Towards More Useful Description Logics of Time, Change and Context

Víctor Didier Gutiérrez Basulto

DISSERTATION

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften

(Dr. -Ing.)

Vorgelegt im Fachbereich 3 (Mathematik und Informatik)

der Universität Bremen

November 2013

Datum des Promotionskolloquiums:
15.11.13

Gutachter:
Prof. Dr. Carsten Lutz, Universität Bremen
Prof. Dr. Michael Zakharyaschev, Birkbeck, University of London

To my parents

Acknowledgments

First and foremost, I would like to thank my supervisor and teacher Carsten Lutz for several reasons. I am grateful to him for giving me the opportunity to join his group; undoubtedly, without his vision and expertise this thesis would not be what it is today. More importantly, I am grateful to him for making of me a ‘special forces’ pupil for this research-battle. I feel very lucky that I have seen him in action looking vehemently for nothing else but perfection in his research.

I also want to thank Michael Zakharyaschev for accepting to review my thesis; I feel honored of having him as a reviewer.

There is no doubt that the every day discussions and the small and big things I learned in daily basis are what made this thesis possible. It is because of this that I am greatly indebted to my colleague-friends Jean Christoph Jung and Szymon Klarman. Jean, thanks for never growing too annoyed with my sloppiness and for helping me so many times to bring to life the ideas living in my head. Szymon, thanks for that first paper together; it was my first one. More importantly, thanks for always being willing to start new research adventures with me. Jean, Szymon, I feel very lucky of having you as my co-authors and friends; without you my life as PhD student, and this thesis would not have been the same.

I would like to thank all the TDKIers for being the best colleagues one could ask for. I particularly thank you for all that time after the working hours; you made of my time in Bremen something special. I want to specially thank Thomas Schneider for several reasons. First, for finding the time to read some parts of my thesis, and giving me very useful feedback. More importantly, thanks for always being sincerely interested in my research, and for helping me disinterestedly in so many ways.

I am incommensurably grateful with my parents for all their love and for being simply the best parents of the world. With my brother, for being such an important part of my life; thanks for all those frenetic life-adventures. Family, thanks for your unconditional love and for being there in the difficult times.

Yazmín, I have no words to describe how thankful I am with you. I am grateful with you for so, so, so many things. I can only say that my time in Germany with you is by far the best time of my life. Thanks for being always so supportive and for believing in my work like nobody else does.

Abstract

Description Logics (DLs) are a family of logic-based formalisms for the representation of and reasoning about knowledge. Classical DLs are fragments of first-order logic and therefore aim at capturing *static* knowledge. Alas, the lack of means of DLs to capture dynamic aspects of knowledge has been often criticized because many important DL applications depend on this kind of knowledge. As a reaction to this shortcoming of DLs, two-dimensional extensions of DLs with capabilities to represent and reason about dynamic knowledge were introduced.

Two-dimesional DLs are constructed by combining DLs with well-known logic-based formalisms (e.g., modal logics) in the style of multi-dimensional modal logics, that is, we apply modal-like operators to different pieces of DL syntax. Clearly, two-dimensional DLs can model different dynamic aspects of knowledge depending on whether the modal-like operators are temporal, epistemic, etc. Notably, several two-dimensional DLs with different expressive power and different computational properties can be designed depending on the level of interaction between the component logics, e.g., depending on which pieces of DL syntax are enriched with modal-like operators.

In this thesis, we further the understanding and utility of two-dimensional DLs. We particularly focus on identifying two-dimensional DLs providing the right expressive power to model more accurately *temporal* and *contextual* aspects of knowledge required by certain DL applications, or providing better computational properties than other possible alternatives. With this in mind, we pursue three lines of research:

1. we study branching-time temporal DLs that emerge from the combination of classical DLs with the classical temporal logics CTL* and CTL;

2. we study description logics of change that emerge from the combination of classical DLs with the modal logic $\mathbf{S5}$;

3. we study description logics of context that emerge from the combination of classical DLs with multi-modal logics.

We particularly investigate temporal and contextual DLs based on the classical DL $\mathcal{ALC}$ and on members of the $\mathcal{EL}$-family of DLs. We moreover consider different levels of interaction between the component logics. Our main technical contributions are algorithms for satisfiability and subsumption, and (mostly) tight complexity bounds.

Zusammenfassung

Beschreibungslogiken (*description logics*, DLs) sind eine Familie logikbasierter Formalismen, mit deren Hilfe man Wissen repräsentieren und Schlussfolgerungen daraus ziehen kann. Klassische DLs sind Fragmente von Prädikatenlogik und zielen vor allem darauf ab, *statisches* Wissen zu modellieren. Die Tatsache, dass man mit DLs schlecht dynamische Aspekte von Wissen erfassen kann, wurde oft kritisiert, weil viele wichtige Anwendungen von DLs auf diese Art von Wissen angewiesen sind. Als eine Reaktion auf diesen Mangel von DLs wurden zweidimensionale Erweiterungen von DLs eingeführt, mit deren Hilfe man dynamisches Wissen repräsentieren und Schlussfolgerungen daraus ziehen kann.

Zweidimensionale DLs werden konstruiert, indem man DLs mit bekannten logikbasierten Formalismen (z. B. Modallogik) im Stile von mehrdimensionalen Modallogiken kombiniert. Das heißt, man wendet modale Operatoren auf verschiedene Teile der DL-Syntax an. Offenkundig können zweidimensionale DLs diverse dynamische Aspekte von Wissen modellieren, in Abhängigkeit davon, ob die modalen Operatoren temporal, epistemisch etc. aufgefasst werden. Insbesondere kann man verschiedene zweidimensionale DLs mit unterschiedlicher Ausdrucksstärke und unterschiedlichen Berechenbarkeitseigenschaften entwerfen, je nach angestrebtem Ausmaß der Interaktion zwischen den einzelnen Logikbestandteilen, z. B. indem man festlegt, welche Teile der DL-Syntax durch modale Operatoren erweitert werden.

Mit dieser Doktorarbeit bringen wir das Verständnis und die Benutzbarkeit zweidimensionaler DLs voran. Wir richten unsere Aufmerksamkeit vor allem darauf, zweidimensionale DLs zu identifizieren, die zum einen die richtige Ausdrucksstärke bieten um für bestimmte Anwendungen relevante *temporale* und *kontextuelle* Aspekte von Wissen zu modellieren, oder die andererseits bessere Berechenbarkeitseigenschaften haben als mögliche Alternativen. Um dieses Ziel zu erreichen, verfolgen wir drei Forschungsstränge:

1. Wir untersuchen temporale DLs mit verzweigender Zeitfolge, die man durch Kombination klassischer DLs mit den klassischen Temporallogiken CTL* und CTL erhält.

2. Wir untersuchen DLs zum Beschreiben von Veränderungen, die man durch Kombinationen klassischer DLs mit der Modallogik $\mathbf{S5}$ erhält.

3. Wir untersuchen kontextuelle DLs, die man durch Kombination klassischer DLs mit multimodalen Logiken erhält.

Wir untersuchen insbesondere temporale und kontextuelle DLs, die auf der klassischen DL $\mathcal{ALC}$ und den DLs in der $\mathcal{EL}$-Familie beruhen. Dabei betrachten wir verschiedene Ausmaße von Interaktion zwischen den Logikbestandteilen. Unsere fachlichen Hauptbeiträge sind Algorithmen für die Erfüllbarkeits- und Subsumptionsprobleme sowie (überwiegend scharfe) Komplexitätsschranken.

Contents

1
Introduction

Since the 1960's the area of *knowledge representation and reasoning (KR)* has played a key role in the development of artificial-intelligence systems that provide methods to adequately represent knowledge of a domain, and reasoning services to infer new knowledge from that stored in the system. One of the main challenges faced in KR research is the identification of adequate formalisms in the sense that they provide sufficient expressive power to faithfully capture the main aspects of an application domain while still permitting to efficiently infer, by means of automated reasoning services, new knowledge. Among the different approaches to KR, logic-based ones are probably the most popular since they provide systems with a precise syntax and semantics. In this thesis, we investigate *description logics (DLs)*: a prominent logic-based family of KR formalisms that offer enough expressive power for many application scenarios and at the same time efficient reasoning services. Notably, most DLs are close relatives of modal logics and therefore well-behaved fragments of first-order logic (FO); it is because of this relation with FO that DLs are well-suited to represent and reason about *static* knowledge. Alas, DLs are incapable to capture *dynamic* aspects of knowledge. The main objective of this thesis is to deepen the study of extensions of DLs that emerge from the combination of DLs with well-known logic-based formalisms in a multi-dimensional fashion allowing to capture various dynamic aspects of knowledge, such as time- or context-dependence of knowledge.

Description Logics

Description Logics (DLs) are a well-known family of knowledge representation formalisms used to structure and formally describe the terminology of an application domain [14]. Historically, DLs evolved from KR formalisms emerged in cognitive science, such as *sematic networks* [69] or *frames* [61]. These formalisms provide a well-defined way of representing knowledge by means of directed graphs or structured objects. However, they lack a formal semantics. As a consequence, the meaning associated to these graphs or structures depends on the implementation of the reasoning system. In order to overcome this drawback DLs were introduced in the 1980's as KR languages with a formal semantics.

The basic components of DLs are *concepts* (unary predicates) and *roles* (binary predicates). Naturally, which type of concepts and roles we can build depends on the constructors allowed by a particular DL language. For example, the classical DL $\mathcal{ALC}$ [71] offers the standard Boolean operators '$\sqcup$' (or), '$\sqcap$' (and) and '$\neg$' (not), and existential '$\exists$' and universal '$\forall$' restrictions. Using these constructors we can build, for example, the following concept modeling the term *father* from the family domain.

$$\mathsf{Human} \sqcap \mathsf{Male} \sqcap \exists \mathsf{hasChild}.\mathsf{Human}$$

As one would naturally expect, different applications of DLs require languages with different expressive power and efficiency of the reasoning tasks. With this in mind, expressive DL languages for which reasoning is hard but decidable and lightweight languages allowing for efficient reasoning while still providing enough expressive power for several applications have been introduced. In expressive languages, concepts and roles are built using not only $\mathcal{ALC}$-constructors but also using additional role and concept constructors such as *inverse roles*, *transitive roles*, *number restrictions*, etc. For example, for the role $\mathsf{hasChild}$, we can use its inverse role $\mathsf{hasChild}^-$ to model the relation '*is Child of*'. Among efficient DLs we find the tractable $\mathcal{EL}$, a sub-Boolean fragment of $\mathcal{ALC}$ allowing only for '$(\top)$' (top), '$\sqcap$' and '$\exists$'. Note that the above concept modeling the term father is an $\mathcal{EL}$-concept.

Description logics use a *knowledge base (KB)* or *ontology* to capture the background *terminological* knowledge about the nomenclature of the domain, and the *assertional* knowledge about concrete elements of the domain. The terminological knowledge of a KB is contained in a *terminological box (TBox)* stating the relation between concepts. Formally, a TBox is a finite set of axioms of the form $C \sqsubseteq D$; for example, we can state that all professors have a PhD using the axiom $\mathsf{Professor} \sqsubseteq \exists \mathsf{hasDegree}.\mathsf{PhD}$. The assertional knowledge is contained in an *assertional box (ABox)* capturing information about the concepts and roles that concrete individuals belong to. More precisely, an ABox is a finite set of axioms of the form $C(a)$ and $r(a, b)$; for example, we can state that Mary is a Human using the axiom $\mathsf{Human}(\mathsf{Mary})$, or we can state that Mary has a pet named Sparky using the axiom $\mathsf{hasPet}(\mathsf{Mary}, \mathsf{Sparky})$.

A DL system not only allows to store information in a KB about a domain, but it further permits to infer implicit consequences from the information stored in the KB. Description logics, differently from their predecessors, have a formal semantics given in terms of interpretations over a domain, which via an interpretation function $\cdot^{\mathcal{I}}$ assigns subsets of individuals of the domain to concepts and binary relations over the domain to roles. More precisely, an interpretation $\mathcal{I}$ is a pair $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$ is a domain and an interpretation function. A wide range of inference problems have been considered, central to this thesis are the following classical reasoning services:

- *satisfiability of a concept w.r.t. a TBox*: given a concept C and a TBox $\mathcal{T}$, decide whether C can have an instance in a model of $\mathcal{T}$. Intuitively, an interpretation $\mathcal{I}$ is a model of a TBox if it satisfies all axioms of the TBox.

– *KB satisfiability*: decide whether a KB can have a model, that is, whether there is an interpretation satisfying all the axioms of the KB.

– *subsumption of concepts w.r.t. a TBox*: given concepts C, D and a TBox $\mathcal{T}$, decide whether every instance of C is also an instance of D in every model of $\mathcal{T}$.

These problems play a key role towards the process of ontology development and design. For example, the subsumption problem can be used to induce from an ontology a concept-hierarchy which can be used by an ontology developer to make the structure of knowledge explicit. On the other hand, the concept satisfiability problem helps an ontology designer to check whether a concept makes sense in terms of an ontology.

Description logics have been successfully used in various areas of computer science, such as the semantic web [20] where DLs serve as the basis of the *Web Ontology Language OWL*, recommended by the W3C (World Wide Web Consortium) [66, 50, 16]; or in bioinformatics where DL languages are used to represent medical ontologies such as SNOMED CT [75] or the GeneOntology [1]. Despite the successful application of DLs in these areas, it has been recognized that a major limitation of DLs is that, as fragments of first-order logic, they are only well-suited for the representation of and reasoning about *static* knowledge, that is, they lack any means to capture *dynamic* aspects of knowledge like time- or action-dependence of knowledge. In particular, this limitation has become an important drawback for many relevant applications depending on this type of knowledge. For example, many terms in the medical domain are defined with reference to time such as the term "*concussion with no loss of consciousness*" found in SNOMED CT, which refers to a concussion after which the patient has remained conscious until the time of examination (indicating that the concussion is only mild) [72].

To tackle this shortcoming in the 1990s *two-dimensional* extensions of DLs designed to capture various dynamic aspects of knowledge were proposed [38, Chapter 3.8]. Two-dimensional DLs are constructed by combining DLs with *modal logics* [21] in the style of multi-dimensional modal logics [38], that is, they are constructed by applying modal operators $\Diamond_i, \Box_i$ to different pieces of DL syntax: roles, concepts or axioms. The semantics of two-dimensional DLs is given in terms of two-dimensional interpretations $\mathfrak{I} = (\Delta, W, \{\mathcal{I}_w\}_{w\in W})$ composed by a set of possible worlds (states, time points) W, such that each world $w \in W$ has associated a DL-model $\mathcal{I}_w$ with domain Δ representing the *current state of knowledge*. Naturally, the model associated to each world might change from one world to another, representing the change through time, the application of actions, etc. As an example, the concept $\Diamond_i C$ describes the class of individuals that are instances of C in some possible world accessible from the current one via the accessibility relation associated to $\Diamond_i$. Clearly, we can model different intensional aspects of knowledge depending on whether the operators are temporal, dynamic, epistemic or some other modal-like operators.

Due to the way two-dimensional DLs are constructed, they are also referred to as *modal description logics*. We refer to concepts of the form $\Diamond_i C$ and $\Box_i C$ as *modalized concepts* and to roles of the form $\Diamond_i r$ and $\Box_i r$ as *modalized roles*. Besides the type of modal-like operators used to construct modal DLs, there are some other degrees of freedom in the design of these logics: for example, two-dimensional interpretations $\mathfrak{I}$ introduced above make the so-called *constant domain assumption*, that is, all $\mathcal{I}_w$ share the same domain Δ. One could also assume *varying domains* by associating with each $\mathcal{I}_w$ a domain Δ^w, or *expanding domains* by requiring that $\Delta^w \subseteq \Delta^{w'}$ if w' is accessible from w through the accessibility relation associated to some $\Diamond_i$. Throughout this thesis, we use constant domains since for most of the considered two-dimensional DLs this is the most general option in the sense that varying and expanding domains can be simulated [38]. Further design choices in the definition of the semantics of two-dimensional DLs include the definition of different DL components, such as roles or concepts, as *rigid*, meaning that they do not vary their interpretation across the possible worlds. A DL symbol α is *rigid* in an interpretation $\mathfrak{I}$ if $\alpha^{\mathfrak{I},w} = \alpha^{\mathfrak{I},w'}$ for all $w, w' \in W$; otherwise, α is *local*. Moreover, we can also assume a TBox to be *global* in the sense that it has to be satisfied in every possible world.

For a general discussion on multi-dimensional DLs, please consult the following works by Wolter and Zakharyaschev [83, 82], and the book *Many-Dimensional Modal Logics: Theory and Applications* [38] by Gabbay *et al.*

1.1 Objectives and Motivation

Due to the crucial need of many applications to model various dynamic aspects of knowledge; for example, to model temporal aspects present in medical ontologies [15] or to capture temporal data models [8], many efforts have been devoted to the investigation of several multi-dimensional DLs. In particular, the flexibility on the design of multi-dimensional DLs has led to the construction of several logics with different expressive power and different computational properties. Naturally, the type of modal-like operators used in the construction of multi-dimensional DLs varies depending on the dynamic aspect to be modeled. The objective of this thesis is further the understanding and utility of several forms of dynamic DLs. We particularly focus on identifying two-dimensional description logics providing the right expressive power to model more accurately *temporal* and *contextual* aspects of knowledge required by certain ontology applications, or offering better computational properties than other possible alternatives.

In philosophy and computer science different notions of time have been considered. Often a distinction between a branching and a linear notion of time has been made: intuitively, under the linear-time notion there exists a unique possible future while under the branching-time one many possible futures are allowed. Not surprisingly, the research on *temporal DLs (TDLs)*, following this dichotomy, has considered linear-time and branching-time temporal description logics [58]. However, in contrast to some other areas of computer science such as specification and verification, the investigation of the computational properties of branching-time TDLs has been nearly disregarded. This comes as a surprise since in many applications terms refer to the existence of many *possible* futures: think, for example, that we are trying to model the term *European candidate*, which refers to '*a European country that in the future might join the EU*'; assuming a linear notion of time we can only express that a European candidate will *eventually* become part of the EU, excluding thus the possibility of a European candidate never joining the EU. On the other hand, under a branching notion of time a more accurate representation is obtained by stating that there is a possible future in which a European candidate eventually joins the EU, leaving open the possibility of other futures. Motivated then by the fact that in some applications we indeed need to refer to a branching notion of time, we investigate the computational complexity of various branching-time TDLs based on the branching-time temporal logics CTL and CTL*, introduced for specification and verification purposes by Clarke and Emerson [32], and Emerson and Halpern [34], respectively.

Another important challenge towards the design of more useful TDLs is the identification of TDLs with polytime reasoning. To this end, one can consider TDLs based on the tractable DL $\mathcal{EL}$. Notably, these TDLs can be used as a natural extension of $\mathcal{EL}$ for capturing temporal aspects occurring in medical ontologies such as the $\mathcal{EL}$-based SNOMED CT. Unfortunately, TDLs based on the linear-time temporal logic LTL and $\mathcal{EL}$ seem to be unsuitable to attain polytime reasoning. The reason is that to achieve polynomial time reasoning, the expressive power of the (T)DL under consideration has to be restricted such that no (explicit or implicit) *disjunction* remains in the language. Alas, disjunction is inherent in linear time as $\Diamond \mathsf{X} \sqcap \Diamond \mathsf{Y}$ implies that one of $\Diamond(\mathsf{X} \sqcap \Diamond \mathsf{Y})$, $\Diamond(\mathsf{Y} \sqcap \Diamond \mathsf{X})$ and $\Diamond(\mathsf{X} \sqcap \mathsf{Y})$ is true, where $\Diamond \mathsf{X}$ reads as *eventually* in the future X holds. A starting point to construct tractable branching-time TDLs is to consider TDLs based on $\mathcal{EL}$ in which *universal path quantification* has been dropped, which in principle avoids the implicit disjunctions above.

Continuing with the design of well-behaved TDLs, we consider TDLs with a weaker temporal dimension based on the modal logic $\mathbf{S5}$ instead of the branching-time temporal logics CTL and CTL*. These TDLs provide a weaker temporal dimension in the sense that they allow to reason about the changes of knowledge over time without differentiating between the changes in the past or future, that is, there is no 'directionality' in time . For example, we can express that '*every child evolves to an adult*', that is, all the instances of Child in the current world are instances of Adult in some possible world. Interestingly, having a weaker temporal dimension allows to design well-behaved TDLs with higher interaction between their component logics, that is, we can apply temporal operators to more pieces of DL-syntax than in branching-time TDLs. In particular, in contrast to TDLs based on traditional linear-time and branching-time temporal logics, one can construct effective TDLs based on $\mathbf{S5}$ and expressive DLs in the presence of temporal roles. This will allow us to model, for example, the term *accidental death* which refers to '*a death that has as a possible cause an accident*' with the concept $\mathsf{Death} \sqcap \exists \Diamond \mathsf{hasCause}.\mathsf{Accident}$ describing the class of individuals that are instances of Death in the current world and that in some possible world are related via $\mathsf{hasCause}$ to an individual in the extension of $\mathsf{Accident}$.

In 2009, Hendler and Berners-Lee [45] outlined some of the challenges towards the development of a new generation of more useful Web applications; specially, they pointed out the key role of the semantic web in the emergence of these technologies. Hendler and Berners-Lee particularly argued that the design of logic-based mechanisms to specify *contexts* is vital to properly reuse and integrate knowledge in an open and distributed environment like the Web. In other words, these KR mechanisms must be able to represent and take into account while reasoning the meta-information describing the situation under which a certain piece of knowledge is valid. With this in mind, different semantic web formalisms to capture contexts have been proposed [17, 73, 26]. However, up to now, none of them provides the semantic web with a generic and formal framework to treat contextualized knowledge. Motivated by this fact, we investigate the adequacy of two-dimensional DLs to capture contextual aspects of knowledge. In particular, we design two-dimensional DLs importing the well-known McCarthy's theory of formalizing contexts. We provide thus the semantic web with a formal and generic approach to specify contexts.

1.2 Related Work

Temporal Description Logics Since the appearance, in the early 90's, of the first research on temporal DLs, a lot of investigations have been conducted. One prominent approach to TDLs is to combine classical DLs with the standard temporal logics LTL, CTL or CTL* in the style of multi-dimensional modal logics. In particular, special efforts have been devoted to the investigation of TDLs based on LTL, so that now a fairly clear landscape of the computational complexity has emerged and several algorithmic approaches have been presented. Naturally, depending on the chosen DL and on the level of interaction between LTL and the DL component, these combinations have different expressive power and different computational properties. We next highlight some relevant results; please see the survey by Lutz *et al.* [58] for a detailed discussion. The research on linear-time TDLs began by considering combinations based on the classical DL $\mathcal{ALC}$. The two most basic results [58] are: first, an ExpTime Pratt-style type elimination algorithm for satisfiability in $\mathrm{LTL}_{\mathcal{ALC}}$ in the case where temporal operators are applied to concepts and a global TBox is considered. In particular, this result shows that reasoning can be reduced to the component logics. Second, a reduction of the $\mathbb{N}\times\mathbb{N}$-tiling problem showing that if we also allow for *rigid* roles satisfiability becomes undecidable. The case where temporal operators are applied not only to concepts but also to concept inclusions (no rigid or temporal roles allowed) has been also considered: Wolter *et al.* [46, 38] presented a tight ExpSpace algorithm for satisfiability based on the notion of *quasimodels* [38].

An orthogonal way of constructing linear-time TDLs is to disallow temporal concepts while permitting temporal axioms, more precisely, these logics are obtained by applying temporal operators to TBoxes and ABoxes, and additionally rigid roles or rigid concepts are allowed. Following this direction, Baader *et al.* [15] proved that KB satisfiability in $\mathrm{LTL}_{\mathcal{ALC}}$ is 2ExpTime-complete in the case where both rigid concepts and rigid roles are available. It is important to note that *without* temporal concepts it becomes impossible to enforce the matching conditions of adjacent tiles and then to properly encode a $\mathbb{N}\times\mathbb{N}$-tiling. They further showed that if *only* rigid concepts (but no rigid roles) are available then satisfiability becomes NExpTime-complete. This shows that to some extent the reasoning can be reduced to the component logics by an adequate management of rigid concepts, that is, a set of rigid concepts satisfied in a temporal model needs to be guessed. Finally, another prominent result is that if the only operators allowed are $\Diamond$ and $\Box$, and both rigid concepts and rigid roles are available, KB satisfiability then becomes ExpTime-complete.

From the investigations previously discussed, one can observe that these combinations become undecidable as soon as temporal concepts and rigid (or temporal) roles are allowed. In the view of this, TDLs based on LTL and lightweight DLs of the $\mathcal{EL}$- and *DL-Lite*-family [4] have been investigated. Note that the rather weak expressiveness of the DL component could in principle lead to the construction of effective TDLs allowing for rigid roles or temporal roles. Recall that some members of these families allow for tractable reasoning. Alas, for combinations based on $\mathcal{EL}$ the result is a negative one [7]: subsumption in $\mathrm{LTL}_{\mathcal{EL}}$ is undecidable in the case where temporal operators (with only the $\Diamond$ operator is enough) are applied to concepts, and rigid roles and a global TBox are considered. The main reason of this result is that this logic is *non-convex*, that is, disjunctive knowledge is reintroduce through the temporal operators. In particular, it has been shown that non-convex extensions of $\mathcal{EL}$ are as complex as the $\mathcal{ALC}$ variant. On the other hand, if members of the *DL-Lite* family are considered this type of combination turns out to be decidable [7]. Remarkably, KB satisfiability in LTL-*DL-Lite*$^{\mathcal{N}}_{bool}$ is EXPSPACE-complete in the case where temporal operators are applied to concepts, TBoxes and ABoxes, and rigid roles are available. After decidability was established, Artale *el al.* [8] made further efforts towards the identification of well-behaved variants. In particular, they demonstrated that by considering global TBoxes and a controlled version of temporal ABox axioms instead of temporal TBoxes and ABoxes, satisfiability of LTL-*DL-Lite*$^{\mathcal{N}}_{bool}$ KBs goes down from EXPSPACE-complete to PSPACE-complete.

For combinations based on the branching-time logics CTL and CTL* only few investigations have been conducted, mainly establishing decidability boundaries. Remarkably, Hodkinson *et al.* [48] showed in the context of monodic temporal first-order logic that TDLs based on CTL are typically decidable whereas TDLs based on CTL* have to be appropriately restricted in order to attain decidability. As a consequence of the investigation by Hodkinson *et al.* [48], we have that TDLs based on CTL and CTL* allowing for rigid or temporal roles are most probably undecidable. The work by Hodkinson *et al.* [48] provides only non-elementary upper complexity bounds leaving open then the establishment of tight complexity bounds.

Description Logics of Change Description logics of change are well-behaved temporal DLs characterized by admitting effective reasoning in the presence of temporal roles. DLs of change are constructed by combining classical DLs with the modal logic $\mathbf{S5}$ and support reasoning about changes of knowledge without differentiating between changes in the past or the future. The rather weak expressiveness of the temporal component (compared with that of traditional temporal logics) allows to construct effective DLs of change based on *expressive* DLs that allow the application of temporal operators to roles and concepts. In this direction, Artale *et al.* [10] presented a 2EXPTIME type-based algorithm for satisfiability in $\mathbf{S5}_{\mathcal{ALCQI}}$ in the case where temporal operators are applied to concepts and roles, and a global TBox is considered. Moreover, they show that 2EXPTIME hardness already holds when $\mathcal{ALC}$ is considered instead of $\mathcal{ALCQI}$. Notably, it has been shown [10, 11] that $\mathbf{S5}_{\mathcal{ALCQI}}$ captures a considerable fragment of *temporal entity-relationship models* used to design temporal databases. In another research effort, DLs of change based on the lightweight DL $\mathcal{EL}$ have been investigated. Remarkably, a tractable DL of change has been identified and therefore shown to be easier than its $\mathcal{ALC}$ variant. Specifically, Lutz *et al.* [57] provided a PTIME completion algorithm for concept subsumption in $\mathbf{S5}_{\mathcal{EL}}$ in the case where temporal operators are applied *only* to concepts and a global TBox is considered. Note that for the $\mathcal{ALC}$ variant an EXPTIME lower bound is inherited from $\mathcal{ALC}$. Another important result from this investigation by Lutz *et al.* is a PSPACE lower bound for concept subsumption in $\mathbf{S5}_{\mathcal{EL}}$ in the case where temporal operators are applied not only to concepts but also to roles.

One of the contributions of this thesis is the design of two-dimensional DLs enabling to model contextual aspects of knowledge. To the best of our knowledge, this is the first attempt to use multi-dimensional DLs to reason about contextualized knowledge. We next discuss some of the works on contexts in AI.

Contexts in Artificial Intelligence (AI) In AI many attempts have been made towards the development of adequate mechanisms to reason about contextualized knowledge, that is, establishing the validity of certain knowledge taking into account the *context* in which this knowledge was generated, see [2] for a survey. An inherent problem of contextualization is what to consider as context, or how to define a context. Intuitively, a context has been understood as the relevant (meta) information describing the situation in which certain knowledge is valid. As one can note the intuitive notion of context is very vague, and it might vary from one application to another. An important theory of formalizing contexts is that of McCarthy's (discussed in length in Section 5.2), characterized by providing a generic (application-free) definition of context: a *context* is a formal object $\boldsymbol{c}$ use in assertions of the form $ist(\boldsymbol{c}, \varphi)$, stating that the formula φ is true in the context $\boldsymbol{c}$. This theory is therefore not interested in an exact definition of context but rather in defining how to use it as a first-class citizen in knowledge-based systems.

McCarthy's theory has been translated into a number of logic systems [30, 28, 64] based on modal logics such as $\mathbf{K}_n, \mathbf{DAlt}_n, \mathbf{Alt}_n$ or $\mathbf{D}_n$. For instance, in the *propositional logic of context* [30] an assertion $ist(\boldsymbol{c}, \varphi)$ can be restated as a modal formula $\Box_{\boldsymbol{c}}\varphi$, where the behavior of $\Box_{\boldsymbol{c}}$ is suitably axiomatized in order to capture possibly many context-based operations, e.g., entering and exiting contexts, lifting knowledge from one context to another, etc. Another dominant tradition in the field originates from the paradigm of *multi-context logics* (MCLs), introduced by Giunchiglia *et al.* [40, 39], where the main objective is to provide mechanisms of bridging multiple local representations. These two perspectives on operationalizing contexts in knowledge systems can be further linked to two areas of research within the field of DLs: two-dimensional DLs and logic-based ontology integration. Two-dimensional DLs can be thought as formalisms capturing the dependency of knowledge in some 'built-in' context-states, such as time points. The area of logic-based ontology integration focuses on the problem of integrating knowledge contained in multiple, independent sources (DL-based ontologies). Among many existing solutions there are Package-based DLs [18], Distributed DLs [24], $\mathcal{E}$-Connections [55], semantic imports [67], and others. Each offers a formal mechanism of relating the vocabularies belonging to different sources by means of certain semantic relations for linking models of the respective ontologies.

Finally, we acknowledge the substantial work by Homola and Serafini [73] in the DL paradigm, proposing the framework of *Contextualized Knowledge Repositories*. In particular, their proposal incorporates the following features: DL-based representation of object knowledge, contexts as formal objects, a mechanism of knowledge integration, meta-level descriptions of contexts.

1.3 Results

In this thesis, we focus on the study of two-dimensional DLs modeling branching-time *temporal*, *evolutionary* and *contextual* aspects of knowledge. In particular, we study the impact of varying the DL component and the degree of interaction between the component logics on the computational complexity. The prime contributions are algorithms for satisfiability and subsumption, and tight complexity bounds. Specifically, we identify the following as our prime technical contributions.

TEMPORAL DESCRIPTION LOGICS:

- A uniform approach based on a combination of Pratt-style type elimination and methods based on non-deterministic automata over infinite trees for satisfiability for $\mathrm{CTL}_{\mathcal{ALC}}$ and $\mathrm{CTL}^*_{\mathcal{ALC}}$ in the case where temporal operators are applied to concepts and a global TBox is considered.

- A uniform approach based on a combination of nondeterministic automata and two-way alternating over infinite trees for satisfiability for $\mathrm{CTL}_{\mathcal{ALC}}$ and $\mathrm{CTL}^*_{\mathcal{ALC}}$ in the case where temporal operators are applied to concepts and to concept inclusions.

- A fairly complete landscape of the computational complexity of fragments of $\mathrm{CTL}_{\mathcal{EL}}$ in the case where temporal operators are applied only to concepts and a global TBox is considered. Notably, a tractable fragment based on the temporal operator $\mathbf{E}\Diamond$ is identified.

- Mostly tight elementary complexity bounds that range from PTIME to 2EXPTIME and 3EXPTIME.

DESCRIPTION LOGICS OF CHANGE

- A NEXPTIME lower bound for satisfiability in $\mathbf{S5}_{\mathcal{ALCO}}$ in the case where modalities are applied to concepts and a global TBox is considered. We show this by a reduction of the $2^n \times 2^n$-tiling problem.

- A PSPACE completion algorithm for subsumption in $\mathbf{S5}_{\mathcal{EL}}$ in the case where modalities are applied to concepts and roles, and a global TBox is considered.

- A 2EXPTIME lower bound for subsumption in $\mathbf{S5}_{\mathcal{ELI}}$ in the case where modalities are applied to concepts and roles, and a global TBox is considered. We demonstrate this by a reduction of the word problem of exponentially space bounded Turing machines.

- Tight complexity bounds that range from PSPACE to NEXPTIME and 2EXPTIME.

DESCRIPTION LOGICS OF CONTEXT

- A stepwise integration of McCarthy's theory of contexts into the DL paradigm through two-dimensional DLs.

- First, we consider the modal DLs $(\mathbf{K}_n)_{\mathcal{ALC}}, (\mathbf{DAlt}_n)_{\mathcal{ALC}}, (\mathbf{Alt}_n)_{\mathcal{ALC}}, (\mathbf{D}_n)_{\mathcal{ALC}}$ in the case where modalities are applied to concepts and a global TBox is considered. For these logics we demonstrate a 2EXPTIME lower bound for satisfiability by a reduction of the word problem of exponentially space bounded Turing machines.

- A 2EXPTIME algorithm based on a variant of quasistate elimination techniques for satisfiability in $(\mathbf{K}_n)_{\mathcal{ALC}}, (\mathbf{DAlt}_n)_{\mathcal{ALC}}, (\mathbf{Alt}_n)_{\mathcal{ALC}}, (\mathbf{D}_n)_{\mathcal{ALC}}$.

– Second, we introduce of a two-dimensional *two-sorted* family of DLs implementing full McCarthy's postulates into the DL paradigm.

– We provide tight complexity bounds for these expressive variants ranging from NEXPTIME to 2EXPTIME.

1.4 Structure of the Thesis

Apart from the preliminaries and the conclusions, this thesis contains three main chapters with the results of our investigation on two-dimensional branching-time temporal description logics (CHAPTER 3), description logics of change (CHAPTER 4), and description logics of context (CHAPTER 5).

– CHAPTER 2. We introduce basic notions and results in DLs and branching-time temporal logics. In particular, we present the syntax and semantics of these logics and further discuss the complexity of their reasoning problems.

– CHAPTER 3. We investigate two-dimensional DLs for representing and reasoning about *temporal* aspects of knowledge. *Branching temporal* DLs are constructed by combining the temporal logics CTL and CTL* with classical DLs in the style of multi-dimensional DLs. We concentrate on the study of the computational complexity of combinations, based on the classical DLs $\mathcal{ALC}$ and $\mathcal{EL}$. First, we present algorithms for satisfiability for CTL$_{\mathcal{ALC}}$ and CTL$^*_{\mathcal{ALC}}$ in the case where temporal operators are applied only to concepts and a global TBox is considered, yielding tight EXPTIME and 2EXPTIME upper bounds. Later, we continue our research on temporal concepts by considering *fragments* of CTL$_{\mathcal{EL}}$: we identify a tractable fragment based on the temporal operator $\mathbf{E}\Diamond$, and further show that most of the other fragments are hard for PSPACE and EXPTIME. Finally, we reconsider CTL$_{\mathcal{ALC}}$ and CTL$^*_{\mathcal{ALC}}$, now in the case where temporal operators are applied to concepts and concept inclusions. We obtain 2EXPTIME and 3EXPTIME upper bounds for CTL$_{\mathcal{ALC}}$ temporal TBoxes and CTL$^*_{\mathcal{ALC}}$ temporal TBoxes, respectively. For CTL$_{\mathcal{ALC}}$ temporal TBoxes we prove a matching 2EXPTIME lower bound.

– CHAPTER 4. We investigate two-dimensional DLs for representing and reasoning about *changes* of knowledge. DLs of *change* are constructed by combining the modal logic $\mathbf{S5}$ with classical DLs in the style of multi-dimensional DLs. We concentrate on the study of the computational complexity of combinations based on the classical DLs $\mathcal{ALCO}$, $\mathcal{EL}$ and $\mathcal{ELI}$. First, we consider $\mathbf{S5}_{\mathcal{ALCO}}$ in the case where $\mathbf{S5}$-modalities are applied only to concepts and a global TBox is considered. We show that satisfiability for this logic is NEXPTIME-complete. Later, we investigate $\mathbf{S5}_{\mathcal{EL}}$ in the case where $\mathbf{S5}$-modalities are applied not only to concepts but also to roles, and a global TBox is considered. We develop a completion algorithm for subsumption, yielding a PSPACE tight upper bound. Finally, we demonstrate that for $\mathbf{S5}_{\mathcal{ELI}}$ subsumption becomes 2EXPTIME-hard.

– CHAPTER 5. We investigate two-dimensional DLs for representing and reasoning about *contextualized* knowledge. We begin by discussing the use of two-dimensional DLs to faithfully import McCarthy's theory of contexts into the DL paradigm. Our investigation of DLs of *context* is two-fold: first, we investigate DLs of context constructed by combining the multi-modal logics $\mathbf{K}_n, \mathbf{DAlt}_n, \mathbf{Alt}_n, \mathbf{D}_n$ with the classical DL $\mathcal{ALC}$ in the style multi-dimensional DLs. We concentrate on the study of the computational complexity of combinations where modal operators are applied only to concepts and a global TBox is considered. We demonstrate that satisfiability for these logics is 2EXPTIME-complete. Second, we construct expressive DLs of context importing all McCarthy's postulates into the DL paradigm. In particular, these logics contain two interacting DL languages for explicitly modeling both the (contextualized) object-level knowledge and the meta-level, describing properties of contexts. Notably, we show that reasoning in these expressive languages is also 2EXPTIME-complete. We moreover show the relation between the modal DLs discussed above and these expressive DLs of context. Finally, we discuss the applicability of DLs of context to diverse problems, such as modeling inherently contextualized knowledge and expressing interoperability constraints over DL ontologies.

1.5 Summary of Publications

Most of the content of this thesis (discussed in CHAPTERS 3-5) has been presented in the following journal, conference, and workshop publications.

CHAPTER 3

- Víctor Gutiérrez Basulto, Jean Christoph Jung, Carsten Lutz. Complexity of Branching Temporal Description Logics. *European Conference on Artificial Intelligence (ECAI 2012), 2012*. (Best Student Paper Award)

CHAPTER 4

- Víctor Gutiérrez Basulto, Jean Christoph Jung, Carsten Lutz and Lutz Schröder. A Closer Look at the Probabilistic Description Logic Prob-$\mathcal{EL}$. *Conference on Artificial Intelligence (AAAI 2011), 2011.*

- Szymon Klarman and Víctor Gutiérrez Basulto. Two-Dimensional Description Logics for Context-Based Semantic Interoperability. *Conference on Artificial Intelligence (AAAI 2011), 2011.*

- Víctor Gutiérrez Basulto, Jean Christoph Jung, Carsten Lutz and Lutz Schröder. The Complexity of Probabilistic $\mathcal{EL}$. *International Workshop on Description Logics (DL 2011), 2011.*

CHAPTER 5

- Szymon Klarman and Víctor Gutiérrez Basulto. Description Logics of Context. *Journal of Logic and Computation*. In press, 2013.

– Szymon Klarman and Víctor Gutiérrez Basulto. Two-Dimensional Description Logics for Context-Based Semantic Interoperability. *Conference on Artificial Intelligence (AAAI 2011), 2011.*

– Szymon Klarman and Víctor Gutiérrez Basulto. Two-Dimensional Description Logics of Context. *International Workshop on Description Logics (DL 2011), 2011.* (Best Student Paper Award)

– Szymon Klarman and Víctor Gutiérrez Basulto. $\mathcal{ALC}_{\mathcal{ALC}}$: A Description Logic of Context. *European Conference on Logics in Artificial Intelligence (JELIA 2010), 2010*

2
Preliminaries

This chapter is dedicated to the presentation of standard definitions and computational complexity results in description logics and branching-time temporal logics.

2.1 Description Logics

Description logics [14], as introductorily discussed in Chapter 1, permit to represent terminological knowledge by means of *concepts*, denoting classes of objects of the domain of discourse (e.g., $\mathsf{Scientist}$, $\mathsf{Teacher}$), and *roles* (e.g., $\mathsf{publishes}$, $\mathsf{teaches}$), denoting binary relations between instances of concepts. Concepts and roles are inductively built from atomic concepts and atomic roles through the application of the concept and role constructors provided by a particular DL language. Remarkably, most DLs are well-behaved fragments of first-order logic, hence these DLs come with the standard Boolean constructors '$\sqcup$' (or), '$\sqcap$' (and) and '$\neg$' (not); moreover, they allow for the concepts '$\top$' (top) and '$\bot$' (bottom), representing *everything* and *nothing*, respectively. Furthermore, standard DLs include the quantifiers '$\forall$' and '$\exists$' used in combination with roles to restrict relations between individuals from two classes. We will begin, later on, our discussion on DLs by introducing the DL $\mathcal{ALC}$ which allows for these constructors.

Recall that DLs model the *terminological* and *assertional* knowledge of the domain of discourse via a *knowledge base* (or *ontology*). Intuitively, the terminological knowledge describes the interrelation between concepts while the assertional one describes the current state of affairs through the specification of the participation of particular individuals in a concept or in a role. The terminological component of an ontology is given through the *TBox* formalism, and the assertional one through the *ABox* formalism.

We start by defining the *vocabulary* over which a DL concept language is defined. A *vocabulary* is a triple $\Sigma = (\mathsf{N_C}, \mathsf{N_R}, \mathsf{N_I})$, where $\mathsf{N_C}$, $\mathsf{N_R}$ and $\mathsf{N_I}$ are infinite countably disjoint sets of *concept names*, *role names* and *individual names*, respectively.

Recall that one of the main differences of DLs with respect to their predecessors is that they come equipped with a formal *model-theoretic* semantics, providing the meaning of the symbols of the vocabulary through *interpretations*, containing a *domain* and an *interpretation function*. Such an interpretation function maps each concept name to a subset of the domain, each role name to a binary relation over the domain, and each individual name to an object of the domain.

Definition 2.1. *An* interpretation *is a pair* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$*, where* $\Delta^{\mathcal{I}}$ *is a non-empty set of individuals, called the* domain*, and* $\cdot^{\mathcal{I}}$ *is an* interpretation function *mapping each* $a \in \mathsf{N_I}$ *to an element* $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$*, each* $A \in \mathsf{N_C}$ *to a subset* $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ *and each* $r \in \mathsf{N_R}$ *to a binary relation* $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

In this thesis, we adopt the *unique name assumption*, that is, $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ for all distinct individuals $a, b \in \mathsf{N_I}$.

We now proceed to introduce different DL languages, which are characterized by the set of constructors they allow for defining complex concepts and roles. We begin with the presentation of the basic Boolean complete DL $\mathcal{ALC}$, introduced by Schmidt-Schauss and Smolka [71], allowing for constructors '$\sqcap$' (and), '$\neg$' (not) and $\exists r.C$.

Definition 2.2. $\mathcal{ALC}$ concepts C *are formed according to the following grammar:*

$$C ::= \top \mid A \mid \neg C \mid C \sqcap D \mid \exists r.C$$

where A *ranges over* $\mathsf{N_C}$*,* r *ranges over* $\mathsf{N_R}$ *and* C, D *range over concepts.*

Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ *be an interpretation. We extend the interpretation function* $\cdot^{\mathcal{I}}$ *to* $\mathcal{ALC}$ *concepts as follows:*

$$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}; \qquad (\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}; \qquad (C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}};$$
$$(\exists r.C)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} \mid \exists e \in \Delta^{\mathcal{I}} : (d, e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}.$$

We use the following standard abbreviations: $\bot = \neg\top$, $C \sqcup D = \neg(\neg C \sqcap \neg D)$, and $\forall r.C = \neg\exists r.\neg C$.

Before introducing knowledge bases, we present the minor extension of $\mathcal{ALC}$ that allows for *nominals*, which are concepts that represent a single individual. Nominals are used to express concepts that have only one instance (e.g., modeling the term *pope*) or to provide individuals with 'names' (e.g., modeling terms *Bremen*, *Weser*). Formally, the DL $\mathcal{ALCO}$ extends $\mathcal{ALC}$ with concepts $\{a\}$, where a ranges over $\mathsf{N_I}$. Furthermore, given an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, we interpret $(\{a\})^{\mathcal{I}}$ as $\{a^{\mathcal{I}}\}$.

Naturally, in order to be able to talk about computational complexity, we need to define the *size* of the different components of a DL. We begin by defining the size of a concept, that is, the number of symbols needed to write it down. We also introduce the standard notion of *subconcept*.

Definition 2.3. *The* size *of concepts over* $\Sigma = (\mathsf{N_C}, \mathsf{N_R}, \mathsf{N_I})$ *is inductively defined as follows:*

$|A| = |\top| = |\{a\}| := 1$*, where* A *ranges over* $\mathsf{N_C}$ *and* a *over* $\mathsf{N_I}$*,*

$|C \sqcap D| = |C|+|D|+1$, $|\neg C| = |\exists r.C| = |C|+1$, *where* C, D *range over concepts and* r *over* $\mathsf{N_R}$.

The set of subconcepts $\mathsf{sub}(C)$ *of a concept* C *is inductively defined as follows:*

$\mathsf{sub}(C) := \{C\}$, *if* C *is* $\top$, *a concept name, or a nominal,*

$\mathsf{sub}(C) := \{C\} \cup \mathsf{sub}(D_1) \cup \mathsf{sub}(D_2)$, *if* C *is of the form* $D_1 \sqcap D_2$,

$\mathsf{sub}(C) := \{C\} \cup \mathsf{sub}(D)$, *if* C *is of the form* $\exists r.D$.

In $\mathcal{ALC}$ and $\mathcal{ALCO}$ knowledge bases are composed of a *TBox*, representing the terminological knowledge of a domain by means of *concept inclusions*, and of an *ABox*, representing the assertional knowledge of a domain by means of *concept* and *role assertions*.

Definition 2.4. *A* concept inclusion (CI) *is an axiom of the form* $C \sqsubseteq D$, *a* concept assertion *is an axiom of the form* $C(a)$, *and a* role assertion *is an axiom of the form* $r(a, b)$, *where* C *is a concept,* r *is a role and* a, b *are individual names. An* ABox assertion *(or simply* assertion*) is a concept or role assertion.*

– A TBox *is a finite set of concept inclusions, an* ABox *is a finite set of assertions, and a* knowledge base (KB) *(or* ontology*) is a pair* $(\mathcal{T}, \mathcal{A})$, *where* $\mathcal{T}$ *is a TBox and* $\mathcal{A}$ *is an ABox.*

– An interpretation $\mathcal{I}$ satisfies *a CI* $C \sqsubseteq D$ *iff* $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, *a concept assertion* $C(a)$ *iff* $a^{\mathcal{I}} \in C^{\mathcal{I}}$, *and a role assertion* $r(a, b)$ *iff* $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$. *We respectively denote the satisfaction of a CI by* $\mathcal{I} \models C \sqsubseteq D$, *of a concept assertion by* $\mathcal{I} \models C(a)$ *and of a role assertion by* $\mathcal{I} \models r(a, b)$.

We proceed to introduce the classical reasoning problems in DLs: KB satisfiability, and concept satisfiability and subsumption w.r.t. a TBox. We are particularly interested in these problems since later on we study variants of them.

Definition 2.5. *An interpretation* $\mathcal{I}$ *is a* model of a concept C *if* $C^{\mathcal{I}} \neq \emptyset$; *moreover,* $\mathcal{I}$ *is a* model of a TBox $\mathcal{T}$ *if it satisfies all concept inclusions in* $\mathcal{T}$; *analogously, it is a* model of an ABox $\mathcal{A}$ *if it satisfies all assertions in* $\mathcal{A}$. *Finally,* $\mathcal{I}$ *is a* model *of a KB* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ *if it is a model of* $\mathcal{T}$ *and* $\mathcal{A}$. *We write* $\mathcal{I} \models \mathcal{K}$ *to denote that* $\mathcal{I}$ *is a model of* $\mathcal{K}$, *and* $\mathcal{I} \models \mathcal{T}$ *and* $\mathcal{I} \models \mathcal{A}$ *for a TBox* $\mathcal{T}$ *and an ABox* $\mathcal{A}$.

We next define the size of TBoxes, ABoxes, and KBs. The *size* $|\mathcal{A}|$ *of an ABox* $\mathcal{A}$ is the sum of the sizes of all assertions in $\mathcal{A}$, where the *size* $|C(a)|$ *of concept assertions* is $|C|$ and the *size* $|r(a, b)|$ *of role assertions* is 2. The *size* $|\mathcal{T}|$ *of a TBox* $\mathcal{T}$ is

$$|\mathcal{T}| := \sum_{C \sqsubseteq D \in \mathcal{T}} |C| + |D|.$$

The *size* $|\mathcal{K}|$ *of a KB* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is the sum of $|\mathcal{T}|$ and $|\mathcal{A}|$.

We now have the necessary ingredients to introduce the reasoning tasks in DLs relevant for this thesis, namely, KB satisfiability, concept satisfiability and subsumption w.r.t. a TBox.

	Concepts		
	Name	Syntax	Semantics
(1)	top concept	$\top$	Δ
(2)	disjunction	$C \sqcap D$	$\{x \in \Delta \mid x \in C^{\mathcal{I}} \cap D^{\mathcal{I}}\}$
(3)	negation	$\neg C$	$\{x \in \Delta \mid x \notin C^{\mathcal{I}}\}$
(4)	existential restriction	$\exists r.C$	$\{x \in \Delta \mid \exists y : (x, y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
(5)	nominal	$\{a\}$	$\{a^{\mathcal{I}}\}$
(6)	inverse roles	r^{-}	$\{(x, y) \in \Delta \times \Delta \mid (y, x) \in r^{\mathcal{I}}\}$
	Axioms		
(8)	concept assertion	$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
(9)	role assertion	$r(a, b)$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$
(10)	concept inclusion	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$

Figure 2.1: Semantics of concepts and roles

Definition 2.6. *A KB $\mathcal{K}$ is* satisfiable *if it has a model. A concept C is* satisfiable w.r.t. a TBox *$\mathcal{T}$ if there is a common model of $\mathcal{T}$ and C. A concept D* subsumes *a concept C* w.r.t. a TBox *$\mathcal{T}$, written $\mathcal{T} \models C \sqsubseteq D$, if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for every model $\mathcal{I}$ of $\mathcal{T}$.*

For brevity, we sometimes write $C \sqsubseteq_{\mathcal{T}} D$ instead of $\mathcal{T} \models C \sqsubseteq D$. Now, we introduce the reasoning problems of interest for this thesis.

Note that using conjunction and negation we can mutually reduce the concept satisfiability and subsumption problems, that is, $\mathcal{T} \models C \sqsubseteq D$ if $C \sqcap \neg D$ is unsatisfiable w.r.t. $\mathcal{T}$, and C is satisfiable w.r.t. $\mathcal{T}$ if $\mathcal{T} \not\models C \sqsubseteq \bot$.

The complexity of these reasoning tasks has been already established for $\mathcal{ALC}$ and $\mathcal{ALCO}$.

Theorem 2.1. *Concept satisfiability w.r.t. TBoxes and the KB satisfiability for $\mathcal{ALC}$ and $\mathcal{ALCO}$ is* ExpTime*-complete [14].*

The research on description logics has considered not only expressive languages, but also lightweight DLs with low computational complexity. A prominent lightweight family of DLs is the *$\mathcal{EL}$-family* [12], which is characterized by disallowing $\neg$ and therefore the abbreviations $C \sqcup D$ and $\forall r.C$. Notably, many members of the $\mathcal{EL}$-family admit polytime reasoning while still providing sufficient expressiveness for many applications. For example, in bioinformatics they are used to represent medical ontologies such as SNOMED CT, or in the semantic web, where they serve as the underlying language of the OWL 2 EL profile of the OWL 2 ontology language [62].

Formally, *$\mathcal{EL}$ concepts C* are formed by the following grammar:

$$C ::= \top \mid A \mid C \sqcap D \mid \exists r.C$$

where A ranges over $\mathsf{N_C}$ and r ranges over $\mathsf{N_R}$. Moreover, the extension $\mathcal{ELI}$ of $\mathcal{EL}$ further allows for inverse roles.

For example, the following CI stating that '*each professor is an academic with a PhD degree*' is formulated in $\mathcal{EL}$.

$$\mathsf{Professor} \sqsubseteq \mathsf{Academic} \sqcap \exists \mathsf{hasDegree}.\mathsf{PhD}$$

Note that, due to the absence of negation in $\mathcal{ELI}$ and $\mathcal{EL}$, the satisfiability problem is trivial for them in the sense that every concept is satisfiable w.r.t. every TBox. However, the subsumption problem w.r.t TBoxes is not trivial.

Theorem 2.2. *Concept subsumption w.r.t. TBoxes is* PTIME*-complete for $\mathcal{EL}$, and* EXPTIME*-complete for $\mathcal{ELI}$ [12, 13].*

2.2 Branching Temporal Logics

We study in this thesis, as previously discussed in Chapter 1, temporal description logics based on the branching-time temporal logics CTL and CTL*, introduced for specification and verification purposes by Clarke and Emerson [32], and Emerson and Halpern [34], respectively. Later on, in order to develop decision procedures for the proposed TDLs, we make use of known techniques and results for CTL and CTL*. Hence we devote this section to the presentation of the syntax and semantics of these logics.

The syntax of CTL* extends the standard constructors from *propositional logic* with the *temporal operators* $\bigcirc$ (next), $\Box$ (always) and $\mathcal{U}$ (until), and with the *path quantifier* $\mathbf{E}$. Intuitively, a formula $\bigcirc\psi$ states that ψ must be true in the next time point; $\Box\psi$ states that ψ must be true in all future time points; and $\psi_1\mathcal{U}\psi_2$ states that there is a time point in the future where ψ_2 is true, and ψ_1 is true in all time points between the current one and that satisfying ψ_2. Moreover, since we have a branching-time structure, we can quantify over possible futures by means of the path quantifier $\mathbf{E}$: $\mathbf{E}\psi$ states that there exists a possible future, starting from the current time point, in which ψ is true. In particular, CTL* formulas are defined in terms of state and path formulas as follows. For the rest of the thesis, we fix a countably infinite set of *atomic propositions* AP.

Definition 2.7. CTL* state formulas φ *and* CTL* path formulas ψ *are defined by the following grammar:*

$$\begin{aligned}
\varphi &::= p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathbf{E}\psi \\
\psi &::= \varphi \mid \neg\psi \mid \psi_1 \wedge \psi_2 \mid \bigcirc\psi \mid \Box\psi \mid \psi_1\mathcal{U}\psi_2
\end{aligned}$$

where p ranges over AP*, $\varphi, \varphi_1, \varphi_2$ are state formulas, and ψ, ψ_1, ψ_2 are path formulas.*

Without further quantification, a CTL* *formula* is a state formula. We use the following standard Boolean abbreviations:

$$\mathsf{true} = p \vee \neg p,\ \text{for some } p \in \mathsf{AP}; \qquad \mathsf{false} = \neg\mathsf{true}; \qquad \varphi_1 \vee \varphi_2 = \neg(\neg\varphi_1 \wedge \neg\varphi_2);$$

$$\varphi_1 \rightarrow \varphi_2 = \neg\varphi_1 \vee \varphi_2; \qquad \varphi_1 \leftrightarrow \varphi_2 = (\varphi_1 \rightarrow \varphi_2) \wedge (\varphi_2 \rightarrow \varphi_1).$$

Moreover, we use the following temporal abbreviations: $\mathbf{A}\varphi = \neg\mathbf{E}\neg\varphi$, $\Diamond\psi = \neg\Box\neg\psi$, $\Box^{<}\psi = \bigcirc\Box\psi$ and $\Diamond^{<}\psi = \bigcirc\Diamond\psi$.

We say that $\Box^{<}$ and $\Diamond^{<}$ are *strict* in the sense that $\Box^{<}\psi$ and $\Diamond^{<}\psi$ require ψ to happen in the strict future.

The precedence of the operators is the following: the unary operators bind stronger than the binary operators. In particular, the unary operators bind equally strong. In the case of the binary operators, the temporal operator $\mathcal{U}$ has precedence over the Boolean binary operators.

CTL is the *fragment* of CTL* in which temporal operators $\bigcirc, \Box$ and $\mathcal{U}$ must be *immediately* preceded by the path quantifier $\mathbf{E}$. Formally, CTL *state formulas* φ and CTL *path formulas* ψ are defined by the following grammar:

$$\begin{aligned} \varphi &::= p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathbf{E}\psi \\ \psi &::= \bigcirc\varphi \mid \Box\varphi \mid \varphi_1\mathcal{U}\varphi_2 \end{aligned}$$

where p ranges over AP, $\varphi, \varphi_1, \varphi_2$ are state formulas, and ψ is a path formula.

As in the case of CTL* we use the Boolean abbreviations, plus the following temporal ones:

$$\begin{array}{lll} \mathbf{A}\bigcirc\varphi{=}\neg\mathbf{E}\bigcirc\neg\varphi, & \mathbf{E}\Diamond\varphi{=}\mathbf{E}(\mathsf{true}\,\mathcal{U}\varphi), & \mathbf{A}\Box\varphi{=}\neg\mathbf{E}\Diamond\neg\varphi, \\ \mathbf{A}\Diamond\varphi{=}\neg\mathbf{E}\Box\neg\varphi, & \multicolumn{2}{l}{\mathbf{A}(\varphi_1\,\mathcal{U}\,\varphi_2) = \neg\mathbf{E}(\neg\varphi_2\,\mathcal{U}\,(\neg\varphi_1 \wedge \neg\varphi_2)) \wedge \neg\mathbf{E}\Box\neg\varphi_2.} \end{array}$$

For example, the following is a correct CTL* formula, but it is not a correct CTL one:

$$\mathbf{A}\Box(\bigcirc\Diamond p \wedge \neg(p\,\mathcal{U}\,q)).$$

In contrast, the following is a well-formed CTL formula: $\mathbf{A}\Box(\mathbf{A}\bigcirc\mathbf{E}\Diamond p \wedge \neg\mathbf{E}(p\,\mathcal{U}\,q))$. Note that indeed every temporal operator is immediately preceded by a path quantifier.

Clearly, in order to be able to talk about the computational complexity of CTL*, we need to define the *size* of a formula.

Definition 2.8. *The* size of a CTL* state formula φ *is inductively defined as follows:*

$$|p| = 1, \quad |\neg\varphi| = |\varphi| + 1, \quad |\varphi_1{\wedge}\varphi_2| = |\varphi_1| + |\varphi_2| + 1, \quad |\mathbf{E}\psi| = |\psi| + 1;$$

the size of a CTL* path formula ψ *is inductively defined as follows:*

$$|\neg\psi| = |\bigcirc\psi| = |\Box\psi| = |\psi| + 1, \quad |\psi_1{\wedge}\psi_2| = |\psi_1\mathcal{U}\psi_2| = |\psi_1| + |\psi_2| + 1.$$

The semantics of CTL* is given in terms of 2^{AP}-labeled trees. In the context of temporal logics, we sometimes refer to nodes of a tree as *time points* or *worlds*. Particularly, as expected, the satisfaction of a state formula is related to a specific time point, and that of a path formula to a specific path. Before defining the semantics of CTL*, we introduce the type of trees over which it is defined.

Definition 2.9. *A* tree *is a directed graph $T = (W, E)$ where $W \subseteq (\mathbb{N} \setminus \{0\})^*$ is a prefix-closed non-empty set of* nodes *and $E = \{(w, wc) \mid wc \in W, w \in \mathbb{N}^*, c \in \mathbb{N}\}$ a set of* edges*; we generally assume that $wc \in W$ and $c' < c$ implies $wc' \in W$ and that $E \subseteq W \times W$ is a total relation. We say that wc is a* successor *of w, and that the node $\varepsilon \in W$ is the* root *of T.*

For brevity and since E can be reconstructed from W, we will usually identify T with W. Furthermore, we say that T is a *k-ary tree*, $k \geq 1$ if every node of T has exactly k successors. We next introduce some auxiliary notions: a *path* in a tree $T = (W, E)$ starting at a node w is a minimal set $\pi \subseteq W$ such that $w \in \pi$ and for each $w' \in \pi$, there is exactly one $c \in \mathbb{N}$ with $w'c \in \pi$. We use $\mathsf{Paths}(w)$ to denote the set of all paths starting at the node w; and for a path $\pi = w_0 w_1 w_1 \cdots$ and $i \geq 0$, we use $\pi[i]$ to denote w_i and $\pi[i..]$ to denote the path $w_i w_{i+1} \cdots$.

We now have the necessary ingredients to define trees with nodes labeled with elements of an alphabet. Let Σ be a finite alphabet. A Σ*-labeled tree* $\mathfrak{T}$ is a pair (T, τ) where T is a tree and $\tau : T \to \Sigma$ assigns a letter from Σ to each node. We sometimes identify (T, τ) with τ.

Clearly, to define the semantics of CTL*, we consider Σ-labeled trees with $\Sigma = 2^{\mathsf{AP}}$. Intuitively, the label of a time point contains the propositional letters holding at this time point.

Definition 2.10. *Let $\mathfrak{T} = (T, \tau)$ be a 2^{AP}-labeled tree. For a time point w in $\mathfrak{T}$, the* truth relation *$\models$ for* CTL* *state formulas is defined as follows.*

$\mathfrak{T}, w \models p \in \mathsf{AP}$ *iff* $p \in \tau(w)$;

$\mathfrak{T}, w \models \neg\varphi$ *iff* $\mathfrak{T}, w \not\models \varphi$;

$\mathfrak{T}, w \models \varphi_1 \wedge \varphi_2$ *iff* $\mathfrak{T}, w \models \varphi_1$ *and* $\mathfrak{T}, w \models \varphi_2$;

$\mathfrak{T}, w \models \mathbf{E}\psi$ *iff* $\mathfrak{T}, \pi \models \psi$ *for some* $\pi \in \mathsf{Paths}(w)$.

For a path π in $\mathfrak{T}$, the truth relation $\models$ for path formulas is defined as follows:

$\mathfrak{T}, \pi \models \varphi$ *iff* $\mathfrak{T}, \pi[0] \models \varphi$;

$\mathfrak{T}, \pi \models \neg\psi$ *iff* $\mathfrak{T}, \pi \not\models \psi$;

$\mathfrak{T}, \pi \models \psi_1 \wedge \psi_2$ *iff* $\mathfrak{T}, \pi \models \psi_1$ *and* $\mathfrak{T}, \pi \models \psi_2$;

$\mathfrak{T}, \pi \models \bigcirc\psi$ *iff* $\mathfrak{T}, \pi[1..] \models \psi$;

$\mathfrak{T}, \pi \models \Box\psi$ *iff* $\forall j \geq 0. \mathfrak{T}, \pi[j..] \models \psi$;

$\mathfrak{T}, \pi \models \psi_1 \,\mathcal{U}\, \psi_2$ *iff* $\exists j \geq 0. (\mathfrak{T}, \pi[j..] \models \psi_2 \wedge \forall 0 \leq k < j. (\mathfrak{T}, \pi[k..] \models \psi))$.

In CTL*, as in DLs, one of the classical reasoning problems is the *satisfiability problem*: a 2^{AP}-labeled tree $\mathfrak{T}$ is a *model of a* CTL* *formula* φ if $\mathfrak{T}, \varepsilon \models \varphi$. A CTL* formula φ is *satisfiable* if there exists a 2^{AP}-labeled tree $\mathfrak{T}$ such that $\mathfrak{T}$ is a model of φ.

For CTL* and its fragment CTL the computational complexity of the satisfiability problem has been established.

Theorem 2.3. *Satisfiability for CTL* is 2*EXPTIME*-complete and for CTL is* EXPTIME*-complete [35, 33].*

3 Branching Temporal Description Logics

We dedicate this chapter to the study of two-dimensional DLs for representing and reasoning about (branching-time) temporal aspects of knowledge. *Branching Temporal Description Logics* emerge from the combination of classical DLs with the branching-time temporal logics CTL and CTL*. We concentrate on the investigation of branching TDLs based on the DLs $\mathcal{EL}$ and $\mathcal{ALC}$. The main technical contributions are algorithms for satisfiability that are more direct than existing approaches, and (mostly) tight elementary complexity bounds that range from PTIME to 2EXPTIME and 3EXPTIME.

3.1 Introduction

Classical description logics, as fragments of first order logic, aim at the representation of and reasoning about *static* knowledge. The inability to capture dynamic and temporal aspects has often been criticized because many relevant applications depend on this type of knowledge, for example:

(1) these capabilities are needed in medical ontologies such as SNOMED CT and FMA [22] to accurately describe terms that refer to dynamic aspects. For example, think about the disease *malaria* which refers to repeating patterns, or about the finding *hyperplasia* which refers to a proliferation of cells that potentially develops into a tumor in the future.

(2) Classical DLs are used as a language for describing the conceptual database models, hence considerable research has been devoted to extending this approach to capture also the evolution of databases over time [3, 9].

To tackle this shortcoming of classical DLs, since the publication of Schild's seminal work in 1993 [70], a vast amount of investigation on *temporal description logics (TDLs)* has been carried out. These studies have resulted in several proposals for constructing TDLs; meticulously, surveyed by Artale *et al.* and Lutz *et al.* [5, 58]. A prominent approach to TDLs, following Schild's original proposal, is to combine static DLs with the standard temporal logics LTL, CTL and CTL* in the style of multi-dimensional description logics. While a large body of literature is available for linear-time TDLs based on combinations of DLs with the temporal logic LTL [7, 15, 8, 37], only limited research was devoted to branching-time TDLs based on CTL and CTL* [48, 19]. From the perspective of ontology applications such as those discussed under (1) above, this is slightly surprising because using LTL operators often results in a modeling that is unrealistically strict. As an example, consider the statement '*each student will eventually be a graduate*'. In TDLs based on LTL, this is usually modeled using one of the following axioms.

$$\mathsf{Student} \sqsubseteq \Diamond \mathsf{Graduate}$$

$$\mathsf{Student} \sqsubseteq \mathsf{Student}\,\mathcal{U}\,\mathsf{Graduate}$$

Intuitively, the first axiom states that each student will be graduated at some point in the future, while the second axiom further requires each student to remain a student until he graduates. Alas, due to the linear structure of time, these axioms exclude the possibility that a student leaves university without a degree. On the other hand, in TDLs based on the branching-time TL CTL, it is possible to use the much more cautious following statement based on the existential path quantifier $\mathbf{E}$, stating that there is a *possible future* in which the student obtains a degree and leaving open the possibility of other possible futures.

$$\mathsf{Student} \sqsubseteq \mathbf{E}(\mathsf{Student}\,\mathcal{U}\,\mathsf{Graduate})$$

Furthermore, based on the universal path quantifier $\mathbf{A}$, strict statements such as '*each human will eventually die, and stay dead*' can be expressed as

$$\mathsf{Human} \sqsubseteq \mathbf{A}\Diamond\mathbf{A}\Box\mathsf{Dead}$$

Decidability boundaries for TDLs have been obtained through the prominent research on *first-order temporal logics ($\mathcal{QTL}$)* by Hodkinson, Wolter and Zakharyaschev [47, 48]. The main result of the research in [47] is the identification of a syntactic restriction, *monodicity*, allowing to design decidable expressive fragments of $\mathcal{QTL}s$ of linear-time. The rough idea to obtain decidable first-order temporal languages is, on the one hand, restrict the non-temporal part of the language to a decidable fragment of first-order logic. On the other hand, in the temporal part only monodic formulas are allowed: formulas in which temporal operators are applied *only* to first-order formulas with at most one free variable. Remarkably, later on, Hodkinson, Wolter and Zakharyaschev [48] showed that the monodicity condition is not enough to design decidable fragments of $\mathcal{QTL}s$ of branching-time based on CTL*. In particular, they showed that the *one-variable* fragment of first-order CTL* is undecidable. Since this fragment is clearly monodic, the introduction of further restrictions to attain decidability in $\mathcal{QTL}s$ of branching-time was necessary. They showed that decidable fragments of $\mathcal{QTL}s$ of branching-time can be designed as for linear-time by restricting the application of first-order quantifiers to temporal *state* formulas, and the application of temporal operators and path quantifiers to monodic formulas. Note that if CTL is considered the additional restrictions are fulfilled.

The results obtained by Hodkinson *et al.* on first-order temporal logics of branching-time [48] have the following consequences on the decidability of branching-time TDLs. First, note that $\exists\Diamond r.C$ is not a monodic formula since roles correspond to first-order formulas with two free variables while $\Diamond\exists r.C$ is indeed a monodic formula since concepts correspond to first-order formulas with one free variable. As a consequence of this, the application of temporal operators to roles, or permitting rigid roles most probably results in undecidability. Second, if temporal or rigid roles are not present: TDLs based on CTL are decidable while TDLs based on CTL* have to be appropriately restricted in order to attain decidability, namely, inside concept inclusions, only state concepts should be allowed, but no path concepts (these correspond to state formulas and path formulas in CTL* –*cf.* Section 2.2).

Since Hodkinson *et al.* [48] obtained decidability by translating into monadic second order logic on trees, these results only come with a non-elementary upper complexity bound. The aim of this chapter is then to reconsider branching-time TDLs based on CTL and CTL* (under the mentioned restriction), to develop more direct algorithms for the satisfiability problem, and to analyze the computational complexity. We concentrate on TDLs that are most natural from the perspective of ontology applications: we consider the basic DLs $\mathcal{ALC}$ and $\mathcal{EL}$, allow the application of temporal operators to concepts and (sometimes) to TBox statements but never to roles, and assume constant domains.

Contributions: Our investigation starts with the TDLs CTL$_{\mathcal{ALC}}$ and CTL$^*_{\mathcal{ALC}}$ in the case where temporal operators can only be applied to concepts and a global TBox is considered. We use a uniform approach to both logics that consists of a combination of Pratt-style type elimination and methods based on nondeterministic tree automata. We obtain EXPTIME-completeness for satisfiability in CTL$_{\mathcal{ALC}}$ and 2EXPTIME-completeness for satisfiability in CTL$^*_{\mathcal{ALC}}$, thus the combined logics are computationally no more complex than their components. Moreover, we consider combinations of the light DL $\mathcal{EL}$ with fragments of CTL. Recall that the crucial advantage of $\mathcal{EL}$ over $\mathcal{ALC}$ is that it admits efficient (polytime) reasoning and our main aim is to understand how far this property transfers to a TDL based on $\mathcal{EL}$. We are able to identify a polytime TDL that could be viewed as an analog of non-temporal $\mathcal{EL}$; it includes the temporal operator $\mathbf{E}\Diamond$. Most other versions of CTL$_{\mathcal{EL}}$ turn out to be hard for PSPACE or EXPTIME. Finally, we reconsider CTL$_{\mathcal{ALC}}$ and CTL$^*_{\mathcal{ALC}}$, but now we additionally allow temporal operators to be applied to TBoxes. To establish an elementary upper bound, we again use a uniform approach that consists of a careful combination of alternating 2-way tree automata and nondeterministic tree automata for CTL and CTL*. We obtain a 2EXPTIME upper bound for CTL$_{\mathcal{ALC}}$ and a 3EXPTIME upper bound for CTL$^*_{\mathcal{ALC}}$. For CTL$_{\mathcal{ALC}}$, we prove a matching lower bound using a reduction of the word problem of exponentially space bounded alternating Turing machines, which shows that, in the presence of temporal TBoxes, the combination of $\mathcal{ALC}$ and CTL results in an increase of computational complexity by one exponential. For CTL$^*_{\mathcal{ALC}}$, the complexity remains open between 2EXPTIME and 3EXPTIME.

Organization: In the next section we formally introduce the syntax and semantics of branching temporal description logics. Section 3.3 is devoted to the study of $\mathrm{CTL}_{\mathcal{ALC}}$ and $\mathrm{CTL}^*_{\mathcal{ALC}}$ in the case where temporal operators are applied only to concepts and a global TBox is considered. Continuing with the study of temporal concepts, Section 3.4 is committed to investigate combinations of $\mathcal{EL}$ with fragments of CTL. Section 3.6 revisits $\mathrm{CTL}_{\mathcal{ALC}}$ and $\mathrm{CTL}^*_{\mathcal{ALC}}$, now in the case where temporal operators are applied not only to concepts but also to TBox formulas. Section 3.6 presents some final conclusions.

3.2 Introducing Branching Temporal Description Logics

Branching Temporal Description Logics emerge from the combination of the branching-time temporal logics CTL and CTL* with classical DLs in the style of multi-dimensional modal logics. In contrast to linear-time TDLs, this family of TDLs is able to differentiate between possible and necessary future developments of knowledge via the use of path quantifiers. In this chapter, we concentrate on branching TDLs based on the traditional DL $\mathcal{ALC}$ and its fragment $\mathcal{EL}$.

3.2.1 Syntax and Semantics

Definition 3.1. *Fix countably infinite disjoint sets* $\mathsf{N_C}$ *and* $\mathsf{N_R}$ *of* concept names *and* role names*, respectively.* $\mathrm{CTL}^*_{\mathcal{ALC}}$-state concepts C *and* $\mathrm{CTL}^*_{\mathcal{ALC}}$-path concepts $\mathcal{C}$*,* $\mathcal{D}$ *are defined by the following grammar:*

$$\begin{aligned} C \quad &::= \quad \top \mid A \mid \neg C \mid C \sqcap D \mid \exists r.C \mid \mathbf{E}\mathcal{C} \\ \mathcal{C},\mathcal{D} \quad &::= \quad C \mid \mathcal{C} \sqcap \mathcal{D} \mid \neg\mathcal{C} \mid \bigcirc\mathcal{C} \mid \Box\mathcal{C} \mid \mathcal{C}\mathcal{U}\mathcal{D} \end{aligned}$$

where A *ranges over* $\mathsf{N_C}$*,* r *ranges over* $\mathsf{N_R}$*,* C, D *range over state concepts, and* $\mathcal{C}$*,* $\mathcal{D}$ *range over path concepts.*

$\mathrm{CTL}_{\mathcal{ALC}}$ is the fragment of $\mathrm{CTL}^*_{\mathcal{ALC}}$ where only CTL operators are allowed. Formally, $\mathrm{CTL}_{\mathcal{ALC}}$*-state concepts* C and $\mathrm{CTL}_{\mathcal{ALC}}$*-path concepts* $\mathcal{C}$ are defined by the following grammar:

$$\begin{aligned} C \quad &::= \quad \top \mid A \mid \neg C \mid C \sqcap D \mid \exists r.C \mid \mathbf{E}\mathcal{C} \\ \mathcal{C} \quad &::= \quad \bigcirc C \mid \Box C \mid C\mathcal{U}D \end{aligned}$$

Without further qualification, the term *concept* refers to a state concept. Moreover, we assume the standard Boolean abbreviations used in classical DLs and the standard temporal abbreviations used in branching-time temporal logics (*cf.* Section 2.2).

