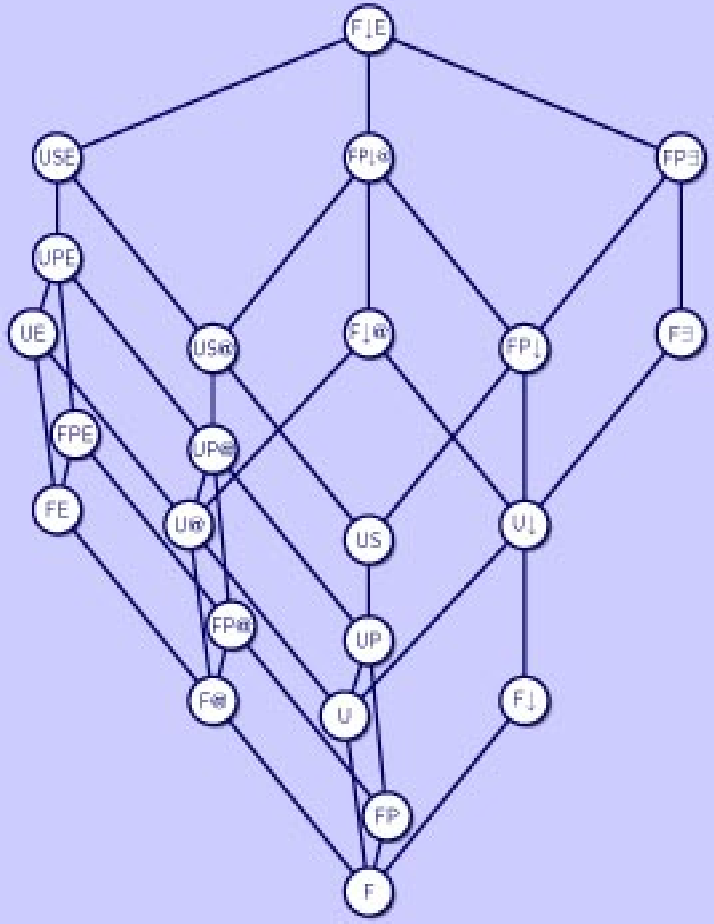
Thomas Schneider

The Complexity of Hybrid Logics over Restricted Frame Classes



Cuvillier Verlag Göttingen

The Complexity of Hybrid Logics over Restricted Frame Classes

Dissertation
zur Erlangung des akademischen Grades
doctor rerum naturalium (Dr. rer. nat.)

vorgelegt dem Rat der
Fakultät für Mathematik und Informatik
der Friedrich-Schiller-Universität Jena

von Diplom-Mathematiker Thomas Schneider
geboren am 15. Juni 1976 in Leipzig

Bibliografische Information Der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.ddb.de abrufbar.
1. Aufl. - Göttingen : Cuvillier, 2007
Zugl.: Jena, Univ., Diss., 2007

ISBN 978-3-86727-298-8

Gutachter

1. Prof. Dr. Martin Mundhenk, Friedrich-Schiller-Universität Jena
2. Prof. Dr. Heribert Vollmer, Leibniz-Universität Hannover

Tag der letzten Prüfung des Rigorosums: 12. 07. 2007
Tag der öffentlichen Verteidigung: 16. 07. 2007

Nonnenstieg 8, 37075 Göttingen
Telefon: 0551-54724-0
Telefax: 0551-54724-21
www.cuvillier.de

1. Auflage, 2007
Gedruckt auf säurefreiem Papier

ISBN 978-3-86727-298-8

There are times when all the world's asleep,
the questions run too deep
for such a simple man.
Won't you please, please tell me what we've learned,
I know it sounds absurd,
but please tell me who I am.

Roger Hodgson, Rick Davies: The Logical Song

Zusammenfassung

Diese Dissertation untersucht die Komplexität von Entscheidungsproblemen für eine Familie hybrider Logiken über verschiedenen Klassen von Strukturen.

