf{E}(\mathsf{Gray}\,\mathcal{U}\,\mathsf{Black})$

Figure 2.1: Visualization of the semantics of some CTL formulas

Chapter 3

Temporal concepts: $\text{CTL}_{\mathcal{ALC}}$ concepts

In this chapter, we investigate the temporal description logic $\text{CTL}_{\mathcal{ALC}}$. First, we introduce its syntax and semantics. Thereafter, we give an alternative semantics of $\text{CTL}_{\mathcal{ALC}}$. Finally, we establish a relation between the μ-calculus and $\text{CTL}_{\mathcal{ALC}}$. Moreover, this relation allows to determine the complexity of reasoning in $\text{CTL}_{\mathcal{ALC}}$.

3.1 Introducing $\text{CTL}_{\mathcal{ALC}}$

During the last 15 years, various approaches to temporal reasoning with description logics have been proposed (Artale & Franconi, 2000; Artale & Franconi, 2005; Lutz *et al.*, 2008). An important one is the combination of description logics with standard temporal logics, which has first been suggested in (Schild, 1993). In this section, we introduce the temporal description logic $\text{CTL}_{\mathcal{ALC}}$. We construct $\text{CTL}_{\mathcal{ALC}}$ using the previous approach.

The temporal description logic $\text{CTL}_{\mathcal{ALC}}$ emerges from the combination of the description logic $\mathcal{ALC}$ and the temporal logic computation tree logic (CTL). Besides choosing the DL and the temporal logic to combine, some other design decisions have to be made. In particular, we have to decide which pieces of syntax temporal operators can be applied to. In this work, our interest focuses on the temporal evolution of concepts. Thus, we define the temporal description logic $\text{CTL}_{\mathcal{ALC}}$ whose concepts are formed using the concept constructors of $\mathcal{ALC}$ (as in Section 2.1) enriched with the CTL temporal operators.

3.1.1 CTL$_{\mathcal{ALC}}$ syntax

Definition 3.1 (CTL$_{\mathcal{ALC}}$ syntax). Let N_C and N_R be disjoint sets of *concept names* and *role names*, respectively. The set of CTL$_{\mathcal{ALC}}$-*concept descriptions* is defined inductively as follows:

1. Each concept name $A \in N_C$ is a CTL$_{\mathcal{ALC}}$-concept description.
2. $\top$ and $\bot$ are CTL$_{\mathcal{ALC}}$-concept descriptions.
3. If C, D are CTL$_{\mathcal{ALC}}$-concept descriptions, and $r \in N_R$, then the following are also CTL$_{\mathcal{ALC}}$-concept descriptions:
   - $C \sqcap D$, $C \sqcup D$, $\neg C$,
   - $\exists r.C$, $\forall r.C$,
   - $\mathsf{A}\bigcirc C$, $\mathsf{E}\bigcirc C$, $\mathsf{A}(C\mathcal{U}D)$, $\mathsf{E}(C\mathcal{U}D)$.

Example 3.2.

The following concept describes processes which necessarily have subprocesses that start in the next time.

$$\mathsf{Process} \sqcap \mathsf{A}\bigcirc(\exists\, \mathsf{starting.SubProcess})$$

The next concept describes processes that necessarily have access to their critical section infinitely often.

$$\mathsf{Process} \sqcap \mathsf{A}\Box\mathsf{A}\Diamond(\exists\ \mathsf{has\_access.CriticalSection})$$

We define TBoxes in the same way as in the case of $\mathcal{ALC}$, but now using CTL$_{\mathcal{ALC}}$ concepts instead of $\mathcal{ALC}$ concepts.

Example 3.3. The following GCI says that the property that at each moment of time each red light phase is necessarily preceded by a yellow light phase implies a safety property of a traffic light system.

$$\mathsf{A}\Box(\mathsf{YellowPhase} \sqcup \mathsf{A}\bigcirc\neg\mathsf{RedPhase}) \sqsubseteq \mathsf{SafetyProperty}.$$

The next GCI states that, each single man has the possibility at some point in the future to have a woman until he dies.

$$\mathsf{Male} \sqcap \mathsf{Single} \sqsubseteq \mathsf{E}\Diamond\mathsf{A}(\exists \mathsf{has.Female}\,\mathcal{U}\,\neg\mathsf{LivingBeing}).$$

3.1.2 CTL$_{\mathcal{ALC}}$ semantics

We interpret CTL$_{\mathcal{ALC}}$ in models based on a tree in which every state s comes equipped with an $\mathcal{ALC}$-model describing the domain at state s. In particular, we focus on temporal interpretations with *expanding domains*, i.e., it is assumed that the domain of the $\mathcal{ALC}$-model at state s is included in all states following s. In other words, objects can be created over time, but not destroyed.

Definition 3.4 (CTL$_{\mathcal{ALC}}$ semantics).

A *temporal interpretation* $\mathfrak{J} = (S, <, I)$ consists of a tree $T = (S, <)$ and a function I associating with each $s \in S$ an $\mathcal{ALC}$-interpretation

$$I(s) = (\Delta^{I(s)}, \cdot^{I(s)})$$

such that

* for all $s, s' \in S$, $s < s'$ implies $\Delta^{I(s)} \subseteq \Delta^{I(s')}$,
* for every individual name $a \in N_I$, $a^{I(s)} = a^{I(s')}$ for any $s, s' \in S$.

The temporal interpretation of

- a concept name $A \in N_C$ is the set $A^{\mathfrak{J}} = \{(s, d) \mid s \in S \text{ and } d \in A^{I(s)}\}$,
- a role name $r \in N_R$ is the set $r^{\mathfrak{J}} = \{(s, d_1, d_2) \mid s \in S \text{ and } (d_1, d_2) \in r^{I(s)}\}$.

The temporal interpretation is then inductively extended to the rest of CTL$_{\mathcal{ALC}}$ concept descriptions as follows, we use the same clauses as in Section 2.1 for the Booleans, plus the following ones:

- $(\forall r.C)^{\mathfrak{J}} = \{(s, d) \mid (s, d, d_1) \in r^{\mathfrak{J}} \text{ implies } (s, d_1) \in C^{\mathfrak{J}}\}$,
- $(\exists r.C)^{\mathfrak{J}} = \{(s, d) \mid \exists (s, d_1) \in C^{\mathfrak{J}} \text{ with } (s, d, d_1) \in r^{\mathfrak{J}}\}$,
- $(\mathbf{E}\bigcirc C)^{\mathfrak{J}} = \{(s, d) \mid \exists s_1 \in S \text{ with } s < s_1 \text{ and } (s_1, d) \in C^{\mathfrak{J}}\}$,
- $(\mathbf{A}\bigcirc C)^{\mathfrak{J}} = \{(s, d) \mid \forall s_1 \in S,\ s < s_1 \text{ implies } (s_1, d) \in C^{\mathfrak{J}}\}$,

- $(\mathbf{E}(C\mathcal{U}D))^{\mathfrak{J}} = \{(s,d) \mid \exists s_0 < s_1 < s_2 \ldots$ with $s = s_0$ such that there is an $m \geq 0$ with $(s_m, d) \in D^{\mathfrak{J}}$ and $(s_k, d) \in C^{\mathfrak{J}}$ for all $k < m\}$,

- $(\mathbf{A}(C\mathcal{U}D))^{\mathfrak{J}} = \{(s,d) \mid \forall s_0 < s_1 < s_2 \ldots$ with $s = s_0$ implies there is an $m \geq 0$ with $(s_m, d) \in D^{\mathfrak{J}}$ and $(s_k, d) \in C^{\mathfrak{J}}$ for all $k < m\}$,

- $(\mathbf{E}(C\mathcal{R}D))^{\mathfrak{J}} = \{(s,d) \mid \exists s_0 < s_1 < s_2 \ldots$ with $s = s_0$ such that for all $j \geq 0$: if $(s_k, d) \notin C^{\mathfrak{J}}$ for all $k < j$ then $(s_j, d) \in D^{\mathfrak{J}}\}$,

- $(\mathbf{A}(C\mathcal{R}D))^{\mathfrak{J}} = \{(s,d) \mid \forall s_0 < s_1 < s_2 \ldots$ with $s = s_0$ implies for all $j \geq 0$: if $(s_k, d) \notin C^{\mathfrak{J}}$ for all $k < j$ then $(s_j, d) \in D^{\mathfrak{J}}\}$.

As in the case of CTL, we introduce the release operator as the dual of the until operator, see Section 2.3.

A temporal interpretation $\mathfrak{J} = (S, <, I)$ is a *model* of a concept C if C is satisfied in the root of $(S, <)$, i.e., $(s_0, d) \in C^{\mathfrak{J}}$ such that s_0 is the root of $(S, <)$. It is a *model* of a TBox $\mathcal{T}$ if and only if $C^{\mathfrak{J}} \subseteq D^{\mathfrak{J}}$ for all $C \sqsubseteq D$. Thus, the GCIs are regarded as temporally *global* constraints in the sense that they should hold at every state. A CTL$_{\mathcal{ALC}}$ concept C is *satisfiable* with respect to a TBox $\mathcal{T}$ if there is a common model of C and $\mathcal{T}$.

Recall that in description logics providing all the Boolean connectives, there are mutual polynomial-time reductions between satisfiability and subsumption, see Section 2.1. Therefore, we restrict ourselves to the satisfiability problem in CTL$_{\mathcal{ALC}}$.

An important remark regarding expanding domains and constant domains, i.e., it is assumed that the domain of the $\mathcal{ALC}$-model is the same at every state, is that they give rise to different versions of concept satisfiability. For example, the following TBox has a (temporal) model with expanding domains, but no model with constant domains:

$$\top \sqsubseteq \mathbf{A}\bigcirc(A \sqcap \exists r.\neg A). \qquad (3.1)$$

We can transform any CTL$_{\mathcal{ALC}}$ formula into a canonical form, the so-called *negation normal form*.

Definition 3.5 (CTL$_{\mathcal{ALC}}$ NNF). A CTL$_{\mathcal{ALC}}$ concept C is in *negation normal form (NNF)* if negation occurs only in front of concept names. Given a CTL$_{\mathcal{ALC}}$ concept, we can transform this concept into an equivalent CTL$_{\mathcal{ALC}}$ concept in NNF. To this aim, we can use the standard rules for the Booleans, see Section 2.1, plus the following rules:

$$\begin{array}{rclcrcl} \neg\mathsf{E}\bigcirc C & \equiv & \mathsf{A}\bigcirc\neg C & \quad & \neg\mathsf{A}(C\,\mathcal{U}\,D) & \equiv & \mathsf{E}(\neg C\,\mathcal{R}\neg D) \\ \neg\mathsf{A}\bigcirc C & \equiv & \mathsf{E}\bigcirc\neg C & & \neg\mathsf{E}(C\,\mathcal{R}\,D) & \equiv & \mathsf{A}(\neg C\,\mathcal{U}\neg D) \\ \neg\mathsf{E}(C\,\mathcal{U}\,D) & \equiv & \mathsf{A}(\neg C\,\mathcal{R}\neg D) & & \neg\mathsf{A}(C\,\mathcal{R}\,D) & \equiv & \mathsf{E}(\neg C\,\mathcal{U}\neg D). \end{array}$$

3.2 CTL$_{\mathcal{ALC}}$ fusion semantics

An important observation regarding CTL$_{\mathcal{ALC}}$ with expanding domains is that, CTL$_{\mathcal{ALC}}$ is closely connected to the *fusion* of CTL and $\mathcal{ALC}$. A fusion is a general combination method for modal logics (Baader *et al.*, 2002; Gabbay *et al.*, 1994). To make the previous connection explicit, we introduce an alternative semantics for CTL$_{\mathcal{ALC}}$.

Definition 3.6 (CTL$_{\mathcal{ALC}}$ fusion semantics). Let $\mathsf{succ} \notin N_R$ be a special role name. A (non temporal) interpretation $\mathcal{I}$ is a *fusion interpretation* if and only if $\mathsf{succ}^{\mathcal{I}}$ is a total role, i.e., for each $d \in \Delta^{\mathcal{I}}$, there is a $d' \in \Delta^{\mathcal{I}}$ such that $(d, d') \in \mathsf{succ}^{\mathcal{I}}$. To interpret a CTL$_{\mathcal{ALC}}$ concept in $\mathcal{I}$, we use $\mathcal{ALC}$ clauses together with the following clauses

- $(\mathsf{E}\bigcirc C)^{\mathcal{I}} = \{ d \in \Delta^{\mathcal{I}} \mid \exists\, d_1 \in \Delta^{\mathcal{I}}$ with $(d, d_1) \in \mathsf{succ}^{\mathcal{I}}$ and $d_1 \in C^{\mathcal{I}}\}$,
- $(\mathsf{A}\bigcirc C)^{\mathcal{I}} = \{ d \in \Delta^{\mathcal{I}} \mid \forall\, d_1 \in \Delta^{\mathcal{I}}$, $(d, d_1) \in \mathsf{succ}^{\mathcal{I}}$ implies $d_1 \in C^{\mathcal{I}}\}$,
- $(\mathsf{E}(C\mathcal{U}D))^{\mathcal{I}} = \{ d \in \Delta^{\mathcal{I}} \mid \exists\, d_0, d_1, d_2, \ldots \in \Delta^{\mathcal{I}}$ with $(d_i, d_{i+1}) \in \mathsf{succ}^{\mathcal{I}}$ for all $i \geq 0$ and $d = d_0$ such that there is an $m \geq 0$ with $d_m \in D^{\mathcal{I}}$ and $d_k \in C^{\mathcal{I}}$ for all $k < m\}$,
- $(\mathsf{A}(C\mathcal{U}D))^{\mathcal{I}} = \{ d \in \Delta^{\mathcal{I}} \mid \forall\, d_0, d_1, d_2, \ldots \in \Delta^{\mathcal{I}}$, $(d_i, d_{i+1}) \in \mathsf{succ}^{\mathcal{I}}$ for all $i \geq 0$ and $d = d_0$ implies there is an $m \geq 0$ with $d_m \in D^{\mathcal{I}}$ and $d_k \in C^{\mathcal{I}}$ for all $k < m\}$,

- $(\mathsf{E}(C\mathcal{R}D))^{\mathcal{I}} = \{ d \in \Delta^{\mathcal{I}} \mid \exists\, d_0, d_1, d_2, \ldots \in \Delta^{\mathcal{I}}$ with $(d_i, d_{i+1}) \in \mathsf{succ}^{\mathcal{I}}$ for all $i \geq 0$ and $d = d_0$ such that for all $j \geq 0$: if $d_k \notin C^{\mathcal{I}}$ for all $k < j$ then $d_j \in D^{\mathcal{I}} \}$,

- $(\mathsf{A}(C\mathcal{R}D))^{\mathcal{I}} = \{ d \in \Delta^{\mathcal{I}} \mid \forall\, d_0, d_1, d_2, \ldots \in \Delta^{\mathcal{I}}$, $(d_i, d_{i+1}) \in \mathsf{succ}^{\mathcal{I}}$ for all $i \geq 0$ and $d = d_0$ implies for all $j \geq 0$: if $d_k \notin C^{\mathcal{I}}$ for all $k < j$ then $d_j \in D^{\mathcal{I}} \}$.