Definition 3.2. *A* $\mathrm{CTL}^*_{\mathcal{ALC}}$-TBox $\mathcal{T}$ *is a finite set of* concept inclusions (CIs) $C \sqsubseteq D$ *with* C, D $\mathrm{CTL}^*_{\mathcal{ALC}}$*-state concepts. A* $\mathrm{CTL}_{\mathcal{ALC}}$-TBox *is defined analogously.*

$\mathrm{CTL}^*_{\mathcal{ALC}}$-TBoxes are thus constructed as for classical DLs but using $\mathrm{CTL}^*_{\mathcal{ALC}}$-state concepts. Note that inclusions between path concepts are not admitted as they result in undecidability [48]. We next present an example of a CI based on $\mathrm{CTL}_{\mathcal{ALC}}$ concepts, and an example of a CI based on $\mathrm{CTL}^*_{\mathcal{ALC}}$ concepts, respectively.

$\mathsf{Student} \sqsubseteq \mathbf{E}\Diamond(\mathsf{Graduated} \sqcap \mathbf{A}\Box\exists\mathsf{worksFor.Company})$
$\mathsf{Prof} \sqsubseteq \mathbf{A}(\mathsf{Prof}\,\mathcal{U}\mathsf{Retired} \ \sqcap \ (\mathsf{Retired} \rightarrow \bigcirc\mathsf{Retired}))$

Intuitively, the first CI states that '*it is possible that each student will eventually graduate, and from there on lastingly will work in a company*'. The second CI states that '*a professor remains a professor until he retires, and if he is retired today, he will remain retired tomorrow*'.

The semantics of branching TDLs is given in terms of temporal interpretations, which are infinite trees in which every node w is associated with a classical interpretation $\mathcal{I}_w$.

Definition 3.3. *A* temporal interpretation *is a structure* $\mathfrak{I} = (\Delta, T, \{\mathcal{I}_w\}_{w \in W})$ *where* $T = (W, E)$ *is an infinite tree, and for each* $w \in W$, $\mathcal{I}_w$ *is an interpretation with domain* Δ. *The mapping* $\cdot^{\mathfrak{I},w}$ *is extended from concept names to* CTL$^*_{\mathcal{ALC}}$*-state concepts as follows:*

$$
\begin{aligned}
\top^{\mathfrak{I},w} &= \Delta; \\
(\neg C)^{\mathfrak{I},w} &= \Delta \backslash C^{\mathfrak{I},w}; \\
(C \sqcap D)^{\mathfrak{I},w} &= C^{\mathfrak{I},w} \cap D^{\mathfrak{I},w}; \\
(\exists r.C)^{\mathfrak{I},w} &= \{d \in \Delta \mid \exists e : (d,e) \in r^{\mathfrak{I},w} \wedge e \in C^{\mathfrak{I},w}\}; \\
(\mathbf{E}\mathcal{C})^{\mathfrak{I},w} &= \{d \in \Delta \mid d \in \mathcal{C}^{\mathfrak{I},\pi} \text{ for some } \pi \in \mathsf{Paths}(w)\};
\end{aligned}
$$

where $\mathcal{C}^{\mathfrak{I},\pi}$ *refers to the extension of* CTL$^*_{\mathcal{ALC}}$*-path concepts on a given path* π*, defined as:*

$$
\begin{aligned}
C^{\mathfrak{I},\pi} &= C^{\mathfrak{I},\pi[0]} \quad \text{for state concepts } C; \\
(\neg\mathcal{C})^{\mathfrak{I},\pi} &= \Delta \setminus \mathcal{C}^{\mathfrak{I},\pi}; \\
(\mathcal{C} \sqcap \mathcal{D})^{\mathfrak{I},\pi} &= \mathcal{C}^{\mathfrak{I},\pi} \cap \mathcal{D}^{\mathfrak{I},\pi}; \\
(\bigcirc\mathcal{C})^{\mathfrak{I},\pi} &= \{d \in \Delta \mid d \in \mathcal{C}^{\mathfrak{I},\pi[1..]}\}; \\
(\Box\mathcal{C})^{\mathfrak{I},\pi} &= \{d \in \Delta \mid \forall j \geq 0.\ d \in \mathcal{C}^{\mathfrak{I},\pi[j..]}\}; \\
(\mathcal{C}\mathcal{U}\mathcal{D})^{\mathfrak{I},\pi} &= \{d \in \Delta \mid \exists j \geq 0.\ (d \in \mathcal{D}^{\mathfrak{I},\pi[j..]} \wedge (\forall 0 \leq k < j.\ d \in \mathcal{C}^{\mathfrak{I},\pi[k..]}))\}.
\end{aligned}
$$

We usually write $A^{\mathfrak{I},w}$ instead of $A^{\mathcal{I}_w}$, and intuitively $d \in A^{\mathfrak{I},w}$ means that in the interpretation $\mathfrak{I}$, the object d is an instance of the concept name A *at time point* w. Moreover, note that in the previous definition we make the *constant domain assumption*, that is, each time point shares the same domain Δ. Intuitively that means that objects are not created or destroyed over time. This is the most general assumption in the TDLs based on $\mathcal{ALC}$ since expanding, decreasing and varying domains can be simulated [38].

In this investigation, we are interested in studying the complexity of the concept *satisfiability* problem w.r.t. CTL$^*_{\mathcal{ALC}}$ and CTL$_{\mathcal{ALC}}$ TBoxes.

Definition 3.4. *A temporal interpretation $\mathfrak{I}$ is a* model *of a concept C if $C^{\mathfrak{I},\varepsilon} \neq \emptyset$; it is a* model *of a TBox $\mathcal{T}$ if $C^{\mathfrak{I},w} \subseteq D^{\mathfrak{I},w}$ for all $w \in W$ and all $C \sqsubseteq D$ in $\mathcal{T}$. A concept C is* satisfiable w.r.t. *$\mathcal{T}$ if there is a common model of $\mathcal{T}$ and C.*

3.3 Reasoning in $\mathrm{CTL}^*_{\mathcal{ALC}}$ and $\mathrm{CTL}_{\mathcal{ALC}}$

This section begins our investigation on the computational complexity of branching-time TDLs. We present algorithms for satisfiability for $\mathrm{CTL}_{\mathcal{ALC}}$ and $\mathrm{CTL}^*_{\mathcal{ALC}}$, developed under a uniform approach that consists of a combination of Pratt-style type elimination and techniques based on nondeterministic automata on infinite trees. Before introducing our algorithms, we present the basics on nondeterministic automata.

The automata-theoretic approach has been used to devise elementary decision procedures for many logics. In particular, automata working on infinite trees are used to decide whether a formula in a logic with the *tree-model property* is satisfiable [77]. Hence, these type of automata are a crucial ingredient for the development of decision procedures for the satisfiability problem for CTL and CTL*. We next introduce *nondeterministic tree automata* and show their use in temporal reasoning.

Definition 3.5. *Let Σ be a finite alphabet. A* nondeterministic tree automaton *(NTA) on Σ-labeled k-ary trees is a tuple $\mathcal{A} = (Q, \Sigma, Q^0, \delta, F)$ where Q is a finite set of* states*, $Q^0 \subseteq Q$ is the set of* initial states*, $F \subseteq Q$ is a set of* recurring states*, and $\delta : Q \times \Sigma \to 2^{Q^k}$ is the* transition function*.*

A run *r* of an NTA *$\mathcal{A}$ on a Σ-labeled k-ary tree (T, τ) is a Q-labeled k-ary tree (T, r) such that $r(\varepsilon) \in Q_0$ and for each node $w \in T$, we have*

$$\langle r(w\cdot 1), \ldots, r(w\cdot k)\rangle \in \delta(r(w), \tau(w)).$$

(T, r) is accepting *if all its paths satisfy the acceptance condition.*

Intuitively, the nondeterminism of an NTA results from the transition relation, that is, when the automaton is at state $q \in Q$ and reads a node w labeled by $\sigma \in \Sigma$, it proceeds to nondeterministically choose a k-tuple $\langle q_1, \ldots, q_k\rangle$ from $\delta(q, \sigma)$, and then sends copies of itself to each successor node $w\cdot i$ in state q_i.

In this thesis, we consider the so-called *Büchi acceptance condition*, defined as follows: given a run (T, r), a path $\pi = w_0 w_1 \cdots$ which starts at ε *satisfies the Büchi acceptance condition* if $r(w_i) \in F$ for infinitely many i. Moreover, we say that a Σ-labeled tree (T, τ) is *accepted* by an NTA $\mathcal{A}$ if there is an accepting run of $\mathcal{A}$ on (T, τ). We denote by $L(\mathcal{A})$ the set of trees accepted by $\mathcal{A}$. From now on, we denote by NBTA, NTAs using the Büchi acceptance condition.

A fundamental automata-theoretic problem is to decide whether an automaton accepts some input. Formally, the *nonemptiness problem* is to decide, given an automaton $\mathcal{A}$ whether $L(\mathcal{A}) \neq \emptyset$.

Theorem 3.1 ([81])**.** *The nonemptiness problem for NBTAs can be decided in quadratic time.*

Later on, we use the fact that the class of NBTA recognizable languages is *closed under intersection* [41], that is, given Büchi automata $\mathcal{A}$ and $\mathcal{B}$, we can construct an NBTA $\mathcal{A}'$ such that $L(\mathcal{A}') = L(\mathcal{A}) \cap L(\mathcal{B})$. Moreover, $\mathcal{A}'$ can be constructed with only a polynomial blow up [41].

Automata-theoretic techniques on infinite words and trees have been successfully used to develop decision procedures in linear-time and branching-time temporal logics, respectively [54, 79]. The core idea for deciding whether a (branching-time) temporal formula φ is satisfiable is to construct an automaton $\mathcal{A}_\varphi$ that accepts all the (tree) models of φ, and then check whether $L(\mathcal{A}_\varphi)$ is nonempty. In other words, we reduce the problem of satisfiability for temporal logics to the automata-theoretic problem of nonemptiness.

We now assert the existence of NBTAs for CTL and CTL*, as well as their constructability within certain time bounds. For $n{>}0$, we use $\mathsf{Mod}_n(\varphi)$ to denote the set of all n-ary models of φ. The following *sufficient degree property* shows that it is sufficient to only consider models of certain arity.

Proposition 3.2 ([54]). *A CTL* formula φ is satisfiable iff $\mathsf{Mod}_{\#_{\mathbf{E}}(\varphi)} \neq \emptyset$, where $\#_{\mathbf{E}}(\varphi)$ is the number of subformulas of φ that are of the form $\mathbf{E}\psi$.*

The following theorem shows the precise relation between the satisfiability problem for temporal logics and the nonemptiness problem for NBTAs. We use $\mathsf{ap}(\varphi)$ to denote the set of atomic propositions in a CTL* formula φ.

Theorem 3.3 ([53, 79]). *For a CTL*-formula φ and $n \geq 0$, one can construct an NBTA $\mathcal{A}_\varphi = (Q, \Sigma, \delta, Q^0, F)$ in time $\mathsf{poly}(|Q|{+}n)$ such that $L(A_\varphi) = \mathsf{Mod}_n(\varphi)$, $\Sigma = 2^{\mathsf{ap}(\varphi)}$, $|Q| \in 2^{2^{\mathsf{poly}(|\varphi|)}}$, and $|Q| \in 2^{\mathsf{poly}(|\varphi|)}$ when φ is a CTL formula.*

3.3.1 Algorithms for Concept Satisfiability w.r.t. TBoxes for $\mathrm{CTL}^*_{\mathcal{ALC}}$ and $\mathrm{CTL}_{\mathcal{ALC}}$

We now are ready to present our algorithm for satisfiability for $\mathrm{CTL}^*_{\mathcal{ALC}}$ and $\mathrm{CTL}_{\mathcal{ALC}}$. This uniform decision procedure yields a tight EXPTIME upper bound for the former case and a tight 2EXPTIME upper bound for the latter. The lower bounds are inherited from CTL and CTL* [36, 80]. Notably, the proposed approach is enabled by the fact that the interaction between the DL dimension and the temporal dimension is limited, similar to the *fusion* of modal logics [38]. Note, however, that fusions correspond to expanding domains while we use constant domains which impose additional technical difficulties. We emphasize that the careful combination of types and existing tree automata for CTL and CTL* allows us to avoid many of the technical intricacies of CTL*, resulting in a rather transparent overall approach.

Let us fix a concept C and a TBox $\mathcal{T}$, formulated in $\mathrm{CTL}^*_{\mathcal{ALC}}$ or its fragment $\mathrm{CTL}_{\mathcal{ALC}}$. We assume w.l.o.g. that $\mathcal{T}$ is of the form $\{\top \sqsubseteq C_\mathcal{T}\}$ and use $\mathsf{cl}(\mathcal{T})$ to denote the set of state concepts that occur in $\mathcal{T}$, closed under subconcepts and single negation.

Definition 3.6. *A* type *for $\mathcal{T}$ is a set $t \subseteq \mathsf{cl}(\mathcal{T})$ such that $C_\mathcal{T} \in t$. A* temporal type *for $\mathcal{T}$ has the form (t, i) with t a type for $\mathcal{T}$ and $i \geq 0$ a* distance *that denotes how far a time point w of a tree structure is from the root (i.e., the length $|w|$ of the word w).*

Algorithm 1: $\mathrm{CTL}^*_{\mathcal{ALC}}$ and $\mathrm{CTL}_{\mathcal{ALC}}$ SATISFIABILITY

Input: Concept C, TBox $\mathcal{T}$ formulated in $\mathrm{CTL}^*_{\mathcal{ALC}}$ or $\mathrm{CTL}_{\mathcal{ALC}}$
Initialize: $i := 0$; $S_0 := \mathsf{ttp}_{n_0}(\mathcal{T})$

repeat
$S_{i+1} := \{(t,j) \in S_i \mid (t,j) \text{ is realizable in } S_i\}$
until $S_i = S_{i+1}$

if exists $(t,0) \in S_i$ such that $C \in t$, return *satisfiable*
otherwise, return *unsatisfiable*

For any $n \geq 0$, we use $\mathsf{ttp}_n(\mathcal{T})$ to denote the set of all temporal types (t,i) for $\mathcal{T}$ with $i \leq n$. Moreover, we introduce some notational conventions. For a type t, let $\overline{t}$ denote the result of replacing every concept $C \in t \backslash \mathsf{N_C}$ with a fresh concept name X_C, and let cn denote the set of all resulting concept names, including those in $\mathcal{T}$. For $C \in \mathsf{cl}(\mathcal{T})$, let $\overline{C}$ denote the result of replacing in C every subconcept $\exists r.D$ with $X_{\exists r.D}$, and $\sqcap$ with $\wedge$. Let $\sharp_{\mathbf{E}}(\mathcal{T})$ denote the number of state concepts in $\mathsf{cl}(\mathcal{T})$ that are of the form $\mathbf{E}\mathcal{C}$.

We now formally describe the *elimination condition* in which n_0 is an appropriate bound in the sense that our type elimination algorithm is correct if it starts with $\mathsf{ttp}_{n_0}(\mathcal{T})$.

Definition 3.7. *Let S be a set of temporal types for $\mathcal{T}$. A temporal type (t,i) is* realizable in S *if the following conditions are satisfied:*

(DL) *if $\exists r.C \in t$, then there is a $(t',i) \in S$ such that $\{C\} \cup \{\neg D \mid \neg\exists r.D \in t\} \subseteq t'$;*

(TL) *(t,i) is* temporal-realizable in *S, that is, there is a 2^{cn}-labeled $\sharp_{\mathbf{E}}(\mathcal{T})$-ary tree (T,τ) that satisfies the following conditions, where $\rho(i) = \min\{n_0, i\}$:*

a) for some $w \in T$ with $|w| = i$, we have $\tau(w) = \overline{t}$;

b) for each $w \in T$ with $|w| = j$, there is a $(t', \rho(j)) \in S$ with $\tau(w) = \overline{t'}$;

c) ε satisfies $\varphi = \mathbf{A}\Box \bigwedge_{X_C \in \mathsf{cn}} X_C \leftrightarrow \overline{C}$.

Intuitively, Condition DL takes care of the DL dimension of $\mathrm{CTL}^*_{\mathcal{ALC}}$ while Condition TL takes care of the temporal dimension, that is, the tree (T,τ) describes the temporal evolution of a single domain element. Note that, by definition of $\overline{C}$, Condition TL(c) takes care of the Boolean constructors. Hence we do not require types to 'respect' Booleans.

Algorithm 1 above implements a type elimination algorithm for deciding satisfiability for $\mathrm{CTL}^*_{\mathcal{ALC}}$ and $\mathrm{CTL}_{\mathcal{ALC}}$: it starts with the set of temporal types $\mathtt{ttp}_{n_0}(\mathcal{T})$ for some appropriate bound n_0 to be determined later and then generates a decreasing sequence $S_0 \supseteq S_1 \supseteq \ldots$ where $S_0 = \mathtt{ttp}_{n_0}(\mathcal{T})$ and S_{j+1} is obtained from S_j by eliminating temporal types that, intuitively, cannot occur in any model of $\mathcal{T}$. The algorithm terminates when no further types are eliminated, that is, when $S_j = S_{j+1}$. It returns *satisfiable* if there is a surviving (t, i) with $C \in t$ and $i = 0$, and *unsatisfiable* otherwise.

Now, it remains to determine the bound n_0 for which Algorithm 1 is correct. The intuition behind the number n_0 and the use of $\rho(\cdot)$ in Condition TL is that time points which are close to the root of the temporal structure behave in a special way. For example, if

$$\mathcal{T} = \{\top \sqsubseteq \mathbf{A}\bigcirc\bigcirc\neg A\},$$

then time points w with distance $|w| < 2$ are special in the sense that they can satisfy A. Using binary counting, one can construct similar examples where time points with exponential distance are still special. The main objective is thus that, by means of an adequate n_0, the final result S of Algorithm 1 represents the infinite expansion $S_\omega := S \cup \{(t, m) \mid (t, n_0) \in S \wedge m > n_0\}$, such that all $(t, i) \in S_\omega$ satisfy Conditions DL and TL when in Condition TL $\rho(i)$ is replaced with i. This means that we can actually extend the result of Algorithm 1 to construct a (infinite) temporal model. This suggests the main property to attain by choosing an appropriate bound n_0:

$(*)$ if (t, n_0) is realizable in S, then $(t, n_0 + \ell)$ is realizable in S for any $\ell \geq 0$.

One might be tempted to choose $n_0 = |\mathtt{tp}(\mathcal{T})|$. While this is indeed sufficient for $\mathrm{CTL}_{\mathcal{ALC}}$, it does not work for $\mathrm{CTL}^*_{\mathcal{ALC}}$, where types do not capture enough information about models and time points of double exponential distance can still be special. To solve this problem, we observe that NBTAs for $\mathrm{CTL}^*_{\mathcal{ALC}}$ and $\mathrm{CTL}_{\mathcal{ALC}}$ can be used to verify Condition TL above, and that this suggests a concrete bound n_0. Specifically, let $\mathcal{A}_\varphi$ be the corresponding NBTA from Theorem 3.3 with set of states Q for the formula φ from Condition TL(c). The following shows that $n_0 := |Q|\cdot|\mathtt{tp}(\mathcal{T})|$ can serve as the required bound.

Lemma 3.4. *When choosing* $n_0 := |Q|\cdot|\mathtt{tp}(\mathcal{T})|$ *as a bound for Algorithm 1, then Property* $(*)$ *is satisfied and* satisfiable *is returned iff* C *is satisfiable w.r.t.* $\mathcal{T}$.

We demonstrate Lemma 3.4 in two steps: first, we show the satisfaction of Property $(*)$. Second, we show that Algorithm 1 is correct. The proof of the first part of Lemma 3.4 is rather subtle and involves the use of automata techniques.

The proof of the satisfaction of Property $(*)$ requires several steps. Since we plan to use the states Q of $\mathcal{A}_\varphi$ for the formula φ from Condition TL(c) to establish the correct n_0, we begin by extending the temporal types from S with states occurring in accepting runs of $\mathcal{A}_\varphi$. Then we show that the obtained $\widehat{S}$ fulfills the correspondent realizability conditions. Thereafter, we analyze the behavior of the extended types through a monotonicity lemma. Finally, based on this, we show that $n_0 := |Q|\cdot|\mathsf{tp}(\mathcal{T})|$ is the required bound.

We begin by proving the following auxiliary lemma, stating that for formulas of the form $\mathbf{A}\Box\psi$ (as the one from Condition TL(c)) we can assume all states of $\mathcal{A}_\psi$ to be initial.

Lemma 3.5. *For every CTL*-formula $\mathbf{A}\Box\psi$ with corresponding NBTA $\mathcal{A}_{\mathbf{A}\Box\psi} = (Q, \Sigma, \delta, Q^0, F)$, we can construct in time $\mathsf{poly}(|Q\times\Sigma|)$ an NBTA $\widehat{\mathcal{A}}_{\mathbf{A}\Box\psi}$ such that $L(\mathcal{A}_{\mathbf{A}\Box\psi}) = L(\widehat{\mathcal{A}}_{\mathbf{A}\Box\psi})$ and every state in $\widehat{\mathcal{A}}_{\mathbf{A}\Box\psi}$ is an initial state.*

Proof. The proof is given in two steps: first, we introduce a variant $\mathcal{A}'$ of $\mathcal{A}_{\mathbf{A}\Box\psi}$. In the second step, we define $\widehat{\mathcal{A}}_{\mathbf{A}\Box\psi}$ based on $\mathcal{A}'$.

We define $\mathcal{A}' = (Q', \Sigma, \delta', Q'^0, F')$ by setting

$$Q' = Q\times\Sigma, \qquad Q'^0 = Q^0\times\Sigma, \qquad F' = F\times\Sigma.$$

Furthermore, we define the transition relation $\delta'((q,\sigma),\sigma')$ as follows:

- if $\sigma = \sigma'$ and $\langle q_1,\dots,q_k\rangle \in \delta(q,\sigma)$, then $\langle (q_1,\sigma_1),\dots,(q_k,\sigma_k)\rangle \in \delta'((q,\sigma),\sigma)$ for all $\sigma_1,\dots,\sigma_k \in \Sigma$.
- Otherwise, if $\sigma \neq \sigma'$, then $\delta((q,\sigma),\sigma') = \emptyset$.

The definition of the transition relation δ' already provides some intuition about the behavior of $\mathcal{A}'$, that is, $\mathcal{A}'$ in state (q,σ) when reading σ behaves as $\mathcal{A}_{\mathbf{A}\Box\psi}$ does in state q when reading σ. Otherwise, $\mathcal{A}'$ stops.

Following the previous intuition, it is not hard to see that $\mathcal{A}_{\mathbf{A}\Box\psi}$ and $\mathcal{A}'$ accept the same language. Moreover, if (T,r) is an accepting run of $\mathcal{A}'$ on some Σ-labeled tree (T,τ), then for all $w \in T$ we have

$$\tau(w) = \sigma \quad \Leftrightarrow \quad \exists q_w \in Q'.r(w) = (q_w,\sigma). \tag{3.1}$$

We call a state $q \in Q'$ *active* if there is some model (T,τ) of $\mathbf{A}\Box\psi$ that admits an accepting run (T,r) of $\mathcal{A}'$ such that $r(w) = q$ for some $w \in T$. Since inactive states (states that are not active) do not participate in the accepting runs on any model of $\mathbf{A}\Box\psi$, dropping inactive states does not change the language of $\mathcal{A}'$.

Now, let $\widehat{\mathcal{A}}_{\mathbf{A}\Box\psi} = (\widehat{Q}, \Sigma, \widehat{\delta}, \widehat{Q}^0, F')$ be the variant of $\mathcal{A}'$ where

- $\widehat{Q}$ is the set of active states in Q',
- $\widehat{Q}^0 = \widehat{Q}$, that is, all states in $\widehat{\mathcal{A}}_{\mathbf{A}\Box\psi}$ are initial states.

Furthermore, we define $\widehat{\delta}(q,\sigma)$ as follows:

- $\langle q_1,\dots q_k\rangle \in \widehat{\delta}(q,\sigma)$ if $\langle q_1,\dots q_k\rangle \in \delta'(q,\sigma)$ and q_i is active for all $i \in (1,k)$.

Now, we are ready to show that $\mathcal{A}'$ and $\widehat{\mathcal{A}}_{\mathbf{A}\Box\psi}$ recognize the same language.

Claim. $L(\mathcal{A}') = L(\widehat{\mathcal{A}}_{\mathbf{A}\Box\psi})$.

Proof of the claim. The direction "$\subseteq$" is immediate.

"$\supseteq$." Assume that $\mathfrak{T} = (T,\tau) \in L(\widehat{\mathcal{A}}_{\mathbf{A}\Box\psi})$. Hence, there is an accepting run (T,r) of $\widehat{\mathcal{A}}_{\mathbf{A}\Box\psi}$ on $\mathfrak{T}$. Let $r(\varepsilon) = (q,\sigma)$. Since, by construction, (q,σ) is active, there is an accepting run (T,r') of $\mathcal{A}'$ on some Σ-labeled tree $\mathfrak{T}' = (T,\tau')$ with $r'(w) = (q,\sigma)$ for some $w \in T$. Moreover, by Equation (3.1), $\tau'(w) = \sigma$.

Now, we construct the Σ-labeled tree $\mathfrak{T}'' = (T,\tau'')$ from $\mathfrak{T}'$ by replacing the subtree rooted at w by $\mathfrak{T}$. It is not hard to see that r'' obtained from r' by replacing the subtree rooted at w by r is an accepting run of $\widehat{\mathcal{A}}_{\mathbf{A}\Box\psi}$ on $\mathfrak{T}''$: clearly the critical points are $w \in r'' \cap r'$. Note that for all $w \in r'' \cap r'$ and $r''(w) = (q,\sigma)$, (q,σ) is active, and moreover by Equation (3.1) $\tau''(w) = \sigma$. Thus, $\widehat{\delta}((q,\sigma),\sigma)$ behaves like $\delta'((q,\sigma),\sigma)$. Finally, note that all states are initial states in $\widehat{\mathcal{A}}_{\mathbf{A}\Box\psi}$; in particular, $r'(\varepsilon)$.

Further, note that, by definition of $\widehat{\mathcal{A}}_{\mathbf{A}\Box\psi}$, $\widehat{\delta} \subseteq \delta'$, then, by this fact and Equation (3.1), it is not hard to see that $\mathfrak{T}''$ is accepted by $\mathcal{A}'$ and, further $\mathfrak{T}''$ is a model of $\mathbf{A}\Box\psi$.

Finally, due to the semantics of $\mathbf{A}\Box\psi$, the subtree of $\mathfrak{T}''$ rooted at w is a model of $\mathbf{A}\Box\psi$. Hence, it is accepted by $\mathcal{A}'$. By construction, the subtree of $\mathfrak{T}''$ rooted at w is exactly $\mathfrak{T}$, thus, $\mathfrak{T}$ is accepted by $\mathcal{A}'$.

This finishes the proof of the claim.

It remains to argue that we can check if a given state is active. To this aim we construct $\mathcal{A}'' = (Q', Q'\times\Sigma, \delta'', Q'^0, F')$ on $Q'\times\Sigma$ labeled trees, which is a variant of $\mathcal{A}'$ where

$$\delta''(q,(q',\sigma)) = \begin{cases} \delta'(q,\sigma) & \text{if } q = q' \\ \emptyset & \text{otherwise.} \end{cases}$$

Intuitively, when reading a symbol (q,σ), a state q behaves in $\mathcal{A}''$ just as in $\mathcal{A}'$ when reading σ. Otherwise, it stops when reading (q',σ) for $q' \neq q$.

This implies that (T,r) is an accepting run of $\mathcal{A}'$ on some tree (T,τ) if and only if (T,r) is an accepting run of $\mathcal{A}''$ on (T,τ') where $\tau'(w) = (r(w),\tau(w))$ for all $w \in T$. Finally, we can easily devise an NBTA $\mathcal{B}_q$ that checks whether in a $Q'\times\Sigma$-tree there is some world labeled with (q,σ) for some σ. Hence, a state q is active if and only if $L(\mathcal{A}'') \cap L(\mathcal{B}_q)$ is not empty.

Since the intersection of a constant number of NBTAs can be constructed with only a polynomial blowup and the emptiness check can be decided in quadratic time, this also yields that $\widehat{\mathcal{A}}_{\mathbf{A}\Box\psi}$ can be computed in polynomial time. ❏

First, recall that S denotes the final result of Algorithm 1. We extend S to a set $\widehat{S}$ that contains temporal types annotated with states Q from the automaton $\mathcal{A}_\varphi$.

Definition 3.8. *An* extended type *for* $\mathcal{T}$ *is a triple* (t,q,i) *with* (t,i) *a temporal type for* $\mathcal{T}$ *and* $q \in Q$. *Moreover, we denote by* $\widehat{S}$ *the set of all extended types* (t,q,i) *for* $\mathcal{T}$ *satisfying the following conditions:*

– *$(t, i) \in S$;*

– *there is a 2^{cn}-labeled $\sharp_{\mathbf{E}}(\mathcal{T})$-ary tree (T, τ) and an accepting run (T, r) of $\mathcal{A}_\varphi$ on (T, τ) such that*

  – *for some $w \in T$ with $|w| = i$, we have $\tau(w) = \bar{t}$ and $r(w) = q$;*

  – *for each $w \in T$ with $|w| = j$, there is a $(t', \rho(j)) \in S$ with $\tau(w) = \overline{t'}$.*

Since the final result S of Algorithm 1 satisfies Condition TL from Definition 3.7, S is the projection of $\widehat{S}$ to the first and last component of triples. We observe the following.

Lemma 3.6. *For all $(t, q, i) \in \widehat{S}$, we have*

$\widehat{\mathsf{DL}}$. *if $\exists r.C \in t$, then there is a $(t', q', i) \in \widehat{S}$ such that $\{C\} \cup \{\neg D \mid \neg\exists r.D \in t\} \subseteq t'$;*

$\widehat{\mathsf{TL}}$. *there is a 2^{cn}-labeled $\sharp_{\mathbf{E}}(\mathcal{T})$-ary tree (T, τ) and an accepting run (T, r) of $\mathcal{A}_\varphi$ on (T, τ) such that*

a) for some $w \in T$ with $|w| = i$, we have $\tau(w) = \bar{t}$ and $r(w) = q$;

b) for each $w \in T$ with $|w| = j$, there is a $(t', q', \rho(j)) \in \widehat{S}$ with $\tau(w) = \overline{t'}$ and $r(w) = q'$.

Proof. Condition $\widehat{\mathsf{DL}}$ is immediate. For Condition $\widehat{\mathsf{TL}}$ observe that $(t, q, i) \in \widehat{S}$ because there is a 2^{cn}-labeled $\sharp_{\mathbf{E}}(\mathcal{T})$-ary tree (T, τ) and an accepting run (T, r) of $\mathcal{A}_\varphi$ such that there is some $w^* \in T$ with $|w^*| = i$, $\tau(w^*) = \bar{t}$ and $r(w^*) = q$ and for all $w \in T$ with $|w| = j$, there is $(t, \rho(j)) \in S$ with $\tau(w) = \bar{t}$. By definition of $\widehat{S}$, r and τ also witness that for all $w \in T$ we have $(t', r(w), \rho(|w|)) \in \widehat{S}$ where $\tau(w) = \bar{t}'$. Thus, τ and r together with w^* show that Condition $\widehat{\mathsf{TL}}$ is satisfied. ❑

We use the following monotonicity property to establish the n_0 for Algorithm 1 is correct. It is worth noticing that a similar property was shown for LTL$_{\mathcal{ALC}}$ [58]; the steps we follow in our proof are similar in spirit to those for the LTL$_{\mathcal{ALC}}$ case.

Let $S_0, \dots, S_m = S$ be the sets computed by Algorithm 1. Set $n_0 = |\mathsf{tp}(\mathcal{T})| \cdot |Q|$ and $\mathfrak{T}_i = \{(t, q) \mid (t, q, i) \in \widehat{S}\}$, for all $i \le n_0$.

Lemma 3.7 (monotonicity). *For all $i \le n_0$, we have*

1. *$\mathfrak{T}_{i+1} \subseteq \mathfrak{T}_i$;*
2. *$\mathfrak{T}_i = \mathfrak{T}_{i+1}$ implies $\mathfrak{T}_i = \mathfrak{T}_{i+\ell}$ for all $i + \ell \le n_0$.*

Proof. 1. Let

$$M = \widehat{S} \cup \{(t, q, j) \mid (t, q, \ell) \in \widehat{S}, \text{ for some } \ell \ge j\}.$$

We show that (1) holds in two steps: first, we show that M is consistent under Conditions $\widehat{\mathsf{DL}}$ and $\widehat{\mathsf{TL}}$. Second, based on the previous point, we show that $(t, q, i) \in M$ and $i \le n_0$ implies that $(t,i) \in S$, that is, (t,i) is not deleted in any of the iterations of Algorithm 1. By definition of M and $\widehat{S}$, the latter implies (1).

We proceed to show the first point. Let $(t,q,j) \in M$, we show it satisfies Conditions $\widehat{\mathsf{DL}}$ and $\widehat{\mathsf{TL}}$. If $(t,q,j) \in \widehat{S}$, we are done. Otherwise, there is a $(t,q,\ell) \in \widehat{S}$, for some $\ell \geq j$.

- We proceed to show that (t,q,j) satisfies Condition $\widehat{\mathsf{DL}}$. Note that Condition $\widehat{\mathsf{DL}}$ of Lemma 3.6 implies that if $\exists r.C \in t$, then there is a $(t',q',\ell) \in \widehat{S}$ witnessing Condition $\widehat{\mathsf{DL}}$ for (t,q,ℓ), that is, $\{C\} \cup \{\neg D \mid \neg\exists r.D \in t\} \subseteq t'$. Moreover, by definition of M, $(t',q',j) \in M$. We use for (t,q,j), (t',q',j) as a witness of $\exists r.C \in t$. Therefore, (t,q,j) satisfies Condition $\widehat{\mathsf{DL}}$.

- We proceed to show that (t,q,j) satisfies Condition $\widehat{\mathsf{TL}}$. Note that Condition $\widehat{\mathsf{TL}}$ of Lemma 3.6 implies that there is a 2^{cn}-labeled $\sharp_{\mathbf{E}}(\mathcal{T})$-ary tree (T,τ) and an accepting run (T,r) of $\mathcal{A}_\varphi$ on (T,τ) such that the following hold:
  a) for some world $w \in T$ with $|w| = \ell$, we have $\tau(w) = \bar{t}$ and $r(w) = q$,
  b) for all $v \in T$ with $|v| = j$, there is a $(t',q',\rho(j)) \in \widehat{S}$ with $\tau(v) = \bar{t}'$ and $r(v) = q'$.

  Now, let $w = uv$ be such that $|v| = j$ and let (T,τ') be the subtree of (T,τ) rooted at u and (T,r') be the subtree of (T,r) rooted at u. Since all states of $\mathcal{A}_\varphi$ are initial states, (T,r') is an accepting run of $\mathcal{A}_\varphi$ on (T,τ'). By construction, $\tau'(v) = \bar{t}$ and $r'(v) = q$, hence Condition $\widehat{\mathsf{TL}}(a)$ is satisfied. Also Condition $\widehat{\mathsf{TL}}(b)$ is satisfied by definition of M.

This finishes the proof of the first point. Now, we proceed to show the second point:

Claim. Let $S_M = \{(t,i) \mid (t,q,i) \in M\}$. $(t,i) \in S_M$ implies $(t,i) \in S$.

Proof of the claim. To show that $(t,i) \in S$, that is, (t,i) is not deleted by Algorithm 1, we need to show that (t,i) satisfies the DL and TL Conditions at each step of Algorithm 1.

We show this by induction on the number of iterations of Algorithm 1. For the induction base, $i = 0$, it holds since $S_0 = \mathsf{ttp}_{n_0}(\mathcal{T})$.

We show (t,i) is not deleted at the step S_i, $i > 0$.

- Condition DL. Assume that (t,i) is deleted at step S_i because it violates Condition DL, that is, $\exists r.C \in t$ and there is no $(t',i) \in S_i$ such that t' is witnesses for $\exists r.C$.

  On the other hand, we know, by definition of S_M, that there is a $(t,q,i) \in M$ for some q and, by the previous point, we moreover know that (t,q,i) satisfies Condition $\widehat{\mathsf{DL}}$, that is, there is a $(\widehat{t},q',i)$ such that $\widehat{t}$ witnesses $\exists r.C \in t$.

  Finally, note that, by definition of S_M, $(\widehat{t},i)$ is also in S_M. This leads to a contradiction, therefore (t,i) does satisfy the DL condition in S_i.

- Condition TL. Assume that (t,i) is deleted at step S_i because it violates Condition TL, that is, it violates one of points (a)-(c) from Condition TL.

  On the other hand, we know, by definition of S_M, that there is a $(t,q,i) \in M$ for some q and, by the previous point, we moreover know that (t,q,i) satisfies Condition $\widehat{\mathsf{TL}}$, that is, there is a 2^{cn}-labeled $\sharp_{\mathbf{E}}(\mathcal{T})$-ary tree (T,τ) and an accepting run (T,r) of $\mathcal{A}_\varphi$ on (T,τ) such that

1. for some $w \in T$ with $|w| = i$, we have $\tau(w) = \bar{t}$ and $r(w) = q$;
2. for each $w \in T$ with $|w| = j$, there is a $(t', q', \rho(j)) \in M$ with $\tau(w) = \overline{t'}$ and $r(w) = q'$.

Again, by definition of S_M, for all $(t, q', \rho(k))$ used to witness points 1 and 2 above $(t', \rho(k)) \in S_M$. Clearly, points (a) and (b) of Condition TL are satisfied. Finally, note that (T,r) is an accepting run of A_φ on (T,τ), then point (c) is also satisfied. This leads to a contradiction, therefore (t,i) does satisfy Condition TL in S_i.

This finishes the proof of the claim.

Now, by definition of $\widehat{S}$ and the previous claim, we have that $M = \widehat{S}$. Therefore, by definition of M, the first point of this lemma holds.

2. Assume $\mathfrak{T}_i = \mathfrak{T}_{i+1}$, and let

$$M = \widehat{S} \cup \{(t,q,j) \mid (t,q,i) \in \widehat{S} \text{ and } i \leq j \leq n_0\}.$$

As in the previous case, it is enough to check that all elements of M satisfy Conditions $\widehat{\mathsf{DL}}$ and $\widehat{\mathsf{TL}}$. Then, one can similarly proof that $(t,q,i) \in M$ implies that (t,i) is not deleted in any of the iterations of Algorithm 1. As in the previous case, by definition of $\widehat{S}$ and M, the latter implies (2).

We proceed to prove that M satisfies Conditions $\widehat{\mathsf{DL}}$ and $\widehat{\mathsf{TL}}$. For triples $(t,q,j) \in \widehat{S}$, we are done. We next show the conditions hold for triples in $M \backslash \widehat{S}$.

– We proceed to show Condition $\widehat{\mathsf{DL}}$. Let $(t,q,j) \in M\backslash\widehat{S}$. By definition of M, there is a $(t,q,i) \in \widehat{S}$ for some $i \leq j$. Note that, by Lemma 3.6, Condition $\widehat{\mathsf{DL}}$ implies that if $\exists r.C \in t$, then there is some $(t',q',i) \in \widehat{S}$ such that $\{C\} \cup \{\neg D \mid \neg\exists r.D \in t\} \subseteq t'$. By definition of M, $(t',q',j) \in M$. Then, we use for (t,q,j), (t',q,j) as a witness of $\exists r.C \in t$. Therefore, (t,q,j) satisfies Condition $\widehat{\mathsf{DL}}$.

– We proceed to show that Condition $\widehat{\mathsf{TL}}$ holds. We show that: (†) for every $(t,q,j) \in M\backslash\widehat{S}$ there is a 2^{cn}-labeled $\sharp_{\mathbf{E}}(\mathcal{T})$-ary tree (T,τ) and an accepting run (T,r) of $\mathcal{A}_\varphi$ on (T,τ) such that the following hold:

  a) for some world $w \in T$ with $|w| = j$, we have $\tau(w) = \bar{t}$ and $r(w) = q$,

  b) for all $v \in T$ with $|v| = \ell$, there is a $(t', q', \rho(\ell)) \in \widehat{S}$ with $\tau(v) = \overline{t}'$ and $r(v) = q'$.

  The proof is by induction on $i \leq j$. The cases $j = i$ and $j = i + 1$ are trivial. For the induction step, assume that (†) holds for every $(t,q,k) \in M$ for all $k < j$. Fix an arbitrary $(t,q,j) \in M$. By definition of M, $(t,q,j{-}1) \in M$; moreover, by I.H., there is a 2^{cn}-labeled $\sharp_{\mathbf{E}}(\mathcal{T})$-ary tree (T,τ) and an accepting run (T,r) of $\mathcal{A}_\varphi$ on (T,τ) such that the following hold:

a) there is a world $w \in T$ with $|w| = j - 1$ and $\tau(w) = \bar{t}$ and $r(w) = q$,

b) for all $v \in T$ with $|v| = \ell$, there is a $(t', q', \rho(\ell)) \in \widehat{S}$ with $\tau(v) = \bar{t}'$ and $r(v) = q'$.

Now, let $w = u{\cdot}c$ for some $c \in \mathbb{N}$ and $\tau(u) = \bar{t}'$ and $r(u) = q'$. Note that since $|u| = j-2$, by definition of M, we have $(t', q', j-2) \in M$ and $(t', q', j-1) \in M$. Furthermore, by I.H., there is a 2^{cn}-labeled $\sharp_{\mathbf{E}}(\mathcal{T})$-ary tree (T, τ') and an accepting run (T, r') of $\mathcal{A}_\varphi$ on (T, τ') such that the following hold:

a) there is a world $v \in T$ with $|v| = j - 1$ such that $\tau'(v) = \bar{t}'$ and $r'(v) = q'$,

b) for all $v \in T$ with $|v| = \ell$, there is a $(t'', q'', \rho(\ell)) \in \widehat{S}$ with $\tau(v) = \bar{t}''$ and $r(w) = q''$.

We define the 2^{cn}-labeled $\sharp_{\mathbf{E}}(\mathcal{T})$-ary tree (T, τ'') (and the run (T, r''), respectively) to be the tree that is obtained from (T, τ') (from (T, r'), respectively) by replacing the subtree rooted at v by the subtree of (T, τ) (of (T, r), respectively) rooted at u. Since $\tau'(v) = \tau(u)$ and $r'(v) = r(u)$, (T, r'') is an accepting run of $\mathcal{A}_\varphi$ on (T, τ''). By construction, $\tau''(v{\cdot}c) = t$, $r''(v{\cdot}c) = q$ and $|v{\cdot}c| = j$. Thus, (T, τ'') and (T, r'') satisfy condition $a)$. By construction, they also satisfy condition $b)$.

Clearly, since all the elements of M satisfy (†), then they also satisfy $\widehat{\mathsf{TL}}$.

❑

We define the infinite expansion $\widehat{S}_\omega$ of $\widehat{S}$ as $\{(t, q, i) \mid (t, q, \rho(i)) \in \widehat{S}\}$. Further, we derive the Conditions $\widehat{\mathsf{DL}'}$ and $\widehat{\mathsf{TL}'}$ from $\widehat{\mathsf{DL}}$ and $\widehat{\mathsf{TL}}$ by replacing $\rho(i)$ by i in $\widehat{\mathsf{TL}}$.

Now, we are in the position of showing with the help of Lemma 3.7 that the expansion $\widehat{S}_\omega$ of $\widehat{S}$ satisfies the required conditions.

Lemma 3.8. *Every* $(t, q, i) \in \widehat{S}_\omega$ *satisfies Conditions* $\widehat{\mathsf{DL}'}$ *and* $\widehat{\mathsf{TL}'}$.

Proof. Since $|\mathfrak{T}_0| \leq n_0$, Lemma 3.7 implies that $\mathfrak{T}_{n_0} = \emptyset$ or $\mathfrak{T}_{n_0} = \mathfrak{T}_{n_0-1}$. In the first case $\widehat{S}_\omega = \emptyset$ and we are done.

For the case $\mathfrak{T}_{n_0} = \mathfrak{T}_{n_0-1}$, we show by induction on $i \geq 0$ that all triples $(t, q, n_0 + i - 1) \in \widehat{S}_\omega$ satisfy Conditions $\widehat{\mathsf{DL}'}$ and $\widehat{\mathsf{TL}'}$. The induction base for $i = 0$ and $i = 1$ is trivial. For the induction step, let

$$M = \{(t, q, i) \mid (t, q, \ell) \in \widehat{S}_\omega \wedge \ell \leq i\}$$

and assume that Conditions $\widehat{\mathsf{DL}'}$ and $\widehat{\mathsf{TL}'}$ are satisfied by all $(t, q, j) \in M$ for $j < i$.

For the induction step, we next show that Conditions $\widehat{\mathsf{DL}'}$ and $\widehat{\mathsf{TL}'}$ are satisfied by all triples of the form $(t, q, i) \in M$.

– We proceed to show that Condition $\widehat{\mathsf{DL}'}$ holds. Let $(t, q, i) \in M$. By definition of M, there is a $(t, q, \ell) \in M$, for some $\ell \leq i$. Note that, by I.H., (t, q, ℓ) satisfies the $\widehat{\mathsf{DL}'}$ condition, that is, if $\exists r.C \in t$ then there is some $(t', q', \ell) \in M$ such that $\{C\} \cup \{\neg D \mid \neg\exists r.D \in t\} \subseteq t'$. By definition of M, $(t', q', i) \in M$. Then, we use for (t, q, i), (t', q', i) as a witness of $\exists r.C \in t$. Therefore, (t, q, i) satisfies Condition $\widehat{\mathsf{DL}'}$.

– We proceed to show that Condition $\widehat{\mathsf{TL}'}$ holds. We show that: $(\dagger)$ for every triple of the form $(t, q, i) \in M$ there is a 2^{cn}-labeled $\sharp_{\mathbf{E}}(\mathcal{T})$-ary tree (T, τ) and an accepting run (T, r) of $\mathcal{A}_\varphi$ on (T, τ) such that the following hold:

a) for some world $w \in T$ with $|w| = i$, we have $\tau(w) = \bar{t}$ and $r(w) = q$,

b) for all $v \in T$ with $|v| = \ell$, there is a $(t', q', \ell) \in \widehat{S}_\omega$ with $\tau(v) = \bar{t}'$ and $r(v) = q'$.

Fix an arbitrary $(t, q, i) \in M$. By definition of M, $(t, q, i-1) \in M$; moreover, by I.H., there is a 2^{cn}-labeled $\sharp_{\mathbf{E}}(\mathcal{T})$-ary tree (T, τ) and an accepting run (T, r) of $\mathcal{A}_\varphi$ on (T, τ) such that the following hold:

a) there is a world $w \in T$ with $|w| = i - 1$ and $\tau(w) = \bar{t}$ and $r(w) = q$,

b) for all $v \in T$ with $|v| = \ell$, there is a $(t', q', \ell) \in \widehat{S}_\omega$ with $\tau(v) = \bar{t}'$ and $r(v) = q'$.

Now, let $w = u{\cdot}c$ for some $c \in \mathbb{N}$ and $\tau(u) = \bar{t}'$ and $r(u) = q'$. Note that since $|u| = i-2$, by definition of M, we have $(t', q', i-2) \in M$ and $(t', q', i-1) \in M$. Furthermore, by I.H., there is a 2^{cn}-labeled $\sharp_{\mathbf{E}}(\mathcal{T})$-ary tree (T, τ') and an accepting run (T, r') of $\mathcal{A}_\varphi$ on (T, τ') such that

a) there is a world $v \in T$ with $|v| = i - 1$ such that $\tau'(v) = \bar{t}'$ and $r'(v) = q'$,

b) for all $v \in T$ with $|v| = \ell$, there is a $(t'', q'', \ell) \in \widehat{S}$ with $\tau(v) = \bar{t}''$ and $r(w) = q''$.

We define the 2^{cn}-labeled $\sharp_{\mathbf{E}}(\mathcal{T})$-ary tree (T, τ'') (and the run (T, r''), respectively) to be the tree that is obtained from (T, τ') (from (T, r'), respectively) by replacing the subtree rooted at v by the subtree of (T, τ) (of (T, r), respectively) rooted at u. Since $\tau'(v) = \tau(u)$ and $r'(v) = r(u)$, (T, r'') is an accepting run of $\mathcal{A}_\varphi$ on (T, τ''). By construction, $\tau''(v{\cdot}c) = t$, $r''(v{\cdot}c) = q$ and $|v{\cdot}c| = i$. Thus, (T, τ'') and (T, r'') satisfy condition $a)$. By construction, they also satisfy condition $b)$.

❏

Lemma 3.7 provides us with the right choice of n_0. We show the first part of Lemma 3.4 asserting that Property $(*)$ holds for $n_0 = |Q| \cdot |\mathsf{tp}(\mathcal{T})|$.

Lemma 3.9 (Property $(*)$)**.** *Let $n_0 = |Q| \cdot |\mathsf{tp}(\mathcal{T})|$. Then, if S is the result of Algorithm 1 and $(t, n_0) \in S$ then $(t, n_0 + \ell)$ is realizable in S, for any $\ell \geq 0$.*

Proof. We define Conditions DL' and TL', as variants of Conditions DL and TL from Definition 3.7, by replacing $\rho(i)$ in TL with i. Recall that S_ω is defined as follows:

$$S_\omega = S \cup \{(t,m) \mid (t,n_0) \in \widehat{S} \wedge m{>}n_0\}.$$

Let $(t,n_0) \in S$. Since (t,n_0) satisfies Condition TL, by the second condition of $\widehat{S}$'s definition (*cf.* Definition 3.8), there is some q such that $(t,q,n_0) \in \widehat{S}$. Now, by Lemma 3.8, for every $\ell \geq 0$, $(t,q,n_0+\ell) \in \widehat{S}_\omega$ satisfies Conditions $\widehat{\mathsf{DL}'}$ and $\widehat{\mathsf{TL}'}$. Based on this, we show the following:

– $(t,q,n_0+\ell)$ satisfies the $\widehat{\mathsf{DL}'}$ condition. Note that there is some $(t',q',n_0+\ell) \in \widehat{S}_\omega$ witnessing the Condition $\widehat{\mathsf{DL}}'$ for $(t,q,n_0+\ell)$, that is, if $\exists r.C \in t$, then $\{C\} \cup \{\neg D \mid \neg\exists r.D \in t\} \subseteq t'$. Now, by definition of $\widehat{S}_\omega$, $(t',q',n_0) \in \widehat{S}$ this implies that $(t',n_0) \in S$. Furthermore, by definition of S_ω, $(t',n_0+\ell) \in S_\omega$. We can use then for $(t,n_0+\ell)$, $(t',n_0+\ell)$ as a witness of $\exists r.C \in t$. Therefore, $(t,n_0+\ell)$ satisfies Condition DL'.

– We also have that $(t,q,n_0+\ell)$ satisfies the $\widehat{\mathsf{TL}'}$ condition, that is, there is a 2^{cn}-labeled $\sharp_{\mathbf{E}}(\mathcal{T})$-ary tree (T,τ) and an accepting run (T,r) of $\mathcal{A}_\varphi$ on (T,τ) such that

  1. for some $w \in T$ with $|w| = n_0+\ell$, we have $\tau(w) = \overline{t}$ and $r(w) = q$;
  2. for each $w \in T$ with $|w| = j$, there is a $(t',q',j) \in \widehat{S}_\omega$ with $\tau(w) = \overline{t'}$ and $r(w) = q'$.

  Now, by definition of $\widehat{S}_\omega$, each $(t',q',j) \in \widehat{S}_\omega$ witnessing condition 2 above for a world w with $|w| = j > n_0$ belongs to $\widehat{S}$ (trivially, this is also the case for the witnesses of condition 2 for a world w with $|w| = j \leq n_0$), implying that $(t',n_0) \in S$. Note that in particular $(t,n_0) \in S$. Hence, (T,τ) satisfies points $(a)-(c)$ of Condition TL'. Therefore, $(t,n_0+\ell)$ satisfies Condition TL'.

❏

Now, we can proceed to proof the second part of Lemma 3.4, that is, the correctness of Algorithm 1.

Lemma 3.10. *When choosing* $n_0 = |Q| \cdot |\mathsf{tp}(\mathcal{T})|$, *Algorithm 1 returns* satisfiable *iff* C *is satisfiable w.r.t.* $\mathcal{T}$.

Proof. "$\Rightarrow$": Let S be the result of the type elimination procedure. In the following fix $k = \sharp_{\mathbf{E}}(\mathcal{T})$. Due to Lemma 3.9, for every $(t,i) \in S_\omega$ there is a 2^{cn}-labeled k-ary tree $(T,\tau_{t,i})$ that is a model of φ from condition 2(c) of Definition 3.7. We proceed to define the temporal interpretation $\mathfrak{I} = (\Delta, T, \{\mathcal{I}_w\}_{w\in T})$ by taking $\Delta = S_\omega$ and

$$\begin{aligned} A^{\mathfrak{I},w} &= \{(t,i) \mid A \in \tau_{t,i}(w)\}; \\ r^{\mathfrak{I},w} &= \{((t,i),(t',j)) \mid X_{\neg\exists r.C} \in \tau_{t,i}(w) \text{ implies } X_{\neg C} \in \tau_{t',j}(w)\}. \end{aligned}$$

For a temporal type (t,i), we write $C \in (t,i)$ if $C \in t$.

Claim. For every $C \in \mathsf{cl}(\mathcal{T})$, $w \in W$ and $(t,i) \in \Delta$, we have that

$$(t, i) \in C^{\mathfrak{I},w} \text{ iff } X_C \in \tau_{t,i}(w),$$

for every $\pi \in \mathsf{Paths}(w)$ and path concept $\mathcal{C}$, we have that

$$(t, i) \in \mathcal{C}^{\mathfrak{I},\pi} \text{ iff } (T, \tau_{t,i}), \pi \models \overline{\mathcal{C}}$$

Proof of the claim. The proof is by a simultaneous induction on the structure of C and $\mathcal{C}$. The induction start, where C is a concept name is immediate by the definition of $\mathfrak{I}$. For the induction step we distinguish the following cases.

- $C = \neg D$ "if:" $(t, i) \in \neg D^{\mathfrak{I},w}$, that is, $(t, i) \notin D^{\mathfrak{I},w}$. Now, by I.H., $X_D \notin \tau_{t,i}(w)$. Furthermore, by Condition TL'(c), $(T, \tau_{t,i}), w \models \neg\overline{D}$. Finally, again by Condition TL'(c), $X_{\neg D} \in \tau_{t,i}(w)$.

  "only if:" $X_{\neg D} \in \tau_{t,i}(w)$, by Condition TL'(c), $(T, \tau_{t,i}), w \not\models \overline{D}$. Now, by Condition TL'(c), $X_D \notin \tau_{t,i}(w)$. By, I.H., $(t, i) \notin D^{\mathfrak{I},w}$. Therefore, $(t, i) \in (\neg D)^{\mathfrak{I},w}$.

- $C = D \sqcap E$ "if:" $(t, i) \in (D \sqcap E)^{\mathfrak{I},w}$, that is, $(t, i) \in D^{\mathfrak{I},w}$ and $(t, i) \in E^{\mathfrak{I},w}$. By I.H., $X_D \in \tau_{t,i}(w)$ and $X_E \in \tau_{t,i}(w)$. Now, by Condition TL'(c), $(T, \tau_{t,i}), w \models \overline{D}$ and $(T, \tau_{t,i}), w \models \overline{E}$. So, $(T, \tau_{t,i}), w \models \overline{D} \wedge \overline{E}$. Once again, by Condition TL'(c), $(T, \tau_{t,i}), w \models X_{D\sqcap E}$. Therefore, $X_{D\sqcap E} \in \tau_{t,i}(w)$.

  "only if:" $X_{D\sqcap E} \in \tau_{t,i}(w)$, by Condition TL'(c), we have that $(T, \tau_{t,i}), w \models \overline{D} \wedge \overline{E}$, that is, $(T, \tau_{t,i}), w \models \overline{D}$ and $(T, \tau_{t,i}), w \models \overline{E}$. Once again, by Condition TL'(c), we have that $X_D, X_E \in \tau_{t,i}(w)$. Now, by I.H., $(t, i) \in D^{\mathfrak{I},w}$ and $(t, i) \in E^{\mathfrak{I},w}$. Therefore, $(t, i) \in (D \sqcap E)^{\mathfrak{I},w}$.

- $C = \exists r.C$. "if" Follows from the fact that $\tau_{t,i}(w) = \overline{t'}$ for some $(t', |w|) \in S_\omega$. Therefore, by definition of $\overline{t'}$, $X_{\exists r.C} \in \tau_{t,i}(w)$.

  "only if:" $X_{\exists r.C} \in \tau_{t,i}(w) = \overline{t'}$ then $\exists r.C \in t'$. Now, since $(t', |w|) \in S_\omega$ then it satisfies Condition DL', that is, there is a $(t'', |w|)$ such that $\{C\} \cup \{\neg E \mid \neg\exists r.E \in t'\} \subseteq t''$. Note that, by definition of $\tau_{t'',j} = \overline{t''}$ for some j, $\{X_{\neg E} \mid X_{\neg\exists r.E} \in \overline{t'}\} \subseteq \tau_{t'',j}(w)$. Then, by definition of $\mathfrak{I}$, $((t, i)(t'', |w|)) \in r^{\mathfrak{I},w}$. Furthermore, by I.H., $(t'', |w|) \in C^{\mathfrak{I},w}$. Therefore, $(t, i) \in (\exists r.C)^{\mathfrak{I},w}$.

- $C = \mathbf{E}\mathcal{C}$ "if:" $(t, i) \in (\mathbf{E}\mathcal{C})^{\mathfrak{I},w}$. This implies that, by semantics, $(t, i) \in \mathcal{C}^{\mathfrak{I},\pi}$ for some $\pi \in \mathsf{Paths}(w)$. Now, by the second point of the claim, $(T, \tau_{t,i}), \pi \models \overline{\mathcal{C}}$. Therefore, by semantics, $(T, \tau_{t,i}), w \models \overline{\mathbf{E}\mathcal{C}}$. Since, $(T, \tau_{t,i}) \models \varphi$ from condition 2(c), then $X_{\mathbf{E}\mathcal{C}} \in \tau_{t,i}(w)$.

  "only if:" $X_{\mathbf{E}\mathcal{C}} \in \tau_{t,i}(w) = \overline{t'}$. By Condition TL'(c) we have that $(T, \tau_{t,i}), w \models \overline{\mathbf{E}\mathcal{C}}$, that is, $(T, \tau_{t,i}), \pi \models \overline{\mathcal{C}}$ for some $\pi \in \mathsf{Paths}(w)$. Now, by the second point of the claim, $(t, i) \in \mathcal{C}^{\mathfrak{I},\pi}$. Therefore, $(t, i) \in (\mathbf{E}\mathcal{C})^{\mathfrak{I},w}$.

  This finishes the proof of the first point of the claim.

  We proceed to show the second point of the claim:

– $\mathcal{C} = D$ with D a state concept. "if:" $(t,i) \in D^{\mathfrak{I},\pi[0]}$. Note that $\pi[0] = w$, then, by the first point of the claim, $(T, \tau_{t,i}), w \models \overline{D}$.

"only if:" $(T, \tau_{t,i}), \pi[0] \models \overline{D}$. By Condition TL'(c), $X_D \in \tau_{t,i}(\pi[0])$. Note that $\pi[0] = w$, then, by first point of the claim, $(t,i) \in D^{\mathfrak{I},w}$.

– $\mathcal{C} = \neg\mathcal{D}$. "if:"$(t,i) \in \neg\mathcal{D}^{\mathfrak{I},\pi}$, that is, $(t,i) \notin \mathcal{D}^{\mathfrak{I},\pi}$. By I.H. $(T, \tau_{(t,i)}), \pi \not\models \overline{\mathcal{D}}$. Therefore, $(T, \tau_{t,i}), \pi \models \neg\overline{\mathcal{D}}$.

"only if:" $(T, \tau_{t,i}), \pi \not\models \mathcal{D}$. By I.H., we have that $(t,i) \notin (\mathcal{D})^{\mathfrak{I},\pi}$. Therefore, $(t,i) \in (\neg\mathcal{D})^{\mathfrak{I},\pi}$.

– $\mathcal{C} = \mathcal{C}_1 \sqcap \mathcal{C}_2$, similar to the analogous case for state concepts.

– $\mathcal{C} = \bigcirc\mathcal{D}$. "if:" $(t,i) \in (\bigcirc\mathcal{D})^{\mathfrak{I},\pi}$, that is, $(t,i) \in \mathcal{D}^{\mathfrak{I},\pi[1]}$. Now, by I.H., $(T, \tau_{t,i}), \pi[1] \models \overline{\mathcal{D}}$. Therefore, by semantics, $(T, \tau_{t,i}), \pi \models \bigcirc\overline{\mathcal{D}}$.

'only if:" $(T, \tau_{t,i}), \pi \models \bigcirc\overline{\mathcal{D}}$. Hence, $(T, \tau_{t,i}), \pi[1] \models \overline{\mathcal{D}}$. Now, by I.H., $(t,i) \in \mathcal{D}^{\mathfrak{I},\pi[1]}$. Therefore, by semantics, $(t,i) \in (\bigcirc\mathcal{D})^{\mathcal{I},\pi}$.

– $\mathcal{C} = \Box\mathcal{D}$. "if:" $(t,i) \in (\Box\mathcal{D})^{\mathfrak{I},\pi}$, that is, for all $j \geq 0$, $(t,i) \in \mathcal{D}^{\mathfrak{I},\pi[j..]}$. Now, by I.H., $(T, \tau_{t,i}), \pi[j..] \models \overline{\mathcal{D}}$. Therefore, by semantics, $(T, \tau_{t,i}), \pi \models \Box\overline{\mathcal{D}}$.

"only if:" $(T, \tau_{t,i}), \pi \models \Box\overline{\mathcal{D}}$. This means that for all $j \geq 0$, $(T, \tau_{t,i}), \pi[j..] \models \overline{\mathcal{D}}$. Now, by I.H., for all $j \geq 0$, $(t,i) \in \mathcal{D}^{\mathfrak{I},\pi[j..]}$. Therefore, by semantics, $(t,i) \in \Box\mathcal{D}^{\mathfrak{I},\pi}$.

– $\mathcal{C} = \mathcal{C}_1\mathcal{U}\mathcal{C}_2$. "if" $\exists j \geq 0.((t,i) \in \mathcal{C}_2^{\mathfrak{I},\pi[j..]} \wedge \forall 0 \leq k < j.(t,i) \in \mathcal{C}_1^{\mathfrak{I},\pi[k..]})$. Now, by I.H., $(T, \tau_{t,i}), \pi[j..] \models \overline{\mathcal{C}}_2 \wedge \forall 0 \leq k < j.((T, \tau_{t,i}), \pi[k..] \models \overline{\mathcal{C}}_1)$. Therefore, $(T, \tau_{t,i}), \pi \models \overline{(\mathcal{C}_1\mathcal{U}\mathcal{C}_2)}$.

"only if:" $\exists j \geq 0.((T, \tau_{t,i}), \pi[j..] \models \overline{\mathcal{C}}_2 \wedge \forall 0 \leq k < j.((T, \tau_{t,i}), \pi[k..] \models \overline{\mathcal{C}}_1)$. Now, by I.H., $(t,i) \in \mathcal{C}_2^{\mathfrak{I},\pi[j..]} \wedge \forall 0 \leq k < j.(t,i) \in \mathcal{C}_1^{\mathfrak{I},\pi[k..])}$. Therefore, $(t,i) \in (\mathcal{C}_1\mathcal{U}\mathcal{C}_2)^{\mathfrak{I},\pi}$.

This finishes the proof of the second point of the claim.

Now, by definition, $C_{\mathcal{T}} \in (t,i)$ for all $(t,i) \in S_\omega$ and $X_{C_{\mathcal{T}}} \in \tau_{t,i}(w)$ for all $w \in W$. By the previous claim, $(t,i) \in C_{\mathcal{T}}^{\mathfrak{I},w}$ for all $w \in W$. Hence, $\mathfrak{I}$ is a model of $\mathcal{T}$. Finally, there is a $(t,0)$ such that $C \in t$. By definition of $\mathfrak{I}$ and the claim, $(t,0) \in C^{\mathfrak{I},\varepsilon}$.

"$\Leftarrow$": Let $\mathfrak{I} = (\Delta, T, \{\mathcal{I}_w\}_{w \in W})$ a model of C and $\mathcal{T}$. Define for every $d \in \Delta$ the $2^{\mathsf{cl}(\mathcal{T})}$-labeled tree (T, τ_d) by

$$\tau_d(w) = \{C \in \mathsf{cl}(\mathcal{T}) \mid d \in C^{\mathfrak{I},w}\}$$

and the 2^{cn}-labeled tree $\overline{\tau}_d$ by $\overline{\tau}_d(w) = \overline{\tau_d(w)}$ for all $w \in T$. Now define

$$S = \{(\tau_d(w), i) \mid w \in T, d \in \Delta, i \leq \rho(|w|)\}.$$

Now, we check that every $(t,i) \in S$ is realizable in S, that is, we need to verify that all elements of S satisfy the DL and TL conditions from Definition 3.7.

– Condition DL is immediately satisfied by definition of S.

– We proceed to show Condition TL. Let $(t, i) \in S$, then there is some $w \in T$, $d \in \Delta$ such that $t = \tau_d(w)$ and $i \leq \rho(|w|)$. Now, let $w = uv$ with $|v| = i$ (possible, since $i \leq \rho(|w|)$ and thus $i \leq |w|$). It can be easily seen that the subtree $(T, \overline{\tau}'_d)$ of $(T, \overline{\tau}_d)$ rooted at u satifies precisely the requirements of Condition TL for (t, i).

  – Condition (a) is satisfied by definition of $(T, \overline{\tau}'_d)$.
  – For condition (b) note that for all $v \in T$ with $|v| = j$, by definition of S, $(\tau_d(v), \rho(j)) \in S$. So, condition (b) is satisfied.
  – For condition (c), note that, by definition of $\overline{\tau_d}$, $C \in \tau_d(w)$ iff $X_C \in \overline{\tau_d}(w)$ for all $w \in T$. Moreover, by the following claim, we have that $C \in \tau_d(w)$ implies that $(T, \overline{\tau_d}), w \models \overline{C}$. Therefore, φ from TL(c) is satisfied.

Claim. For each $C \in \mathsf{cl}(\mathcal{T})$, $w \in W$ and $d \in \Delta$, we have that

$$C \in \tau_d(w) \text{ implies } (T, \overline{\tau_d}), w \models \overline{C},$$

for every $\pi \in \mathsf{Paths}(w)$ and path concept $\mathcal{C}$, we have that

$$d \in \mathcal{C}^{\mathfrak{I},\pi} \text{ implies } (T, \tau_d), \pi \models \overline{\mathcal{C}}.$$

Proof of the claim. The proof is by a simultaneous induction on the structure of C and $\mathcal{C}$. For concept names it follows trivially.

  – $C = \neg C$. Since $\neg C \in \tau_d(w)$, by definition of τ_d, $d \in (\neg C)^{\mathfrak{I},w}$. Hence, $d \notin C^{\mathfrak{I},w}$. Now, by I.H., $(T, \overline{\tau_d}), w \not\models \overline{C}$. Hence, $(T, \overline{\tau_d}), w \models \neg\overline{C}$.
  – $C = C \sqcap D$. Since $(C \sqcap D) \in \tau_d(w)$, by definition of τ_d, $d \in (C \sqcap D)^{\mathfrak{I},w}$. Hence, $d \in C^{\mathfrak{I},w}$ and $d \in D^{\mathfrak{I},w}$. Now, by I.H., $(T, \overline{\tau_d}), w \models \overline{C}$ and $(T, \overline{\tau_d}), w \models \overline{D}$. Therefore, $(T, \overline{\tau_d}), w \models \overline{C} \wedge \overline{D}$.
  – $C = \mathbf{E}\mathcal{C}$. Since $(C \sqcap D) \in \tau_d(w)$, by definition of τ_d, $d \in (\mathbf{E}\mathcal{C})^{\mathfrak{I},w}$. Hence, there is a $\pi \in \mathsf{Paths}(w)$ such that $d \in \mathcal{C}^{\mathfrak{I},\pi}$. Now, by the second point of the claim, $(T, \overline{\tau_d}), \pi \models \overline{\mathcal{C}}$. Therefore, $(T, \overline{\tau_d}), w \models \overline{\mathbf{E}\mathcal{C}}$.

This finishes the proof of the first point of the claim.

We proceed to prove the second point of the claim.

  – $\mathcal{C} = D$ with D a state concept. We have that $d \in D^{\mathfrak{I},\pi[0]}$. Note that $\pi[0] = w$, then by the first point of the claim $(T, \tau_d), w \models \overline{D}$.
  – $\mathcal{C} = \neg\mathcal{D}$. We have that $d \in \neg\mathcal{D}^{\mathfrak{I},\pi}$, that is, $d \notin \mathcal{D}^{\mathfrak{I},\pi}$. By I.H. $(T, \tau_d), \pi \not\models \overline{\mathcal{D}}$. Therefore, $(T, \tau_d), \pi \models \neg\overline{\mathcal{D}}$.
  – $\mathcal{C} = \mathcal{C}_1 \sqcap \mathcal{C}_2$, similar to the analogous case for state concepts.
  – $\mathcal{C} = \bigcirc\mathcal{D}$. We have that $d \in (\bigcirc\mathcal{D})^{\mathfrak{I},\pi}$, that is, $d \in \mathcal{D}^{\mathfrak{I},\pi[1]}$. Now, by I.H., $(T, \tau_d), \pi[1] \models \overline{\mathcal{D}}$. Therefore, by semantics, $(T, \tau_d), \pi \models \bigcirc\overline{\mathcal{D}}$.

– $\mathcal{C} = \Box\mathcal{D}$. We have that $d \in (\Box\mathcal{D})^{\mathfrak{I},\pi}$, that is, for all $j \geq 0$, $d \in \mathcal{D}^{\mathfrak{I},\pi[j..]}$. Now, by I.H., $(T, \tau_d), \pi[j..] \models \overline{\mathcal{D}}$. Therefore, by semantics, $(T, \tau_d), \pi \models \Box\overline{\mathcal{D}}$.

– $\mathcal{C} = \mathcal{C}_1\mathcal{U}\mathcal{C}_2$. We have that $\exists j \geq 0.(d \in \mathcal{C}_2^{\mathfrak{I},\pi[j..]} \wedge \forall 0 \leq k < j.(t,i) \in \mathcal{C}_1^{\mathfrak{I},\pi[k..]})$. Now, by I.H., $(T, \tau_d), \pi[j..] \models \overline{\mathcal{C}}_2 \wedge \forall 0 \leq k < j.((T, \tau_d), \pi[k..] \models \overline{\mathcal{C}}_1)$. Therefore, $(T, \tau_d), \pi \models \overline{(\mathcal{C}_1\mathcal{U}\mathcal{C}_2)}$.

This finishes the proof of the claim.

Now, one can easily see by induction on i that $S \subseteq S_i$ for $0 \leq i \leq m$, where $S_0, \ldots, S_m$ is the sequence computed by Algorithm 1, that is, none of the elements of S is deleted in any of the iterations of Algorithm 1. Since $\mathfrak{I}$ is a model of C, there is a $(t, 0) \in S$ with $C \in t$. Thus, Algorithm 1 returns *satisfiable*.

❏

We have not yet said how NBTAs can be used to verify Condition TL from Definition 3.7. The idea is to construct three NBTAs, one for each of the parts (a) to (c), build the intersection NBTA which accepts precisely the 2^{cn}-trees required for Condition 2, and then to perform an emptiness test. For part (c), we can simply use $\mathcal{A}_\varphi$. Moreover, it is easy to define an NBTA $\mathcal{A}_{t,i}$ with $i \leq n_0$ states that verifies the condition in part (a), and the same is true for part (b) and an NBTA $\mathcal{A}_{S_j}$ with n_0 states.

Now, it remains to show that the algorithm runs in double exponential time in the case of CTL$^*_{\mathcal{ALC}}$ and in exponential time for CTL$_{\mathcal{ALC}}$. From Theorem 3.3 and Lemma 3.7, we have that the bound n_0 is in $O(2^{2^{\mathsf{poly}(|\mathcal{T}|)}})$ for CTL$_{\mathcal{ALC}}$ and in $O(2^{\mathsf{poly}(|\mathcal{T}|)})$ for CTL$^*_{\mathcal{ALC}}$. The number of steps of the type elimination procedure is bounded by $2^{O(|\mathcal{T}|)} \cdot n_0$. The number of states in $\mathcal{A}_\varphi$ is n_0 and thus it remains to recall that the intersection of a constant number of NBTAs can be constructed with only a polynomial blowup and that emptiness can be decided in quadratic time.

Theorem 3.11. *Concept satisfiability w.r.t. TBoxes for* CTL$_{\mathcal{ALC}}$ *is* EXPTIME*-complete and* 2EXPTIME*-complete for* CTL$^*_{\mathcal{ALC}}$.

3.4 Reasoning in Fragments of CTL$_{\mathcal{EL}}$

We continue our investigation by studying the computational complexity of *fragments* of the combination of CTL with the lightweight DL $\mathcal{EL}$. Remarkably we identify a polytime fragment of CTL$_{\mathcal{EL}}$. Note that linear-time TDLs based on $\mathcal{EL}$ and LTL are computationally not very attractive as they turn out to be of the same complexity as the corresponding combination of $\mathcal{ALC}$ and LTL [7]. Moreover, we show that most of the remaining candidate fragments turn out to be hard for PSPACE or EXPTIME. The EXPTIME lower bounds are established using well-known techniques for extensions of $\mathcal{EL}$ based on the notion of non-convexity of a logic.

Formally, CTL$_{\mathcal{EL}}$ is the fragment of CTL$_{\mathcal{ALC}}$ that disallows the constructor $\neg$ (and thus also the abbreviations $C \sqcup D$ and $\forall r.C$). CTL$_{\mathcal{EL}}$ *concepts* are formed by the following grammar:

$$C, D ::= \top \mid A \mid C \sqcap D \mid \exists r.C \mid *C \mid \mathbf{P}C\mathcal{U}D \mid \mathbf{P}C\mathcal{R}D$$

where A ranges over $\mathsf{N_C}$, r ranges over $\mathsf{N_R}$, $\mathbf{P}$ is either an existential path quantifier $\mathbf{E}$ or a universal path quantifier $\mathbf{A}$, and $* \in \{\mathbf{E}\Diamond, \mathbf{A}\Diamond, \mathbf{E}\Box, \mathbf{A}\Box, \mathbf{E}\bigcirc, \mathbf{A}\bigcirc\}$. As an example, consider the following CTL$_{\mathcal{EL}}$-TBox:

$$\mathsf{PhDStudent} \sqsubseteq \mathbf{E}\Diamond(\mathsf{Phd} \sqcap \mathbf{E}\Diamond\exists\mathsf{worksFor.Uni})$$
$$\exists\mathsf{worksFor.Uni} \sqsubseteq \mathbf{E}\Diamond\mathbf{E}\Box\mathsf{Professor}$$

Intuitively, the first CI states that each student has the possibility of eventually obtaining a PhD degree and from that point possibly work in the future for a university. The second CI, states that each worker of the university has the possibility of eventually and lastingly becoming a professor.

Because of the absence of negation, satisfiability in CTL$_{\mathcal{EL}}$ is trivial in the sense that every concept is satisfiable w.r.t. every TBox. We therefore consider *subsumption* as the central reasoning problem for CTL$_{\mathcal{EL}}$: a concept D *subsumes* a concept C *w.r.t. a* $\mathrm{CTL}_{\mathcal{EL}}$ *TBox* $\mathcal{T}$, if $C^{\mathfrak{J}} \subseteq D^{\mathfrak{J}}$ for every temporal interpretation $\mathfrak{J}$ that is a model of $\mathcal{T}$. For example, the above TBox implies that every PhD student has the possible future of becoming a professor, formally $\mathcal{T} \models \mathsf{PhDStudent} \sqsubseteq \mathbf{E}\Diamond\mathsf{Professor}$.