Hybride Logiken sind Erweiterungen modaler Logik. Diese ist ein mächtiger, gut handhabbarer und wohluntersuchter Formalismus, mit dem man Eigenschaften von und Anforderungen an Anwendungen beschreiben kann, die durch relationale Strukturen modellierbar sind – zum Beispiel das Verhalten von Dingen mit der Zeit, das Wissen von Agenten oder die Verifikation von Programmen. In all diesen Anwendungen bietet modale Logik eine lokale Perspektive, das heißt, man kann damit Dinge beschreiben, die in einzelnen Zuständen relationaler Strukturen und in deren Nachfolgerzuständen stattfinden. In Abhängigkeit von der jeweiligen Anwendung werden bestimmte Varianten (oder Erweiterungen) modaler Logik verwendet – beispielsweise temporale, epistemische oder dynamische Logik.

Hybride Logiken bieten mehr Ausdruckskraft durch Namen für Zustände von Strukturen und direkten Zugriff auf diese Zustände. Das ist in vielen Anwendungen begehrt. So ist es im temporalen Fall nur natürlich, Namen für Zeitpunkte zu haben und, unabhängig von der „später-als"-Relation, darauf zuzugreifen. Außerdem kann man mittels hybrider Logik viele temporal relevanten Eigenschaften von Strukturen wie Irreflexivität, Antisymmetrie oder Trichotomie axiomatisieren. Wegen dieser Eigenschaften sind hybride Sprachen sehr begehrt, wann immer die modale Grundsprache an ihre Grenzen stößt.

Die Hauptbestandteile hybrider Logik, von einem temporalen Standpunkt aus betrachtet, sind die folgenden Ausdrucksmittel.

- *Zukunfts- und Vergangenheitsoperatoren* – drücken aus:
  "irgendwann in der Zukunft φ" oder "immer in der Vergangenheit φ"
- *Until- und Since-Operatoren* – drücken zum Beispiel aus:
  "irgendwann in der Zukunft ψ, und von jetzt an bis dahin φ"
- *Nominale* – geben Punkten in einer Struktur feste Namen
- *Sprungoperatoren* – gestatten Sprünge zu benannten Punkten
- *Binder* – binden Namen dynamisch an Punkte
- *die globale Modalität* – drückt aus: "irgendwann φ"

In Abhängigkeit von der jeweiligen Anwendung ist es angebracht, die Klasse der Strukturen auf diejenigen einzuschränken, die die Anforderungen dieser Anwendung modellieren. Relevante Strukturenklassen für temporale oder epistemische Anwendungen sind beispielsweise transitive Strukturen, transitive Bäume, lineare Ordnungen, die natürlichen Zahlen, Strukturen mit Äquivalenzrelationen (ÄR-Strukturen) oder vollständige Strukturen.

Die Entscheidbarkeit und die Komplexität von Entscheidungsproblemen für Logiken sind relevant für das automatisierte Lösen dieser Probleme. Wir untersuchen systematisch das Erfüllbarkeitsproblem und das Model-Checking-Problem für alle relevanten hybriden Sprachen, die beliebige Kombinationen der oben aufgeführten Operatoren enthalten, bezüglich der genannten Strukturenklassen. Das schließt ein, eine Hierarchie aller dieser Sprachen aufzustellen, Ergebnisse aus der Literatur dort einzuordnen und eigene Resultate beizusteuern. Im Einzelnen beweisen wir die folgenden Hauptergebnisse unter Zuhilfenahme einer breiten Palette von Techniken.