Recall that, $\mathcal{ALC}$ has the tree-model property, see Section 2.1. Thus, we can assume w.l.o.g. that a fusion model $\mathcal{I}$ is tree-shaped.

3.3 Relating CTL$_{\mathcal{ALC}}$ temporal and fusion semantics

In this section, we establish a connection between CTL$_{\mathcal{ALC}}$ temporal and fusion semantics. More precisely, we construct a temporal model of a concept and TBox given a fusion model of this concept and TBox, and vice versa. First, we focus on the former problem.

Recall that we consider temporal interpretations with expanding domains, i.e., we can create elements over the time, but do not destroy them. The constructed temporal model has to conform with the latter notion. However, a fusion model neither uses the notion of expanding domains nor codifies it in a natural way. In order to solve this problem, we have to identify the elements that should be part of the domain.

We differentiate in a fusion model between two types of elements, namely *DL elements* and *temporal elements*. The former, are part of the domain of the temporal model to be constructed. We use the temporal elements to define the states of the temporal model.

Definition 3.7 (Temporal & DL element). Let $\mathcal{I}$ be a tree-shaped fusion model. $e \in \Delta^{\mathcal{I}}$ is a *temporal element* if $(d, e) \in \mathsf{succ}^{\mathcal{I}}$ for some $d \in \Delta^{\mathcal{I}}$ or e is the root node. We call e a *DL element* if $(d, e) \in r^{\mathcal{I}}$ for some $d \in \Delta^{\mathcal{I}}$ and $r \in N_R$ or e is the root node.

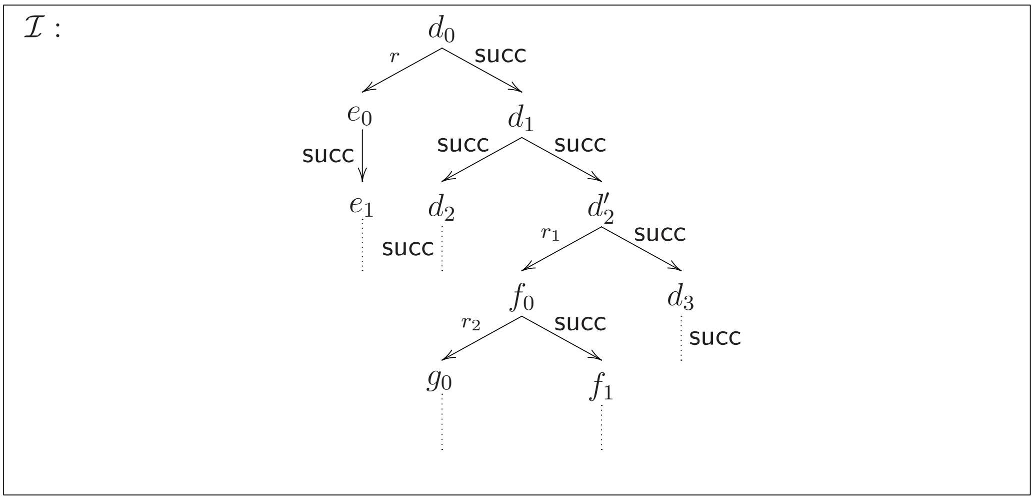


Figure 3.1: An example of a tree-shaped fusion model

Note that, the root node is a temporal and a DL element.

Example 3.8. In the tree-shaped fusion model of the Figure 3.1, the elements d_0, e_0, f_0, g_0 are DL elements and $d_0, d_1, d_2, d'_2, d_3, d_4, e_1, f_1$ are temporal elements.

The next definition allows to relate a temporal element with a DL element. We can see a temporal element d as a possible temporal evolution of the DL element $\mu_b(d)$.

Definition 3.9 (Function μ_b). Let $\mathcal{I}$ be a tree-shaped fusion model. We define $\mu_b : \Delta^{\mathcal{I}} \rightarrow \Delta^{\mathcal{I}}$ as follows

$$\mu_b(d) = e \quad \text{if there are } d_0, \ldots, d_n \in \Delta^{\mathcal{I}}, n \geq 0 \text{ such that } (d_i, d_{i+1}) \in \mathsf{succ}^{\mathcal{I}} \text{ for all } i < n \text{ with } d_0 \text{ the DL element } e \text{ and } d_n = d.$$

Example 3.10. In the fusion model of the Figure 3.1, $\mu_b(d_i) = d$ for all $0 \leq i \leq 4$, $\mu_b(f_j) = f$ for all $0 \leq j \leq 1$, $\mu_b(e_k) = e$ for all $0 \leq k \leq 1$, and $\mu_b(g_0) = g_0$.

The following definition allows to decide if an element is at a certain "succ-distance" from another.

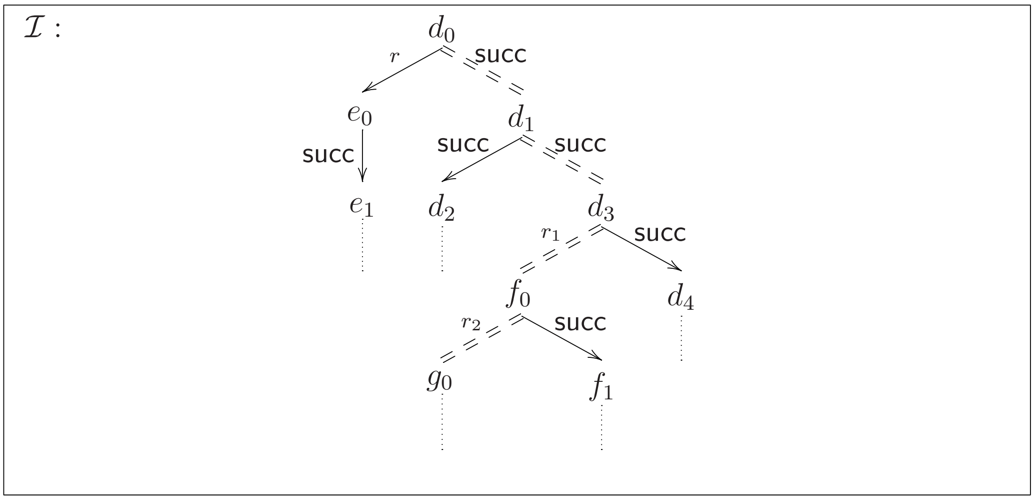


Figure 3.2: A 2-succ path from d_0 to g_0

Definition 3.11 (n-succ path).

Let $\mathcal{I}$ be a tree-shaped fusion model and $n \in \mathbb{N}$. We say that $e_0, \ldots, e_k \in \Delta^{\mathcal{I}}$, $k \geq 0$ is an *n-succ path from e_0 to e_k* if there are $r_0, \ldots, r_{k-1} \in N_R \cup \{\mathsf{succ}\}$ such that $(e_i, e_{i+1}) \in r_i^{\mathcal{I}}$ for all $i < k$ and $|\{i \mid i < k \text{ and } r_i = \mathsf{succ}\}| = n$.

Then, in an n-succ path from e_0 to e_k, we can think of n as the number of time-steps needed to travel from e_0 to e_k. Figure 3.2 shows a 2-succ path.

We define a special role name $\mathsf{ch} \notin N_R$ with $\mathsf{ch}^{\mathcal{I}}$ a total function, $\mathsf{ch}^{\mathcal{I}}(d) = e$ such that $(d, e) \in \mathsf{succ}^{\mathcal{I}}$. We should remark that ch serves as a choice function and we can choose it arbitrarily.

Definition 3.12 (n-ch path).

Let $\mathcal{I}$ be a tree-shaped fusion model and $n \in \mathbb{N}$. We say that $e_0, \ldots, e_k \in \Delta^{\mathcal{I}}, k \geq 0$ is an *n-ch path from e_0 to e_k* if there are $r_0, \ldots, r_{k-1} \in N_R \cup \{\mathsf{ch}\}$ such that $(e_i, e_{i+1}) \in r_i^{\mathcal{I}}$ or $r_i^{\mathcal{I}}(e_i) = e_{i+1}$ for all $i < k$ and $|\{i \mid i < k \text{ and } r_i = \mathsf{ch}\}| = n$.

Note that, we define an n-ch path analogously as we do for an n-succ path, but we replace succ with ch.

Definition 3.13 (Relevant).

Let $\mathcal{I}$ be a tree-shaped fusion model. We say that $d \in \Delta^{\mathcal{I}}$ is *relevant* to a temporal element $e \in \Delta^{\mathcal{I}}$, if there is a $d' \in \Delta^{\mathcal{I}}$ such that there is an n-succ

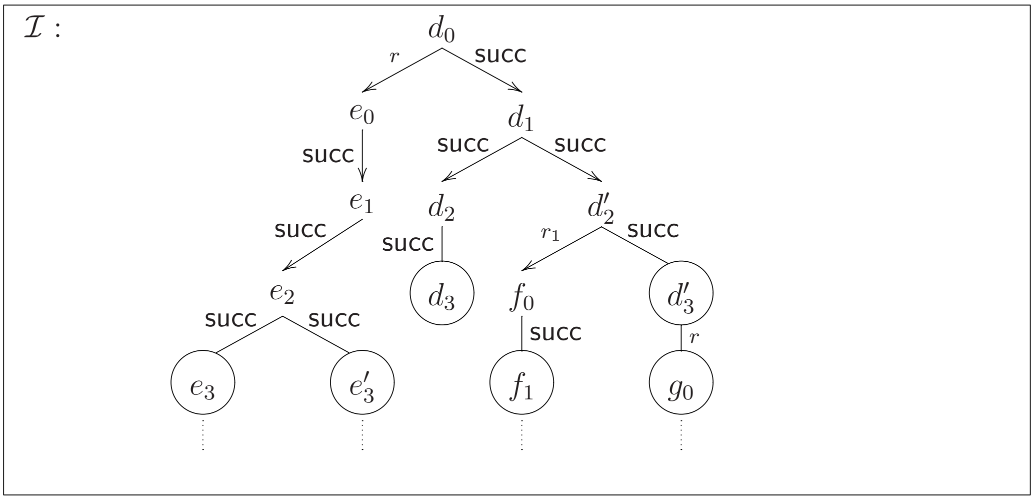


Figure 3.3: n succ depth

path $e_0, \ldots, e_k$ from d' to e, an n-ch path $e'_0, \ldots, e'_{k'}$ from d' to d for some $n \geq 0$, and for each $0 < i \leq k$, $0 < j \leq k'$ $\mu_b(e_i) \neq \mu_b(e'_j)$.

The last definition relates elements that have walked the same number of "time-steps" starting from a common element. In other words, we relate a temporal element to an element that is at the same "succ-depth" –we do not count role edges– in a tree-shaped fusion model. Note that, we can not relate a temporal element with *all* the elements at the same "succ-depth". More precisely, we relate a temporal element at succ-depth n with only one of the temporal evolutions at succ-depth n of a DL-element. The last statement holds in the Definition 3.13, thanks to the definition of n-ch path and, to the condition that for each $0 < i \leq k$, $0 < j \leq k'$ $\mu_b(e_i) \neq \mu_b(e'_j)$.

Example 3.14. We want to identify in the model of the Figure 3.3 some of the elements that are relevant to d'_3. The circled elements in the Figure 3.3 are those that are at d'_3's succ-depth. However, not all the circled elements are relevant to d'_3. As an instance, if we choose $\mathsf{ch}^{\mathcal{I}}(d_1) = d_2$, then there is a 3-ch path from d_0 to d_3 and a 3-succ path from d_0 to d'_3 but $\mu_b(d_3) = \mu_b(d'_3)$. Thus, d_3 is not relevant to d'_3. Another example, if we choose $\mathsf{ch}^{\mathcal{I}}(e_2) = e_3$, then there is no ch path from some $d' \in \Delta^{\mathcal{I}}$ to e'_3. Thus, e'_3 is not relevant to d'_3.