Now, with the aim of identifying a computationally efficient branching-time TDL, we consider various fragments of CTL$_{\mathcal{EL}}$ obtained by admitting sets of temporal operators from the set

$$\{\mathbf{P}\bigcirc, \mathbf{P}\Diamond, \mathbf{P}\Box, \mathbf{P}\mathcal{U}\}.$$

For uniformity, we denote fragments of CTL$_{\mathcal{EL}}$ by putting the available temporal operators in superscript; for example, CTL$_{\mathcal{EL}}^{\mathbf{E}\Diamond,\mathbf{E}\Box}$ is CTL$_{\mathcal{EL}}$ with only the operators $\mathbf{E}\Diamond$ and $\mathbf{E}\Box$.

3.4.1 A tractable Fragment of $\mathrm{CTL}_{\mathcal{EL}}$

We identify the tractable branching-time TDL $\mathrm{CTL}_{\mathcal{EL}}^{\mathbf{E}\Diamond}$. In particular, we show that subsumption in $\mathrm{CTL}_{\mathcal{EL}}^{\mathbf{E}\Diamond}$ is in PTIME by reducing it to subsumption in the extension $\mathcal{EL}^{++}$ of $\mathcal{EL}$.

In what follows we assume that the input TBox is in the following normal form. A *basic concept* is a concept of the form $\top, A, \exists r.A, \mathbf{E}\Diamond A$ where A is a concept *name*. Now, every CI in the input TBox is required to be of the form

$$X_1 \sqcap \ldots \sqcap X_n \sqsubseteq X$$

with $X_1, \ldots, X_n, X$ basic concepts. Every TBox in $\mathrm{CTL}_{\mathcal{EL}}^{\mathbf{E}\Diamond}$ can be transformed into this normal form in polytime such that (non-)subsumption between the concept names that occur in the original TBox is preserved, *cf.* [12]. We proceed to present a straightforward reduction of CTL$_{\mathcal{EL}}^{\mathbf{E}\Diamond}$ to $\mathcal{EL}^{++}$.

First, note that $\mathcal{EL}^{++}$ allows to specify properties on roles, such as reflexivity and transitivity [13]. We introduce a fresh role name $\mathsf{succ}_\Diamond$ to represent the '*going to the future*' relation, and require that

C1 Let $d \in \Delta_i, w \in W_i$ such that $\pi_i(d,w) = e$ and $(e,f) \in r^{\mathcal{I}}$ for some $r \in \mathsf{N_R}$. Then, add a fresh element d' to Δ_i, and set $\pi_i(d',w) := f$.

C2 Let $d \in \Delta_i, w \in W_i$ such that $\pi_i(d,w) = e$ and $(e,f) \in \mathsf{succ}_{\Diamond}^{\mathcal{I}}$. Then, add a fresh world $w{\cdot}j$ to W_i, and set $\pi_i(d,w') := f$.

C3 Let $d \in \Delta_i$ and $w_0 \ldots w_n \in W_i$ such that for all $0 \le j < n$, $\pi_i(d,w_j)$ is not defined and $\pi_i(d,w_n)$ defined. Then, for all $j < n$ set $\pi_i(d',w_j) := \pi_i(d',w_k)$.

C4 Let $d \in \Delta_i$ and $\pi_i(d,w{\cdot}j)$ is not defined and $\pi_i(d,w)$ defined. Then, set $\pi_i(d,w{\cdot}j) := \pi_i(d,w)$.

Figure 3.1: Rules for the induction step of the construction of $\mathfrak{I}$

$\mathsf{succ}_{\Diamond}$ is *transitive*, *reflexive* and *total*.

Now, we obtain an $\mathcal{EL}^{++}$-TBox $\mathcal{T}'$ from a CTL$_{\mathcal{EL}}^{\mathbf{E}\Diamond}$-TBox $\mathcal{T}$ by (1) replacing every subconcept $\mathbf{E}\Diamond A$ with $\exists\mathsf{succ}_{\Diamond}.A$ and (2) stating the transitivity, reflexivity and totality of $\mathsf{succ}_{\Diamond}$.

Lemma 3.12. *Let A, B be two concept names occurring in $\mathcal{T}$. Then, $\mathcal{T} \models A \sqsubseteq B$ iff $\mathcal{T}' \models A \sqsubseteq B$.*

Proof. $\Rightarrow$) We show the contrapositive, that is, $\mathcal{T}' \not\models A \sqsubseteq B$, then $\mathcal{T} \not\models A \sqsubseteq B$. $\mathcal{T}' \not\models A \sqsubseteq B$ if and only if there is a model $\mathcal{I}$ of $\mathcal{T}'$ such that there is a $d \in A^{\mathcal{I}}$, but $d \notin B^{\mathcal{I}}$. Then, we construct a temporal model $\mathfrak{I} = (\Delta, T, \{\mathcal{I}_w\}_{w \in W})$ of $\mathcal{T}$ based on $\mathcal{I}$ such that $d \in A^{\mathfrak{I},w}$, but $d \notin B^{\mathfrak{I},w}$ for some w. From now on, w.l.o.g. we assume that $\mathcal{I}$ is tree shaped, and we use $\cdot^{\dagger}$ to denote the reduction introduced above.

We define sequences $\Delta_0, \Delta_1, \ldots$, $W_0, W_1, \ldots$ and partial mappings $\pi_0, \pi_1, \ldots$ with $\pi_i : \Delta_i \times W_i \to \Delta^{\mathcal{I}}$. We obtain our desired sets Δ, W in the limit.

To start the construction of $\mathfrak{I}$, set

- $\Delta_0 := \{d_0\}$,
- $W_0 := \{\varepsilon\}$,
- $\pi_0(d_0, \varepsilon) := d$, with d the *root* of $\mathcal{I}$.

For the *induction step*, we start by setting $\Delta_i = \Delta_{i-1}, W_i = W_{i-1}$and $\pi_i = \pi_{i-1}$, and then proceed according to the rules in Figure 3.1.

Finally, set $\Delta := \bigcup_{i \ge 0} \Delta_i$, $W := \bigcup_{i \ge 0} W_i$. The temporal interpretation $\mathfrak{I} = (\Delta, T, \{\mathcal{I}_w\}_{w \in W})$ is then given by:

$$A^{\mathfrak{I},w} = \{d \in \Delta \mid \pi(d,w) \in A^{\mathcal{I}}\};$$
$$r^{\mathfrak{I},w} = \{(d,d') \in \Delta \times \Delta \mid (\pi(d,w),\pi(d',w)) \in r^{\mathcal{I}}\}.$$

Claim: For all $d, e \in \Delta^{\mathfrak{I}}$, $w \in W$ and basic concepts C we have:

- $d \in C^{\mathfrak{I},w}$ iff $\pi(d,w) \in (C^{\dagger})^{\mathcal{I}}$;
- $(d,e) \in r^{\mathfrak{I},w}$ iff $(\pi(d,w),\pi(e,w)) \in r^{\mathcal{I}}$.

Proof of the Claim. We prove the statement by structural induction.

- $C = A \in \mathsf{CN}$ follows from definition of $\mathfrak{I}$.
- $C = C \sqcap D$. "if" $d \in (C \sqcap D)^{\mathfrak{I},w}$, that is, $d \in C^{\mathfrak{I},w}$ and $d \in D^{\mathfrak{I},w}$. Now, by I.H., $\pi(d,w) \in C^{\mathcal{I}}$ and $\pi(d,w) \in D^{\mathcal{I}}$. Therefore, $\pi(d,w) \in (C \sqcap D)^{\mathcal{I}}$. The other direction is analogous.
- $C = \exists r.A$. "if:" $d \in (\exists r.A)^{\mathfrak{I},w}$, that is, there exists a $e \in \Delta$ such that $(d,e) \in r^{\mathfrak{I},w}$ and $e \in A^{\mathfrak{I},w}$. Now, by I.H, $\pi(e,w) \in A^{\mathcal{I}}$. Moreover, by the second point of the claim, $(\pi(d,w),\pi(e,w)) \in r^{\mathcal{I}}$. Therefore, $\pi(d,w) \in (\exists r.A)^{\mathcal{I}}$.

  "only if": $\pi(d,w) \in (\exists r.A)^{\mathcal{I}}$, that is, there is $e \in \Delta^{\mathcal{I}}$ such that $(\pi(d,w),e) \in r^{\mathcal{I}}$ and $e \in A^{\mathcal{I}}$. Now, by rule **C1**, there exists a $d' \in \Delta$ such that $\pi(d',w) = e$. By I.H., $d' \in A^{\mathfrak{I},w}$ and, by the second point of the claim, $(d,d') \in r^{\mathfrak{I},w}$. Therefore, $d \in (\exists r.A)^{\mathfrak{I},w}$.
- $C = \mathbf{E}\Diamond A$. "if:" Let $\pi(d,w) \in C^{\dagger}$. Hence, $\pi(d,w) \in (\exists \mathsf{succ}_{\Diamond}.A)^{\mathcal{I}}$, that is, there is $e \in \Delta^{\mathcal{I}}$ such that $(\pi(d,w),e) \in \mathsf{succ}_{\Diamond}^{\mathcal{I}}$. By rule **C2**, there exists $w{\cdot}j \in W_i$ such that $\pi(d,w{\cdot}j) = e$. Now, by I.H., $\pi(d,w{\cdot}j) \in A^{\mathfrak{I},w\cdot j}$. Therefore, $\pi(d,w) \in (\mathbf{E}\Diamond A)^{\mathfrak{I},w}$.

  "only if:" Let $d \in (\mathbf{E}\Diamond A)^{\mathfrak{I},w}$, that is, there exists a path $\pi = w_0 \ldots w_n \ldots$ such that $w = w_0$ and $d \in A^{\mathfrak{I},w_n}$. To show that $(\pi(d,w_i),\pi(d,w_{i+1})) \in \mathsf{succ}_{\Diamond}^{\mathcal{I}}$ for $0 \le i < n$, and then by transitivity $(\pi(d,w_0),\pi(d,w_n)) \in r^{\mathcal{I}}$. Let i be arbitrary from $[0,\ldots,n]$. First note that no both $\pi(d,w_i)$ and $\pi(d,w_{i+1}))$ are defined by rule **C1**. Hence, we distinguish the following cases:

  - $\pi(d,w_{i+1})$ was defined by rule **C2**, then by definition $(\pi(d,w_i),\pi(d,w_{i+1})) \in \mathsf{succ}_{\Diamond}^{\mathcal{I}}$.
  - $\pi(d,w_{i+1})$ was defined by rule **C3**, then $\pi(d,w_i) = \pi(d,w_{i+1})$. Since $\mathsf{succ}_{\Diamond}$ is reflexive, $(\pi(d,w_i),\pi(d,w_{i+1})) \in \mathsf{succ}_{\Diamond}^{\mathcal{I}}$.
  - $\pi(d,w_{i+1})$ was defined by rule **C4**. Analogous to the previous case.

  By I.H., $\pi(d,w_n) \in A^{\mathcal{I}}$ and, by transitivity of $\mathsf{succ}_{\Diamond}$, $(\pi(d,w_0),\pi(d,w_n)) \in \mathsf{succ}_{\Diamond}^{\mathcal{I}}$. Therefore, $\pi(d,w) \in (\exists \mathsf{succ}_{\Diamond}.A)^{\mathcal{I}}$.

It remains to show that $\mathfrak{I} \models \mathcal{T}$. Let $X_1 \sqcap \ldots \sqcap X_n \sqsubseteq X \in \mathcal{T}$. Assume $d \in X_i^{\mathfrak{I},w}$ for $1 \le i \le n$. By our claim, $\pi(d,w) \in (X_i^{\dagger})^{\mathcal{I}}$. Since $\mathcal{I} \models \mathcal{T}'$ and $X_1^{\dagger} \sqcap \ldots \sqcap X_n^{\dagger} \sqsubseteq X^{\dagger} \in \mathcal{T}$, we also have $\pi(d,w) \in (X^{\dagger})^{\mathcal{I}}$. Applying again our claim, we have that $d \in X^{\mathfrak{I},w}$. Obviously, $d \in A^{\mathfrak{I},w} \backslash B^{\mathfrak{I},w}$. Therefore, $\mathcal{T} \not\models A \sqsubseteq B$.

"$\Leftarrow$:" We show the contrapositive, that is, if $\mathcal{T} \not\models A \sqsubseteq B$ then $\mathcal{T}' \not\models A \sqsubseteq B$. Let $\mathfrak{I} = (\Delta, T, \{\mathcal{I}_w\}_{w \in W})$ be a model of $\mathcal{T}$ such that $d \in A^{\mathfrak{I},w} \backslash B^{\mathfrak{I},w}$ for some $d \in \Delta$ and $w \in W$. We construct a model $\mathcal{J}$ of $\mathcal{T}'$ such that $d' \in A^{\mathcal{J}} \backslash B^{\mathcal{J}}$ for some $d' \in \Delta^{\mathcal{J}}$.

To construct $\mathcal{J}$, we define $\Delta^{\mathcal{J}} = W \times \Delta$ and $\cdot^{\mathcal{J}}$ as follows:

$$
\begin{array}{rcl}
A^{\mathcal{J}} & = & \{(w,d) \mid d \in A^{\mathfrak{I},w}\}; \\
r^{\mathcal{J}} & = & \{((w,d),(w,d')) \mid (d,d') \in r^{\mathfrak{I},w}\}; \\
\mathsf{succ}_{\diamond}^{\mathcal{J}} & = & \mathsf{cl}^{rt}(\{((w,d),(w{\cdot}i,d)) \mid d \in \Delta \wedge w, w{\cdot}i \in W\});
\end{array}
$$

where cl^{rt} is the transitive and reflexive closure of E where $T = (W, E)$.

Claim. For all $(w,d) \in \Delta^{\mathcal{J}}$ and basic concepts C we have that

$$(w,d) \in (C^{\dagger})^{\mathcal{J}} \text{ iff } d \in C^{\mathfrak{I},w}$$

Proof of the claim. The proof of the claim is by induction on the structure of C. The induction start, where C is a concept name follows directly from the definition of $\mathcal{J}$.

- $C = C \sqcap D$ "if:" $(w,d) \in (C \sqcap D)^{\mathcal{J}}$, that is, $(w,d) \in C^{\mathcal{J}}$ and $(w,d) \in D^{\mathcal{J}}$. Now, by I.H., $d \in C^{\mathfrak{I},w}$ and $d \in D^{\mathfrak{I},w}$. Therefore, $d \in (C \sqcap D)^{\mathfrak{I},w}$.

  "only if:" $d \in (C \sqcap D)^{\mathfrak{I},w}$, that is, $d \in C^{\mathfrak{I},w}$ and $d \in D^{\mathfrak{I},w}$. Now, by I.H., $(w,d) \in C^{\mathcal{J}}$ and $(w,d) \in D^{\mathcal{J}}$. Therefore, $(w,d) \in (C \sqcap D)^{\mathcal{J}}$.

- $C = \exists r.A$ "if:" $(w,d) \in (\exists r.A)^{\mathcal{J}}$, that is, there is a $(w,d') \in \Delta^{\mathcal{J}}$ with $((w,d),(w,d')) \in r^{\mathcal{J}}$ and $(w,d') \in A^{\mathcal{J}}$. By construction, $(d,d') \in r^{\mathfrak{I},w}$ and, by I.H, $d' \in A^{\mathfrak{I},w}$. Therefore, $d \in (\exists r.A)^{\mathfrak{I},w}$.

  "only if:" $d \in (\exists r.A)^{\mathfrak{I},w}$, that is, there is a $d' \in \Delta$ with $(d,d') \in r^{\mathfrak{I},w}$ and $d' \in A^{\mathfrak{I},w}$. By construction, $((w,d),(w,d')) \in r^{\mathcal{J}}$ and, by I.H., $(w,d') \in A^{\mathcal{J}}$. Therefore, $(w,d) \in (\exists r.A)^{\mathcal{J}}$.

- $C = \mathbf{E}\Diamond A$ Hence $C^{\dagger} = \exists \mathsf{succ}_{\diamond}.A$. "if:" $(w,d) \in (\exists \mathsf{succ}_{\diamond}.A)^{\mathcal{J}}$, that is, there is $(w',d') \in \Delta^{\mathcal{J}}$ with $((w,d),(w',d)) \in \mathsf{succ}^{\mathcal{J}}$ and $(w',d) \in A^{\mathcal{J}}$. By construction, w' is a successor of w, and by I.H., $d' \in A^{\mathfrak{I},w'}$. Therefore, $d \in (\mathbf{E}\Diamond A)^{\mathfrak{I},w}$.

  "only if:" $d \in (\mathbf{E}\Diamond A)^{\mathfrak{I},w}$, that is, there is a path $\pi = w_0 \ldots w_n \ldots$ such that $w = w_0$ and $d \in A^{\mathfrak{I},w_n}$. By construction, $((w,d),(w_n,d)) \in \mathsf{succ}_{\diamond}^{\mathcal{J}}$, and by I.H., $(w_n,d) \in A^{\mathcal{J}}$. Therefore, $(w,d) \in (\exists \mathsf{succ}_{\diamond}.A)^{\mathcal{J}}$.

It remains to show that $\mathfrak{I} \models \mathcal{T}'$. Let $X_1^{\dagger} \sqcap \ldots \sqcap X_n^{\dagger} \sqsubseteq X^{\dagger} \in \mathcal{T}'$. Assume $(w,d) \in (X_i^{\dagger})^{\mathcal{J}}$ for $1 \leq i \leq n$. By our claim, $d \in (X_i^{\dagger})^{\mathfrak{I},w}$. Since $\mathfrak{I} \models \mathcal{T}$ and $X_1 \sqcap \ldots \sqcap X_n \sqsubseteq X \in \mathcal{T}$, we also have $d \in (X^{\dagger})^{\mathfrak{I},w}$. Applying again our claim, we have that $(w,d) \in X^{\mathcal{J}}$. Obviously, $(w,d) \in A^{\mathcal{J}} \backslash B^{\mathcal{J}}$. Therefore, $\mathcal{T}' \not\models A \sqsubseteq B$. ❑

Since concept subsumption w.r.t. TBoxes for $\mathcal{EL}^{++}$ can be decided in PTIME [12], we obtain the desired result.

Theorem 3.13. *Concept subsumption w.r.t. TBoxes for* $\mathrm{CTL}_{\mathcal{EL}}^{\mathbf{E}\Diamond}$ *can be decided in* PTIME.

We note that this is the first temporal description logic based on $\mathcal{EL}$ that turns out to admit PTIME reasoning; see also [7]. While the expressive power of $\mathrm{CTL}_{\mathcal{EL}}^{\mathbf{E}\Diamond}$ is clearly rather restricted, we believe that it might still be sufficient for some applications.

3.4.2 Intractable Fragments of $\mathrm{CTL}_{\mathcal{EL}}$

We show that $\mathrm{CTL}_{\mathcal{EL}}^{\mathbf{E}\Diamond}$ is a maximal tractable fragment of $\mathrm{CTL}_{\mathcal{ALC}}$ in the sense that adding further temporal operators destroys tractability. We start with the extension $\mathrm{CTL}_{\mathcal{EL}}^{\mathbf{E}\Diamond,\mathbf{A}\Box}$ and prove that subsumption becomes PSPACE-hard by a reduction of QBF validity.

Theorem 3.14. *Concept subsumption w.r.t. TBoxes for* $\mathrm{CTL}_{\mathcal{EL}}^{\mathbf{E}\Diamond,\mathbf{A}\Box}$ *is* PSPACE*-hard.*

Proof. The proof is by reduction of the validity problem for quantified Boolean formulas which is PSPACE-complete [68]. A *quantified Boolean formula (QBF)* Ψ is of the form

$$\Psi = Q_1 x_1 \dots Q_n x_n . \varphi$$

where $Q_i \in \{\exists, \forall\}$ for $1 \leq i \leq n$ and φ is a Boolean formula with only variables $x_1, \dots, x_n$. The validity of QBFs is defined inductively:

$$\exists x.\Psi \text{ is } \textit{valid} \text{ if } \Psi[x/\mathsf{true}] \text{ or } \Psi[x/\mathsf{false}] \text{ is valid.}$$

$$\forall x.\Psi \text{ is } \textit{valid} \text{ if } \Psi[x/\mathsf{true}] \text{ and } \Psi[x/\mathsf{false}] \text{ are valid.}$$

Given a QBF Ψ, our aim is to construct in polynomial time a $\mathrm{CTL}_{\mathcal{EL}}^{\mathbf{E}\Diamond,\mathbf{A}\Box}$ TBox $\mathcal{T}_\Psi$ such that for certain concept names A_0, B_0, we have $\mathcal{T}_\Psi \models A_0 \sqsubseteq B_0$ iff Ψ is valid. We use the following signature:

- concepts names X_φ for every subformula φ of Ψ, In particular, $X_{\chi\wedge\theta}$ for all subformulas $\chi \wedge \theta$ and $X_{\chi\vee\theta}$ for all subformulas $\chi \vee \theta$.
- concept names X_{x_i} and $X_{\neg x_i}$ for each variable x_i.
- concept names $\mathsf{L}_1, \dots, \mathsf{L}_n$ to distinguish the levels of a binary tree of depth n.

We begin by constructing a binary tree of depth n rooted at L_0 representing all possible evaluations of $\{x_1, \dots, x_n\}$.

$$\mathsf{L}_i \sqsubseteq \mathbf{E}\Diamond(\mathsf{L}_{i+1} \sqcap X_{x_{i+1}}) \sqcap \mathbf{E}\Diamond(\mathsf{L}_{i+1} \sqcap X_{\neg x_{i+1}})$$

The truth value of a variable is kept through all descendants.

$$\begin{aligned} X_{x_i} &\sqsubseteq \mathbf{A}\Box X_{x_i} \text{ for all } 0 \leq i \leq n \\ X_{\neg x_i} &\sqsubseteq \mathbf{A}\Box X_{\neg x_i} \text{ for all } 0 \leq i \leq n \end{aligned}$$

We evaluate the subformulas in each leaf of the tree.

$$\mathsf{L}_n \sqcap X_\chi \sqcap X_\theta \sqsubseteq X_{\chi\wedge\theta} \text{ for all subformulas } \chi \wedge \theta \text{ of } \Psi$$
$$\mathsf{L}_n \sqcap X_\chi \text{ and } \mathsf{L}_n \sqcap X_\theta \sqsubseteq X_{\chi\vee\theta} \text{ for all subformulas } \chi \vee \theta \text{ of } \Psi$$

To evaluate the formula we proceed from the leafs to the root. We identify each level with a quantifier $\exists$ or $\forall$ in Ψ.
For every $1 \leq i \leq n$ with $Q_i = \exists$ we have

$$\begin{aligned} \mathsf{L}_{i-1} \sqcap \mathbf{E}\Diamond(\mathsf{L}_i \sqcap X_{x_i} \sqcap X_\varphi) &\sqsubseteq X_\varphi \\ \mathsf{L}_{i-1} \sqcap \mathbf{E}\Diamond(\mathsf{L}_i \sqcap X_{\neg x_i} \sqcap X_\varphi) &\sqsubseteq X_\varphi \end{aligned}$$

while for $Q_i = \forall$ we have that

$$\mathsf{L}_{i-1} \sqcap \mathbf{E}\Diamond(\mathsf{L}_i \sqcap X_{x_i} \sqcap X_\varphi) \sqcap \mathbf{E}\Diamond(\mathsf{L}_i \sqcap X_{\neg x_i} \sqcap X_\varphi) \sqsubseteq X_\varphi$$

This finishes the construction of $\mathcal{T}_\Psi$ which contains the CIs introduced above. Following the intuitions provided before, it is clear that Ψ is valid iff $\mathcal{T}_\Psi \models \mathsf{L}_0 \sqsubseteq X_\Psi$.

❏

The remaining candidate operators for extending $\mathrm{CTL}^{\mathbf{E}\Diamond}_{\mathcal{EL}}$ are $\mathbf{E}\bigcirc$, $\mathbf{A}\bigcirc$, $\mathbf{A}\Diamond$, $\mathbf{E}\mathcal{U}$, $\mathbf{A}\mathcal{U}$. It turns out that subsumption is EXPTIME-complete in any of the resulting extensions.

Theorem 3.15. *Subsumption is* EXPTIME*-complete in*

(a) $\mathrm{CTL}^{\mathbf{A}\Diamond,\mathbf{E}\Diamond}_{\mathcal{EL}}$	(b) $\mathrm{CTL}^{\mathbf{E}\Diamond,\mathbf{E}\bigcirc}_{\mathcal{EL}}$	(c) $\mathrm{CTL}^{\mathbf{A}\Diamond,\mathbf{A}\bigcirc}_{\mathcal{EL}}$
(d) $\mathrm{CTL}^{\mathbf{E}\mathcal{U}}_{\mathcal{EL}}$	(e) $\mathrm{CTL}^{\mathbf{A}\mathcal{U}}_{\mathcal{EL}}$	(f) $\mathrm{CTL}^{\mathbf{A}\bigcirc}_{\mathcal{EL}}$
(g) $\mathrm{CTL}^{\mathbf{E}\Diamond,\mathbf{E}\Box}_{\mathcal{EL}}$	(h) $\mathrm{CTL}^{\mathbf{E}\Diamond^<,\mathbf{E}\Box^<}_{\mathcal{EL}}$	(i) $\mathrm{CTL}^{\mathbf{E}\mathcal{U}^<}_{\mathcal{EL}}$

The upper bounds are obvious since all listed TDLs are fragments of $\mathrm{CTL}_{\mathcal{ALC}}$. To prove the matching lower bound for a listed fragment, it suffices to show that it is *non-convex*, and then use standard techniques introduced by Baader *et al.* [12] to reduce satisfiability w.r.t. TBoxes in $\mathcal{ALC}$ to subsumption in the corresponding fragment.

Definition 3.9. *A fragment* $\mathrm{CTL}^{-}_{\mathcal{EL}}$ *of* $\mathrm{CTL}_{\mathcal{EL}}$ *is* non-convex *if there are a* $\mathrm{CTL}^{-}_{\mathcal{EL}}$ *TBox* $\mathcal{T}$ *and concepts* $C, D_1, \ldots D_n$*,* $n \geq 2$ *such that* $\mathcal{T} \models C \sqsubseteq D_1 \sqcup \cdots \sqcup D_n$ *but* $\mathcal{T} \not\models C \sqsubseteq D_i$ *for all* i*.*

Consider the fragment $\mathrm{CTL}^{\mathbf{A}\Diamond,\mathbf{E}\Diamond}_{\mathcal{EL}}$, then set

$$\mathcal{T} = \emptyset,\ C = \mathbf{A}\Diamond A \sqcap \mathbf{A}\Diamond B,\ D_1 = \mathbf{E}\Diamond(A \sqcap \mathbf{E}\Diamond B),\ D_2 = \mathbf{E}\Diamond(B \sqcap \mathbf{E}\Diamond A)$$

Now, we show that the above indeed witnesses non-convexity.

Lemma 3.16. *$\mathcal{T} \models C \sqsubseteq \bigsqcup D_i$ but $\mathcal{T} \not\models C \sqsubseteq D_i$ for $1 \leq i \leq 2$.*

Proof. For the former, let $\mathfrak{I}$ a model of $\mathcal{T}$, $d \in C^{\mathfrak{I},w}$ for some $w \in W$. Since $d \in (\mathbf{A}\Diamond A \sqcap \mathbf{A}\Diamond B)^{\mathfrak{I},w}$ for each $\pi \in \mathsf{Paths}(w)$, there are $j \geq 0$ and $k \geq 0$ such that $d \in A^{\mathfrak{I},\pi[j]}$ and $d \in B^{\mathfrak{I},\pi[k]}$. Now, for a path $\pi' \in \mathsf{Paths}(w)$, depending on whether $k \leq j$ or $j \leq k$ implies $d \in D_1^{\mathfrak{I},w}$ or $d \in D_2^{\mathfrak{I},w}$.

For the latter, we construct a temporal model $\mathfrak{I} = (\Delta, T, \{\mathcal{I}_w\}_{w \in W})$ with $\Delta = \{d\}$ and T a 1-ary tree such that $w_1 = \varepsilon{\cdot}1$ and $w_2 = w_1{\cdot}1$, such that $d \in C^{\mathfrak{I},\varepsilon}$ and $d \notin D_2^{\mathfrak{I},\varepsilon}$ by setting

$$A^{\mathfrak{I},w_1} := \{d\}; \qquad A^{\mathfrak{I},w} := \emptyset, \text{ for } w \neq w_1;$$

$$B^{\mathfrak{I},w_2} := \{d\}; \qquad B^{\mathfrak{I},w} := \emptyset, \text{ for } w \neq w_2.$$

It is clear that $d \in (\mathbf{A}\Diamond A \sqcap \mathbf{A}\Diamond B)^{\mathfrak{I},\varepsilon}$ but $d \notin (\mathbf{E}\Diamond(B \sqcap \mathbf{E}\Diamond A))^{\mathfrak{I},\varepsilon}$. Following the previous ideas we can also construct a model $\mathfrak{I}'$ such that $d \in C^{\mathfrak{I}',\varepsilon}$ but $d \notin D_1^{\mathfrak{I}',\varepsilon}$. ❏

Now that we have established the non-convexity of $\mathrm{CTL}_{\mathcal{EL}}^{\mathbf{A}\Diamond,\mathbf{E}\Diamond}$ a standard proof technique from [12, 7] can be used to show EXPTIME-hardness. More precisely, we show the lower bound by a reduction of the satisfiability problem w.r.t. TBoxes in $\mathcal{ALC}$ which is known to be EXPTIME-complete [14].

Lemma 3.17. *Concept subsumption w.r.t. TBoxes for $\mathrm{CTL}_{\mathcal{EL}}^{\mathbf{A}\Diamond,\mathbf{E}\Diamond}$ is EXPTIME-hard.*

Proof. The proof is by reduction of the satisfiability problem w.r.t. TBoxes in $\mathcal{ALC}$. Suppose that an $\mathcal{ALC}$ TBox $\mathcal{T}$ and a concept name A_0 are given for which satisfiability is to be decided. First, we manipulate the TBox $\mathcal{T}$ with some satisfiability preserving operations:

* Ensure that negation $\neg$ occurs *only* in front of concept names: for every subconcept $\neg C$ in $\mathcal{T}$ with C complex, introduce a fresh concept name A, replace $\neg C$ with $\neg A$, and add $A \sqsubseteq C$ and $C \sqsubseteq A$ to $\mathcal{T}$.

* Eliminate negation: for every subconcept $\neg A$, introduce a fresh concept name $\overline{A}$, replace every occurrence of $\neg A$ with $\overline{A}$, and add $\top \sqsubseteq A \sqcup \overline{A}$ and $A \sqcap \overline{A} \sqsubseteq \bot$ to $\mathcal{T}$.

* Eliminate disjunction: modulo introduction of new concept names, we may assume that $\sqcup$ occurs in $\mathcal{T}$ only in the form

  $$\text{(i) } A \sqcup B \sqsubseteq C \text{ and (ii) } C \sqsubseteq A \sqcup B,$$

  where A and B are concept names and C is disjunction free.

  The former kind of inclusion is replaced with $A \sqcap \mathsf{M} \sqsubseteq C$ and $\mathsf{M} \sqcap B \sqsubseteq C$. The latter one is replaced with

$$\begin{aligned}\mathsf{M} \sqcap C &\sqsubseteq \mathbf{A}\Diamond\mathsf{X} \sqcap \mathbf{A}\Diamond\mathsf{Y}\\ \mathsf{M} \sqcap C \sqcap \mathbf{E}\Diamond(\mathsf{X} \sqcap \mathbf{E}\Diamond\mathsf{Y}) &\sqsubseteq A\\ \mathsf{M} \sqcap C \sqcap \mathbf{E}\Diamond(\mathsf{Y} \sqcap \mathbf{E}\Diamond\mathsf{X}) &\sqsubseteq B\end{aligned}$$

where $\mathsf{M}, \mathsf{X}, \mathsf{Y}$ are fresh concept names.

Moreover, we apply the following replacement:

– for each CI $C \sqsubseteq D \in \mathcal{T}$, replace C in $\mathcal{T}$ with $C \sqcap \mathsf{M}$ and replace every subconcept $\exists r.E$ of D with $\exists r.(E \sqcap \mathsf{M})$.

Let $\mathcal{T}'$ be the $\mathrm{CTL}_{\mathcal{EL}_\bot}^{\mathbf{E}\Diamond,\mathbf{A}\Diamond}$ TBox obtained by these manipulations, that is, it is the extension of $\mathrm{CTL}_{\mathcal{EL}}^{\mathbf{E}\Diamond,\mathbf{A}\Diamond}$ with bottom $(\bot)$.

Claim. A_0 is satisfiable w.r.t. $\mathcal{T}$ iff $A_0 \sqcap \mathsf{M}$ is satisfiable w.r.t. $\mathcal{T}'$.

Proof of the claim. Clearly, if $A_0 \sqcap \mathsf{M}$ is satisfiable w.r.t. $\mathcal{T}$', A_0 is satisfiable w.r.t. $\mathcal{T}$.

For the other direction, we assume that A_0 is satisfiable w.r.t. $\mathcal{T}$. Let $\mathcal{I}$ be a model of A_0 and $\mathcal{T}$, and consider a CI $C \sqsubseteq A \sqcup B$. We construct a temporal interpretation $\mathfrak{I} = (\Delta, T, \{\mathcal{I}_w\}_{w \in W})$ with $\Delta = \Delta^\mathcal{I}$ as follows:

(I) for all $w \in W$,

$$A^{\mathfrak{I},w} = A^\mathcal{I} \text{ for all } A \in \mathsf{N_C} \backslash \{\mathsf{M}, \mathsf{X}, \mathsf{Y}\};$$
$$r^{\mathfrak{I},w} = r^\mathcal{I} \text{ for all } r \in \mathsf{N_R}.$$

(II) $\mathsf{M}^{\mathfrak{I},\varepsilon} = \Delta$ and $\mathsf{M}^{\mathfrak{I},w} = \emptyset$, for all $w \neq \varepsilon$.

(III) Assume $d \in C^\mathcal{I}$, then $d \in (A \sqcup B)^\mathcal{I}$, we distinguish the following cases:

1. if $d \in B^\mathcal{I}$, include $d \in \mathsf{Y}^{\mathfrak{I},\varepsilon}$ and $d \in \mathsf{X}^{\mathfrak{I},w}$ for all w of the form $\varepsilon{\cdot}i$;
2. if $d \in A^\mathcal{I} \backslash B^\mathcal{I}$, include $d \in \mathsf{X}^{\mathfrak{I},\varepsilon}$ and $d \in \mathsf{Y}^{\mathfrak{I},w}$ for all w of the form $\varepsilon{\cdot}i$.

Claim. For all concepts C and $d \in \Delta$

$$d \in \widehat{C}^{\mathfrak{I},\varepsilon} \text{ iff } d \in C^\mathcal{I}$$

where $\widehat{C}$ is obtained by the above replacement. This claim can be readily checked: note that neither X nor Y occur in $\widehat{C}$ and moreover $\mathsf{M}^{\mathfrak{I},\varepsilon} = \Delta = \Delta^\mathcal{I}$.

Claim. $\mathfrak{I}$ is a model of $\mathcal{T}'$.

We show that $\mathfrak{I}$ satisfies each CI of $\mathcal{T}$'. We focus on showing that $\mathfrak{I}$ satisfies the CIs introduced to replace CIs of the form $A \sqcup B \sqsubseteq C$ and $C \sqsubseteq A \sqcup B$. First, we note that, for all $w \neq \varepsilon$, because of the above replacement and $\mathsf{M}^{\mathfrak{I},w} = \emptyset$, the CIs are trivially satisfied in such worlds w. We take a look at the case for ε.

– '$A \sqcap \mathsf{M} \sqsubseteq C$'. Assume $d \in (A \sqcap \mathsf{M})^{\mathfrak{I},\varepsilon}$, that is, $d \in A^{\mathfrak{I},\varepsilon}$. Now, by the previous claim, $d \in A^{\mathcal{I}}$. Moreover, since $\mathcal{T} \models A \sqcup B \sqsubseteq C$, $d \in C^{\mathcal{I}}$. Again by the previous claim, $d \in \widehat{C}^{\mathfrak{I},\varepsilon}$.

– '$\mathsf{M} \sqcap C \sqsubseteq (\mathbf{A}\Diamond\mathsf{X}\sqcap\mathbf{A}\Diamond\mathsf{Y})$'. Assume $d \in (\mathsf{M} \sqcap C)^{\mathfrak{I},\varepsilon}$, that is, $d \in C^{\mathfrak{I},\varepsilon}$. Now, by the previous claim, $d \in C^{\mathcal{I}}$. Since $\mathcal{T} \models C \sqsubseteq A \sqcup B$, then $d \in (A \sqcup B)^{\mathcal{I}}$, that is, $d \in A^{\mathcal{I}}$ or $d \in B^{\mathcal{I}}$. We can distinguish two cases according to (III):

  * if $d \in B^{\mathcal{I}}$, we have that $d \in \mathsf{Y}^{\mathfrak{I},\varepsilon}$ and $d \in \mathsf{X}^{\mathfrak{I},w}$ for all w of the form $\varepsilon{\cdot}i$. Therefore, $d \in (\mathbf{A}\Diamond\mathsf{X})^{\mathfrak{I},\varepsilon}$ and $d \in (\mathbf{A}\Diamond\mathsf{Y})^{\mathfrak{I},\varepsilon}$. Hence, $d \in (\mathbf{A}\Diamond\mathsf{X} \sqcap \mathbf{A}\Diamond\mathsf{Y})^{\mathfrak{I},\varepsilon}$.
  * Analogous to the previous one.

– '$\mathsf{M} \sqcap C \sqsubseteq \mathbf{E}\Diamond(\mathsf{X} \sqcap \mathbf{E}\Diamond Y)$'. Assume $d \in (\mathsf{M} \sqcap C \sqcap \mathbf{E}\Diamond(\mathsf{X} \sqcap \mathbf{E}\Diamond Y))^{\mathfrak{I},\varepsilon}$, that is, $d \in C^{\mathfrak{I},\varepsilon}$ and $d \in \mathbf{E}\Diamond(\mathsf{X} \sqcap \mathbf{E}\Diamond Y)^{\mathfrak{I},\varepsilon}$. Now, by the previous claim, $d \in C^{\mathcal{I}}$. Since $\mathcal{T} \models C \sqsubseteq A \sqcup B$, we have that $d \in A^{\mathcal{I}}$ or $d \in B^{\mathcal{I}}$, so either case 1 or 2 of (III) is applied.

  Note that case 1 cannot be applied, otherwise $d \notin \mathbf{E}\Diamond(\mathsf{X} \sqcap \mathbf{E}\Diamond Y)^{\mathfrak{I},\varepsilon}$. By applying 2, we have that $d \in \mathsf{X}^{\mathfrak{I},\varepsilon}$ and $d \in \mathsf{X}^{\mathfrak{I},w}$ for all w of the form $\varepsilon{\cdot}i$. Then, $d \in \mathbf{E}\Diamond(\mathsf{X} \sqcap \mathbf{E}\Diamond Y))^{\mathfrak{I},\varepsilon}$. Therefore, by construction, $d \in A^{\mathcal{I}}$, and by the previous claim, $d \in A^{\mathfrak{I},\varepsilon}$.

– '$\mathsf{M} \sqcap C \sqsubseteq \mathbf{E}\Diamond(\mathsf{Y} \sqcap \mathbf{E}\Diamond X)$'. Analogous to the previous one.

For the remaining CIs, it readily follows, from the previous claim and the fact that $\mathsf{M}^{\mathfrak{I},\varepsilon} = \Delta$, that $\mathfrak{I}$ satisfies them. Analogously, by the previous claim, $d \in (A_0 \sqcap \mathsf{M})^{\mathfrak{I},\varepsilon}$.

This finishes the proof of the claim.

* The TBox $\mathcal{T}'$ contains only the operators $\sqcap$, $\exists$, $\top$, $\bot$, and $\mathbf{E}\Diamond, \mathbf{A}\Diamond$. We now reduce satisfiability of A_0 w.r.t. $\mathcal{T}'$ to (non-)subsumption in $\mathrm{CTL}_{\mathcal{EL}}^{\mathbf{E}\Diamond,\mathbf{A}\Diamond}$. To this aim we use the reduction proposed by Baader *et al.* [12] for the extension of $\mathcal{EL}$ with $\bot$.

  Introduce a fresh concept name L and replace every occurrence of $\bot$ with L and extend $\mathcal{T}'$ with

$$\exists r.\mathsf{L} \sqsubseteq \mathsf{L}, \text{ for every role } r \text{ from } \mathcal{T}';$$
$$\mathbf{E}\Diamond\mathsf{L} \sqsubseteq \mathsf{L}.$$

  It is not hard to see that A_0 is satisfiable w.r.t. $\mathcal{T}'$ iff $\mathcal{T}'' \not\models A_0 \sqsubseteq \mathsf{L}$.

❏

One would expect that $\mathrm{CTL}_{\mathcal{EL}}^{\mathbf{E}\Diamond,\mathbf{E}\Box}$ could remain tractable since both temporal operators are existentially quantified and thus it could be seen in principle as a temporal analog of $\mathcal{EL}$. Alas, we show that this is not the case. Consider the following setting:

$$\mathcal{T} = \{C \sqsubseteq \mathbf{E}\Diamond(D \sqcap A),\ D \sqsubseteq \mathbf{E}\Diamond(C \sqcap A)\}$$
$$D_1 = \mathbf{E}\Diamond(C \sqcap D),\ D_2 = \mathbf{E}\Box\mathbf{E}\Diamond A.$$

$\mathrm{CTL}_{\mathcal{EL}}^{\mathbf{E}\Diamond,\mathbf{E}\bigcirc}$	$\mathcal{T} = \emptyset$, $C = \mathbf{E}\Diamond A$, $D_1 = A$, $D_2 = \mathbf{E}\bigcirc\mathbf{E}\Diamond A$
$\mathrm{CTL}_{\mathcal{EL}}^{\mathbf{E}\mathcal{U}}$	$\mathcal{T} = \emptyset$, $C = \mathbf{E}(A\mathcal{U}B)$, $D_1 = B$, $D_2 = A$
$\mathrm{CTL}_{\mathcal{EL}}^{\mathbf{E}\Diamond,\mathbf{E}\Box}$	$\mathcal{T} = \{C \sqsubseteq \mathbf{E}\Diamond D \sqcap A,\ D \sqsubseteq \mathbf{E}\Diamond C \sqcap A\}$ $D_1 = \mathbf{E}\Diamond(C \sqcap D)$, $D_2 = \mathbf{E}\Box A$.
$\mathrm{CTL}_{\mathcal{EL}}^{\mathbf{E}\Diamond^<,\mathbf{E}\Box^<}$	$\mathcal{T} = \{C \sqsubseteq \mathbf{E}\Diamond^< B,\ B \sqsubseteq D \sqcap \mathbf{E}\Box^< D\}$ $D_1 = \mathbf{E}\Diamond^<\mathbf{E}\Diamond^< B$, $D_2 = \mathbf{E}\Box^< D$.
$\mathrm{CTL}_{\mathcal{EL}}^{\mathbf{A}\Diamond,\mathbf{A}\bigcirc}$	Analogous to the corresponding **E**-setting above.
$\mathrm{CTL}_{\mathcal{EL}}^{\mathbf{A}\mathcal{U}}$	Analogous to the corresponding **E**-setting above.

Figure 3.2: Non-convexity witness

We show that the above setting witnesses non-convexity

Lemma 3.18. $\mathcal{T} \models C \sqsubseteq \bigsqcup D_i$ *but* $\mathcal{T} \not\models C \sqsubseteq D_i$ *for* $1 \leq i \leq 2$.

Proof. For the former, let $\mathfrak{I}$ be a model of $\mathcal{T}$ and $d \in C^{\mathfrak{I},w}$ for some $w \in W$. Since

$$d \in (\mathbf{E}\Diamond(D \sqcap A))^{\mathfrak{I},w} \text{ and } D \sqsubseteq \mathbf{E}\Diamond(C \sqcap A)$$

there exists a $j \geq 0$ such that $d \in (D \sqcap A \sqcap \mathbf{E}\Diamond(C \sqcap A))^{\mathfrak{I},\pi[j]}$ for some $\pi \in \mathsf{Paths}(w)$. Then, by semantics, there exists a $k \geq j$ such that $d \in (C \sqcap A)^{\mathfrak{I},\pi'[k]}$ for some $\pi' \in \mathsf{Paths}(\pi[j])$. We can distinguish two cases (1) $k = j$ or (2) $k > j$:

1. if the first case holds, then $d \in (C \sqcap D \sqcap A)^{\mathfrak{I},\pi[j]}$. Therefore, $d \in \mathbf{E}\Diamond(C \sqcap D)^{\mathfrak{I},\pi[\varepsilon]}$;
2. if the second case holds, then $d \in (C \sqcap \mathbf{E}\Diamond(D \sqcap A))^{\mathfrak{I},\pi'[k]}$.

Clearly, if we are in the second case, then the same two cases can be distinguish again. Hence, if always the second case holds, then $d \in (\mathbf{E}\Box\mathbf{E}\Diamond A)^{\mathfrak{I},w}$. Therefore, $\mathcal{T} \models C \sqsubseteq \mathbf{E}\Diamond(C \sqcap D) \sqcup \mathbf{E}\Box\mathbf{E}\Diamond A$.

For the latter, we construct a temporal model $\mathfrak{I} = (\Delta, T, \{\mathcal{I}_w\}_{w \in W})$ of $\mathcal{T}$ with $\Delta = \{d\}$ and T

a 1-ary tree with $w_1 = \varepsilon{\cdot}1$ such that $d \in C^{\mathfrak{I},\varepsilon}$ and $d \notin D_2^{\mathfrak{I},\varepsilon}$ by setting

$$\begin{array}{ll} C^{\mathfrak{I},\{\varepsilon,w_1\}} := \{d\}; & C^{\mathfrak{I},w} := \emptyset, \text{ for } w \notin \{\varepsilon, w_1\}; \\ A^{\mathfrak{I},\{\varepsilon,w_1\}} := \{d\}; & A^{\mathfrak{I},w} := \emptyset; \text{ for } w \notin \{\varepsilon, w_1\}; \\ D^{\mathfrak{I},w_1} := \{d\}; & D^{\mathfrak{I},w} := \emptyset, \text{ for } w \neq w_1. \end{array}$$

Clearly, $\mathfrak{I}$ is a model of $\mathcal{T}$. Now, since $d \in (C \sqcap D)^{\mathfrak{I},w_1}$, by semantics, $d \in \mathbf{E}\Diamond(C \sqcap D)^{\mathfrak{I},\varepsilon}$. On the other hand, since $d \notin A^{\mathfrak{I},w}$ for all $w \neq \{\varepsilon, w_i\}$, $d \notin (\mathbf{E}\Box\mathbf{E}\Diamond A)^{\mathfrak{I},\varepsilon}$.

❏

Although the non-convexity of the previous case relies on the non-strict interpretation of the temporal operators, we can also show that the fragment $\mathrm{CTL}_{\mathcal{EL}}^{\mathbf{E}\Diamond^<,\mathbf{E}\Box^<}$ (allowing for the strict version of $\mathbf{E}\Diamond$ and $\mathbf{E}\Box$) is also non-convex. Consider the following setting:

$$\mathcal{T} = \{C \sqsubseteq \mathbf{E}\Diamond^< B,\ B \sqsubseteq D \sqcap \mathbf{E}\Box^< D\}$$
$$D_1 = \mathbf{E}\Diamond^<\mathbf{E}\Diamond^< B,\ D_2 = \mathbf{E}\Box^< D.$$

Lemma 3.19. *$\mathcal{T} \models C \sqsubseteq \bigsqcup D_i$ but $\mathcal{T} \not\models D_i$ for $1 \leq i \leq 2$.*

Proof. For the former, let $\mathfrak{I}$ be a model of $\mathcal{T}$, and $d \in C^{\mathfrak{I},w}$ for some $w \in W$. Hence, $d \in (\mathbf{E}\Diamond^< B)^{\mathfrak{I},w}$, that is, there exists a $j > 0$ such that $d \in B^{\mathfrak{I},\pi[j]}$ for some $\pi \in \mathsf{Paths}(w)$. Then, we can distinguish two cases (1) $j = 1$ or (2) $j > 1$:

– if the first case holds, then $d \in (D \sqcap \mathbf{E}\Box^< D)^{\mathfrak{I},\pi[1]}$. Hence, by semantics, $d \in (\mathbf{E}\Box^< D)^{\mathfrak{I},w}$.

– if the second case holds, then $d \in (\mathbf{E}\Diamond^< B)^{\mathfrak{I},\pi[1]}$. Therefore, by semantics $d \in (\mathbf{E}\Diamond^<\mathbf{E}\Diamond^< B)^{\mathfrak{I},w}$.

From these cases, we can conclude $\mathcal{T} \models C \sqsubseteq \mathbf{E}\Box^< D \sqcup \mathbf{E}\Diamond^<\mathbf{E}\Diamond^< B$.

For the latter, we construct a model $\mathfrak{I} = (\Delta, T, \{\mathcal{I}_w\}_{w\in W})$ of $\mathcal{T}$ with $\Delta = \{d\}$ and T a 1-ary tree with $w_1 = \varepsilon{\cdot}1$ and $w_2 = w_1{\cdot}1$, such that $d \in C^{\mathfrak{I},\varepsilon}$ and $d \notin D_2^{\mathfrak{I},\varepsilon}$ by setting

$$\begin{array}{ll} C^{\mathfrak{I},\varepsilon} := \{d\}; & C^{\mathfrak{I},w} := \emptyset, \text{ for } w \neq \varepsilon; \\ B^{\mathfrak{I},w_2} := \{d\}; & B^{\mathfrak{I},w} := \emptyset; \text{ for } w \neq w_2; \\ D^{\mathfrak{I},\{w_2,w\}} := \{d\}, \text{ for all } w \text{ of the form } w_2{\cdot}u; & D^{\mathfrak{I},\{\varepsilon,w_1\}} = \emptyset. \end{array}$$

Clearly, $\mathfrak{I}$ is a model of $\mathcal{T}$. Now, since $d \in B^{\mathfrak{I},w_2}$, by semantics, $d \in (\mathbf{E}\Diamond^<\mathbf{E}\Diamond^< B)^{\mathfrak{I},\varepsilon}$. Therefore, by semantics, $d \in (\mathbf{E}\Diamond^< B)^{\mathfrak{I},\varepsilon}$. On the other hand since $d \notin D^{\mathfrak{I},w_1}$, $d \notin (\mathbf{E}\Box^< B)^{\mathfrak{I},\varepsilon}$. Following these ideas we can also construct a model $\mathfrak{I}'$ of $\mathcal{T}$ such that $d \in C^{\mathfrak{I},\varepsilon}$ but $d \notin D_1^{\mathfrak{I},\varepsilon}$.

❏

Now, a similar reduction to that of Lemma 3.17 can be done for these two fragments. Moreover, for the remaining fragments of $\mathrm{CTL}_{\mathcal{EL}}$ listed in Theorem 3.15, non-convexity can be shown using the settings shown in Figure 3.2 and then similar reductions to that of Lemma 3.17 can be done.

The logic $\mathrm{CTL}^{\mathbf{A}\bigcirc}_{\mathcal{EL}}$ can be proved to be convex. However, it is nevertheless EXPTIME-hard, which follows from the observation that, after dropping the contructor $\exists r.C$, $\mathrm{CTL}^{\mathbf{A}\bigcirc}_{\mathcal{EL}}$ is a notational variant of the description logic $\mathcal{FL}_0$ which is shown to be EXPTIME-complete in [12, 49].

The complexity of some fragments of $\mathrm{CTL}_{\mathcal{EL}}$ remains open. In particular, we conjecture that the PSPACE lower bound for $\mathrm{CTL}^{\mathbf{E}\Diamond,\mathbf{A}\Box}_{\mathcal{EL}}$ is indeed tight. We also conjecture that, as in the case of $\mathrm{CTL}^{\mathbf{E}\Diamond}_{\mathcal{EL}}$, we can reduce $\mathrm{CTL}^{\mathbf{E}\Box}_{\mathcal{EL}}$ to $\mathcal{EL}^{++}$ by introducing a total role $\mathsf{succ}_\Box$ and then replacing every temporal concept $\mathbf{E}\Box A$ with a fresh concept M_A and adding the CI $M_A \sqsubseteq A \sqcap \exists \mathsf{succ}_\Box.M_A$.

3.5 Reasoning about $\mathrm{CTL}^*_{\mathcal{ALC}}$ and $\mathrm{CTL}_{\mathcal{ALC}}$ Temporal TBoxes

We finalize our study on the computational complexity of branching-time TDLs by reconsidering $\mathrm{CTL}^*_{\mathcal{ALC}}$ and $\mathrm{CTL}_{\mathcal{ALC}}$, studied in Section 3.3, but now allowing temporal operators to be applied not only to concepts but to also to concept inclusions. We present a uniform decision procedure for satisfiability of $\mathrm{CTL}^*_{\mathcal{ALC}}$ and $\mathrm{CTL}_{\mathcal{ALC}}$-temporal TBoxes based on a careful combination of alternating 2-way tree automata and nondeterministic tree automata over infinite trees. We obtain a 2EXPTIME upper bound for $\mathrm{CTL}_{\mathcal{ALC}}$ and a 3EXPTIME upper bound for $\mathrm{CTL}^*_{\mathcal{ALC}}$. Moreover, we show that the presence of temporal TBoxes leads to an increase in the complexity. In particular, we provide a matching 2EXPTIME lower bound for $\mathrm{CTL}_{\mathcal{ALC}}$. We begin by introducing the syntax and semantics of branching-time temporal TBoxes.

3.5.1 Syntax and Semantics

Branching temporal TBoxes are constructed using TBox formulas in which CIs are the *atomic formulas* such that the concepts forming them are temporal concepts. Formally, temporal TBoxes are defined as follows. *$\mathrm{CTL}^*_{\mathcal{ALC}}$-state TBoxes* φ and *$\mathrm{CTL}^*_{\mathcal{ALC}}$-path TBoxes* ψ, ϑ are formed according to the following grammar:

$$\begin{aligned} \varphi &::= C \sqsubseteq D \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathbf{E}\psi \\ \psi, \vartheta &::= \varphi \mid \neg\psi \mid \vartheta \wedge \psi \mid \bigcirc\psi \mid \Box\psi \mid \psi\mathcal{U}\vartheta \end{aligned}$$

where C, D are $\mathrm{CTL}^*_{\mathcal{ALC}}$ concepts. A *temporal $\mathrm{CTL}^*_{\mathcal{ALC}}$-TBox* is a $\mathrm{CTL}^*_{\mathcal{ALC}}$-state TBox; temporal *$\mathrm{CTL}_{\mathcal{ALC}}$-TBoxes* are defined in the expected way.

Definition 3.10. *Let $\mathfrak{I}$ be a temporal interpretation. For a time point w in $\mathfrak{I}$, the* truth relation *$\models$ for temporal $\mathrm{CTL}^*_{\mathcal{ALC}}$-state TBoxes is defined as follows:*

$$
\begin{array}{lll}
\mathfrak{I}, w \models C \sqsubseteq D & \textit{iff} & C^{\mathfrak{I},w} \subseteq D^{\mathfrak{I},w}, \\
\mathfrak{I}, w \models \neg\varphi & \textit{iff} & \mathfrak{I}, w \not\models \varphi, \\
\mathfrak{I}, w \models \varphi_1 \wedge \varphi_2 & \textit{iff} & \mathfrak{I}, w \models \varphi_1 \textit{ and } \mathfrak{I}, w \models \varphi_2, \\
\mathfrak{I}, w \models \mathbf{E}\psi & \textit{iff} & \mathfrak{I}, \pi \models \psi \textit{ for some } \pi \in \mathsf{Paths}(w).
\end{array}
$$

For a path π in $\mathfrak{I}$, the truth relation $\models$ for path TBox formulas is defined as follows:

$$
\begin{array}{lll}
\mathfrak{I}, \pi \models \varphi & \textit{iff} & \mathfrak{I}, \pi[0] \models \varphi, \\
\mathfrak{I}, \pi \models \neg\psi & \textit{iff} & \mathfrak{I}, \pi \not\models \psi, \\
\mathfrak{I}, \pi \models \psi_1 \wedge \psi_2 & \textit{iff} & \mathfrak{I}, \pi \models \psi_1 \textit{ and } \mathfrak{I}, \pi \models \psi_2, \\
\mathfrak{I}, \pi \models \bigcirc\psi & \textit{iff} & \mathfrak{I}, \pi[1..] \models \psi, \\
\mathfrak{I}, \pi \models \Box\psi & \textit{iff} & \forall j \geq 0. \pi[j..] \models \psi, \\
\mathfrak{I}, \pi \models \psi_1 \mathcal{U} \psi_2 & \textit{iff} & \exists j \geq 0. (\mathfrak{I}, \pi[j..] \models \psi_2 \wedge \forall 0 \leq k < j. (\mathfrak{I}, \pi[k..] \models \psi_1)).
\end{array}
$$

We say that a temporal interpretation $\mathfrak{I}$ is a *model* of a temporal CTL$^*_{\mathcal{ALC}}$-TBox φ if $\mathfrak{I}, \varepsilon \models \varphi$. Temporal TBoxes are useful for expressing the *dynamics of policies*. For example, the following temporal CTL$_{\mathcal{ALC}}$-TBox says that, in all possible futures, there will be eventually a policy such that all students who fail a single major exam will immediately and lastingly be exmatriculated.

$$
\mathbf{A}\Diamond\mathbf{A}\Box(\mathsf{Student} \sqcap \exists\mathsf{fails}.\mathsf{MajorExam} \sqsubseteq \mathbf{A}\Box\neg\mathsf{Student})
$$

The central reasoning problem we consider is temporal TBox satisfiability. We say that a temporal CTL$^*_{\mathcal{ALC}}$-TBox is *satisfiable* if it has a model. Note that it is not necessary to consider satisfiability of a concept w.r.t. a temporal TBox, since a concept C is satisfiable w.r.t. a temporal TBox φ iff the temporal TBox $\neg(\top \sqsubseteq \neg C) \wedge \varphi$ is satisfiable.

We present a uniform decision procedure for temporal TBox satisfiability for CTL$^*_{\mathcal{ALC}}$ and CTL$_{\mathcal{ALC}}$ based on a careful amalgamation of alternating 2-way tree automata and nondeterministic tree automata running on quasi-models, that is, infinite trees in which each node is associated with an *abstraction* of an $\mathcal{ALC}$ model. Before proceeding to present our algorithms, we introduce some basic notions of alternating automata.

Alternating automata, introduced by Muller and Schupp [63], generalize standard nondeterministic automata by allowing several successor states along a specific branch of the tree, that is, an alternating automaton might send several copies of itself to a single branch. In this thesis, we focus on *two-way* alternating automata, which extend alternating automata by allowing not only successor states, but also a predecessor state.

For a set X, let $\mathcal{B}^+(X)$ be the set of Boolean formulas built from elements in X using $\wedge$, $\vee$, true and false. Let $Y \subseteq X$, we say that Y *satisfies* a formula $\theta \in \mathcal{B}^+(X)$ if assigning true to the members of Y and assigning false to the members of $X \setminus Y$ makes θ true. For example, the sets $\{q_1, q_2\}$ and $\{q_1, q_4\}$ both satisfy the formula $(q_1 \vee q_5) \wedge (q_2 \vee q_4)$.

Definition 3.11. *Let* $[k] = \{-1, 0, \ldots, k\}$. *An* alternating 2-way tree automaton (2ATA) *on* Σ*-labeled* k*-ary trees is a tuple* $\mathcal{A} = (Q, \Sigma, Q^0, \delta, F)$ *where all components except* δ *are as for NTAs (cf. Definition 3.5). The transition function* δ *of a 2ABTA is a function* $\delta : Q \times \Sigma \to \mathcal{B}^+([k] \times Q)$.

For example, the following transition δ of a 2ATA on a binary tree

$$\delta(q, \sigma) = (1, q_1) \wedge (1, q_2) \vee (2, q_1) \wedge (2, q_2) \wedge (1, q_1)$$

means that the automaton can choose between two possibilities: in the first possibility two copies proceed in direction 1, one in state q_1 and the other in state q_2. In the second possibility two copies proceed in direction 2, one in state q_1 and the other in state q_2, and a third copy proceeds in direction 1 in state q_1. This example also shows that the transition function of a 2ATA allows to send several copies of the automaton to the same direction and it is not forced to send copies to all directions.

Now, consider an NTA $\mathcal{A} = (Q, \Sigma, Q^0, \delta, F)$. First, recall that the transition function δ of an NTA maps each state $q \in Q$ and input symbol $\sigma \in \Sigma$ to a set of k-tuples. Thus we can represent δ by the transition relation of a 2ATA. For example, the transition relation $\delta(q, \sigma) = \{\langle q_2, q_3\rangle, \langle q_1, q_4\rangle\}$ of an NTA on a 2-ary tree can be written as

$$\delta(q, \sigma) = (1, q_2) \wedge (2, q_3) \vee (1, q_1) \wedge (2, q_4).$$

A *run of a 2ATA* $\mathcal{A}$ on a Σ-labeled k-ary tree (T, τ) is a $T{\times}Q$-labeled tree (T_r, r). Intuitively, a node in T_r labeled with (w, q) describes a copy of $\mathcal{A}$ that is at state q and reads the node w from T. Moreover, the labels of a node and its successors must satisfy the transition function. We next introduce the definition of a run. For any $w \in (\mathbb{N}\backslash\{0\})^*$ and $m \in k$, we put $\mathsf{mov}(w, m) = w$ if $m = 0$, $\mathsf{mov}(w, m) = w{\cdot}m$ if $m > 0$, and $\mathsf{mov}(w, m) = u$ if $m = -1$ and $w = uc$ with $c \in \mathbb{N}$. A *run* of $\mathcal{A}$ on τ is a $T{\times}Q$-labeled tree (T_r, r) such that the following hold:

- $r(\varepsilon) = (\varepsilon, q_0)$ for some $q_0 \in Q^0$ and
- for all $x \in T_r, r(x) = (w, q)$ with $\delta(q, \tau(w)) = \theta$, there is a set

  $$\mathcal{S} = \{(m_1, q_1), \ldots, (m_n, q_n)\} \subseteq [k]{\times}Q$$

  such that the following hold:

  1. $\mathcal{S}$ satisfies θ, and
  2. for $1 \leq i \leq n$, we have $x{\cdot}i \in T_r$, $\mathsf{mov}(w, m_i)$ is defined, and $\tau_r(x{\cdot}i) = (\mathsf{mov}(w, m_i), q_i)$.

Note that the automaton cannot go backwards from the root of the input tree since $\mathsf{mov}(w, x{\cdot}i)$ needs to be defined.

A run (T_r, r) is *accepting* if all its paths satisfy the acceptance condition. We consider the so-called *Büchi acceptance condition*: given a run (T_r, r), a path π *satisfies the Büchi acceptance condition* if $\mathsf{inf}(\pi) \cap F \neq \emptyset$, where $\mathsf{inf}(\pi) \subseteq Q$ is such that $q \in \mathsf{inf}(\pi)$ if and only if there are infinitely many $w \in \pi$ for which $r(w) = \mathbb{N}^*{\times}\{q\}$. Analogously to NBTAs, we use 2ABTA to denote 2ATAs using this acceptance condition.

Theorem 3.20 ([78]). *The nonemptiness problem for 2ABTAs is* ExpTime*-complete.*

Notably, the possibility of using conjunctions in 2ATAs allows for a straightforward construction for the closure under intersection [63], that is, take the disjoint union of two given 2ATAs and then combine them by taking the conjunction of the initial states.

In order to define 2ATAs more compactly later on, we add a third component, serving as a *root flag*, to the transition function: the transition function δ of a 2ATA is a function

$$\delta : Q\times\Sigma\times\{\mathsf{t},\mathsf{f}\} \to \mathcal{B}^+([k]\times Q).$$

Then, the definition of a run is extended accordingly, by putting for $w \in T$, $\mathsf{root}(w) = \mathsf{t}$ if $w = \varepsilon$ and $\mathsf{root}(w) = \mathsf{f}$ otherwise. Note that this unorthodox assumption does not cause any problems since for a 2ATA $\mathcal{A} = (Q, \Sigma, Q^0, \delta, F)$ we can construct a 2ATA $\mathcal{A}'$ with alphabet $\Sigma\times\{\mathsf{t},\mathsf{f}\}$ that behaves like $\mathcal{A}$ and then define a 2ATA $\mathcal{A}'$ which accepts $\Sigma\times\{\mathsf{t},\mathsf{f}\}$-labeled trees with the root node labeled with alphabet letters of the form (σ, t), and the other nodes with alphabet letters of the form (σ, f). Finally, we can simply intersect them.

3.5.2 An Algorithm for Temporal TBox Satisfiability for $\mathrm{CTL}^*_{\mathcal{ALC}}$ and $\mathrm{CTL}_{\mathcal{ALC}}$

Now, we have the required ingredients to present our algorithms for temporal TBox satisfiability for $\mathrm{CTL}^*_{\mathcal{ALC}}$ and $\mathrm{CTL}_{\mathcal{ALC}}$.

Let φ be a temporal TBox formulated in $\mathrm{CTL}^*_{\mathcal{ALC}}$ or $\mathrm{CTL}_{\mathcal{ALC}}$ whose satisfiability is to be decided. We use $\mathsf{cl}(\varphi)$ to denote the set of state concepts that occur in φ, closed under subconcepts and single negation. A *concept type for* φ is a set $t \subseteq \mathsf{cl}(\varphi)$ and $\mathsf{tp}(\varphi)$ denotes the set of all concept types for $\mathcal{T}$. We use $\mathsf{sub}(\varphi)$ to denote the set of all state subformulas of φ. A *formula type for* φ is a subset of $\mathsf{sub}(\varphi)$.

Definition 3.12. *A* quasi-world for φ *is a pair* (S_1, S_2) *with* $S_1 \subseteq \mathsf{tp}(\varphi)$ *a set of concept types and* $S_2 \subseteq \mathsf{sub}(\varphi)$ *a formula type for* φ *such that*

1. *if* $t \in S_1$ *and* $\exists r.C \in t$*, then there is a* $t' \in S_1$ *with* $\{C\} \cup \{\neg D \mid \neg\exists r.D \in t\} \subseteq t'$*;*
2. *for all* $C \sqsubseteq D \in \mathsf{sub}(\varphi)$*, we have* $C \sqsubseteq D \in S_2$ *iff, for all* $t \in S_1$*,* $C \in t$ *implies* $D \in t$*.*

Let $\mathsf{qw}(\varphi)$ denote the set of all quasi-worlds for φ. Furthermore, a *quasi-model* $\mathfrak{M}$ *of* φ is a $\mathsf{qw}(\varphi)$-labeled tree, of any outdegree.

For $t \in \mathsf{tp}(\varphi)$, $\bar{t}$ is the result of replacing every $C \in t\backslash\mathsf{N_C}$ with a fresh concept name X_C, and cn_X denotes the set of all resulting concept names, including those in $\mathcal{T}$. For $C \in \mathsf{cl}(\mathcal{T})$, $\overline{C}$ denotes the result of replacing in C every subconcept $\exists r.D$ with $X_{\exists r.D}$, and $\sqcap$ with $\wedge$. For every $\psi \in \mathsf{sub}(\varphi)$, $\overline{\psi}$ denotes the result of replacing every subformula $C \sqsubseteq D$ of ψ with a fresh concept $Y_{C\sqsubseteq D}$ (which plays the role of a propositional letter for CTL / CTL*) and cn_Y is the set of all concept names thus introduced. For $S \subseteq \mathsf{sub}(\varphi)$, we set $\overline{S} = \{\overline{\psi} \mid \psi \in S\}$. For $\mathfrak{M}$ a quasi-model, we use $\mathfrak{M}_2$ to denote the 2^{cn_Y}-labeled tree obtained by associating each node $w \in \mathfrak{M}$ with the label $\overline{S_2(w)}$.

Definition 3.13. *A quasi-model $\mathfrak{M} = (T, \tau)$ of φ is* proper *if the following conditions are satisfied:*

1. $\mathfrak{M}_2, \varepsilon \models \overline{\varphi}$;
2. *for all $w \in T$ with $\tau(w) = (S_1, S_2)$ and all $s \in S_1$, there is a 2^{cn_X}-labeled tree (T, τ') such that*
   a) $\tau'(w) = \overline{s}$;
   b) *for all $w' \in T$ with $\tau(w') = (S_1', S_2')$, there is an $s' \in S_1'$ such that $\tau'(w') = \overline{s'}$;*
   c) *ε satisfies $\mathbf{A}\Box \bigwedge_{X_C \in \mathsf{cn}_X} (X_C \leftrightarrow \overline{C})$.*

Intuitively, Condition 1 ensures that $\mathfrak{M}$ satisfies the temporal TBox φ and Condition 2 guarantees that, for each required domain element, we can consistently select a type from the quasi-world at each node of $\mathfrak{M}$. The following result shows that to decide satisfiability of φ, it suffices to check the existence of a proper quasi-model for φ.

Lemma 3.21. *φ is satisfiable iff there is a proper quasi-model for φ.*

Proof. "$\Rightarrow$:" Let $\mathfrak{I} = (\Delta, T, \{\mathcal{I}_w\}_{w \in W})$ be a temporal model of φ. We define a $\mathsf{qw}(\varphi)$-labeled tree structure $\mathfrak{M} = (T, \tau)$ such that for all $w \in T$, $\tau(w)$ is defined as follows:

$$\begin{aligned} S_2(w) &= \{\Psi \in \mathsf{sub}(\varphi) \mid \mathfrak{I}, w \models \Psi\}; \\ \pi(d, w) &= \{C \in \mathsf{cl}(\varphi) \mid d \in C^{\mathfrak{I},w}\}; \\ S_1(w) &= \{\pi(d, w) \mid d \in \Delta\}. \end{aligned}$$

We argue that $\mathfrak{M}$ is proper, that is, $\mathfrak{M}$ satisfies condition (1) and (2) of Definition 3.13.

– For condition (1), we obtain the 2^{cn_Y}-labeled tree $\mathfrak{M}_2$ by associating each $w \in \mathfrak{M}$ with the label $\overline{S_2(w)}$. Clearly, by definition of $S_2(w)$, $\mathfrak{M}_2, \varepsilon \models \overline{\varphi}$.

– For condition (2), for all $w \in T$ with $\tau(w) = (S_1, S_2)$ and all $\pi(d, w) \in S_1$ there is a 2^{cn_X}-labeled tree (T, τ') satisfying 2(a)-(c), where τ_d' is defined as follows. For all $w' \in T$, set

$$\tau_d'(w') = \overline{\pi(d, w').}$$

  Observe that indeed conditions (a)-(c) are fulfilled:
  – Condition (a) follows by definition of τ_d', that is, $\tau'(w) = \overline{\pi(d, w)}$.
  – Condition (b) follows by definition of $\mathfrak{M}$ and τ_d'.
  – Condition (c): note that, by definition of $\tau_d'(w)$, $C \in \pi(d, w)$ iff $X_C \in \tau_d'(w)$ for all $w \in W$. Moreover, by the following claim, $C \in \pi(d, w)$ implies $(T, \tau_d'), w \models \overline{C}$. Therefore, condition 2(c) holds.

Claim. For each $C \in \mathsf{cl}(\mathcal{T})$, $w \in W$ and $d \in \Delta$, we have that

$$C \in \pi(d,w) \text{ implies } (T,\tau'_d), w \models \overline{C},$$

for every $\pi \in \mathsf{Paths}(w)$ and path concept $\mathcal{C}$

$$d \in \mathcal{C}^{\mathfrak{I},\pi'} \text{ implies } (T,\tau'_d), \pi' \models \overline{\mathcal{C}}.$$

Proof of the claim. The proof is by a simultaneous induction on the structure of C and $\mathcal{C}$. For concept names it follows trivially.

- $C = \neg C$. Since $(\neg C) \in \pi(d,w)$, then $d \in (\neg C)^{\mathfrak{I},w}$, that is, $d \notin C^{\mathfrak{I},w}$. By I.H., $(T,\tau'_d), w \not\models \overline{C}$. Hence, $(T,\tau'_d), w \models \neg\overline{C}$.
- $C = C \sqcap D$. Since $(C \sqcap D) \in \pi(d,w)$, then $d \in (C \sqcap D)^{\mathfrak{I},w}$, that is, $d \in C^{\mathfrak{I},w}$ and $d \in D^{\mathfrak{I},w}$. By I.H., $(T,\tau'_d), w \models \overline{C}$ and $(T,\tau'_d), w \models \overline{D}$. Therefore, $(T,\tau'_d), w \models \overline{C} \wedge \overline{D}$.
- $C = \mathbf{E}\mathcal{C}$. Since $\mathbf{E}C \in \pi(d,w)$, then $d \in (\mathbf{E}\mathcal{C})^{\mathfrak{I},w}$, that is, there is a $\pi' \in \mathsf{Paths}(w)$ such that $d \in \mathcal{C}^{\mathfrak{I},\pi'}$. By the second point of the claim, $(T,\tau'_d), \pi \models \overline{\mathcal{C}}$. Therefore, $(T,\tau'_d), w \models \overline{\mathbf{E}\mathcal{C}}$.

This finishes the proof of the first point of the claim.

We proceed to prove the second point of the claim.

- $\mathcal{C} = D$ with D a state concept. We have that $d \in D^{\mathfrak{I},\pi[0]}$ then, by definition of $\mathfrak{M}$, $D \in \pi(d,\pi[0])$. Note that $\pi[0] = w$, then by the first point of the claim, $(T,\tau'_d), w \models \overline{D}$.
- $\mathcal{C} = \neg\mathcal{D}$. We have that $d \in \neg\mathcal{D}^{\mathfrak{I},\pi'}$, that is, $d \notin \mathcal{D}^{\mathfrak{I},\pi'}$. By I.H. $(T,\tau'_d), \pi' \not\models \overline{\mathcal{D}}$. Therefore, $(T,\tau'_d), \pi' \models \neg\overline{\mathcal{D}}$.
- $\mathcal{C} = \mathcal{C}_1 \sqcap \mathcal{C}_2$, similar to the analogous case for state concepts.
- $\mathcal{C} = \bigcirc\mathcal{D}$. We have that $d \in (\bigcirc\mathcal{D})^{\mathfrak{I},\pi'}$, that is, $d \in \mathcal{D}^{\mathfrak{I},\pi'[1]}$. Now, by I.H., $(T,\tau'_d), \pi'[1] \models \overline{\mathcal{D}}$. Therefore, by semantics, $(T,\tau'_d), \pi' \models \bigcirc\overline{\mathcal{D}}$.
- $\mathcal{C} = \Box\mathcal{D}$. We have that $d \in (\Box\mathcal{D})^{\mathfrak{I},\pi'}$, that is, for all $j \geq 0$, $d \in \mathcal{D}^{\mathfrak{I},\pi[j..]}$. Now, by I.H., $(T,\tau'_d), \pi'[j..] \models \overline{\mathcal{D}}$. Therefore, by semantics, $(T,\tau'_d), \pi' \models \Box\overline{\mathcal{D}}$.
- $\mathcal{C} = \mathcal{C}_1\mathcal{U}\mathcal{C}_2$. We have that $\exists j \geq 0.(d \in \mathcal{C}_2^{\mathfrak{I},\pi'[j..]} \wedge \forall 0 \leq k < j. d \in \mathcal{C}_1^{\mathfrak{I},\pi'[k..]})$. Now, by I.H., $(T,\tau'_d), \pi'[j..] \models \overline{\mathcal{C}_2} \wedge \forall 0 \leq k < j.((T,\tau'_d), \pi'[k..] \models \overline{\mathcal{C}_1})$. Therefore, $(T,\tau'_d), \pi' \models \overline{(\mathcal{C}_1\mathcal{U}\mathcal{C}_2)}$.

  This finishes the proof of the second point of the claim.

This finishes the proof of the claim.