- Das Model-Checking-Problem für Sprachen mit Bindern bleibt PSPACE-vollständig über allen genannten Strukturenklassen. (Kapitel 4)
- Das Erfüllbarkeitsproblem für hybride Until-/Since-Sprachen über transitiven Strukturen ist EXPTIME-hart und in 2EXPTIME. (Abschnitt 5.4)
- Das Erfüllbarkeitsproblem für fast alle Erweiterungen der kleinsten Bindersprache über transitiven Strukturen ist unentscheidbar. (Abschn. 5.4)
- Das Erfüllbarkeitsproblem für hybride Until-/Since-Sprachen über transitiven Bäumen ist EXPTIME-vollständig. (Abschnitt 5.5)
- Das Erfüllbarkeitsproblem für alle Sprachen mit Bindern über transitiven Bäumen ist nichtelementar entscheidbar. (Abschnitt 5.5)
- Das Erfüllbarkeitsproblem für hybride Sprachen ohne Binder über ÄR-Strukturen ist NP-vollständig. (Abschnitt 5.8)
- Das Erfüllbarkeitsproblem für die Sprache mit allen Operatoren über ÄR-Strukturen ist N2EXPTIME-vollständig. (Abschnitt 5.8)
- Das Erfüllbarkeitsproblem für alle übrigen hybriden Sprachen über ÄR-Strukturen ist NEXPTIME-vollständig. (Abschnitt 5.8)
- Das Erfüllbarkeitsproblem für die bimodale Version der kleinsten Bindersprache über vielen Strukturenklassen, darunter fast alle der oben genannten, ist unentscheidbar. (Kapitel 6)

Diese Dissertation enthält Material, das auf den Workshops „Methods for Modalities" (2005, für eine Sonderausgabe des „Journal of Logic, Language and Information" angenommen) und „Hybrid Logic" (2006 und 2007) präsentiert und in den zugehörigen Tagungsbänden veröffentlicht wurde.

Abstract

This dissertation examines the computational complexity of decision problems for a collection of hybrid logics over different classes of frames.

Hybrid logics are extensions of modal logic that allow, in addition to the usual perspective on states and their successors in relational structures, for naming and accessing states of a structure explicitly. These features are very desirable in many applications, for example, in the temporal or epistemic context.

We will systematically examine decidability and the computational complexity of satisfiability and the model-checking problem for a systematic collection of hybrid languages with respect to temporally and epistemically relevant classes of structures. This includes establishing a hierarchy of all relevant languages over these classes, arranging results from the literature into this hierarchy, and contributing our own results. In particular, we prove the following main results, involving a wide range of techniques for establishing complexity bounds of logics.

- The model-checking problem for all binder languages remains PSPACE-complete over restricted classes of structures. (Chapter 4)
- The satisfiability problem for hybrid until/since languages over transitive structures is EXPTIME-hard and in 2EXPTIME. (Section 5.4)
- The satisfiability problem for almost all extensions of the smallest binder language over transitive structures is undecidable. (Section 5.4)
- The satisfiability problem for hybrid until/since languages over transitive trees is EXPTIME-complete. (Section 5.5)
- The satisfiability problem for all binder languages over transitive trees is nonelementarily decidable. (Section 5.5)
- The satisfiability problem for binder-free hybrid languages over ER structures is NP-complete. (Section 5.8)
- The satisfiability problem for the language with all operators over ER structures is N2EXPTIME-complete. (Section 5.8)
- The satisfiability problem for all remaining hybrid languages over ER structures is NEXPTIME-complete. (Section 5.8)

- The satisfiability problem for the bi-modal version of the smallest binder language over many classes of structures, including many restricted ones, is undecidable. (Chapter 6)

This dissertation contains material presented at, and published in the proceedings of, the workshops “Methods for Modalities” (2005, accepted for a special issue of the Journal of Logic, Language and Information) and “Hybrid Logic” (2006 and 2007).

Acknowledgements

I am very grateful to Martin Mundhenk for his constant supervision, cooperation and generous support during my time as a PhD student in his group. It was he who suggested examining the complexity of modal (and related) logics as a Diplom (MSc) thesis topic, and who offered to supervise my research in this field afterwards. His suggestions and motivation helped me to tackle problems and get in touch with the modal and hybrid logic community.

Motivation during the writing of this dissertation also emanated from the theoretical computer science group at the University of Hannover (consisting of Heribert Vollmer, Michael Bauland, Ilka Schnoor, and Henning Schnoor). Our collaboration on LTL decision problems (whose outcome is not part of this thesis) as well as their feedback in talks and conversations was, and still is, inspiring and instructive for me and makes me very much obliged.