The following are some positive examples. There exists a 1-ch path from d'_2 to f_1 and a 1-succ path from d'_2 to d'_3, such that $f_0 = \mu_b(f_1) \neq \mu_b(d'_3)$. Thus, f_1 is relevant to d'_3. Analogously, there is a 0-ch path from d'_3 to g_0 and

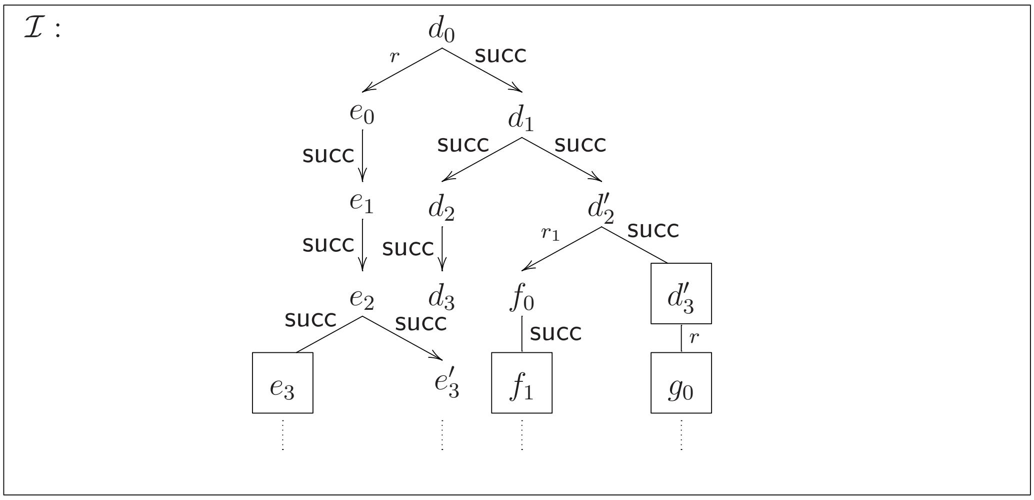


Figure 3.4: Relevant elements to d'_3

a 0-succ path from d'_3 to d'_3, such that $\mu_b(g_0) \neq \mu_b(d'_3)$, then g_0 is relevant to d'_3. In the figure 3.4, the squared elements are the relevant elements to d'_3, if $\mathsf{ch}^{\mathcal{I}}(e_2) = e_3$.

Now, we identify some of the elements that are relevant to f_1. First, we present some negative examples. If we choose $\mathsf{ch}^{\mathcal{I}}(e_2) = e_3$, then there is no ch path from some $d' \in \Delta^{\mathcal{I}}$ to e'_3. Thus, e'_3 is not relevant to f_1. Another example, if we choose $\mathsf{ch}^{\mathcal{I}}(d_1) = d_2$, then there is a 2-ch path from d_1 to d_3 and a 2-succ path from d_1 to f_1 but $\mu_b(d_3) = \mu_b(d'_2)$. Thus, d_3 is not relevant to f_1.

The following are some positive examples. There exists a 1-ch path from d'_2 to d'_3 and a 1-succ path from d'_2 to f_1, such that $f_0 = \mu_b(f_1) \neq \mu_b(d'_3)$. Another example, there exists a 1-ch path from d'_2 to g_0 and a 1-succ path from d'_2 to f_1, such that $f_0 = \mu_b(f_1) \neq g_0$ and $f_0 = \mu_b(f_1) \neq \mu_b(d'_3)$.

Next, we introduce the notion of "temporal paths" to define the states of the temporal model to be constructed. The main idea is that, beginning from the root, we unravel the temporal evolutions of the elements in a fusion model.

Definition 3.15. Let $\mathcal{I}$ be a fusion model and $e \in \Delta^{\mathcal{I}}$ a temporal element. We define $S(e)$ as the set of all temporal elements $e' \in \Delta^{\mathcal{I}}$ such that there is a $d \in \Delta^{\mathcal{I}}$,

- d is relevant to e, and

- $(d, e') \in \mathsf{succ}^{\mathcal{I}}$.

Definition 3.16 (Temporal path). Let $\mathcal{I}$ be a fusion model. A *temporal path* is a sequence $e_0, \ldots, e_n \in \Delta^{\mathcal{I}}$ of temporal elements such that

- e_0 is the root node, and
- $e_{i+1} \in S(e_i)$ for all $i < n$.

An element $d \in \Delta^{\mathcal{I}}$ *is relevant to the temporal path* $e_0, \ldots, e_n$ if d is relevant to e_n.

Example 3.17. In the model of the Figure 3.4, we can construct the temporal paths: d_0, d_0d_1, d_0e_1, $d_0d_1d_2$, $d_0d_1d_2'$, $d_0e_1e_2$, ...

The following lemma establishes the conditions to define how to interpret the elements of the fusion model into the temporal model we are attempting to construct. In particular, we want to guarantee that under the previous conditions we have a unique way to interpret an element in a particular state.

Lemma 3.18. *Let $\mathcal{I}$ be a tree-shaped fusion model, $d \in \Delta^{\mathcal{I}}$ a temporal element and $e, e' \in \Delta^{\mathcal{I}}$. If e and e' are relevant to d, $\mu_b(e) = \mu_b(e')$ implies $e = e'$.*

Proof: This proof is by contradiction. We suppose that $e \neq e'$.

Let $\mu_b(e) = \mu_b(e')$. Then, by definition of μ_b, there are $d_0, \ldots, d_m \in \Delta^{\mathcal{I}}$, $m \geq 0$ such that $(d_i, d_{i+1}) \in \mathsf{succ}^{\mathcal{I}}$ for all $i < m$, $d_0 = \mu_b(e)$ a DL element and $d_m = e$. Analogously, there are $d_0', \ldots, d_{m_1}' \in \Delta^{\mathcal{I}}$, $m_1 \geq 0$ such that $(d_i', d_{i+1}') \in \mathsf{succ}^{\mathcal{I}}$ for all $i < m_1$, $d_0' = d_0$ and $d_{m_1}' = e'$.

Since e is relevant to d, there is a $d' \in \Delta^{\mathcal{I}}$ such that there is an n-succ path $e_0, \ldots, e_k$ from d' to d and a n-ch path $e_0', \ldots, e_{k'}'$ from d' to e for some $n \geq 0$. Analogously, since e' is relevant to d, exists a $d'' \in \Delta^{\mathcal{I}}$ such that there is an n_1-succ path $f_0, \ldots, f_{k_1}$ from d'' to d and an n_1-ch path $f_0', \ldots, f_{k_1'}'$ from d' to e for some $n_1 \geq 0$.

We can distinguish two cases.

($d' = d''$) W.l.o.g. we assume that $m_1 \geq m$. First, assume $m_1 = m$. Since $\mathcal{I}$ is tree-shaped, $d' = d''$ and $e \neq e'$, there is $d_i = d_i' = e_j' = f_j'$, $0 \leq i < m$, $0 \leq j < k' = k_1'$ such that $\mathsf{ch}^{\mathcal{I}}(e_j') \neq \mathsf{ch}^{\mathcal{I}}(f_j')$. But $e_j' = f_j'$ and ch is a function, then this yields a contradiction. Therefore, $e = e'$.

Assume $m_1 > m$. Since $\mathcal{I}$ is tree shaped and $e \neq e$, $n_1 > n$. e, e' are relevant to d, then there is a n_1-succ path from d' to d and a n-succ path from d' to d. But $n_1 > n$ and $\mathcal{I}$ is tree-shaped, then this yields a contradiction. Therefore, $e = e'$.

($d' \neq d''$) W.l.o.g. we assume that there are $d''_0, \ldots, d''_l \in \Delta^{\mathcal{I}}$, $l > 0$ such that there are $r_0, \ldots, r_{l-1} \in N_R \cup \{\mathsf{succ}\}$ with $(d''_i, d''_{i+1}) \in r_i^{\mathcal{I}}$ for all $i < l$, $d'' = d_0$ and $d' = d_l$.

Note that, if all $r_i \in N_R$, then we are in the case $(d = d')$. Analogously, $d' = d_i$ for some $0 < i \leq m$, otherwise we are in the case $(d = d')$.

Then, by definition of succ path, $d' = f_j$ for some $0 \leq i \leq k_1$. Since $\mathcal{I}$ is tree-shaped and $\mu_b(e) = \mu_b(e')$, $\mu_b(d') = \mu_b(e')$. But d' is part of the n_1-succ path and e' is part of the n_1-ch path. This yields a contradiction. Therefore, $e = e'$.

Now, we introduce the function that establishes how to interpret the elements of the fusion model into the constructed temporal model. More precisely, we interpret an element e at state s with the temporal evolution of e that is relevant to s.

Definition 3.19. Let $\mathcal{I}$ be a tree-shaped fusion model and $s = e_0, \ldots, e_n$ a temporal path. μ_b^s is the *restriction* of μ_b to $\{d \in \Delta^{\mathcal{I}} \mid d \text{ is relevant to } s\}$. The function μ_f^s is the converse of μ_b^s.

The following lemma shows that the temporal interpretations with expanding domains and fusion interpretations are equivalent for CTL$_{\mathcal{ALC}}$ concepts and TBoxes.

Lemma 3.20. *Let C be a CTL$_{\mathcal{ALC}}$ concept and $\mathcal{T}$ a TBox. Then there is a temporal model of C and $\mathcal{T}$ with expanding domains if and only if there is a fusion model of C and T.*

Proof: For the 'if' direction, let $\mathcal{I}$ be a tree-shaped fusion model of C and $\mathcal{T}$. We define the temporal interpretation $\mathfrak{J} = (TP, <, I)$ such that

- TP is the set of all temporal paths in $\mathcal{I}$;
- $<= \{(s, s') \in TP \mid s' = se$ for some temporal element $e\}$;
- For $s \in TP$, we define $I(s)$ as follows

$$\begin{array}{rcl} \Delta^{I(s)} & = & \{\, \mu_b(d) \mid d \in \Delta^{\mathcal{I}} \text{ and } d \text{ is relevant to } s\,\}; \\ A^{I(s)} & = & \{d \mid d \in \Delta^{\mathcal{I}} \text{ and } \mu_f^s(d) \in A^{\mathcal{I}}\,\} \text{ for all } A \in N_C; \\ r^{I(s)} & = & \{(d,e) \mid d, e \in \Delta^{\mathcal{I}} \text{ and } (\mu_f^s(d), e) \in r^{\mathcal{I}}\,\} \text{ for all } r \in N_R. \end{array}$$

Example 3.21 (Expanding domains). Given the model of the Figure 3.4, we construct a temporal interpretation $\mathfrak{J} = (TP, <, I)$, where $TP = \{d_0,\ d_0d_1,\ d_0e_1,\ d_0d_1d_2,\ d_0d_1d_2',\ d_0e_1e_2,\ d_0d_1d_2'd_3'\ \ldots\}$. As an instance, it is not hard to see that, $\Delta^{I(d_0d_1d_2')} \subseteq \Delta^{I(d_0d_1d_2'd_3')}$.

Claim : $\mathfrak{J}$ is a temporal model with expanding domains of C and $\mathcal{T}$.

1. For all $r \in N_R$, $s \in TP$ and e, $\mu_f^s(d) \in \Delta^{\mathcal{I}}$, we show that

$$(\mu_f^s(d), e) \in r^{\mathcal{I}} \text{ implies } (d, e) \in r^{I(s)}.$$

   Since $\mu_f^s(d)$ is relevant to s and $(\mu_f^s(d), e) \in r^{\mathcal{I}}$, e is relevant to s.

   Then, by definition of μ_f^s and $\Delta^{I(s)}$, $d \in \Delta^{I(s)}$. Analogously, by definition of μ_b and $\Delta^{I(s)}$, $e \in \Delta^{I(s)}$.

   Therefore, by definition of $r^{I(s)}$, $(d, e) \in r^{I(s)}$.

2. For all concept D_1, $s \in TP$ and $\mu_f^s(d) \in \Delta^{\mathcal{I}}$, we show that

$$\mu_f^s(d) \in D_1^{\mathcal{I}} \text{ implies } d \in D_1^{I(s)}$$

   This proof is by induction on the structure of D_1.

   - $D_1 = A \in N_C$
     It holds, by definition of $\mathfrak{J}$.

   - $D_1 = \neg A \in N_C$
     It holds, by definition of $\mathfrak{J}$.

- $D_1 = C \sqcup D$
  By definition of $\mathcal{I}$, $\mu_f^s(d) \in (C \sqcup D)^{\mathcal{I}}$ implies $\mu_f^s(d) \in C^{\mathcal{I}}$ or $\mu_f^s(d) \in D^{\mathcal{I}}$.

  By definition of μ_f^s and $\Delta^{I^{(s)}}$, $d \in \Delta^{I^{(s)}}$.

  By induction hypothesis, $d \in C^{I(s)}$ or $d \in D^{I(s)}$. Therefore, $d \in (C \sqcup D)^{I(s)}$.

- $D_1 = C \sqcap D$
  By definition of $\mathcal{I}$, $\mu_f^s(d) \in (C \sqcap D)^{\mathcal{I}}$ implies $\mu_f^s(d) \in C^{\mathcal{I}}$ and $\mu_f^s(d) \in D^{\mathcal{I}}$.

  By definition of μ_f^s and $\Delta^{I^{(s)}}$, $d \in \Delta^{I^{(s)}}$.

  By induction hypothesis, $d \in C^{I(s)}$ and $d \in D^{I(s)}$. Therefore, $d \in (C \sqcap D)^{I(s)}$.

- $D_1 = \exists r.D$
  By definition of $\mathcal{I}$, $\mu_f^s(d) \in (\exists r.D)^{\mathcal{I}}$ implies that there is an $e \in \Delta^{\mathcal{I}}$ with $e \in D^{\mathcal{I}}$ and $(\mu_f^s(d), e) \in r^{\mathcal{I}}$.

  Since $\mu_f^s(d)$ is relevant to s and $(\mu_f^s(d), e) \in r^{\mathcal{I}}$, e is relevant to s.

  Then, by definition of $\mu_f^s(d)$ and $\Delta^{I^{(s)}}$, $d \in \Delta^{I^{(s)}}$. Analogously, by definition of μ_b and $\Delta^{I(s)}$, $e \in \Delta^{I(s)}$.

  By point 1, $(d, e) \in r^{I(s)}$ and, by induction hypothesis, $e \in D^{I(s)}$. Therefore, $d \in (\exists r.D)^{I(s)}$.

- $D_1 = \forall r.D$
  By definition of $\mathcal{I}$, $\mu_f^s(d) \in (\forall r.D)^{\mathcal{I}}$ implies that for all $e \in \Delta^{\mathcal{I}}$, $(\mu_f^s(d), e) \in r^{\mathcal{I}}$ implies $e \in D^{\mathcal{I}}$.

  Let $e_1 \in \Delta^{\mathcal{I}}$ such that $(\mu_f^s(d), e_1) \in r^{\mathcal{I}}$. We have that $\mu_f^s(d)$ is relevant to s and $(\mu_f^s(d), e_1) \in r^{\mathcal{I}}$, then e_1 is relevant to s .

  By definition of μ_f^s and $\Delta^{I^{(s)}}$, $d \in \Delta^{I^{(s)}}$. Analogously, by definition of μ_b and $\Delta^{I(s)}$, $e_1 \in \Delta^{I(s)}$.