We can conclude then that $\mathfrak{M}$ is indeed a proper-quasimodel of φ.

"$\Leftarrow$:" Let $\mathfrak{M} = (T, \tau)$ be a proper-quasimodel of φ. We next define a model $\mathfrak{I}$ of φ. First note that according to Condition 2 of Definition 3.13, for all $w \in T$ with $\tau(w) = (S_1, S_2)$ and all $s \in S_1$ there is a 2^{cn_X}-labeled tree $(T, \tau_{w,s})$ satisfying 2(a)-(c). We define the temporal interpretation $\mathfrak{I} = (\Delta, T, \{\mathcal{I}_w\}_{w \in W})$ with $\Delta = \{(w, s) \mid s \in S_1(w)\}$ given by:

$$\begin{aligned}
A^{\mathfrak{I},w} &= \{(v, s) \in \Delta \mid A \in \tau_{v,s}(w)\}; \\
r^{\mathfrak{I},w} &= \{((v, s), (v', s')) \mid X_{\exists r.C} \in \tau_{v,s}(w) \text{ implies} \\
&\qquad \{X_C\} \cup \{X_{\neg E} \mid X_{\neg\exists r.E} \in \tau_{v,s}(w)\} \subseteq \tau_{v',s'}(w)\}.
\end{aligned}$$

Claim. For all $C \in \mathsf{cl}(\mathcal{T})$, $w \in W$ and $(v, s) \in \Delta$, we have

$$(v, s) \in C^{\mathfrak{I},w} \text{ iff } X_C \in \tau_{v,s}(w),$$

for every $\pi \in \mathsf{Paths}(w)$, and path concept $\mathcal{C}$

$$(v, s) \in \mathcal{C}^{\mathfrak{I},\pi} \text{ iff } (T, \tau_{v,s}), \pi \models \overline{\mathcal{C}}.$$

Proof of the claim. The proof is by a simultaneous induction on the structure of C and $\mathcal{C}$. The induction start, where C is a concept name is immediate by the definition of $\mathfrak{I}$. For the induction step we distinguish the following cases.

- $C = \neg D$ "if:" $(v, s) \in \neg D^{\mathfrak{I},w}$, that is, $(v, s) \notin D^{\mathfrak{I},w}$. Now, by I.H., $X_D \notin \tau_{v,s}(w)$. Furthermore, by condition 2(c) of Definition 3.13, $(T, \tau_{v,s}), w \models \neg\overline{D}$. Finally, again by condition 2(c), $X_{\neg D} \in \tau_{v,s}(w)$.

  "only if:" $X_{\neg D} \in \tau_{v,s}(w)$, by condition 2(c) of Definition 3.13, implies that $(T, \tau_{v,s}), w \not\models \overline{D}$. Now, by condition 2(c), $X_D \notin \tau_{v,s}(w)$. By, I.H., $(v, s) \notin D^{\mathfrak{I},w}$. Therefore, $(v, s) \in (\neg D)^{\mathfrak{I},w}$.

- $C = D \sqcap E$ "if:" $(v, s) \in (D \sqcap E)^{\mathfrak{I},w}$, that is, $(v, s) \in D^{\mathfrak{I},w}$ and $(v, s) \in E^{\mathfrak{I},w}$. By I.H., $X_D \in \tau_{v,s}(w)$ and $X_E \in \tau_{v,s}(w)$. Now, condition 2(c) of Definition 3.13, $(T, \tau_{v,s}), w \models \overline{D}$ and $(T, \tau_{v,s}), w \models \overline{E}$. So, $(T, \tau_{v,s}), w \models \overline{D} \wedge \overline{E}$. Once again, by condition 2(c), $(T, \tau_{v,s}), w \models X_{D \sqcap E}$. Therefore, $X_{D \sqcap E} \in \tau_{v,s}(w)$.

  "only if:" $X_{D \sqcap E} \in \tau_{v,s}(w)$, by condition 2(c) of Definition 3.13, we have that $(T, \tau_{v,s}), w \models \overline{D} \wedge \overline{E}$., that is, $(T, \tau_{v,s}), w \models \overline{D}$ and $(T, \tau_{v,s}), w \models \overline{E}$. Once again, by condition 2(c), we have that $X_D, X_E \in \tau_{v,s}(w)$. Now, by I.H., $(v, s) \in D^{\mathfrak{I},w}$ and $(v, s) \in E^{\mathfrak{I},w}$. Therefore, $(v, s) \in (D \sqcap E)^{\mathfrak{I},w}$.

- $C = \exists r.C$. "if" Follows from condition from the fact that $\tau_{v,s}(w) = \overline{s'}$ for some $s' \in S_1(w)$, and s' satisfies condition 1 of the definition of a quasi-world.

  "only if:" $X_{\exists r.C} \in \tau_{v,s}(w) = \overline{s'}$ then $\exists r.C \in s'$. Now, since $s' \in S_1(w)$, then it satisfies condition 1 of the definition of quasi-world, that is, there is a $s'' \in S_1(w)$ such that $\{C\} \cup \{\neg E \mid \neg\exists r.E \in s'\} \subseteq s''$. Note that, by definition of $\tau_{w,s''}$, $\{X_C\} \cup \{X_{\neg E} \mid X_{\neg\exists r.E} \in \overline{s'}\} \subseteq \tau_{w,s''}(w)$. Then, by definition of $\mathfrak{I}$, $((v, s)(w, s'')) \in r^{\mathfrak{I},w}$. Furthermore, by I.H., $(w, t'') \in C^{\mathfrak{I},w}$. Therefore, $(v, s) \in (\exists r.C)^{\mathfrak{I},w}$.

– $C = \mathbf{E}\mathcal{C}$ "if:" $(v,s) \in (\mathbf{E}\mathcal{C})^{\mathfrak{I},w}$. This implies that, by semantics, $(v,s) \in \mathcal{C}^{\mathfrak{I},\pi}$ for some $\pi \in \mathsf{Paths}(w)$. Now, by the second point of the claim, $(T,\tau_{v,s}),\pi \models \overline{\mathcal{C}}$. Therefore, by semantics, $(T,\tau_{v,s}),w \models \overline{\mathbf{E}\mathcal{C}}$. Since, $(T,\tau_{v,s}) \models \varphi$ from condition 2(c), then $X_{\mathbf{E}\mathcal{C}} \in \tau_{v,s}(w)$.

  "only if:" $X_{\mathbf{E}\mathcal{C}} \in \tau_{v,s}(w)$. By condition 2(c) of the Definition 3.13, we have that $(T,\tau_{v,s}),w \models \overline{\mathbf{E}\mathcal{C}}$, that is, $(T,\tau_{v,s}),\pi \models \overline{\mathcal{C}}$ for some $\pi \in \mathsf{Paths}(w)$. Now, by the second point of the claim, $(v,s) \in \mathcal{C}^{\mathfrak{I},\pi}$. Therefore, $(v,s) \in (\mathbf{E}\mathcal{C})^{\mathfrak{I},w}$.

This finishes the proof of the first point of the claim.

We proceed to show the second point of the claim.

– $\mathcal{C} = D$ with D a state concept. "if:" $(v,s) \in D^{\mathfrak{I},\pi[0]}$. Note that $\pi[0] = w$, then by the first point of the claim, $(T,\tau_{v,s}),w \models \overline{D}$.

  "only if:" $(T,\tau_{v,s}),\pi[0] \models \overline{D}$. By condition 2(c) of Definition 3.13, $X_D \in \tau_{t,i}(\pi[0])$. Note that $\pi[0] = w$, then by the outer induction $(v,s) \in D^{\mathfrak{I},w}$.

– $\mathcal{C} = \neg\mathcal{D}$. "if:" $(v,s) \in \neg\mathcal{D}^{\mathfrak{I},\pi}$, that is, $(v,s) \notin \mathcal{D}^{\mathfrak{I},\pi}$. By I.H. $(T,\tau_{(v,s)}),\pi \not\models \overline{\mathcal{D}}$. Therefore, $(T,\tau_{v,s}),\pi \models \neg\overline{\mathcal{D}}$.

  "only if:" $(T,\tau_{v,s}),\pi \not\models \mathcal{D}$. By I.H., we have that $(v,s) \notin (\mathcal{D})^{\mathfrak{I},\pi}$. Therefore, $(v,s) \in (\neg\mathcal{D})^{\mathfrak{I},\pi}$.

– $\mathcal{C} = \mathcal{C}_1 \sqcap \mathcal{C}_2$, similar to the analogous case for state concepts.

– $\mathcal{C} = \bigcirc\mathcal{D}$. "if:" $(v,s) \in (\bigcirc\mathcal{D})^{\mathfrak{I},\pi}$, that is, $(v,s) \in \mathcal{D}^{\mathfrak{I},\pi[1]}$. Now, by I.H., $(T,\tau_{v,s}),\pi[1] \models \overline{\mathcal{D}}$. Therefore, by semantics, $(T,\tau_{v,s}),\pi \models \bigcirc\overline{\mathcal{D}}$.

  "only if:" $(T,\tau_{v,s}),\pi \models \bigcirc\overline{\mathcal{D}}$. Hence, $(T,\tau_{v,s}),\pi[1] \models \overline{\mathcal{D}}$. Now, by I.H., $(v,s) \in \mathcal{D}^{\mathfrak{I},\pi[1]}$. Therefore, by semantics, $(v,s) \in (\bigcirc\mathcal{D})^{\mathcal{I},\pi}$.

– $\mathcal{C} = \Box\mathcal{D}$. "if:" $(v,s) \in (\Box\mathcal{D})^{\mathfrak{I},\pi}$, that is, for all $j \geq 0$, $(v,s) \in \mathcal{D}^{\mathfrak{I},\pi[j..]}$. Now, by I.H., $(T,\tau_{v,s}),\pi[j..] \models \overline{\mathcal{D}}$. Therefore, by semantics, $(T,\tau_{v,s}),\pi \models \Box\overline{\mathcal{D}}$.

  "only if:" $(T,\tau),\pi \models \Box\overline{\mathcal{D}}$. This means that for all $j \geq 0$, $(T,\tau_{v,s}),\pi[j..] \models \overline{\mathcal{D}}$. Now, by I.H., for all $j \geq 0$, $(v,s) \in \mathcal{D}^{\mathfrak{I},\pi[j..]}$. Therefore, by semantics, $(v,s) \in \Box\mathcal{D}^{\mathfrak{I},\pi}$.

– $\mathcal{C} = \mathcal{C}_1\mathcal{U}\mathcal{C}_2$. "if:" $\exists j \geq 0.((v,s) \in \mathcal{C}_2^{\mathfrak{I},\pi[j..]} \wedge \forall 0 \leq k < j.(v,s) \in \mathcal{C}_1^{\mathfrak{I},\pi[k..]})$. Now, by I.H., $(T,\tau_{v,s}),\pi[j..] \models \overline{\mathcal{C}}_2 \wedge \forall 0 \leq k < j.((T,\tau_{v,s}),\pi[k..] \models \overline{\mathcal{C}}_1)$. Therefore, $(T,\tau_{v,s}),\pi \models \overline{(\mathcal{C}_1\mathcal{U}\mathcal{C}_2)}$.

  "only if:" $\exists j \geq 0.((T,\tau_{v,s}),\pi[j..] \models \overline{\mathcal{C}}_2 \wedge \forall 0 \leq k < j.((T,\tau_{v,s}),\pi[k..] \models \overline{\mathcal{C}}_1)$. Now, by I.H., $(v,s) \in \mathcal{C}_2^{\mathfrak{I},\pi[j..]} \wedge \forall 0 \leq k < j.((t,i) \in \mathcal{C}_1^{\mathfrak{I},\pi[k..])}$. Therefore, $(v,s) \in (\mathcal{C}_1\mathcal{U}\mathcal{C}_2)^{\mathfrak{I},\pi}$.

This finishes the proof of second point of the claim.

This finishes the proof of the claim.

Now, recall that according to Definition 3.13 $\mathfrak{M}_2 \models \overline{\varphi}$. We show the following claim.

Claim. For all $\varphi' \in \mathsf{sub}(\varphi)$ and $w \in W$, we have

$$\mathfrak{M}_2, w \models \overline{\varphi'} \text{ implies } \mathfrak{I}, w \models \varphi',$$

for every $\pi \in \mathsf{Paths}(w)$ and path TBox ψ, we have

$$\mathfrak{M}_2, \pi \models \overline{\psi} \text{ implies } \mathfrak{I}, \pi \models \psi.$$

Proof of the claim. The proof is by simultaneous induction on the structure of φ' and ψ.

- $\varphi' = C \sqsubseteq D$. $\mathfrak{M}_2, w \models X_{C \sqsubseteq D}$. We have that $X_{C \sqsubseteq D} \in \tau_2(w) = \overline{S_2(w)}$. Now, by construction, $C \sqsubseteq D \in S_2(w)$. Recall that, by definition of a quasi-world, $C \sqsubseteq D \in S_2(w)$ iff for all $t \in S_1(w)$, $C \in t$ implies $D \in t$. Fix an arbitrary t. Now, since $\mathfrak{M}$ is proper, satisfies condition 2(a)-(c) for some $w' \in W$ and $t' \in S_1(w)$ such that $\tau_{w',t'}(w) = \bar{t}$. By construction, since $C \in t$ implies $D \in t$, $X_C, X_D \in \tau_{w',t'}(w)$. Now, by the previous claim $(w', t') \in C^{\mathfrak{I},w}$ and $(w', t') \in D^{\mathfrak{I},w}$. Thus, $C^{\mathfrak{I},w} \subseteq D^{\mathfrak{I},w}$. Therefore, $\mathfrak{I}, w \models C \sqsubseteq D$.
- $\varphi' = \neg\varphi'$. $\mathfrak{M}_2, w \not\models \overline{\varphi'}$. Then, by I.H., $\mathfrak{I}, w \not\models \varphi'$. Therefore, $\mathfrak{I}, w \models \neg\varphi'$.
- $\varphi' = \varphi'_1 \wedge \varphi'_2$. $\mathfrak{M}_2, w \models \overline{\varphi'_1 \wedge \varphi'_2}$, that is, $\mathfrak{M}_2, w \models \overline{\varphi'_1}$ and $\mathfrak{M}_2, w \models \overline{\varphi'_2}$. Now, by I.H., $\mathfrak{I}, w \models \varphi'_1$ and $\mathfrak{I}, w \models \varphi'_2$. Therefore, $\mathfrak{I}, w \models \varphi'_1 \wedge \varphi'_2$.
- $\varphi' = \mathbf{E}\psi$. $\mathfrak{M}_2, w \models \overline{\mathbf{E}\psi}$, that is, there exists a path $\pi \in \mathsf{Paths}(w)$ such that $\mathfrak{M}_2, \pi \models \overline{\psi}$. Now, by second point of the claim, $\mathfrak{I}, \pi \models \psi$. Therefore, $\mathfrak{I}, \pi \models \mathbf{E}\psi$.

This finishes the proof of the first point of the claim

We proceed to show the second point of the claim

- $\psi = \varphi_1$ with φ_1 a state concept. $\mathfrak{M}_2, \pi \models \overline{\varphi_1}$, that is, $\mathfrak{M}_2, \pi[0] \models \overline{\varphi_1}$. Note that $\pi[0] = w$, then by the first point of the claim $\mathfrak{I}, w \models \varphi_1$.
- $\overline{\neg\psi}$ and $\overline{\psi_1 \wedge \psi_2}$ similar to the analogous case for state concepts.
- $\psi = \bigcirc\psi$. We have that $\mathfrak{M}_2, \pi \models \bigcirc\overline{\psi}$. Hence, $\mathfrak{M}_2, \pi[1] \models \overline{\psi}$. Now, by I.H., $\mathfrak{I}, \pi[1] \models \psi$. Therefore, by semantics, $\mathfrak{I}, \pi \models \bigcirc\psi$.
- $\psi = \Box\psi$. We have that $\mathfrak{M}_2, \pi \models \Box\overline{\psi}$. This means that for all $j \geq 0$, $\mathfrak{M}_2, \pi[j..] \models \overline{\psi}$. Now, by I.H., for all $j \geq 0$, $\mathfrak{I}, \pi[j..] \models \psi$. Therefore, by semantics, $\mathfrak{I}, \pi \models \Box\psi$.
- $\psi = \psi_1 \mathcal{U} \psi_2$. We have that $\exists j \geq 0.(\mathfrak{M}_2, \pi[j..] \models \overline{\psi_2} \wedge \forall 0 \leq k < j.(\mathfrak{M}_2, \pi[k..] \models \overline{\psi_1})$. Now, by I.H., $\mathfrak{I}, \pi[j..] \models \psi_2 \wedge \forall 0 \leq k < j.(\mathfrak{I}, \pi[k..] \models \psi_1)$. Therefore, $\mathfrak{I}, \pi \models \psi_1 \mathcal{U} \psi_2$.

This finishes the claim

This finishes the proof of the claim.

From the previous claim we conclude therefore that $\mathfrak{I}, \varepsilon \models \varphi$.

❏

The following NBTAs will be used in our decision procedure. Let ϑ be the formula in Condition 2(c). By Theorem 3.3 (*cf.* Section 3.3), we find an NBTA $\mathcal{A}_{\overline{\varphi}} = (Q_1, \Sigma_1, \delta_1, Q_1^0, F_1)$ that accepts exactly the 2^{cn_Y}-labeled $\#^f_{\mathbf{E}}(\varphi)$-ary trees which satisfy $\overline{\varphi}$, where $\#^f_{\mathbf{E}}(\varphi)$ denotes the set of state formulas of the form $\mathbf{E}\psi$ in $\mathsf{sub}(\varphi)$; we also find an NBTA $\mathcal{A}_\vartheta = (Q_2, \Sigma_2, \delta_2, Q_2^0, F_2)$ that accepts exactly the 2^{cn_X}-labeled $\#^c_{\mathbf{E}}(\varphi)$-ary trees which satisfy ϑ, where $\#^c_{\mathbf{E}}(\varphi)$ denotes the set of state concepts of the form $\mathbf{E}\mathcal{C}$ in $\mathsf{sub}(\varphi)$.

We aim at constructing a 2ABTA $\mathcal{A}$ on $\mathsf{qw}(\varphi)$-labeled trees that accepts precisely the proper quasi-models for φ. For doing this, we have to restrict the outdegree of quasi-models in an appropriate way. Set $k := |\mathsf{qw}(\varphi)| \cdot |\mathsf{tp}(\varphi)| \cdot |Q_2|$. The following is proved by replacing Condition 2(c) with a version based on the NBTA $\mathcal{A}_\vartheta$ and carefully analyzing its runs.

Lemma 3.22. *There is a proper quasi-model for φ iff there is a proper quasi-model for φ that is a k-ary tree.*

Proof. The "if"-direction is trivial. For the other direction let $\mathfrak{M} = (T, \tau)$ be an arbitrary proper quasi-model. We can assume that every $w \in T$ has outdegree at least k, otherwise we can just duplicate some successors of w. For this proof we replace condition 2(c) from Definition 3.13 with a version based on the NBTA $\mathcal{A}_\vartheta$ and then we analyze its runs.

We next introduce a modified Condition 2 by restating 2(c) in terms of the automaton $\mathcal{A}_\vartheta$. Moreover, we consider only $\sharp^c_E$-ary trees (T', τ') such that $T' \subseteq T$ and for all $w \in T'$ $\tau'(w) = \tau(w)$. This is enough by the sufficient degree property (*cf.* Proposition 3.2) stating that if there is a model of ϑ then there is one with branching degree $\sharp^c_E$. In particular, $L(\mathcal{A}_\vartheta) = \mathsf{Mod}_{\sharp^c_E}(\vartheta)$. We define condition 2' as follows:

2'. for all $w \in T$ with $\tau(w) = (S_1, S_2)$ and all $s \in S_1$, there is a 2^{cn_X}-labeled $\sharp^c_E$-ary tree (T', τ') as described above, such that
   - (a) $\tau'(w) = \overline{s}$;
   - (b) for all $w' \in T$ with $\tau(w') = (S'_1, S'_2)$, there is an $s' \in S'_1$ such that $\tau'(w') = \overline{s'}$;
   - (c) there is an accepting run (T', r) of $\mathcal{A}_\vartheta$ on (T', τ').

We next fix all trees from satisfying condition 2' together with a selection of types from a quasi-world and states from the accepting runs of $\mathcal{A}_\vartheta$. In the following we denote with "$s \in \tau(w)$" the fact that $s \in S_1$ when $\tau(w) = (S_1, S_2)$.

For every $w \in T$, $s \in \tau(w)$, we fix

- the tree $(T_{w,s}, \tau_{w,s})$ witnessing condition 2',
- the corresponding accepting run $r_{w,s}$ from 2'(c), and
- the set of states $Q_{w,s}$ that $\mathcal{A}_\vartheta$ assigns to s in the accepting runs of $\mathcal{A}_\vartheta$ on all $(T_{w',s'}, \tau_{w',s'})$ where s is used to witness condition 2'(b), that is, $\tau_{w',s'}(w) = \overline{s}$. Formally, $Q_{w,s}$ is defined as follows.

$$Q_{w,s} = \{r_{w',s'}(w) \mid w \in T_{w',s'}, \tau_{w',s'}(w) = \overline{s}\}.$$

Now, we duplicate quasi-worlds of $\mathfrak{M}$ and modify the trees $(T_{w,s}, \tau_{w,s})$ and the corresponding runs $r_{w,s}$ such that in every quasi-world precisely one type is chosen by an accepting run of A_ϑ. After the modification we will have that

$$\sum_{s \in \tau(w)} |Q_{w,s}| = 1$$

for every $w \in T$. More formally, we define the quasi-model $\widehat{\mathfrak{M}} = (\widehat{T}, \widehat{\tau})$ with $\widehat{T} = (\widehat{W}, \widehat{E})$ by taking

$$\begin{aligned}
\widehat{W} &= \{(w,s,q) \mid w \neq \varepsilon \in T, s \in \tau(w), q \in Q_{w,s}\} \cup \{\varepsilon\};\\
\widehat{E} &= \{((w,s,q),(w',s',q')) \mid (w,w') \in E\} \cup \{(\varepsilon,(w,s,q)) \mid (\varepsilon,w) \in E\};\\
\widehat{\tau}(w,s,q) &= \tau(w);\\
\widehat{\tau}(\varepsilon) &= \tau(\varepsilon).
\end{aligned}$$

Note that, by definition of $\widehat{\tau}$, Condition 1 is still satisfied for $\widehat{\mathfrak{M}}$. We make use of $(T_{w,s}, \tau_{w,s})$, and $r_{w,s}$ to show that $\widehat{\mathfrak{M}}$ satisfies also Condition 2'. For all $(w,s,q) \in \widehat{T}$ and $t \in \widehat{\tau}(w,s,q)$ define

$$\begin{aligned}
T_{(w,s,q),t} &= \{(w', \tau_{w,t}(w'), r_{w,t}(w')) \mid w' \in T_{w,t} \setminus \{w\}\} \cup \{(w,s,q)\};\\
\tau_{(w,s,q),t}((w',s',q')) &= \tau_{w,t}(w');\\
r_{(w,s,q),t}((w',s',q')) &= r_{w,t}(w').
\end{aligned}$$

Since $T_{(w,s,q),t}$ is properly defined in terms of $(T_{w,t}, \tau_{w,t})$ and $r_{w,t}$, one can readily see that indeed $\widehat{M}$ satisfies condition 2'. Take an arbitrary $(w,s,q) \in \widehat{W}$ and $t \in \widehat{\tau}(w,s,q)$, the 2^{cn_X}-labeled $\sharp^c_E$-ary tree $(T_{(w,s,q),t}, \tau_{(w,s,q),t})$ witnesses condition 2'.

- For condition (a), we have that $\tau_{(w,s,q),t}((w,s,q)) = \tau_{w,t}(w) = \bar{t}$.
- For condition (b), for all $(w,s',q') \in \widehat{T}$, $\tau_{(w,s,q),t}(w',s',q') = \tau_{w,t}(w') = \bar{s} \in \tau(w') = \widehat{\tau}(w',s',q)$.
- For condition (c), we have that for all $(w',s',q') \in T_{(w,s,q),t}$, $r_{(w,s,q),t}(w',s',q') = r_{w,t}$. Moreover, $\tau_{(w,s,q),t}((w',s',q')) = \tau_{w,t}(w')$. Since $r_{w,t}$ is an accepting of $\mathcal{A}_\vartheta$ on $T_{w,t}$, thus $r_{(w,s,q),t}$ is an accepting run of $\mathcal{A}_\vartheta$ on $T_{(w,s,q),t}$.

Finally, we uniformize $\widehat{\mathfrak{M}}$ in the sense that for each $w, (w_1,s,q), (w_2,s,q) \in \widehat{T}$,

$$\text{if } (w,(w_1,s,q)) \in E \text{ and } (w,(w_2,s,q)) \in E \text{ and } \widehat{\tau}(w_1) = \widehat{\tau}(w_2)$$

then, we substitute the subtree $(\widehat{T}, \widehat{\tau}'')$ of $\widehat{\mathfrak{M}}$ rooted at (w_2,s,q) with the subtree $(\widehat{T}, \widehat{\tau}')$ of $\widehat{\mathfrak{M}}$ rooted at (w_1,s,q).

Now, we argue that the resulting $\widehat{\mathfrak{M}}' = (\widehat{T}', \widehat{\tau}')$ continues satisfying condition 2'. The idea is that for every $(T_{w',t}, \tau_{w',t})$ used to witness condition 2' of $\widehat{\mathfrak{M}}$ that contains (w_2, s, q) we construct a $(T_{w',t}, \tau'_{w',t})$ witnessing condition 2' of $\widehat{\mathfrak{M}}'$ by replacing the subtree of $(T_{w',t}, \tau_{w',t})$ rooted in (w_2, s, q) with an appropriate tree $(T_{w',t}, \tau''_{w',t})$ with root (w_1, s, q).

First, note that since $\widehat{\mathfrak{M}}$ satisfies condition 2', there is a 2^{cn_X}-labeled $\sharp^c_E$-ary tree

$$(T_{(w_1,s,q),s}, \tau_{(w_1,s,q),s})$$

for (w_1, s, q) with $\overline{s} \in \widehat{\tau}(w, s, q)$ fulfilling 2'(a)-(c). Now, we obtain $(T_{w',t}, \tau'_{w',t})$ by using the subtree $(T_{(w_1,s,q),s}), \tau'_{(w_1,s,q),s})$ of

$$(T_{(w_1,s,q),s}, \tau_{(w_1,s,q),s})$$

rooted in (w_1, s, q) to replace the subtree of $(T_{w',t}, \tau_{w',t})$ rooted in (w_2, s, q). Formally, we define $\tau'_{w',t}$ and $r'_{w',t}$ as follows:

$$\tau'_{w',t}(w) = \begin{cases} \tau'_{(w_1,s,q),s}(w) & \text{if } w \in (T_{(w_1,s,q),s}), \tau'_{(w_1,s,q),s}) \\ \tau_{w',t}(w) & \text{otherwise} \end{cases}$$

$$r'_{w',t}(w) = \begin{cases} r_{(w_1,s,q),s}(w) & \text{if } w \in (T_{(w_1,s,q),s}), \tau'_{(w_1,s,q),s}) \\ r_{w',t}(w) & \text{otherwise} \end{cases}$$

Now, from the construction of $\widehat{\mathfrak{M}}'$, it is clear that it satisfies condition 2'. In particular, conditions 2'(a) and (b) are trivially satisfied.

Now, we can safely eliminate duplicated subtrees without violating condition 2', and then get a proper quasi-model that is k-ary tree: we eliminate from $\widehat{\mathfrak{M}}'$ duplicated subtrees rooted at the direct successors of ε. Recall that $k = |\mathsf{qw}(\varphi)| \cdot |\mathsf{tp}(\varphi)| \cdot |Q_2|$, so indeed the quasi-models obtained after removing duplicated subtrees are k-ary trees.

❑

The desired 2ABTA $\mathcal{A}$ will thus run on k-ary trees. For simplicity and because Theorem 3.3 admits any outdegree, we can actually assume both $\mathcal{A}_{\overline{\varphi}}$ and $\mathcal{A}_{\vartheta}$ to run on trees of outdegree k (this does not result in a change to the state set Q_2, thus does not impact k). Since 2ABTAs are trivially closed under intersection, it suffices to construct separate 2ABTAs $\mathcal{A}_1$ and $\mathcal{A}_2$ to deal with Conditions 1 and 2 of proper quasi-models. To obtain $\mathcal{A}_1$, manipulate $\mathcal{A}_{\overline{\varphi}}$ so that it has input alphabet $\mathsf{qw}(\varphi)$ and each symbol (S_1, S_2) is treated as $\overline{S_2}$, and view the resulting automaton as a 2ABTA. The 2ABTA $\mathcal{A}_2 = (Q, \Sigma, \delta, \{q_0\}, F)$ verifies Condition 2 by simulating a run of $\mathcal{A}_\vartheta$ for every $w \in T$ with $\tau(w) = (S_1, S_2)$ and every $s \in S_1$. Formally, set

$$Q_2^* = Q_2 \cup \{*\} \quad \text{and} \quad Q = \{q_0\} \cup (Q_2 \times Q_2^*) \cup (Q_2 \times 2^{\mathsf{cn}_X} \times Q_2^*) \cup F_2$$

and the transition relation δ is as follows, for $\omega = (S_1, S_2)$:

$$
\begin{aligned}
\delta(q_0, \omega, \cdot) &= \textstyle\bigwedge_{i=1}^{k}(i, q_0) \wedge \bigwedge_{s\in S_1} \bigvee_{q\in Q_2}(0, (q, s, *)) \\
\delta((q, q'), \omega, \cdot) &= \textstyle\bigvee_{s\in S_1}(0, (q, s, q')) \\
\delta((q, s, q'), \omega, \mathsf{t}) &= \textstyle\bigvee_{(q_1,\ldots,q_k)\in\delta_2(q,s)|q'\in\{q_1,\ldots,q_k\}} \bigwedge_{i=1}^{k}(i, (q_i, *)) \\
\delta((q, s, q'), \omega, \mathsf{f}) &= \textstyle\bigvee_{p\in Q_2}(-1, (p, q'))) \wedge \\
&\qquad \textstyle\bigvee_{(q_1,\ldots,q_k)\in\delta_2(q,s)|q'\in\{q_1,\ldots,q_k\}} \bigwedge_{i=1}^{k}(i, (q_i, *))
\end{aligned}
$$

where $\cdot$ in the third component means that the transition exists both when the component is t and f, and '$*$' behaves like a wildcard for all states of Q_2 with the test $* \in \{q_1, \ldots, q_k\}$ always being successful. Finally, we set $F = F_2$. Note that runs of the original NBTA $\mathcal{A}_\vartheta$ must start at the root of the tree, but when simulating $\mathcal{A}_\vartheta$ in $\mathcal{A}$, we have to start at an arbitrary tree node. In fact, this is the reason why we need a *2-way* automaton and states of the form (q, q') and (q, s, q'), which intuitively mean that we are currently simulating a run of $\mathcal{A}_\vartheta$ in state q and have already decided to assign q' to some successor of the current node (we do not need to memorize *which* successor since the transitions of $\mathcal{A}_\vartheta$ are closed under permuting the successors). The state (q, s, q') additionally selects an $s \in S_1$ for the current tree node, see Condition 2. This finishes the definition of the components of $\mathcal{A}$.

Lemma 3.23. *For a temporal* $\mathrm{CTL}^*_{\mathcal{ALC}}$ *TBox* φ *and* $k \geq 0$*, one can construct a 2ABTA* $\mathcal{A} = (Q, \Sigma, \delta, \{q_0\}, F)$ *in time* $\mathsf{poly}(|Q| + k)$ *such that* $L(\mathcal{A}) = \mathsf{Mod}_k(\varphi)$*,* $\Sigma = 2^{\mathsf{qw}(\varphi)}$*,* $|Q| \in 2^{2^{\mathsf{poly}(|\varphi|)}}$ *and* $|Q| \in 2^{\mathsf{poly}(|\varphi|)}$ *when* φ *is a temporal* $\mathrm{CTL}_{\mathcal{ALC}}$ *TBox.*

Proof. Due to Lemma 3.22, we can set $k := |\mathsf{qw}(\varphi)|\cdot|\mathsf{tp}(\varphi)|\cdot|Q_2|$. Now, $\mathcal{A}$ is constructed as presented in Section 3.6, that is, we construct separately 2ABTAs $\mathcal{A}_1$ and $\mathcal{A}_2$ to deal with conditions 1 and 2 of proper quasimodels. $\mathcal{A}_1$ is a variant of an standard CTL* automaton, hence its correctness is trivial (*cf.* Section 3.6). In the case of $\mathcal{A}_2$, first let us recall that $\mathcal{A}_2$ is devised to check condition 2 of Definition 3.13, that is, to check whether a quasi-model $\mathfrak{M} = (T, \tau)$ of φ satisfies the following:

> for all $w \in T$ with $\tau(w) = (S_1, S_2)$ and all $s \in S_1$, there is a 2^{cn_X}-labeled tree (T, τ') such that
>
> a) $\tau'(w) = \overline{s}$;
>
> b) for all $w' \in T$ with $\tau(w') = (S'_1, S'_2)$, there is an $s' \in S'_1$ such that $\tau'(w') = \overline{s'}$;
>
> c) ε satisfies $\vartheta = \mathbf{A}\Box \bigwedge\limits_{X_C\in\mathsf{cn}_X} (X_C \leftrightarrow \overline{C})$.

Notably, $\mathcal{A}_2$ verifies Condition (c) by simulating a run of the NBTA $\mathcal{A}_\vartheta = (Q_2, \Sigma_2, \delta_2, Q_2^0, F_2)$ accepting the models of ϑ, for every $w \in T$ with $\tau(w) = (S_1, S_2)$ and every $s \in S_1$. Moreover, it takes care of conditions 2(a)-(b) by properly using alternation and the 2-way. From now on we denote with "$s \in \tau(w)$" the fact that $s \in S_1$ when $\tau(w) = (S_1, S_2)$.

$\mathcal{A}_2$ is *sound*. To check: given an accepting run (T_r, r) on $\mathfrak{M} = (T, \tau)$, then $\mathfrak{M}$ fulfills condition 2. Recall that a run of $\mathcal{A}$ on $\mathfrak{M}$ is a $T \times Q$-labeled tree. From the definition of $\mathcal{A}$ is not difficult to see that $\mathfrak{M}$ satisfies condition 2 for $w = \varepsilon$: by definition of a run, $r(\varepsilon) = (\varepsilon, q_0)$, and moreover,

$$\delta(q_0, \tau(\varepsilon), \mathsf{t}) = \bigwedge_{i=1}^{k} (i, q_0) \wedge \bigwedge_{s \in S_1} \bigvee_{q \in Q_2} (0, (q, s, *)) \qquad (\dagger).$$

First, we analyze the second conjunct of the transition relation defined above. We have then that for all $s \in \tau(\varepsilon)$, $\bigvee_{q \in Q_2} (0, (q, s, *))$. Hence, for all $s \in \tau(\varepsilon)$ and q' with the test $* \in \{q_1, \ldots, q_n\}$ successful, by definition of $\delta((q, s, q'), \tau(\varepsilon), \mathsf{t})$, the following holds:

$$\bigvee_{(q_1, \ldots, q_k) \in \delta_2(q,s) | q' \in \{q_1, \ldots, q_k\}} \bigwedge_{i=1}^{k} (i, (q_i, *)).$$

Intuitively, for all $s \in \tau(\varepsilon)$ we will begin to simulate a run of $\mathcal{A}_\vartheta$. For all successors $\varepsilon \cdot i$, $i \in (1, k)$ of ε we have that

$$\delta((q, q'), \tau(\varepsilon \cdot i), \cdot) = \bigvee_{s \in \tau(\varepsilon \cdot i)} (0, (q, s, q'))$$

and therefore

$$\delta((q, s, q'), \omega, \mathsf{f}) = \bigvee_{p \in Q_2} (-1, (p, q'))) \;\; \wedge \bigvee_{(q_1, \ldots, q_k) \in \delta_2(q,s) | q' \in \{q_1, \ldots, q_k\}} \bigwedge_{i=1}^{k} (i, (q_i, *)).$$

Since we began from the root we have that the first conjunct of the transition above is trivially satisfied. With the second conjunct we continue simulating the run of $\mathcal{A}_\vartheta$ in the successors of the current node. In this way we proceed to simulate the runs of $\mathcal{A}_\vartheta$ towards $\mathfrak{M}$. Therefore, for all $s \in \tau(\varepsilon)$ we identify a 2^{cn_X}-labeled tree satisfying 2(a)-(c). In particular, note that since we assume that (T_r, r) is an accepting run of $\mathcal{A}$ on $\mathfrak{M}$, and the recurrent states F of $\mathcal{A}$ are given in terms of the recurrent states F_2 of ϑ, then we obtain an accepting run of $\mathcal{A}_\vartheta$ on all such 2^{cn_X}-labeled trees.

Moreover, it is clear that, by the first conjunct $\bigwedge_{i=1}^{k} (i, q_0)$ of $\delta(q_0, \omega, \cdot)$ of $(\dagger)$ above, we will simulate a run of $\mathcal{A}_\vartheta$ for all $w \neq \varepsilon \in T$ and $s \in \tau(w)$. The simulation will proceed as discussed above. In particular, the first conjunct $\bigvee_{p \in Q_2} (-1, (p, q')))$ of $\delta((q, s, q'), \tau(w), \mathsf{f})$ (f holds since now we are initiating the simulation not at the root) takes care of starting the simulation of $\mathcal{A}_\vartheta$ at the root as required by condition 2.

$\mathcal{A}_2$ is *complete*. To check: given a proper-quasimodel $\mathfrak{M} = (T, \tau)$, then $\mathfrak{M}$ is accepted by $\mathcal{A}_2$, that is, there is an accepting run (T_r, r) of $\mathcal{A}_2$ on $\mathfrak{M}$. Since $\mathfrak{M}$ is proper, for each $w \in T$ and $s \in \tau(w)$, there is a 2^{cn_X}-labeled tree $(T_{w,s}, \tau_{w,s})$ satisfying 2. Note that, by Lemma 3.22, we can assume that $(T_{w,s}, \tau_{w,s})$ is a k-ary tree. Due to 2(c) there is an accepting run (T_r, r) of $\mathcal{A}_\vartheta$ on $(T_{w,s}, \tau_{w,s})$. Since $\mathcal{A}$ simulates the runs of $\mathcal{A}_\vartheta$, its clear that we can use r to define a run of $\mathcal{A}_2$ on $(T_{w,s}, \tau_{w,s})$. Therefore, by simulating the accepting runs of $\mathcal{A}_\vartheta$ on all such $(T_{w,s}, \tau_{w,s})$

we obtain (T_r, τ_r), an accepting run of $\mathcal{A}_2$ on $\mathfrak{M}$. Moreover, the recurrent states F are given in terms of F_2, and thus (T_r, τ_r) is an accepting run of $\mathcal{A}_2$ on $\mathfrak{M}$.

Finally, recall that the number of states $|Q_2|$ of $\mathcal{A}_\vartheta$ is in $2^{2^{\mathsf{poly}(\vartheta)}}$ for $\mathrm{CTL}_{\mathcal{ALC}}{}^*$-TBoxes, and in $2^{\mathsf{poly}(\vartheta)}$ for $\mathrm{CTL}_{\mathcal{ALC}}$-TBoxes. Moreover, recall that the states Q of $\mathcal{A}_2$ are the following,

$$Q = \{q_0\} \cup Q_1 \cup (Q_1 \times 2^{\mathsf{cn}_X}) \cup (Q_1 \times Q_1) \cup (Q_1 \times 2^{\mathsf{cn}_X} \times Q_1)$$

Therefore, $|Q| \in 2^{2^{\mathsf{poly}(\varphi)}}$ for $\mathrm{CTL}_{\mathcal{ALC}}{}^*$-TBoxes, and $|Q| \in 2^{\mathsf{poly}(\varphi)}$ if we consider $\mathrm{CTL}_{\mathcal{ALC}}$-TBoxes.
The number of states Q_1 of $\mathcal{A}_1$ is in $2^{2^{\mathsf{poly}(\vartheta)}}$ for $\mathrm{CTL}_{\mathcal{ALC}}{}^*$-TBoxes, and in $2^{\mathsf{poly}(\vartheta)}$ for $\mathrm{CTL}_{\mathcal{ALC}}$-TBoxes. Therefore, the number of states of $\mathcal{A}$, which is the intersection of $\mathcal{A}_1$ and $\mathcal{A}_2$, is in $2^{2^{\mathsf{poly}(\vartheta)}}$, and $2^{\mathsf{poly}(\vartheta)}$ if we consider $\mathrm{CTL}_{\mathcal{ALC}}$-TBoxes.

❏

Now, it remains to recall that the emptiness problem for 2ABTAs can be decided in exponential time in the number of states, we obtain then the following result.

Theorem 3.24. *Satisfiability of temporal TBoxes is in* 2EXPTIME *for* $\mathit{CTL}_{\mathcal{ALC}}$ *and in* 3EXPTIME *for* $\mathit{CTL}^*_{\mathcal{ALC}}$.

3.5.3 A 2EXPTIME Lower Bound for Temporal TBox Satisfiability for $\mathrm{CTL}_{\mathcal{ALC}}$

In this section, we prove a 2EXPTIME lower bound for temporal TBox satisfiability for $\mathrm{CTL}_{\mathcal{ALC}}$ by a reduction of the word problem of an exponentially space bounded alternating Turing machine. This result shows that the additional expressive power obtained by the introduction of temporal TBoxes is reflected in an exponential increase in the computational complexity, compared to the case when only temporal concepts are allowed. Before presenting the reduction, we introduce the basic notions of alternating Turing machines.

Alternating Turing machines are a generalization of nondeterministic Turing machines in which existential and universal quantification alternate in a computation. Formally, an *alternating Turing machine (ATM)* is a tuple $\mathcal{M} = (Q, \Sigma, \Gamma, q_0, \delta)$, where:

- $Q = Q_\exists \uplus Q_\forall \uplus \{q_a, q_r\}$ is a finite set of *states* containing pairwise disjoint sets of *existential states* $Q_\exists$, *universal states* $Q_\forall$, an *accepting state* $\{q_a\}$ and a *rejecting* state $\{q_r\}$;
- Σ is an *input alphabet* and Γ a *working alphabet*, containing the *blank symbol* $\llcorner\!\lrcorner$, such that $\Sigma \subseteq \Gamma$ and $\llcorner\!\lrcorner \notin \Sigma$;
- $q_0 \in Q_\exists \cup Q_\forall$ is the *initial state*;
- δ is a *transition relation* of the form $\delta \subseteq Q \times \Gamma \times Q \times \Gamma \times \{\ell, r, n\}$.

We write $(q', b, m) \in \delta(q, a)$ for $(q, a, q', b, m) \in \delta$. We assume that $q \in Q_\exists \uplus Q_\forall$ implies $\delta(q, b) \neq \emptyset$ for all $b \in \Gamma$, and $q \in \{q_a, q_r\}$ implies $\delta(q, b) = \emptyset$ for all $b \in \Gamma$. Intuitively, the triple (q', b, m) describes the transition to state q', involving overwriting of symbol a with b and a shift of the head to the left ($m = l$), to the right ($m = r$) or no shift ($m = n$).

A *configuration* of $\mathcal{M}$ is a word wqw' with $w, w' \in \Gamma^*$ and $q \in Q$, stating that the tape contains the word ww' (with only blanks before and behind it), the machine is in state q, and the head is on the leftmost symbol of w'. The *successor configurations* of a configuration wqw' are defined in terms of the transition relation δ. A *halting configuration* is of the form wqw' with $q \in \{q_a, q_r\}$. A *computation path of an ATM* $\mathcal{M}$ on a word w is a (finite or infinite) sequence of configurations $c_1, c_2, \ldots$ such that $c_1 = q_0 w$ and c_{i+1} is a successor configuration of c_i for $i \geq 0$.

We consider in this thesis ATMs that have only *finite* computation paths on any input. Since this case is simpler than the general one, we define acceptance for ATMs with finite computation paths only, and refer to [31] for the full definition. Let $\mathcal{M}$ be such an ATM, a halting configuration is *accepting* iff it is of the form $wq_a w'$; a non-halting configuration $c = wqw'$ is accepting if at least one (all) successor configuration is accepting for $q \in Q_\exists$ ($q \in Q_\forall$, respectively). An ATM *accepts* an input w if the *initial configuration* $q_0 w$ is accepting. We use $L(\mathcal{M})$ to denote the language accepted by $\mathcal{M}$, that is,

$$L(\mathcal{M}) = \{w \in \Sigma^* \mid \mathcal{M} \text{ accepts } w\}.$$

The *word problem for* $\mathcal{M}$ is the following decision problem: given a word $w \in \Sigma^*$, does $w \in L(\mathcal{M})$ holds?

There exists an *exponentially space bounded* ATM $\mathcal{M}$ with only finite computations whose word problem is 2ExpTime-hard [31, Theorem 3.4]. Furthermore, by [31, Theorem 2.6], we can assume that there exists a polynomial p, such that the length of every computation path of $\mathcal{M}$ on $w \in \Sigma^*$ of length n is bounded by $2^{2^{p(n)}}$, and all configurations wqw' in such paths satisfy that $|ww'| \leq 2^{p(n)}$.

We sometimes see an accepting computation of an ATM $\mathcal{M}$ on a word w as an acceptance computation tree. An *acceptance tree of* $\mathcal{M}$ on w is a finite tree whose nodes are labeled with configurations, such that:

- the root node is labeled with the initial configuration $q_0 w$,
- if a node s in the tree is labeled with wqw', $q \in Q_\exists$, then s has exactly one successor, and this successor is labelled with a successor configuration of wqw',
- if a node s in the tree is labeled with wqw', $q \in Q_\forall$, then there is exactly one successor of s for each successor configuration of wqw',
- leaves are labeled with accepting halting configurations.

For a more detailed discussion on alternation and the relation of deterministic (space) time complexity classes with their alternating analogs, please consult the paper *Alternation* [31], by Chandra *et al.*

Theorem 3.25. *Satisfiability of temporal CTL$_{\mathcal{ALC}}$ TBoxes is* 2-ExpTime*-hard.*

Proof. The proof is by reduction of the word problem for exponentially space bounded alternating Turing machines. Let $\mathcal{M} = (Q, \Sigma, \Gamma, q_0, \delta)$ be such an ATM with a 2ExpTime-hard word problem, and $w \in \Sigma$ the input of length n. Our aim is to construct in polynomial time a temporal $\mathrm{CTL}_{\mathcal{ALC}}$ TBox $\varphi_{\mathcal{M},w}$ such that $\varphi_{\mathcal{M},w}$ is satisfiable *iff* $\mathcal{M}$ accepts w.

The basic idea is to construct $\varphi_{\mathcal{M},w}$ such that its models correspond to accepting computation trees of $\mathcal{M}$ on w. In particular, the computation tree is represented by the temporal development of a single domain element d_0 with each time point w corresponding to a tape cell and a configuration of $\mathcal{M}$ being represented by exponentially many consecutive time points (see Figure 3.3). To transport symbols in our ATM reduction, we use a suitable set of binary counters to manage distances in the tree. The resulting TBox $\varphi_{\mathcal{M},w}$ has the form $\mathbf{A}\Box\psi$ with ψ a Boolean combination of the CIs $C \sqsubseteq D$ to be introduced below. We use the following signature:

- concept names S_a, W_a for each $a \in \Gamma$;
- concept names Q_q for each $q \in Q$;
- concept names $N_{q,a,m}, M_{q,a,m}$ for every $(q, a, m) \in \Theta$, where $\Theta = \{(q, a, m) \mid (q', b, q, a, m) \in \delta$ for any $b \in \Gamma$ and $q' \in Q\}$;
- concept name H to mark the cells to the right of the head;
- concept names $X_0, \ldots, X_{n-1}$, $C_0, \ldots, C_{n-1}$, $C'_0, \ldots, C'_{n-1}$, $U_0, \ldots, U_{n-1}$ for encoding exponential counters;
- auxiliary concept names $\mathsf{ZeroTape}$, $\mathsf{EndTape}$, $\mathsf{ZeroHead}$, $\mathsf{ZeroHead}'$, $\mathsf{EndHead}'$.

Throughout the reduction we use several counters over the temporal tree structure which allow to identify time points exponentially far away. Each counter consists of a number of inclusions of polynomial size; constraints (3.2)-(3.3) implement an exemplary *counter* based on atomic concepts X_i for $0 \leq i < n$, which simulate the bits of a number in binary.

For every $0 \leq j < i < n$,

$$\begin{aligned} \neg X_i \sqcap \neg X_j &\sqsubseteq \mathbf{A}\bigcirc\neg X_i, \\ X_i \sqcap \neg X_j &\sqsubseteq \mathbf{A}\bigcirc X_i, \end{aligned} \tag{3.2}$$

For every $0 \leq j < n$,

$$\begin{aligned} \neg X_j \sqcap X_{j-1} \sqcap \ldots \sqcap X_1 &\sqsubseteq \mathbf{A}\bigcirc X_j, \\ X_j \sqcap X_{j-1} \sqcap \ldots \sqcap X_1 &\sqsubseteq \mathbf{A}\bigcirc\neg X_j. \end{aligned} \tag{3.3}$$

We use the abbreviations Zero and End to respectively denote:

$$\bigsqcap_{j=0}^{n-1} \neg X_j \qquad \bigsqcap_{j=0}^{n-1} X_j \tag{3.4}$$

Later on, we instantiate the above pattern to encode different counters X.

Now, we can enforce the conditions that ensure that the model actually represents an accepting configuration of $\mathcal{M}$ on w. We utilize counter X to define constraints over a *configuration* of the ATM. We use $\mathsf{ZeroTape}$ and $\mathsf{EndTape}$, instantiating the abbreviations introduced in (3.4).

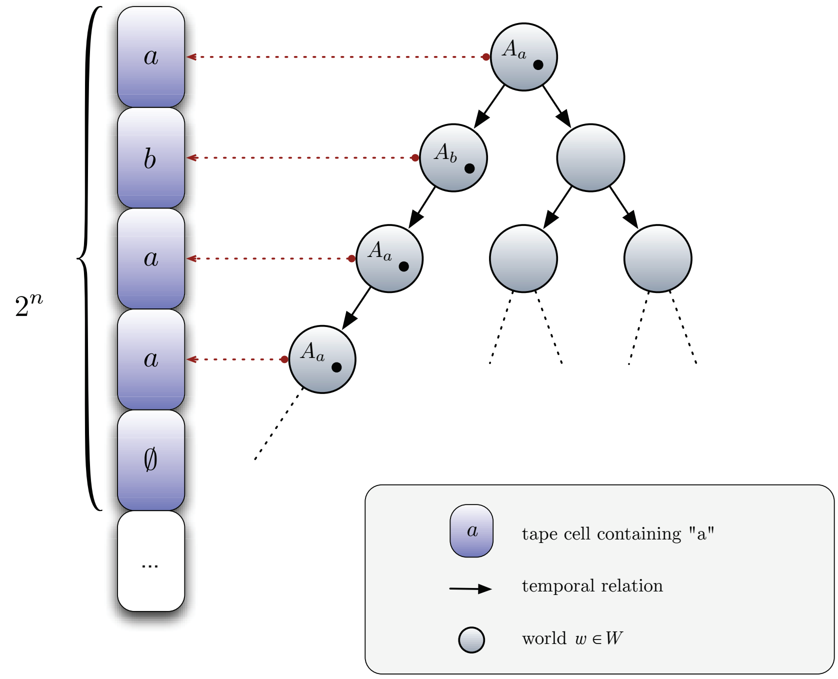

Figure 3.3: Embedding of ATM tapes in $\mathrm{CTL}_{\mathcal{ALC}}$ models.

We begin by enforcing standard structural requirements of ATM; we don't synchronize yet the content of successor configurations. Recall that we use concepts Q_q to denote the current state and the position of the head. First, we ensure that exists always a time successor until we reach the head in a halting configuration (3.5). Moreover, we require that in each configuration every tape cell is labeled with at most one state variable Q_q (3.6), and that in every configuration at most one tape cell is indeed labeled with a state Q_q (3.7). Finally, we enforce that each tape cell is labeled with exactly one alphabet letter (3.8).

$$\neg(Q_a \sqcup Q_r) \sqsubseteq \mathbf{E}\bigcirc\top \tag{3.5}$$

$$\top \sqsubseteq \bigsqcap_{q,q' \in Q} \neg(Q_q \sqcap Q_{q'}) \tag{3.6}$$

$$(\bigsqcup_{q \in Q} Q_q \sqsubseteq \mathbf{A}\bigcirc H) \,\wedge\, (H \sqcap \neg\mathsf{EndTape} \sqsubseteq \mathbf{A}\bigcirc H) \,\wedge\, (H \sqsubseteq \neg \bigsqcup_{q \in Q} Q_q) \tag{3.7}$$

$$\top \sqsubseteq \bigsqcap_{a,a' \in \Gamma, a \neq a'} \neg(A_a \sqcap A_{a'}) \tag{3.8}$$

We introduce a counter C capturing the position of the *head* in the tape; we use $\mathsf{ZeroHead}$ as defined in (3.4). Furthermore, we use concepts $M_{q,a,m}$ for carrying the information generated by the transition function, and use them later to determine the successor configuration. Information about the transitions is generated depending on whether the state is universal (3.9) or existential (3.10) and then carried to the end of the tape (3.11).

For every $a \in \Gamma, q \in Q_\forall$:

$$A_a \sqcap Q_q \sqsubseteq \bigsqcap_{(q'b'm) \in \delta(q,a)} M_{q',b,m} \sqcap \mathsf{ZeroHead} \tag{3.9}$$

For every $a \in \Gamma, q \in Q_\exists$:

$$A_a \sqcap Q_q \sqsubseteq \bigsqcup_{(q'b'm) \in \delta(q,a)} M_{q',b,m} \sqcap \mathsf{ZeroHead} \tag{3.10}$$

$$M_{q,a,m} \sqcap \neg\mathsf{EndTape} \sqsubseteq \mathbf{A}\bigcirc M_{q,a,m} \tag{3.11}$$

When moving to a successor configuration in order to avoid clashes in the information we create copies $N_{q,a,m}$ for concepts $M_{q,a,m}$, carrying this information over the new configuration (3.12)-(3.13). We make the standard requirement that at most one concept $N_{q,a,m}$ is true in a tape cell (3.14). To avoid further clashes while synchronizing adjacent configurations, we create an auxiliary counter C', using the standard abbreviations $\mathsf{ZeroHead}', \mathsf{EndHead}'$ (3.4), which proceeds with counting exactly from where the previous head counter terminated (3.15).

For every $(q, a, m) \in \Theta$:

$$M_{q,a,m} \sqcap \mathsf{EndTape} \sqsubseteq \mathbf{E}\bigcirc N_{q,a,m} \tag{3.12}$$

$$N_{q,a,m} \sqcap \neg\mathsf{EndTape} \sqsubseteq \mathbf{A}\bigcirc N_{q,a,m} \tag{3.13}$$

$$\top \sqsubseteq \bigsqcap_{(q,a,m) \neq (q',a',m') \in \Theta} \neg(N_{q,a,m} \sqcap N_{q',a',m'}) \tag{3.14}$$

$$(\mathsf{EndTape} \sqcap C_i \sqsubseteq C_i') \wedge (\mathsf{EndTape} \sqcap \neg C_i \sqsubseteq \neg C_i') \tag{3.15}$$

The changes imposed by the transition relation are implemented: write the new tape symbol (3.16), and place the state variable in the correct position (3.17) -(3.19). Moreover, we ensure that the transition does not push the head beyond the tape (3.20) - (3.21).

For every $(q, a, m) \in \Theta$

$$N_{q,a,m} \sqcap \mathsf{ZeroHead}' \sqsubseteq A_a \tag{3.16}$$

For every $(q, a, n) \in \Theta$

$$N_{q,a,m} \sqcap \mathsf{ZeroHead}' \sqsubseteq Q_q \tag{3.17}$$

For every $(q, a, r) \in \Theta$

$$N_{q,a,m} \sqcap \mathsf{ZeroHead}' \sqsubseteq \mathbf{A}\bigcirc Q_q \tag{3.18}$$

For every $(q, a, l) \in \Theta$

$$N_{q,a,m} \sqcap \mathsf{EndHead}' \sqsubseteq Q_q \tag{3.19}$$

For every $(q, a, l) \in \Theta$

$$\mathsf{ZeroHead}' \sqcap \mathsf{ZeroTape} \sqsubseteq \neg N_{q,a,m} \tag{3.20}$$

For every $(q, a, r) \in \Theta$

$$\mathsf{EndHead}' \sqcap \mathsf{EndTape} \sqsubseteq \neg N_{q,a,m} \tag{3.21}$$

We propagate the information of each i-th tape cell that do *not change* during the transition. This information is stored in fresh elements, i.e., new r-successors, and synchronized via the counter U with the content of the i*-th cell* in the previous configuration; we use ZeroCell and EndCell for the respective abbreviations in (3.4). Moreover, to store the label of a tape cell concept names W_a, S_a for every $a \in \Gamma$ are used. The information store by W_a and S_a is propagated to the previous configuration (3.22), and aligned at the end of the configuration (3.23). We enforce standard consistency constraints (3.24).

For every $a \in \Gamma$

$$(\neg\mathsf{EndTape} \sqcap \mathbf{E}\bigcirc W_a \sqsubseteq W_a) \wedge (\neg\mathsf{EndTape} \sqcap \mathbf{E}\bigcirc S_a \sqsubseteq S_a) \tag{3.22}$$

$$\mathsf{EndTape} \sqcap \mathbf{E}\bigcirc W_a \sqsubseteq S_a \tag{3.23}$$

$$(\top \sqsubseteq \bigsqcap_{a,a' \in \Gamma, a \neq a'} \neg(W_a \sqcap W_{a'})) \wedge (\top \sqsubseteq \bigsqcap_{a,a' \in \Gamma, a \neq a'} \neg(S_a \sqcap S_{a'})) \tag{3.24}$$

We proceed to propagate the information of each i-th tape cell not meant to change. For implementing successfully this (recall that roles change freely over time), we need to enforce all the individuals of a world to share the same alphabet symbol (3.25). A representative of each not changing cell is generated, and its label A_a is stored in W_a (3.26). Now, we synchronize the content of i-th cell with that of the i-th cell in the previous configuration (3.27).

$$\bigvee_{a \in \Gamma} (\top \sqsubseteq A_a \vee \top \sqsubseteq \neg A_a) \tag{3.25}$$

$$\neg\mathsf{ZeroHead}' \sqcap A_a \sqsubseteq \exists r.W_a \sqcap \mathsf{ZeroCell} \tag{3.26}$$

For every $b \neq a \in \Gamma$

$$A_a \sqcap S_b \sqcap \mathsf{ZeroCell} \sqsubseteq \bot \tag{3.27}$$

Recall that in our setting any computation is terminating, moreover halting configurations are the only configurations without successor configurations. Then, the input $w = a_1 \dots a_n$ is accepted if the rejecting state is not reached (3.28).

$$\top \sqsubseteq \neg Q_{q_r} \tag{3.28}$$

The initial configuration $q_0 w$ starting at A_0 is encoded as follows

$$\begin{aligned}
A_0 &= A_{a_1} \sqcap Q_{q_0} \sqcap \mathsf{ZeroTape} \sqcap \mathbf{E}\bigcirc A_1 \\
A_i &= A_{a_{i+1}} \sqcap \mathbf{E}\bigcirc A_{i+1} \\
A_n &= \mathbf{A}((A_\llcorner \sqcap \neg\mathsf{EndTape})\mathcal{U}(A_\llcorner \sqcap \mathsf{Endtape}))
\end{aligned}$$

$$\neg(\top \sqsubseteq \neg A_0) \tag{3.29}$$

We define ψ as the conjunction of the TBoxes introduced above, and $\varphi_{\mathcal{M},w}$ as $\mathbf{A}\Box\psi$. Now, it is not hard to see that the size of $\varphi_{\mathcal{M},w}$ is polynomial in n. Finally, following the intuitive meaning of each conjunct given above, it is clear that the following holds.

Proposition 3.26. *$\varphi_{\mathcal{M},w}$ is satisfiable iff $\mathcal{M}$ accepts w.*

Assume $w \in L(\mathcal{M})$, and $T_{\mathcal{M}} = (N_{\mathcal{M}}, E_{\mathcal{M}}, \mathsf{conf})$ is an acceptance tree of $\mathcal{M}$ on w, where $\mathsf{conf}(n)$ assigns configurations to nodes $n \in N_{\mathcal{M}}$. Moreover, given $\mathsf{conf}(n) = wqw'$, the function $\sharp_i(\mathsf{conf}(n))$ returns the i-th symbol of ww', and $\mathsf{h}(\mathsf{conf}(n))$ the position of the head. Our aim is to construct a model $\mathfrak{I} = (\Delta, T, \{\mathcal{I}_w\}_{w\in W})$ of $\varphi_{\mathcal{M},w}$. We begin by defining the set $W^{\mathfrak{I},\mathcal{M}} \subseteq W$ as follows:

$$W^{\mathfrak{I},\mathcal{M}} = \{w_i^n \mid n \in N_{\mathcal{M}} \wedge i < 2^{|w|}\}$$

and the set $E^{\mathfrak{I},\mathcal{M}} \subseteq E$ as follows:

$$E^{\mathfrak{I},\mathcal{M}} = \{(w_i^n, w_{i+1}^n) \mid i < 2^{|w|} \wedge n \in N^{\mathcal{M}}\} \cup \{(w^n_{2^{|w|-1}}, w_0^{n'}) \mid (n,n') \in E^{\mathcal{M}}\}.$$

Moreover, we define $\Delta = \{d_{\mathcal{M}}\} \cup \{e_{w_i^n} \mid w_i^n \in W^{\mathfrak{I},\mathcal{M}} \wedge \mathsf{h}(\mathsf{conf}(n)) \neq i\}$ and

- $A_a^{\mathfrak{I},w_i^n} = \{e \in \Delta \mid \sharp_i(\mathsf{conf}(n)) = a\}$;
- $Q_q^{\mathfrak{I},w_i^n} = \{d_{\mathcal{M}} \in \Delta \mid \mathcal{M}$ is in state q $\wedge\, \mathsf{h}(\mathsf{conf}(n)) = i\}$;
- $X_j^{\mathfrak{I},w_i^n} = \{d_{\mathcal{M}} \in \Delta \mid$ the j^{th} bit of the binary representation of i is $1\}$;
- $r^{\mathfrak{I},w_i^n} = \{(d_{\mathcal{M}}, e_{w_i^n}) \in \Delta \times \Delta\}$.

The interpretation can be standardly extended to the remaining auxiliary concepts in the construction, e.g., $M_{q,a,m}, H$, etc. For example, the concepts W_a, S_a are interpreted as follows:

$$W_a^{\mathfrak{I},w_i^n} = \{e_{w_j^n} \in \Delta \mid \sharp_j(\mathsf{conf}(n)) = a \wedge j \geq i\},$$

$$S_a^{\mathfrak{I},w_i^n} = \{e_{w_j^{n'}} \in \Delta \mid (n,n') \in E^{\mathcal{M}} \wedge e_{w_j^{n'}} \in W_a^{\mathfrak{I},w^{n'}_{2^{|w|-1}}}\}.$$

One can interpret analogously the other auxiliary concepts and its copies used to transfer and synchronize information between adjacent configurations. We next discuss how the interpretation of auxiliary counters is defined. For example, consider the Head counter implemented with concept names C_j, and its auxiliary copy Head' implemented with concept names C_j'. We define their interpretation as follows:

$C_j^{\mathfrak{I},w_i^n} = \{d_{\mathcal{M}} \in \Delta \mid h(\mathsf{conf}(n)) = m, m \geq i \wedge$
the j^{th} bit of the binary representation of $\mathsf{dist}(w_i^{n'}, w_m^n)$ is 1 $\}$,

$C'^{\mathfrak{I},w_i^{n'}}_j = \{d_{\mathcal{M}} \in \Delta \mid h(\mathsf{conf}(n)) = m, (n', n) \in E^{\mathcal{M}} \wedge$
the j^{th} bit of the binary representation of $\mathsf{dist}(w_i^{n'}, w_m^n)$ is 1 $\}$,

where $\mathsf{dist}(w_i^{n'}, w_m^n)$ is a function giving the distance modulo $2^{|w|}$ between the nodes $w_i^{n'}$ and w_m^n. We can define analogously the interpretation for the other auxiliary counters.

Finally, note that up to this point we have defined the interpretation $\mathfrak{I}' = (\Delta, T^{\mathfrak{I},\mathcal{M}}, \{\mathcal{I}_w\}_{w \in W^{\mathfrak{I},\mathcal{M}}})$ where $T^{\mathfrak{I},\mathcal{M}} = (W^{\mathfrak{I},\mathcal{M}}, E^{\mathfrak{I},\mathcal{M}})$. However, we can straightforwardly obtain $\mathfrak{I}$ from $\mathfrak{I}'$ by extending each path of $T^{\mathfrak{I},\mathcal{M}}$ to an infinite path. Most of the concepts can be interpreted as the empty set in the new added worlds, the only exception are the counter concepts: we can properly interpreted them by taking the distance from the root ε, that is, for $w \in W \backslash W^{\mathfrak{I},\mathcal{M}}$

$$X_j^{\mathfrak{I},w} = \{d_{\mathcal{M}} \in \Delta \mid \text{ the } j^{th} \text{ bit of the binary representation of } \mathsf{dist}(\varepsilon, w) \text{ is } 1\}.$$

Now, by simply inspecting the conjuncts forming $\varphi_{\mathcal{M},w}$ one can easily see that $\mathfrak{I}, \varepsilon \models \varphi_{\mathcal{M},w}$.

The other direction follows directly from the construction. Let $\mathfrak{I} = (\Delta, T, \{\mathcal{I}_w\}_{w \in W})$ be a model of $\varphi_{\mathcal{M},w}$, to retrieve an acceptance tree of $\mathcal{M}$ on w, we pick a $d \in \Delta$ such that $d \in A_0^{\mathfrak{I},\varepsilon}$, and follow its evolution through the paths of $\mathfrak{I}$, and collect the information of the entire computation. Note that $\mathfrak{I}$ is infinite but since the transitions of the ATM are properly simulated in the encoding, then the ATM trees embedded in $\mathfrak{I}$ are finite.

❏

Theorem 3.27. *Satisfiability of temporal CTL$_{\mathcal{ALC}}$-TBoxes is 2*EXPTIME*-complete.*

3.6 Conclusions

Towards the construction of more useful temporal DLs it is essential to consider different formalisms of time, providing TDLs with different capabilities for modeling specific temporal aspects of knowledge. For example, many ontology applications require to distinguish between *possible* and *necessary* future developments of knowledge. With this in mind, we investigated combinations of DLs with the branching-time temporal logics CTL and CTL*. We focused on the study of the computational complexity of branching-time TDLs based on the traditional DLs $\mathcal{EL}$ and $\mathcal{ALC}$. We began by considering the case where temporal operators can be applied only to concepts: for CTL$_{\mathcal{ALC}}$ and CTL$^*_{\mathcal{ALC}}$, we devised a uniform algorithm for satisfiability based on a combination of type-elimination and automata techniques. This provided us with EXPTIME and 2EXPTIME tight upper bounds for CTL$_{\mathcal{ALC}}$ and CTL$^*_{\mathcal{ALC}}$, respectively. In the case of $\mathcal{EL}$, we concentrated on fragments of CTL$_{\mathcal{EL}}$. Notably, we identified the polytime fragment CTL$^{\mathbf{E}\Diamond}_{\mathcal{EL}}$. Moreover, we showed that most of the remaining candidate fragments of CTL$_{\mathcal{EL}}$ are hard for PSPACE and EXPTIME. Finally, we considered more expressive variants of CTL$_{\mathcal{ALC}}$ and CTL$^*_{\mathcal{ALC}}$, allowing for the application of temporal operators not only to concepts but also to TBoxes. Again, we used a uniform approach to satisfiability of temporal CTL$_{\mathcal{ALC}}$ and

$CTL^*_{\mathcal{ALC}}$ TBoxes based on automata on infinite trees. We obtained 2EXPTIME and 3EXPTIME upper bounds for $CTL_{\mathcal{ALC}}$ and $CTL^*_{\mathcal{ALC}}$, respectively. For $CTL_{\mathcal{ALC}}$, we were able to prove a 2EXPTIME matching lower bound. This shows that the presence of temporal concepts and TBoxes indeed leads to an increase on the computational complexity.

This chapter provided us with a better understanding of the computational complexity of branching-time TDLs, for which only non-elementary upper bounds were known. Remarkably, we identified the first fragment based on traditional TLs (LTL, CTL, CTL*) for which reasoning is easier than in the $\mathcal{ALC}$ variant. Some interesting problems remain open: for instance, to establish the precise complexity of satisfiability of temporal $CTL^*_{\mathcal{ALC}}$-TBoxes, which is currently open between 2EXPTIME and 3EXPTIME. Another interesting research line is the study of branching-time TDLs based on the DL-Lite family of DLs. Remarkably, linear-time TDLs based on DL-Lite allow to effectively reason about the temporal evolution of of roles. It also seems natural to increase the expressive power of the branching time component as demanded by applications. This includes capturing statements such as '*it is likely* that an irregular mole develops into a melanoma in the future' and 'all students will graduate within 8 semesters'. For the former, one can look at TDLs based on *probabilistic CTL* [27], and for the latter, one can look at TDLs based on *metric temporal logics* [65].

4 Description Logics of Change

In this chapter, we investigate two-dimensional DLs for representing and reasoning about changes of knowledge over time. *Description Logics of Change* are constructed by combining the modal logic $\mathbf{S5}$ with classical DLs. We concentrate on the investigation of DLs of change based on $\mathcal{EL}$ and its extension $\mathcal{ELI}$, and on the expressive DL $\mathcal{ALCO}$. The main technical contributions are algorithms for subsumption and satisfiability, and tight complexity bounds that range from PSPACE to NEXPTIME and 2EXPTIME.

4.1 Introduction

Temporal description logics, as discussed previously, appeared as a reaction to the inability of classical description logics to represent and reason about dynamic and temporal aspects of knowledge. Temporal description logics that emerge from combining DLs and TLs in the spirit of multi-dimensional DLs [38] provide different expressive power and have different computational properties depending on whether temporal operators are applied to concepts, roles or axioms. Notably, TDLs constructed by combining the temporal logics LTL or CTL with $\mathcal{ALC}$ become undecidable in the presence of a global TBox as soon as temporal operators are applied not only to concepts but also to roles, or rigid roles are allowed [58]. In other words, in the presence of temporal concepts we cannot effectively reason about the temporal evolution of binary relations. Alas, this becomes an important drawback for many relevant applications. For example, to accurately model the medical term '*genetic disorder*' which refers to a genetic disease, we require a rigid role $\mathsf{hasGeneDisorder}$. Intuitively, this is the case since such a disorder will remain with a person for a life-time. As another example, consider the term '*PhD candidate*' which refers to a student that eventually submits a dissertation. We can properly model this term with the concept $\mathsf{Student} \sqcap \exists\Diamond\mathsf{submits}.\mathsf{Dissertation}$, which (via the temporal role $\Diamond\mathsf{submits}$) states that there is a time point in the future when he will submit a dissertation.

Two recent investigations showed that this limitation can be overcome by weakening either the temporal or the DL component [10, 8]. In the prominent proposal by Artale *et. al.* [8], LTL is maintained as the temporal component while the DL component is weakened to members of the lightweight *DL-Lite family* [4]. A complementary approach was presented by Artale *et al.* [10] where the DL component is given by the highly expressive DL $\mathcal{ALCQI}$, and the temporal one by the modal logic $\mathbf{S5}$. The latter approach gave rise to *Description Logics of Change (DLCh)* in which the application of $\mathbf{S5}$ modalities to different pieces of DL syntax allows to model the change of knowledge over time without differentiating between changes in the past or the future. The main objective of the research carried out by Artale *et al.* [10] was the design of a TDL capable to capture *temporal entity-relationship models (TER)* used in the design of temporal databases. Hence the need of considering the highly expressive DL $\mathcal{ALCQI}$ to faithfully capture constraints present in TER models, such as *disjointness, covering constraints* and *cardinality constraints* [6]. Describing temporal conceptual database models is indeed one prominent application of TDLs, but DLs of change are also well-suited for modeling important dynamic aspects of knowledge such as *versioning* and *evolution*. As an example, consider the term '*java language*' which refers to a language which is a programing language in the computer science field, and it is also a natural language spoken by the Javanese people. We can model this using the following axioms:

$$\mathsf{JavaLanguage} \sqsubseteq \mathsf{ProgrammingLanguage} \sqcap \Diamond \mathsf{NaturalLanguage}$$
$$\mathsf{NaturalLanguage} \sqcap \mathsf{ProgrammingLanguage} \sqsubseteq \bot$$

Intuitively, when identifying each possible world with a possible version, the first axiom states that a java language is a programming language, and that there is a version in which it is a natural language. The second axiom ensures that a language cannot be a natural language and a programming language in the same version. As another example, using the following axiom we can express the term '*Mortal*' which refers to a living being that eventually will be dead.

$$\mathsf{Mortal} \sqsubseteq \neg\mathsf{Dead} \sqcap \Diamond \mathsf{Dead}$$

Intuitively, this axiom states that each instance of mortal eventually will evolve to a dead entity. In the light of the high 2EXPTIME-completeness for $\mathbf{S5}_{\mathcal{ALCQI}}$ [10], the design of well-behaved DLs of change becomes crucial for applications requiring to capture change of knowledge over time. The aim of this chapter is thus to investigate DLs of change based on lighter DLs than $\mathcal{ALCQI}$, which (in principle) could lead to the construction of computationally better-behaved logics. We concentrate on developing algorithms for subsumption and the satisfiability, and on providing tight complexity bounds. First, we study a DLCh based on the DL $\mathcal{ALCO}$ in the case where $\mathbf{S5}$-modalities are applied only to concepts and a global TBox is considered. Note that in the presence of temporal roles already reasoning in $\mathbf{S5}_{\mathcal{ALC}}$ becomes 2EXPTIME-hard, hence we look at $\mathbf{S5}_{\mathcal{ALCO}}$ allowing only for local roles. Later, we revisit the idea of designing TDLs allowing to reason about the temporal evolution of roles. Notably, since the combination of LTL and $\mathcal{EL}$ allowing for temporal concepts in the presence of a global TBox and rigid roles is as complex as the $\mathcal{ALC}$ variant [7] (which, as mentioned above, is undecidable), it is fundamental to investigate whether it is possible to design DLChs based on $\mathcal{EL}$ in which we can reason about the temporal evolution of concepts and roles, and reasoning is less complex.

Contributions: Our investigation starts with $\mathbf{S5}_{\mathcal{ALCO}}$ in the case where modalities can be applied only to concepts, concentrating on global TBoxes. Notably, we provide a NEXPTIME lower bound using a reduction of the $2^n \times 2^n$-tiling problem, showing thus that the computational complexity of $\mathbf{S5}_{\mathcal{ALCO}}$ is higher than in the component logics. This result comes as a surprise, since the computational complexity of TDLs allowing only for temporal concepts normally does not exceed that of their components. Later, we consider $\mathbf{S5}_{\mathcal{EL}}$ in the case where $\mathbf{S5}$-modalities can be applied to both concepts and roles, concentrating again on global TBoxes. For this logic, it was known that subsumption is in 2EXPTIME and PSPACE-hard. It is interesting to note that, until now, any two-dimensional extension of $\mathcal{EL}$ allowing for modalities to be applied to roles has turned out to have the same complexity as the corresponding extension of the DL $\mathcal{ALC}$, see e.g. [7]. Since subsumption in the $\mathcal{ALC}$-variant of $\mathbf{S5}_{\mathcal{EL}}$ is 2EXPTIME-complete [10], it was thus tempting to conjecture that the same holds for $\mathbf{S5}_{\mathcal{EL}}$. We show that this is not the case by establishing a tight PSPACE upper bound for subsumption in $\mathbf{S5}_{\mathcal{EL}}$. In particular, we devise a two-dimensional variant of completion algorithms [12]. We further show that this result does not hold anymore if inverse roles are allowed, that is, $\mathcal{ELI}$ is considered instead of $\mathcal{EL}$. Particularly, we establish a tight 2EXPTIME lower bound using a reduction of the word problem of exponentially space bounded alternating Turing machines. Unfortunately, this implies that subsumption in $\mathbf{S5}_{\mathcal{ELI}}$ is as hard as in the corresponding $\mathcal{ALCQI}$-variant.