Parts of this dissertation have profited from suggestions, discussions, and comments from several people such as Balder ten Cate, Massimo Franceschet, and Maarten Marx (Sections 5.4 and 5.5), as well as Ulrike Sattler and Volker Weber (Chapter 6). Volker Weber and Thomas Schwentick were pleasant coauthors who made a major contribution to the paper where parts of Sections 5.4 and 5.5 appeared.

Furthermore, I appreciate the feedback of the anonymous referees of the workshops where parts of this thesis appeared, namely "Methods for Modalities" (M4M 2005) and "Hybrid Logic" (HyLo 2006 and 2007).

Last, but not least, supporting pillars for dissertations are not only one's colleagues. I am very grateful to my wife, Renate Stein, who has been understanding and encouraging me over the past few years, who experienced the variably difficult phases of preparing (and finishing!) a dissertation long before I did, and who patiently explained the difference in the use of adjectives and adverbs in English to me for the $\mathcal{O}(n)$-th time, although I was (not quite rightfully) sure to have understood it at the $\mathcal{O}(1)$-st time.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Modal, temporal, and hybrid languages

Modal logic is a powerful, easily utilisable, well-understood, and well-behaved formalism for describing and specifying properties of any application that can be modelled by relational structures. Such applications (and their corresponding relational structures) are, for example

(1) the behaviour of things over time
(points in time and the "later-than" relation);

(2) the knowledge of agents
(states describing the knowledge of the agent together with the relation linking actual states with possible states);

(3) verification of programmes
(states of a machine and transitions between them given by executions of programmes).

In all these applications, modal logic offers a local perspective, that is, it allows for describing things that happen in individual states of relational structures and their successor states. Depending on the particular application, certain variants (or extensions) of modal logic are used — for example, temporal (1), epistemic (2), or dynamic (3) logic.

This thesis will not consider logics for one certain application. Therefore we will prefer an abstract view (in terms of relational structures) to a concrete one (as given in the above examples). However, since many of the languages we examine are useful for temporal applications, we will often speak about them in terms of Example (1) from above. Until we provide a formal definition of relational structures in Chapter 2, we will call them *structures* or *frames*, and refer to their elements as *states* or *points*.

Hybrid languages are extensions of modal logic that allow for naming and accessing states of a structure explicitly. These features are very desirable in many applications. Particularly in the temporal case, it is very natural to give names to points in time and refer to them independently of the "later-than" relation. Besides, by means of hybrid logic, it is possible to capture many temporally relevant properties of structures, such as irreflexivity, antisymmetry, trichotomy, directedness, etc. For these reasons, hybrid languages and hybrid *temporal* languages are of great interest where basic modal and temporal logic reach their limits [BT99, Bla00b, ABM01, FdRS03].

Another reason for the interest in hybrid logic is discussed in [ABM99] and [ABM00]. Hybrid languages are proof-theoretically well behaved and "internalise" labelled deduction [Bla00a], an apparatus that guides proof search in modal logic [Gab96].

Hybrid Logic, as well as the foundations of temporal logic, goes back to Arthur Prior [Pri67]. Since then, many — more or less powerful — languages have been studied [Bul70, PT91, Bla93, GG93, BS95, Gor96, BS98, ABM99, ABM00, ABM01, FdRS03]. The main features of hybrid logic that are of special interest for this thesis are the following.

Nominals. They are special atomic propositions that give names to states — a very natural thing for applications, particularly temporal ones. With the help of nominals, it is possible to express properties of structures that are not expressible in modal logic, such as irreflexivity, asymmetry, etc.

The satisfaction operators. They allow for jumping to a point named by a nominal, regardless of the accessibilities in the structure.

Hybrid binders. They allow for binding names to states dynamically and for referring to these states later on. This makes them a very powerful and desirable means of expression, especially if they are combined with satisfaction operators. Unfortunately, due to this high expressive power, binders are dangerous in terms of computational costs.

Furthermore, we will consider operators that occur in the context of modal logic, too.