  By point 1, $(d, e_1) \in r^{I(s)}$ and, by induction hypothesis, $e_1 \in D^{I(s)}$. Therefore, $d \in (\forall r.D)^{I(s)}$.

- $D_1 = (\mathsf{E}\bigcirc C)$
  By definition of $\mathcal{I}$, $\mu_f^s(d) \in (\mathsf{E}\bigcirc C)^{\mathcal{I}}$ implies that there is a $d_1 \in \Delta^{\mathcal{I}}$ with $(\mu_f^s(d), d_1) \in \mathsf{succ}^{\mathcal{I}}$ and $d_1 \in C^{\mathcal{I}}$.

By definition of temporal path, there is an $s^{d_1} \in TP$ such that $s^{d_1} = sd_1$.
By definition of $\mu^s_f(d)$ and $\Delta^{I^{(s)}}$, $d \in \Delta^{I^{(s)}}$. Analogously, since $\mathcal{I}$ is tree-shaped and d_1 is relevant to s^{d_1}, $d \in \Delta^{I^{(s^{d_1})}}$.
By construction, $s < s^{d_1}$ and, by induction hypothesis, $d \in C^{I(s^{d_1})}$.
Therefore, $d \in (\mathbf{E}\bigcirc C)^{I(s)}$.

- $D_1 = (\mathbf{A}\bigcirc C)$
  By definition of $\mathcal{I}$, $\mu^s_f(d) \in (\mathbf{A}\bigcirc C)^{\mathcal{I}}$ implies that for all $e \in \Delta^{\mathcal{I}}$, $(\mu^s_f(d), e) \in \mathsf{succ}^{\mathcal{I}}$ implies $e \in C^{\mathcal{I}}$.
  Let $e_1 \in \Delta^{\mathcal{I}}$ such that $(\mu^s_f(d), e_1) \in \mathsf{succ}^{\mathcal{I}}$. Since $(\mu^s_f(d), e_1) \in \mathsf{succ}^{\mathcal{I}}$ and $\mu^s_f(d)$ is relevant to s, there is an $s^{e_1} \in TP$ such that $s^{e_1} = se_1$.
  By definition of $\mu^s_f(d)$ and $\Delta^{I^{(s)}}$, $d \in \Delta^{I^{(s)}}$. Analogously, since $\mathcal{I}$ is tree-shaped and e_1 is relevant to s^{e_1}, $d \in \Delta^{I^{(s^{e_1})}}$.
  By construction, $s < s^{e_1}$ and, by induction hypothesis, $d \in C^{I(s^{e_1})}$.
  Therefore, $d \in (\mathbf{A}\bigcirc C)^{I(s)}$.

- $D_1 = (\mathbf{E}(C\mathcal{U}D))$
  By definition of $\mathcal{I}$, $\mu^s_f(d) \in (\mathbf{E}(C\mathcal{U}D))^{\mathcal{I}}$ implies that there are $d_0, d_1, d_2, \ldots \in \Delta^{\mathcal{I}}$ such that $(d_i, d_{i+1}) \in \mathsf{succ}^{\mathcal{I}}$ for all $i \geq 0$ and $\mu^s_f(d) = d_0$: there is an $m \geq 0$ ($d_m \in D^{\mathcal{I}}$ and $d_k \in C^{\mathcal{I}}$ for all $k < m$).
  By definition of temporal path, there is an $s^{d_i} \in TP$ for all $i > 0$ such that $s^{d_i} = s, \ldots, d_i$. By construction, $s^{d_i} < s^{d_{i+1}}$.
  By definition of $\mu^s_f(d)$ and $\Delta^{I^{(s)}}$, $d \in \Delta^{I^{(s)}}$. Analogously, since $\mathcal{I}$ is tree-shaped and d_i is relevant to s^{d_i}, $d \in \Delta^{I^{(s^{d_i})}}$ for all $i > 0$.
  We can construct $s^{d_m} = s, \ldots, d_m$, $m \geq 0$ and $s^{d_k} = s, \ldots, d_k$, $k < m$. By induction hypothesis, $d \in D^{I(s^{d_m})}$ and $d \in C^{I(s^{d_k})}$ for all $k < m$. Therefore, $d \in (\mathbf{E}(C\mathcal{U}D))^{I(s)}$.

- $D_1 = (\mathbf{A}(C\mathcal{U}D))$
  By definition of $\mathcal{I}$, $\mu^s_f(d) \in (\mathbf{A}(C\mathcal{U}D))^{\mathcal{I}}$ implies that for all $d_0, d_1, d_2, \ldots \in \Delta^{\mathcal{I}}$, $(d_i, d_{i+1}) \in \mathsf{succ}^{\mathcal{I}}$ for all $i \geq 0$ and $\mu^s_f(d) = d_0$ implies there is an $m \geq 0$ ($d_m \in D^{\mathcal{I}}$ and $d_k \in C^{\mathcal{I}}$ for all $k < m$).
  Let $d'_0, d'_1, d'_2, \ldots \in \Delta^{\mathcal{I}}$ such that $(d'_i, d'_{i+1}) \in \mathsf{succ}^{\mathcal{I}}$ for all $i \geq 0$ and $\mu^s_f(d) = d'_0$. By definition of temporal path, there is an $s^{d'_i} \in TP$ for all $i > 0$ such that $s^{d'_i} = s, \ldots, d'_i$. By construction, $s^{d'_i} < s^{d'_{i+1}}$.

By definition of $\mu_f^s(d)$ and $\Delta^{I(s)}$, $d \in \Delta^{I(s)}$. Analogously, since $\mathcal{I}$ is tree-shaped and d'_i is relevant to $s^{d'_i}$, $d \in \Delta^{I(s^{d'_i})}$ for all $i > 0$.
We can construct $s^{d'_m} = s, \ldots, d'_m$, $m \geq 0$ and $s^{d'_k} = s, \ldots, d'_k$, $k < m$. By induction hypothesis, $d \in D^{I(s^{d'_m})}$ and $d \in C^{I(s^{d'_k})}$ for all $k < m$. Therefore, $d \in (\mathbf{A}(C \mathcal{U} D))^{I(s)}$.

- $D_1 = (\mathbf{E}(C \mathcal{R} D))$
  By definition of $\mathcal{I}$, $\mu_f^s(d) \in (\mathbf{E}(C \mathcal{R} D))^{\mathcal{I}}$ implies that there are $d_0, d_1, d_2, \ldots \in \Delta^{\mathcal{I}}$ such that $(d_i, d_{i+1}) \in \mathsf{succ}^{\mathcal{I}}$ for all $i \geq 0$ and $\mu_f^s(d) = d_0$ such that for all $j \geq 0$: if $d_k \notin C^{\mathcal{I}}$ for all $k < j$ then $d_j \in D^{\mathcal{I}}$.
  By definition of temporal path, there is an $s^{d_i} \in TP$ for all $i > 0$ such that $s^{d_i} = s, \ldots, d_i$. By construction, $s^{d_i} < s^{d_{i+1}}$.
  By definition of $\mu_f^s(d)$ and $\Delta^{I(s)}$, $d \in \Delta^{I(s)}$. Analogously, since $\mathcal{I}$ is tree-shaped and d_i is relevant to s^{d_i}, $d \in \Delta^{I(s^{d_i})}$ for all $i > 0$.
  We can construct $s^{d_j} = s, \ldots, d_j$, $j \geq 0$ and $s^{d_k} = s, \ldots, d_k$, $k < j$.
  By induction hypothesis, $d \notin C^{I(s^{d_k})}$ for all $k < j$ and $d \in D^{I(s^{d_j})}$.
  Therefore, $d \in (\mathbf{E}(C \mathcal{R} D))^{I(s)}$.

- $D_1 = (\mathbf{A}(C \mathcal{R} D))$
  By definition of $\mathcal{I}$, $\mu_f^s(d) \in (\mathbf{A}(C \mathcal{R} D))^{\mathcal{I}}$ implies that for all $d_0, d_1, d_2, \ldots \in \Delta^{\mathcal{I}}$, $(d_i, d_{i+1}) \in \mathsf{succ}^{\mathcal{I}}$ for all $i \geq 0$ and $\mu_f^s(d) = d_0$ implies that for all $j \geq 0$: if $d_k \notin C^{\mathcal{I}}$ for all $k < j$ then $d_j \in D^{\mathcal{I}}$.
  Let $d'_0, d'_1, d'_2, \ldots \in \Delta^{\mathcal{I}}$, such that $(d'_i, d'_{i+1}) \in \mathsf{succ}^{\mathcal{I}}$ for all $i \geq 0$ and $\mu_f(d) = d'_0$. By definition of temporal path, there is an $s^{d'_i} \in TP$ for all $i > 0$ such that $s^{d'_i} = s, \ldots, d'_i$. By construction, $s^{d'_i} < s^{d'_{i+1}}$.
  By definition of $\mu_f^s(d)$ and $\Delta^{I(s)}$, $d \in \Delta^{I(s)}$. Analogously, since $\mathcal{I}$ is tree-shaped and d'_i is relevant to $s^{d'_i}$, $d \in \Delta^{I(s^{d'_i})}$ for all $i > 0$.
  We can construct $s^{d'_j} = s, \ldots, d'_j$, $j \geq 0$ and $s^{d'_k} = s, \ldots, d'_k$, $k < j$.
  By induction hypothesis, $d \notin C^{I(s^{d'_k})}$ for all $k < j$ and $d \in D^{I(s^{d'_j})}$.
  Therefore, $d \in (\mathbf{A}(C \mathcal{R} D))^{I(s)}$.

Therefore, given a tree-shaped fusion model $\mathcal{I}$ of C and $\mathcal{T}$, such that $d \in C^{\mathcal{I}}$ with d the root of $\mathcal{I}$ we can construct a temporal model $\mathfrak{J}$ of C and $\mathcal{T}$ with expanding domains such that $\mu_b(d) \in C^{I(d)}$.

For the 'only if' direction, let $\mathfrak{J} = (S, <, I)$ a temporal model of C and $\mathcal{T}$. We define a fusion interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ such that

$$\begin{array}{rcl} \Delta^{\mathcal{I}} & \subseteq & \{(s,d) \mid d \in \Delta^{I(s)}, s \in S\}; \\ A^{\mathcal{I}} & = & \{(s,d) \mid s \in S \text{ and } d \in A^{I(s)}\} \text{ for all } A \in N_C; \\ r^{\mathcal{I}} & = & \{((s,d),(s,e)) \mid s \in S \text{ and } (d,e) \in r^{I(s)}\} \text{ for all } r \in N_R; \\ \mathsf{succ}^{\mathcal{I}} & = & \{((s,d),(s',d)) \mid s,s' \in S, s < s', d \in \Delta^{I(s)} \text{and } d \in \Delta^{I(s')}\}. \end{array}$$

Claim: $\mathcal{I}$ is a fusion model for C and $\mathcal{T}$.

1. For all $r \in N_R$, $d, e \in \Delta^{\mathfrak{J}}$ and $s \in S$, we show that

$$(s,d,e) \in r^{\mathfrak{J}} \text{ implies } ((s,d),(s,e)) \in r^{\mathcal{I}}.$$

   By definition of $\mathfrak{J}$, $(s,d,e) \in r^{\mathfrak{J}}$ implies that $s \in S$ and $(d,e) \in r^{I(s)}$.

   Therefore, by definition of $r^{\mathcal{I}}$, $((s,d),(s,e)) \in r^{\mathcal{I}}$.

2. For all concept C_1, $d \in \Delta^{\mathfrak{J}}$ and $s \in S$, we show that,

$$(s,e) \in C_1^{\mathfrak{J}} \text{ implies } (s,e) \in C_1^{\mathcal{I}}.$$

   This proof is by induction on the structure of C_1. We proof only some cases.

   - $C_1 = A \in N_C$
     It holds, by definition of $\mathcal{I}$.

   - $C_1 = \neg A \in N_C$
     It holds, by definition of $\mathcal{I}$.

   - $C_1 = C \sqcap D$
     By definition of $\mathfrak{J}$, $(s,e) \in (C \sqcap D)^{\mathfrak{J}}$ implies that $s \in S$, $e \in C^{I(s)}$ and $e \in D^{I(s)}$.

     By induction hypothesis, $(s,e) \in C^{\mathcal{I}}$ and $(s,e) \in D^{\mathcal{I}}$. Therefore, $(s,e) \in (C \sqcap D)^{\mathcal{I}}$.

   - $C_1 = \exists r.D$
     By definition of $\mathfrak{J}$, $(s,e) \in (\exists r.D)^{\mathfrak{J}}$ implies that there is an $e_1 \in \Delta^{I(s)}$ with $(s,e_1) \in D^{\mathfrak{J}}$ and $(s,e,e_1) \in r^{\mathfrak{J}}$.

We have that, $(e, e_1) \in r^{I(s)}$ then, by point 1, $((s, e), (s, e_1)) \in r^{\mathcal{I}}$. By induction hypothesis, $(s, e_1) \in D^{\mathcal{I}}$. Therefore, $(s, e) \in (\exists r.D)^{\mathcal{I}}$.

- $C_1 = (\mathbf{E}\bigcirc D)$
  By definition of $\mathfrak{J}$, $(s, e) \in (\mathbf{E}\bigcirc D)^{\mathfrak{J}}$ implies that there is an $s_1 \in S$ with $s < s_1$ and $(s_1, e) \in D^{\mathfrak{J}}$.

  We have that, $e \in \Delta^{I(s_1)}$ and, by definition of $I(s)$, $\Delta^{I(s)} \subseteq \Delta^{I(s_1)}$ then $e \in \Delta^{I(s)}$. Then, since $s < s_1$, $((s, e), (s_1, e)) \in \mathsf{succ}^{\mathcal{I}}$. By induction hypothesis, $(s_1, e) \in D^{\mathcal{I}}$. Therefore, $(s, e) \in (\mathbf{E}\bigcirc D)^{\mathcal{I}}$.