Organization: The next section formally introduces DLs of change. Further, Section, 4.3 investigates the computational complexity of $\mathbf{S5}_{\mathcal{ALCO}}$ in the case where temporal operators are applied to concepts and a global TBox is considered. In Section 4.4, we investigate DLChs based on $\mathcal{EL}$ and its extension $\mathcal{ELI}$ in the case where temporal operators are applied not only to concepts but also to roles, and a global TBox is considered. This chapter ends with some conclusions presented in Section 4.5.

4.2 Introducing Description Logics of Change

Description Logics of Change (DLCh) emerge from the combination of the modal logic $\mathbf{S5}$ with classical DLs in the style of multi-dimensional DLs. DLs of change enable reasoning about changes of knowledge over time. However, they do not permit to differentiate between changes occurring in the past or in the future. In the investigation carried out in this chapter, we concentrate on the study of DLChs based on the DLs $\mathcal{EL}$, $\mathcal{ELI}$ and $\mathcal{ALCO}$.

4.2.1 Syntax and Semantics

Definition 4.1. *Fix countably infinite disjoint sets* $\mathsf{N_C}, \mathsf{N_R}, \mathsf{N_I}$ *of* concept names, role names *and* individual names, *respectively.* $\mathbf{S5}_{\mathcal{ALCO}}$ concepts *are formed by the following grammar:*

$$C ::= \top \mid A \mid \neg C \mid C \sqcap D \mid \{a\} \mid \exists r.C \mid \Diamond D \mid \exists {*}r.C$$

where A ranges over $\mathsf{N_C}$*, a ranges over* $\mathsf{N_I}$*, r ranges over* $\mathsf{N_R}$*, and* $* \in \{\Diamond, \Box\}$.

We say that a role is *temporal* if it is of the form $\Diamond r$ or $\Box r$, where r ranges over $\mathsf{N_R}$. We use standard Boolean abbreviations, plus $\Box C$ to abbreviate $\neg\Diamond\neg C$.

We define $\mathbf{S5}_{\mathcal{ALCO}}$ TBoxes as for classical DLs but using concepts from $\mathbf{S5}_{\mathcal{ALCO}}$.

Definition 4.2. *An* $\mathbf{S5}_{\mathcal{ALCO}}$ TBox *is a finite set of CIs* $C \sqsubseteq D$*, where* C, D *are* $\mathbf{S5}_{\mathcal{ALCO}}$ *concepts.*

For example, the following $\mathbf{S5}_{\mathcal{ALCO}}$ CI states that Pluton is an igneous rock and that it is a dwarf planet in an astronomy version. Moreover, we ensure that Pluton cannot be an igneous rock and a dwarf planet in the same version.

$$\mathsf{Pluton} \sqsubseteq \mathsf{IgneousRock} \sqcap \Diamond(\mathsf{DwarfPlanet} \sqcap \exists \mathsf{inVersion}.\{\mathsf{astronomy}\})$$
$$\mathsf{DwarfPlanet} \sqcap \mathsf{IgneousRock} \sqsubseteq \bot$$

The possible world semantics of DLs of change is given in terms of *temporal interpretations*, which associate with each possible world w a classical DL interpretation $\mathcal{I}_w$.

Definition 4.3. *A* temporal interpretation $\mathfrak{I}$ *is a structure* $(\Delta, W, \{\mathcal{I}_w\}_{w\in W})$ *where* W *is a non-empty set of* possible worlds *and for each* $w \in W$*,* $\mathcal{I}_w$ *is a classical DL interpretation with domain* Δ*, such that* $a^{\mathcal{I}_w} = a^{\mathcal{I}_{w'}}$ *for all* $a \in \mathsf{N_I}$ *and* $w, w' \in W$*. The mapping* $\cdot^{\mathfrak{I},w}$ *is extended to complex concepts and roles as follows:*

$$\begin{aligned}
\top^{\mathfrak{I},w} &= \Delta;\\
(\neg C)^{\mathfrak{I},w} &= \{d \in \Delta \mid d \notin C^{\mathfrak{I},w}\};\\
(C \sqcap D)^{\mathfrak{I},w} &= \{d \in \Delta \mid d \in C^{\mathfrak{I},w} \wedge d \in D^{\mathfrak{I},w}\};\\
(\exists r.C)^{\mathfrak{I},w} &= \{d \in \Delta \mid \exists e \in \Delta :\ e \in C^{\mathfrak{I},w} \wedge (d,e) \in r^{\mathfrak{I},w}\};\\
(\{a\})^{\mathfrak{I},w} &= \{a^{\mathfrak{I},w}\};\\
(\Diamond C)^{\mathfrak{I},w} &= \{d \in \Delta \mid \exists v \in W :\ d \in C^{\mathfrak{I},v}\};\\
(\Diamond r)^{\mathfrak{I},w} &= \{(d,e) \in \Delta\times\Delta \mid \exists v \in W : (d,e) \in r^{\mathfrak{I},v}\};\\
(\Box r)^{\mathfrak{I},w} &= \{(d,e) \in \Delta\times\Delta \mid \forall v \in W : (d,e) \in r^{\mathfrak{I},v}\};\\
(\exists {*}r.C)^{\mathfrak{I},w} &= \{d \in \Delta \mid \exists e \in \Delta :\ e \in C^{\mathfrak{I},w} \wedge (d,e) \in ({*}r)^{\mathfrak{I},w}\}.
\end{aligned}$$

We usually write $C^{\mathfrak{I},w}$ instead of $C^{\mathcal{I}_w}$ –analogously for $r^{\mathcal{I}_w}$–. Intuitively, $d \in C^{\mathfrak{I},w}$ means that in the temporal interpretation $\mathfrak{I}$, d is an instance of C in the world w. Moreover, note that in the previous definition we make the *constant domain assumption*, that is, each world shares the same domain Δ. Intuitively, this means that objects are not created or destroyed from one world to another. In this thesis, we are interested in the study of the complexity of the concept satisfiability problem w.r.t. TBoxes.

Definition 4.4. *A temporal interpretation* $\mathfrak{I}$ *is a* model of a concept C *if* $C^{\mathfrak{I},w} \neq \emptyset$ *for some* $w \in W$*; it is a* model of a TBox $\mathcal{T}$ *if* $C^{\mathfrak{I},w} \subseteq D^{\mathfrak{I},w}$ *for all* $w \in W$ *and* $C \sqsubseteq D$ *in* $\mathcal{T}$*. A concept* C is satisfiable w.r.t. *a TBox* $\mathcal{T}$ *if there exists a common model of* C *and* $\mathcal{T}$.

Note that in the previous definition TBoxes are *globally* interpreted in the sense that the axioms should hold in each world.

4.3 Reasoning in $\mathrm{S5}_{\mathcal{ALCO}}$ without Temporal Roles

This section begins our investigation on the computational complexity of DLs of change. We start by considering a restricted variant of $\mathbf{S5}_{\mathcal{ALCO}}$ in which concepts of the form $\exists{*}r.C$ are disallowed, that is, we consider only temporal concepts $\Diamond C$ and *local* roles. More precisely, we allow for the concepts defined by the following grammar:

$$C ::= \top \mid A \mid \neg C \mid C \sqcap D \mid \{a\} \mid \exists r.C \mid \Diamond D$$

where A ranges over $\mathsf{N_C}$, a ranges over $\mathsf{N_I}$, r ranges over $\mathsf{N_R}$. For example, the following CI is formulated in this logic:

$$\mathsf{Turtle} \sqsubseteq \mathsf{Reptile} \sqcap \Diamond(\mathsf{City} \sqcap \exists \mathsf{located}.(\mathsf{Canada} \sqcup \mathsf{US}) \sqcap \exists \mathsf{inVersion}.\{\mathsf{geography}\}).$$

Intuitively, this axiom states that a turtle is a reptile and that it is a city located in Canada or US in a geography version.

Recall that we are trying to identify logics computationally less complex than $\mathbf{S5}_{\mathcal{ALCQI}}$. Since in the presence of temporal roles already the combination based on $\mathcal{ALC}$ becomes 2EXPTIME-hard, we look at the case where only local roles are allowed. Note that reasoning in $\mathbf{S5}_{\mathcal{ALC}}$ with only temporal concepts in the presence of a global TBox is EXPTIME-complete [38] and then not harder than in $\mathcal{ALC}$. In this chapter, we show that this is not the case for $\mathbf{S5}_{\mathcal{ALCO}}$ in which reasoning becomes NEXPTIME-hard and then harder than in $\mathcal{ALCO}$.

We next show that the presence of nominals indeed makes reasoning harder. In particular, the interaction of nominals with $\mathbf{S5}$-modalities enables a reduction of the NEXPTIME-complete $2^n \times 2^n$-tiling problem. This shows thus that there is a jump in the complexity from EXPTIME-complete for $\mathcal{ALCO}$ to NEXPTIME-hard for $\mathbf{S5}_{\mathcal{ALCO}}$.

Before presenting the reduction, we proceed to introduce the $2^n \times 2^n$-tiling problem which is a *bounded* version of the undecidable $\mathbb{N} \times \mathbb{N}$-tiling problem [23, 76].

A *tile type* t is a 4-tuple of colors $(\mathsf{left}(t), \mathsf{right}(t), \mathsf{up}(t), \mathsf{down}(t))$. An *instance* $\mathfrak{T}$ is a tuple (T, t_0, n), where T is a finite set of type tiles, $t_0 \in T$ and $n \in \mathbb{N}$ is given in unary. We next define the matching conditions under which a set of tile types tiles the $2^n \times 2^n$ grid. Given an instance $\mathfrak{T} = (T, t_0, n)$, we say that T *tiles* the $2^n \times 2^n$ grid if there exists a function τ from the set $\{(i, j) \mid i, j < 2^n\}$ to T such that the following hold:

– $\mathsf{up}(\tau(i, j)) = \mathsf{down}(\tau(i, j + 1))$, for all $i < 2^n, j < 2^n - 1$.

– $\mathsf{right}(\tau(i,j)) = \mathsf{left}(\tau(i+1,j))$, for all $i < 2^n - 1, j < 2^n$.

– $\tau(0,0) = t_0$.

The following $2^n \times 2^n$*-tiling problem* is NEXPTIME-complete [23, 76]: Given an instance $\mathfrak{T} = (T, t_0, n)$, does T tile the $2^n \times 2^n$ grid?

Theorem 4.1. *Concept satisfiability w.r.t. TBoxes for* $\mathbf{S5}_{\mathcal{ALCO}}$ *without temporal roles is* NEXPTIME*-hard.*

Proof. The proof is by a polynomial reduction of the $2^n \times 2^n$-tiling problem. Let $\mathfrak{T} = (T, t_0, n)$ be an instance. Our aim is to construct in polynomial time a TBox $\mathcal{T}_{\mathfrak{T}}$ and a concept $C_{\mathfrak{T}}$, such that T tiles the $2^n \times 2^n$ grid *iff* $C_{\mathfrak{T}}$ is satisfiable w.r.t. $\mathcal{T}_{\mathfrak{T}}$. We use the following signature:

– concept names A_{τ_i} for each $\tau_i \in T$;

– concept names $X_0, \ldots, X_{2n-1}$ and $Y_0, \ldots Y_{2n-1}$ for encoding exponential counters;

– a single nominal $\{a\}$;

– an auxiliary concepts Grid, $\mathsf{StartGrid}$, $\mathsf{EndGrid}$, $\mathsf{RightEdge}$, $\mathsf{DownNeighbor}$.

First, the inclusions (4.1)-(4.6) enforce a 2^{2n}-long chain of individuals (Grid), uniquely identifiable by counting concepts X_i and Y_i, for $i \in (0, 2n-1)$. Notably, the Y-counter is shifted in the phase w.r.t. the X-counter by exactly 2^n, (i.e., $X + 2^{n-1} = Y$), which further on is used for identifying the top-down neighbors in the tiling. Also, every 2^{n-1}-th individual starting from the beginning of the chain is made an instance of concept $\mathsf{RightEdge}$, marking the right edge of the tiling (4.2):

$$\mathsf{StartGrid} \equiv \mathsf{Grid} \sqcap \bigsqcap_{j=0}^{2n-1} \neg X_j \sqcap \bigsqcap_{j=0}^{n-1} \neg Y_j \sqcap Y_n \sqcap \bigsqcap_{j=n+1}^{2n-1} \neg Y_j, \tag{4.1}$$

$$\mathsf{EndGrid} \equiv \bigsqcap_{j=0}^{2n-1} X_j \qquad \mathsf{Grid} \sqcap \neg\mathsf{EndGrid} \sqsubseteq \exists s.\mathsf{Grid}, \qquad \mathsf{RightEdge} \equiv \bigsqcap_{j=0}^{n-1} X_j. \tag{4.2}$$

For every $0 \leq j < i < 2n$:

$$\begin{aligned} \neg X_i \sqcap \neg X_j &\sqsubseteq \forall s.\neg X_i, \\ X_i \sqcap \neg X_j &\sqsubseteq \forall s.X_i. \end{aligned} \tag{4.3}$$

For every $0 \leq j < 2n$:

$$\begin{aligned} \neg X_j \sqcap X_{j-1} \sqcap \ldots \sqcap X_1 &\sqsubseteq \forall s.X_j, \\ X_j \sqcap X_{j-1} \sqcap \ldots \sqcap X_1 &\sqsubseteq \forall s.\neg X_j. \end{aligned} \tag{4.4}$$

For every $0 \leq j < i < 2n$:

$$\begin{aligned} \neg Y_i \sqcap \neg Y_j &\sqsubseteq \forall s.\neg Y_i, \\ Y_i \sqcap \neg Y_j &\sqsubseteq \forall s.Y_i. \end{aligned} \tag{4.5}$$

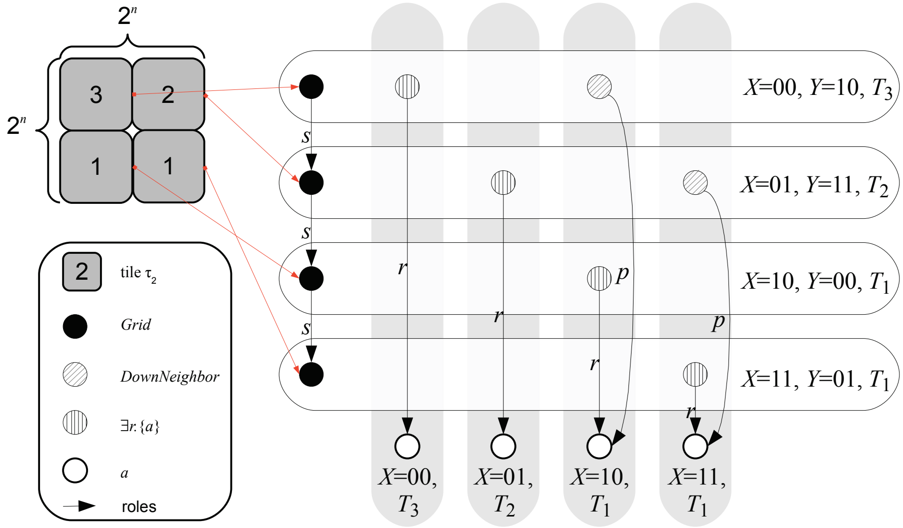

Figure 4.1: Encoding of a $2^n \times 2^n$-tiling in an $\mathbf{S5}_{\mathcal{ALCO}}$-model.

For every $0 \leq j < 2n$:

$$\begin{aligned} \neg Y_j \sqcap Y_{j-1} \sqcap \ldots \sqcap Y_1 &\sqsubseteq \forall s.Y_j, \\ Y_j \sqcap Y_{j-1} \sqcap \ldots \sqcap Y_1 &\sqsubseteq \forall s.\neg Y_j. \end{aligned} \tag{4.6}$$

Next, by means of CIs (4.7)-(4.8), the values of the counting concepts are propagated globally across all $\mathbf{S5}$-worlds.

For every $0 \leq i < 2n$:

$$X_i \sqsubseteq \Box X_i, \quad \neg X_i \sqsubseteq \Box \neg X_i, \tag{4.7}$$

$$Y_i \sqsubseteq \Box Y_i, \quad \neg Y_i \sqsubseteq \Box \neg Y_i. \tag{4.8}$$

Further, we impose the basic coloring constraints over all individuals (4.9). We adjust the coloring of all the left-right neighbors: (4.10). Finally, we propagate the tile types over all $\mathbf{S5}$-worlds (4.11):

For every $\tau_i, \tau_j \in T$,

$$\top \sqsubseteq (\bigsqcup_{\tau_i} A_{\tau_i}) \sqcap \bigsqcap_{\tau_i \neq \tau_j} \neg(A_{\tau_i} \sqcap A_{\tau_j}), \tag{4.9}$$

$$A_{\tau_i} \sqcap \neg \mathsf{RightEdge} \sqsubseteq \forall s.(\bigsqcup_{\mathsf{right}(\tau_i)=\mathsf{left}(\tau_j)} A_{\tau_j}), \tag{4.10}$$

$$A_{\tau_i} \sqsubseteq \Box A_{\tau_i}. \tag{4.11}$$

The key to the reduction is a suitable use of a single nominal $\{a\}$ (see Figure 4.1). By (4.12) every individual in the grid is linked to a via a role r in some $\mathbf{S5}$-world. There, due to (4.13)-(4.14), the value of the X-counter and the tile type assigned to the individual is forced upon a. Consequently, by assuming rigid individual names, we generate 2^{2n} distinct $\mathbf{S5}$-worlds:

$$\mathsf{Grid} \sqsubseteq \Diamond\exists r.\{a\}, \tag{4.12}$$

$$X_i \sqsubseteq \forall r.X_i, \quad \neg X_i \sqsubseteq \forall r.\neg X_i, \tag{4.13}$$

$$A_{\tau_i} \sqsubseteq \forall r.A_{\tau_i}. \tag{4.14}$$

Finally, in every **S5**-world, all individuals are linked to a via p (4.15). Whenever the value of the Y-counter on a grid-individual matches the value of the X-counter on a (4.16), the proper top-down coloring constraints are imposed (4.17):

$$\top \sqsubseteq \exists p.\{a\}, \tag{4.15}$$

$$\mathsf{DownNeighbor} \equiv \prod_{j=0}^{2n-1} ((Y_i \sqcap \exists p.X_i) \sqcup (\neg Y_i \sqcap \exists p.\neg X_i)), \tag{4.16}$$

$$A_{\tau_i} \sqcap \mathsf{DownNeighbor} \sqsubseteq \forall p. \bigsqcup_{\mathsf{down}(\tau_i)=\mathsf{up}(\tau_j)} A_{\tau_j}, \text{ for every } \tau_i, \tau_j \in T. \tag{4.17}$$

The TBox $\mathcal{T}_{\mathfrak{T}}$ is defined as the union of the axioms (4.1)-(4.17). It is easy to see that the size of $\mathcal{T}_{\mathfrak{T}}$ is polynomial in the size of the instance $\mathfrak{T}$. Finally, we define the concept

$$C_{\mathfrak{T}} = \mathsf{StartGrid} \sqcap A_{\tau_0}.$$

Now, following the construction of $\mathcal{T}_{\mathfrak{T}}$ it is not difficult to see that the following proposition holds.

Proposition 4.2. *$\mathfrak{T}$ tiles the $2^n \times 2^n$-grid* iff *$C_{\mathfrak{T}}$ is satisfiable w.r.t. $\mathcal{T}_{\mathfrak{T}}$.*

Let τ be a tiling for $\mathfrak{T}$, that is, a mapping from $2^n \times 2^n$ to T. We define a model $\mathfrak{I} = (\Delta, W, \{\mathcal{I}_w\}_{w \in W})$ of $\mathcal{T}_{\mathfrak{T}}$ and $C_{\mathfrak{T}}$ as follows. First, transform τ into $\pi : 2^{2n} \to T$, such that for every $(x, y) \in 2^n \times 2^n$, $\tau(x, y) = \pi(y * 2^n + x)$. Now, set $W = \{w_i \mid i \in (0, 2^{2n})\}$ and $\Delta = \{d_i \mid i \in (0, 2^{2n})\}$ and ensure that the following interpretation constraints are satisfied:

- $a^{\mathfrak{I},w} = d_0$ for $d_0 \in \Delta$ and every $w \in W$,
- for $w_0 \in W$:
  - $\mathsf{Grid}^{\mathfrak{I},w_0} = \Delta \backslash \{d_0\}$,
  - $\mathsf{StartGrid}^{\mathfrak{I},w_0} = \{d_1 \in \Delta\}$, $\mathsf{EndGrid}^{\mathfrak{I},w_0} = \{d_{2^{2n}} \in \Delta\}$,
  - $\mathsf{RightEdge}^{\mathfrak{I},w_0} = \{d_{2^n * i} \in \Delta\}$,
  - $s^{\mathfrak{I},w_0} = \{(d_i, d_{i+1}) \mid d_i, d_{i+1} \in \Delta, i \geq 1\}$,
- $\{d_i \mid \pi(i) = \tau_j\} \subseteq A_{\tau_j}^{\mathfrak{I},w}$, for every $w \in W$ and $\tau_j \in T$,
- $d_0 \in A_{\tau_j}^{\mathfrak{I},w_i}$ *iff* $\pi(i) = \tau_j$, for every $i \geq 1$ and $\tau_j \in T$,
  - $r^{\mathfrak{I},w_i} = \{(d_i, d_0) \mid d_i \in \Delta\}$ for $i \geq 1$,
  - $p^{\mathfrak{I},w} = \{(d, d_0) \mid d \in \Delta\}$ for every $w \in W$,
  - $\mathsf{DownNeighbor}^{\mathfrak{I},w_i} = \{d_{i-2^n} \in \Delta\}$, for every $w_i \in W$ and $i \geq 2^n + 1$.

The interpretation can be standardly extended to counting concepts X_i and Y_i so that $\mathfrak{I}$ is indeed a model of $\mathcal{T}_{\mathfrak{T}}$ with $d_1 \in (C_{\mathfrak{T}})^{\mathfrak{I},w_0}$.

Conversely, let $\mathfrak{I}$ be a model of $\mathcal{T}_{\mathfrak{T}}$ and $C_{\mathfrak{T}}$. A tiling for $\mathfrak{T}$ can be retrieved from $\mathfrak{I}$ by mapping a chain of s-successors, which instantiate the concept Grid in the $\mathbf{S5}$-world in which $C_{\mathfrak{T}}$ is satisfied, on the $2^n \times 2^n$ grid, where the type of a tile in the grid is determined by the unique concept A_{τ_i} satisfied by the individual in the chain. The coloring constraints have to be satisfied by the construction of the encoding.

❑

We obtain a matching upper bound in Chapter 5, Section 5.8.

Theorem 4.3. *Concept satisfiability w.r.t. TBoxes for $\mathbf{S5}_{\mathcal{ALCO}}$ without temporal roles is* NEXPTIME-*complete.*

4.4 Reasoning in $\mathrm{S5}_{\mathcal{EL}}$ and $\mathrm{S5}_{\mathcal{ELI}}$ with Temporal Roles

We continue our investigation by studying the computational complexity of DLs of change in which $\mathbf{S5}$-modalities are applied to both concepts and roles. With the aim of designing computationally well-behaved logics, we consider DLChs based on the tractable DL $\mathcal{EL}$ and its extension $\mathcal{ELI}$. Notably, we show that by using $\mathcal{EL}$ instead of $\mathcal{ALC}$ reasoning becomes indeed easier: it goes down from 2EXPTIME-complete [10] for $\mathbf{S5}_{\mathcal{ALC}}$ to PSPACE-complete for $\mathbf{S5}_{\mathcal{EL}}$. It is worth noting that $\mathbf{S5}_{\mathcal{EL}}$ is the first two-dimensional extension of $\mathcal{EL}$ *with* temporal roles presenting better computational complexity than the corresponding $\mathcal{ALC}$ variant. Unfortunately, later on, we show that this result does not hold anymore if we use $\mathcal{ELI}$ instead of $\mathcal{EL}$. In particular, we show that reasoning in $\mathbf{S5}_{\mathcal{ELI}}$ is as difficult as in $\mathbf{S5}_{\mathcal{ALC}}$.

$\mathbf{S5}_{\mathcal{EL}}$ is the fragment of $\mathbf{S5}_{\mathcal{ALCO}}$ (with temporal roles) that disallows $\neg$, and thus $C \sqcup D$, $\forall r.C$, and nominals. Formally, $\mathbf{S5}_{\mathcal{EL}}$ *concepts* are formed by the following grammar:

$$C ::= \top \mid A \mid C \sqcap D \mid \exists r.C \mid *C \mid \exists * r.C$$

where A ranges over $\mathsf{N_C}$, r ranges over $\mathsf{N_R}$, and $* \in \{\Diamond, \Box\}$. $\mathbf{S5}_{\mathcal{ELI}}$ further allows for inverse roles, that is, r ranges over $\{r, r^- \mid r \in \mathsf{N_R}\}$.

For example, the following CI is formulated in $\mathbf{S5}_{\mathcal{EL}}$:

$$\mathsf{ViralDisease} \sqsubseteq \mathsf{Disease} \sqcap \exists \Diamond \mathsf{hasCause}.\mathsf{Virus}$$

Intuitively, this axiom states that a viral disease is a disease that has as a possible cause a virus. As in the case of classical $\mathcal{EL}$, due to the lack of negation the satisfiability problem becomes trivial for $\mathbf{S5}_{\mathcal{EL}}$ in the sense that every concept is satisfiable w.r.t. every TBox. We concentrate thus on the *subsumption* problem for $\mathbf{S5}_{\mathcal{EL}}$: a concept D *subsumes* a concept C *w.r.t. a* $\mathbf{S5}_{\mathcal{EL}}$ *TBox* $\mathcal{T}$, if $C^{\mathfrak{I}} \subseteq D^{\mathfrak{I}}$ for every temporal interpretation $\mathfrak{I}$ that is a model of $\mathcal{T}$.

We devote the rest of this section to the development of a *completion algorithm* for deciding subsumption w.r.t. TBoxes for $\mathbf{S5}_{\mathcal{EL}}$ with temporal roles, yielding a tight PSPACE upper bound. Our algorithm can be seen as a 'two-dimensional' variant of completion algorithms for Horn

DLs [12, 52]. The lower bound was established in the context of the research on *probabilistic DLs* with possible world semantics [57] [Theorem 14]. A discussion on the relation of probabilistic DLs based on $\mathcal{EL}$ and $\mathbf{S5}_{\mathcal{EL}}$ can be found in [44].

4.4.1 An Algorithm for Concept Subsumption w.r.t. TBoxes for $\mathrm{S5}_{\mathcal{EL}}$ with Temporal Roles

We concentrate w.l.o.g. on subsumption between concept *names* and assume that the input TBox is in a certain normal form, defined as follows. A *basic concept* is a concept of the form $\top$, A, $\Diamond A$, $\Box A$, or $\exists\alpha.A$, where A is a concept name and, here and in what follows, α is a *role*, i.e., of the form r, $\Diamond r$, or $\Box r$ with r a role name. Now, every concept inclusion in the input TBox is required to be of the form

$$X_1 \sqcap \cdots \sqcap X_n \sqsubseteq X$$

with $X_1, \ldots, X_n, X$ basic concepts. Every TBox in $\mathbf{S5}_{\mathcal{EL}}$ can be transformed into this normal form in polynomial time such that (non-)subsumption between the concept names that occur in the original TBox is preserved [57].

Let $\mathcal{T}$ be the input TBox in normal form, CN the set of concept names that occur in $\mathcal{T}$, BC the set of basic concepts in $\mathcal{T}$, and ROL the set of roles in $\mathcal{T}$. Our algorithm maintains the data structures shown in Figure 4.2, which will be saturated according to a set of completion rules. The definition of the data structures provides already some intuition about their meaning, e.g., $X \in \mathsf{Q}(A)$ means that $\mathcal{T} \models A \sqsubseteq X$. The key characteristic, however, of these structures is that they provide an abstract representation of a model of $\mathcal{T}$:

- $\mathsf{Q(A)}$ describes the concept memberships of a domain element d in a world w with $d \in A^{\mathfrak{I},w}$;
- R describes role memberships, that is, when $(A, B) \in \mathsf{R}(\alpha)$, then $d \in A^{\mathfrak{I},w}$ implies that in some world v, d has an element described by $\mathsf{Q}(B)$ as an α-successor;
- $\mathsf{Q}_{\mathsf{cert}}(A)$ contains all concepts that must be true in *all* worlds for any domain element that satisfies A in *some* world.

The data structures are initialized as follows, for all $A \in \mathsf{CN}$ and relevant roles α:

$$\mathsf{Q}(A) = \{\top, A\} \qquad \mathsf{Q}_{\mathsf{cert}}(A) = \{\top\} \qquad \mathsf{R}(\alpha) = \emptyset$$

The sets $\mathsf{Q}(\cdot)$, $\mathsf{Q}_{\mathsf{cert}}(\cdot)$, and $\mathsf{R}(\cdot)$ are then repeatedly extended by the application of various rules. First, a set of 'local rules' is presented in Figure 4.3, serving the purpose of saturating a set of concepts Γ. These rules are close in spirit to those introduced by Baader *el al.* [12] for classical $\mathcal{EL}$; they saturate a set describing an element in a given world. We use $\mathsf{cl}(\Gamma)$ to denote the set of concepts that results from exhaustively applying the rules in Figure 4.3 to Γ, where any rule can only be applied if the added concept is in BC, but not yet in Γ. The rules will be applied to

Mapping Q that associates with each $A \in \mathsf{CN}$ a subset $\mathsf{Q}(A) \subseteq \mathsf{BC}$ such that $\mathcal{T} \models A \sqsubseteq X$ for all $X \in \mathsf{Q}(A)$
Mapping $\mathsf{Q}_{\mathsf{cert}}$ that associates with each $A \in \mathsf{CN}$ a subset $\mathsf{Q}_{\mathsf{cert}}(A) \subseteq \mathsf{BC}$ such that $\mathcal{T} \models A \sqsubseteq \Box X$ for all $X \in \mathsf{Q}_{\mathsf{cert}}(A)$
Mapping R that associates with each probabilistic role $\alpha \in \mathsf{ROL}$ a binary relation $\mathsf{R}(\alpha)$ on CN such that $\mathcal{T} \models A \sqsubseteq \Diamond(\exists\alpha.B)$ for all $(A, B) \in \mathsf{R}(\alpha)$

Figure 4.2: Data Structures Q, $\mathsf{Q}_{\mathsf{cert}}$, R

the sets $\mathsf{Q}(A)$ and $\mathsf{Q}_{\mathsf{cert}}(A)$, but they will also serve other purposes as described below. It is not hard to see that rule application terminates after polynomially many steps.

The rules that are used for completing the data structures $\mathsf{Q}(\cdot)$, $\mathsf{Q}_{\mathsf{cert}}(\cdot)$, and $\mathsf{R}(\cdot)$ are more complex rules that take into account the two-dimensional nature of $\mathbf{S5}_{\mathcal{EL}}$. In particular, they refer to 'traces' through these data structures, capturing the interpretation of the domain in a possible world.

Definition 4.5. *Let* $B \in \mathsf{CN}$. *A* trace to B *is a finite sequence* $S, A_1, \alpha_2, A_2, \ldots, \alpha_n, A_n$ *where*

– $S = A$ *for some* $\Diamond A \in \mathsf{Q}(A_1)$ *or* $S = (r, B_1)$ *for some* $(A_1, B_1) \in \mathsf{R}(\Diamond r)$;

– *each* $A_i \in \mathsf{CN}$ *and each* $\alpha_i \in \mathsf{ROL}$ *is a temporal role, such that* $A_n = B$;

– $(A_i, A_{i-1}) \in \mathsf{R}(\alpha_i)$ *for* $1 < i \leq n$.

If t is a trace of length n, we use t_k, $k \leq n$, to denote the shorter trace $S, A_1, \alpha_2, \ldots, \alpha_k, A_k$.

Intuitively, the purpose of a trace is to deal with worlds that are generated by concepts $\Diamond A$ and $\exists\Diamond r.A$. Note that there can be infinitely many such worlds as $\mathbf{S5}_{\mathcal{EL}}$ lacks the finite model property [57]. The trace starts at some domain element represented by a set $\mathsf{Q}(A_1)$ in the world generated by the first element S of the trace, then repeatedly follows role edges represented by $\mathsf{R}(\cdot)$ backwards until it reaches the final domain element represented by $\mathsf{Q}(B)$. The importance of traces stems from the fact that information can be propagated along them, as captured by the following notion.

R1	If $X_1 \sqcap \ldots \sqcap X_n \sqsubseteq X \in \mathcal{T}$ and $X_1, \ldots, X_n \in \Gamma$ then add X to Γ
R2	If $\Box A \in \Gamma$ then add A to Γ
R3	If $\exists \Box r.A \in \Gamma$ then add $\exists r.A$ to Γ
R4	If $A \in \Gamma$ then add $\Diamond A$ to Γ
R5	If $\exists r.A \in \Gamma$ then add $\exists \Diamond r.A$ to Γ
R6	If $\exists \alpha.A \in \Gamma$ and $B \in \mathsf{Q}(A)$ then add $\exists \alpha.B$ to Γ

Figure 4.3: Saturation rules for $\mathsf{cl}(\Gamma)$

Definition 4.6. *Let $t = S, A_1, \alpha_2, \ldots, \alpha_n, A_n$ be a trace of length n. Then the* type $\Gamma(t) \subseteq \mathsf{BC}$ *of t is defined as follows:*

- $\mathsf{cl}(\{A\} \cup \mathsf{Q}_{\mathsf{cert}}(A_1))$ *if $n = 1$ and $S = A$;*
- $\mathsf{cl}(\mathsf{Q}_{\mathsf{cert}}(A_1) \cup \{\exists r.B' \in \mathsf{BC} \mid B' \in \mathsf{Q}_{\mathsf{cert}}(B_1)\})$ *if $n = 1$ and $S = (r, B_1)$;*
- $\mathsf{cl}(\mathsf{Q}_{\mathsf{cert}}(A_n) \cup \{\exists \alpha_n.B' \in \mathsf{BC} \mid B' \in \Gamma(t_{n-1})\})$ *if $n > 1$.*

The propagation of information along traces is now as follows: if there is a trace t to B, then any domain element that satisfies B in *some* world must satisfy the concepts in $\Gamma(t)$ in some other world. So if for example $\Diamond A \in \Gamma(t)$, we need to add $\Diamond A$ also to $\mathsf{Q}_{\mathsf{cert}}(B)$ and to $\mathsf{Q}(B)$.

Figure 4.4 shows the rules used for completing the data structures $\mathsf{Q}(\cdot)$, $\mathsf{Q}_{\mathsf{cert}}(\cdot)$, and $\mathsf{R}(\cdot)$, where rules **S6** and **S7** implement, using traces, the propagation of information across both dimensions.

Our algorithm for deciding subsumption starts with the initial data structures defined above and then exhaustively applies the rules shown in Figure 4.4. To decide whether $\mathcal{T} \models A \sqsubseteq B$, it then simply checks whether $B \in \mathsf{Q}(A)$.

Lemma 4.4. *Let $\mathcal{T}$ be a $\mathbf{S5}_{\mathcal{EL}}$-TBox in normal form and A, B be concept names. Then $\mathcal{T} \models A \sqsubseteq B$ iff, after exhaustive rule application, $B \in Q(A)$.*

Proof. For the “if” direction we show that the following invariants of the algorithm hold, i.e.,

$$C \in \mathsf{Q}(A) \text{ implies } A \sqsubseteq_{\mathcal{T}} C \tag{I1}$$
$$C \in \mathsf{Q}_{\mathsf{cert}}(A) \text{ implies } A \sqsubseteq_{\mathcal{T}} \Box C \tag{I2}$$
$$(A, B) \in \mathsf{R}(\alpha) \text{ implies } A \sqsubseteq_{\mathcal{T}} \Diamond(\exists \alpha.B) \tag{I3}$$

S1	apply **R1-R6** to $\mathsf{Q}(A)$ and $\mathsf{Q}_{\mathsf{cert}}(A)$
S2	if $*B \in \mathsf{Q}(A)$ then add $*B$ to $\mathsf{Q}_{\mathsf{cert}}(A)$
S3	if $C \in \mathsf{Q}_{\mathsf{cert}}(A)$ then add $\Box C$ and C to $\mathsf{Q}(A)$
S4	If $\exists\alpha.B \in \mathsf{Q}(A)$ with α a temporal role then add (A, B) to $\mathsf{R}(\alpha)$.
S5	If $\Diamond B_1 \in \mathsf{Q}(A)$, $(B_1, B_2) \in \mathsf{R}(\alpha)$, $B_3 \in \mathsf{Q}_{\mathsf{cert}}(B_2)$ then add $\exists\alpha.B_3$ to $\mathsf{Q}_{\mathsf{cert}}(A)$
S6	if t is a trace to B and $*A \in \Gamma(t)$ then add $*A$ to $\mathsf{Q}_{\mathsf{cert}}(B)$
S7	if t is a trace to B and $\exists\alpha.A \in \Gamma(t)$ with α a temporal role then add (B, A) to $\mathsf{R}(\alpha)$

Figure 4.4: The rules for completing the data structures.

The proof is by induction on the number of applications of the rules in Figure 4.4. The induction base is trivial since $A \sqsubseteq_{\mathcal{T}} A$ and $A \sqsubseteq_{\mathcal{T}} \top$. For the induction step we start with showing soundness of the rules **R1-R6**, i.e., for every set of concepts Γ it holds

$$\bigsqcap \Gamma \sqsubseteq_{\mathcal{T}} \bigsqcap \mathsf{cl}(\Gamma) \tag{*}$$

For the rules **R1-R5** it follows directly by the semantics. For **R6** assume $\exists\alpha.A \in \Gamma$ and $B \in \mathsf{Q}(A)$. Invariant (I1) implies $A \sqsubseteq_{\mathcal{T}} B$, which means that we can certainly add $\exists\alpha.B$ to Γ.

Next, we analyze traces a little closer and prove the following claim.

Claim. If t is a trace to B, then $B \sqsubseteq_{\mathcal{T}} \Diamond(\bigsqcap \Gamma(t))$.

Proof of the Claim. Let $t = S, A_1, \alpha_2, \ldots, \alpha_n, A_n$. The proof is by induction on the length n of t. For the induction base, let $n = 1$ and consider first the case that the trace starts with $S = A$, i.e., $\Diamond A \in \mathsf{Q}(A_1)$. From the invariants (I1) and (I2) follows that $A_1 \sqsubseteq_{\mathcal{T}} \Diamond(A \sqcap \bigsqcap \mathsf{Q}_{\mathsf{cert}}(A_1))$. Since $\Gamma(t) = \mathsf{cl}(\{A\} \cup \mathsf{Q}_{\mathsf{cert}}(A_1))$, by (*), we obtain $A_1 \sqsubseteq_{\mathcal{T}} \Diamond(\bigsqcap \Gamma(t))$.

Assume now that the trace starts with $S = (r, B)$, i.e., $(A_1, B) \in \mathsf{R}(\Diamond r)$. From the invariant (I2), we get that $A_1 \sqsubseteq_{\mathcal{T}} \Box(\bigsqcap \mathsf{Q}_{\mathsf{cert}}(A_1))$ and $B \sqsubseteq_{\mathcal{T}} \Box(\bigsqcap \mathsf{Q}_{\mathsf{cert}}(B))$. Further, (I3) implies that $A_1 \sqsubseteq_{\mathcal{T}} \Diamond(\exists \Diamond r.B)$. Thus, $A_1 \sqsubseteq_{\mathcal{T}} \Diamond(\exists r.\Diamond B)$. Overall, we obtain that

$$A_1 \sqsubseteq_{\mathcal{T}} \Diamond\left(\bigsqcap \mathsf{Q}_{\mathsf{cert}}(A_1) \sqcap \exists r.\bigsqcap \mathsf{Q}_{\mathsf{cert}}(B)\right)$$

Since $\Gamma(t) = \mathsf{cl}(\mathsf{Q}_{\mathsf{cert}}(A_1) \cup \{\exists r.B' \mid B' \in \mathsf{Q}_{\mathsf{cert}}(B)\})$ then, by (*), it follows

$$A_1 \sqsubseteq_{\mathcal{T}} \Diamond(\bigsqcap \Gamma(t))$$

this finishes the proof of the induction base.

For the induction step, let $n > 1$. By Definition 4.5, $(A_n, A_{n-1}) \in \mathsf{R}(\alpha_n)$. By (I3), we have $A_n \sqsubseteq_\mathcal{T} \Diamond(\exists\alpha_n.A_{n-1})$. Applying the induction hypothesis, we get

$$A_n \sqsubseteq_\mathcal{T} \Diamond\left(\exists\alpha_n.\Diamond(\bigsqcap\Gamma(t_{n-1}))\right)$$

Since $\exists\alpha_n.\Diamond C \sqsubseteq_\mathcal{T} \Diamond\exists\alpha_n.C$, then

$$A_n \sqsubseteq_\mathcal{T} \Diamond\left(\exists\alpha_n.\bigsqcap\Gamma(t_{n-1})\right)$$

On the other hand, (I2) implies $A_n \sqsubseteq_\mathcal{T} \Box\bigsqcap\mathsf{Q}_{\mathsf{cert}}(A_n)$. Hence, we obtain the following:

$$A_n \sqsubseteq_\mathcal{T} \Diamond\left(\bigsqcap\mathsf{Q}_{\mathsf{cert}}(A_n)\sqcap\exists\alpha_n.\bigsqcap\Gamma(t_{n-1})\right)$$

Since $\Gamma(t) = \mathsf{cl}(\mathsf{Q}_{\mathsf{cert}}(A_n)\cup\{\exists\alpha_n.B \mid B\in\Gamma(t_{n-1})\})$ then, by (*), we get:

$$A_n \sqsubseteq_\mathcal{T} \Diamond \bigsqcap_{C\in\Gamma(t)} C$$

This finishes the proof of the claim.

It remains to show that the rules in Figure 4.4 preserve the invariants:

S1 Direct consequence of (*).

S2 Direct by the semantics: $\Diamond B \sqsubseteq_\mathcal{T} \Box(\Diamond B)$ and $\Box B \sqsubseteq_\mathcal{T} \Box(\Box B)$.

S3 $C \in \mathsf{Q}_{\mathsf{cert}}(A)$ implies $A \sqsubseteq_\mathcal{T} \Box C$ by invariant (I2), hence also $A \sqsubseteq_\mathcal{T} C$.

S4 $\exists\alpha.B \in \mathsf{Q}(A)$ implies $A \sqsubseteq_\mathcal{T} \exists\alpha.B$ by invariant (I1), thus also $A \sqsubseteq_\mathcal{T} \Diamond(\exists\alpha.B)$.

S5 By (I1), we get $A \sqsubseteq \Diamond B_1$. Then, by invariant (I3), $B_1 \sqsubseteq_\mathcal{T} \Diamond\exists\alpha.B_2$ and, by invariant (I2), $B_2 \sqsubseteq_\mathcal{T} \Box B_3$. Combining these inclusions yields $A \sqsubseteq_\mathcal{T} \Diamond(\exists\alpha.\Box B_3)$. The semantics then implies $A \sqsubseteq_\mathcal{T} \Box(\exists\alpha.B_3)$.

S6 Let t be a trace to B and $\Gamma = \Gamma(t)$ its type. By the above claim $B \sqsubseteq_\mathcal{T} \Diamond C$ for every $C \in \Gamma$. Thus in particular $B \sqsubseteq_\mathcal{T} *A$, if $*A \in \Gamma$. Hence $B \sqsubseteq_\mathcal{T} \Box(*A)$, so $*A$ can be added to $\mathsf{Q}_{\mathsf{cert}}(B)$.

S7 Analogously to **S6**.

Assume now that $B \in Q(A)$. Invariant (I1) implies $A \sqsubseteq_\mathcal{T} B$ which finishes the proof of the "if"-direction.

For showing the "only if" direction, we provide a temporal model $\mathfrak{I} = (\Delta, W, \{\mathcal{I}_w\}_{w\in W})$ of $\mathcal{T}$, such that there is a world $w \in W$ and a domain element $d \in \Delta$ with $d \in A^{\mathfrak{I},w}$ but $d \notin B^{\mathfrak{I},w}$.

We define sequences $\Delta_0, \Delta_1, \ldots, W_0, W_1, \ldots$, and partial maps $\pi_1, \pi_2, \ldots$ with $\pi_i : \Delta_i \times W_i \to 2^{\mathsf{BC}}$. Our desired sets Δ and W are then obtained in the limit. The elements of the sets Δ_i are

---

C1 If $\exists\alpha.A \in \pi(\sigma, w)$ for some $\sigma \in \Delta_i$ and $w \in W_i$, then
- add $\sigma\cdot(\alpha, w, A)$ to Δ_i (if it does not yet exist);
- set $\pi_i(\sigma\cdot(\alpha, w, A), w) = Q(A)$ and $\pi_i(\sigma\cdot(\alpha, w, A), v) = Q_{\mathsf{cert}}(A)$ for all $v \in W \setminus \{w\}$.

C2 If $\Diamond B \in \pi(\sigma, w)$ for some $\sigma \in \Delta_i$ and $w \in W_i$, then
- add (σ, B) to W_i (if it does not yet exist);
- set $\pi_i(\sigma|_j, (\sigma, B)) = \Gamma_j(B, \sigma)$ for all $j \leq n$;
- set $\pi_i(\sigma'\cdot(\alpha, w, A), (\sigma, B)) = Q_{\mathsf{cert}}(A)$, for all $\sigma'\cdot(\alpha, w, A) \in \Delta_i$ that are not a prefix of σ.

C3 If $\sigma = (\alpha_1, w_1, A_1) \cdots (\alpha_n, w_n, A_n) \in \Delta_i$ and $(A_n, B) \in R(\Diamond r)$, then
- add (σ, r, B) to W_i (if it does not yet exist);
- set $\pi_i(\sigma|_j, (\sigma, r, B)) = \Gamma_j((r, B), \sigma)$ for all $j \leq n$;
- set $\pi_i(\sigma'\cdot(\alpha, w, A), (\sigma, r, B)) = Q_{\mathsf{cert}}(A)$, for all $\sigma'\cdot(\alpha, w, A) \in \Delta_i$ that are not a prefix of σ.

Figure 4.5: Rules for the induction step

sequences of triples (α, w, A) where α is a role, $w \in W_i$, and A is a concept name. For such a sequence σ, we use $\sigma|_j$ to denote the prefix of σ that consists of the first j triples.

It is possible to view the sequences in Δ as traces, in analogy to the traces from Definition 4.5. Assume $\sigma = (\alpha_1, w_1, A_1) \cdots (\alpha_n, w_n, A_n) \in \Delta_i$ for some $i \geq 0$ (this $\mathcal{I}_i$ is yet to be defined), and that S is either A for some $\Diamond A \in Q(A_n)$ or (r, B) for some $(A_n, B) \in R(\Diamond r)$. We have then the following: for $j \leq n$, the *type* $\Gamma_j(S, \sigma) \in 2^{\mathsf{BC}}$ is defined as follows:

- $\mathsf{cl}(\{A\} \cup \mathsf{Q}_{\mathsf{cert}}(A_n))$ if $j = n$ and $S = A$;
- $\mathsf{cl}(\mathsf{Q}_{\mathsf{cert}}(A_n) \cup \{\exists r.B' \in \mathsf{BC} \mid B' \in \mathsf{Q}_{\mathsf{cert}}(B)\})$ if $j = n$ and $S = (r, B)$;
- $\mathsf{cl}(\mathsf{Q}_{\mathsf{cert}}(A_j) \cup \{\exists\hat{\alpha}_{j+1}.B' \in \mathsf{BC} \mid B' \in \Gamma_{j+1}(S, \sigma)\})$ if $j < n$, where $\hat{\alpha}_{j+1} = \alpha_{j+1}$ if α_{j+1} is a temporal role and $\hat{\alpha}_{j+1} = \Diamond r$ if α_{j+1} is the role name r.

To start the construction of $\mathfrak{I}$, set

- $\Delta_0 = \{(\alpha, \varepsilon, A_0)\}$ where α is any role (not important) and A_0 is the concept name from the left-hand side of the subsumption;
- $W_0 = \{\varepsilon\}$;
- $\pi((\alpha, \varepsilon, A_0), \varepsilon) = \mathsf{Q}(A_0)$.

For the *induction step*, we start with setting $\Delta_i = \Delta_{i-1}$, $W_i = W_{i-1}$, and $\pi_i = \pi_{i-1}$, and then inductively proceed according to the rules in Figure 4.5.

Finally, set $\Delta = \bigcup_{i \geq 0} \Delta_i$ and $W = \bigcup_{i \geq 0} W_i$. It remains to define the interpretation of concept and role names:

$$\begin{aligned} A^{\mathfrak{I},w} &= \{\sigma \in \Delta \mid A \in \pi(\sigma, w)\}; \\ r^{\mathfrak{I},w} &= \{(\sigma, \sigma\cdot(\Diamond r, v, A)) \mid \sigma\cdot(\Diamond r, v, A) \in \Delta,\ w = (\sigma, r, A)\} \cup \\ &\quad\ \{(\sigma, \sigma\cdot(r, w, A)) \mid \sigma\cdot(r, w, A) \in \Delta\} \cup \\ &\quad\ \{(\sigma, \sigma\cdot(\Box r, v, A)) \mid \sigma\cdot(\Box r, v, A) \in \Delta\}. \end{aligned}$$

First, we show a correspondence between types in the above construction and types of a trace.

Claim. For all $\sigma = (\alpha_1, w_1, A_1) \cdots (\alpha_n, w_n, A_n) \in \Delta$ and $w \in W$, we have that the following hold.

(a) $(A_j, A_{j+1}) \in \mathsf{R}(\hat{\alpha}_{j+1})$ for $1 \leq j < n$, and

(b) $\pi(\sigma, w)$ is either $\mathsf{Q}(A_n)$, $\mathsf{Q}_{\mathsf{cert}}(A_n)$ or $\Gamma(t)$ for some trace t to A_n.

Proof of the Claim. First, note that due to the definition of $\mathfrak{I}$, each $\pi(\sigma, w)$ is defined. The proof is by induction on the number of applications of rules **C1**-**C3**. The induction base follows directly from the definition of the initial step to define $\mathfrak{I}$; in particular of π_0 and W_0. For the induction step, we distinguish different cases depending on which rule is applied.

– Rule **C1**: first note that (b) remains trivially satisfied.

Now, consider (a), that is, assume $\exists\alpha.A \in \pi(\sigma, w)$ for some $w \in W$ and $\sigma' = \sigma\cdot(\alpha, w, A)$ is added to Δ. Because of the induction hypothesis it suffices to show that $(A_n, A) \in \mathsf{R}(\hat{\alpha})$. First, since $\pi_i(\sigma, w)$ is closed under cl, by rule **R5**, $\exists\hat{\alpha}.A \in \pi_i(\sigma, w)$. Now, it remains to note that, by I.H., $\pi_i(\sigma, w)$ is either $\mathsf{Q}(A_n)$, $\mathsf{Q}_{\mathsf{cert}}(A_n)$ or $\Gamma(t)$ for some trace t to A_n. In the first case, rule **S4** yields $(A_n, A) \in \mathsf{R}(\hat{\alpha})$; in the second case, by rule **S3**, $\exists\hat{\alpha}.A \in \mathsf{Q}(A_n)$, and then we argue as in the previous case. In the last case, by rule **S7**, $(A_n, A) \in \mathsf{R}(\hat{\alpha})$.

– Rule **C2**: the interesting point to show is (b), that is, assume there is some $\hat{\sigma} \in \Delta$ and $w \in W$ such that $\Diamond B \in \pi(\hat{\sigma}, w), \sigma = \hat{\sigma}|_j$ and $\pi(\sigma, (\hat{\sigma}, B)) = \Gamma(B, \hat{\sigma})$. Let

$$\hat{\sigma} = \sigma\cdot(\alpha_{n+1}, w_{n+1}, A_{n+1}) \cdots (\alpha_{n+k}, w_{n+k}, A_{n+k})$$

for some $k \geq 0$. Moreover, define the following sequence:

$$t = B, A_{n+k}, \hat{\alpha}_{n+k}, A_{n+k-1}, \ldots \hat{\alpha}_{n+1}, A_n.$$

We proceed to verify that t is a trace to A_n, that is, (1) $\Diamond B \in \mathsf{Q}(A_{n+k})$ and (2) for all $0 \leq i \leq k$ we have that $(A_{n+i}, A_{n+i+1}) \in \mathsf{R}(\hat{\alpha}_{n+i+1})$. Note that the later holds by I.H. in point (a).

Now, we show the first point. Note that, by I.H. of point (b), $\pi(\sigma, w)$ is defined as one of the following $\mathsf{Q}(A_{n+k})$, $\mathsf{Q}_{\mathsf{cert}}(A_{n+k})$ or $\Gamma(t')$ for some trace t' to A_{n+k}. We analyze the different cases: in the first case, by assumption, $\Diamond B \in \mathsf{Q}(A_{n+k})$; in the second case, $\Diamond B \in \mathsf{Q}_{\mathsf{cert}}(A_{n+k})$ and **S3** implies that $\Diamond B \in \mathsf{Q}(A_{n+k})$; in the last case, by rule **S6**, $\Diamond B \in \mathsf{Q}_{\mathsf{cert}}(A_{n+k})$, and then we argue as in the previous case.

Now, it remains to show that $\pi(\sigma, w) = \Gamma(t)$. To this aim we show by induction that for all $0 \leq j \leq k$ the following holds:

$$\Gamma_{n+k-j}(B, \hat{\sigma}) = \Gamma(t_{j+1})$$

and observe that in the equation above for $j = k$ we have that $\pi(\sigma, w) = \Gamma_n(B, \hat{\sigma}) = \Gamma(t_{k+1}) = \Gamma(t)$.

For the induction base, $j = 0$ we have that $t_1 = B, A_{n+k}$, and then we obtain the following:

$$\Gamma_{n+k}(B, \hat{\sigma}) = \mathsf{cl}(\{B\} \cup \mathsf{Q}_{\mathsf{cert}}(A_{n+k})) = \Gamma(t_1).$$

The induction step follows by applying the I.H., that is, we have that

$$t_j = B, A_{n+k}, \hat{\alpha}_{n+k}, \ldots, \hat{\alpha}_{n+k-j+1}, A_{n+k-j}.$$

Now, we have that by I.H., $\Gamma_{n+k-j-1}(B, \hat{\sigma}) = \Gamma(t_j)$, and then, by definition of the type of a trace, $\Gamma_{n+k-j}(B, \hat{\sigma}) = \Gamma(t_{j+1})$.

– Rule **C3** is analogous to the previous case.

This finishes the proof of the claim.

One consequence of this claim is that

$$*A \in \pi(\sigma, w) \text{ if and only if } *A \in \pi(\sigma, v) \tag{A1}$$

for all $\sigma = (\alpha_1, w_1, A_1) \cdots (\alpha_n, w_n, A_n) \in \Delta$ and $w, v \in W$.

We can check by case distinction that this is the case: if $*A \in Q(B)$, then by rule **S2** it will be in $\mathsf{Q}_{\mathsf{cert}}(B)$. Then, by definition, $*A \in \pi(\sigma, v)$ for all $v \in W$. If $*A \in \Gamma(t)$ for some trace t to A_n, then by rule **S6** $*A \in \mathsf{Q}_{\mathsf{cert}}(B)$, and then argue as in the previous case.

Another property that we will need later is that for all temporal roles α, it holds that

$$\sigma' = \sigma{\cdot}(\alpha, v, B) \wedge A \in \pi(\sigma', w) \Rightarrow \exists\alpha.A \in \pi(\sigma, w) \tag{A2}$$

This can be shown by a case distinction:

– $\pi(\sigma', w) = \mathsf{Q}(B)$. Then rule **C1** of the construction yields $\exists\alpha.B \in \pi(\sigma, w)$ and moreover $A \in \pi(\sigma', w) = \mathsf{Q}(B)$. Now by rule **R6** we obtain $\exists\alpha.A \in \pi(\sigma, w)$.

– $\pi(\sigma', w) = \mathsf{Q}_{\mathsf{cert}}(B)$. Thus $A \in \mathsf{Q}_{\mathsf{cert}}(B) \subseteq \mathsf{Q}(B)$. Further, by rule **C1** of the construction, $\sigma' \in \Delta$ implies that $\exists\alpha.B \in \pi(\sigma, v)$.

We have that $\exists\alpha.B \in \pi(\sigma, v)$ implies that $(A_n, B) \in \mathsf{R}(\alpha)$. To see this we distinguish the following cases: because of **S4** if $\pi(\sigma, v)$ equals $\mathsf{Q}(A_n)$ or $\mathsf{Q}_{\mathsf{cert}}(A_n)$, or because of **S7** if $\pi(\sigma, v)$ is the type of some trace.

Further note that, by definition and **R4**, $\Diamond A_n \in \mathsf{Q}(A_n)$, then $\Diamond A_n \in \mathsf{Q}_{\mathsf{cert}}(A_n)$. Now, we can apply **S5** in order to obtain $\exists\alpha.A \in \mathsf{Q}_{\mathsf{cert}}(A_n)$.

- $\pi(\sigma', w) = \Gamma(t)$ for some trace t to B. Now we define the trace $t' = t, \alpha, A_n$ which clearly is a trace to A_n. By definition of a type $\exists\alpha.B \in \Gamma(t')$ and by assumption $A \in \pi(\sigma', w)$. Moreover, $\pi(\sigma, w) = \Gamma(t')$. Therefore, $\exists\alpha.A \in \pi(\sigma, w)$.

We are now ready to show the central property of our model construction.

Claim. For all $\sigma \in \Delta$, $w \in W$, and $C \in \mathsf{BC}$, we have

$$\sigma \in C^{\mathfrak{I},w} \text{ iff } C \in \pi(\sigma, w).$$

Proof of the Claim. We prove the claim by a case distinction on the form of C. Throughout the following we assume $\sigma = (\alpha_1, w_1, A_1) \cdots (\alpha_n, w_n, A_n)$.

- $C = \top$.

  Then both $\sigma \in \top^{\mathfrak{I},w}$ and $\top \in \pi(\sigma, w)$ for all $\sigma \in \Delta$ and $w \in W$.

- $C = A \in \mathsf{CN}$. For this case, the lemma holds trivially by definition of the interpretation of concept names.

- $C = \Diamond A$. "if": Let $\sigma \in (\Diamond A)^{\mathfrak{I},w}$, that is, by the semantics, $\sigma \in A^{\mathfrak{I},v}$ for some $v \in W$. By I.H., this implies $A \in \pi(\sigma, v)$. Now, by **R4**, $\Diamond A \in \pi(\sigma, v)$, and by (A1) $\Diamond A \in \pi(\sigma, w)$.

  "only if": Let $\Diamond A \in \pi(\sigma, w)$. By rule **C2** of the construction, $A \in \pi(\sigma, (\sigma, A)) = \Gamma_n(A, \sigma)$. Now, by I.H., $\sigma \in A^{\mathfrak{I},(\sigma,A)}$. Thus, by semantics, $\sigma \in (\Diamond A)^{\mathfrak{I},w}$.

- $C = \Box A$. "if": Let $\sigma \in (\Box A)^{\mathfrak{I},w}$, that is, $\sigma \in A^{\mathfrak{I},v}$ for all $v \in W$. Now, by I.H., $A \in \pi(\sigma, v)$ for all $v \in W$. In particular, $A \in \pi(\sigma, \varepsilon)$. By construction, $\pi(\sigma, \varepsilon)$ is not the type of a trace, since the newly added σ's are realized in worlds different from ε. Furthermore, by rule **C1** the newly added domain elements σ will have $\pi(\sigma, w) = \mathsf{Q}(B)$ for some B in exactly one world w'. Thus, $\pi(\sigma, \varepsilon)$ will be $\mathsf{Q}_{\mathsf{cert}}(B)$, where $\sigma = \sigma' \cdot (\alpha, w', B)$. Now, by **S3** $\Box A \in Q(B)$, and by **S2** also $\Box A \in \mathsf{Q}_{\mathsf{cert}}(B) = \pi(\sigma, \varepsilon)$. Then, by (A1), $\Box A \in \pi(\sigma, w)$.

  "only if": Let $\Box A \in \pi(\sigma, w)$. By (A1), $\Box A \in \pi(\sigma, v)$ for all $v \in W$. Since all $\pi(\sigma, v)$ are closed under **R2**, $A \in \pi(\sigma, v)$ for all v. Now, by I.H., $\sigma \in A^{\mathfrak{I},v}$ for all $v \in W$. Therefore, by semantics, $\sigma \in (\Box A)^{\mathfrak{I},w}$.

- $C = \exists r.A$. "if": $\sigma \in (\exists r.A)^{\mathfrak{I},w}$, that is, there is a $\sigma' \in \Delta$ such that $\sigma' \in A^{\mathfrak{I},w}$ and $(\sigma, \sigma') \in r^{\mathfrak{I},w}$. Due to the model construction there are three possibilities for (σ, σ') being in $r^{\mathfrak{I},w}$:

* $\sigma' = \sigma\cdot(\Diamond r, v, B)$ for some concept B and $w = (\sigma, r, B)$: By construction, since σ' is not a prefix of σ, rule **C3** implies $\pi(\sigma', w) = \mathsf{Q}_{\mathsf{cert}}(B)$. Then, by I.H., $A \in \mathsf{Q}_{\mathsf{cert}}(B)$. Now, the definition of a type yields $\exists r.A \in \Gamma_n((r, B), \sigma) = \pi(\sigma, w)$.
* $\sigma' = \sigma\cdot(r, w, B)$: By rule **C1** of the construction, $\pi(\sigma', w) = \mathsf{Q}(B)$ Then, by I.H., $A \in \mathsf{Q}(B)$. Also by rule **C1**, $\exists r.B \in \pi(\sigma, w)$. Hence by **R6**, $\exists r.A \in \pi(\sigma, w)$.
* $\sigma' = \sigma\cdot(\Box r, v, B)$: We can apply (A2) in order to obtain $\exists\Box r.A \in \pi(\sigma, w)$. Now rule **R3** yields $\exists r.A \in \pi(\sigma, w)$.

"only if": Let $\exists r.A \in \pi(\sigma, w)$. By rule **C1** of the construction, there is a domain element $\sigma' = \sigma\cdot(r, w, A)$ with $\pi(\sigma', w) = \mathsf{Q}(A)$, thus $A \in \pi(\sigma', w)$. Then, by I.H., $\sigma' \in A^{\mathfrak{I},w}$. By definition of the interpretation of role names, $(\sigma, \sigma') \in r^{\mathfrak{I},w}$. Hence, $\sigma \in (\exists r.A)^{\mathfrak{I},w}$.

– $C = \exists\Box r.A$. "if": Let $\sigma \in (\exists\Box r.A)^{\mathfrak{I},w}$, that is, there is a σ' with $\sigma' \in A^{\mathfrak{I},w}$ and $(\sigma, \sigma') \in r^{\mathfrak{I},v}$ for all $v \in W$. Consider $v = \varepsilon$: from $(\sigma, \sigma') \in r^{\mathfrak{I},\varepsilon}$ follows that $\sigma' = \sigma\cdot(\Box r, v, B)$ for some world $v \in W$ and concept name B. Applying (A2) yields $\exists\Box r.A \in \pi(\sigma, w)$.

"only if": Let $\exists\Box r.A \in \pi(\sigma, w)$. By rule **C1** of the construction, there is a domain element $\sigma' = \sigma\cdot(\Box r, w, A)$ with $\pi(\sigma', w) = \mathsf{Q}(A)$, thus $A \in \pi(\sigma', w)$. Then, by I.H., $\sigma' \in A^{\mathfrak{I},w}$. By definition of the interpretation of role names, $(\sigma, \sigma') \in r^{\mathfrak{I},v}$ for all $v \in W$. Hence, $\sigma \in (\exists\Box r.A)^{\mathfrak{I},w}$.

– $C = \exists\Diamond r.A$."if": Let $\sigma \in (\exists\Diamond r.A)^{\mathfrak{I},w}$, that is, there is a σ' with $\sigma' \in A^{\mathfrak{I},w}$ and $(\sigma, \sigma') \in r^{\mathfrak{I},v}$ for some $v \in W$. Again we distinguish the three cases of the interpretation of the roles.

* $\sigma' = \sigma\cdot(\Diamond r, v', B)$ and $w = (\sigma, r, B)$ for some concept B. By construction, since σ' is not a prefix of σ, rule **C3** implies $\pi(\sigma', w) = \mathsf{Q}_{\mathsf{cert}}(B)$, thus $A \in \mathsf{Q}_{\mathsf{cert}}(B)$. Also by construction rule **C3**, $\pi(\sigma, w) = \Gamma_n((r, B), \sigma)$, and by the definition of Γ_n this yields $\exists r.A \in \Gamma_n((r, B), \sigma)$. So by rule **R5**, $\exists\Diamond r.A \in \pi(\sigma, w)$.
* $\sigma' = \sigma\cdot(r, w, B)$. By rule **C1**, $\pi(\sigma', w) = \mathsf{Q}(B)$ and $\exists r.B \in \pi(\sigma, w)$. This implies $A \in \mathsf{Q}(B)$, and since $\pi(\sigma, w)$ is closed under **R6**, $\exists r.A \in \pi(\sigma, w)$. Thus, by **R5**, $\exists\Diamond r.A \in \pi(\sigma, w)$.
* $\sigma' = \sigma\cdot(\Box r, v, B)$. Applying (A2) yields $\exists\Box r.A \in \pi(\sigma, w)$. Using rules **R3** and **R5** we obtain $\exists\Diamond r.A \in \pi(\sigma, w)$.

"only if". Let $\exists\Diamond r.A \in \pi(\sigma, w)$. By rule **C1** of the construction there is a domain element $\sigma' = \sigma\cdot(\Diamond r, w, A)$ with $\pi(\sigma', w) = \mathsf{Q}(A)$. Then, by I.H., $\sigma' \in A^{\mathfrak{I},w}$. By definition of the interpretation of role names $(\sigma, \sigma') \in r^{\mathfrak{I},v}$ for $v = (\sigma', A, r)$. Hence $\sigma \in (\exists\Diamond r.A)^{\mathfrak{I},w}$.

This finishes the proof of the claim.

It remains to show that for $\sigma_0 = (\alpha, \varepsilon, A_0)$ we have $\sigma_0 \in A_0^{\mathfrak{I},\varepsilon}$, but $\sigma_0 \notin B_0^{\mathfrak{I},\varepsilon}$. However, both are obviously true: first we note that, by construction, $\pi(\sigma_0, \varepsilon) = \mathsf{Q}(A_0)$. By definition, $A_0 \in Q(A_0)$, hence $\sigma_0 \in A_0^{\mathfrak{I},\varepsilon}$ by the above claim. On the other hand, by assumption we have $B_0 \notin \mathsf{Q}(A_0)$, thus by the above claim $\sigma_0 \notin B_0^{\mathfrak{I},\varepsilon}$.

❏

We now argue that the algorithm can be implemented using only polynomial space. First, note that there can be only polynomially many rule applications: every rule application extends the data structures $\mathsf{Q}(\cdot)$, $\mathsf{Q}_{\mathsf{cert}}(\cdot)$, and $\mathsf{R}(\cdot)$, but these structures consist of polynomially many sets, each with at most polynomially many elements. It thus remains to verify that each rule application can be executed using only polyspace, which is obvious for all rules except those involving traces, that is, **S6** and **S7**. For these rules, we first note that it is not necessary to consider all (infinitely many!) traces. In fact, we next show that it is enough to only consider *non-repeating traces*.

Proposition 4.5. *Let $B \in \mathsf{CN}$. If there is a trace t to B, then there is a trace t' to B with $|t'| \leq |\mathcal{T}| \cdot 2^{|\mathcal{T}|}$ such that $\Gamma(t) = \Gamma(t')$.*

Proof. Let $t = S, A_1, \alpha_2, \ldots, \alpha_n, A_n$ and $|t| \geq |\mathcal{T}| \cdot 2^{|\mathcal{T}|}$. We next denote by Γ_i the type of the trace t_i, that is, $\Gamma(t_i)$. We consider the sequence $(A_1, \Gamma_1), \ldots, (A_n, \Gamma_n)$ of concept names from t with their types. First note that there are at most $2^{|\mathcal{T}|}$ types and at most $|\mathcal{T}|$ concept names. Now, since we assume that $|t| \geq |\mathcal{T}| \cdot 2^{|\mathcal{T}|}$, then there are $1 \leq i < j \leq n$ such that $A_i = A_j$ and $\Gamma_i = \Gamma_j$. Now, we can straightforwardly construct a trace t' such that $\Gamma(t') = \Gamma(t)$ as follows:

$$t' = S, A_1, \alpha_2, \ldots, \alpha_i, A_i, \alpha_{j+1}, A_{j+1}, \ldots, A_n.$$

Clearly, by construction, t' is shorter than t. Now, we proceed as follows: if $|t'| \leq |\mathcal{T}| \cdot 2^{|\mathcal{T}|}$, we are done. Otherwise, repeat the identification procedure above.

❏

To get to polyspace, we use a nondeterministic approach, enabled by Savitch's theorem: to check whether there is a trace t to B with $C \in \Gamma(t)$, we guess t *step-by-step*, at each time keeping only a single A_i, α_i and $\Gamma(t_i)$ in memory. When we reach a situation where $A_i = B$ and $C \in \Gamma(t_i)$, our guessing was successful and we apply the rule. We also maintain a binary counter of the number of steps that have been guessed so far. As soon as this counter exceeds $|\mathcal{T}| \cdot 2^{|\mathcal{T}|}$, the maximum length of non-repeating traces, we stop the guessing and do not apply the rule. Clearly, this yields a polyspace algorithm.

Theorem 4.6. *Concept subsumption w.r.t. TBoxes for $\mathbf{S5}_{\mathcal{EL}}$ with temporal roles is* PSPACE-*complete.*

We finalize our study by investigating the computational complexity of $\mathbf{S5}_{\mathcal{ELI}}$. We show, as previously discussed, that the presence of inverse roles in $\mathbf{S5}_{\mathcal{ELI}}$ makes subsumption harder than for $\mathbf{S5}_{\mathcal{EL}}$. In particular, it increases from PSPACE-complete for $\mathbf{S5}_{\mathcal{EL}}$ to 2EXPTIME-complete for $\mathbf{S5}_{\mathcal{ELI}}$. This shows that the computational complexity of $\mathbf{S5}_{\mathcal{ELI}}$ coincides with that of $\mathbf{S5}_{\mathcal{ALCQI}}$ [10]. Note that $\mathbf{S5}_{\mathcal{ALCQI}}$ provides a 2EXPTIME upper bound for $\mathbf{S5}_{\mathcal{ELI}}$. We next commit ourselves to the development of a matching lower bound.

4.4.2 A 2EXPTIME Lower Bound for Concept Subsumption w.r.t. TBoxes for $\mathrm{S5}_{\mathcal{ELI}}$ with Temporal Roles

We demonstrate a 2EXPTIME lower bound for concept subsumption w.r.t. TBoxes for $\mathbf{S5}_{\mathcal{ELI}}$ by a reduction of the word problem of an exponentially space bounded alternating Turing machine (*cf.* Section 3.5.3). The core idea of the reduction is to combine the techniques: first, for showing EXPTIME-hardness for concept subsumption w.r.t. TBoxes for $\mathcal{ELI}$ [13]. Second, for showing 2EXPTIME-hardness for concept subsumption w.r.t. TBoxes for $\mathbf{S5}_{\mathcal{ALC}}$ [10]. In particular, we use the possibility of introducing exponentially many worlds associated with each node of the $\mathcal{ELI}$-tree implementing the ATM computation. Notably, due to the weak expressiveness of $\mathcal{ELI}$, a careful synchronization of the information propagated across the temporal and DL dimension is needed to correctly amalgamate these techniques.

Theorem 4.7. *Concept subsumption w.r.t. TBoxes for* $\mathbf{S5}_{\mathcal{ELI}}$ *with temporal roles is* 2EXPTIME-*hard.*

Proof. The proof is by a reduction of the word problem of an exponentially spaced bounded alternating Turing machine (*cf.* Section 3.5.3). Let $\mathcal{M} = (Q, \Sigma, \Gamma, q_0, \delta)$ be such an ATM with a 2EXPTIME-hard word problem, and let $w \in \Sigma^*$ be the input of length n. Our aim is to construct in polynomial time a TBox $\mathcal{T}_{\mathcal{M},w}$ and concepts A, B such that $\mathcal{T}_{\mathcal{M},w} \models A \sqsubseteq B$ iff $\mathcal{M}$ accepts w.

The basic idea of our reduction is to capture the configurations of an ATM by means of $\mathbf{S5}$ models, where each $\mathbf{S5}$-world will be identified with a tape cell. As proposed by Artale *et al.* [10], an acceptance tree for an ATM is then encoded as an $\mathcal{ELI}$ tree with an $\mathbf{S5}$ model attached to each node. We use the following signature:

- concept names $A_a, \overline{A_a}$ for every $a \in \Gamma$;
- concept names Q_q for every $q \in Q$;
- concept names $C_0, \overline{C_0}, \ldots, C_{n-1}, \overline{C_{n-1}}$ and $C'_0, \overline{C'_0}, \ldots, C'_{n-1}, \overline{C'_{n-1}}$ for encoding exponential counters;
- concept names $C'_{=i}$ for $i \in [0, n-1]$ (polynomially many $i's$) to represent that the value of the counter C' is i;
- concept names $\mathsf{Level}_0, \ldots, \mathsf{Level}_n$ for identifying the levels of a binary tree;
- concept names $\mathsf{Head}_0, \overline{\mathsf{Head}_0} \ldots \mathsf{Head}_{n-1}, \overline{\mathsf{Head}_{n-1}}$ for encoding a counter representing the position of the head of an ATM;
- concept name Fail indicating any kind of *failure*;
- additional (auxiliary) concept names $\mathsf{Correct}, \overline{\mathsf{Correct}}, \mathsf{Correct}^{\mathsf{H}}, \overline{\mathsf{Correct}^{\mathsf{H}}}, \mathsf{MCell}$.