The "until" and "since" operators. They permit temporal statements such as: "Until some point with property ψ, it is always the case that φ." This notion of "betweenness" cannot be expressed by the usual temporal operators, which only allow for accessing some successor or predecessor state while immediately forgetting about the original one. Again, the increased expressivity makes the until/since operators worthwhile, and fortunately, the computational costs paid are not as dramatically high as in the case of hybrid binders.

The global modality. It simply grants access to any point in the structure and can thus be seen as a generalisation of satisfaction operators. Similarly to the until/since operators, it adds expressive power to the language, which sometimes makes reasoning harder.

1.2 Towards a systematic study of the complexity of hybrid logics

This thesis systematically examines decidability and the computational complexity of decision problems for a collection of hybrid languages with respect to several classes of structures. More precisely speaking, we will establish, in the usual terms of computational complexity theory, what amount of resources (space, time) are necessary for an algorithm to decide each of these problems, and whether such an algorithm exists at all. We will focus on the satisfiability problem and provide results on the model-checking problem. These problems ask whether a given formula from a certain hybrid logic is satisfiable in some structure or a given structure, respectively. Decidability and the computational complexity of decision problems are of great interest whenever those shall be solved automatically, see [Wos85] for an introduction into automated reasoning.

Satisfiability for hybrid logic tends to have a high computational complexity in general, which is due to the increased expressive power of hybrid languages. For instance, satisfiability for hybrid logic is known to require exponential time [ABM00] in the presence of past or until operators, and to be even undecidable if a fairly restricted form of a binder is admitted [ABM99]. This is in contrast to modal logic, whose satisfiability problem is solvable in polynomial space [Lad77]. Furthermore, model checking for modal languages is solvable in polynomial time, but in the presence of binders, polynomial time most probably does not suffice because the model-checking problem is complete for polynomial space here [FdR06].

It is well-known that many applications for modal or hybrid logic do not require the full language or do not permit all possible frames. Hence, restricting the language and/or the class of relevant frames could be a way to "tame" a very expressive logic. And indeed, there is much literature where very different complexities for more or less expressive hybrid languages over different classes of frames have been established [ABM99, ABM00, FdRS03, tCF05b, FdR06, MSSW05, MS07b, MS07a]. There are combinations of hybrid languages and frame classes, for which the satisfiability problem, for instance, is known to be complete for the complexity classes NP, PSPACE, EXPTIME, NEXPTIME, N2EXPTIME; nonelementarily decidable; or even undecidable. However, we are not aware of any systematic study that involves several frame classes and, independently from those, a self-contained collection of hybrid languages.

Such a systematic study is pursued by this thesis and will show problems that have not been solved in the literature yet. We will fix a set of modal, temporal, and hybrid operators and consider a hierarchy of all hybrid languages defined by subsets of this set of operators. We will then arrange known results from the literature into this hierarchy, separately for several classes of frames. This will show that there are many combinations of languages and frame classes whose complexity is not known. We will provide results for most of them, applying a wide range of well-known techniques for establishing lower and upper complexity bounds in modal and hybrid logic.

We do not claim that either collection (of frame classes or languages) is complete, but, at least, our study covers all hybrid languages with the most commonly used operators and many temporally and epistemically relevant frame classes. Here, the notion of the "relevance of frame classes for applications" deserves a more precise explanation.

In view of temporal applications, it is apparent that only frames with "later-than" relations satisfying certain properties need be considered. Such properties include — but are not restricted to — transitivity, irreflexivity, or trichotomy. (The latter refers to the condition that given two distinct points, at least one is related to the other.) One of the most special frame class in this context is the class that consists of only one frame, namely the natural numbers with the greater-than relation. This class underlies the widely used and well-understood Linear Temporal Logic (see, e.g., [CGP01]) and represents a discrete view on time. It is possible to consider the integers or the reals instead of natural numbers [Rey92]. Furthermore, there are two generalisations of these singleton frame classes. One is the class of linear frames that merely requires the above three properties and contains frames with discrete as well as dense flows of time (among them, the natural numbers, integers, and reals). Another generalisation is the class of transitive trees that adds branching to the natural numbers and underlies the expressive Computation Tree Logic (which is described in [CGP01], too).