- $C_1 = (\mathbf{E}(D\mathcal{U}D_1))$
  By definition of $\mathfrak{J}$, $(s, e) \in (\mathbf{E}(D\mathcal{U}D_1))^{\mathfrak{J}}$ implies that there are $s_0 < s_1 < s_2 \ldots$ with $s = s_0 : \exists\, m \geq 0\, ((s_m, e) \in D_1^{\mathfrak{J}}$ and $(s_k, e) \in D^{\mathfrak{J}}$ for all $k < m)$.

  By definition of $\mathcal{I}$, there are $(s_0, e), (s_1, e), (s_2, e) \ldots \in \Delta^{\mathcal{I}}$ such that $((e, s_i), (e, s_{i+1})) \in \mathsf{succ}^{\mathcal{I}}$ for all $i \geq 0$ and $(s, e) = (s_0, e)$.

  By induction hypothesis, $(s_m, e) \in D_1^{\mathcal{I}}$, $m \geq 0$ and $(s_k, e) \in D^{\mathcal{I}}$ for all $k < m$. Therefore, $(s, e) \in (\mathbf{E}(D\mathcal{U}D_1))^{\mathcal{I}}$.

- $C_1 = (\mathbf{E}(D\mathcal{R}D_1))$
  By definition of $\mathfrak{J}$, $(s, e) \in (\mathbf{E}(D\mathcal{R}D_1))^{\mathfrak{J}}$ implies there are $s_0 < s_1 < s_2 \ldots$ with $s = s_0$ such that for all $j \geq 0$: if $(s_k, e) \notin D^{\mathfrak{J}}$ for all $k < j$ then $(s_j, e) \in D_1^{\mathfrak{J}}$.

  By definition of $\mathcal{I}$, there are $(s_0, e), (s_1, e), (s_2, e) \ldots \in \Delta^{\mathcal{I}}$ such that $((e, s_i), (e, s_{i+1})) \in \mathsf{succ}^{\mathcal{I}}$ for all $i \geq 0$ and $(s, e) = (s_0, e)$.

  By induction hypothesis, for all $j \geq 0$, $(s_k, e) \notin D^{\mathcal{I}}$ for all $k < j$ and $(s_j, e) \in D_1^{\mathcal{I}}$. Therefore, $(s, e) \in (\mathbf{E}(D\mathcal{R}D_1))^{\mathcal{I}}$.

Thus, given a temporal model $\mathfrak{J}$ of C and $\mathcal{T}$ we can construct a fusion model $\mathcal{I}$ of C and $\mathcal{T}$.

Therefore, there is a temporal model of C and $\mathcal{T}$ if and only if there is a fusion model of C and $\mathcal{T}$. □

3.4 Introducing the μ-calculus

In 1983, Dexter Kozen introduced the μ-calculus (Kozen, 1983; Kozen & Parikh, 1983). The μ-calculus comes not from the philosophical tradition of modal logic, but from the application of modal and temporal logics to program verification. This logic, the μ-calculus, is a propositional modal logic augmented with least and great fixpoint operators. Intuitively, the μ-calculus makes it possible to characterize the modalities in terms of recursively defined tree-like patterns.

3.4.1 μ-calculus syntax

Definition 3.22 (μ-calculus syntax). Formulae $\phi, \psi, \ldots$ are formed inductively from atomic formulae $A, \ldots$ and variables $X, \ldots$ according to the following abstract syntax:

$$\phi, \psi ::= A \mid \top \mid \bot \mid \neg\psi \mid \psi \wedge \phi \mid \psi \vee \phi \mid \langle a \rangle \psi \mid [a]\psi \mid \mu X.\psi \mid \nu X.\psi \mid X$$

where a is a generic element of a set of labels $\mathcal{L}$, and every bounded occurrence of every variable X must be in the scope of an even number of negation signs.

We call μ and ν *fixpoint operators*. A *sentence* is a formula without free variables.

Example 3.23. We can use the μ-calculus to express the usual operators of temporal logics. As an instance, consider the CTL formula $\mathsf{A}\Box\phi$. Another way of expressing $\mathsf{A}\Box\phi$ is the following: there is a property X such that if X is true, then ϕ is true, and wherever we go next, X remains true. The last statement can be be described by the next formula.

$$\nu Z.\phi \wedge [-]Z$$

The following formula states that,"on some a-path, P holds until Q holds".

$$\mu Z.Q \vee (P \wedge \langle a \rangle Z)$$

3.4.2 μ-calculus semantics

The semantics of the μ-calculus is based on the notions of a structure and a valuation. A *Kripke structure* $\mathcal{M}$ is a triple $(\mathcal{S}, \{\mathcal{R}_i \mid i \in \mathcal{L}\}, \mathcal{V})$, where $\mathcal{S}$ is a set of states, each $\mathcal{R}_i$ is a binary relation on $\mathcal{S}$, and $\mathcal{V}$ is a mapping from atomic formulae to subsets of $\mathcal{S}$. A *valuation* ρ on $\mathcal{M}$ is a mapping from the variables to subsets of $\mathcal{S}$. Given a valuation ρ, we denote by $\rho[X/\mathcal{E}]$ the valuation identical to ρ except for $\rho[X/\mathcal{E}](X) = \mathcal{E}$.

Definition 3.24 (μ-calculus semantics). Let $\mathcal{M}$ be a Kripke structure and ρ a valuation on $\mathcal{M}$. The *extension function* $\cdot_\rho^{\mathcal{M}}$ is inductively defined as follows:

$$
\begin{array}{lcl}
X_\rho^{\mathcal{M}} & = & \rho(X) \subseteq \mathcal{S}, \\
A_\rho^{\mathcal{M}} & = & \mathcal{V}(A) \subseteq \mathcal{S}, \\
\top_\rho^{\mathcal{M}} & = & \mathcal{S}, \\
\bot_\rho^{\mathcal{M}} & = & \emptyset, \\
(\neg\Phi)_\rho^{\mathcal{M}} & = & \mathcal{S} \setminus \Phi_\rho^{\mathcal{M}}, \\
(\Phi \wedge \Psi)_\rho^{\mathcal{M}} & = & \Phi_\rho^{\mathcal{M}} \cap \Psi_\rho^{\mathcal{M}}, \\
(\Phi \vee \Psi)_\rho^{\mathcal{M}} & = & \Phi_\rho^{\mathcal{M}} \cup \Psi_\rho^{\mathcal{M}}, \\
(\langle a \rangle\Phi)_\rho^{\mathcal{M}} & = & \{s \in \mathcal{S} \mid \exists s' \text{ with } (s,s') \in R_a \text{ and } s' \in \Phi_\rho^{\mathcal{M}}\}, \\
([a]\Phi)_\rho^{\mathcal{M}} & = & \{s \in \mathcal{S} \mid \forall s', (s,s') \in R_a \text{ implies } s' \in \Phi_\rho^{\mathcal{M}}\}, \\
(\mu X.\Phi)_\rho^{\mathcal{M}} & = & \bigcap\{\mathcal{E} \subseteq S \mid \Phi_{\rho[X/\mathcal{E}]}^{\mathcal{M}} \subseteq \mathcal{E}\}, \\
(\nu X.\Phi)_\rho^{\mathcal{M}} & = & \bigcup\{\mathcal{E} \subseteq S \mid \mathcal{E} \subseteq \Phi_{\rho[X/\mathcal{E}]}^{\mathcal{M}}\}.
\end{array}
$$

A formula Φ is *satisfiable* if there exists a Kripke structure $\mathcal{M}$ and a valuation ρ on $\mathcal{M}$ such that $\Phi_\rho^{\mathcal{M}} \neq \emptyset$. If Φ is a sentence we can omit the valuation.

Definition 3.25. A Kripke structure $\mathcal{M} = (\mathcal{S}, \{\mathcal{R} \mid i \in \mathcal{L}\}, \mathcal{V})$ is *rooted* if there is a $s_0 \in \mathcal{S}$ such that $\mathcal{S} = \{s \in \mathcal{S} \mid s_0\mathcal{R}^*s\}$, where

$$\mathcal{R} = \bigcup_{i \in \mathcal{L}} \mathcal{R}_i.$$

Definition 3.26. Let $\mathcal{M}$ be a Kripke structure, ρ a valuation on $\mathcal{M}$ and Φ a formula. $\mathcal{M}$ is a *rooted model* for Φ if $\mathcal{M}$ is rooted and $s_0 \in \Phi_\rho^{\mathcal{M}}$ where s_0 is the root of $\mathcal{M}$.

Lemma 3.27 ((Vardi, 1997)). *Every μ-calculus formula has a rooted model $\mathcal{M}$.*

3.5 Relating CTL$_{\mathcal{ALC}}$ and the μ-calculus

In this section, we relate satisfiability in CTL$_{\mathcal{ALC}}$ with satisfiability in the μ-calculus. First, we show that we can view a Kripke structure as a DL interpretation and vice versa.

Given a Kripke structure $\mathcal{M} = (\mathcal{S}, \{R_i \mid i \in \mathcal{L}\}, \mathcal{V})$, we can construct a DL interpretation $\mathcal{I}_\mathcal{M}$ such that $\Delta^{\mathcal{I}_\mathcal{M}} = \mathcal{S}$; $r_i^{\mathcal{I}_\mathcal{M}} = R_i$ for each $i \in \mathcal{L}$ and $A^{\mathcal{I}_\mathcal{M}} = \mathcal{V}(A)$ for each atomic formula A. Vice versa, given a DL interpretation $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, we can construct a Kripke structure $\mathcal{M}_\mathcal{I} = (\Delta^\mathcal{I}, \{R_i \mid i \in \mathcal{L}\}, \mathcal{V})$ such that $\mathcal{L} = N_R$; $R_r = r^\mathcal{I}$ for each $r \in N_R$ and $\mathcal{V}(A) = A^\mathcal{I}$ for each $A \in N_C$.

Now, we set the correspondence between CTL$_{\mathcal{ALC}}$ and the μ-calculus. To this aim, we define the translation $\cdot^\dagger$ of CTL$_{\mathcal{ALC}}$ concepts into the μ-calculus formulas.

$$
\begin{array}{rcl}
A \in {N_C}^\dagger & = & A, \\
(\neg C)^\dagger & = & \neg C^\dagger, \\
(C \sqcap D)^\dagger & = & C^\dagger \wedge D^\dagger, \\
(\exists r.C)^\dagger & = & \langle r \rangle C^\dagger, \\
(\forall r.C)^\dagger & = & [r] C^\dagger, \\
(\mathsf{E} \bigcirc C)^\dagger & = & \langle \mathsf{succ} \rangle C^\dagger, \\
(\mathsf{A} \bigcirc C)^\dagger & = & [\mathsf{succ}] C^\dagger, \\
(\mathsf{E}(C \mathcal{U} D))^\dagger & = & \mu Y.(D^\dagger \vee (C^\dagger \wedge \langle \mathsf{succ} \rangle Y)), \\
(\mathsf{A}(C \mathcal{U} D))^\dagger & = & \mu Y.(D^\dagger \vee (C^\dagger \wedge [\mathsf{succ}] Y)), \\
(\mathsf{E}(C \mathcal{R} D))^\dagger & = & \nu Y.(D^\dagger \wedge (C^\dagger \vee \langle \mathsf{succ} \rangle Y)), \\
(\mathsf{A}(C \mathcal{R} D))^\dagger & = & \nu Y.(D^\dagger \wedge (C^\dagger \vee [\mathsf{succ}] Y)).
\end{array}
$$

We define the translation of the pair $(C, \mathcal{T})$ as follows

$$
(C, \mathcal{T})^\dagger \quad = \quad C^\dagger \wedge \nu Y.(C_\mathcal{T}^\dagger \wedge [\mathsf{succ}] Y),
$$

where C is a concept, $\mathcal{T} = \{C_1 \sqsubseteq D_1, \ldots, C_q \sqsubseteq D_q\}$ and $C_\mathcal{T} = (\neg C_1 \sqcup D_1) \sqcap \ldots \sqcap (\neg C_q \sqcup D_q)$.

Lemma 3.28. *Let C be a CTL$_{\mathcal{ALC}}$ concept and $\mathcal{T}$ a TBox.*

1. *If $\mathcal{I}$ is a model of C and $\mathcal{T}$, then $\mathcal{M}_\mathcal{I}$ is a model of $(C, \mathcal{T})^\dagger$.*

2. If $\mathcal{M}$ is a rooted model of $(C, \mathcal{T})^{\dagger}$, then $\mathcal{I}_{\mathcal{M}}$ is a model of C and $\mathcal{T}$.

Proof Sketch: For *1*, let $\mathcal{I}$ be a model of C and $\mathcal{T}$. We can define the Kripke structure $\mathcal{M}_{\mathcal{I}}$ as above. Then, we have to prove that for all concepts D and for all $d \in \Delta^{\mathcal{I}}$

$$d \in D^{\mathcal{I}} \text{ implies } d \in (D^{\dagger})^{\mathcal{M}_{\mathcal{I}}}.$$

Note that, any resulting μ-calculus formula from $\cdot^{\dagger}$ is a sentence. Thus, we can omit the valuation.

We can prove the last statement by induction on the structure of D.

Since $\mathcal{I}$ is a model of $\mathcal{T}$, $\mathcal{T}$ holds in every state. Therefore, $M_{\mathcal{I}}$ is a model of $(C, \mathcal{T})^{\dagger}$.

For *2*, let $\mathcal{M}$ be a rooted model of $(C, \mathcal{T})^{\dagger}$. We can define a DL interpretation $\mathcal{I}_{\mathcal{M}}$ as above. Then, we have to prove that for all concepts D and for all $d \in \mathcal{S}$

$$d \in (D)^{\mathcal{M}} \text{ implies } d \in D^{\mathcal{I}_{\mathcal{M}}}.$$

We can prove this by induction on the structure of D.