We begin by constructing a $\mathcal{ELI}$ binary tree of depth n rooted in $C_{\mathcal{M},w}$ for identifying exponentially many tape cells. Note that this is only an auxiliary tree, it does not encode the ATM computation. For $1 \leq i < n$, $\mathcal{T}_{\mathcal{M},w}$ contains the following CIs:

$$C_{\mathcal{M},w} \sqsubseteq \mathsf{Level}_0 \sqcap \exists\Box r.C_0 \sqcap \exists\Box r.\overline{C_0}, \tag{4.18}$$

$$C_i \sqsubseteq \mathsf{Level}_i \sqcap \exists\Box r.C_{i+1} \sqcap \exists\Box r.\overline{C_{i+1}}, \tag{4.19}$$

$$\overline{C_i} \sqsubseteq \mathsf{Level}_i \sqcap \exists\Box r.C_{i+1} \sqcap \exists\Box r.\overline{C_{i+1}}, \tag{4.20}$$

$$\exists r^-.(\mathsf{Level}_i \sqcap C_i) \sqsubseteq C_i, \qquad \exists r^-.(\mathsf{Level}_i \sqcap \overline{C_i}) \sqsubseteq \overline{C_i}, \tag{4.21}$$

$$\exists r^-.\mathsf{Level}_i \sqsubseteq \mathsf{Level}_{i+1}. \tag{4.22}$$

We create copies of each leave (nodes at Level_n) of the $\mathcal{ELI}$-tree across the $\mathbf{S5}$ dimension via an auxiliary marker M (4.23). This forces the introduction of exponentially many $\mathbf{S5}$-worlds. The main objective of these copies is to store the content of i-tape cells from different configurations. We also ensure that the values C_i of the counter and the levels Level_i are globally propagated across the $\mathbf{S5}$-worlds (4.24)-(4.25).

$$\mathsf{Level}_n \sqsubseteq \overline{\mathsf{M}} \sqcap \Diamond\mathsf{M}, \tag{4.23}$$

$$C_i \sqsubseteq \Box C_i, \qquad \overline{C_i} \sqsubseteq \Box\overline{C_i}, \tag{4.24}$$

$$\mathsf{Level}_i \sqsubseteq \Box\mathsf{Level}_i. \tag{4.25}$$

We introduce an auxiliary counter C', which is coordinated with the main counter C through the marker M (4.26). We moreover ensure that *all* the elements of the $\mathcal{ELI}$-tree share the same counter value C' (4.27)-(4.28), –role s is used later in the implementation of the ATM.

$$\mathsf{M} \sqcap C_i \sqsubseteq C'_i, \qquad \mathsf{M} \sqcap \overline{C_i} \sqsubseteq \overline{C'_i}, \tag{4.26}$$

$$\exists\{r,s\}.C'_i \sqsubseteq C'_i, \qquad \exists\{r,s\}.\overline{C'_i} \sqsubseteq \overline{C'_i}, \tag{4.27}$$

$$\exists\{r^-,s^-\}.C'_i \sqsubseteq C'_i, \qquad \exists\{r^-,s^-\}.\overline{C'_i} \sqsubseteq \overline{C'_i}. \tag{4.28}$$

Now that we have the desired structure for representing an ATM (see Figure 4.6), we can proceed to encode an ATM computation. We represent, as discussed above, each $q \in Q$ with the concept name Q_q and each $a \in \Sigma$ with the concept name A_a.

We enforce the *initial configuration*: $\mathcal{M}$ is in the initial state q_0, the head is in the left-most tape cell, and the input word is $w = a_0, \ldots, a_{n-1}$ followed by blanks ($\llcorner\!\lrcorner$). One can standardly write concepts $C'_{=i}$ –polynomially many– for representing that the value of the counter C' is i. Analogously, one can write a concept $C'_{>n-1}$.

$$\mathsf{Level}_0 \sqcap (C'_{=i}) \sqsubseteq A_{a_i}, \tag{4.29}$$

$$\mathsf{Level}_0 \sqcap (C'_{=0}) \sqsubseteq Q_{q_0}, \tag{4.30}$$

$$\mathsf{Level}_0 \sqcap (C'_{>n-1}) \sqsubseteq A_{\llcorner\!\lrcorner}. \tag{4.31}$$

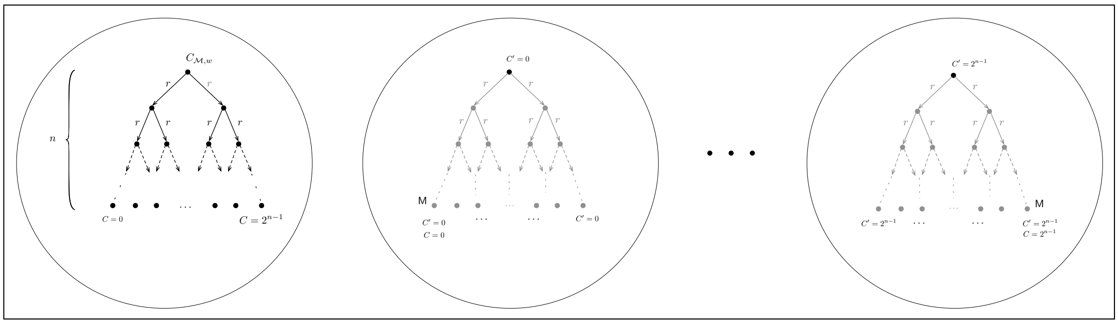

Figure 4.6: Exponentially many worlds: each representing the content of a tape cell in a possible configuration.

We use a counter Head to set the position of the head (in the initial configuration is at position 0) (4.32). We moreover ensure that the value of the counter is propagated across the $\mathbf{S5}$-dimension (4.33).

$$\mathsf{Level}_0 \sqsubseteq \mathsf{Head}_{=0} \sqcap \mathsf{Initial}, \tag{4.32}$$

$$\mathsf{Head}_i \sqsubseteq \Box\mathsf{Head}_i, \qquad \overline{\mathsf{Head}_i} \sqsubseteq \Box\overline{\mathsf{Head}_i}. \tag{4.33}$$

For each $q \in Q, a \in \Sigma$ and $m \in \{l, r, n\}$, we use marker concepts $M_{q,a,m}$ to represent the moves of $\mathcal{M}$. At this point, we begin using the role s to encode the computation of an ATM.

$$Q_q \sqcap A_a \sqsubseteq \exists\Box s.M_{q',b,m} \text{ for all } (q', b, m) \in \delta(q, a). \tag{4.34}$$

We proceed to ensure that the head is set for one tape cell. In particular, we compare the value of the head counter of the s-predecessor with the position of a cell (4.35)-(4.38). This allows us to verify what is the position of the head in the previous configuration

$$C'_0 \sqcap \exists s^-.\mathsf{Head}_0 \sqsubseteq \mathsf{Correct}_0, \qquad \overline{C'_0} \sqcap \exists s^-.\overline{\mathsf{Head}_0} \sqsubseteq \mathsf{Correct}_0, \tag{4.35}$$

$$\mathsf{Correct}_{i-1} \sqcap C'_i \sqcap \exists s^-.\mathsf{Head}_i \sqsubseteq \mathsf{Correct}_i, \quad \mathsf{Correct}_{i-1} \sqcap \overline{C'_i} \sqcap \exists s^-.\overline{\mathsf{Head}_i} \sqsubseteq \mathsf{Correct}_i, \tag{4.36}$$

$$\mathsf{Correct}_{n-1} \sqsubseteq \mathsf{Correct}, \tag{4.37}$$

$$C'_i \sqcap \exists s^-.\overline{\mathsf{Head}_i} \sqsubseteq \overline{\mathsf{Correct}}, \qquad \overline{C'_i} \sqcap \exists s^-.\mathsf{Head}_i \sqsubseteq \overline{\mathsf{Correct}}. \tag{4.38}$$

Now, if we are at the cell in which the head was positioned in the previous configuration (Correct) and encounter the $M_{q,a,r}$ marker, then we *increment* the head-counter Head (4.39)-(4.40). This captures the fact that the head should move to the *right*.:

For every $0 \leq j < i < n$

$$\begin{aligned} \mathsf{Correct} \sqcap M_{q,a,r} \sqcap \exists s^-.(\overline{\mathsf{Head}_i} \sqcap \overline{\mathsf{Head}_j}) &\sqsubseteq \overline{\mathsf{Head}_i}, \\ \mathsf{Correct} \sqcap M_{q,a,r} \sqcap \exists s^-.(\mathsf{Head}_i \sqcap \overline{\mathsf{Head}_j}) &\sqsubseteq \overline{\mathsf{Head}_i}. \end{aligned} \tag{4.39}$$

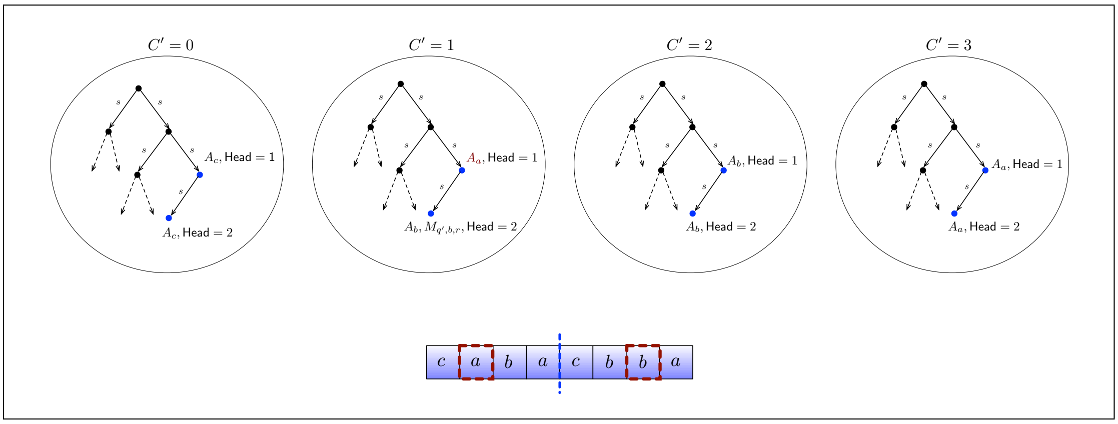

Figure 4.7: A transition between succeeding configurations for $n = 2$ and $(q', b, r) \in \delta(q, a)$.

For every $0 \leq j < n$

$$\begin{aligned}\mathsf{Correct} \sqcap M_{q,a,r} \sqcap \exists s^-.(\overline{\mathsf{Head}_j} \sqcap \mathsf{Head}_{j-1} \sqcap \ldots \mathsf{Head}_0) &\sqsubseteq \overline{\mathsf{Head}_j}, \\ \mathsf{Correct} \sqcap M_{q,a,r} \sqcap \exists s^-.(\mathsf{Head}_j \sqcap \mathsf{Head}_{j-1} \sqcap \ldots \mathsf{Head}_0) &\sqsubseteq \overline{\mathsf{Head}_j}.\end{aligned} \tag{4.40}$$

(*) Analogously we can *decrease* the head-counter to capture the movement of the head to the left, when we encounter the marker $M_{q,a,l}$.

We again use an auxiliary marker ($\mathsf{Correct}^{\mathsf{H}}$) to check whether the value of the head counter and that of a tape cell coincide. This allows us to check the current position of the head.

$$C'_0 \sqcap \mathsf{Head}_0 \sqsubseteq \mathsf{Correct}^{\mathsf{H}}_0, \qquad \overline{C'_0} \sqcap \overline{\mathsf{Head}_0} \sqsubseteq \mathsf{Correct}^{\mathsf{H}}_0, \tag{4.41}$$

$$\mathsf{Correct}^{\mathsf{H}}_{i-1} \sqcap C'_i \sqcap \mathsf{Head}_i \sqsubseteq \mathsf{Correct}^{\mathsf{H}}_i, \qquad \mathsf{Correct}^{\mathsf{H}}_{i-1} \sqcap \overline{C'_i} \sqcap \overline{\mathsf{Head}_i} \sqsubseteq \mathsf{Correct}^{\mathsf{H}}_i, \tag{4.42}$$

$$\mathsf{Correct}^{\mathsf{H}}_{n-1} \sqsubseteq \mathsf{Correct}^{\mathsf{H}}, \tag{4.43}$$

$$C'_i \sqcap \overline{\mathsf{Head}_i} \sqsubseteq \overline{\mathsf{Correct}^{\mathsf{H}}}, \qquad \overline{C'_i} \sqcap \mathsf{Head}_i \sqsubseteq \overline{\mathsf{Correct}^{\mathsf{H}}}. \tag{4.44}$$

Now, we have the necessary ingredients to implement the transitions: for $q \in Q$, and $a \in \Sigma$, set

$$\mathsf{Correct} \sqcap M_{q,a,n} \sqsubseteq A_a \sqcap Q_q, \tag{4.45}$$

$$\mathsf{Correct} \sqcap M_{q,a,\{l,r\}} \sqsubseteq A_a, \tag{4.46}$$

$$\mathsf{Correct}^{\mathsf{H}} \sqcap \Diamond(M_{q,a,\{l,r\}} \sqcap \mathsf{Correct}) \sqsubseteq Q_q. \tag{4.47}$$

We establish standard structural requirements for ATMs by identifying potential defects via the '*failure*' concept Fail. For example, if we have an inconsistency in the information of a tape cell, if we have inconsistent counters, etc. Some of these defects are the following:

$$A_a \sqcap A_b \sqsubseteq \mathsf{Fail}, \qquad A_a \sqcap \overline{A_a} \sqsubseteq \mathsf{Fail}, \qquad \bigsqcap_{a \in \Gamma} \overline{A_a} \sqsubseteq \mathsf{Fail}, \tag{4.48}$$

$$Q_q \sqcap Q_{q'} \sqsubseteq \mathsf{Fail}, \tag{4.49}$$

$$C_i \sqcap \overline{C_i} \sqsubseteq \mathsf{Fail}, \quad C'_i \sqcap \overline{C'_i} \sqsubseteq \mathsf{Fail}, \quad \mathsf{Head}_i \sqcap \overline{\mathsf{Head}_i} \sqsubseteq \mathsf{Fail}, \quad \mathsf{MCell}_i \sqcap \overline{\mathsf{MCell}_i} \sqsubseteq \mathsf{Fail}, \tag{4.50}$$

$$\mathsf{Correct} \sqcap \overline{\mathsf{Correct}} \sqsubseteq \mathsf{Fail}, \qquad \mathsf{Correct}_i^{\mathsf{H}} \sqcap \overline{\mathsf{Correct}_i^{\mathsf{H}}}. \tag{4.51}$$

The concept Fail is propagated to the initial configuration (4.52).

$$\exists s.\mathsf{Fail} \sqsubseteq \mathsf{Fail}. \tag{4.52}$$

We capture the fact that tape cells that are not in the head position do not change their content.

$$\exists s^-.(\overline{\mathsf{Correct}} \sqcap A_a) \sqsubseteq A_a. \tag{4.53}$$

We identify *accepting* configurations (4.54) (4.56). In particular, we verify that they whether the states are *universal* or *existential*. For all $q \in Q_\forall, q' \in Q_\exists$ and $a \in \Sigma$ set:

$$Q_{q_a} \sqsubseteq A, \tag{4.54}$$

$$Q_q \sqcap \mathsf{Correct}^{\mathsf{H}} \sqcap A_a \sqcap \bigsqcap_{(q'',b,m)\in\delta(q,a)} \exists s.(M_{q'',b,m} \sqcap \Diamond A) \sqsubseteq A, \tag{4.55}$$

$$Q_{q'} \sqcap \exists s.(M_{q'',b,m} \sqcap \Diamond A) \sqsubseteq A \text{ for all } (q'', b, m) \in \delta(q, a). \tag{4.56}$$

The initial configuration is Good if it is either accepting or a defect has been detected:

$$C_{\mathcal{M},w} \sqcap \Diamond A \sqsubseteq \mathsf{Good}, \tag{4.57}$$

$$C_{\mathcal{M},w} \sqcap \Diamond \mathsf{Fail} \sqsubseteq \mathsf{Good}. \tag{4.58}$$

This finishes the construction of the TBox, now following the construction is not difficult to see that the following proposition holds

Proposition 4.8. *$\mathcal{M}$ accepts w iff $\mathcal{T}_{\mathcal{M},w} \models C_{\mathcal{M},w} \sqsubseteq \mathsf{Good}$.*

First assume $w \notin L(\mathcal{M})$. We construct a model $\mathfrak{I} = (W, \Delta, \{\mathcal{I}_w\}_{w\in W})$ of $\mathcal{T}_{\mathcal{M},w}$ such that $\mathcal{T}_{\mathcal{M},w} \not\models C_{\mathcal{M},w} \sqsubseteq \mathsf{Good}$ as follows. Let $W = \{0, \dots, 2^n\}$ be the set of worlds and $\Delta = \mathsf{conf}(\mathcal{M}) \cup \{n_1, \dots, n_{2^{n+1}}\}$, where $\mathsf{conf}(\mathcal{M})$ is the set of all configurations of $\mathcal{M}$. For $i \in W, i < 2^n$ we define:

- $C'^{\mathfrak{I},i}_j = \{n \in \mathsf{conf}(\mathcal{M}) \mid \text{the } j^{th} \text{ bit of the binary representation of } i \text{ is } 1\}$;
- $A_a^{\mathfrak{I},i} = \{n \in \mathsf{conf}(\mathcal{M}) \mid a \text{ is the } i^{th} \text{ symbol on the tape in } n\}$;
- $Q_q^{\mathfrak{I},i} = \{n \in \mathsf{conf}(\mathcal{M}) \mid \mathcal{M} \text{ is in state } q \text{ and the head is at position } i \text{ in } n\}$;
- $\mathsf{Head}_i = \{n \in \mathsf{conf}(\mathcal{M}) \mid \mathcal{M} \text{ is in state } q \text{ and the head is at position } i \text{ in } n\}$;
- $s^{\mathfrak{I},i} = \{(n, n') \in \Delta\times\Delta \mid n' \text{ is a successor configuration of } n\}$.

This interpretation can be immediately extended to the remaining auxiliary concepts used in the construction, e.g., the markers $M_{q,a,m}$, $\mathsf{Correct}$, $\mathsf{Correct}^{\mathsf{H}}$, etc.

The additional world $2^n \in W$ is used as the *initial world* from which the original construction starts: we simply set $C_{\mathcal{M},w}^{\mathfrak{I},2^n} = \{q_0 w\}$. In addition, we need to attach the auxiliary $\mathcal{ELI}$ tree formed by the role r and the nodes $\{n_1, \ldots, n_{2^n+1}\}$ to the root of this interpretation. Moreover, we interpret the counter C appropriately:

– $C_j{}^{\mathfrak{I},i} = \{n \in \{n_1, \ldots, n_{2^n+1}\} \mid \text{the } j^{th} \text{ bit of the binary representation of } i \text{ is } 1\}$.

Finally, we interpret $A^{\mathfrak{I},i}$ as the set of accepting configurations, and $\mathsf{Fail}^{\mathfrak{I},i}$ and $\mathsf{Good}^{\mathfrak{I},i}$ as the empty set. By inspection of the axioms in $\mathcal{T}_{\mathcal{M},w}$ one can see that $\mathfrak{I} \models \mathcal{T}_{\mathcal{M}}$. Moreover, $\{q_0 w\} \in C_{\mathcal{M},w}^{\mathfrak{I},2^n}$ but $\{q_0 w\} \notin \mathsf{Good}^{\mathfrak{I},2^n}$.

Conversely, assume that $\mathcal{T}_{\mathcal{M},w} \not\models C_{\mathcal{M},w} \sqsubseteq \mathsf{Good}$, that is, there is a model $\mathfrak{I} = (W, \Delta, \{\mathcal{I}_w\}_{w \in W})$ of $\mathcal{T}_{\mathcal{M},w}$ such that there is a $n \in C_{\mathcal{M},w}^{\mathfrak{I},v}$ but $n \notin \mathsf{Good}^{\mathfrak{I},v}$ for some $v \in W$. From the construction of $\mathcal{T}_{\mathcal{M},w}$ it is clear that there are $v_0, \ldots, v_{2^n-1}$ such that $n \in (C'_{=i})^{\mathfrak{I},v_i}$ for $0 \le i < 2^n$. If there are more than one world for a i we just simply pick one. Now we define

$$\mathsf{conf}(m) = a_0 a_1 \ldots a_{j-1} q a_j \ldots a_{2^n-1}$$

for m a s^ℓ successor of n for $\ell \ge 0$ such that $m \in A_{a_i}^{\mathfrak{I},v_i}$ for all $0 \le i < 2^n$, and $m \in Q_q^{\mathfrak{I},v_j}$. Moreover, note that $\mathcal{T}_{\mathcal{M}}$ guarantees that there is exactly one head per configuration and exactly one symbol per tape cell. Thus the $\mathsf{conf}(.)$ function is well defined. Now we map the pair (n, v) to the root of the tree and use the s role to inductively construct the complete tree. Now, since $n \notin \mathsf{Good}^{\mathfrak{I},v}$, then $n \notin \Diamond A^{\mathfrak{I},v}$, and also $n \notin \Diamond \mathsf{Fail}^{\mathfrak{I},v}$. This means that $q_0 w$ is not accepting, and thus $w \notin L(\mathcal{M})$.

❑

Theorem 4.9. *Concept subsumption w.r.t. TBoxes for* $\mathbf{S5}_{\mathcal{ELI}}$ *with temporal roles is* 2EXPTIME-*complete.*

4.5 Conclusions

Finding the right trade-off between expressiveness and complexity is one of the main challenges towards the design of useful TDLs. In particular, the construction of effective TDLs allowing for temporal or rigid roles is crucial for many applications; for example, in medical ontologies such as SNOMED CT [15] or for temporal data modeling [8]. Alas, combinations of standard TLs and the lightweight DL $\mathcal{EL}$ allowing for temporal concepts and temporal (or rigid) roles turned out undecidable. A possibility to attain decidability is to use weaker logics for the DL or temporal component. In this chapter, we focused on the study of TDLs with a weaker temporal dimension given by the modal logic $\mathbf{S5}$. Notably, these TDLs allow to reason about the change of knowledge without differentiating between changes in the past or future. We investigated the impact of having members of the $\mathcal{EL}$ family instead of $\mathcal{ALC}$ as the DL component on the computational complexity of DLChs. We showed that reasoning in the TDL $\mathbf{S5}_{\mathcal{EL}}$ with temporal roles is indeed easier than in the $\mathcal{ALC}$ variant (in contrast to TDLs based on LTL and CTL): the complexity goes down from 2EXPTIME-complete to PSPACE-complete. We moreover showed that pushing further this result to $\mathbf{S5}_{\mathcal{ELI}}$ is not possible. Reasoning in $\mathbf{S5}_{\mathcal{ELI}}$ becomes hard for 2EXPTIME and then as hard as in the $\mathcal{ALC}$ variant. We also investigated the DLCh based on the extension of $\mathcal{ALC}$ with nominals –$\mathcal{ALCO}$–, allowing for $\mathbf{S5}$-modalities to be applied only to concepts. We showed that the computational complexity increases from EXPTIME for $\mathcal{ALCO}$ to NEXPTIME for $\mathbf{S5}_{\mathcal{ALCO}}$. Interestingly, this jump in the complexity is not present in the $\mathcal{ALC}$-variant for which the complexity remains in EXPTIME. This shows that the presence of nominals is responsible for the increase in the complexity of $\mathbf{S5}_{\mathcal{ALCO}}$.

The work presented in this chapter broadened the understanding of the computational complexity of DLs of change. We showed that in DLs of change the computational complexity indeed varies depending on whether they are based on either $\mathcal{ALCO}$, $\mathcal{ALC}$ or $\mathcal{EL}$. Some interesting theoretical and practical problems, however, remain open. One important research line is to investigate the adequacy of *temporal TBoxes* –allowing for the application $\mathbf{S5}$-modalities to TBoxes– for modeling the change of policies over time, and their impact on the computational complexity. Another possibility is to investigate DLs of change based on the lightweight members of the DL-Lite-family.

5 Description Logics of Context

In this chapter, we investigate two-dimensional DLs for representing and reasoning about contextualized knowledge. We introduce a novel family of two-dimensional, two-sorted description logics implementing McCarthy's theory of formalizing contexts. The main technical contribution are algorithms for KB satisfiability, and tight complexity bounds that range from NEXPTIME to 2EXPTIME. We also show the relation of the proposed formalism with well-known modal description logics (which we consider as simple DLs of context), and its applicability to diverse problems such as modeling inherently contextualized knowledge or expressing interoperability constraints over DL ontologies.

5.1 Introduction

One of the consequences of the inability of classical description logics to capture dynamic aspects of knowledge is the impossibility of representing heterogeneous viewpoints on an application domain or to represent context-sensitive knowledge. Alas, this becomes a drawback for many practical applications. For example, in ontology applications related with reasoning over distributed knowledge sources on the semantic web [43, 17]. In particular, the way classical DLs are designed, and their semantics is defined force an ontology to impose a unique, global and uniform view on the represented domain. The axioms of an ontology $\mathcal{O}$ are thus interpreted as unconditionally and universally true in all models of $\mathcal{O}$. For example,

$$\mathsf{Heart} \sqsubseteq \mathsf{HumanOrgan} \in \mathcal{O}$$

enforces all domain individuals of type Heart to be of type HumanOrgan in all possible models of $\mathcal{O}$. These capabilities offered by classical DLs are well-suited for applications where everyone shares the same conceptual perspective on the domain or if there is no need for considering alternative viewpoints. However, in many important applications the domain should be in fact modeled differently depending on the context –viewpoint– in which it is considered, where the context might depend on a spatio-temporal coordinate, the thematic focus, a subjective perspective of the modeler, the adopted level of granularity of the representation, an intended application of the ontology, etc. For instance, the axiom $\mathsf{Heart} \sqsubseteq \mathsf{HumanOrgan}$ is valid in the domain of human anatomy, but this might not be necessarily the case once a broader perspective of mammal anatomy is considered. Moreover, the intrinsic inability of accounting for contexts or possible viewpoints in DLs seems to hinder the usability of DLs in two very basic application scenarios:

`(I)`**.** It is impossible to create ontologies that would be at the same time general enough as to cover all relevant knowledge about the domain and yet sufficiently detailed as to capture all context-related peculiarities occurring in this knowledge. This challenge is commonly faced by the creators of huge knowledge bases, aiming at maximum coverage of the representation, such as SNOMED CT [74] or CYC [56], and typically leads to the development of ad hoc, application-driven mechanisms of contextualization.

`(II)`**.** The second problem concerns the reuse of knowledge from multiple existing sources –such as the numerous DL-based ontologies already published on the Web– in new applications. Naturally, portions of such knowledge retrieved from different ontologies are likely to pertain to different, heterogenous contexts, which are implicitly assumed during the creation of the sources. Consequently, a faithful reuse of such data cannot be achieved without special semantic mechanisms which acknowledge and respect its local, context-specific character [43, 17].

The research on description logics, as noticed in Section 1.2, has considered variants of these two types of problems. On the one hand, DLs have been commonly extended with constructors facilitating direct modeling of contextualized ontologies. Prominently, multi-dimensional DLs can be seen as convenient formalisms for capturing the dependency of knowledge on some contextual states built-in in the semantics, such as time points, epistemic states, computation states, etc. On the other hand, well-known frameworks for supporting context-aware ontology integration that allow to link knowledge from a set of classical DL ontologies without violating their local character have been developed, e.g., $\mathcal{E}$-Connections [55], Package-based DLs [18] or Distributed DLs [24]

The solutions proposed so far for management of contextualized knowledge are notoriously specialized in their scope, leaving open then the problem of formulating a broad and well-grounded theory of contexts within the DL paradigm. The aim of this chapter is to systematically develop a framework of two-dimensional *Description Logics of Context (DLC)*. Our proposal is inspired by J. McCarthy's theory of formalizing contexts [59], whose gist is to replace logical formulas φ, as the basic knowledge carriers, with assertions of the form $ist(\boldsymbol{c}, \varphi)$. Such assertions state that φ *is true* in $\boldsymbol{c}$, where $\boldsymbol{c}$ denotes an abstract first-order entity called a *context*. Further, contexts can be on their own described in a first-order language. For example, the formula:

$$ist(\boldsymbol{c}, Heart(a)) \wedge \boldsymbol{HumanAnatomy}(\boldsymbol{c})$$

states that the object a is a heart in a certain context $\boldsymbol{c}$ of type human anatomy. Formally, we interpret McCarthy's theory in terms of two-dimensional possible world semantics, where one dimension represents a usual object domain, while the other a (possibly infinite) domain of contexts. Thus, the notion of context is identified with that of *possible world*, which provides the former with a philosophically neutral, yet technically substantial reading, presupposed at the core of McCarthy's theory. Our investigation is two-fold: first, we investigate well-known modal DLs, which we see as *Simple Description Logics of Context*, as a natural and basic way of defining DLCs. Particularly, we concentrate on the traditional modal DLs $(\mathbf{K}_n)_{\mathcal{ALC}}, (\mathbf{DAlt}_n)_{\mathcal{ALC}}, (\mathbf{Alt}_n)_{\mathcal{ALC}}$ and $(\mathbf{D}_n)_{\mathcal{ALC}}$ in the case where modal operators are applied only to concepts and a global TBox is considered. Second, we extend simple DLCs with two interacting DL languages – the object and the context language – interpreted over the respective domains. These languages allow for explicit modeling of both: the (contextualized) object-level knowledge and the meta-level knowledge, i.e., descriptions of contexts as first-class citizens. Consequently, we define a whole family of *two-sorted*, *two-dimensional* DLs, comprising the most expressive DLC framework: *Expressive Description Logics of Context*, which are characterized by importing McCarthy's theory of formalizing contexts to its full extend.

Contributions: We introduce description logics of context, a family of two-dimensional DLs for representing and reasoning about contextualized knowledge. The proposed DLs are the result of a careful amalgamation of the principles of McCarthy's theory of formalizing contexts and the capabilities of two-dimensional DLs for capturing dynamic aspects of knowledge. Our main technical contribution is the study of the computational complexity of the satisfiability problem in DLCs. For simple DLCs, we provide a 2EXPTIME quasistate elimination algorithm deciding satisfiability, and a matching lower bound using a reduction of the word problem of an exponentially space bounded alternating Turing machine. Interestingly, we show that in some cases the transition from simple to expressive DLCs comes without an increment in the complexity. Particularly, we provide tight complexity bounds for expressive DLCs that range from NEXPTIME (for a restricted logic) to 2EXPTIME. Finally, we show several application scenarios of DLs of context.

Organization: The next section provides a comprehensive analysis of McCarthy's theory of contexts, and the relation of its principles with the modeling capabilities offered by two-dimensional DLs. In Section 5.3, we investigate the use of two-dimensional DLs as simple DLCs. In particular, we develop algorithms for satisfiability, and provide tight complexity bounds. Section 5.5 introduces expressive DLCs, extending simple DLCs with means to explicitly describe contexts and therefore fully complying with McCarthy's principles. Further, Section 5.6 presents a formal comparison between simple and expressive DLCs. In Section 5.7, we present algorithms for KB satisfiability based on type-like techniques for multi-dimensional modal logics, such as *quasistate elimination*. We moreover prove tight complexity bounds. Finally, Section 5.9 discusses the application of the DLC framework to a diversity of problems.

5.2 Towards the Design of Description Logics of Context

We begin our investigation on two-dimensional DLs of context by analyzing and motivating the use of two-dimensional DLs with product-like semantics to capture contextual aspects of knowledge. Particularly, we argue how these logics properly implement the principles of McCarthy's theory of formalizing contexts within the DL framework. Our first step towards this analysis is to introduce the principles of McCarthy's theory.

Over two decades ago John McCarthy introduced the AI community to a new paradigm of formalizing contexts in logic-based knowledge systems. This idea, presented in his Turing Award Lecture [59], was quickly picked up by others and by now has led to a significant body of work studying different implementations of the approach in a variety of formal frameworks and applications [30, 29, 28, 60, 42, 64]. The great appeal of McCarthy's paradigm stems from the simplicity and intuitiveness of the three major postulates it is based on:

1. Contexts are formal objects. More precisely, a context is anything that can be denoted by a first-order formula and used meaningfully in a statement of the form $ist(\boldsymbol{c}, \varphi)$, saying that formula φ *is true* (*ist*) in context $\boldsymbol{c}$, e.g., $ist(Hamlet,$ 'Hamlet is a prince.') [59, 60, 42, 30]. By adopting a strictly formal view on contexts, one can bypass unproductive debates on what they really are and instead take them as primitives underlying practical models of contextual reasoning.

2. Contexts have properties and can be described. As first-order objects, contexts can be in a natural way described in a first-order language [28, 42]. This allows for addressing them generically through quantified formulas such as $\forall x(\boldsymbol{C}(x) \rightarrow ist(x, \varphi))$, expressing that φ is true in every context of type $\boldsymbol{C}$, e.g., $\forall x(\textit{barbershop}(x) \rightarrow ist(x,$ 'Main service is a haircut.')).

3. Contexts are organized in relational structures. In the commonsense reasoning, contextual assumptions are dynamically and directionally altered [64, 30]. Contexts are entered and then exited, accessed from other contexts or transcended to broader ones. A simple way of handling their complex organization in formal systems is therefore by means of relational structures, which naturally support representation of diverse relationships and dynamic aspects in first-order domains. On the syntactic level, the use of such structures can be further reflected by permitting nested formulas of type $ist(\boldsymbol{c}, ist(\boldsymbol{d}, \varphi))$. For instance, $ist(France, ist(capital,$ 'The city river is Seine.')) implies that there exists certain relationship between $France$ and $capital$ such that 'The city river is Seine' is true in the latter context if accessed from (or seen from) the former, but not necessarily when accessed from any other arbitrary context.

Now that we have introduced McCarthy's principles, we argue about the convenience of two-dimensional DLs to design DLs of context that import McCarthy's theory of contexts into the DL framework. We start from the basic semantic considerations on contexts and further trace their impact on the selection of specific logical languages capturing them.

The first key step to importing McCarthy's theory into the DL framework is to faithfully reinterpret his three postulates on the model-theoretic grounds of DLs. Our main objective is then to find a form of extending classical DLs, such that contexts are treated as first-class citizens and therefore being able to reason with knowledge according to its contextual scope. Figure 5.1 shows a formal model, based on McCarthy's postulates, of an application domain supporting multiple contexts of representation, that is, each context supports the representation of a particular viewpoint on the domain.

We observe that the dynamic aspect that context-dependency adds to knowledge translates McCarthy's postulates in a two-dimensional model. Essentially, the *context-level* consists of context entities (postulate 1), which are possibly interlinked with certain relations (postulate 3) and described in a language containing individual names, concepts and relation names (postulate 2). For instance, in Figure 5.1, context $\boldsymbol{c}$ is of type $\boldsymbol{D}$ and is related to $\boldsymbol{d}$ through a relation of type $\boldsymbol{t}$. Instead of a unique one-dimensional global model of the object domain, we associate therefore

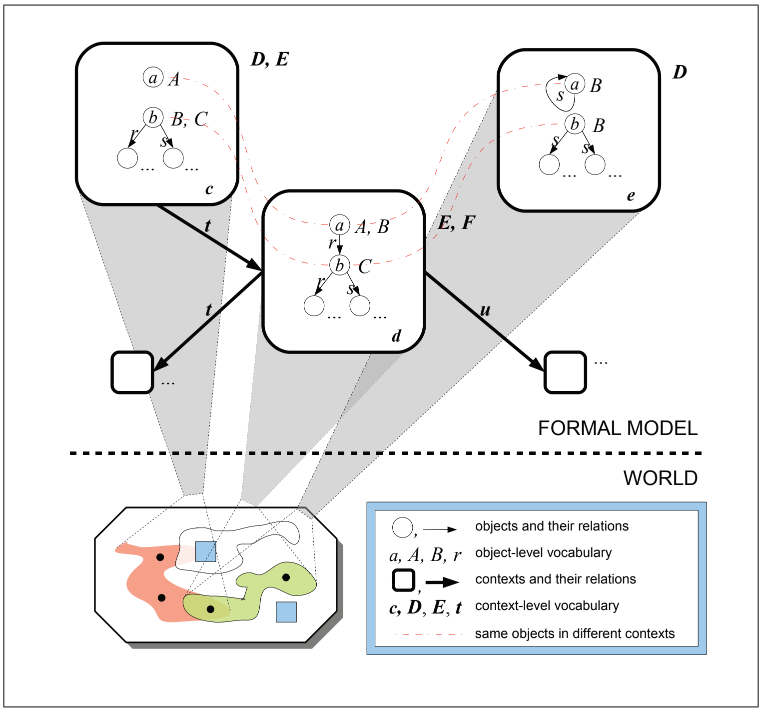

Figure 5.1: A formal domain model complying to McCarthy's postulates.

a local model of the object-domain with every context, giving rise then to a two-dimensional model. Intuitively, these local models reflect a specific viewpoint on the object domain, and they might then not necessarily cover the same fragment or aspect of the application domain and not necessarily use the same fragment of the object language for describing it. For instance, in Figure 5.1, objects a and b occur at the same time in contexts $\boldsymbol{c}, \boldsymbol{d}, \boldsymbol{e}$, but in each of them they are described differently and remain in different relations to other objects. From this analysis, one can straightforwardly realize that the context-level structures can be seen as Kripke frames, with possible worlds representing context entities and accessibility relations capturing relations between contexts. Consequently, we obtain a very clear-cut formal reading of the notion of *context* that coincides with the philosophically neutral and application-free notion of *context-as-formal-object* lying at the heart of McCarthy's theory.

CONTEXT = POSSIBLE WORLD

This view of contexts as possible worlds justifies the use of modal description logics with product-like semantics for capturing contextual aspects of knowledge. In particular, various contextualization and lifting operations, that is, context-sensitive transfers of knowledge between different contexts [59], can be naturally modeled by means of modal operators $\Diamond_i, \Box_i$. It is worth noting that the convenience of (one-dimensional) modal logics has already been

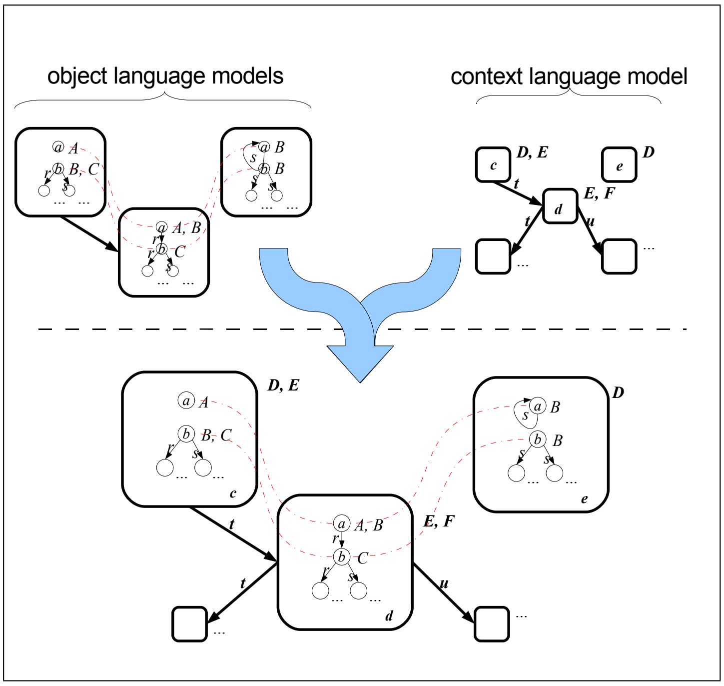

Figure 5.2: Combining models of two DLs.

exploited in the design of other context logics in the literature, such as [30, 28, 64]. There, however, contexts are usually identified with syntactic modalities rather than possible worlds in the semantics. As a consequence, these logics are restricted to the modeling of contextualized knowledge, lacking of support for the integration of independent knowledge sources. On the other hand, the interpretation of contexts as possible worlds allows for both: 'postulating' contexts implicitly in the representation, thus accounting for the inherently contextual character of the modeled knowledge, as well as accommodating standard DL ontologies in broader context-based systems, simply by seeing them as separate possible worlds.

From this analysis it is clear that modal DLs can appropriately serve as the underlying formalism for designing DLs of context. More precisely, as argued above, one can easily augment a DL language with modal '*contextualization*' operators for traversing the context dimension of the models and quantifying over the context entities. Note that, however, two-dimensional DLs do not conform with all of McCarthy's postulates. In particular, modal DLs do not offer a direct methodology for describing contexts *per se*. In other words, it is not possible to explicitly assert properties of the accessed contexts. For instance, to express global contextual dependencies, such as '*In every context of type human anatomy, a heart is a human organ*'. Intuitively, such functionality seems essential for obtaining a fine-grained contextualization machinery. The solution we propose is to extend modal DLs with a second DL language for describing the context dimension. In this way, we obtain *expressive* DLCs, which are *two-sorted*, *two-dimensional* DLs, where each sort of the language is interpreted over the respective dimension in the semantics. The two languages are suitably integrated on the syntactic and semantic level, so that their models can be eventually combined as presented in Figure 5.2. This style of combination is naturally fully compatible with the underlying modal DLs. In principle, the two-dimensional models of the object language are embedded in the models of the context language, where possible worlds are mapped on context individuals and accessibility relations are mapped on context roles.

5.3 Introducing Simple Description Logics of Context

We initiate our investigation on *Description Logics of Context (DLCs)* by studying modal description logics. In particular, as argued before, by identifying the notion of context with that of possible world, modal DLs prove well-suited for reasoning about contextual aspects of knowledge. In fact, the adequacy of modal DLs to capture contextual aspects of knowledge is supported by their faithful implementation within the DL framework of postulates 1 and 3 of McCarthy's theory of contexts (*cf.* Section 5.2). It is due to this partial implementation of McCarthy's theory that we see modal DLs as *simple* DLs of context.

In this chapter we focus on the investigation of the modal DLs $(\mathbf{DAlt}_n)_{\mathcal{ALC}}, (\mathbf{D}_n)_{\mathcal{ALC}}, (\mathbf{Alt}_n)_{\mathcal{ALC}}$ and $(\mathbf{K}_n)_{\mathcal{ALC}}$ in the case where modal operators are applied only to concepts, and a global TBox is considered. The choice of the modal components is based on previous research considering *classical* modal logics as context logics [30, 28, 64].

5.3.1 Syntax and Semantics

Definition 5.1. *Fix countably infinite disjoint sets* $\mathsf{N_C}$ *and* $\mathsf{N_R}$ *of* concept names *and* role names, *respectively.* Multi-modal $\mathcal{ML}_{\mathcal{ALC}}$-concepts *are formed by the following grammar:*

$$C ::= \top \mid A \mid \neg C \mid C \sqcap D \mid \exists r.C \mid \Diamond_i C$$

where A ranges over $\mathsf{N_C}$ and r ranges over $\mathsf{N_R}$, and $i \in (1, n)$ for some $n \in \mathbb{N}$.

Standard Boolean abbreviations are used, plus $\Box_i C$ to abbreviate $\neg\Diamond_i\neg C$.

We define $\mathcal{ML}_{\mathcal{ALC}}$ TBoxes as for classical DLs but using $\mathcal{ML}_{\mathcal{ALC}}$ concepts.

Definition 5.2. *An $\mathcal{ML}_{\mathcal{ALC}}$* TBox *is a finite set of CIs $C \sqsubseteq D$ with C, D $\mathcal{ML}_{\mathcal{ALC}}$ concepts.*

As an example consider the following $\mathcal{ML}_{\mathcal{ALC}}$ CI about the wine domain contextualized w.r.t. to geographic locations:

$$\mathsf{RedWine} \sqsubseteq \Box_{eu}\mathsf{PopDrink} \sqcap \Diamond_{amer}(\neg\mathsf{PopDrink}).$$

Intuitively, this axiom states that in all contexts accessible through the accessibility relation *eu* (Europe) red wine is a popular drink, and there is a possible context accessible through the accessibility relation *amer* (America) in which is not a popular drink.

In what follows we sometimes refer to the DL dimension as the *object dimension*, and to the modal dimension as the *context dimension* (e.g., Figure 5.3).

The possible world semantics of $\mathcal{ML}_{\mathcal{ALC}}$ is given in terms of *modal interpretations*, which associate with each possible world w a classical DL interpretation $\mathcal{I}_w$.

Definition 5.3. *A* modal interpretation $\mathfrak{I}$ *is a structure* $(\Delta, W, \{R_i\}_{i\in(1,n)}, \{\mathcal{I}_w\}_{w\in W})$ *where* W *is a non-empty set of* possible worlds, R_i *is an accessibility relation over* W *associated with the operator* $\Diamond_i$*, and for each* $w \in W$*,* $\mathcal{I}_w$ *is a classical DL-interpretation with domain* Δ*. The mapping* $\cdot^{\mathfrak{I},w}$ *is extended to complex concepts as follows:*

$$\top^{\mathfrak{I},w} = \Delta;$$

$$(\neg C)^{\mathfrak{I},w} = \{d \in \Delta \mid d \notin C^{\mathfrak{I},w}\};$$

$$(C \sqcap D)^{\mathfrak{I},w} = \{d \in \Delta \mid d \in C^{\mathfrak{I},w} \wedge d \in D^{\mathfrak{I},w}\};$$

$$(\exists r.C)^{\mathfrak{I},w} = \{d \in \Delta \mid \exists e \in \Delta : \ e \in C^{\mathfrak{I},w} \wedge (d,e) \in r^{\mathfrak{I},w}\};$$

$$(\Diamond_i C)^{\mathfrak{I},w} = \{d \in \Delta \mid \exists v \in W : wR_iv \wedge d \in C^{\mathfrak{I},v}\}.$$

We usually write $C^{\mathfrak{I},w}$ instead of $C^{\mathcal{I}_w}$; intuitively $d \in C^{\mathfrak{I},w}$ means that in the modal interpretation $\mathfrak{I}$, d is an instance of C in the world w. In the previous definition we make the *constant domain assumption*, i.e., each world shares the same domain Δ. Intuitively, this means that objects are not created or destroyed while making a transition from one world to another. This is the most general choice since expanding, decreasing and varying domains can all be simulated.

In this thesis, we are interested in studying the computational complexity of the concept satisfiability problem w.r.t. $\mathcal{ML}_{\mathcal{ALC}}$ TBoxes.

Definition 5.4. *A modal interpretation* $\mathfrak{I}$ *is a* model of a concept C *if* $C^{\mathfrak{I},w} \neq \emptyset$ *for some* $w \in W$*; it is a* model of a TBox $\mathcal{T}$ *if* $C^{\mathfrak{I},w} \subseteq D^{\mathfrak{I},w}$ *for all* $w \in W$ *and* $C \sqsubseteq D$ *in* $\mathcal{T}$*. A concept* C is satisfiable w.r.t. a TBox $\mathcal{T}$ *if there exists a common model of* C *and* $\mathcal{T}$*.*

Note that in the previous definition a TBox is regarded global in the sense that it must hold at each *world*.

Note that without further restrictions on the accessibility relations $\mathcal{ML}_{\mathcal{ALC}}$ corresponds to $(\mathbf{K}_n)_{\mathcal{ALC}}$, the combination of the modal logic $\mathbf{K}_n$ with $\mathcal{ALC}$. We also consider the modal DL $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$ which extends $\mathcal{ALC}$ with a set of functional modalities $\bigcirc_i$, that is, operators associated with accessibility relations R_i satisfying the properties of *seriality* $(\mathbf{D})$ and *quasi-functionality* $(\mathbf{Alt})$:

seriality :	$\forall w \in W \ \exists v \in W.(wR_iv)$
quasi-functionality :	$\forall w, v, u \in V(wR_iv \wedge wR_iu \rightarrow v = u)$

A natural question that emerges is whether we can allow for rigid roles in our logics. Alas, the answer is a negative one since the presence of rigid roles leads to undecidability.

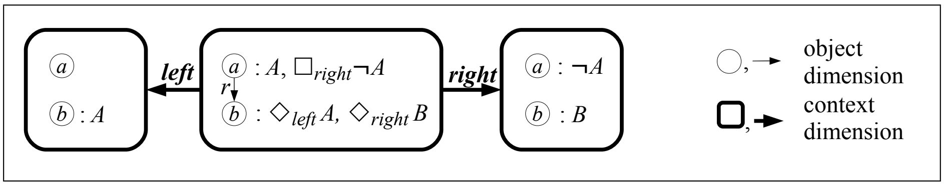

Figure 5.3: A context structure modeling concept $A \sqcap \Box_{right}\neg A \sqcap \exists r.(\Diamond_{left}A \sqcap \Diamond_{right}B)$.

Theorem 5.1 ([58])**.** *Concept satisfiability w.r.t. TBoxes* $\mathbf{DAlt}_{\mathcal{ALC}}$ *with a single rigid role is undecidable.*

Intuitively, the reason is that $\mathbf{DAlt}_{\mathcal{ALC}}$ corresponds to the fragment of LTL$_{\mathcal{ALC}}$ with the *next-time* operator, which is enough to construct a usual encoding of the undecidable $\mathbb{N}\times\mathbb{N}$-*tiling problem* [58]. Now, it is not hard to see that the variation of the modal component in our logics has no impact on the previous result.

Proposition 5.2. *Concept satisfiability w.r.t. TBoxes is polynomially reducible between the following logics* (*where* $\mapsto$ *means* reduces to)*:*

$$(\mathbf{DAlt}_n)_{\mathcal{ALC}} \mapsto \{(\mathbf{D}_n)_{\mathcal{ALC}}, (\mathbf{Alt}_n)_{\mathcal{ALC}}\} \mapsto (\mathbf{K}_n)_{\mathcal{ALC}}.$$

Proof. If $(C, \mathcal{T})$ is an instance of the concept satisfiability problem w.r.t. TBoxes in some lefthandside logic, then one can decide it in the righthandside logic by applying simple transformations of C and $\mathcal{T}$, encoding the missing conditions:

– *Quasi-functionality:* Assume, w.l.o.g., that $C = \mathsf{nnf}(C)$, where nnf stands for *negation normal form*, and $\mathcal{T} = \{\top \sqsubseteq \mathsf{nnf}(C_{\mathcal{T}})\}$. Let C' and $C'_{\mathcal{T}}$ be the result of replacing every subconcept $\Diamond_i B$ occurring in C and $C_{\mathcal{T}}$, respectively, with $(\Diamond_i\top)\sqcap(\Box_i B)$. Then, $(C, \mathcal{T})$ is satisfiable on a quasi-functional frame *iff* $(C', \{\top \sqsubseteq C'_{\mathcal{T}}\})$ is satisfiable.

– *Seriality:* Let $\mathcal{T}' = \mathcal{T} \cup \{\top \sqsubseteq \Diamond_i\top \mid 1 \leq i \leq n\}$, where n is the number of all modalities occurring in $\mathcal{T}$ and C. Then, $(C, \mathcal{T})$ is satisfiable on a serial frame *iff* $(C, \mathcal{T}')$ is satisfiable.

❏

Theorem 5.1 together with Proposition 5.2 immediately entails the following:

Lemma 5.3. *For any* $\mathcal{ML} \in \{\mathbf{DAlt}_n, \mathbf{D}_n, \mathbf{Alt}_n, \mathbf{K}_n\}$*, concept satisfiability in* $\mathcal{ML}_{\mathcal{ALC}}$ *w.r.t. TBoxes with a single rigid role is undecidable.*

5.4 Reasoning in Simple Description Logics of Context

This section begins our investigation on the computational complexity of the simple DLs of context: $\mathcal{ML}_{\mathcal{ALC}}$ for $\mathcal{ML} \in \{\mathbf{DAlt}_n, \mathbf{D}_n, \mathbf{Alt}_n, \mathbf{K}_n\}$. We present algorithms for concept satisfiability w.r.t. TBoxes based on quasistate elimination techniques commonly used to devise decision procedures for multi-dimensional modal logics. This decision procedure yields a 2ExpTime upper bound. Surprisingly, we demonstrate a matching lower bound; note that the computational complexity of two-dimensional DLs allowing for such limited interaction of the component logics usually is not higher than for the component logics. We prove the lower bound by a reduction of the word problem of exponentially space bounded alternating Turing machines.

5.4.1 An Algorithm for Concept Satisfiability w.r.t. TBoxes for $(\mathrm{K}_n)_{\mathcal{ALC}}$

We next present a 2ExpTime algorithm for satisfiability in $(\mathbf{K}_n)_{\mathcal{ALC}}$[1]. Our algorithm implements a variant of the *quasistate elimination technique* [38]. The main idea is to abstract from the domains W and Δ and consider only a *finite* (double exponential) number of quasistates representing possible worlds inhabited by a finite number of possible types. We then iteratively eliminate all quasistates that do not satisfy necessary conditions.

Let us fix a concept C and a TBox $\mathcal{T}$ formulated in $(\mathbf{K}_n)_{\mathcal{ALC}}$. We assume w.l.o.g. that $\mathcal{T}$ is of the form $\{\top \sqsubseteq C_{\mathcal{T}}\}$, and use $\mathsf{cl}(\mathcal{T})$ to denote the set of concepts that occur in $\mathcal{T}$, closed under negation and subconcepts.

Definition 5.5. *A* type for $\mathcal{T}$ *is a set* $t \subseteq \mathsf{cl}(\mathcal{T})$ *satisfying the following conditions:*

- $C \in t$ *iff* $\neg C \notin t$*, for all* $C \in \mathsf{cl}(\mathcal{T})$,
- $C \sqcap D \in t$ *iff* $\{C, D\} \subseteq t$*, for all* $C \sqcap D \in \mathsf{cl}(\mathcal{T})$,
- $C_{\mathcal{T}} \in t$.

We denote by $\mathtt{tp}(\mathcal{T})$ be the set of all types for $\mathcal{T}$.

The next notion establishes when two types are compatible according to a transition k: we say that two types $t, t' \in \mathtt{tp}(\mathcal{T})$ are *k-compatible* for $k \in (1, n)$ if $\{\neg C \mid \neg\Diamond_k C \in t\} \subseteq t'$.

Definition 5.6. *A* quasistate for $\mathcal{T}$ *is a set* $q \subseteq \mathtt{tp}(\mathcal{T})$*, such that for every* $t \in q$ *and every* $\exists r.D \in \mathsf{cl}(\mathcal{T})$ *the following holds:*

$(\mathbf{QS})$ *if* $\exists r.D \in t$ *then there is a type* $t' \in q$ *such that* $\{D\} \cup \{\neg C \mid \neg\exists r.C \in t\} \subseteq t'$.

We denote by $\mathsf{qs}(\mathcal{T})$ the set of all quasistates for $\mathcal{T}$.

We extend the notion of k-compatibility to quasistates by ensuring that all their types are k-compatible: we say that two quasistates $q, q' \in \mathsf{qs}(\mathcal{T})$ are *k-compatible* if there exists a pair of functions $f : q \to q'$ and $g : q' \to q$ such that, for every $t \in q$ and $t' \in q'$, t and $f(t)$ are k-compatible and $g(t')$ and t' are k-compatible.

The following definition allows us to identify those quasistates that, intuitively, can occur in a model of $\mathcal{T}$.

[1] Mind that the NExpTime-completeness result for concept satisfiability in $\mathbf{K}_{\mathcal{ALC}}$ [38, Theorem 15.15] applies to $\mathcal{ALC}$ with a single pair of $\mathbf{K}$ operators, full booleans on modalized formulas and no global TBoxes.

Algorithm 2: $(\mathbf{K}_n)_{\mathcal{ALC}}$ SATISFIABILITY

Input: Concept C, TBox $\mathcal{T}$ formulated in $(\mathbf{K}_n)_{\mathcal{ALC}}$
Initialize: $i := 0$; $S_0 := \mathsf{qs}(\mathcal{T})$

repeat
$S_{i+1} := \{q \in S_i \mid q \text{ is realizable in } S_i\}$
until $S_i = S_{i+1}$

if exists $q \in S_i$ and a $t \in q$ such that $C \in t$, return *satisfiable*
otherwise, return *unsatisfiable*

Definition 5.7. *Let q be a quasistate for $\mathcal{T}$, such that $\Diamond_k C \in t$ for some $t \in q$. A quasistate q' is a* witness *for the triple $(\Diamond_k C, t, q)$* if *q and q' are k-compatible and there is $t' \in q'$ such that $C \in t'$.*

Now, we can define the elimination condition of our algorithm, that is, we will eliminate those quasistates that cannot occur in any model of $\mathcal{T}$.

Definition 5.8. *Let $S \subseteq \mathsf{qs}(\mathcal{T})$ be a set of quasistates for $\mathcal{T}$. A quasistate q for $\mathcal{T}$ is realizable in S if the following condition is satisfied: for every $t \in q$ and every $\Diamond_k D \in \mathsf{cl}(\mathcal{T})$,*

if $\Diamond_k D \in t$ then there is a witness for $(\Diamond_k D, t, q)$ in S.

Algorithm 2 above implements a quasistate elimination procedure for deciding satisfiability for $(\mathbf{K}_n)_{\mathcal{ALC}}$.

Lemma 5.4. *Algorithm 2 returns* 'satisfiable' iff *C is satisfiable w.r.t. $\mathcal{T}$.*

Proof. ($\Rightarrow$) Let S_j be the final set computed by Algorithm 2. We construct a model $\mathfrak{I} = (\Delta, W, \{R_i\}_{i\in(1,n)}, \{\mathcal{I}_w\}_{w\in W})$ of C and $\mathcal{T}$ as follows. First, define the $\mathbf{K}_n$*-frame* $(W, \{R_i\}_{i\in(1,n)})$. In particular, we define sequences $W_0, W_1 \ldots, R_i^0, R_i^1, \ldots$ and partial mappings $\pi : W_i \to S_j$. Our desired sets W and R_i are obtained in the limit. To start the construction of the $\mathbf{K}_n$-frame, set

– $W_0 = \{w_0\}$, $\pi_0(w_0) = q$ such that there exists a t in q with $C \in t$, $R_i^0 = \emptyset$, for all $i \in (1, n)$.

For the inductive step, we start by setting $W_i = W_{i-1}, R_j^i = R_j^{i-1}$ and $\pi_{i-1} = \pi_i$, and then proceed as follows:

(I) For every $1 \leq k \leq n$, every $w \in W_i$, if $\Diamond_k C \in t$ for some $t \in \pi_i(w)$, then

– add w' to W_i, (w, w') to R_k^i, and set $\pi_i(w') = q'$ such that q' is a witness for $(\Diamond_k C, t, q)$.

Finally, set $W = \bigcup_{i\geq 0} W_i$ and $R_k = \bigcup_{i\geq 0} R_k^i$.
We continue our definition of $\mathfrak{I}$ by defining a run through W. A *run* ρ is a *choice function* which for every $w \in W$ selects a type $\rho(w) \in \pi(w)$. A set of runs $\mathfrak{R}$ is *coherent* if the following conditions are satisfied:

(a) for every $w \in W$ and every $t \in \pi(w)$, there is a run $\rho \in \mathfrak{R}$ such that $\rho(w) = t$;

(b) for every $\rho \in \mathfrak{R}$, $1 \leq k \leq n$ and $(w, w') \in R_k$, it holds that $\rho(w)$ and $\rho(w')$ are k-compatible;

(c) for every $\rho \in \mathfrak{R}$, $\Diamond_k D \in \mathsf{cl}(\mathcal{T})$ and $w \in W$, if $\Diamond_k D \in \rho(w)$ then there exists $w' \in W$ such that $(w, w') \in R_k$ and $D \in \rho(w')$.

Finally, set $\Delta = \mathfrak{R}$ with $\mathfrak{R}$ a coherent set of runs through W. It remains to define the interpretation function for concept and role names:

$$A^{\mathfrak{I},w} = \{\rho \in \Delta \mid A \in \rho(w)\};$$

$$r^{\mathfrak{I},w} = \{(\rho, \rho') \in \Delta\times\Delta \mid \exists r.D \in \rho(w) \text{ implies } \{D\}\cup\{\neg C \mid \neg\exists r.C \in \rho(w)\} \subseteq \rho'(w)\}.$$

Claim For each $C \in \mathsf{cl}(\mathcal{T})$, $\rho \in \Delta$, $w \in W$

$$\rho \in C^{\mathfrak{I},w} \text{ iff } C \in \rho(w)$$

Proof of the claim: The proof is by induction on the structure of C. The induction start, where C is a concept name is immediate by definition of $\mathfrak{I}$. For the induction step, we distinguish the following cases:

– $C = \neg D$ “if:” $\rho \in \neg D^{\mathfrak{I},w}$, that is, $\rho \notin D^{\mathfrak{I},w}$. Now, by I.H., $D \notin \rho(w)$. Then, by definition of type, $\neg D \in \rho(w)$. “only if” $\neg D \in \rho(w)$. By definition of type, $D \notin \rho(w)$. Now, by I.H., $\rho \notin D^{\mathfrak{I},w}$. Therefore, by semantics, $\rho \in (\neg D)^{\mathfrak{I},w}$.

– $C = D_1 \sqcap D_2$ “if:” $\rho \in (D_1 \sqcap D_2)^{\mathfrak{I},w}$, that is, $\rho \in D_1^{\mathfrak{I},w}$ and $\rho \in D_2^{\mathfrak{I},w}$. Now, by I.H., $D_1 \in \rho(w)$ and $D_2 \in \rho(w)$. Therefore, by definition of type, $D_1 \sqcap D_2 \in \rho(w)$. “only if:” $D_1 \sqcap D_2 \in \rho(w)$, then by definition of type, $D_1 \in \rho(w)$ and $D_2 \in \rho(w)$, Now, by I.H., $\rho \in D^{\mathfrak{I},w}$ and $\rho \in D_2^{\mathfrak{I},w}$. Therefore, by semantics, $\rho \in (D_1 \sqcap D_2)^{\mathfrak{I},w}$.

– $C{=}\exists r.D$ “if:” $\rho \in (\exists r.D)^{\mathfrak{I},w}$, that is, there exists a ρ' such that $(\rho, \rho') \in r^{\mathfrak{I},w}$ and $\rho' \in D^{\mathfrak{I},w}$. Now, by I.H., we know that $D \in \rho'(w)$. Therefore, by definition of $r^{\mathfrak{I},w}$ and I.H., $\exists r.D \in \rho(w)$.

  “only if:” $\exists r.D \in \rho(w)$. By construction, $\rho(w) = t \in \pi(w)$. Now, by condition (**QS**) of definition of quasistate, there exists a $t' \in \pi(w)$ such that $\{D\} \cup \{\neg C \mid \neg\exists r.C \in t\} \subseteq t'$. Furthermore, by (c), there is a ρ' such that $\rho'(w) = t'$. Then, by I.H., $\rho' \in D^{\mathfrak{I},w'}$, and moreover, by definition of $r^{\mathfrak{I},w}$, $(\rho, \rho') \in r^{\mathfrak{I},w}$. Therefore, by semantics, $\rho \in (\exists r.D)^{\mathfrak{I},w}$.

– $C = \Diamond_k D$. "if:" $\rho \in (\Diamond_k D)^{\mathfrak{I},w}$, that is, there exists a $w' \in W$ such that $wR_k w'$ and $\rho \in D^{\mathfrak{I},w'}$. By construction (b), $\rho(w)$ and $\rho(w')$ are k-compatible and, by (**I**), $\pi(w)$ and $\pi(w')$ are k-compatible. Moreover, by I.H., $D \in \rho(w')$. Hence, by definition of witness, $\pi(w')$ is a witness of $(\Diamond_k D, \rho(w), \pi(w))$. Therefore, $\Diamond_k D \in \rho(w)$.

  "only if:" $\Diamond_k D \in \rho(w)$. Note that $\rho(w) = t \in \pi(w)$. Now, by construction (**I**), there is a $w' \in W$ such that $\pi(w') \in \mathsf{qs}(\mathcal{T})$ is a witness of $(\Diamond_k D, t, \pi(w))$. Hence, $wR_k w'$. Now, by condition (c), $\rho(w)$ and $\rho(w')$ are k-compatible, and moreover by I.H., $\rho \in D^{\mathfrak{I},w'}$. Therefore, by semantics, $\rho \in (\Diamond_k D)^{\mathfrak{I},w}$.

Note that by definition of type, $C_{\mathcal{T}} \in \rho(w)$ for all $\rho \in \Delta$ and $w \in W$. Now, by the previous claim $\rho \in (C_{\mathcal{T}})^{\mathfrak{I},w}$ for all $\rho \in \Delta$ and $w \in W$. Hence, $\mathfrak{I}$ is a model of $\mathcal{T}$. Moreover, by assumption, there is a $\rho(w) \in \pi(w)$ for some $\rho \in \mathfrak{R}$ and $w \in W$ such that $C \in \rho(w)$. Thus, by the previous claim, $\rho \in C^{\mathfrak{I},w}$. Therefore, $\mathfrak{I}$ is a model of C and $\mathcal{T}$.

($\Leftarrow$) Let $\mathfrak{I} = (\Delta, W, \{R_i\}_{i\in(1,n)}, \{\mathcal{I}_w\}_{w\in W})$ be a model of $\mathcal{T}$ and C. We next define S such that there is a subset S of S_0 such that none of its elements can be eliminated by Algorithm 2 and for some $q \in S$ there is a $t \in q$ such that $C \in t$.

For every $d \in \Delta$ and $w \in W$ we set:

$$\mathsf{tp}(d, w) := \{C \in \mathsf{cl}(\mathcal{T}) \mid d \in C^{\mathfrak{I},w}\}.$$

Further, we associate with every $w \in W$ the set of types

$$q_w = \{\mathsf{tp}(d, w) \mid d \in \Delta\}.$$

It is clear that each such q_w is a quasistate since all existential restrictions are witnessed. Finally, we fix

$$S = \{q_w \mid w \in W\}.$$

It is clear from the definition of S that each element of S is realizable: assume $\Diamond_k C \in t$ for some arbitrary t in $q_w \in S$. By construction, we know that $t = \mathsf{tp}(d, w)$ for some $d \in \Delta$, and moreover we know that $d \in (\Diamond_k C)^{\mathfrak{I},w}$. Now, by semantics, $d \in C^{\mathfrak{I},w'}$ for some $w' \in W$. $(w, w') \in R_k$. Now by definition of S, there is a $q_{w'}$ such that there is a $t' = \mathsf{tp}(d, w')$. Then, by definition of $\mathsf{tp}(d, w')$, $C \in t'$. One can argue analogously to see that q_w and $q_{w'}$ are k-compatible. Therefore, S is realizable.

Now one can directly see by induction on i that $S \subseteq S_i$ for $0 \leq i \leq m$, where $S_0, \ldots, S_m$ is the sequence computed by Algorithm 2, that is, none of the elements of S is deleted in any of the iterations of Algorithm 2.

Since $\mathfrak{I}$ is a model of C, one of the elements of S contains a type t, such that $C \in t$, and all types in all quasistates contain the TBox concept $C_{\mathcal{T}}$. Thus, the Algorithm 2 returns *satisfiable*. ❑

Now, it remains to show that Algorithm 2 runs in double exponential time. First, note that the number of types is in $O(2^{\mathsf{poly}(|\mathcal{T}|)})$ and the number of quasistates is in $O(2^{2^{\mathsf{poly}(|\mathcal{T}|)}})$. In the worst case, in order to verify whether the elimination criterion applies to a quasistate at a given stage of the run of the Algorithm 2, it is necessary to compare each of its types against all types from the remaining quasistates, where each comparison can be performed in the polynomial time. Thus the whole algorithm cannot take more than $((2^{2^{\mathsf{poly}(|\mathcal{T}|)}} \cdot 2^{\mathsf{poly}(|\mathcal{T})|}) \cdot 2^{\mathsf{poly}(|\mathcal{T}|)}) \cdot 2^{2^{\mathsf{poly}(|\mathcal{T}|)}}$ steps in total to terminate, and thus remains clearly in $O(2^{2^{\mathsf{poly}(|\mathcal{T}|)}})$.

Theorem 5.5. *Concept satisfiability w.r.t. TBoxes for* $(\mathbf{K}_n)_{\mathcal{ALC}}$ *is in* 2EXPTIME.

5.4.2 A 2EXPTIME Lower Bound for Concept Satisfiability w.r.t. TBoxes for $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$

In the light of the results on two-dimensional DLs found in the literature, one could expect the 2EXPTIME upper bound presented above not to be optimal. This is the case, since $(\mathbf{K}_n)_{\mathcal{ALC}}$ with modalities applied only to concepts is very similar to the fusion of $\mathbf{K}_n$ and $\mathcal{ALC}$, which in principle should make possible to devise an EXPTIME algorithm. However, the following example shows that there are important differences between $(\mathbf{K}_n)_{\mathcal{ALC}}$ and the fusion of its component logics. Consider the $(\mathbf{K}_n)_{\mathcal{ALC}}$ concept in the l.h.s. and its reduction to a fusion language in the r.h.s.

$$(\dagger)\ \Diamond_i C \sqcap \exists r.\Box_i \bot \qquad (\ddagger)\ \exists \mathsf{succ}_i.C \sqcap \exists r.\forall \mathsf{succ}_i.\bot$$

Note that although $(\dagger)$ clearly does not have a model, its reduction $(\ddagger)$ to a fusion language, where modal operators are translated to restrictions on *fresh* $\mathcal{ALC}$ roles succ_i, is satisfiable. The reason is that while in the former case the information about the structure of the $\mathbf{K}_n$-frame is global for all individuals, in the latter it becomes local. The r-successor in $(\ddagger)$ is simply not '*aware*' that it should actually have a succ_i-successor.[2] This effect, amplified by the presence of global TBoxes (which can enforce infinite $\mathbf{K}$-trees), makes the reasoning harder.

We next prove a 2EXPTIME lower bound for concept satisfiability w.r.t. TBoxes in $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$ by a reduction of the word problem of an exponentially space bounded alternating Turing machine.

Theorem 5.6. *Concept satisfiability w.r.t. TBoxes for* $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$ *is* 2EXPTIME*-hard.*

Proof. This proof is by a reduction of the word problem of an exponentially space bounded alternating Turing machine (*cf.* Section 3.5.3). Let $\mathcal{M} = (Q, \Sigma, \Gamma, q_0, \delta)$ be such an ATM, with a 2EXPTIME-hard word problem, and $w \in \Sigma^*$ be the input of length n. Our aim is to construct in polynomial time a TBox $\mathcal{T}_{\mathcal{M},w}$ and a concept $C_{\mathcal{M},w}$ such that $C_{\mathcal{M},w}$ is satisfiable w.r.t. $\mathcal{T}_{\mathcal{M},w}$ *iff* $\mathcal{M}$ accepts w.

The core idea of the reduction is to use separate $\mathbf{DAlt}_n$ modalities for representing symbols of the alphabet. By isolating then specific fragments of $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$ tree models we can thus embed the syntactic structure of an ATM computation tree (see Figure 5.4) within $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$

[2] Demonstrating the corresponding phenomenon in $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$ is not that straightforward due to the seriality condition, as then the global information concerns only the existence of succ_i-predecessors. Thus, one needs role inverses in the fusion language to observe the loss of such information. This is in general explained by the fact that fusions correspond to expanding domains instead of constant ones.

models. At the same time, using special counting concepts over the modal dimension, we align the succeeding configurations, ensuring then the satisfaction of the constraints of the respective ATM transitions (see Figure 5.5). We use the following signature:

- concept names W_a, S_a, for every $a \in \Gamma$;
- concept names Q_q, for every $q \in Q$;
- concept names $X_0 \dots X_{n-1}$, $U_0 \dots U_{n-1}$, $C_0 \dots C_{n-1}$, $C'_0 \dots C'_{n-1}$ for encoding exponential counters;
- concept names $M_{q,a,m}, N_{q,a,m}$ for every $(q, a, m) \in \Theta$, where

$$\Theta = \{(q, a, m) \mid (q', b, q, a, m) \in \delta \text{ for any } b \in \Gamma \text{ and } q' \in Q\};$$

- auxiliary concept names ZeroTape, EndTape, ZeroHead, ZeroHead′, ZeroCell related with the value zero and 2^n-1 of an associated counter, and a marker concept name Tape.

We moreover use for every $a \in \Gamma$ modal operators $\bigcirc_a$, and the following abbreviations (for any concept B):

$$\Box B = \bigsqcap_{a \in \Gamma} \bigcirc_a B,$$
$$\Diamond B = \bigsqcup_{a \in \Gamma} \bigcirc_a B,$$

Throughout the reduction we use several counters, consisting of a number of inclusions of a total polynomial size, which allow to identify worlds on the branches at a fixed distance 2^n. Constraints (5.1)-(5.2) implement an exemplary *counter*, based on atomic concepts X_i, which simulate the bits of a number in binary.

For every $0 \le j < i < n$,

$$\begin{aligned} \neg X_i \sqcap \neg X_j &\sqsubseteq \Box \neg X_i, \\ X_i \sqcap \neg X_j &\sqsubseteq \Box X_i, \end{aligned} \tag{5.1}$$

For every $0 \le j < n$

$$\begin{aligned} \neg X_j \sqcap X_{j-1} \sqcap \ldots \sqcap X_1 &\sqsubseteq \Box X_j, \\ X_j \sqcap X_{j-1} \sqcap \ldots \sqcap X_1 &\sqsubseteq \Box \neg X_j. \end{aligned} \tag{5.2}$$

We use the abbreviations Zero and End to denote, respectively:

$$X = \bigsqcap_{j=0}^{n-1} \neg X_j \qquad X = \bigsqcap_{j=0}^{n-1} X_j \tag{5.3}$$

We use the counter X to define constraints which encode a single tape on a branch of a model. We use ZeroTape and EndTape to refer to the abbreviations introduce above in (5.3). In (5.4) we define the beginning and the end of such a tape, while with (5.5)-(5.6) we ensure that there is a unique path connecting the two. We will consider such a tape path as determining the content of the tape, as presented in Figure 5.4. In fact, in our models we will need only one individual

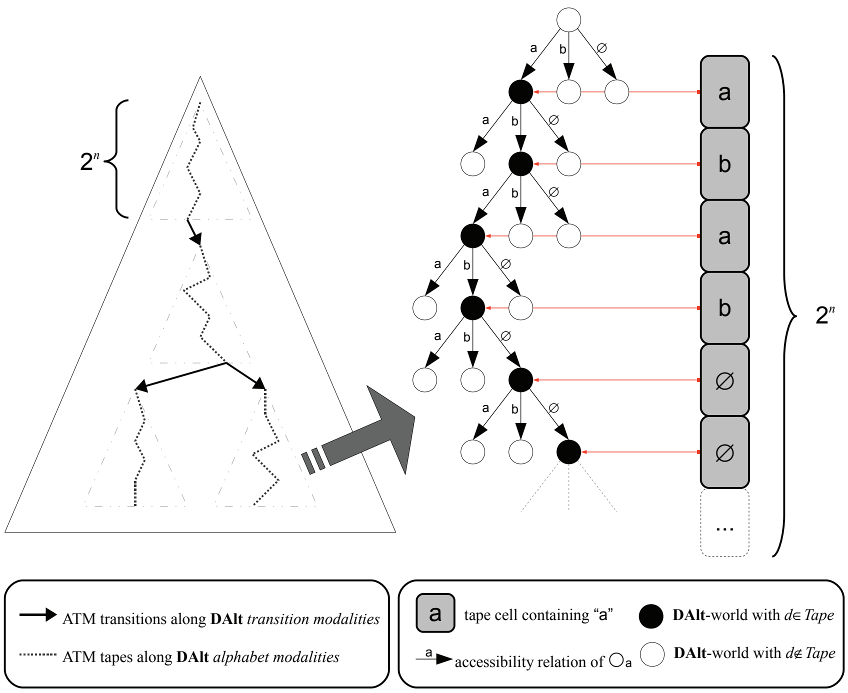

Figure 5.4: Embedding of ATM computation trees (left) and ATM tapes (right) in $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$-tree-models.

which will single out the whole structure of the ATM tree. Constraint (5.7) ensures that the blank symbol is followed only by blank symbols on the tape.

$$\mathsf{BeginTape} \equiv \mathsf{Tape} \sqcap \mathsf{ZeroTape} \qquad \mathsf{FinishTape} \equiv \mathsf{Tape} \sqcap \mathsf{EndTape}, \tag{5.4}$$

$$\mathsf{Tape} \sqsubseteq \Diamond\mathsf{Tape} \qquad \Diamond\mathsf{Tape} \sqsubseteq \mathsf{Tape}, \tag{5.5}$$

$$\neg\mathsf{FinishTape} \sqcap \bigcirc_a \mathsf{Tape} \sqcap \bigcirc_b \mathsf{Tape} \sqsubseteq \bot, \text{ for every } a \neq b \in \Gamma, \tag{5.6}$$

$$\bigcirc_{\llcorner\!\lrcorner}(\mathsf{Tape} \sqcap \bigcirc_a \mathsf{Tape}) \sqsubseteq \bot, \text{ for every } a \neq \llcorner\!\lrcorner \in \Gamma. \tag{5.7}$$

We introduce a counter C capturing the position of the head in the tape. We use ZeroHead, as defined in (5.3). We further use Q_q concepts to denote the current state and the position of the head, and concepts $M_{q,a,m} \in \Theta$ to carry this information about the following transitions. Information about the transitions is generated depending on whether the state is universal (5.8) or existential (5.9) and then carried to the end of the tape (5.10).