For epistemic applications, equivalence relations and weaker notions are necessary to model knowledge and belief of agents [FHMV95, Section 3.1]. If the states and accessibility relations in a frame are to represent possible worlds of agents and if the agents' knowledge or beliefs are assumed to satisfy certain soundness properties (in particular: only true things are known/believed, and the agent is aware of what she knows/believes and what she does not know/believe), then this is captured by equivalence relations. If some of the soundness properties are abandoned or weakened, then one has to use more general kinds of relations.

For both kinds of applications, transitivity plays a very important rôle. First, in all of the above examples of temporally relevant frame classes, the future relation is transitive (and has other properties as well). The class of transitive frames is a general case of all these temporal applications. Second, transitivity is similarly fundamental in epistemic applications because it corresponds to the property that agents are aware of their knowledge or their belief. As in the temporal case, other properties can — but need not — be added, but transitivity is rarely left out.

Modal, hybrid, and first-order logics over transitive models have been studied recently in [ABM00, GMV99, ST01, Kie02, Kie03, IRR$^+$04, DO05]. Although the complexity of satisfiability for hybrid (temporal) logics has been extensively examined [BS95, Gor96, ABM99, ABM00, FdRS03], there are highly expressive hybrid languages for whose satisfiability problems only results over arbitrary, but not over restricted, temporally or epistemically relevant frame classes have been known. This confirms the need for a classification of complexity for satisfiability of hybrid logic over such frame classes.

Furthermore, for the (general) model checking problem, only results over arbitrary frames have been known [FdR06]. We will find out whether the above mentioned level of complexity for binder languages persists if we restrict the class of frames. (The word "general" means that we will examine the model-checking problem considered in [FdR06], restricted to certain classes of frames, as opposed to the linear-time model-checking problem from [SC85] and [FdRS03].)

The frame classes that we will consider are the class of all frames, transitive frames, transitive trees, linear frames, the natural numbers, frames with equivalence relations, and complete frames.

1.3 The complexity of multi-modal hybrid logics

The classification of the satisfiability problem for hybrid languages over different frame classes will show that satisfiability for the language with the more restricted form of a hybrid binder, which is undecidable over arbitrary frames [ABM99], will become decidable over transitive frames [MSSW05]. We will not only show that satisfiability for languages combining this binder with other operators is undecidable over transitive frames. We will also examine another extension of this binder language over a wide range of frame classes, namely its multi-modal version. Our (undecidability) results will cover, among others, frame classes that are important for epistemic applications, because the multi-modal setting corresponds to multi-agent scenarios.

1.4 Legend to this thesis

This thesis is organised as follows. In Chapter 2, we will give all definitions and notations that are necessary for modal, temporal, hybrid, and first-order logic. We will also introduce the basic concepts of computational complexity and tools used to establish complexity bounds of certain logics. Chapter 3 is concerned with expressivity issues and establishes hierarchies of hybrid languages over different classes of frames. Chapters 4 and 5 examine the model-checking problem and satisfiability of hybrid languages over these frame classes. Satisfiability of multi-modal binder logic is considered in Chapter 6. Chapter 7 gives an overview of all achieved results and contains remarks on each group of results from the previous chapters.

Parts of this thesis have appeared in proceedings of workshops or in journals. In particular, Sections 5.4 and 5.5 contain results from [MSSW05] and [MSSW07], Section 5.8 has appeared as [MS07a], and Chapter 6 improves on [MS07b].

Chapter 2

Preliminaries

2.1 Hybrid logic

We define the basic concepts and notations of modal and hybrid logic that are relevant for this thesis. The fundamentals of modal logic are mainly taken from [BdRV05]; those of hybrid logic from [ABM99, Bla00b, AtC06].

2.1.1 Syntax

As indicated in the previous chapter, *the* hybrid language does not exist. Rather there are several extensions of the modal language that allow for explicit references to states and incorporate very restricted versions of first-order quantifiers — hence the attribute "hybrid". We will introduce the largest and most expressive hybrid language that will interest us in this thesis. It contains four temporal operators, two hybrid binders, satisfaction operators, and the global modality. Later on, we will define fragments of this full language.