Since $\mathcal{M}$ is a rooted model of $(C, \mathcal{T})^{\dagger}$, by definition of $(C, \mathcal{T})^{\dagger}$, $\mathcal{I}_{\mathcal{M}}$ is a model of C and $\mathcal{T}$. $\square$

Theorem 3.29. *Concept satisfiability w.r.t. TBoxes with expanding domains in CTL$_{\mathcal{ALC}}$ is EXPTIME-complete.*

The satisfiability problem for the μ-calculus is EXPTIME complete (Emerson & Jutla, 1988), by lemma 3.28, EXPTIME is transfered as the upper bound for concept satisfiability w.r.t. TBoxes with expanding domains in CTL$_{\mathcal{ALC}}$ and the lower bound carries over from $\mathcal{ALC}$.

Note that, in the constant domain case Lemma 3.20 does not hold. For example, the TBox 3.1 has a fusion model, but no temporal model with constant domains. Thus, in the constant domain case, we cannot use the translation $\cdot^{\dagger}$ to relate CTL$_{\mathcal{ALC}}$ with the μ-calculus.

Chapter 4

Temporal concepts: $\mathrm{CTL}_{\mathcal{EL}}$ concepts

In this chapter, we investigate the temporal description logic $\mathrm{CTL}_{\mathcal{EL}}$. First, we introduce its syntax and semantics. Thereafter, we determine the computational complexity of reasoning in $\mathrm{CTL}_{\mathcal{EL}}$.

We combine the description logic $\mathcal{EL}$ and the temporal logic computation tree logic (CTL) to construct the temporal description logic $\mathrm{CTL}_{\mathcal{EL}}$. As in the case of $\mathrm{CTL}_{\mathcal{ALC}}$, we focus on the temporal evolution of concepts. Thus, we define the temporal description logic $\mathrm{CTL}_{\mathcal{EL}}$ whose concepts are formed using the constructors of $\mathcal{EL}$ enriched with the CTL temporal operators.

4.1 $\mathrm{CTL}_{\mathcal{EL}}$ syntax

Definition 4.1 ($\mathrm{CTL}_{\mathcal{EL}}$ syntax). Let N_C and N_R be disjoint sets of concept names and role names, respectively. The set of $\mathrm{CTL}_{\mathcal{EL}}$-*concept descriptions* is defined inductively as follows:

1. Each concept name $A \in N_C$ is a $\mathrm{CTL}_{\mathcal{EL}}$-concept description.
2. $\top$ is a $\mathrm{CTL}_{\mathcal{EL}}$-concept description.
3. If C, D are $\mathrm{CTL}_{\mathcal{EL}}$-concept descriptions and $r \in N_R$, then
    - $C \sqcap D$ and $\exists r.C$ are also $\mathrm{CTL}_{\mathcal{EL}}$-concept descriptions.
    - $\mathsf{A}\bigcirc C$, $\mathsf{E}\bigcirc C$, $\mathsf{A}(C\mathcal{U}D)$, $\mathsf{E}(C\mathcal{U}D)$ are also $\mathrm{CTL}_{\mathcal{EL}}$-concept descriptions.

Example 4.2. The following concept describes processes which necessarily have subprocesses that start at some point in the future.

$$\mathsf{Process} \sqcap \mathbf{A}\Diamond(\exists\,\mathsf{starting.SubProcess})$$

The next concept describes processes that have the possibility to access their critical section.

$$\mathsf{Process} \sqcap \mathbf{E}\Diamond(\exists\,\mathsf{has\_access.CriticalSection})$$

We define TBoxes in the same way as in the case of $\mathcal{EL}$ but now using CTL$_{\mathcal{EL}}$ concepts instead of $\mathcal{EL}$ concepts.

Example 4.3. The following GCI states that, there is the possibility that exists a time in the future which on US citizens will always have health insurance.

$$\mathsf{UScitizen} \sqsubseteq \mathbf{A}\Box\mathbf{E}\Diamond(\exists \mathsf{insured\_by.HealthInsurer})$$

4.2 CTL$_{\mathcal{EL}}$ semantics

We interpret CTL$_{\mathcal{EL}}$ in models based on a tree in which every state s comes equipped with an $\mathcal{EL}$-model describing the domain at state s. In particular, we focus on temporal interpretations with *expanding domains*, i.e., it is assumed that the domain of the $\mathcal{EL}$-model at state s is included in all states following s. In other words, objects can be created over time, but do not destroyed.

Definition 4.4 (CTL$_{\mathcal{EL}}$ semantics). We define the semantics of CTL$_{\mathcal{EL}}$ concept descriptions as we did for CTL$_{\mathcal{ALC}}$, see Definition 3.4. Next, we give the semantics of $\mathbf{A}\Diamond$ and $\mathbf{E}\Diamond$. Let $\mathfrak{J} = (S, <, I)$ be a temporal interpretation. Then,

- $(\mathbf{E}\Diamond C)^{\mathfrak{J}} = \{(s, d) \mid \exists s_0 < s_1 < s_2 \ldots \text{ with } s = s_0 \text{ such that there is a } k > 0 \ : (s_k, d) \in C^{\mathfrak{J}}\}$,
- $(\mathbf{A}\Diamond C)^{\mathfrak{J}} = \{(s, d) \mid \forall s_0 < s_1 < s_2 \ldots \text{ with } s = s_0 \text{ implies there is a } k > 0 \ : (s_k, d) \in C^{\mathfrak{J}}\}$.

As in the case of $\mathcal{EL}$, the satisfiability problem in CTL$_{\mathcal{EL}}$ is not interesting. Observe that every concept is satisfiable w.r.t. every TBox: they are satisfied in the model where all the concepts and roles are interpreted by the whole domain at every state. In fact, the interesting reasoning problem for $\mathrm{CTL}_{\mathcal{EL}}$ is concept subsumption.

A temporal interpretation $\mathfrak{J}$ is a *model* of a TBox $\mathcal{T}$ if and only if $C^{\mathfrak{J}} \subseteq D^{\mathfrak{J}}$ for all $C \sqsubseteq D \in \mathcal{T}$. Thus, the GCIs are regarded as temporally global constraints in the sense that they should hold at every state. A $\mathrm{CTL}_{\mathcal{EL}}$ concept C *is subsumed by a* $\mathrm{CTL}_{\mathcal{EL}}$ concept D *w.r.t.* $\mathcal{T}$ ($C \sqsubseteq_{\mathcal{T}} D$) if and only if $C^{\mathfrak{J}} \subseteq D^{\mathfrak{J}}$ for all models $\mathfrak{J}$ of $\mathcal{T}$.

4.3 $\mathrm{CTL}_{\mathcal{EL}}$ computational complexity

In the following theorem, we prove that the computational complexity of reasoning in $\mathrm{CTL}_{\mathcal{EL}}$ does not remain tractable as in the case of pure $\mathcal{EL}$. More precisely, we prove that the computational complexity of reasoning in $\mathrm{CTL}_{\mathcal{EL}}$ is EXPTIME-complete.

Theorem 4.5. *Concept subsumption w.r.t. TBoxes with expanding domains in $CTL_{\mathcal{EL}}$ is EXPTIME complete.*

The upper bound follows from $\mathrm{CTL}_{\mathcal{EL}}$ being a fragment of $\mathrm{CTL}_{\mathcal{ALC}}$. For the lower bound, we reduce the satisfiability problem w.r.t. TBoxes for $\mathcal{ALC}$ to the subsumption problem w.r.t. TBoxes for $\mathrm{CTL}_{\mathcal{EL}}$. Recall that, the former is EXPTIME-hard.

4.3.1 From $\mathcal{ALC}$ satisfiability to $\mathrm{CTL}_{\mathcal{EL}}$ subsumption

In this section, we give a stepwise reduction from $\mathcal{ALC}$ satisfiability to $\mathrm{CTL}_{\mathcal{EL}}$ subsumption.

Suppose that an $\mathcal{ALC}$ concept C and a TBox $\mathcal{T}$ are given. We assume that in C and $\mathcal{T}$ do not occur subconcepts of the form $\forall r.D$, i.e., for all restrictions are given in terms of existential restrictions and negation. First, we perform a number of satisfiability preserving operations.

($\mathcal{ALC} \rightarrow \mathcal{ALC}_1$) We ensure that negation occurs only in front of concept names. For every concept $\neg D$ with D complex,

1. we introduce a fresh concept name A',
2. we replace $\neg D$ with $\neg A'$,

3. we add $A' \sqsubseteq D$ and $D \sqsubseteq A'$ to $\mathcal{T}$.

We denote the resulting concept by C_0 and the TBox by $\mathcal{T}_0$. We have to show that C is satisfiable w.r.t. $\mathcal{T}$ iff C_0 is satisfiable w.r.t. $\mathcal{T}_0$.

Proof Sketch: ($\rightarrow$) Let $\mathcal{I}$ be a model of C and $\mathcal{T}$. We construct the interpretation $\mathcal{J}$ as follows

$$\begin{array}{rcl} \Delta^{\mathcal{J}} & = & \Delta^{\mathcal{I}}, \\ A^{\mathcal{J}} & = & A^{\mathcal{I}} \text{ for all } A \in N_C, \\ r^{\mathcal{J}} & = & r^{\mathcal{I}} \text{ for all } r \in N_R, \\ (\neg A')^{\mathcal{J}} & = & (\neg D)^{\mathcal{I}}. \end{array}$$

We have to prove that for all concepts C' and for all $d \in \Delta^{\mathcal{I}}$

$$d \in C'^{\mathcal{I}} \text{ iff } d \in C'^{\mathcal{J}}.$$

We can show the last statement by induction on the structure of C'.

We have to show that $\mathcal{J}$ is a model of $\mathcal{T}_0$. Then, we have to prove that $\mathcal{J}$ is a model of every GCI in $\mathcal{T}_0$. First, we show that $\mathcal{J}$ is a model of the GCIs added at point 3, i.e., $\{A' \sqsubseteq D,\ D \sqsubseteq A'\}$. We must show that $d \in A'^{\mathcal{J}}$ implies $d \in D^{\mathcal{J}}$ and $d \in D^{\mathcal{J}}$ implies $d \in A'^{\mathcal{J}}$. We suppose that $d \in A'^{\mathcal{J}} = \Delta^{\mathcal{J}} \setminus (\neg A')^{\mathcal{J}} = \Delta^{\mathcal{I}} \setminus (\neg D)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus (\Delta^{\mathcal{I}} \setminus D^{\mathcal{I}}) = \Delta^{\mathcal{J}} \setminus (\Delta^{\mathcal{J}} \setminus D^{\mathcal{J}}) = \Delta^{\mathcal{J}} \setminus (\neg D)^{\mathcal{J}} = D^{\mathcal{J}}$. Because of the statement above $\mathcal{J}$ is a model of the rest of the GCIs in $\mathcal{T}_0$.

($\leftarrow$) Analogously. □

($\mathcal{ALC}_1 \rightarrow \mathcal{ALC}_2$) We ensure that negation does not occur at all (except for $\bot$, which abbreviates $\neg\top$), neither in C_0 nor in $\mathcal{T}_0$. For every concept $\neg A$,

1. we introduce a fresh concept name $\bar{A}$,
2. we replace every occurrence of $\neg A$ with $\bar{A}$,
3. we add $\top \sqsubseteq A \sqcup \bar{A}$ and $A \sqcap \bar{A} \sqsubseteq \bot$ to $\mathcal{T}_0$.

We denote the resulting concept by C_1 and the TBox by $\mathcal{T}_1$. We have to show that C_0 is satisfiable w.r.t. $\mathcal{T}_0$ iff C_1 is satisfiable w.r.t. $\mathcal{T}_1$.

Proof Sketch: ($\rightarrow$) Let $\mathcal{I}$ be a model of C_0 and $\mathcal{T}_0$. We construct the interpretation $\mathcal{J}$ as follows

$$\begin{array}{rcl} \Delta^{\mathcal{J}} &=& \Delta^{\mathcal{I}}, \\ A^{\mathcal{J}} &=& A^{\mathcal{I}} \text{ for all } A \in N_C, \\ r^{\mathcal{J}} &=& r^{\mathcal{I}} \text{ for all } r \in N_R, \\ \bar{A}^{\mathcal{J}} &=& (\neg A)^{\mathcal{I}}. \end{array}$$

We have to show that for all concepts D and for all $d \in \Delta^{\mathcal{I}}$

$$d \in D^{\mathcal{I}} \text{ iff } d \in D^{\mathcal{J}}.$$

We can show the last statement by induction on the structure of D.

We have to show that $\mathcal{J}$ is a model of $\mathcal{T}_1$. Then, we have to prove that $\mathcal{J}$ is a model of every GCI in $\mathcal{T}_1$. First, we show that $\mathcal{J}$ is a model of the GCIs added at point 3, i.e., $\{\top \sqsubseteq \bar{A} \sqcup A, \bar{A} \sqcap A \sqsubseteq \bot\}$. We must show that $d \in \top^{\mathcal{J}}$ implies $d \in (\bar{A} \sqcup A)^{\mathcal{J}}$ and $d \in (\bar{A} \sqcap A)^{\mathcal{J}}$ implies $d \in \bot^{\mathcal{J}}$. First, we prove the former. We suppose that $d \in \top^{\mathcal{J}} = \Delta^{\mathcal{J}} = \Delta^{\mathcal{I}} = A^{\mathcal{I}} \cup (\neg A)^{\mathcal{I}} = A^{\mathcal{J}} \cup \bar{A}^{\mathcal{J}} = (A \sqcup \bar{A})^{\mathcal{J}}$. Thus, $d \in (A \sqcup \bar{A})^{\mathcal{J}}$ Now, we prove the latter. We suppose that $d \in (A \sqcap \bar{A})^{\mathcal{J}} = A^{\mathcal{J}} \cap \bar{A}^{\mathcal{J}} = A^{\mathcal{I}} \cap (\neg A)^{\mathcal{I}} = \emptyset = \bot^{\mathcal{J}}$. Because of the statement above $\mathcal{J}$ is a model of the rest of the GCIs in $\mathcal{T}_1$.

($\leftarrow$) Analogously. □

($\mathcal{ALC}_2 \rightarrow \mathcal{ALC}_3$) We ensure that disjunction does not occur at all in C_1. For every concept $D_1 \sqcup D_2$ in C_1

1. we introduce a fresh concept name A^*,
2. we replace $D_1 \sqcup D_2$ with A^*,
3. we add $A^* \sqsubseteq D_1 \sqcup D_2$ and $D_1 \sqcup D_2 \sqsubseteq A^*$ to $\mathcal{T}_1$.