For every $a \in \Gamma, q \in Q_\forall$:

$$\bigcirc_a(Q_q \sqcap \mathsf{Tape}) \sqsubseteq \bigcirc_a\Big(\bigsqcap_{(q'b'm) \in \delta(q,a)} M_{q',b,m} \sqcap \mathsf{ZeroHead}\Big) \tag{5.8}$$

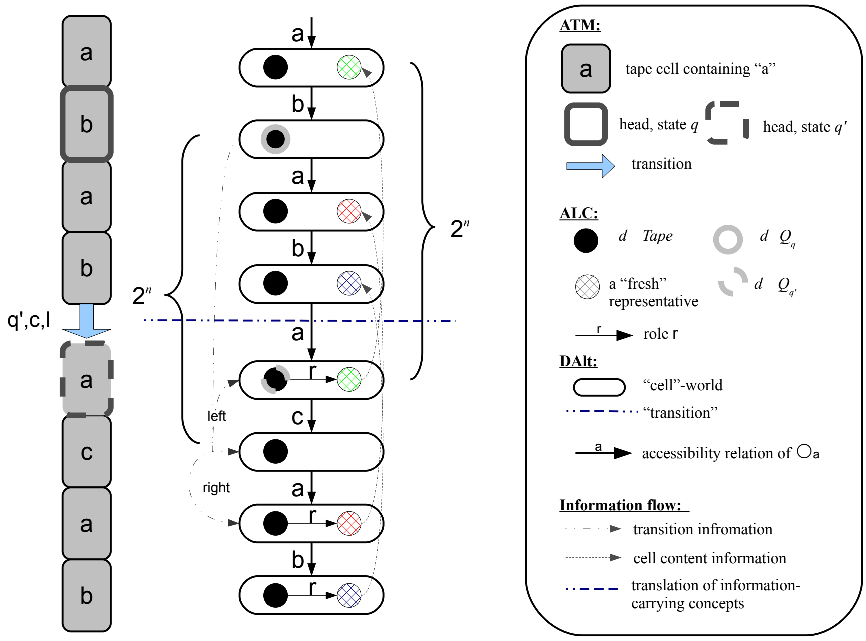

Figure 5.5: A transition between succeeding configurations in $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$-tree-models for $n = 2$ and $(q', c, l) \in \delta(q, b)$.

For every $a \in \Gamma, q \in Q_\exists$:

$$\bigcirc_a(Q_q \sqcap \mathsf{Tape}) \sqsubseteq \bigcirc_a\Big(\bigsqcup_{(q'b'm)\in\delta(q,a)} M_{q',b,m} \sqcap \mathsf{ZeroHead}\Big) \tag{5.9}$$

$$\neg\mathsf{FinishTape} \sqcap M_{q,a,m} \sqsubseteq \Box M_{q,a,m} \tag{5.10}$$

To avoid potential clashes with the information generated on the succeeding configurations, we create copies $N_{q,a,m}$ for all concepts $M_{q,a,m}$, which continue to carry their information over the new configuration (5.11)-(5.12). Further, we introduce an auxiliary counter C', which proceeds with the counting exactly from the point where the previous head counter (C) terminated (5.13).

For every $(q, a, m) \in \Theta$:

$$M_{q,a,m} \sqcap \mathsf{FinishTape} \sqsubseteq \Diamond(N_{q,a,m} \sqcap \mathsf{BeginTape}) \tag{5.11}$$

$$\mathsf{FinishTape} \sqcap N_{q,a,m} \sqsubseteq \Box N_{q,a,m}, \tag{5.12}$$

$$\mathsf{FinishTape} \sqcap C_i \sqsubseteq C'_i \qquad \mathsf{FinishTape} \sqcap \neg C_i \sqsubseteq \neg C'_i \tag{5.13}$$

The necessary changes in the configuration are imposed through constraints (5.14)-(5.17), which place the head in the appropriate position, marking it with the new state concept, and force the old position to be overwritten with the new symbol. The inclusions (5.18)-(5.19) ensure that the transition does not push the head beyond the tape.

$$\bigcirc_b(N_{q,a,m} \sqcap \mathsf{Tape} \sqcap \mathsf{ZeroHead}) \sqsubseteq \bot \tag{5.14}$$

For every $(q, a, n) \in \Theta$:

$$N_{q,a,n} \sqcap \mathsf{Tape} \sqcap \mathsf{ZeroHead}' \sqsubseteq Q_q \tag{5.15}$$

For every $(q, a, r) \in \Theta$:

$$N_{q,a,r} \sqcap \mathsf{Tape} \sqcap \mathsf{ZeroHead}' \sqsubseteq \Box Q_q \tag{5.16}$$

For every $(q, a, l) \in \Theta$:

$$N_{q,a,r} \sqcap \mathsf{Tape} \sqcap \mathsf{EndHead}' \sqsubseteq Q_q \tag{5.17}$$

$$\mathsf{ZeroHead} \sqcap \mathsf{BeginTape} \sqsubseteq \neg N_{q,a,l}, \text{ for every } q \in Q, a \in \Gamma, \tag{5.18}$$

$$\mathsf{ZeroHead} \sqcap \mathsf{FinishTape} \sqsubseteq \neg N_{q,a,r}, \text{ for every } q \in Q, a \in \Gamma. \tag{5.19}$$

Now we transfer the information about the content of the cells which are not meant to change during the transition. This information is carried by newly generated ‘representatives’, i.e., new r-successors of the individual instantiating Tape. Observe that since our models are tree-shaped, it follows that whenever a representative at a i-th position reaches the $2^n - 1$ ancestor world, it is exactly the world representing the i-th cell at the previous configuration. This enables to align the content of the two versions.

For each $a \in \Gamma$ we introduce two concept names W_a, S_a storing the content of a tape cell, whose interpretation is propagated to the previous configuration and aligned at the end of the configuration (5.20). We further use a counter U representing the position of a cell; we use accordingly ZeroCell. Constraint (5.21) generates a representative of each cell, and equips it with the corresponding concept W describing the cell’s content. Once this information arrives to the previous version of that cell we prevent the cells from having different content (5.22). We standardly ensure that the representative of the cell is uniquely associated with an alphabet letter (5.23).

For every $a \in \Gamma$

$$\neg\mathsf{FinishTape} \sqcap \Diamond W_a \sqsubseteq W_a \qquad \neg\mathsf{FinishTape} \sqcap \Diamond S_a \sqsubseteq S_a \qquad \mathsf{FinishTape} \sqcap \Diamond W_a \sqsubseteq S_a \tag{5.20}$$

$$\bigcirc_a(\mathsf{Tape} \sqcap \neg\mathsf{ZeroHead}) \sqsubseteq \bigcirc_a \exists r.(\mathsf{ZeroCell} \sqcap W_a), \tag{5.21}$$

For every $b \neq a \in \Gamma$

$$\bigcirc_a(S_b \sqcap \mathsf{ZeroCell}) \sqsubseteq \bot, \tag{5.22}$$

$$W_a \sqcap W_b \sqsubseteq \bot \qquad S_a \sqcap S_b \sqsubseteq \bot \tag{5.23}$$

Finally, it suffices to ensure that nowhere in the model is the rejecting state satisfied.

$$\top \sqsubseteq \neg Q_{q_r} \tag{5.24}$$

This completes the construction of the TBox $\mathcal{T}_{\mathcal{M},w}$. The initial configuration q_0w is encoded as concept $C_{\mathcal{M},w}$. Let $w = a_1 \dots a_n$. For $2 \leq i \leq n$ define recursively:

$$A_i = \bigcirc_{a_i}(\mathsf{Tape} \sqcap A_{i+1})$$
$$A_{n+1} = \bigcirc_{\llcorner}\mathsf{Tape}$$

and set $C_{\mathcal{M},w} = \bigcirc_{a_1}(\mathsf{BeginTape} \sqcap Q_{q_0} \sqcap A_2)$.

Following the construction is not hard to see that the following holds.

Proposition 5.7. *$C_{\mathcal{M},w}$ is satisfiable w.r.t. $\mathcal{T}_{\mathcal{M},w}$* iff *$\mathcal{M}$ accepts w.*

First assume $w \in L(\mathcal{M})$ and $T_{\mathcal{M}} = (N_{\mathcal{M}}, E_{\mathcal{M}}, \mathsf{conf}(\cdot))$ is an accepting tree of $\mathcal{M}$ on w, where $\mathsf{conf}(n)$ assigns configurations to nodes $n \in N_{\mathcal{M}}$. Moreover, given $\mathsf{conf}(n) = wqw'$, the function $\sharp_i(\mathsf{conf}(n))$ returns the i-th symbol of ww', and $\mathsf{h}(\mathsf{conf}(n))$ the position of the head. We describe the construction of a model $\mathfrak{I} = (\Delta, W, \{R_x\}_{x\in\Gamma}, \{\mathcal{I}_w\}_{w\in W})$ of $\mathcal{T}_{\mathcal{M},w}$ and $C_{\mathcal{M},w}$. Let $m = |w|$, we begin by defining

$$W = \{w_n^i \mid n \in N_{\mathcal{M}} \wedge 0 \leq i \leq 2^m\} \cup \{\varepsilon\}$$

and set

$$R_x = \{(w_n^i, w_n^{i+1}) \mid \sharp_{i+1}(\mathsf{conf}(n)) = x\} \cup \{(w_n^{2^{m-1}}, w_{n'}^0) \mid \sharp_0(\mathsf{conf}(n')) = x \wedge (n,n') \in E_{\mathcal{M}}\}$$
$$\cup\{(\varepsilon, w_n^0) \mid n \text{ the root of } T_{\mathcal{M}} \wedge \sharp_0(\mathsf{conf}(n)) = x\}.$$

Moreover, set $\Delta = \{d_{\mathcal{M}}\} \cup \{e_w \mid w \in W\}$ and

- $\mathsf{Tape}^{\mathfrak{I},w} = \{d_{\mathcal{M}}\}$;
- $\mathsf{BeginTape}^{\mathfrak{I},w_n^0} = \{d_{\mathcal{M}}\}$; $\mathsf{EndTape}^{\mathfrak{I},w_n^{2^{m-1}}} = \{d_{\mathcal{M}}\}$;
- $Q_q^{\mathfrak{I},w_n^i} = \{d_{\mathcal{M}} \mid \mathcal{M} \text{ is in state } q \wedge \mathsf{h}(\mathsf{conf}(n)) = i\}$;
- $X_j^{\mathfrak{I},w_n^i} = \{d_{\mathcal{M}} \mid \text{ the } j^{th} \text{ bit of the binary representation of } i \text{ is } 1\}$;
- $r^{\mathfrak{I},w_n^i} = \{(d_{\mathcal{M}}, e_{w_n^i})\}$.

The interpretation can be straightforwardly extended to the remaining auxiliary concepts in the construction, e.g., $M_{q,a.m}, N_{q,a,r}$, etc. For example, the concepts W_a, S_a are interpreted as follows:

$$W_a^{\mathfrak{I},w_n^i} = \{e_{w_n^j} \in \Delta \mid \sharp_j(\mathsf{conf}(n)) = a \wedge j \geq i\},$$
$$S_a^{\mathfrak{I},w_n^i} = \{e_{w_{n'}^j} \in \Delta \mid (n,n') \in E^{\mathcal{M}} \wedge e_{w_{n'}^j} \in W_a^{\mathfrak{I},w_{n'}^{2^{|w|-1}}}\}.$$

Note that up to this point we have defined the interpretation of a finite subtree of $\mathfrak{I}$. However, we can straightforwardly extend it by introducing new worlds and extending each path to an infinite path. Most of the concepts can be interpreted as the empty set in the new added worlds, the only exception are the counter concepts: we can properly interpreted them by taking the distance from the root ε, that is, for each added w

$$X_j^{\mathfrak{I},w} = \{d_{\mathcal{M}} \in \Delta \mid \text{ the } j^{th} \text{ bit of the binary representation of } \mathsf{dist}(\varepsilon, w) \text{ is } 1\},$$

where $\mathsf{dist}(\varepsilon, w)$ is a function giving the distance modulo 2^m between the ε and w. Finally, set $C_{\mathcal{M},w}^{\mathfrak{I},\varepsilon} = \{d_{\mathcal{M}}\}$. Now by simply inspecting the axioms of $\mathcal{T}_{\mathcal{M},w}$ one can see that $\mathfrak{I} \models \mathcal{T}_{\mathcal{M},w}$.

The other direction of the claim follows straightforwardly from the reduction. In order to retrieve an ATM tree accepting w from a $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$-tree-model we only need to pick an individual d, such that $d \in C_{\mathcal{M},\omega}^{\mathfrak{I},w_0}$ and follow the paths of worlds $w \in W$ for which $d \in \mathsf{Tape}^{\mathfrak{I},w}$, just as presented in Figure 5.4. On the way we collect information about the entire configuration. Two important comments are in order. First, note that the reduction is somewhat underconstrained in the sense that the models might represent also some surplus states or transitions. However, the proper computation tree, i.e., the one directly enforced by the encoding, has to appear within this structure. Secondly, we recall that the ATM trees we consider are all finite. Since the transitions in the reduction properly simulate those of an ATM, therefore the ATM trees embedded in $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$-tree-models have to be also finite, even though the models themselves are always infinite.

❏

The two complexity bounds from Theorem 5.5 and 5.6 together with the reductions established in Proposition 5.2 provide us with the following.

Theorem 5.8. *For any $\mathcal{ML} \in \{\mathbf{DAlt}_n, \mathbf{D}_n, \mathbf{Alt}_n, \mathbf{K}_n\}$, concept satisfiability w.r.t. TBoxes for $\mathcal{ML}_{\mathcal{ALC}}$ is* 2EXPTIME*-complete.*

What follows then from this analysis is that by sacrificing the generality of $\mathbf{K}_n$-frames one does not immediately obtain a better computational behavior as long as multi-modal operators are permitted. For this reason, later on, we adopt $(\mathbf{K}_n)_{\mathcal{ALC}}$ as the baseline for constructing expressive DLCs.

5.5 Introducing Expressive Description Logics of Context

In this section we introduce expressive extensions of simple DLs of context: an *Expressive Description Logic of Context* $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ consists of the DL context language $\mathcal{L}_C$, supporting context descriptions, and of the object language $\mathcal{L}_O$ equipped with context operators for representing object knowledge relative to contexts. Expressive DLCs extend simple DLCs with the possibility of explicitly asserting properties of the contexts and therefore conforming with the second postulate of McCarthy's theory. From here on, when convenient, we drop the *expressive* qualification and simply refer to them as description logics of context.

5.5.1 Syntax and semantics

Definition 5.9. *The* context language *of* $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ *is a DL language* $\mathcal{L}_C$ *over the vocabulary* $\Gamma = (\mathsf{M_C}, \mathsf{M_R}, \mathsf{M_I})$*, where* $\mathsf{M_C}$ *is a set of* context concepts*,* $\mathsf{M_R}$ *a set of* context roles *and* $\mathsf{M_I}$ *a set of* context names*.*[3]

The object language extends standard DLs with special contextualization operators applicable to concepts.

Definition 5.10. *Let* $\mathcal{L}_O$ *be a DL language over the vocabulary* $\Sigma = (\mathsf{N_C}, \mathsf{N_R}, \mathsf{N_I})$*. The* object language *of* $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ *is the smallest language containing* $\mathcal{L}_O$ *and closed under the constructors of* $\mathcal{L}_O$ *and at least one of the two types —* $\mathfrak{F}_1$ resp. $\mathfrak{F}_2$ *— of concept-forming operators:*

$$\langle \boldsymbol{r}.\boldsymbol{C}\rangle\, D \;\mid\; [\boldsymbol{r}.\boldsymbol{C}]\, D \tag{$\mathfrak{F}_1$}$$

$$\langle \boldsymbol{C}\rangle D \;\mid\; [\boldsymbol{C}]D \tag{$\mathfrak{F}_2$}$$

where $\boldsymbol{C}$ *and* $\boldsymbol{r}$ *are a concept and a role of the context language and* D *is a concept of the object language.*

We use standard Boolean abbreviations, plus $[\boldsymbol{r}.\boldsymbol{C}]D$ to abbreviate $\neg\langle \boldsymbol{r}.\boldsymbol{C}\rangle\neg D$, and $[\boldsymbol{C}]D$ to abbreviate $\neg\langle \boldsymbol{C}\rangle\neg D$.

Intuitively, the concept $\langle \boldsymbol{r}.\boldsymbol{C}\rangle D$ denotes all objects which are D in *some* context of type $\boldsymbol{C}$ accessible from the current one through $\boldsymbol{r}$. Analogically, $[\boldsymbol{r}.\boldsymbol{C}]D$ denotes all objects which are D in *every* such context. In the case of $\mathfrak{F}_2$ operators, the concept $\langle \boldsymbol{C}\rangle D$ denotes all objects which are D in *some* context of type $\boldsymbol{C}$, whereas $[\boldsymbol{C}]D$ all objects which are D in *every* such context. For example, $\langle \boldsymbol{neighbor}.\boldsymbol{Country}\rangle\, Citizen$, refers to the concept $Citizen$ in some context of type ***Country*** accessible through the ***neighbor*** relation from the current context. Analogically, $\langle \boldsymbol{HumanAnatomy}\rangle Heart$ refers to the concept $Heart$ in some context of ***HumanAnatomy***.

Definition 5.11. *A* $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$-knowledge base (CKB) *is a pair* $\mathcal{K} = (\mathcal{C}, \mathcal{O})$*, where* $\mathcal{C}$ *is a set of axioms over the context language (in the syntax allowed by* $\mathcal{L}_C$*), and* $\mathcal{O}$ *is a set of formulas of the form:*

$$\boldsymbol{c} : \varphi \mid \boldsymbol{C} : \varphi$$

where φ *is an axiom over the object language (in the syntax allowed by* $\mathcal{L}_O$*),* $\boldsymbol{c} \in M_I$ *and* $\boldsymbol{C}$ *is a concept of the context language.*

A formula $\boldsymbol{c} : \varphi$ states that axiom φ holds in the context denoted by the context name $\boldsymbol{c}$. Note, that this corresponds directly to McCarthy's $ist(\boldsymbol{c}, \varphi)$. Axioms of the form $\boldsymbol{C} : \varphi$ assert the truth of φ in all contexts of type $\boldsymbol{C}$. For example, the following formula states that in every country, the citizens of its neighbor countries do not require visas.

[3] In certain scenarios it might be useful to consider only a subset of $\mathsf{M_I}$ as proper contexts, while the remaining individuals serving merely for context descriptions in $\mathcal{L}_C$. For instance in provenance applications, a context $\boldsymbol{c}$, associated with a single knowledge source, might be described with an axiom $\boldsymbol{hasAuthor}(\boldsymbol{c}, \boldsymbol{henry})$, where ***henry*** is an individual related to $\boldsymbol{c}$, but not a context itself [17].

$$\textit{\textbf{Country}} : \langle \textit{\textbf{neighbor}}.\textit{\textbf{Country}}\rangle\, Citizen \sqsubseteq No\,VisaRequirement$$

The possible world semantics of DLCs is given through $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$*-interpretations* and $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$*-models*, which combine the interpretations of $\mathcal{L}_C$ with those of $\mathcal{L}_O$.

Definition 5.12. *A* $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$*-interpretation is a tuple* $\mathfrak{I} = (\mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\mathcal{I}_i\}_{i\in\mathfrak{C}})$*, such that:*

1. $(\mathfrak{C}, \cdot^{\mathcal{J}})$ *is a classical DL interpretation of* $\mathcal{L}_C$*, where* $\mathfrak{C}$ *is a non-empty domain of contexts and* $\cdot^{\mathcal{J}}$ *an interpretation function defined for* $\mathcal{L}_C$ *as usual,*
2. $\mathcal{I}_i$ *is a classical DL interpretation of* $\mathcal{L}_O$ *with domain* Δ *such that* $a^{\mathcal{I}_i} = a^{\mathcal{I}_j}$ *for all* $a \in \mathsf{M_I}$ *and* $i, j \in \mathfrak{C}$*. The mapping* $\cdot^{\mathfrak{I},i}$ *is extended to complex concepts as follows:*

   $(\mathfrak{F}_1)$ *for every* $\langle \boldsymbol{r}.\boldsymbol{C}\rangle D$ *and* $[\boldsymbol{r}.\boldsymbol{C}]D$*:*
   - $(\langle \boldsymbol{r}.\boldsymbol{C}\rangle\, D)^{\mathfrak{I},i} = \{d \in \Delta \mid \exists j \in \mathfrak{C} : (i,j) \in \boldsymbol{r}^{\mathcal{J}} \wedge j \in \boldsymbol{C}^{\mathcal{J}} \wedge d \in D^{\mathfrak{I},j}\}$,
   - $([\boldsymbol{r}.\boldsymbol{C}]\, D)^{\mathfrak{I},i} = \{d \in \Delta \mid \forall j \in \mathfrak{C} : (i,j) \in \boldsymbol{r}^{\mathcal{J}} \wedge j \in \boldsymbol{C}^{\mathcal{J}} \rightarrow d \in D^{\mathfrak{I},j}\}$.

   $(\mathfrak{F}_2)$ *for every* $\langle \boldsymbol{C}\rangle D$ *and* $[\boldsymbol{C}]D$*:*
   - $(\langle \boldsymbol{C}\rangle D)^{\mathfrak{I},i} = \{d \in \Delta \mid \exists j \in \mathfrak{C} : j \in \boldsymbol{C}^{\mathcal{J}} \wedge d \in D^{\mathfrak{I},j}\}$,
   - $([\boldsymbol{C}]D)^{\mathfrak{I},i} = \{d \in \Delta \mid \forall j \in \mathfrak{C} : j \in \boldsymbol{C}^{\mathcal{J}} \rightarrow d \in D^{\mathfrak{I},j}\}$.

We usually write $C^{\mathfrak{I},i}$ instead of $C^{\mathcal{I}_i}$; intuitively $d \in C^{\mathfrak{I},i}$ means that in the interpretation $\mathfrak{I}$, d is an instance of C in the context i. In the previous definition we make the *constant domain assumption*, i.e., each context shares the same domain Δ. Intuitively, this means that objects are not created or destroyed while making a transition from one context to another. This is the most general choice since expanding, decreasing and varying domains can all be simulated.

Clearly, the difference between the context operators of type $\mathfrak{F}_1$ and $\mathfrak{F}_2$ lies in the choice of the relational structures observed when quantifying over the context domain. $\mathfrak{F}_1$-operators bind contexts only along the roles of the context language (as $\mathbf{K}$-modalities), while $\mathfrak{F}_2$-operators follow the universal relation over $\mathfrak{C}$ (as $\mathbf{S5}$-modalities). This leads to some clear consequences in the scope and the character of the distribution of the object knowledge over contexts in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-models. For instance, in Figure 5.1, the concept $\langle \boldsymbol{t}.\boldsymbol{F}\rangle B$ is satisfied by object a only in context $\boldsymbol{c}$, while $\langle \boldsymbol{F}\rangle B$ is satisfied by a in all contexts in the model. From the perspective of McCarthy's theory, employing operators $\mathfrak{F}_2$, rather than $\mathfrak{F}_1$, is equivalent to sacrificing postulate (3). This means that every two contexts in the model become in principle accessible to each other. The focus on $\mathbf{K}$-like and $\mathbf{S5}$-like modalities is quite arbitrary here and driven merely by the formal simplicity of the two types of operators and easiness of their integration with the DL semantics. In principle, however, nothing prevents from constructing logics containing contextualization operators which mimic other common modalities.

Definition 5.13. *A* $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$*-interpretation* $\mathfrak{I} = (\mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\mathcal{I}_i\}_{i\in\mathfrak{C}})$ *is a* model *of a CKB* $\mathcal{K} = (\mathcal{C}, \mathcal{O})$ *if the following hold:*

– *for every* $\varphi \in \mathcal{C}$, $(\mathfrak{C}, \cdot^{\mathcal{J}})$ *satisfies* φ,

– *for every* $\boldsymbol{c} : \varphi \in \mathcal{O}$, $\mathcal{I}_{\boldsymbol{c}^{\mathcal{J}}}$ *satisfies* φ,

– *for every* $\boldsymbol{C} : \varphi \in \mathcal{O}$ *and* $i \in \mathfrak{C}$, *if* $i \in \boldsymbol{C}^{\mathcal{J}}$ *then* $\mathcal{I}_i$ *satisfies* φ.

The central reasoning problem we study is KB satisfiability: a $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ KB $\mathcal{K}$ is *satisfiable* if $\mathcal{K}$ has a model.

5.6 Simple vs Expressive Description Logics of Context

We next present a rough analysis of the expressive limits of DLCs by direct comparisons to simple DLCs and therefore to standard two-dimensional DLs. The results which we deliver here are *not* exhaustive, but nevertheless, they offer a good limiting characterization of the proposed logics. We show that the expressive power of the DLC framework properly subsumes the expressiveness of $(\mathbf{K}_n)_{\mathcal{L}}$ and $\mathbf{S5}_{\mathcal{L}}$.

The following result shows that concept satisfiability w.r.t. TBoxes for $(\mathbf{K}_n)_{\mathcal{L}}$ can be immediately restated as the problem of KB satisfiability for $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ with only $\mathfrak{F}_1$ operators.

Theorem 5.9 ($(\mathbf{K}_n)_{\mathcal{L}}$ vs. $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$)**.** *Deciding concept satisfiability w.r.t. TBoxes for* $(\mathbf{K}_n)_{\mathcal{L}}$ *is linearly reducible to KB satisfiability in* $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$, *for* $\mathcal{L}_O = \mathcal{L}$, *with only context operators of type* $\mathfrak{F}_1$.

Proof. Let $(C, \mathcal{T})$ be a problem instance in $(\mathbf{K}_n)_{\mathcal{L}}$. Define the corresponding KB $\mathcal{K} = (\mathcal{C}, \mathcal{O})$ in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ as follows. First, set $\mathcal{C} = \emptyset$ and $\mathcal{O} = \{\top : B \sqsubseteq D \mid B \sqsubseteq D \in \mathcal{T}\} \cup \{\top : (\langle \boldsymbol{s}.\top\rangle C)(a)\}$, for a context role $\boldsymbol{s}$ and some fresh individual object name a. Then, with every pair of $\mathbf{K}$-modalities $\Diamond_i$, $\Box_i$ in $(\mathbf{K}_n)_{\mathcal{L}}$ associate a distinct context role name $\boldsymbol{r}_i$ and replace every occurrence of $\Diamond_i$ in $\mathcal{O}$ with $\langle \boldsymbol{r}_i.\top\rangle$ and every occurrence of $\Box_i$ with $[\boldsymbol{r}_i.\top]$. Then, C is satisfiable w.r.t. $\mathcal{T}$ in $(\mathbf{K}_n)_{\mathcal{L}}$ *iff* the resulting KB $\mathcal{K}$ is satisfiable in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$. This conclusion follows immediately by observing the direct correspondence between the semantics of both languages; in particular the semantics of the $\mathbf{K}$-modalities and global TBox axioms in $(\mathbf{K}_n)_{\mathcal{L}}$ and of the corresponding $\mathfrak{F}_1$ operators and formulas $\top : \varphi$ in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$. Based on this observation, it is easy to see that $(\Delta, \{R_i\}_{i\in(1,n)}, W, \{\mathcal{I}_w\}_{w\in W})$ is a model of $\mathcal{T}$ *iff* $(W, \cdot^{\mathcal{J}}, \Delta, \{\mathcal{I}_w\}_{w\in W})$ is a model of $\mathcal{K}$, where $R_i = (\boldsymbol{r}_i)^{\mathcal{J}}$, for every $i \in (1, n)$, and the concept C is satisfied in some $w \in W$ by the individual $a^{\mathfrak{I},w} \in \Delta$.

❑

In the same manner, we devise a reduction from $\mathbf{S5}_{\mathcal{L}}$ to $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ with only $\mathfrak{F}_2$ operators.

Theorem 5.10 ($\mathbf{S5}_{\mathcal{L}}$ vs. $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$)**.** *Deciding concept satisfiability w.r.t. TBoxes for* $\mathbf{S5}_{\mathcal{L}}$ *is linearly reducible to KB satisfiability in* $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$, *for* $\mathcal{L}_O = \mathcal{L}$, *with only context operators of type* $\mathfrak{F}_2$.

Proof. Let $(C, \mathcal{T})$ be a problem instance in $\mathbf{S5}_{\mathcal{L}}$. Again, define the KB $\mathcal{K} = (\mathcal{C}, \mathcal{O})$ in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ by setting $\mathcal{C} = \emptyset$ and $\mathcal{O} = \{\top : B \sqsubseteq D \mid B \sqsubseteq D \in \mathcal{T}\} \cup \{\top : (\langle\top\rangle C)(a)\}$, for some fresh individual name a. Then, replace every occurrence of $\Diamond$ in $\mathcal{O}$ with $\langle\top\rangle$ and every occurrence of $\Box$ with $[\top]$. Consequently, C is satisfiable w.r.t. $\mathcal{T}$ in $\mathbf{S5}_{\mathcal{L}}$ *iff* the resulting KB $\mathcal{K}$ is satisfiable in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$. Analogically to the previous case, observe that the semantics of $\mathbf{S5}$-modalities coincides with that of $\mathfrak{F}_2$ operators and so $(\Delta, W, \{\mathcal{I}_w\}_{w\in W})$ is a model of $\mathcal{T}$ *iff* $(W, \cdot^{\mathcal{J}}, \Delta, \{\mathcal{I}_w\}_{w\in W})$ is a model of $\mathcal{K}$, where the concept C is satisfied in some $w \in W$ by the individual $a^{\mathfrak{I},w} \in \Delta$.

❑

Observe that for the reductions we use only a residual context language. In the former case we merely require the top concept and a set of context role names, while in the latter only the top concept. Clearly, there is also no need for employing axioms of the context language. This suggests that the expressive power of DLCs might be in general even greater and strictly subsume that of the union of $(\mathbf{K}_n)_{\mathcal{L}}$ and $\mathbf{S5}_{\mathcal{L}}$. Indeed, it is not difficult to instantiate this intuition with concrete examples of properties which are expressible in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ but cannot be captured by any of the underlying two-dimensional languages. For instance, in formulas of the form $\boldsymbol{c} : \varphi$ the context name $\boldsymbol{c}$ uniquely identifies a world in which the formula φ must be satisfied. Moreover, we can express complex modalities, e.g.,

$$\langle \boldsymbol{A}\rangle C \sqcup [\boldsymbol{A} \sqcap \neg \boldsymbol{B}]C,$$

describing the set of objects which are C in any context of type $\boldsymbol{A}$ or in all contexts of type $\boldsymbol{A}$ and $\neg\boldsymbol{B}$. Obviously neither $(\mathbf{K}_n)_{\mathcal{L}}$ or $\mathbf{S5}_{\mathcal{L}}$ allows for expressing such properties, as they require a more fine-grained mechanism of quantifying over possible worlds, offered by the context language in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$.

Although we do not have a precise characterization result for the expressiveness of $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$, it seems that at least to some extent its behavior can be simulated in two-dimensional DLs extended with *global concepts*, that is, concepts C such that for every $w \in W$ one of the following holds: $C^{\mathfrak{I},w} = \Delta$ or $C^{\mathfrak{I},w} = \emptyset$. Technically, global concepts can be used to simulate the context language by associating with every context concept $\boldsymbol{C}$ its global counterpart $C_{\boldsymbol{C}}$ and requiring that for every $w \in W$ the following holds: $w \in \boldsymbol{C}^{\mathcal{J}}$ iff $C_{\boldsymbol{C}}^{\mathfrak{I},w} = \Delta$. Given this restriction, for example, we can translate concepts of the form $\langle \boldsymbol{r}.\boldsymbol{C}\rangle D$ into a $(\mathbf{K}_n)_{\mathcal{ALC}}$ concept $\Diamond_r(C_{\boldsymbol{C}} \sqcap D)$.

However, even if a complete reduction to two-dimensional DLs with global vocabulary was possible, this approach would be still conceptually inadequate to our motivation, as the semantics of global expressions would be defined purely in terms of the object domain and not the domain of contexts. Moreover, the interaction between the two levels of representation would be highly obscured, making it hard to define fragments of $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ in a modular fashion –simply by selecting DLs of desired expressiveness for $\mathcal{L}_C$ and $\mathcal{L}_O$.

5.7 Reasoning in Expressive Description Logics of Context

This section is dedicated to the study of the computational complexity of DLCs based on the DL $\mathcal{ALCO}$. We provide a 2EXPTIME upper bound for the KB satisfiability problem for $\mathfrak{C}^{\mathcal{ALCO}}_{\mathcal{ALCO}}$. Remarkably, in the light of the analysis carried out in Section 5.6, this result shows that we can equip $(\mathbf{K}_n)_{\mathcal{ALC}}$ with means for describing contexts and comply therefore with all McCarthy's postulates without an increase in the complexity.

5.7.1 An Algorithm for KB Satisfiability for $\mathfrak{C}_{\mathcal{ALCO}}^{\mathcal{ALCO}}$

In order to prove decidability of the KB satisfiability problem for $\mathfrak{C}_{\mathcal{ALCO}}^{\mathcal{ALCO}}$, we devise a *quasistate elimination algorithm*. Particularly, this algorithm can be seen as an extension of the algorithm for concept satisfiability w.r.t. TBoxes for $(\mathbf{K}_n)_{\mathcal{ALC}}$ presented in Section 5.4. Again, instead of looking directly for a model of a KB, we abstract from the possibly infinite domains $\mathfrak{C}$ and Δ, and consider only a finite number of quasistates which represent possible *context* types inhabited by a finite number of possible *object* types. Further all quasistates which do not satisfy certain criteria are iteratively eliminated.

Let $\mathcal{K} = (\mathcal{C}, \mathcal{O})$ be a KB formulated in $\mathfrak{C}_{\mathcal{ALCO}}^{\mathcal{ALCO}}$ with context operators $\mathfrak{F}_1$ and $\mathfrak{F}_2$ whose satisfiability is to be decided. Further, we apply the following replacements of all respective (sub)formulas with their equivalents:

$$\begin{array}{rclcrcl} C(a) & \Rightarrow & \{a\} \sqsubseteq C, & \qquad & r(a,b) & \Rightarrow & \{a\} \sqsubseteq \exists r.\{b\}, \\ \boldsymbol{C}(\boldsymbol{a}) & \Rightarrow & \{\boldsymbol{a}\} \sqsubseteq \boldsymbol{C}, & \qquad & \boldsymbol{r}(\boldsymbol{a},\boldsymbol{b}) & \Rightarrow & \{\boldsymbol{a}\} \sqsubseteq \exists \boldsymbol{r}.\{\boldsymbol{b}\}. \end{array}$$

The following notation is used to denote particular sets of *object* symbols occurring in $\mathcal{K}$:

– $\mathsf{cl}_o(\mathcal{K})$:	set of object concepts that occur in $\mathcal{K}$, closed under subconcepts and negation;
– $\mathsf{rol}_o(\mathcal{K})$:	set of object roles that occur in $\mathcal{K}$;
– $\mathsf{ind}_o(\mathcal{K})$:	set of object individual names that occur in $\mathcal{K}$;
– $\mathsf{sub}_o(\mathcal{K})$:	set of axioms from $\{\varphi \mid \boldsymbol{C} : \varphi \in \mathcal{O}$ for any $\boldsymbol{C}\}$.

Moreover, we use $\mathsf{cl}_c(\mathcal{K})$, $\mathsf{rol}_c(\mathcal{K})$, $\mathsf{ind}_c(\mathcal{K})$ to denote the analogous sets over *context* concepts and roles occurring in $\mathcal{K} \cup \{\exists \boldsymbol{r}.\boldsymbol{C}, \neg\exists \boldsymbol{r}.\boldsymbol{C} \mid \langle \boldsymbol{r}.\boldsymbol{C}\rangle D, \neg\langle \boldsymbol{r}.\boldsymbol{C}\rangle D$ occurs in $\mathcal{K}\}$. We proceed to define a *context* and *object type*.

Definition 5.14. *A* context type for $\mathcal{K}$ *is a subset* $t_c \subseteq \mathsf{cl}_c(\mathcal{K})$*, where:*

- $\boldsymbol{C} \in t_c$ *iff* $\neg\boldsymbol{C} \notin t_c$*, for all* $\boldsymbol{C} \in \mathsf{cl}_c(\mathcal{K})$*;*
- $\boldsymbol{C} \sqcap \boldsymbol{D} \in t_c$ *iff* $\{\boldsymbol{C}, \boldsymbol{D}\} \subseteq t_c$*, for all* $\boldsymbol{C} \sqcap \boldsymbol{D} \in \mathsf{cl}_c(\mathcal{K})$.

Furthermore, an object type for $\mathcal{K}$ *is a subset* $t_o \subseteq \mathsf{cl}_o(\mathcal{K})$*, where:*

- $C \in t_o$ *iff* $\neg C \notin t_o$*, for all* $C \in \mathsf{cl}_o(\mathcal{K})$*;*
- $C \sqcap D \in t_o$ *iff* $\{C, D\} \subseteq t_o$*, for all* $C \sqcap D \in \mathsf{cl}_o(\mathcal{K})$.

We can now define the notion of quasistate: intuitively, a quasistate captures a '*slice*' of a model representing one possible context inhabited by a set of possible objects.

Definition 5.15. *A* quasistate for $\mathcal{K}$ *is a tuple* $q = \langle t_{c_q}, f_q, O_q \rangle$ *with* t_{c_q} *a context type for* $\mathcal{K}$, $f_q \subseteq \mathsf{sub}_o(\mathcal{K})$ *and* O_q *a non-empty set of object types for* $\mathcal{K}$, *such that for every* $t_o \in O_q$ *the following holds:*

($\mathbf{QS}$) *if* $\exists r.D \in t_o$ *then there is a type* $t'_o \in O_q$ *such that* $\{D\} \cup \{\neg E \mid \neg\exists r.E \in t_o\} \subseteq t'_o$.

Furthermore, we say that q *is* coherent *if the following conditions hold:*

($\mathbf{QC1}$*) for every* $a \in \mathsf{ind}_o(\mathcal{K})$ *there exists a unique* $t_o \in O_q$ *such that* $\{a\} \in t_o$*;*

($\mathbf{QC2}$*) for every* $\boldsymbol{C} : \varphi \in \mathcal{O}$*, if* $\boldsymbol{C} \in t_{c_q}$ *then* $\varphi \in f_q$*;*

($\mathbf{QC3}$*) for every* $C \sqsubseteq D \in f_q$ *and* $t_o \in O_q$*, if* $C \in t_o$ *then* $D \in t_o$*;*

($\mathbf{QC4}$*) for every* $t_o \in O_q$ *and* $\neg\langle \boldsymbol{C} \rangle D \in t_o$*, if* $\boldsymbol{C} \in t_{c_q}$ *then* $\neg D \in t_o$*.*

We denote by $\mathsf{qs}(\mathcal{K})$ the set of all quasistates for $\mathcal{K}$. Moreover, a *linkage* between two quasistates $q = \langle t_{c_q}, f_q, O_q \rangle$ and $q' = \langle t_{c_{q'}}, f_{q'}, O_{q'} \rangle$ for $\mathcal{K}$ is a mapping $\lambda = g \cup h$, where $g : O_q \to O_{q'}$ and $h : O_{q'} \to O_q$, such that for every $a \in \mathsf{ind}_o(\mathcal{K})$ and $t_o \in O_q \cup O_{q'}$, $\{a\} \in t_o$ *iff* $\{a\} \in \lambda(t_o)$.

The following definition will be used to reconstruct the accessibility relations between individuals of the object dimension.

Definition 5.16. *Let* $q = \langle t_{c_q}, f_q, O_q \rangle$ *and* $q' = \langle t_{c_{q'}}, f_{q'}, O_{q'} \rangle$ *be two quasistates for* $\mathcal{K}$*. Then* q' *is a* matching $\mathfrak{F}_2$-successor for q via a linkage λ *if for every* $t_o \in O_q \cup O_{q'}$, $\{\langle \boldsymbol{C} \rangle D, \neg\langle \boldsymbol{C} \rangle D \in t_o\} = \{\langle \boldsymbol{C} \rangle D, \neg\langle \boldsymbol{C} \rangle D \in \lambda(t_o)\}$.

Furthermore, for any $\boldsymbol{r} \in \mathsf{rol}_c(\mathcal{K})$*, we say that* q' *is a* matching $\boldsymbol{r}$- successor for q via a linkage λ *if* q' *is a matching* $\mathfrak{F}_2$*-successor for* q *via* λ *and the following conditions are satisfied:*

- $\{\neg \boldsymbol{C} \mid \neg\exists \boldsymbol{r}.\boldsymbol{C} \in t_{c_q}\} \subseteq t_{c_{q'}}$*;*
- *for every* $t_o \in O_q$ *and* $t'_o \in O_{q'}$*,* $\{\neg D \mid \neg\langle \boldsymbol{r}.\boldsymbol{C} \rangle D \in t_o, \boldsymbol{C} \in t_{c_{q'}}\} \subseteq \lambda(t_o)$ *and* $\{\neg D \mid \neg\langle \boldsymbol{r}.\boldsymbol{C} \rangle D \in \lambda(t'_o), \boldsymbol{C} \in t_{c_{q'}}\} \subseteq t'_o$*.*

Moreover, we say that a set of quasistates Q is *saturated* if for every quasistate $q \in Q$ with $q = \langle t_{c_q}, f_q, O_q \rangle$ the following hold:

($\mathbf{QS1}$) for every $\exists \boldsymbol{r}.\boldsymbol{C} \in t_{c_q}$ there is a matching $\boldsymbol{r}$-successor q' for q in Q via some linkage λ, such that $\boldsymbol{C} \in t_{c_{q'}}$;

($\mathbf{QS2}$) for every $t_o \in O_q$ and $\langle \boldsymbol{C} \rangle D \in t_o$ there is a matching $\mathfrak{F}_2$-successor $q' = \langle t_{c_{q'}}, f_{q'}, O_{q'} \rangle$ for q in Q via some linkage λ, such that $\boldsymbol{C} \in t_{c_{q'}}$ and $D \in \lambda(t_o)$;

($\mathbf{QS3}$) for every $t_o \in O_q$ and $\langle \boldsymbol{r}.\boldsymbol{C} \rangle D \in t_o$, there is a matching $\boldsymbol{r}$-successor $q' = \langle t_{c_{q'}}, f_{q'}, O_{q'} \rangle$ for q in Q via some linkage λ, such that $\boldsymbol{C} \in t_{c_{q'}}$ and $D \in \lambda(t_o)$, and $\exists \boldsymbol{r}.\boldsymbol{C} \in t_{c_q}$.

Now, we have the necessary ingredients to define a finite abstraction of a $\mathfrak{C}^{\mathcal{ALCO}}_{\mathcal{ALCO}}$ model where the DL-model associated to a context is represented by elements in a quasitate and the correspondence between the elements of different quasistates is defined via the matching conditions introduced above.

Definition 5.17. *A* quasimodel $\mathfrak{M}$ *for $\mathcal{K}$ is a non-empty, saturated set of coherent quasistates for $\mathcal{K}$ satisfying the following conditions:*

*(**M1**) For every $\boldsymbol{c} \in \mathsf{ind}_c(\mathcal{K})$ there is a unique $q = \langle t_{c_q}, f_q, O_q \rangle$ in $\mathfrak{M}$, such that $\{\boldsymbol{c}\} \in t_{c_q}$.*

*(**M2**) For every $\boldsymbol{C} \sqsubseteq \boldsymbol{D} \in \mathcal{C}$ and $q = \langle t_{c_q}, f_q, O_q \rangle$ in $\mathfrak{M}$, if $\boldsymbol{C} \in t_{c_q}$ then $\boldsymbol{D} \in t_{c_q}$.*

The next lemma shows that to decide satisfiability of $\mathcal{K}$, it suffices to check the existence of a quasimodel.

Lemma 5.11. *There is a quasimodel for $\mathcal{K}$* iff *there is an $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-model of $\mathcal{K}$.*

Proof. The key observation which we exploit in this proof is that the constraints (**QS1**)-(**QS3**) imposed on quasimodels ensure existence of certain specific quasistates, which represent successors in the context dimension, and existence of special linkage relations allowing for a proper choice of types for the same object in different contexts. To ease reference to these elements we amend the corresponding conditions with the following naming conventions:

(**QS1***) In such case call q' a *witness* for $(\exists \boldsymbol{r}.\boldsymbol{C}, q)$ and a linkage λ, enforced by the condition, a *witnessing linkage*.

(**QS2***) In such case call q' a *witness* for $(\langle \boldsymbol{C} \rangle D, t, q)$ and a linkage λ, enforced by the condition, a *witnessing linkage*

(**QS3***) In such case call q' a *witness* for $(\langle \boldsymbol{r}.\boldsymbol{C} \rangle D, t, q)$ and a linkage λ, enforced by the condition, a *witnessing linkage*.

($\Rightarrow$) Let $\mathfrak{M}$ be a quasimodel for $\mathcal{K} = (\mathcal{C}, \mathcal{O})$. We present the construction of an $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-model $\mathfrak{I} = (\mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\mathcal{I}_i\}_{i \in \mathfrak{C}})$ of $\mathcal{K}$. We start by inductively defining the interpretation $\mathcal{J} = (\mathfrak{C}, \cdot^{\mathcal{J}})$ of the context dimension. We define sequences $\mathfrak{C}_0, \mathfrak{C}_1, \ldots$, $\mathsf{R}_0(\boldsymbol{r}), \mathsf{R}_1(\boldsymbol{r}) \ldots$, for all $\boldsymbol{r} \in \mathsf{rol}_c(\mathcal{K})$, and the partial mappings $\pi_i : \mathfrak{C}_i \to \mathfrak{M}$. We obtain the desired sets $\mathfrak{C}$ and $\mathsf{R}_{\boldsymbol{r}}$ in the limit. From now on, for $q = \langle t_{c_q}, f_q, O_q \rangle$ we denote by '$\boldsymbol{C} \in q$' the fact that $\boldsymbol{C} \in t_{c_q}$.

- To start the construction of $\mathcal{J}$, set
  - for every $\boldsymbol{a} \in \mathsf{ind}_c(\mathcal{K})$ and $q \in \mathfrak{M}$ such that $\{\boldsymbol{a}\} \in q$, add $\boldsymbol{c}$ to $\mathfrak{C}^0$, set $\pi_0(\boldsymbol{c}) = q$, and $\mathsf{R}_0(\boldsymbol{r}) = \emptyset$, for all $\boldsymbol{r} \in \mathsf{rol}_c(\mathcal{K})$;
  - if $\mathsf{ind}_c(\mathcal{K}) = \emptyset$, set $\mathfrak{C} = \{\boldsymbol{c}\}$, $\pi(\boldsymbol{c}) = q$ for some $q \in \mathfrak{M}$, and $\mathsf{R}_0(\boldsymbol{r}) = \emptyset$, for all $\boldsymbol{r} \in \mathsf{rol}_c(\mathcal{K})$.
- For the induction step, we start by setting $\mathfrak{C}_i = \mathfrak{C}_{i-1}$, $\mathsf{R}_i(\boldsymbol{r}) = \mathsf{R}_{i-1}(\boldsymbol{r})$ and $\pi_i = \pi_{i-1}$, and then proceed as follows: if $\boldsymbol{c} \in \mathfrak{C}_i$ with $\pi(\boldsymbol{c}) = \langle t_{c_q}, f_q, O_q \rangle$, then

C1 if $\exists \boldsymbol{r}.\boldsymbol{C} \in \pi_i(\boldsymbol{c})$, then add $\boldsymbol{c}'$ to $\mathfrak{C}_i$ and $(\boldsymbol{c}, \boldsymbol{c}')$ to $\mathsf{R}_i(\boldsymbol{r})$, and set $\pi(\boldsymbol{c}') = q'$ such that q' is a witness for $(\exists \boldsymbol{r}.\boldsymbol{C}, q)$;

C2 if $t \in O_q$ and $\langle \boldsymbol{C} \rangle D \in t$, then add $\boldsymbol{c}'$ to $\mathfrak{C}_i$ and set $\pi(\boldsymbol{c}') = q'$ such that q' is a witness for $(\langle \boldsymbol{C} \rangle D, t, q)$;

C3 if $t \in O_q$ and $\langle \boldsymbol{r}.\boldsymbol{C} \rangle D \in t$, then add $\boldsymbol{c}'$ to $\mathfrak{C}_i$ and $(\boldsymbol{c}, \boldsymbol{c}')$ to $\mathsf{R}_i(\boldsymbol{r})$, and set $\pi(\boldsymbol{c}') = q'$ such that q' is a witness for $(\langle \boldsymbol{r}.\boldsymbol{C} \rangle D, t, q)$.

Finally, set $\mathfrak{C} = \bigcup_{i \geq 0} \mathfrak{C}_i$ and $\mathsf{R}(\boldsymbol{r}) = \bigcup_{i \geq 0} \mathsf{R}_i(\boldsymbol{r})$. We define the interpretation function $\cdot^{\mathcal{J}}$ as follows:

- $\boldsymbol{a}^{\mathcal{J}} = \boldsymbol{c}$ if $\{\boldsymbol{a}\} \in \pi(\boldsymbol{c})$, for all $\boldsymbol{a} \in \mathsf{ind}_c(\mathcal{K})$;
- $\boldsymbol{A}^{\mathcal{J}} = \{\boldsymbol{c} \in \mathfrak{C} \mid \boldsymbol{A} \in \pi(\boldsymbol{c})\}$;
- $\boldsymbol{r}^{\mathcal{J}} = \{(\boldsymbol{c}, \boldsymbol{c}') \in \mathfrak{C} \times \mathfrak{C} \mid (\boldsymbol{c}, \boldsymbol{c}') \in \mathsf{R}(\boldsymbol{r})\}$.

This is a standard one-dimensional construction. In particular, by structural induction it follows that all complex context concepts are satisfied by $\mathcal{J}$ in the expected contexts.

Claim. For all $\boldsymbol{C} \in \mathsf{cl}_c(\mathcal{K})$ and $\boldsymbol{c} \in \mathfrak{C}$, we have that

$$\boldsymbol{c} \in \boldsymbol{C}^{\mathcal{J}} \text{ iff } \boldsymbol{C} \in \pi(\boldsymbol{c})$$

Proof of the claim: The proof is by induction on the structure of C. The induction start, where C is a concept name is immediate by the definition of $\mathcal{J}$. For the induction step, we distinguish the following cases:

- $\boldsymbol{C} = \neg \boldsymbol{C}$ and $\boldsymbol{C} = \boldsymbol{C}_1 \sqcap \boldsymbol{C}_2$ are standard.
- $\boldsymbol{C} = \{\boldsymbol{a}\}$. By semantics, $\{\boldsymbol{a}\}^{\mathcal{J}} = \{\boldsymbol{a}^{\mathcal{J}}\}$. By definition of $\mathcal{J}$, $\boldsymbol{a}^{\mathcal{J}} = \boldsymbol{c}$ such that $\{\boldsymbol{a}\} \in \pi(\boldsymbol{c})$. Finally, by (**M1**), such $\pi(\boldsymbol{c})$ exists and it is unique. Therefore, $\{\boldsymbol{a}\} \in \pi(\boldsymbol{c})$ iff $\boldsymbol{c} \in \{\boldsymbol{a}\}^{\mathcal{J}}$.
- $\boldsymbol{C} = \exists \boldsymbol{r}.C$ "if:" $\boldsymbol{c} \in (\exists \boldsymbol{r}.\boldsymbol{C})^{\mathcal{J}}$, that is, there exists a $\boldsymbol{c}'$ such that $(\boldsymbol{c}, \boldsymbol{c}') \in \boldsymbol{r}^{\mathcal{J}}$ and $\boldsymbol{c}' \in \boldsymbol{C}^{\mathcal{J}}$. By definition of $\boldsymbol{r}^{\mathcal{J}}$, we have that $(\boldsymbol{c}, \boldsymbol{c}') \in \mathsf{R}(\boldsymbol{r})$. Now, by construction, $(\boldsymbol{c}, \boldsymbol{c}')$ were added to $\mathsf{R}(\boldsymbol{r})$ either by rule **C1** or **C3**. First note that by I.H., $\boldsymbol{C} \in \pi(\boldsymbol{c}')$. But then, conditions (**QS1**) and (**QS3**), respectively, imply that $\exists \boldsymbol{r}.\boldsymbol{C} \in \pi(\boldsymbol{c})$.

  "only if:" $\exists \boldsymbol{r}.\boldsymbol{C} \in \pi(\boldsymbol{c})$. Now, by (**QS1**), there is a $q = \langle t_{c_q}, f_q, O_q \rangle$ such that q' is matching $\boldsymbol{r}$ successor for q via some linkage λ with $\boldsymbol{C} \in q$. By rule **C1**, there is a $\boldsymbol{c}'$ with $\pi(\boldsymbol{c}') = q$, and then, by I.H., $\boldsymbol{c} \in \boldsymbol{C}^{\mathcal{J}}$. Moreover, also by rule **C1**, $(\boldsymbol{c}, \boldsymbol{c}') \in r^{\mathcal{J}}$. Therefore, by semantics, $\boldsymbol{c} \in (\exists \boldsymbol{r}.\boldsymbol{C})^{\mathcal{J}}$.

This finishes the proof of the claim.

Now, by condition (**M2**), we have that for all $\boldsymbol{C} \sqsubseteq \boldsymbol{D}$ and $q = \langle t_{c_q}, f_q, O_q \rangle$, if $\boldsymbol{C} \in t_{c_q}$, then $\boldsymbol{D} \in t_{c_q}$. We have thus, by the previous claim, that for all $\boldsymbol{c} \in \mathfrak{C}$, if $\boldsymbol{c} \in \boldsymbol{C}^{\mathcal{J}}$, then $\boldsymbol{c} \in \boldsymbol{D}^{\mathcal{J}}$. Therefore, all axioms from the context KB $\mathcal{C}$ must be satisfied.

Next we turn to the object dimension; note that this part of the construction is similar to that for $(\mathbf{K}_n)_{\mathcal{ALC}}$.

A *run* ρ through $\mathfrak{C}$ is a *choice function* which for every $\boldsymbol{c} \in \mathfrak{C}$ selects an object type $\rho(\boldsymbol{c}) \in O_q$, such that $\pi(\boldsymbol{c}) = \langle t_{c_q}, f_q, O_q \rangle$.

Intuitively, runs are used for representing the behavior of object individuals across contexts. The easiest way to properly constrain this behavior is by employing the witnessing linkages introduced in conditions (**QS1**)-(**QS3**). Note that the way the interpretation $(\mathfrak{C}, \cdot^{\mathcal{J}})$ is constructed ensures that for every two contexts there exists a witnessing linkage we can refer to in order to align the interpretations of object individuals inhabiting these contexts.

A set of runs $\mathfrak{R}$ is *coherent* if for every $\boldsymbol{c}, \boldsymbol{c}' \in \mathfrak{C}$, with $\pi(\boldsymbol{c}) = \langle t_{c_q}, f_q, O_q \rangle$ and $\pi(\boldsymbol{c}') = \langle t_{c_{q'}}, f_{q'}, O_{q'} \rangle$ and λ being the witnessing linkage between q and q', the following conditions are satisfied:

- for every $a \in \mathsf{ind}_o(\mathcal{K})$, there is exactly one run $\rho_a \in \mathfrak{R}$ such that $\{a\} \in \rho_a(\boldsymbol{c})$;
- for every $\rho \in \mathfrak{R}$, $\lambda(\rho(\boldsymbol{c})) = \rho(\boldsymbol{c}')$;
- for every $t \in O_q$ and $t' \in O_{q'}$, if $\lambda(t) = t'$ then there exists $\rho \in \mathfrak{R}$, such that $\rho(\boldsymbol{c}) = t$ and $\rho(\boldsymbol{c}') = t'$.

Let $\Delta = \mathfrak{R}$, with $\mathfrak{R}$ a set of coherent runs through $\mathfrak{C}$. It remains to define the interpretation function $\cdot^{\mathfrak{I},\boldsymbol{c}}$:

- $a^{\mathfrak{I},\boldsymbol{c}} = \rho$ if $\{a\} \in \rho(\boldsymbol{c})$, for all $a \in \mathsf{ind}_o(\mathcal{K})$;
- $A^{\mathfrak{I},\boldsymbol{c}} = \{\rho \in \mathfrak{R} \mid A \in \rho(\boldsymbol{c})\}$;
- $r^{\mathfrak{I},\boldsymbol{c}} = \{(\rho, \rho') \mid \exists r.D \in \rho(\boldsymbol{c}) \text{ implies } \{D\} \cup \{\neg E \mid \neg\exists r.E \in \rho(\boldsymbol{c})\} \subseteq \rho(\boldsymbol{c}')\}$.

Note that by aligning runs with the witnessing linkages we automatically ensure that each object obtains compatible interpretations in every two related contexts. In particular, whenever $d \in (\langle \boldsymbol{r}.\boldsymbol{C} \rangle D)^{\mathfrak{I},\boldsymbol{c}}$ for some $d \in \Delta$ and $\boldsymbol{c} \in \mathfrak{C}$, there has to exist a context $\boldsymbol{c}' \in \boldsymbol{C}^{\mathcal{J}}$ accessible from $\boldsymbol{c}$ through $\boldsymbol{r}$ in which $d \in D^{\mathfrak{I},\boldsymbol{c}'}$. By the same token, whenever $d \in (\langle \boldsymbol{C} \rangle D)^{\mathfrak{I},\boldsymbol{c}}$, there must be a context $\boldsymbol{c}' \in \boldsymbol{C}^{\mathcal{J}}$ such that $d \in D^{\mathfrak{I},\boldsymbol{c}'}$.

Claim. For each $C \in \mathsf{cl}_o(\mathcal{K})$, $\rho \in \Delta$ and $\boldsymbol{c} \in \mathfrak{C}$, we have that

$$\rho \in C^{\mathfrak{I},\boldsymbol{c}} \text{ iff } C \in \rho(\boldsymbol{c}).$$

Proof of the claim: The proof is by induction on the structure of C. The induction start, where C is a concept name is immediate by the definition of $\mathfrak{I}$. For the induction step, we distinguish the following cases:

- $C = \neg D$ "if:" $\rho \in (\neg D)^{\mathfrak{I},\boldsymbol{c}}$, that is, $\rho \notin D^{\mathfrak{I},\boldsymbol{c}}$. Now, by I.H., $D \in \rho(\boldsymbol{c})$. Then, by definition of a type, $\neg D \in \rho(\boldsymbol{c})$. "only if:" $\neg D \in \rho(\boldsymbol{c})$. By definition of a type $D \notin \rho(\boldsymbol{c})$. Now, by I.H., $\rho \notin D^{\mathfrak{I},\boldsymbol{c}}$. Therefore, by semantics, $\rho \notin (\neg D)^{\mathfrak{I},\boldsymbol{c}}$.

– $C = D_1 \sqcap D_2$ "if:" $\rho \in (D_1 \sqcap D_2)^{\mathfrak{I},\boldsymbol{c}}$, that is, $\rho \in D^{\mathfrak{I},\boldsymbol{c}}$ and $D_2^{\mathfrak{I},\boldsymbol{c}}$. Now, by I.H., $D_1 \in \rho(\boldsymbol{c})$ and $D_2 \in \rho(\boldsymbol{c})$. Therefore, by definition of type, $D_1 \sqcap D_2 \in \rho(\boldsymbol{c})$. "only if:" $D_1 \sqcap D_2 \in \rho(\boldsymbol{c})$, then by definition of a type, $D_1 \in \rho(\boldsymbol{c})$ and $D_2 \in \rho(\boldsymbol{c})$. Moreover, by I.H., $\rho \in D_1^{\mathfrak{I},\boldsymbol{c}}$ and $\rho \in D_2^{\mathfrak{I},\boldsymbol{c}}$. Therefore, by semantics, $\rho \in (D_1 \sqcap D_2)^{\mathfrak{I},\boldsymbol{c}}$.

– $C = \exists r.D$ "if" $\rho \in (\exists r.D)^{\mathfrak{I},\boldsymbol{c}}$, that is, there exists a ρ' such that $(\rho, \rho') \in r^{\mathfrak{I},\boldsymbol{c}}$ and $\rho' \in D^{\mathfrak{I},\boldsymbol{c}}$. Now, by I.H., $\{D\}$. Therefore, by definition of $r^{\mathfrak{I},\boldsymbol{c}}$ and I.H., $\exists r.D \in \rho(\boldsymbol{c})$.

  "only if:" $\exists r.D \in \rho(\boldsymbol{c}) = t$. Now, by definition of **(QS)**, there is a $t' \in O_q \in \pi(\boldsymbol{c})$ such that $D \in \rho'(\boldsymbol{c})$. Now, by definition of $\mathfrak{R}$, there exists a ρ' such that $\rho'(\boldsymbol{c}) = t'$. Then, by I.H., $\rho' \in D^{\mathfrak{I},\boldsymbol{c}}$, and moreover, by definition of $r^{\mathfrak{I},\boldsymbol{c}}$, $(\rho, \rho') \in r^{\mathfrak{I},\boldsymbol{c}}$. Therefore, by semantics, $\rho \in (\exists r.D)^{\mathfrak{I},\boldsymbol{c}}$.

– $C = \{a\}$. $\{a\}^{\mathfrak{I},\boldsymbol{c}}$, that is, $\{a^{\mathfrak{I},\boldsymbol{c}}\}$. Now, by definition of $\mathfrak{I}$, $a^{\mathfrak{I},\boldsymbol{c}} = \rho$ such that $\{a\} \in \rho(\boldsymbol{c})$. Now, by **(QC1)**, such $\rho(\boldsymbol{c})$ exists and it is unique. Therefore, $\rho \in \{a\}^{\mathfrak{I},\boldsymbol{c}}$ iff $\{a\} \in \rho(\boldsymbol{c})$.

  Note that the first condition of coherent runs ensures that $a^{\mathfrak{I},\boldsymbol{c}} = a^{\mathfrak{I},\boldsymbol{c}'}$ for all $\boldsymbol{c} \neq \boldsymbol{c}'$.

– $C = \langle \boldsymbol{r}.\boldsymbol{C}\rangle D$. "if:" $\rho \in (\langle \boldsymbol{r}.\boldsymbol{C}\rangle D)^{\mathfrak{I},\boldsymbol{c}}$, that is, there exists a $\boldsymbol{c}'$ such that $(\boldsymbol{c}, \boldsymbol{c}') \in \boldsymbol{r}^{\mathcal{J}}$ and $\boldsymbol{c}' \in \boldsymbol{C}^{\mathcal{J}}$, and $\rho \in D^{\mathfrak{I},\boldsymbol{c}'}$. By construction $\pi(\boldsymbol{c}')$ is a matching $\boldsymbol{r}$-successor of $\pi(\boldsymbol{c})$ via some linkage λ between $\pi(\boldsymbol{c})$ and $\pi(\boldsymbol{c}')$. Moreover, by construction of runs, $\lambda(\rho(\boldsymbol{c})) = \rho(\boldsymbol{c}')$. Finally, by I.H., $D \in \rho(\boldsymbol{c}')$ and $\boldsymbol{C} \in \pi(\boldsymbol{c}')$. Furthermore, by semantics, $\boldsymbol{c} \in (\exists \boldsymbol{r}.\boldsymbol{C})^{\mathcal{J}}$, and then by the former claim $\exists \boldsymbol{r}.\boldsymbol{C} \in \pi(\boldsymbol{c})$. Therefore, by (**QS3**), $\langle \boldsymbol{r}.\boldsymbol{C}\rangle D \in \rho(\boldsymbol{c})$.

  "only if:" $\langle \boldsymbol{r}.\boldsymbol{C}\rangle D \in \rho(\boldsymbol{c})$. Recall that $\rho(\boldsymbol{c}) = t$, for some object type $t \in O_q \in \pi(\boldsymbol{c})$. Now, by construction there is $\boldsymbol{c}'$ such that $\pi(\boldsymbol{c}') = \langle t_{c_{q'}}, f_{q'}, O_{q'}\rangle$ such that $\pi(\boldsymbol{c}')$ is a matching $\boldsymbol{r}$ successor via λ for $\pi(\boldsymbol{c})$ such that $\boldsymbol{C} \in \pi(\boldsymbol{c}')$ and $D \in \lambda(t)$. Note that, by (**QS3**), such $\boldsymbol{c}$ exists. Now, by definition of the runs, $\lambda(\rho(t))$ is defined. Moreover, by I.H., $\boldsymbol{c}' \in \boldsymbol{C}^{\mathcal{J}}$, $(\boldsymbol{c}, \boldsymbol{c}') \in \boldsymbol{r}^{\mathcal{J}}$ and $\rho \in D^{\mathfrak{I},\boldsymbol{c}'}$. Therefore, by semantics, $\rho \in (\langle \boldsymbol{r}.\boldsymbol{C}\rangle D)^{\mathfrak{I},\boldsymbol{c}}$.

– $C = \langle \boldsymbol{C}\rangle D$. "if:" $\rho \in (\langle \boldsymbol{C}\rangle D)^{\mathfrak{I},\boldsymbol{c}}$, that is, there exists a a $\boldsymbol{c}'$ such that $\boldsymbol{c}' \in \boldsymbol{C}^{\mathcal{J}}$ and $\rho \in D^{\mathfrak{I},\boldsymbol{c}'}$. By construction, $\pi(\boldsymbol{c}')$ is an $\mathfrak{F}_2$-succesor of $\pi(\boldsymbol{c})$ via some link λ between $\pi(\boldsymbol{c})$ and $\pi(\boldsymbol{c}')$. Now, by (**QC4**), $\langle \boldsymbol{C}\rangle D \in \pi(\boldsymbol{c}')$. Moreover, by construction of runs, $\lambda(\rho(\boldsymbol{c})) = \rho(\boldsymbol{c}')$. Finally, by I.H., $D \in \rho(\boldsymbol{c}')$ and $\boldsymbol{C} \in \pi(\boldsymbol{c})$. Therefore, by (**QS2**), $\langle \boldsymbol{C}\rangle D \in \rho(\boldsymbol{c})$.

  "only if:" $\langle \boldsymbol{C}\rangle D \in \rho(\boldsymbol{c})$. Recall that $\rho(\boldsymbol{c}) = t$, for some object type $t \in O_q \in \pi(\boldsymbol{c})$. Now, by construction there is a $\boldsymbol{c}' \in \mathfrak{C}$ such that $\pi(\boldsymbol{c}') = \langle t_{c'_q}, f_{q'}, O_{q'}\rangle$ such that $\pi(\boldsymbol{c}')$ is a matching $\mathfrak{F}_2$ successor via λ for $\pi(\boldsymbol{c})$ such that $\boldsymbol{C} \in \pi(\boldsymbol{c})'$ and $D \in \lambda(t)$. Note that, by definition $\boldsymbol{c}'$ exists. Now, by definition of the runs, $\rho(\lambda(t))$ is defined. Moreover, by I.H., $\boldsymbol{c}' \in \boldsymbol{C}^{\mathcal{J}}$ and $\rho \in D^{\mathfrak{I},\boldsymbol{c}'}$ Therefore, by semantics, $\rho \in (\langle \boldsymbol{C}\rangle D)^{\mathfrak{I},q}$.

This finishes the proof of the claim.

Now, since $\mathfrak{M}$ satisfies conditions (**QC2**) and (**QC3**), all axioms from the object knowledge base $\mathcal{O}$ must be also satisfied. More precisely, by (**QC2**), for every $\boldsymbol{C} : \varphi \in \mathcal{O}$ and $q = \langle t_{c_q}, f_q, O_q\rangle$, if $\boldsymbol{C} \in t_{c_q}$ then $C \sqsubseteq D \in f_q$. Moreover, by (**QC3**), for every $t_o \in O_q$ if $C \in t_o$ then $D \in t_o$. Now, from the previous claims and the construction, we have that there is a $\boldsymbol{c} \in \mathfrak{C}$ with $\pi(\boldsymbol{c}) = q$

Algorithm 3: $\mathfrak{C}^{\mathcal{ALCO}}_{\mathcal{ALCO}}$ KB SATISFIABILITY

Input: A KB $\mathcal{K}$ formulated in $\mathfrak{C}^{\mathcal{ALCO}}_{\mathcal{ALCO}}$
Initialize: $i := 0$; $S_0 := \mathsf{qs}(\mathcal{T})$; $\gamma \in \Gamma$

Eliminate(S_0, γ) {
delete γ from Γ
repeat
$S_{i+1} := \{q \in S_i \mid q$ is realizable in S_i w.r.t $\gamma\}$
until $S_i = S_{i+1}$ }

if exists $q \in S_i$, return *satisfiable*
otherwise, if $\Gamma \neq \emptyset$, then choose a γ from Γ and perform Eliminate(S_0, γ)
otherwise, return *unsatisfiable*

such that $\boldsymbol{c} \in \boldsymbol{C}^{\mathcal{J}}$ and $\rho \in C^{\mathfrak{I},\boldsymbol{c}}$ implies $\rho \in D^{\mathfrak{I},\boldsymbol{c}}$, where $\rho(\boldsymbol{c}) = t_o$. Therefore, $\mathfrak{I}$ is a model of $\mathcal{O}$. Since $(\mathfrak{C}, \cdot^{\mathcal{J}})$ is a model of $\mathcal{C}$ and $\mathfrak{I}$ is a model of $\mathcal{O}$. Therefore, $\mathfrak{I}$ is a model of $\mathcal{K}$.

($\Leftarrow$) This direction follows straightforwardly from the construction. Let $\mathfrak{I} = (\mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\mathcal{I}_i\}_{i\in\mathfrak{C}})$ be a $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-model of $\mathcal{K}$. We construct a quasimodel $\mathfrak{M}$ for $\mathcal{K}$ as follows. We define context types determined by the interpretation $\mathfrak{I}$.

For every $\boldsymbol{c} \in \mathfrak{C}$, we set

$$\mathsf{tp}(\boldsymbol{c}) := \{\boldsymbol{C} \in \mathsf{cl}_c(\mathcal{K}) \mid \boldsymbol{c} \in \boldsymbol{C}^{\mathcal{J}}\};$$

$$f(\boldsymbol{c}) := \{\varphi \in \mathsf{sub}_o(\mathcal{K}) \mid \mathcal{I}_{\boldsymbol{c}} \models \varphi\}.$$

In the same way we use tp to denote object types for objects. For every $d \in \Delta$ and $\boldsymbol{c} \in \mathfrak{C}$, we define $\mathsf{tp}(d, \boldsymbol{c})$ as:

$$\mathsf{tp}(d, \boldsymbol{c}) = \{C \in \mathsf{cl}_o(\mathcal{K}) \mid d \in C^{\mathfrak{I},\boldsymbol{c}}\}$$

Further, for every $\boldsymbol{c} \in \mathfrak{C}$, let $O_{\boldsymbol{c}} = \{\mathsf{tp}(d, \boldsymbol{c}) \mid d \in \Delta\}$ be the set of object types represented in the context $\boldsymbol{c}$. We can then define a quasistate for every $c \in \mathfrak{C}$ as $q_{\boldsymbol{c}} = \langle \mathsf{tp}(\boldsymbol{c}), f(\boldsymbol{c}), O_{\boldsymbol{c}}\rangle$.
Finally, let

$$\mathfrak{M} = \{q_{\boldsymbol{c}} \mid \boldsymbol{c} \in \mathfrak{C}\}.$$

Clearly $\mathfrak{M}$ is a quasimodel for $\mathcal{K}$. In particular, it is guaranteed that for all existential restrictions and context operators occurring in the context and object types from the quasistates, there must exist suitable witnesses and witnessing linkages, and thus that all conditions constituting quasimodels have to be satisfied.

❏

We now describe the *elimination condition* used in our algorithm. Intuitively, we eliminate quasistates that cannot occur in any quasimodel for $\mathcal{K}$.

Definition 5.18. *Let $S \subseteq \mathsf{qs}(\mathcal{K})$ be a set of quasistates for $\mathcal{K}$ and γ a mapping from $\mathsf{ind}_c(\mathcal{K})$ to $\mathsf{qs}(\mathcal{K})$. A quasistate q is* realizable *in S w.r.t. γ if the following conditions are satisfied:*

1. *for all $\boldsymbol{c} \in \mathsf{ind}_c(\mathcal{K})$, if $\{\boldsymbol{c}\} \in t_{c_q} \in q$ then $\gamma(\boldsymbol{c}) = q$.*
2. *q satisfies (**QC1**)-(**QC4**), (**QS1**)-(**QS3**), (**M1**)-(**M2**) in S.*

We denote by Γ the set of all mappings from $\mathsf{ind}_c(\mathcal{K}) \to \mathsf{qs}(\mathcal{K})$. Algorithm 3 decides whether a quasimodel for $\mathcal{K}$ exists by implementing a straightforward extension of the quasistate elimination method [38].

Intuitively, Algorithm 3 *succeeds* if not all the quasistates get eliminated for some $\gamma \in \Gamma$. In such case the result of the elimination is clearly a quasimodel and the search is finished with the answer *satisfiable*. Otherwise, if all quasistates get eliminated, the algorithm selects another mapping γ and repeats the elimination procedure. If none of the mappings allow for a successful termination then clearly no quasimodel exists and the algorithm returns *unsatisfiable*.

The whole algorithm runs in double exponential time in the size of $\mathcal{K}$. To show this, we observe that the following inequalities hold.

$$|\mathsf{cl}_c(\mathcal{K}) \cup \mathsf{cl}_o(\mathcal{K})| \leq 2\ |\mathcal{K}|,$$
$$|\mathsf{ind}_c(\mathcal{K})| \leq |\mathcal{K}|,\ |\ \mathsf{sub}_o(\mathcal{K})\ | \leq |\mathcal{K}|,$$

SIZE OF A QUASISTATE

$$|q| \leq |\mathcal{K}|\ \cdot (|\mathsf{cl}_c(\mathcal{K})| + |\mathsf{sub}_o(\mathcal{K})| + 2^{|\mathsf{cl}_o(\mathcal{K})|}) \leq |\mathcal{K}|\ \cdot (2\ |\mathcal{K}|\ +\ |\mathcal{K}|\ + 2^{2|\mathcal{K}|}),$$

NUMBER OF QUASISTATES IN A QUASIMODEL

$$|\mathsf{qs}(\mathcal{K})| = 2^{|\mathsf{cl}_c(\mathcal{K})|} \cdot 2^{|\mathsf{sub}_o(\mathcal{K})|} \cdot 2^{2^{|\mathsf{cl}_o(\mathcal{K})|}} = 2^{2|\mathcal{K}|} \cdot 2^{|\mathcal{K}|} \cdot 2^{2^{2|\mathcal{K}|}}.$$

Since deciding whether a quasistate can be eliminated at a given stage of Algorithm 3; in particular, checking if there exist appropriate witnesses for it, (**QS1**)-(**QS3**), cannot take more than $|q|^2 \cdot |\mathsf{qs}(\mathcal{K})|$ steps. Thus, a single run of the elimination procedure takes no more than $(|q| \cdot |\mathsf{qs}(\mathcal{K})|)^2$ steps. Finally, note that there can be at most $|\mathfrak{M}|^{|\mathsf{ind}_c(\mathcal{K})|}$ different mappings γ. Therefore, the whole procedure must terminate in time belonging to $O(2^{2^{\mathsf{poly}|\mathcal{K}|}})$.

Theorem 5.12. *KB satisfiability for $\mathfrak{C}^{\mathcal{ALCO}}_{\mathcal{ALCO}}$ for any combination of context operators $\mathfrak{F}_1/\mathfrak{F}_2$ is in* 2EXPTIME.

5.8 Reasoning in Description Logics of Context with only $\mathfrak{F}_2$ Operators

We finalize our study on the computational complexity of DLs of context by considering DLCs allowing only for $\mathfrak{F}_2$ operators. The increase in the computational complexity of DLCs by one exponential, in comparison with that of $\mathcal{ALC}$, can be observed already in the simple DLC

$(\mathbf{K}_n)_{\mathcal{ALC}}$ (*cf.* Theorem 5.6). A possible way of obtaining better-behaved DLCs is to reduce the expressiveness of the underlying formalisms. In particular, we can constraint the allowed type of context operators. It turns out that when only operators of type $\mathfrak{F}_2$ are allowed, the complexity of the KB satisfiability problem for $\mathfrak{C}^{\mathcal{ALCO}}_{\mathcal{ALCO}}$ goes down from 2EXPTIME-complete to NEXPTIME-complete. Moreover, we show that NEXPTIME-hardness holds already for $\mathfrak{C}^{\mathcal{ALC}}_{\mathcal{ALC}}$.