We denote the resulting concept by C_2 and the TBox by $\mathcal{T}_2$. We have to show that, C_1 is satisfiable w.r.t. $\mathcal{T}_1$ iff C_2 is satisfiable w.r.t. $\mathcal{T}_2$.

The proof is similar to the proofs above.

($\mathcal{ALC}_3 \rightarrow \mathbf{CTL}_{\mathcal{EL}_\bot}$) We ensure that disjunction $\sqcup$ does not occur at all in $\mathcal{T}_2$. We assume that the only occurrences of disjunction $\sqcup$ in $\mathcal{T}_2$ are of the form

$$\begin{array}{rl} (i) & A \sqcup B \sqsubseteq D \\ (ii) & D \sqsubseteq A \sqcup B \end{array}$$

where A, B are concept names and D is disjunction free.

1. We replace (i) in $\mathcal{T}_2$ by $A \sqcap M \sqsubseteq D$ and $B \sqcap M \sqsubseteq D$.
2. We replace (ii) in $\mathcal{T}_2$ with the following GCIs

$$(a) \quad M \sqcap D \sqsubseteq \mathsf{A}\Diamond X \sqcap \mathsf{A}\Diamond Y,$$

$$(b) \quad M \sqcap D \sqcap \mathsf{E}\Diamond(X \sqcap \mathsf{E}\Diamond Y) \sqsubseteq A,$$

$$(c) \quad M \sqcap D \sqcap \mathsf{E}\Diamond(Y \sqcap \mathsf{E}\Diamond X) \sqsubseteq A,$$

$$(d) \quad M \sqcap D \sqcap \mathsf{E}\Diamond(X \sqcap Y) \sqsubseteq B,$$

   where M, X and Y are fresh concept names (for each $D \sqsubseteq A \sqcup B$).
3. We replace every subconcept $\exists r.E$ of C_2 with $\exists r.(E \sqcap M)$.
4. For every GCI $C \sqsubseteq D \in \mathcal{T}_2$, we replace C with $C \sqcap M$ and every subconcept $\exists r.E$ of D with $\exists r.(E \sqcap M)$.

We denote the resulting concept by C_3 and the TBox by $\mathcal{T}_3$. We have to show that C_2 is satisfiable w.r.t. $\mathcal{T}_2$ iff $C_3 \sqcap M$ is satisfiable w.r.t. $\mathcal{T}_3$.

Proof: ($\rightarrow$) Let $\mathcal{I}$ be a model of C_2 and $\mathcal{T}_2$. We construct a temporal interpretation $\mathfrak{J} = (S, <, I)$ as follows

- $(S, <)$ is a tree.
- For all $s \in S$, we define $I(s)$ as follows

$$\begin{array}{rcl} \Delta^{I(s)} & = & \Delta^{\mathcal{I}}, \\ A^{I(s)} & = & A^{\mathcal{I}} \text{ for all } A \in N_C \setminus \{M, X, Y\}, \\ r^{I(s)} & = & r^{\mathcal{I}} \text{ for all } r \in N_R. \end{array}$$

  We interpret M, X, Y as follows,

If s **is the root of** $(S,<)$, then

- $M^{I(s)} = \top^{I(s)}$.

  Let $d \in D^{\mathcal{I}}$. Then, by the GCI $D \sqsubseteq A \sqcup B$, $d \in (A \sqcup B)^{\mathcal{I}}$.

- If $d \in B^{\mathcal{I}}$, then $d \in X^{I(s')}$ and $d \in Y^{I(s')}$ for all $s' > s$.
- If $d \in A^{\mathcal{I}} \setminus B^{\mathcal{I}}$, then $d \in X^{I(s')}$ and $d \in Y^{I(s'')}$ for all $s'' > s' > s$.

If s **is not the root of** $(S,<)$, then

- $M^{I(s)} = \emptyset$
- $X^{I(s'')} = Y^{I(s'')} = \emptyset$ for all $s < s' < s''$.

We have to prove that for all concepts C' and for all $d \in \Delta^{I(s_0)}$

$$d \in C'^{\mathcal{I}} \text{ iff } d \in C'^{I(s_0)} \qquad (*)$$

where s_0 is the root of $(S,<)$.

We can prove the last statement by induction on the structure of C'. For the proof, note that X, Y do not occur in all C' and $M^{I(s_0)} = \top^{I(s_0)}$. Now, since $M^{I(s_0)} = \top^{I(s_0)}$, $C_3 \sqcap M$ is satisfiable.

It remains to show that $\mathfrak{J}$ is a model of $\mathcal{T}_3$. Then, we have to prove that $\mathfrak{J}$ is a model of every GCI in $\mathcal{T}_3$. First, we prove that $\mathfrak{J}$ is a model of the GCIs added at points 1 and 2.

(i) We have to prove that $d \in (A \sqcap M)^{\mathfrak{J}}$ implies $d \in D^{\mathfrak{J}}$ and $d \in (B \sqcap M)^{\mathfrak{J}}$ implies $d \in D^{\mathfrak{J}}$. First, we prove the former. Let $d \in (A \sqcap M)^{I(s)}$ with s the root of $(S,<)$. Since $M^{I(s)} = \top^{I(s)}$, $(A \sqcap M)^{I(s)} = A^{\mathcal{I}}$. Then, $d \in (A \sqcup B)^{\mathcal{I}}$. By the GCI $A \sqcup B \sqsubseteq D$, $d \in D^{\mathcal{I}} = D^{I(s)}$. Analogously for the latter.

(ii) (a) Let $d \in (M \sqcap D)^{I(s)}$ with s the root of $(S,<)$. Since $M^{I(s)} = \top^{I(s)}$, $(M \sqcap D)^{I(s)} = D^{\mathcal{I}}$. Then, by the GCI $D \sqsubseteq A \sqcup B$, $d \in (A \sqcup B)^{\mathcal{I}}$. We can distinguish two cases.

- If $d \in B^{\mathcal{I}}$, then $d \in X^{I(s')}$ and $d \in Y^{I(s')}$ for all $s' > s$. Therefore, $d \in (\mathbf{A}\Diamond X)^{I(s)}$ and $d \in (\mathbf{A}\Diamond Y)^{I(s)}$. Hence, $d \in (\mathbf{A}\Diamond X \sqcap \mathbf{A}\Diamond Y)^{I(s)}$
- If $d \in A^{\mathcal{I}} \setminus B^{\mathcal{I}}$, then $d \in X^{I(s')}$ and $d \in Y^{I(s'')}$ for all $s'' > s' > s$. Therefore, $d \in (\mathbf{A}\Diamond X)^{I(s)}$ and $d \in (\mathbf{A}\Diamond Y)^{I(s)}$. Hence, $d \in (\mathbf{A}\Diamond X \sqcap \mathbf{A}\Diamond Y)^{I(s)}$

(b) Let $d \in (M \sqcap D \sqcap \mathsf{E}\Diamond(X \sqcap \mathsf{E}\Diamond Y))^{I(s)}$ with s the root of $(S,<)$. Thus, $d \in (M^{I(s)} \cap D^{I(s)} \cap \mathsf{E}\Diamond(X \sqcap \mathsf{E}\Diamond Y)^{I(s)})$. Since $M^{I(s)} = \top^{I(s)}$, $d \in (D^{I(s)} \cap \mathsf{E}\Diamond(X \sqcap \mathsf{E}\Diamond Y)^{I(s)})$. Then, by the GCI $D \sqsubseteq A \sqcup B$, $d \in ((A \sqcup B)^{\mathcal{I}} \cap \mathsf{E}\Diamond(X \sqcup \mathsf{E}\Diamond Y)^{I(s)})$. Hence, there is a $s'' > s' > s$ such that $d \in X^{I(s')}$ and $d \in Y^{I(s'')}$, and $d \in (A \sqcup B)^{\mathcal{I}}$. Then, by construction, $d \in A^{\mathcal{I}} \setminus B^{\mathcal{I}} = A^{\mathcal{I}} = A^{I(s)}$.

(c) Let $d \in (M \sqcap D \sqcap \mathsf{E}\Diamond(Y \sqcap \mathsf{E}\Diamond X))^{I(s)}$ with s the root of $(S,<)$. Thus, $d \in (M^{I(s)} \cap D^{I(s)} \cap \mathsf{E}\Diamond(Y \sqcap \mathsf{E}\Diamond X)^{I(s)})$. Since $M^{I(s)} = \top^{I(s)}$, $d \in (D^{I(s)} \cap \mathsf{E}\Diamond(Y \sqcap \mathsf{E}\Diamond X)^{I(s)})$. Then, by the GCI $D \sqsubseteq A \sqcup B$, $d \in ((A \sqcup B)^{\mathcal{I}} \cap \mathsf{E}\Diamond(Y \sqcup \mathsf{E}\Diamond X)^{I(s)})$. Hence, there are $s'' > s' > s$ such that $d \in Y^{I(s')}$ and $d \in X^{I(s'')}$, and $d \in (A \sqcup B)^{\mathcal{I}}$. Thus, no such d exists.

(d) Let $d \in (M \sqcap D \sqcap \mathsf{E}\Diamond(X \sqcap Y))^{I(s)}$ with s the root of $(S,<)$. Thus, $d \in (M^{I(s)} \cap D^{\mathfrak{I}(\mathfrak{s})} \cap \mathsf{E}\Diamond(X \sqcap Y)^{I(s)})$. Since $M^{I(s)} = \top^{I(s)}$, $d \in (D^{I(s)} \cap \mathsf{E}\Diamond(X \sqcap Y)^{I(s)}$. Then, by the GCI $D \sqsubseteq A \sqcup B$, $d \in ((A \sqcup B)^{\mathcal{I}} \cap \mathsf{E}\Diamond(X \sqcap Y)^{I(s)})$. Hence, there is a $s' > s$ such that $d \in Y^{I(s')}$ and $d \in X^{I(s'')}$, and $d \in (A \sqcup B)^{\mathcal{I}}$. Then, by construction, $d \in B^{\mathcal{I}} = d \in B^{I(s)}$.

By the statement $(*)$ and the fact that $M = \top^{I(s)}$, $I(s)$ is a model of the rest of the GCIs in $\mathcal{T}_3$.

If s is not the root, then since $M^{I(s)} = \emptyset$ and M appears intersecting in the lefthand side of each GCI in $\mathcal{T}_3$, $I(s)$ is a model of $\mathcal{T}_3$.

Therefore, $\mathfrak{I}$ is a model of $\mathcal{T}_3$.

$(\leftarrow)$ Let $\mathfrak{I} = (S, <, I)$ be a model of $C_3 \sqcap M$ and $\mathcal{T}_3$. We construct the interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ as follows

$$\begin{aligned} \Delta^{\mathcal{I}} &= M^{I(s_0)}, \\ A^{\mathcal{I}} &= A^{I(s_0)} \cap \Delta^{\mathcal{I}} \text{ for all } A \in N_C \setminus \{X, Y\}, \\ r^{\mathcal{I}} &= r^{I(s_0)} \cap \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \text{ for all } r \in N_R, \end{aligned}$$

where s_0 is the root of $(S,<)$.

We have to prove that for all $\mathcal{ALC}$ concepts C' and for all $d \in \Delta^{I(s_0)}$

$$d \in \Delta^{I(s_0)} \text{ implies } d \in \Delta^{\mathcal{I}} \,(*), \text{ and}$$

$$d \in (C' \sqcap M)^{I(s_0)} \text{ implies } d \in C'^{\mathcal{I}} \; (**).$$

Observe that all concepts occurring in C_3 and $\mathcal{T}_3$ are $\mathcal{ALC}$ concepts. We can prove the last statement by structural induction.

Now, it remains to prove that $\mathcal{I}$ is a model of $\mathcal{T}_2$. First, we prove that $\mathcal{I}$ is a model of (i) and (ii). We have to show that $d \in (A \sqcup B)^{\mathcal{I}}$ implies $d \in D^{\mathcal{I}}$ and $d \in D^{\mathcal{I}}$ implies $d \in (A \sqcup B)^{\mathcal{I}}$. First, we prove the former. $d \in (A \sqcup B)^{\mathcal{I}} = ((A \sqcup B) \sqcap M)^{I(s_0)}$. Then, $d \in (A \sqcap M)^{I(s_0)}$ or $d \in (B \sqcap M)^{I(s_0)}$. By the GCI $A \sqcap M \sqsubseteq D$, $d \in (A \sqcap M)^{I(s_0)}$ implies $d \in D^{I(s_0)}$. Then, $d \in D^{\mathcal{I}}$, and by the GCI $B \sqcap M \sqsubseteq D$, $d \in (B \sqcap M)^{I(s_0)}$ implies $d \in D^{I(s_0)}$. Then, $d \in D^{\mathcal{I}}$.

Now, we prove the latter. Let $d \in D^{\mathcal{I}} = (D \sqcap M)^{I(s_0)}$. Then, by the GCI $D \sqcap M \sqsubseteq \mathsf{A}\Diamond X \sqcap \mathsf{A}\Diamond Y$, $d \in (\mathsf{A}\Diamond X \sqcap \mathsf{A}\Diamond Y)^{I(s_0)}$. Hence, for all $s_0 < s_1 < s_2 \ldots$ there are $k, k' > 0$ such that $d \in X^{I(s_k)}$ and $d \in Y^{I(s_{k'})}$. We can distinguish three cases, $k = k'$, $k > k'$ or $k' > k$.

- If $k = k'$, there is a $s_0 < s_1 < s_2 \ldots$ such that $X^{I(s_k)}$ and $Y^{I(s_k)}$. Then, $d \in \mathsf{E}\Diamond(Y \sqcap X)^{I(s_0)}$.
- If $k > k'$, there is a $s_0 < s_1 < s_2 \ldots$ such that $X^{I(s_k)}$ and $Y^{I(s_{k'})}$. Then, $d \in \mathsf{E}\Diamond(X \sqcap \mathsf{E}\Diamond Y)^{I(s_0)}$.
- If $k' > k$, there is a $s_0 < s_1 < s_2 \ldots$ such that $X^{I(s_k)}$ and $Y^{I(s_{k'})}$. Then, $d \in \mathsf{E}\Diamond(Y \sqcap \mathsf{E}\Diamond X)^{I(s_0)}$.

Therefore, $d \in (\mathsf{A}\Diamond X \sqcap \mathsf{A}\Diamond Y)^{I(s_0)} = \mathsf{E}\Diamond(X \sqcap Y)^{I(s_0)} \cup \mathsf{E}\Diamond(X \sqcap \mathsf{E}\Diamond Y)^{I(s_0)} \cup \mathsf{E}\Diamond(Y \sqcap \mathsf{E}\Diamond X)^{I(s_0)}$. Then, by the GCIs (a)–(d), $d \in (A^{I(s_0)} \cup B^{I(s_0)}) = (A \sqcup B)^{I(s_0)}$. Therefore $(A \sqcup B)^{\mathcal{I}}$.

By the statement $(**)$, $\mathcal{I}$ is a model of the rest of the GCIs in $\mathcal{T}_2$. □

Now, we can reduce satisfiability in $\text{CTL}_{\mathcal{EL}_\bot}$ to subsumption in $\text{CTL}_{\mathcal{EL}}$.

($\mathbf{CTL}_{\mathcal{EL}_\bot} \rightarrow \mathbf{CTL}_{\mathcal{EL}}$) We ensure that $\bot$ does not occur at all, neither in C_3 nor in $\mathcal{T}_3$.

1. We introduce a fresh concept name L.
2. We replace every occurrence of $\bot$ with L.
3. We extend $\mathcal{T}_3$ with $(a)\, \exists r.L \sqsubseteq L$ for every role from C_3 and $\mathcal{T}'$.

4. We add the following GCI to $\mathcal{T}_3$ (b) $\mathsf{E}\Diamond L \sqsubseteq L$.

We denote the resulting concept by C_4 and the TBox by $\mathcal{T}_4$. We have to show that C_3 is satisfiable w.r.t. $\mathcal{T}_3$ iff $C_4 \not\sqsubseteq_{\mathcal{T}_4} L$.

Proof: $(\rightarrow)$ Let $\mathfrak{I} = (S, <, I)$ be a model of C_3 and $\mathcal{T}_3$. We construct the interpretation $\mathfrak{J} = (S, <, J)$ as follows

- $(S, <)$ is a tree.
- For $s \in S$, we define $J(s)$ as follows

$$\begin{array}{rcl} \Delta^{J(s)} & = & \Delta^{I(s)}, \\ A^{J(s)} & = & A^{I(s)} \text{ for all } A \in N_C \setminus \{L\}, \\ r^{J(s)} & = & r^{I(s)} \text{ for all } r \in N_R, \\ L^{J(s)} & = & \bot^{I(s)}. \end{array}$$

We have to prove that for all concepts D and for all $d \in \Delta^{\mathfrak{I}}$

$$d \in D^{\mathfrak{J}} \text{ iff } d \in D^{\mathfrak{I}} \qquad (*)$$

We can prove the last statement by induction on the structure of D.

Then, by the previous statement, ${C_4}^{\mathfrak{J}} \neq \emptyset$ and $L^{\mathfrak{J}} = \emptyset$. Therefore, ${C_4}^{\mathfrak{J}} \not\subseteq L^{\mathfrak{J}}$.

It remains to show that $\mathfrak{J}$ is a model of $\mathcal{T}_4$. First, we prove that $\mathfrak{J}$ is model of the GCIs added at points 3 and 4.

(a) Let $(s, d) \in \exists r.L$. Then, there is a $(s, e) \in L^{\mathfrak{J}} = \bot^{\mathfrak{J}} = \emptyset$ with $(s, d, e) \in r^{\mathfrak{J}}$. Thus, no such (s, d) exists.

(b) Let $(s, d) \in (\mathsf{E}\Diamond L)^{\mathfrak{J}}$. Then, there exists $s_0 < s_1 < s_2 \ldots$ with $s = s_0$ such that there is a $k > 0$: $(s_k, d) \in L^{\mathfrak{J}} = \bot^{\mathfrak{J}} = \emptyset$. Thus, no such (s_k, d) exists.

By the statement $(*)$ $\mathfrak{J}$ is a model of the rest of the GCIs in $\mathcal{T}_4$

$(\leftarrow)$ Consider the contrapositive. Then, C_3 is not satisfiable w.r.t. $\mathcal{T}_3$, i.e., for all models $\mathfrak{I}$ of $\mathcal{T}_3$, $C_3^{\mathfrak{I}} = \emptyset$.

Let $\mathfrak{I} = (S, <, I)$ be a model of $\mathcal{T}_3$. We construct an interpretation $\mathfrak{J} = (S, <, J)$ as follows

- $(S, <)$ is a tree.
- For $s \in S$, we define $J(s)$ as follows

$$\begin{array}{rcl} \Delta^{J(s)} & = & \Delta^{I(s)}, \\ A^{J(s)} & = & A^{I(s)} \text{ for all } A \in N_C \setminus \{L\}, \\ r^{J(s)} & = & r^{I(s)} \text{ for all } r \in N_R, \\ L^{J(s)} & = & \bot^{I(s)}. \end{array}$$

We have to prove that for all concepts D and for all $d \in \Delta^{\mathfrak{I}}$

$$d \in D^{\mathfrak{I}} \text{ iff } d \in D^{\mathfrak{J}}. \qquad (**)$$

We can prove the last statement by induction on the structure of D.

Then, by the last statement and since C_3 is not satisfiable w.r.t $\mathcal{T}_3$, $C_4^{\mathfrak{J}} = \emptyset$ and $L^{\mathfrak{J}} = \bot^{\mathfrak{J}} = \emptyset$. Therefore, $C_4^{\mathfrak{J}} \subseteq L^{\mathfrak{J}}$.

It remains to show that $\mathfrak{J}$ is a model of $\mathcal{T}_4$. First, we prove that $\mathfrak{J}$ is model of the GCIs added at points 3 and 4.

(a) Let $(s, d) \in \exists r.L$. Then, there is a $(s, e) \in L^{\mathfrak{J}} = \bot^{\mathfrak{J}} = \emptyset$ with $(s, d, e) \in r^{\mathfrak{J}}$. Thus, no such (s, d) exists.

(b) Let $(s, d) \in (\mathsf{E}\Diamond L)^{\mathfrak{J}}$. Then, there exists $s_0 < s_1 < s_2 \ldots$ with $s = s_0$ such that there is a $k > 0\ : (s_k, d) \in L^{\mathfrak{J}} = \bot^{\mathfrak{J}} = \emptyset$. Thus, no such (s_k, d) exists.
By the statement $(**)$ $\mathfrak{J}$ is a model of the rest of the GCIs in $\mathcal{T}_4$ $\square$

Chapter 5

Conclusions

This chapter summarizes the work we developed in this thesis and presents some possibilities of future work.

The main goal of this thesis was to investigate branching temporal extensions of description logics. To this aim, we decided to investigate the combination of description logics with the temporal branching logic computation tree logic (CTL). We obtained the temporal description logics (TDLs), $\mathrm{CTL}_{\mathcal{ALC}}$ and $\mathrm{CTL}_{\mathcal{EL}}$ from the combination of CTL with $\mathcal{ALC}$ and $\mathcal{EL}$, respectively. In $\mathrm{CTL}_{\mathcal{ALC}}$ and $\mathrm{CTL}_{\mathcal{EL}}$, we focused our attention on the temporal evolution of concepts. More specifically, we proved complexity results of reasoning in $\mathrm{CTL}_{\mathcal{ALC}}$ and $\mathrm{CTL}_{\mathcal{EL}}$. On the one hand, we showed that satisfiability w.r.t. TBoxes with expanding domains in $\mathrm{CTL}_{\mathcal{ALC}}$ is EXPTIME -complete. Therefore, we consider $\mathrm{CTL}_{\mathcal{ALC}}$ as computationally rather well-behaved, i.e., concept satisfiability is not harder than in the component logics. The key observation in the proof of the upper bound was the close relation of $\mathrm{CTL}_{\mathcal{ALC}}$ with the fusion of CTL and $\mathcal{ALC}$. Moreover, the previous observation allows to establish a connection between $\mathrm{CTL}_{\mathcal{ALC}}$ and the standard μ-calculus. Finally, we use the latter connection to determine the complexity of reasoning in $\mathrm{CTL}_{\mathcal{ALC}}$. On the other hand, we showed that concept subsumption in $\mathrm{CTL}_{\mathcal{EL}}$ is not tractable as in the case of pure $\mathcal{EL}$. More precisely, reasoning in $\mathrm{CTL}_{\mathcal{EL}}$ is EXPTIME-complete, i.e., it is equally complex to reason in $\mathrm{CTL}_{\mathcal{ALC}}$. To obtain the lower bound, we reduced satisfiability in $\mathcal{ALC}$ to $\mathrm{CTL}_{\mathcal{EL}}$ subsumption.

As we have discussed, in the design of a temporal description logic there are several degrees of freedom. The previous fact, gives a wide spectrum of future work. In the immediate future, we can extend this work in the following ways. First, in the case of $\mathrm{CTL}_{\mathcal{ALC}}$, we can continue reasoning about the temporal evolution of concepts but in the "constant domain" case. Then, we can constrain to the case of "rigid roles" to increase the expressive power. Second, in the case of $\mathrm{CTL}_{\mathcal{EL}}$, we can look for polytime fragments.

Another important extension is to reason about the temporal evolution of axioms. For example, as discussed in (Baader *et al.*, 2008). We can also vary the DL component. For example, we can establish similar results for the *DL-Lite* family as in (Artale *et al.*, 2007). Finally, we can sketch some application scenarios in which the decision procedures could be optimized.

References

Artale, A., & Franconi, E. (1998). A temporal description logic for reasoning about actions and plans. *Journal of Artificial Intelligence Research*, 463–506.

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. (2005). Temporal description logics. *Pages 375–388 of: Handbook of Time and Temporal Reasoning in Artificial Intelligence.* Elsevier.

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., Sturm, H., & Wolter, F. (2002). Fusions of description logics and abstract description systems. *Journal of Artificial Intelligence Research*, **16**, 1–58.

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., & Patel-Schneider, P. F. (eds). (2003). *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press.

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

Blackburn, P., van Benthem, J. F. A. K., & Wolter, F. (2006). *Handbook of modal logic*. New York, NY, USA: Elsevier Science Inc.

Brachman, R. J., & Schmolze, J. (1985). An Overview of the KL-ONE Knowledge Representation System. *Cognitive science*, **9**(2), 171–216.

Brandt, S. (2004). Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and—what else? *Pages 298–302 of: Proceedings of the 16th European Conference on Artificial Intelligence (ECAI-2004)*. IOS Press.

Clarke, E. M., & Emerson, E. A. (1982). Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic. *Pages 52–71 of: Logic of programs, workshop*. London, UK: Springer-Verlag.

Emerson, E. A. (1990). Temporal and modal logic. *Pages 995–1072 of: Handbook of Theoretical Computer Science (vol. b): Formal Models and Semantics*. MIT Press.

Emerson, E. A. & Jutla, C. (1988). The complexity of tree automata and logics of programs. *Pages 368–377 of: Proceedings of the 29th IEEE Symposium of Foundations of Computer Science*. White Plains.

Gabbay, D., Hodkinson, I., & Reynolds, M. (1994). *Temporal Logic: Mathematical Foundations and Computational Aspects*. Vol. 1. Oxford University Press.

Gabbay, D., Reynolds, M., & Finger, M. (2000). *Temporal Logic: Mathematical Foundations and Computational Aspects*. Vol. 2. Oxford University Press.

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

Kozen, D. (1983). Results on the propositional μ- calculus. *Theoretical Computer Science*, **27**(3), 333–354.

Kozen, D., & Parikh, R. (1983). A decision procedure for the propositional μ-calculus. *Pages 313–325 of: Proceedings of the 2nd Workshop on Logic Programs*. LNCS, no. 164. Springer-Verlag.

Lamport, L. (1980). "sometime" is sometimes "not never": on the temporal logic of programs. *Pages 174–185 of: POPL '80: Proceedings of the 7th Symposium on Principles of Programming Languages*. ACM.

Lutz, C. (2004). Combining interval-based temporal reasoning with general TBoxes. *Artificial Intelligence*, **152**(2), 235–274.

Lutz, C., Sattler, U., & Wolter, F. (2001). Description logics and the two variable fragment. *Pages 66–75 of:* McGuiness, I D., Pater-Schneider, P., Goble, C., & Mller, R. (eds), *Proceedings of the 2001 International Workshop in Description Logics (DL'01).*

Lutz, C., Wolter, F., & Zakharyaschev, M. (2008). Temporal description logics: A survey. *Proceedings of the Fifteenth International Symposium on Temporal Representation and Reasoning.* IEEE Computer Society Press.

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

Pnueli, A. (1977). The temporal logic of programs. *Pages 46–57 of: Proceedings of the 18th IEEE Symposium on Foundations of Computer Science.*

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

Schild, K. (1991). A correspondence theory for terminological logics: preliminary report. *Pages 466–471 of: In proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91).*

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

Schild, K. (1994). Terminological cycles and the propositional μ-calculus. *Pages 509–520 of:* Doyle, P. T. J., & Sandewall, E. (eds), *Proceedings of the Fourth International Conference on Principles of Knowledge Representation and Reasoning (KR'94).* Morgan Kaufmann.

Schmidt-Schauß, M. (1989). Subsumption in KL-ONE is undecidable. *Page 21 of:* Brachman, R. J., Levesque, H. J., & Reiter, Raymond (eds), *Proceedings of the First International Conference on the Principles of Knowledge Representation and Reasoning (KR-89).* Morgan Kaufmann.

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

Schmiedel, A. (1990). Temporal terminological logic. Dietterich, W., & Swartout, T. (eds), *Proceedings of AAAI-90.* MIT press.

Vardi, M. Y. (1997). What makes modal logic so robustly decidable? *Pages 149–183 of: Descriptive Complexity and Finite Moldels*. American Mathematical Society.

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