We next demonstrate a NEXPTIME lower bound for KB satisfiability for $\mathfrak{C}^{\mathcal{ALC}}_{\mathcal{ALC}}$ by a reduction of $2^n\times 2^n$-tiling problem. This result shows that the presence of the context language makes indeed the reasoning harder, if compared with that of $\mathbf{S5}_{\mathcal{ALC}}$ for which satisfiability is EXPTIME-complete when modal operators are applied only to concepts and a global TBox is considered [38]. This jump in the complexity is intuitively explained by the need of guessing the interpretation of the context language before finding a model of the object component. We concentrate w.l.o.g. on concept satisfiability w.r.t. global TBoxes.

Theorem 5.13. *Concept satisfiability w.r.t. TBoxes for $\mathfrak{C}^{\mathcal{ALC}}_{\mathcal{ALC}}$ is* NEXPTIME-*hard.*

Proof. The proof is by a reduction of the $2^n\times 2^n$-tiling problem. Let $\mathfrak{T} = (n, T)$ be an instance. Our aim is to construct in polynomial time a TBox $\mathcal{T}_{\mathfrak{T}}$ and a concept $C_{\mathfrak{T}}$, such T tiles the $2^n\times 2^n$ grid *iff* $C_{\mathfrak{T}}$ is satisfiable w.r.t. $\mathcal{T}_{\mathfrak{T}}$.

The encoding utilizes the possibility of constructing and constraining a "diagonal" in the $\mathfrak{C}^{\mathcal{ALC}}_{\mathcal{ALC}}$-models, as depicted in Figure 5.6, representing then the whole tiling in a linear projection. We use the following signature:

- concept names A_{τ_i}, A'_{τ_i} for each $\tau_i \in T$;
- concept names (from the context language) $\boldsymbol{U}_i$ for each $\tau_i \in T$;
- concept names $X_0 \dots X_{2n-1}, Y_0 \dots Y_{2n-1}$ for encoding an exponential counter;
- concept names (from the context language) $\boldsymbol{Z}_0 \dots \boldsymbol{Z}_{2n-1}$;
- auxiliary concept names $\mathsf{RightEdge}, \mathsf{BottomEdge}, \mathsf{DownNeighbor}, \mathsf{StartGrid}, \mathsf{EndGrid}$.

The inclusions (5.25)-(5.28) enforce a 2^{2n}-long chain of individuals, uniquely identifiable by counting concepts X_i, for $i \in (0, 2n-1)$. Moreover, every 2^n-th individual, starting from the beginning of the chain, is an instance of concept RightEdge, marking the right edge of the tiling, while the last 2^n individuals are instances of BottomEdge, marking the bottom of the tiling.

$$\mathsf{StartGrid} \equiv \bigsqcap_{j=0}^{2n-1} \neg X_j, \quad \mathsf{EndGrid} \equiv \bigsqcap_{j=0}^{2n-1} X_j, \quad \neg\mathsf{EndGrid} \sqsubseteq \langle\top\rangle\exists r.\top, \tag{5.25}$$

For every $0 \leq j < i < 2n$:

$$\begin{aligned} \neg X_i \sqcap \neg X_j &\sqsubseteq [\top]\forall r.\neg X_i, \\ X_i \sqcap \neg X_j &\sqsubseteq [\top]\forall r.X_i \end{aligned} \tag{5.26}$$

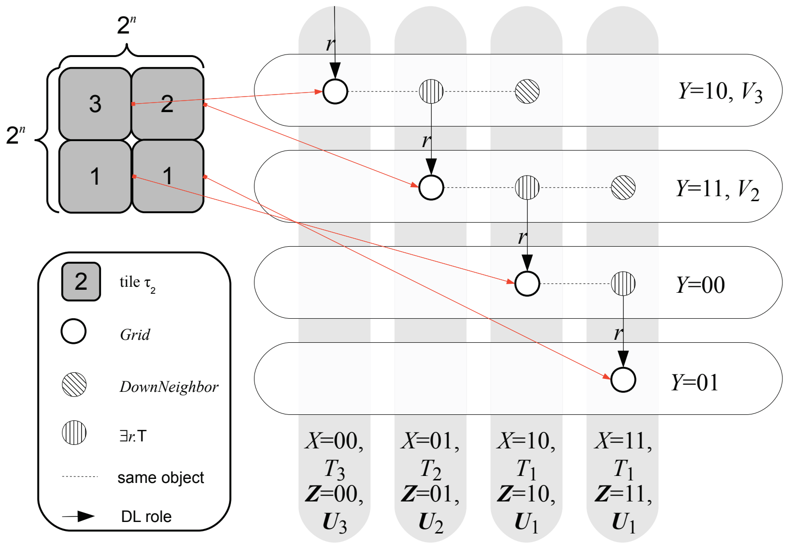

Figure 5.6: Encoding of a $2^n \times 2^n$ tiling in an $\mathfrak{C}^{\mathcal{ALC}}_{\mathcal{ALC}}$-model.

For every $0 \leq j < 2n$:

$$\begin{aligned} \neg X_j \sqcap X_{j-1} \sqcap \ldots \sqcap X_1 &\sqsubseteq [\top]\forall r.X_j, \\ X_j \sqcap X_{j-1} \sqcap \ldots \sqcap X_1 &\sqsubseteq [\top]\forall r.\neg X_j, \end{aligned} \tag{5.27}$$

$$\mathsf{RightEdge} \equiv \bigsqcap_{j=0}^{n-1} X_j, \quad \mathsf{BottomEdge} \equiv \bigsqcap_{j=n}^{2n-1} X_j. \tag{5.28}$$

The values of these counting concepts are then propagated over all the objects in the given context, by involving an interaction with concepts of the metalanguage $\boldsymbol{Z}_i$, for $i \in (0, 2n-1)$ (5.29).

For every $0 \leq i < 2n$:

$$\top \sqsubseteq [\boldsymbol{Z}_i]X_i, \quad \top \sqsubseteq [\neg\boldsymbol{Z}_i]\neg X_i \tag{5.29}$$

Each individual is required to satisfy exactly one concept A_{τ_i}, representing a tile type $\tau_i \in T$ (5.30). This type is then propagated to all individuals in a given world (5.31) and used to adjust the coloring of the left-right neighbors (5.32).

For every $\tau_i, \tau_j \in T$:

$$\top \sqsubseteq (\bigsqcup_{\tau_i} A_{\tau_i}) \sqcap \bigsqcap_{\tau_i \neq \tau_j} \neg(A_{\tau_i} \sqcap A_{\tau_j}), \tag{5.30}$$

$$\top \sqsubseteq [\boldsymbol{U}_i]A_{\tau_i}, \quad \top \sqsubseteq [\neg\boldsymbol{U}_i]\neg A_{\tau_i}, \tag{5.31}$$

$$A_{\tau_i} \sqcap \neg\mathsf{RightEdge} \sqsubseteq [\top]\forall r.(\bigsqcup_{\mathsf{right}(\tau_i)=\mathsf{left}(\tau_j)} A_{\tau_j}). \tag{5.32}$$

For each individual we identify the counter of its down neighbor and encode this value rigidly across the context dimension, by means of concepts Y_i (5.33)-(5.35). In the same manner, the tile type is propagated (5.36).

For every $n \leq j < i < 2n$,

$$\begin{aligned} \neg X_i \sqcap \neg X_j &\sqsubseteq \forall r.[\top]\neg Y_i, \\ X_i \sqcap \neg X_j &\sqsubseteq \forall r.[\top] Y_i, \end{aligned} \tag{5.33}$$

For every $n \leq j < 2n$

$$\begin{aligned} \neg X_j \sqcap X_{j-1} \sqcap \ldots \sqcap X_n &\sqsubseteq \forall r.[\top] Y_j, \\ X_j \sqcap X_{j-1} \sqcap \ldots \sqcap X_n &\sqsubseteq \forall r.[\top]\neg Y_j, \end{aligned} \tag{5.34}$$

For every $1 \leq i \leq n$:

$$\begin{aligned} X_i &\sqsubseteq \forall r.[\top] Y_i, \\ \neg X_i &\sqsubseteq \forall r.[\top]\neg Y_i, \end{aligned} \tag{5.35}$$

$$\neg\mathsf{BottomEdge} \sqcap A_{\tau_i} \sqsubseteq \forall r.[\top] A'_{\tau_i}, \text{ for every } \tau_i \in T. \tag{5.36}$$

Finally, the up-down coloring constraints are enforced whenever the value of Y_i's agrees with the X_i-counter. (5.37-5.38).

$$\mathsf{DownNeighbor} \equiv \bigsqcap_{0 \leq i < 2n} ((X_i \sqcap Y_i) \sqcup (\neg X_i \sqcap \neg Y_i)), \tag{5.37}$$

For every $\tau_i \in T$:

$$\mathsf{DownNeighbor} \sqcap A'_{\tau_i} \sqsubseteq \bigsqcap_{\mathsf{down}(\tau_i) \neq \mathsf{up}(\tau_j)} \neg A_{\tau_j}. \tag{5.38}$$

The TBox $\mathcal{T}_{\mathfrak{T}}$ is defined as the union of the axioms (5.25)-(5.38), moreover we define

$$C_{\mathfrak{T}} = \exists r.(\mathsf{StartGrid} \sqcap A_{\tau_0}).$$

It is easy to see that the size of $\mathcal{T}_{\mathfrak{T}}$ is polynomial in the size of the instance $\mathfrak{T}$, and it is not hard to see from the construction that the following holds.

Proposition 5.14. *$\mathfrak{T}$ tiles the $2^n \times 2^n$ iff $C_{\mathfrak{T}}$ is satisfiable w.r.t. $\mathcal{T}_{\mathfrak{T}}$.*

Let τ be a tiling for $\mathfrak{T}$, that is, a mapping from $2^n \times 2^n$ to T. Define a model $\mathfrak{I} = (\mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\mathcal{I}_c\}_{c \in \mathfrak{C}})$ of $\mathcal{T}_{\mathfrak{T}}$ and $C_{\mathfrak{T}}$ as follows. First, transform τ into $\pi : 2^{2n-1} \mapsto T$, such that for every $(x, y) \in 2^n \times 2^n$, $\tau(x, y) = \pi(y * 2^{n-1} + x)$, and then fix $\mathfrak{C} = \{c_i \mid i \in (0, 2^{2n-1})\}$ and $\Delta = \{d_i \mid i \in (0, 2^{2n})\}$ and ensure that the following interpretation constraints are satisfied:

- $r^{\mathfrak{I}, c_i} = \{(d_{i-1}, d_i) \mid d_{i-1}, d_i \in \Delta\}$,
- $\mathsf{StartGrid}^{\mathfrak{I}, c_0} = \{d_1 \in \Delta\}$, $\mathsf{EndGrid}^{\mathfrak{I}, c_{2^{2n-1}})} = \{d_{2^{2n}} \in \Delta\}$,

- for every $c_i \in \mathfrak{C}$, $\mathsf{DownNeighbor}^{\mathfrak{I},c_i} = \{d_{i-2^{n-1}} \in \Delta\}$,
- for every $\tau_j \in T$ and $i \in (0, 2^{2n-1})$, $A_{\tau_j}^{\mathfrak{I},c_i} = \Delta$, if $\pi(i) = \tau_j$, and else $A_{\tau_j}^{\mathfrak{I},c_i} = \emptyset$.

The interpretations can be straightforwardly extended over the remaining concepts so that $\mathfrak{I}$ is indeed a model of $\mathcal{T}_{\mathfrak{T}}$ where $C_{\mathfrak{T}}^{\mathfrak{I},c_0} = \{d_0\}$.

For the other direction, let $\mathfrak{I}$ be model of $\mathcal{T}_{\mathfrak{T}}$ and $C_{\mathfrak{T}}$. Then, a tiling for $\mathfrak{T}$ can be retrieved from $\mathfrak{I}$ by mapping the diagonal of the model on the $2^n \times 2^n$ grid, where the type of a tile in the grid is determined by the unique concept A_{τ_i} satisfied by the individual in the chain. The coloring constraints have to be satisfied by the construction of the encoding.

❑

Now, we present an algorithm for KB satisfiability for $\mathfrak{C}_{\mathcal{ALCO}}^{\mathcal{ALCO}}$ with only $\mathfrak{F}_2$ context operators, yielding a NEXPTIME matching upper bound. As in the case of the decision procedure for KB satisfiability in $\mathfrak{C}_{\mathcal{ALCO}}^{\mathcal{ALCO}}$ allowing for $\mathfrak{F}_1$ and $\mathfrak{F}_2$ operators (*cf.* Section 5.7), the proposed decision procedure is essentially a variant of type-based techniques.

Let $\mathcal{K} = (\mathcal{C}, \mathcal{O})$ be a KB formulated in $\mathfrak{C}_{\mathcal{ALCO}}^{\mathcal{ALCO}}$ with only $\mathfrak{F}_2$ context operators whose satisfiability is to be decided. Further, we apply the following replacements of all respective (sub)formulas with their equivalents:

$$\begin{array}{rclcrcl} C(a) & \Rightarrow & \{a\} \sqsubseteq C, & \quad & r(a,b) & \Rightarrow & \{a\} \sqsubseteq \exists r.\{b\}, \\ \boldsymbol{C}(\boldsymbol{a}) & \Rightarrow & \{\boldsymbol{a}\} \sqsubseteq \boldsymbol{C}, & & \boldsymbol{r}(\boldsymbol{a},\boldsymbol{b}) & \Rightarrow & \{\boldsymbol{a}\} \sqsubseteq \exists \boldsymbol{r}.\{\boldsymbol{b}\}. \end{array}$$

The following notation is used to denote particular sets of *object* symbols occurring in $\mathcal{K}$:

- $\mathsf{cl}_o(\mathcal{K})$: set of object concepts that occur in $\mathcal{K}$, closed under subconcepts and negation;
- $\mathsf{ind}_o(\mathcal{K})$: set of object individual names that occur in $\mathcal{K}$;
- $\mathsf{sub}_o(\mathcal{K})$: set of axioms from $\{\varphi \mid \boldsymbol{C} : \varphi \in \mathcal{O} \text{ for any } \boldsymbol{C}\}$.

Moreover, we use $\mathsf{cl}_c(\mathcal{K})$, $\mathsf{ind}_c(\mathcal{K})$ to denote the analogous sets over *context* concepts and roles occurring in $\mathcal{K}$. We proceed to define a *context* and an *object type*.

Definition 5.19. *A* context type for $\mathcal{K}$ *is a subset* $t_c \subseteq \mathsf{cl}_c(\mathcal{K})$ *such that the following hold:*

- $\boldsymbol{C} \in t_c$ *iff* $\neg\boldsymbol{C} \notin t_c$*, for all* $\boldsymbol{C} \in \mathsf{cl}_c(\mathcal{K})$*,*
- $\boldsymbol{C} \sqcap \boldsymbol{D} \in t_c$ *iff* $\{\boldsymbol{C}, \boldsymbol{D}\} \subseteq t_c$*, for all* $\boldsymbol{C} \sqcap \boldsymbol{D} \in \mathsf{cl}_c(\mathcal{K})$.

Furthermore, an object type for $\mathcal{K}$ *is a subset* $t_o \subseteq \mathsf{cl}_o(\mathcal{K})$ *such that the following hold:*

- $C \in t_o$ *iff* $\neg C \notin t_o$*, for all* $C \in \mathsf{cl}_o(\mathcal{K})$*,*
- $C \sqcap D \in t_o$ *iff* $\{C, D\} \subseteq t_o$*, for all* $C \sqcap D \in \mathsf{cl}_o(\mathcal{K})$.

We denote the set of all context types for $\mathcal{K}$ by $\mathsf{tp}_c(\mathcal{K})$ and the set of all object types for $\mathcal{K}$ is denoted by $\mathsf{tp}_o(\mathcal{K})$. Furthermore, we denote by $\mathbf{m}(t_o)$ the set of all object concepts containing context operators in $t_o \in \mathsf{tp}_o(\mathcal{K})$, that is, $\mathbf{m}(t_o) = \{\langle \boldsymbol{C}\rangle D, \neg\langle \boldsymbol{C}\rangle D \in t_o \mid \boldsymbol{C} \in \mathsf{cl}_c(\mathcal{K}), D \in \mathsf{cl}_o(\mathcal{K})\}$. We say that two object types $t_o, t'_o \in \mathsf{tp}_o(\mathcal{K})$ are *matching $\mathfrak{F}_2$-successors iff* $\mathbf{m}(t_o) = \mathbf{m}(t'_o)$.

We next establish the conditions of a set of context types representing a model of the context language. We say that $S \subseteq \mathsf{tp}_c(\mathcal{K})$ is *$\mathcal{C}$-admissible* if the following conditions are satisfied:

(**A1**) for every $\boldsymbol{C} \sqsubseteq \boldsymbol{D} \in \mathcal{C}$ and $t_c \in S$, if $\boldsymbol{C} \in t_c$ then $\boldsymbol{D} \in t_c$;

(**A2**) for every $\boldsymbol{a} \in \mathsf{ind}_c(\mathcal{K})$ there exists a unique $t_c \in S$ such that $\{\boldsymbol{a}\} \in t_c$;

(**A3**) for every $\exists \boldsymbol{s}.\boldsymbol{C} \in \mathsf{cl}_c(\mathcal{K})$ and $t_c \in S$, if $\exists \boldsymbol{s}.\boldsymbol{C} \in t_c$ then there exists $t_{c'} \in S$, such that $\{\boldsymbol{C}\} \cup \{\neg \boldsymbol{E} \mid \neg\exists \boldsymbol{s}.\boldsymbol{E} \in t_c\} \subseteq t_{c'}$.

We are now in the position to define a *context structure* containing all the pieces necessary for reconstructing a single $\mathfrak{C}^{\mathcal{ALCO}}_{\mathcal{ALCO}}$-interpretation.

Definition 5.20. *A* context structure *$\langle S, \mathfrak{S}\rangle$ for $\mathcal{K}$ is a pair consisting of a set $S \subseteq \mathsf{tp}_c(\mathcal{K})$ of context types for $\mathcal{K}$ and a non-empty set $\mathfrak{S}$ of tuples of the form $\langle t_c, f, \nu\rangle$, where $t_c \in S$, $f \subseteq \mathsf{sub}_o(\mathcal{K})$, $\nu : \mathsf{ind}_o(\mathcal{K}) \to \mathsf{tp}_o(\mathcal{K})$ assigns unique object types to individual object names such that the following conditions are satisfied:*

*(**CS1**) for every $t_c \in S$, there exists at least one $\langle t_c, f, \nu\rangle$ in $\mathfrak{S}$;*

*(**CS2**) for every $\boldsymbol{a} \in \mathsf{ind}_c(\mathcal{K})$ there is at most one $\langle t_c, f, \nu\rangle \in \mathfrak{S}$;*

*(**CS3**) S is $\mathcal{C}$-admissible;*

*(**CS4**) for every $\langle t_c, f, \nu\rangle \in \mathfrak{S}$ and $\boldsymbol{C} : \varphi \in \mathcal{O}$, if $\boldsymbol{C} \in t_c$ then $\varphi \in f$.*

However, not all such interpretations might correspond to a genuine $\mathfrak{C}^{\mathcal{ALCO}}_{\mathcal{ALCO}}$-model. To filter out the proper ones some additional conditions need to be imposed. These are, later on, introduced in the notion of *quasimodel candidate*, and further, in the notion of *quasimodel*.

Definition 5.21. *A* quasimodel candidate *$\mathfrak{Q}^S_{\mathfrak{S}}$ for $\mathcal{K}$, where $\langle S, \mathfrak{S}\rangle$ is a context structure for $\mathcal{K}$, is a set of pairs $\langle k, t_o\rangle$ such that $k \in \mathfrak{S}$ and $t_o \in \mathsf{tp}_o(\mathcal{K})$ satisfying the following conditions:*

*(**QC1**) for every $k \in \mathfrak{S}$ with $k = \langle t_c, f, \nu\rangle$ and $a \in \mathsf{ind}_o(\mathcal{K})$, it holds that $\langle k, \nu(a)\rangle \in \mathfrak{Q}^S_{\mathfrak{S}}$;*

for every $\langle k, t_o\rangle \in \mathfrak{Q}^S_{\mathfrak{S}}$, with $k = \langle t_c, f, \nu\rangle$, the following hold:

*(**QC2**) if $\boldsymbol{C} \in t_c$ and $D \in t_o$ then $\langle \boldsymbol{C}\rangle D \in t_o$, for all $\langle \boldsymbol{C}\rangle D \in \mathsf{cl}_o(\mathcal{K})$;*

*(**QC3**) for every $k' \in \mathfrak{S}$, there is some $\langle k', t'_o\rangle \in \mathfrak{Q}^S_{\mathfrak{S}}$ such that t_o, t'_o are matching $\mathfrak{F}_2$-successors;*

*(**QC4**) if $\langle \boldsymbol{C}\rangle D \in t_o$ then there is $\langle k', t'_o\rangle \in \mathfrak{Q}^S_{\mathfrak{S}}$, such that $k' = \langle t'_c, f', \nu'\rangle$, $\boldsymbol{C} \in t'_c$, $D \in t'_o$ and t_o, t'_o are matching $\mathfrak{F}_2$-successors. Moreover, if $t_o \neq t'_o$ then $k' \neq k$;*

(**QC5**) *for every* $\exists r.C \in t_o$ *there is* $\langle k, t'_o \rangle \in \mathfrak{Q}^S_{\mathfrak{S}}$*, such that* $\{C\} \cup \{\neg E \mid \neg\exists r.E \in t_o\} \subseteq t'_o$.

Now, we have the main building blocks to define the *quasimodel* structure. Intuitively, this structure codifyfies an abstraction of a $\mathfrak{C}^{\mathcal{ALCO}}_{\mathcal{ALCO}}$ model.

Definition 5.22. *A quasimodel candidate* $\mathfrak{Q}^S_{\mathfrak{S}}$ *for* $\mathcal{K}$ *is called a* quasimodel for $\mathcal{K}$ if *the following conditions are satisfied:*

(**QM1**) *for every* $\langle k, t_o \rangle \in \mathfrak{Q}^S_{\mathfrak{S}}$ *with* $k = \langle t_c, f, \nu \rangle$ *and* $a \in \mathsf{ind}_o(\mathcal{K})$*, if* $t_o = \nu(a)$*, then* $\{a\} \in t_o$*;*

(**QM2**) *for every* $k \in \mathfrak{S}$ *with* $k = \langle t_c, f, \nu \rangle$*,* $C \sqsubseteq D \in f$ *if for every* $\langle k, t_o \rangle \in \mathfrak{Q}^S_{\mathfrak{S}}$ *if* $C \in t_o$ *then* $D \in t_o$*, for every* $C \sqsubseteq D \in \mathsf{sub}_o(\mathcal{K})$.

The next lemma shows that to decide satisfiability of $\mathcal{K}$, it suffices to check the existence of a quasimodel.

Lemma 5.15. *There is a quasimodel for* $\mathcal{K}$ iff $\mathcal{K}$ *has a model* .

Proof. ($\Rightarrow$) Let $\mathfrak{Q}^S_{\mathfrak{S}}$ be a quasimodel for $\mathcal{K}$. In the following steps we define a model $\mathfrak{I} = (\mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\mathcal{I}_i\}_{i \in \mathfrak{C}})$ of $\mathcal{K}$. The interpretation of the context dimension follows immediately from the definition of the context structure. In particular, note that, since $\langle S, \mathfrak{S} \rangle$ is a context structure, then S must be a $\mathcal{C}$-admissible set of context types. We use the mapping π from $\mathfrak{C}$ to $\mathfrak{S}$. Moreover, for $k = \langle t_c, f, \nu \rangle$, we denote by '$\boldsymbol{C} \in k$' the fact that $\boldsymbol{C} \in t_c$. Now, we define $\mathfrak{C}$ as follows:

- for each $\boldsymbol{a} \in \mathsf{ind}_c$, add $\boldsymbol{c}$ to $\mathfrak{C}$ and set $\pi(\boldsymbol{c}) = k$ such that $\{\boldsymbol{a}\} \in k$;
- for each $k \in \langle S, \mathfrak{S} \rangle \backslash \{k' \mid \{\boldsymbol{c}\} \in k'$ for some $\boldsymbol{c} \in \mathsf{ind}_c(\mathcal{K})\}$, add a fresh $\boldsymbol{c}$ to $\mathfrak{C}$, and set $\pi(\boldsymbol{c}) = k$.

Finally, we set the interpretation for concept and role names as follows:

- $\boldsymbol{a}^{\mathcal{J}} = \boldsymbol{c}$ if $\{\boldsymbol{a}\} \in \pi(\boldsymbol{c})$, for all $\boldsymbol{a} \in \mathsf{ind}_c(\mathcal{K})$;
- $\boldsymbol{A}^{\mathcal{J}} = \{\boldsymbol{c} \in \mathfrak{C} \mid \boldsymbol{A} \in \pi(\boldsymbol{c})\}$;
- $\boldsymbol{r}^{\mathcal{J}} = \{(\boldsymbol{c}, \boldsymbol{c}') \in \mathfrak{C} \times \mathfrak{C} \mid \exists \boldsymbol{r}.\boldsymbol{C} \in \pi(\boldsymbol{c})$ implies $\{\boldsymbol{C}\} \cup \{\neg \boldsymbol{E} \mid \neg\exists \boldsymbol{r}.\boldsymbol{E} \in \pi(\boldsymbol{c})\} \subseteq \pi(\boldsymbol{c}')\}$.

We can now straightforwardly prove the following claim.

Claim. For all $\boldsymbol{c} \in \mathfrak{C}$ and $\boldsymbol{C} \in \mathsf{cl}_c(\mathcal{K})$, we have that

$$\boldsymbol{c} \in \boldsymbol{C}^{\mathcal{J}} \text{ iff } \boldsymbol{C} \in \pi(\boldsymbol{c}).$$

Proof of the claim: The proof is by induction on the structure of $\boldsymbol{C}$. The induction start, where C is a concept name is immediate by the definition of $\mathcal{J}$. For the induction step, we distinguish the following cases:

- $\boldsymbol{C} = \neg \boldsymbol{D}$, $\boldsymbol{C} = \boldsymbol{C}_1 \sqcap \boldsymbol{C}_2$ standard from the definition of type.

– $\boldsymbol{C} = \{\boldsymbol{a}\}$. By semantics, $\{\boldsymbol{a}\}^{\mathcal{J}} = \{\boldsymbol{a}^{\mathcal{J}}\}$. By definition of $\mathcal{J}$, $\boldsymbol{a}^{\mathcal{J}} = \boldsymbol{c}$ such that $\{\boldsymbol{a}\} \in \pi(\boldsymbol{c})$. Finally, by (**A2**) and (**CS2**), such $\pi(\boldsymbol{c})$ exists and it is unique. Therefore, $\{\boldsymbol{a}\} \in \pi(\boldsymbol{c})$ iff $\boldsymbol{c} \in \{\boldsymbol{a}\}^{\mathcal{J}}$.

– $\boldsymbol{C} = \exists \boldsymbol{r}.\boldsymbol{C}$ "if:" $\boldsymbol{c} \in (\exists \boldsymbol{r}.\boldsymbol{C})^{\mathcal{J}}$, that is, there exists a $\boldsymbol{c}'$ such that $(\boldsymbol{c}, \boldsymbol{c}') \in \boldsymbol{r}^{\mathcal{J}}$ and $\boldsymbol{c}' \in \boldsymbol{C}^{\mathcal{J}}$. Now by I.H., $\{\boldsymbol{C}\}$. Therefore, by I.H., and definition of $r^{\mathcal{J}}$, $\exists \boldsymbol{r}.\boldsymbol{C} \in \pi(\boldsymbol{c})$.

  "only if:" $\exists \boldsymbol{r}.\boldsymbol{C} \in \pi(\boldsymbol{c})$. Now, since S is $\mathcal{C}$-admissible (**CS3**), then there is a $t'_c \in S$ such that $\{\boldsymbol{C}\} \cup \{\neg \boldsymbol{E} \mid \neg\exists \boldsymbol{r}.\boldsymbol{E} \in \pi(\boldsymbol{c})\} \subseteq t'_c$ (**A3**). Furthermore, by (**CS2**), there exists a $k = \langle t'_c, f, \nu\rangle \in \mathfrak{S}$. Now, by construction of $\mathcal{J}$, there exists a $\boldsymbol{c}' \in \mathfrak{C}$ with $\pi(\boldsymbol{c}') = k$. By I.H., $\boldsymbol{c}' \in (\{\boldsymbol{C}\} \cup \{\neg \boldsymbol{E} \mid \neg\exists \boldsymbol{r}.\boldsymbol{E} \in \pi(\boldsymbol{c})\})^{\mathcal{J}}$, and, by definition of $\boldsymbol{r}^{\mathcal{J}}$, $(\boldsymbol{c}, \boldsymbol{c}') \in \boldsymbol{r}^{\mathcal{J}}$. Therefore, by semantics, $\boldsymbol{c} \in (\exists \boldsymbol{r}.\boldsymbol{C})^{\mathcal{J}}$

This finishes the proof of the claim.

Since S is $\mathcal{C}$-admissible for all $\boldsymbol{C} \sqsubseteq \boldsymbol{D}$, and $t_c \in S$, if $\boldsymbol{C} \in t_c$, then $\boldsymbol{D} \in t_c$. Thus, by the previous claim, for all $\boldsymbol{c} \in \mathfrak{C}$, $\boldsymbol{c} \in \boldsymbol{C}^{\mathcal{J}}$ implies $\boldsymbol{c} \in \boldsymbol{D}^{\mathcal{J}}$. Therefore, $\mathcal{J}$ is a model of $\mathcal{C}$.

Now, consider the object dimension. For every $\boldsymbol{c} \in \mathfrak{C}$ with $\pi(\boldsymbol{c}) = k$, we fix the set of object types $T_k = \{t_o \mid \langle k, t_o\rangle \in \mathfrak{Q}^S_{\mathfrak{S}}\}$ realized in this context.

A *run* ρ through $\mathfrak{Q}^S_{\mathfrak{S}}$ is a *choice function* which to every $\boldsymbol{c} \in \mathfrak{C}$ with $\pi(\boldsymbol{c}) = k$ assigns a single type from T_k such that the following hold:

(**C1**) for every $\boldsymbol{c}, \boldsymbol{c}' \in \mathfrak{C}$ it is the case that $\rho(\boldsymbol{c}), \rho(\boldsymbol{c}')$ are matching $\mathfrak{F}_2$-successors;

(**C2**) for every $\boldsymbol{c} \in \mathfrak{C}$, if $\langle \boldsymbol{C}\rangle D \in \rho(\boldsymbol{c})$ then there is $\boldsymbol{c}' \in \mathfrak{C}$, such that $\boldsymbol{C} \in \pi(\boldsymbol{c}')$ and $D \in \rho(\boldsymbol{c}')$.

A set $\mathfrak{R}$ of runs through $\mathfrak{Q}^S_{\mathfrak{S}}$ is called *coherent* if the following conditions are satisfied:

– for every $\boldsymbol{c} \in \mathfrak{C}$ with $\pi(\boldsymbol{c}) = k$ and $t_o \in T_k$, there is a $\rho \in \mathfrak{R}$ such that $\rho(\boldsymbol{c}) = t_o$;

– for every $a \in \mathsf{ind}_o(\mathcal{K})$ and $\boldsymbol{c} \in \mathfrak{C}$ with $\pi(\boldsymbol{c}) = \langle t_c, f, \nu\rangle$ there is a unique $\rho \in \mathfrak{R}$, such that $\rho(\boldsymbol{c}) = \nu(a)$.

Next, we define the interpretation of the object dimension as follows. First, fix the object domain as $\Delta := \mathfrak{R}$, with $\mathfrak{R}$ a set of coherent runs through $\mathfrak{Q}^S_{\mathfrak{S}}$. Then, set the interpretation function $\cdot^{\mathfrak{I},\boldsymbol{c}}$ as follows:

– $a^{\mathfrak{I},\boldsymbol{c}} = \rho$ if $\nu(a) = \rho(\boldsymbol{c})$, for every $a \in \mathsf{ind}_o(\mathcal{K})$;

– $A^{\mathfrak{I},\boldsymbol{c}} = \{\rho \in \Delta \mid A \in \rho(\boldsymbol{c})\}$;

– $r^{\mathfrak{I},\boldsymbol{c}} = \{(\rho, \rho') \in \Delta \times \Delta \mid \exists r.C \in \rho(\boldsymbol{c}) \text{ implies } \{C\} \cup \{\neg E \mid \neg\exists r.E \in \rho(\boldsymbol{c})\} \subseteq \rho'(\boldsymbol{c})\}$.

Claim For all $\rho \in \Delta$, $\boldsymbol{c} \in \mathfrak{C}$ and $C \in \mathsf{cl}_o(\mathcal{K})$, we have that

$$\rho \in C^{\mathfrak{I},\boldsymbol{c}} \text{ iff } C \in \rho(\boldsymbol{c}).$$

Proof of the claim: The proof is by induction on the structure of C. The induction start, where C is a concept name is immediate by the definition of $\mathfrak{I}$. For the induction step, we distinguish the following

- $C = \neg D$ and $C = D_1 \sqcap D_2$ standard from the definition of type.
- $C = \{a\}$. $\{a\}^{\mathfrak{I},\boldsymbol{c}}$, that is, $\{a^{\mathfrak{I},\boldsymbol{c}}\}$. Now, by definition of $\mathfrak{I}$, $a^{\mathfrak{I},\boldsymbol{c}} = \rho$ such that $\nu(a) = \rho(\boldsymbol{c})$. Moreover, by **(QM1)**, $\{a\} \in \rho(\boldsymbol{c})$. Finally, by **(QC1)**, such $\nu(a)$ exists and it is unique. Therefore, $\rho \in \{a\}^{\mathfrak{I},\boldsymbol{c}}$ iff $\{a\} \in \rho(\boldsymbol{c})$.

  Note that the second condition of coherent runs ensures that $a^{\mathfrak{I},\boldsymbol{c}} = a^{\mathfrak{I},\boldsymbol{c}'}$, for all $\boldsymbol{c} \neq \boldsymbol{c}'$.
- $C = \exists r.D$ "if" $\rho \in (\exists r.D)^{\mathfrak{I},\boldsymbol{c}}$, that is, there exists a ρ' such that $(\rho, \rho') \in r^{\mathfrak{I},\boldsymbol{c}}$ and $\rho' \in D^{\mathfrak{I},\boldsymbol{c}}$. Now, by I.H., $\{D\} \cup \{\neg E \mid \neg\exists r.E \in \rho(\boldsymbol{c})\} \subseteq \rho'(\boldsymbol{c}')$. Therefore, by definition of $r^{\mathfrak{I},\boldsymbol{c}}$, $\exists r.D \in \rho(\boldsymbol{c})$.

  "only if:" $\exists r.D \in \rho(\boldsymbol{c}) = t_o \in T_k$, where $\pi(\boldsymbol{c}) = k$. Now, by (**QC5**), there is a $t'_o \in T_k$ such that $\{D\} \cup \{\neg E \mid \neg\exists r.E \in t_o\} \subseteq t'_o$. Then, by I.H., $\rho' \in D^{\mathfrak{I},\boldsymbol{c}}$, and moreover by definition of $r^{\mathfrak{I},\boldsymbol{c}}$, $(\rho, \rho') \in r^{\mathfrak{I},\boldsymbol{c}}$. Therefore, by semantics, $\rho \in (\exists r.D)^{\mathfrak{I},\boldsymbol{c}}$.
- $C = \langle \boldsymbol{C} \rangle D$. "if" $\rho \in (\langle \boldsymbol{C} \rangle D)^{\mathfrak{I},\boldsymbol{c}}$, that is, there is a $\boldsymbol{c}' \in \mathfrak{C}$ such that $\boldsymbol{c}' \in \boldsymbol{C}^{\mathcal{J}}$ and $\rho \in D^{\mathfrak{I},\boldsymbol{c}'}$. By I.H., $D \in \rho(\boldsymbol{c}')$, and by the former claim $\boldsymbol{C} \in \pi(\boldsymbol{c}')$. Hence, by (**QC2**), $\langle \boldsymbol{C} \rangle D \in \rho(\boldsymbol{c}')$. Now, by (**QC3**) and definition of $\mathfrak{R}$ (**C1**), $\rho(\boldsymbol{c})$ and $\rho(\boldsymbol{c}')$ are matching $\mathfrak{F}_2$-successors. Therefore, by (**QC4**), $\langle \boldsymbol{C} \rangle D \in \rho(\boldsymbol{c})$.

  "only if:" $(\langle \boldsymbol{C} \rangle D) \in \rho(\boldsymbol{c}) = t_o \in T_k$, where $\pi(\boldsymbol{c}) = k$. Then, by (**QC4**), there exists a $\langle k', t'_o \rangle$ such that $\boldsymbol{C} \in k'$, and $D \in t'_o$, and t_o, t'_o are matching $\mathfrak{F}_2$-successors. By construction, there is a $\boldsymbol{c}' \in \mathfrak{C}$ such that $\pi(\boldsymbol{c}') = k'$, and by I.H., $\boldsymbol{c}' \in \boldsymbol{C}^{\mathcal{J}}$. Moreover, by (**C2**) from $\mathfrak{R}$, $\rho(\boldsymbol{c}') = t'_o$, and by I.H., $\rho \in D^{\mathfrak{I},\boldsymbol{c}'}$. Therefore, by semantics, $\rho \in (\langle \boldsymbol{C} \rangle D)^{\mathfrak{I},\boldsymbol{c}}$.

This finishes the proof of the claim.

Now, by (**CS4**), for every $\boldsymbol{C} : \varphi \in \mathcal{O}$ and $k = \langle t_c, f, \nu \rangle \in \mathfrak{S}$, if $\boldsymbol{C} \in t_c$, then $\varphi \in f$. Moreover, by (**QM2**), for every $\langle k, t_o \rangle$, and $C \sqsubseteq D \in f$, if $C \in t_o$, then $D \in t_o$. Now, from the construction we have that for every k we have a $\boldsymbol{c} \in \mathfrak{C}$, such that $\pi(\boldsymbol{c}) = k$. Furthermore, by the previous claims, we have that $\boldsymbol{c} \in \boldsymbol{C}^{\mathcal{J}}$, and $\rho \in C^{\mathfrak{I},\boldsymbol{c}}$ implies $\rho \in D^{\mathfrak{I},\boldsymbol{c}}$, where $\rho(\boldsymbol{c}) = t_o$. Therefore, $\mathcal{J}$ is a model of $\mathcal{K}$.

($\Leftarrow$) This direction follows straightforwardly from the construction. Let $\mathfrak{I} = (\mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\mathcal{I}_i\}_{i \in \mathfrak{C}})$ be a model of $\mathcal{K}$. We define context and object types determined by the model $\mathfrak{I}$.
For every $\boldsymbol{c} \in \mathfrak{C}$, we set

$$\mathsf{tp}(\boldsymbol{c}) := \{\boldsymbol{C} \in \mathsf{cl}_c(\mathcal{K}) \mid \boldsymbol{c} \in \boldsymbol{C}^{\mathcal{J}}\};$$

$$f(\boldsymbol{c}) := \{\varphi \in \mathsf{sub}_o(\mathcal{K}) \mid \mathcal{I}_{\boldsymbol{c}} \models \varphi\}.$$

In the same way we use tp to denote the object types. For every $d \in \Delta$ and $\boldsymbol{c} \in \mathfrak{C}$

$$\mathsf{tp}(d, \boldsymbol{c}) := \{D \in \mathsf{cl}_o(\mathcal{K}) \mid d \in D^{\mathfrak{I},\boldsymbol{c}}\}$$

and for every $a \in \mathsf{ind}_o(\mathcal{K})$, set the mapping $\nu_{\boldsymbol{c}}(a) := \mathsf{tp}(a^{\mathfrak{I},\boldsymbol{c}}, \boldsymbol{c})$. Fix

$$S = \{t \mid t = \mathsf{tp}(\boldsymbol{c}), \boldsymbol{c} \in \mathfrak{C}\} \text{ and } \mathfrak{S} = \{\langle \mathsf{tp}(\boldsymbol{c}), f(\boldsymbol{c}), \nu_c \rangle \mid c \in \mathfrak{C}\}.$$

Finally, we can define the quasimodel:

$$\mathfrak{Q}^S_{\mathfrak{S}} = \{\langle \mathsf{tp}(\boldsymbol{c}), \mathsf{tp}(d, \boldsymbol{c})\rangle \mid \boldsymbol{c} \in \mathfrak{C}, d \in \Delta\}.$$

Clearly, all conditions (**QC1**)-(**QC5**) and (**QM1**)-(**QM3**) have to be satisfied. Therefore, $\mathfrak{Q}^S_{\mathfrak{S}}$ is a quasimodel for $\mathcal{K}$.

❑

Note that the size of a quasimodel candidate is exponentially bounded in the size of $\mathcal{K}$:

$$|\mathsf{tp}_o(\mathcal{K})| \leq 2^{|\mathsf{cl}_o(\mathcal{K})|} \leq 2^{2|\mathcal{K}|}, \quad |\mathsf{tp}_c(\mathcal{K})| \leq 2^{|\mathsf{cl}_c(\mathcal{K})|} \leq 2^{2|\mathcal{K}|}$$
$$|\mathfrak{S}| \leq |\mathsf{tp}_c(\mathcal{K})| \cdot |\mathsf{tp}_o(\mathcal{K})^{|\mathsf{ind}_o(\mathcal{K})|}| \leq 2^{2|\mathcal{K}|^2} + 4|\mathcal{K}|$$
$$|\mathfrak{Q}^S_{\mathfrak{S}}| \leq |\mathfrak{S}| \cdot |\mathsf{tp}_o(\mathcal{K})| \leq 2^{2|\mathcal{K}|^2+6|\mathcal{K}|}$$

Since the maximum size of a single tuple in a quasimodel candidate is polynomial in $|\mathcal{K}|$ therefore the maximum size of a quasimodel is never greater than $2^{p(|\mathcal{K}|)}$, where p is a fixed polynomial.
The simplest brute-force NEXPTIME algorithm for checking satisfiability of $\mathcal{K}$ first guesses a quasimodel and then checks whether all conditions (**CS**x), (**QC**x) and (**QM**x) are satisfied. Clearly, such a check can be accomplished in a polynomial time in the size of the quasimodel, and thus in at most an exponential time in the size of $\mathcal{K}$. This combined with Theorem 5.13 provides us with the following.

Theorem 5.16. *KB satisfiability for $\mathfrak{C}^{\mathcal{ALCO}}_{\mathcal{ALCO}}$ with only context operators $\mathfrak{F}_2$ is* NEXPTIME-*complete.*

In the light of the relation between $\mathbf{S5}_{\mathcal{ALCO}}$ and $\mathfrak{C}^{\mathcal{ALCO}}_{\mathcal{ALCO}}$ with only context operators $\mathfrak{F}_2$ (*cf.* Theorem 5.10) and Theorem 4.1, we obtain the following:

Theorem 5.17. *Concept satisfiability w.r.t. TBoxes for $\mathbf{S5}_{\mathcal{ALCO}}$ without temporal roles is* NEXPTIME-*complete.*

5.9 Application Scenarios

In this section, we commit ourselves to show the applicability of DLCs to diverse problems. The typical uses of contexts in knowledge systems, as argued by Bouquet *et al.*[25], can be classified into two categories, reflecting two generic knowledge representation scenarios: *divide-and-conquer* and *compose-and-conquer*. The first one concerns the problem of representing inherently contextualized knowledge, while the latter, the problem of integrating multiple, non-contextualized knowledge models in a context-sensitive manner. In what follows, we make these two scenarios more concrete by grounding them in the practice of knowledge engineering, explain how they translate into the DL setting, and outline how they can be supported using DLCs.

$\mathcal{C}$:	$\boldsymbol{Country}(\boldsymbol{germany})$	(1)
	$\boldsymbol{neighbor}(\boldsymbol{france}, \boldsymbol{germany})$	(2)
$\mathcal{O}$:	$\boldsymbol{germany} : \exists hasParent.Citizen(john)$	(3)
	$\boldsymbol{Country} : \exists hasParent.Citizen \sqsubseteq Citizen$	(4)
	$\boldsymbol{france} : \langle \boldsymbol{neighbor}.\boldsymbol{Country} \rangle Citizen \sqsubseteq NoVisaRequirement$	(5)

Table 5.1: A sample knowledge base in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ with $\mathfrak{F}_1$-operators.

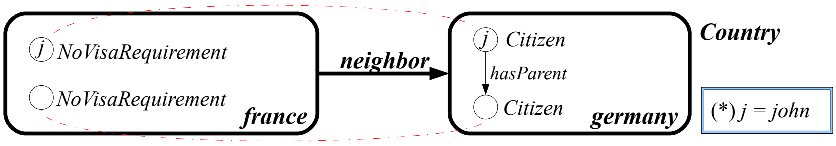

Figure 5.7: A possible model of the CKB in Table 5.1.

5.9.1 Divide-and-conquer

Picture a complex application domain and a modeler intending to formally represent knowledge about this domain in a possibly generic, application-agnostic manner. His task is to construct a representation model that can be reused for different purposes and in different situations, always providing adequate information under the specified conditions. The divide-and-conquer philosophy builds on the observation that in most such cases knowledge is likely to be inherently contextualized, i.e., implicitly partitioned over a collection of interrelated contextual states, which must be taken into account when reasoning about the domain, as they determine which information applies in a given situation. The challenge for the modeler is then to elicit this underlying context structure and explicitly represent it in the model, so that the context-dependency of knowledge is faithfully reflected and operationalized in the system. From the perspective of the DL paradigm, such scenarios require expressive extensions of the standard DL languages, capable of representing contexts. Below we present two examples of applying DLCs to divide-and-conquer scenarios.

A contextualized knowledge base with $\mathfrak{F}_1$-operators. Consider a simple representation of knowledge about the legal status of people, contextualized with respect to geographic locations. In Table 5.1, we define a CKB $\mathcal{K} = (\mathcal{C}, \mathcal{O})$ with $\mathfrak{F}_1$-operators, consisting of the context (geographic) ontology $\mathcal{C}$ and the object (people) ontology $\mathcal{O}$. Visibly, ***france*** and ***germany*** play here the role of contexts, described in the context language by axioms (1) and (2). In the context of ***germany***, it is known that $john$ has a parent who is a citizen (3). Since in every ***Country*** context — thus including ***germany*** — the concept $\exists hasParent.Citizen$ is subsumed by $Citizen$ (4), therefore it must be true that $john$ is an instance of $Citizen$ in ***germany***. Finally, since ***germany*** is related to ***france*** via the role ***neighbor***, it follows that $john$ (assuming rigid interpretation of this name across contexts) has to be an instance of $NoVisaRequirement$ in the context of ***france*** (5). A sample $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-model of $\mathcal{K}$ is depicted in Figure 5.7.

$\mathcal{C}$:	$\textbf{\textit{Geometry}} \sqsubseteq \textbf{\textit{Math}}$	(1)
$\mathcal{O}$:	$\textbf{\textit{disambiguation}} : Ring \sqsubseteq \langle \textbf{\textit{Math}} \rangle Ring \sqcup \langle \textbf{\textit{People}} \rangle Ring$	(2)
	$\textbf{\textit{Math}} : Ring \sqsubseteq AlgebStruct \sqcup \langle \textbf{\textit{Geometry}} \rangle Annulus$	(3)
	$\textbf{\textit{People}} : Ring \sqsubseteq \{nickRing\}$	(4)

Table 5.2: A sample knowledge base in $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ with $\mathfrak{F}_2$-operators.

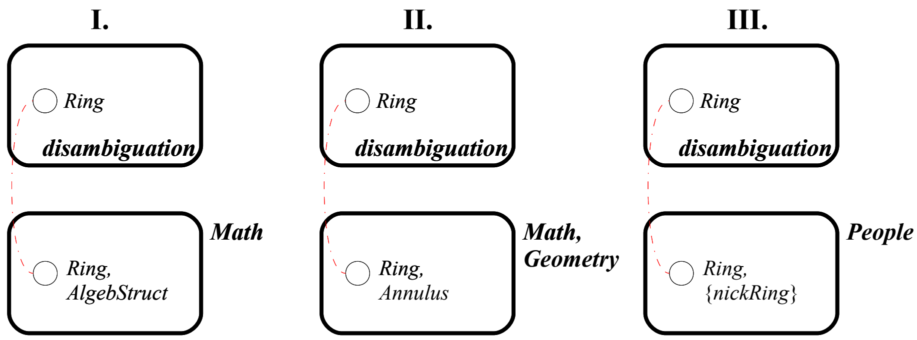

Figure 5.8: Possible models of the CKB in Table 5.2.

A contextualized knowledge base with $\mathfrak{F}_2$-operators. In Table 5.2, we model a piece of information presented on the disambiguation website of Wikipe-dia on querying for the term $Ring$. In particular, $Ring$ is contextualized according to whether it is defined as a mathematical object or as person.[4] Observe, that the named context ***disambiguation*** provides basic distinction on $Ring$ in some ***Math*** context and in some ***People*** context (2). This is further enhanced, by the distinction defined on the level of all ***Math*** contexts. There, $Ring$ denotes either $AlgebStruct$ or $Annulus$ in some further ***Geometry*** context (3), where ***Geometry*** contexts are known to be a subset of ***Math*** contexts. In case of ***People*** context, $Ring$ actually denotes an individual $nickRing$ (4). Some possible $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$-models of this representation are depicted in Figure 5.8.

5.9.2 Compose-and-conquer

Unlike considered in the divide-and-conquer scenario, we might observe that many existing knowledge models very often adopt unique, purpose-driven viewpoints on the domain, determined by the particular applications at hand. In certain situations, one might need to reuse a number of such models in one system. To this end, the models must be composed into a reasonably coordinated, single representation. According to the compose-and-conquer philosophy this can be achieved by acknowledging the presence of the implicit-contexts (assumed during the creation of each individual model) and reflecting then on how these contexts interrelate. The contextualization process is thus considered here as an a posteriori effort of integrating context-specific knowledge models. In the DL paradigm, this problem corresponds to a variety of tasks involving ontology alignment (coordination). Arguably, DLCs can naturally support such sce-

[4] See http://en.wikipedia.org/wiki/Ring.

$\mathcal{O}_{\boldsymbol{c}}$:	$\mathit{Staff} \sqsubseteq \exists \mathit{isEmployed}.\mathit{Company}$	(1)
	$\mathit{Staff}(J.\mathit{Smith})$	(2)
$\mathcal{O}_{\boldsymbol{d}}$:	$\mathit{Employee} \sqsubseteq \exists \mathit{employedIn}.\top$	(3)
	$\mathit{Employee}(\mathit{JohnSmith})$	(4)
	$\top : \langle\{\boldsymbol{c}\}\rangle \mathit{Staff} \equiv \langle\{\boldsymbol{d}\}\rangle \mathit{Employee}$	(5)
	$\top : \langle\{\boldsymbol{c}\}\rangle \{J.\mathit{Smith}\} \equiv \langle\{\boldsymbol{d}\}\rangle \{\mathit{JohnSmith}\}$	(6)

Table 5.3: Integration of ontologies $\mathcal{O}_{\boldsymbol{c}}$, $\mathcal{O}_{\boldsymbol{d}}$ via DLC formulas in $\mathcal{O}_{\boldsymbol{e}}$.

narios. Observe, that a collection of DL ontologies $\mathcal{O}_1, \ldots, \mathcal{O}_n$ in some language $\mathcal{L}_O$ can be seen as a set of formulas $\mathcal{O} = \{\boldsymbol{c}_i : \varphi \mid \varphi \in \mathcal{O}_i,\ i \in (1, n)\}$ in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$, where every ontology is associated with a unique context name. Using DLC formulas one can then impose a number of interesting interoperability constraints over the contents of the ontologies, as presented in the following examples.

Simple vocabulary mappings. Consider two ontologies $\mathcal{O}_{\boldsymbol{c}}$ and $\mathcal{O}_{\boldsymbol{d}}$ describing overlapping domains, as shown in Figure 5.3. Using context operators $\langle\{\boldsymbol{c}\}\rangle, \langle\{\boldsymbol{d}\}\rangle$ we can easily define vocabulary mappings, such as (5)-(6). Given the semantics of DLCs, it follows that Staff must have the same meaning in the context $\boldsymbol{c}$ as $\mathit{Employee}$ in $\boldsymbol{d}$ (5). Similarly, the denotation of individual names $J.\mathit{Smith}$ and $\mathit{JohnSmith}$ is the same across $\boldsymbol{c}$ and $\boldsymbol{d}$ (6). Note, that in this case the context language is restricted to context names only. In this form, the DLCs provide similar functionality to other known logic-based ontology integration formalisms such as DDLs and Package DLs.

Interoperability constraints for ontology alignment and reuse. Consider an architecture such as the NCBO BioPortal project[5], which gathers diverse biohealth ontologies, and categorizes them via thematic tags, e.g.: ***Cell***, ***Health***, ***Anatomy***, etc., organized in a meta-ontology. The intention of the project is to facilitate the reuse of the collected resources in new applications. Note, that the division between the context and the object language is already present in the architecture of the BioPortal, this is naturally reflected in the example of Table 5.4 where (2) maps the concept Heart from any ***HumanAnatomy*** ontology to the concept $\mathit{HumanHeart}$ in every ***Anatomy*** ontology; (3) imposes the axiom $\mathit{Heart} \sqsubseteq \mathit{Organ}$ of an upper anatomy ontology over all ***Anatomy*** ontologies, which due to axiom (1) carries over to all ***HumanAnatomy*** ontologies.

In general, $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ provides logic-based explications of some interesting notions, relevant to the problem of semantic interoperability of ontologies, such as:

concept alignment: $\top : \langle \boldsymbol{A} \rangle C \sqsubseteq [\boldsymbol{B}] D$
every instance of C in any ontology of type $\boldsymbol{A}$ is D in every ontology of type $\boldsymbol{B}$

semantic importing: $\boldsymbol{c} : \langle \boldsymbol{A} \rangle C \sqsubseteq D$
every instance of C in any ontology of type $\boldsymbol{A}$ is D in ontology $\boldsymbol{c}$

[5] See http://bioportal.bioontology.org/.

$\mathcal{C}$:	***HumanAnatomy*** $\sqsubseteq$ ***Anatomy***	(1)
$\mathcal{O}$:	$\top$: $\langle$***HumanAnatomy***$\rangle Heart \sqsubseteq [$***Anatomy***$] HumanHeart$	(2)
	Anatomy : $Heart \sqsubseteq Organ$	(3)

Table 5.4: A set of interoperability constraints expressed as a knowledge base in $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ with $\mathfrak{F}_2$-operators.

upper ontology axiom: ***A*** : $C \sqsubseteq D$
axiom $C \sqsubseteq D$ holds in every ontology of type ***A***

Interoperability constraints for ontology evolving. The context operators can be also interpreted as change operators, in the style of DL of change (*cf.* Chapter 4) for instance, for representing and studying dynamic aspects of ontology versioning, especially when evolutionary constraints apply to a whole collection of semantically interoperable ontologies. Some central issues arising in this setup are integrity (constraining the scope of changes allowed due to versioning), evolvability (ability of coordinating the evolution of ontologies) and formal analysis of differences between the versions [51]. In the examples below, we assume that each ontology version is associated with a unique context, each context concept denotes all versions of a particular ontology and ***updatedBy*** denotes the relation of being an immediate updated version.

version-invariant concepts: $\top$: $\langle$***A***$\rangle C \equiv [$***A***$]C$
C is a version-invariant concept within the scope of versions of type ***A***.

evolvability constraints: ***A*** : $C \sqsubseteq \langle$***updatedBy.B***$\rangle D$
in any version of type ***A***, every instance of C has to evolve into D in some immediate updated version of type ***B***.

5.10 Conclusions

Representing inherently contextualized knowledge as well as reasoning with multiple, heterogeneous, but semantically interoperating knowledge sources are both interesting and practically vital problems within the area of the DL-based knowledge representation. It is our strong belief that these two challenges are in fact two sides of the same coin. Consequently, they should be approached within a unifying formal framework. In this chapter, we have proposed such a framework founded on a novel family of two-dimensional, two-sorted *Description Logics of Context*. The pivotal premise of this theory is that contexts should be interpreted as possible worlds in the second modal dimension added to the standard semantics of DLs. In this way the instrumental, application-agnostic spirit of McCarthy's theory of contexts can be successfully combined with the formal machinery of modal logics.

The work presented in this chapter establishes the generic foundations for the DLC framework and opens up a number of theoretical and practical problems which should be addressed in future research. One important direction is to investigate how different notions common to traditional context-based systems (e.g., managing local inconsistencies or modeling generality hierarchy of contexts, etc.) can be effectively restated within DLCs. Another course of research should be dedicated to identification and formal analysis of specific fragments of the framework that could be especially useful in practice, particularly considering semantic web applications. For instance, a scenario of integrating a finite number of ontologies does not in principle require the full expressiveness of DLCs. Similarly, an efficient support for reasoning with contextually annotated semantic web data could be likely provided via a more lightweight fragment. Finally, on a more abstract level, it could be interesting to investigate whether a similar methodology of constructing two-dimensional, two-sorted formalisms could be applicable to combinations of DLs with other modal logics, e.g. spatial or temporal, in order to support fine-grained descriptions of the second semantic dimension by means of a dedicated vocabulary.

6
Conclusions

In this thesis, we investigated several two-dimensional extensions of classical description logics allowing to represent and reason about temporal and contextual knowledge. We particularly focused on the development of algorithms for satisfiability and subsumption, and on the establishment of tight complexity bounds. For the former, we explored various kinds of techniques based on automata over infinite trees, type-elimination, completion-algorithms and quasistate elimination.

The main objective of this thesis was to make further steps towards the design of more useful two-dimensional DLs. With this in mind, we focused on identifying logics providing the right expressive power to model more accurately temporal or contextual aspects of knowledge required by certain ontology applications, or offering better computational properties than other possible alternatives. More precisely, we pursued the following research lines:

Branching-time TDLs. We investigated TDLs providing capabilities to differentiate between possible and necessary future developments of knowledge. We particularly looked at TDLs that are obtained from the combination of the standard TL CTL* and its fragment CTL with classical DLs.

We presented, in Chapter 3, algorithms to reason about the temporal evolution of concepts. We obtained a tight ExpTime upper bound for $\mathrm{CTL}_{\mathcal{ALC}}$ and a tight 2ExpTime upper bound for $\mathrm{CTL}^*_{\mathcal{ALC}}$. These results show that they are no more complex than their components. Intuitively, these results are explained by the fact that the interaction of the component logics is rather weak, similar to the fusion of modal logics. Later on, we studied fragments of $\mathrm{CTL}_{\mathcal{EL}}$ with the objective of identifying a computationally efficient branching-time TDL. We successfully identified the polytime fragment $\mathrm{CTL}^{\mathbf{E}\Diamond}_{\mathcal{EL}}$. Notably, this is the first TDL based on standard TLs and $\mathcal{EL}$ for which reasoning is easier than in the $\mathcal{ALC}$ variant. As discussed above, this upper bound is rather fragile in the sense that if we allow further temporal operators then tractability is destroyed.

We further presented algorithms to reason about the temporal evolution of TBoxes, that is, we also apply temporal operators to CIs. In this case, we obtained a 2EXPTIME upper bound for temporal $\mathrm{CTL}_{\mathcal{ALC}}$-TBoxes and a 3EXPTIME upper bound for temporal $\mathrm{CTL}^*_{\mathcal{ALC}}$-TBoxes, respectively. We also showed that for temporal $\mathrm{CTL}_{\mathcal{ALC}}$-TBoxes the 2EXPTIME upper bound is indeed tight. The latter shows that the increase in the expressivity is reflected in an increase in the computational complexity.

Description Logics of Change. We investigated TDLs based on a weaker temporal component given by the modal logic $\mathbf{S5}$. These TDLs allow to reason about the changes of knowledge over time without differentiating between changes in the past and in the future. A key characteristic of DLs of change based on expressive DLs is that they allow to effectively reason about the temporal evolution of roles and concepts. We particularly took a look at DLs of change based on $\mathcal{EL}$ and its extension $\mathcal{ELI}$ with the objective of designing lightweight temporal logics allowing to reason about the temporal evolution of concepts and roles.

We presented, in Chapter 4, an algorithm to reason about the temporal evolution of concepts and roles in $\mathbf{S5}_{\mathcal{EL}}$, yielding a PSPACE tight upper bound. We have thus identified the first two-dimensional DL based on $\mathcal{EL}$ allowing for modalities to be applied to roles and concepts for which reasoning is easier than in the $\mathcal{ALC}$ variant. Alas, we showed that this result does not hold anymore if $\mathcal{ELI}$ is considered instead of $\mathcal{EL}$. In particular, we showed that reasoning in $\mathbf{S5}_{\mathcal{ELI}}$ is 2EXPTIME-complete and then as complex as the $\mathcal{ALC}$ variant.

Furthermore, we showed that reasoning about the temporal evolution of concepts in $\mathbf{S5}_{\mathcal{ALCO}}$ is NEXPTIME-complete. Since reasoning about the temporal evolution of concepts in $\mathbf{S5}_{\mathcal{ALC}}$ is EXPTIME-complete, and moreover reasoning in $\mathcal{ALCO}$ is EXPTIME-complete, the former result shows that interaction of nominals with $\mathbf{S5}$-modalities makes the reasoning harder in $\mathbf{S5}_{\mathcal{ALCO}}$.

Description Logics of Context. We investigated the adequacy of two-dimensional DLs to represent and reason about contextualized knowledge. Notably, by interpreting contexts as possible worlds, we have successfully imported McCarthy's theory of contexts into the DL paradigm. We did a stepwise integration of McCarthy's theory into the DL paradigm through two-dimensional DLs.

In Chapter 5, we considered classical two-dimensional DLs. In particular, we took a look at the prominent $(\mathbf{K}_n)_{\mathcal{ALC}}$. As discussed above, by importing two of the three postulates of McCarthy's theory, $(\mathbf{K}_n)_{\mathcal{ALC}}$ is capable to capture contextual aspects of knowledge. Surprisingly, we showed that reasoning in $(\mathbf{K}_n)_{\mathcal{ALC}}$ in the case where modalities are applied only to concepts and a global TBox is present is 2EXPTIME-complete. This indeed comes as surprise since normally reasoning in combinations allowing for such limited interaction of the component logics is no harder than in the components.

Furthermore, we extended $(\mathbf{K}_n)_{\mathcal{L}}$ with two-interacting DL languages: the object-level language and the meta-level language. In particular, the latter allows descriptions of contexts as first-class citizens. These extensions, as discussed above, fully implement McCarthy's theory into the DL paradigm. We moreover showed that reasoning in these extensions based on $\mathcal{ALCO}$ is also 2EXPTIME-complete and thus no more difficult than in $(\mathbf{K}_n)_{\mathcal{ALCO}}$.

This investigation leaves open some important problems as future work. Many of the directions to follow have already been discussed in the conclusions of each chapter. Besides these, applications to medical ontologies suggest that a more light approach to temporal DLs or context DLs may be more appropriate than combinations of classical DLs with standard TLs, or modal logics. This naturally is a big challenge since currently is rather unclear how such an approach could look like. Another important challenge is the development of mechanisms to extend the *ontology-based access* approach towards accessing temporal or spatio-temporal data.

Bibliography

[1] Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature genetics*, 25(1):25–29, 2000.

[2] V. Akman and M. Surav. Steps toward formalizing context. *AI Magazine*, 17:55–72, 1996.

[3] A. Artale. Reasoning on temporal conceptual schemas with dynamic constraints. In *Proceedings of TIME*, pages 79–86. IEEE Computer Society, 2004.

[4] A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyaschev. The DL-Lite family and relations. *J. Artif. Intell. Res. (JAIR)*, 36:1–69, 2009.

[5] A. Artale and E. Franconi. Temporal description logics. In *Handbook of Time and Temporal Reasoning in Artificial Intelligence*, pages 375–388, 2005.

[6] A. Artale, E. Franconi, and F. Mandreoli. Description logics for modeling dynamic information. In *Logics for Emerging Applications of Databases*, pages 239–275, 2003.

[7] A. Artale, R. Kontchakov, C. Lutz, F. Wolter, and M. Zakharyaschev. Temporalising tractable description logics. In *TIME*, pages 11–22, 2007.

[8] A. Artale, R. Kontchakov, V. Ryzhikov, and M. Zakharyaschev. Past and future of DL-Lite. In *AAAI*, 2010.

[9] A. Artale, R. Kontchakov, V. Ryzhikov, and M. Zakharyaschev. Tailoring temporal description logics for reasoning over temporal conceptual models. In *FroCos*, volume 6989 of *Lecture Notes in Computer Science*, pages 1–11. Springer, 2011.

[10] A. Artale, C. Lutz, and D. Toman. A description logic of change. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI-07)*, 2007.

[11] A. Artale and D. Toman. Decidable reasoning over timestamped conceptual models. In *Description Logics*, 2008.

[12] F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In *IJCAI*, pages 364–369, 2005.

[13] F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope further. In K. Clark and P.F. Patel-Schneider, editors, *Proc. of the Workshop on OWL: Experiences and Directions (OWLED-08 DC)*, 2008.

[14] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2003.

[15] F. Baader, S. Ghilardi, and C. Lutz. LTL over description logic axioms. In *KR*, pages 684–694, 2008.

[16] F. Baader, I. Horrocks, and U. Sattler. Description logics as ontology languages for the semantic web. In *Mechanizing Mathematical Reasoning: Essays in Honor of Jörg H. Siekmann on the Occasion of His 60th Birthday*, volume 2605 of *LNAI*, pages 228–248. Springer-Verlag, 2005.

[17] J. Bao, J. Tao, D. L. McGuinness, and P. Smart. Context representation for the semantic web. *Proc. of Web Science Conference*, 2010.

[18] J. Bao, G. Voutsadakis, G. Slutzki, and V. Honavar. Package-based description logics. In Heiner Stuckenschmidt, Christine Parent, and Stefano Spaccapietra, editors, *Modular Ontologies*, pages 349–371. 2009.

[19] S. Bauer, I. M. Hodkinson, F. Wolter, and M Zakharyaschev. On non-local propositional and weak monodic quantified ctl. *J. Log. Comput.*, 14(1):3–22, 2004.

[20] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, May 2001.

[21] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2001.

[22] O. Bodenreider and S. Zhang. Comparing the representation of anatomy in the FMA and SNOMED CT. In *Proceedings of the AMIA Annual Symposium*, pages 46–50, 2006.

[23] E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem*. Perspectives in Mathematical Logic. Springer, 1997.

[24] A. Borgida and L. Serafini. Distributed description logics: Assimilating information from peer sources. *Journal of Data Semantics*, 1, 2003.

[25] P. Bouquet, C. Ghidini, F. Giunchiglia, and E. Blanzieri. Theories and uses of context in knowledge representation and reasoning. *Journal of Pragmatics*, 2003.

[26] L. Bozzato, M. Homola, , and L. Serafini. Context on the semantic web: Why and how. In *Proc. of the 4th International Workshop on Acquisition, Representation and Reasoning with Contextualized Knowledge (ARCOE-12)*, 2012.

[27] T. Brázdil, V. Forejt, J. Kretínský, and A. Kucera. The satisfiability problem for probabilistic CTL. In *LICS*, pages 391–402. IEEE Computer Society, 2008.

[28] S. Buvač. Quantificational logic of context. In *Proc. of the Conference on Artificial Intelligence (AAAI-96)*, 1996.

[29] S. Buvač, V Buvac, and I. A. Mason. Metamathematics of contexts. *Fundamenta Informaticae*, 23:412–419, 1995.

[30] S. Buvač and I. A. Mason. Propositional logic of context. In *Proc. of the Conference on Artificial Intelligence (AAAI-93)*, 1993.

[31] A. K. Chandra, D. Kozen, and L. J. Stockmeyer. Alternation. *J. ACM*, 28(1):114–133, 1981.

[32] E. M. Clarke and E. A. Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In *Logic of Programs*, pages 52–71, 1981.

[33] E. A. Emerson. Temporal and modal logic. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 16, pages 995–1072. Elsevier Science, 1990.

[34] E. A. Emerson and J. Y. Halpern. "sometimes" and "not never" revisited: on branching versus linear time temporal logic. *J. ACM*, 33(1), 1986.

[35] E. A. Emerson and C. S. Jutla. The complexity of tree automata and logics of programs. *SIAM Journal on Computing*, 29(1):132–158, September 1999.

[36] M. J. Fischer and R. E. Ladner. Propositional dynamic logic of regular programs. *J. Comput. Syst. Sci.*, 18(2):194–211, 1979.

[37] E. Franconi and D. Toman. Fixpoints in temporal description logics. In *Proceedings of IJCAI*, pages 875–880. AAAI, 2011.

[38] D. Gabbay, A. Kurucz, F. Wolter, and M. Zakharyaschev. *Many-Dimensional Modal Logics: Theory and Applications*. Studies in Logic, 148. Elsevier Science, 2003.

[39] C. Ghidini and F. Giunchiglia. Local models semantics, or contextual reasoning=locality+compatibility. *Artificial Intelligence*, 127(2):221 – 259, 2001.

[40] F. Giunchiglia. Contextual reasoning. *Epistemologia*, XVI:345–364, 1993.

[41] E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002.

[42] R. Guha. *Contexts: a formalization and some applications*. PhD thesis, Stanford University, 1991.

[43] R. Guha, R. McCool, and R. Fikes. Contexts for the semantic web. In *Proc. of the International Semantic Web Conference (ISWC-04)*, 2004.

[44] V. Gutiérrez-Basulto, C. Jung, J, C. Lutz, and L. Schröder. A closer look at the probabilistic description logic prob-$\mathcal{EL}$. In *Proceedings of Twenty-Fifth Conference on Artificial Intelligence (AAAI-11)*, 2011.

[45] J. Hendler and T. Berners-Lee. From the semantic web to social machines: A research challenge for ai on the world wide web. *Artif. Intell.*, 174(2):156–161, 2010.

[46] I. M. Hodkinson, R. Kontchakov, A. Kurucz, F. Wolter, and M. Zakharyaschev. On the computational complexity of decidable fragments of first-order linear temporal logics. In *TIME*, pages 91–98, 2003.

[47] I. M. Hodkinson, F. Wolter, and M. Zakharyaschev. Decidable fragment of first-order temporal logics. *Ann. Pure Appl. Logic*, 106(1-3):85–134, 2000.

[48] I. M. Hodkinson, F Wolter, and M Zakharyaschev. Decidable and undecidable fragments of first-order branching temporal logics. In *LICS*, pages 393–402, 2002.

[49] M. Hofmann. Proof-theoretic approach to description-logic. In *LICS*, pages 229–237, 2005.

[50] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From SHIQ and RDF to OWL: the making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, 2003.

[51] Z. Huang and H. Stuckenschmidt. Reasoning with multi-version ontologies: A temporal logic approach. In *Proc. of the Intenational Semantic Web Conference (ISWC-05)*. 2005.

[52] Y. Kazakov. Consequence-driven reasoning for Horn SHIQ ontologies. In *IJCAI*, pages 2040–2045, 2009.

[53] O. Kupferman and M. Y. Vardi. Safraless decision procedures. In *FOCS*, pages 531–542, 2005.

[54] O. Kupferman, M. Y. Vardi, and P Wolper. An automata-theoretic approach to branching-time model checking. *J. ACM*, 47(2):312–360, 2000.

[55] O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev. E-connections of abstract description systems. *Artificial Intelligence*, 156:1–73, June 2004.

[56] D. Lenat. The dimensions of context space. Technical report, CYCORP, 1998.

[57] C. Lutz and L. Schröder. Probabilistic description logics for subjective uncertainty. In *KR*, 2010.

[58] C. Lutz, F. Wolter, and M. Zakharyaschev. Temporal description logics: A survey. In *TIME*, pages 3–14, 2008.

[59] J. McCarthy. Generality in artificial intelligence. *Communications of the ACM*, 30:1030–1035, 1987.

[60] J. McCarthy. Notes on formalizing context. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI-93)*, 1993.

[61] M. Minsky. A framework for representing knowledge. In P. Winston, editor, *The psychology of computer vision*, pages 211–277. McGraw-Hill, 1975.

[62] B. Motik, B. Cuenca Grau, I. Horrocks, Z. Wu, A. Fokoue, and C. Lutz. OWL 2 Web Ontology Language: Profiles. W3C Recommendation, W3C, `http://www.w3.org/TR/owl2-profiles/`, October 2009.

[63] D. E. Muller and P. E. Schupp. Alternating automata on infinite trees. *Theor. Comput. Sci.*, 54:267–276, 1987.

[64] R. Nossum. A decidable multi-modal logic of context. *Journal of Applied Logic*, 1(1-2):119 – 133, 2003.

[65] J. Ouaknine and J. Worrell. Some recent results in metric temporal logic. In F. Cassez and C. Jard, editors, *FORMATS*, volume 5215 of *LNCS*, pages 1–13, 2008.

[66] W3C OWL Working Group. *OWL 2 Web Ontology Language: Document Overview*. W3C Recommendation, 2009. Available at `http://www.w3.org/TR/owl2-overview/`.

[67] J. Z. Pan, L. Serafini, and Y. Zhao. Semantic import: an approach for partial ontology reuse. In *Proc. of the Workshop on Modular Ontologies (WoMO-06)*, 2006.

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

[69] M. R. Quillian. Word Concepts: A Theory and Simulation of Some Basic Semantic Capabilities. *Behavioral Science*, 12(5), 1967.

[70] K. Schild. Combining terminological logics with tense logic. In *Proceedings of EPIA*, volume 727 of *Lecture Notes in Computer Science*, pages 105–120. Springer, 1993.

[71] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.

[72] S. Schulz, K. Markó, and B. Suntisrivaraporn. Complex occurrents in clinical terminologies and their representation in a formal language. In *Proc. of the First European Conference on SNOMED CT (SMCS 06)*, 2006.

[73] L. Serafini and M. Homola. Contextualized knowledge repositories for the semantic web. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 12, 2012.

[74] K. Spackman. SNOMED CT style guide: Situations with explicit context. Technical report, SNOMED CT, 2008.

[75] K. A. Spackman. Managing clinical terminology hierarchies using algorithmic calculation of subsumption: Experience with SNOMED-RT. *Journal of the American Medical Association*, 2000.

[76] P. van Emde Boas. The convenience of tiling. In A. Sorbi, editor, *Complexity, Logic and Recursion Theory*, volume 187 of *Lecture Notes in Pure and Applied Mathematics*, pages 331–363. Marcel Dekker Inc., February 1997.

[77] M. Y. Vardi. Why is modal logic so robustly decidable? In *Descriptive Complexity and Finite Models*, pages 149–184, 1996.

[78] M. Y. Vardi. Reasoning about the past with two-way automata. In *ICALP*, pages 628–641, 1998.

[79] M. Y. Vardi. Automata-theoretic techniques for temporal reasoning. In *In Handbook of Modal Logic*, pages 971–989. Elsevier, 2006.

[80] M. Y. Vardi and L. J. Stockmeyer. Improved upper and lower bounds for modal logics of programs: Preliminary report. In *STOC*, pages 240–251, 1985.

[81] M. Y. Vardi and P. Wolper. Automata-theoretic techniques for modal logics of programs. *J. Comput. Syst. Sci.*, 32(2):183–221, 1986.

[82] F. Wolter and M. Zakharyaschev. Modal description logics: modalizing roles. *Fundamenta Informaticae*, 39(4):411–438, 1999.

[83] F. Wolter and M. Zakharyaschev. Multi-dimensional description logics. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI-99)*, 1999.