We will give the syntax of hybrid logic inductively in the usual manner. For Boolean, modal, and hybrid operators that appear in duals, Definition 2.1 gives only the "existential" operators $\bot$, $\vee$, F, etc. in the induction and defines the remaining operators as abbreviations.

Definition 2.1 *Let* PROP *be a countable set of propositional atoms,* NOM *be a countable set of nominals,* SVAR *be a countable set of state variables, and let* ATOM $=$ PROP $\cup$ NOM $\cup$ SVAR.

(1) *The full hybrid language* $\mathcal{HL}(\mathsf{F}, \mathsf{P}, \mathsf{U}, \mathsf{S}, \downarrow, \exists, @, \mathsf{E})$ *is the set of all formulae of the form*

$$\varphi ::= a \mid \bot \mid \neg\varphi \mid \varphi \vee \varphi' \mid \mathsf{F}\varphi \mid \mathsf{P}\varphi \mid \varphi\mathsf{U}\psi \mid \varphi\mathsf{S}\psi \mid {\downarrow} x.\varphi \mid \exists x.\varphi \mid @_t\varphi \mid \mathsf{E}\varphi\,,$$

where $a \in$ ATOM, $t \in$ NOM $\cup$ SVAR, *and* $x \in$ SVAR.

(2) *We use the following abbreviations.*

$$\begin{aligned} \top &= \neg\bot & \mathsf{G}\varphi &= \neg\mathsf{F}\neg\varphi \\ \varphi \wedge \psi &= \neg(\neg\varphi \vee \neg\psi) & \mathsf{H}\varphi &= \neg\mathsf{P}\neg\varphi \\ \varphi \to \psi &= \neg\varphi \vee \psi & \forall x.\varphi &= \neg\exists x.\neg\varphi \\ \varphi \leftrightarrow \psi &= (\varphi \to \psi) \wedge (\psi \to \varphi) & \mathsf{A}\varphi &= \neg\mathsf{E}\neg\varphi \end{aligned}$$

(3) *Let φ be a formula and x be a state variable.*

- *For any occurrence of the $\downarrow$ or $\exists$ operator in φ that begins a subformula $\downarrow x.\psi$ or $\exists x.\psi$ of φ, its* scope *is ψ.*
- *Any occurrence of x in φ is called* bound *iff it is within the scope of some occurrence of the $\downarrow$ or $\exists$ operator in φ.*
- *x is* free *in φ iff it does not occur bound in φ.*

(4) *A hybrid formula is called*

- pure *iff it contains no propositional atoms;*
- nominal-free *iff it contains no nominals;* *and*
- *a* sentence *iff it contains no free state variables.*

(5) *For each formula φ, we use $\mathrm{PROP}(\varphi)$, $\mathrm{NOM}(\varphi)$, and $\mathrm{SVAR}(\varphi)$ to denote, respectively, the set of all atomic propositions, nominals, and state variables that occur in φ.*

It is common practice to denote propositional atoms by $p, q, \dots$; nominals by $i, j, \dots$; and state variables by $x, y, \dots$. The operators F, G, P, H, U, and S are called *temporal operators*, $\downarrow$, $\exists$, and $\forall$ are called *hybrid binders*, $@_t$ are *satisfaction operators*, and E and A are referred to as *global modalities*. The operators $\wedge$, G, H, $\forall$, and A are said to be the *duals* of $\vee$, F, P, $\exists$, and E, respectively.

2.1.2 Semantics

Semantics is defined in terms of *Kripke models*. In order to evaluate formulae with binders, an assignment from the set of all state variables to the set of states is necessary. This assignment can be omitted whenever binder-free sublanguages or only sentences are considered.

Definition 2.2

(1) *A* frame *is a pair $\mathcal{F} = (M, R)$ with the following components.*

- *M is a nonempty set of states.*[1]
- *$R \subseteq M \times M$ is a binary relation — the accessibility relation.*

(2) *A (hybrid Kripke) model is a triple $\mathcal{M} = (M, R, V)$, where (M, R) is a frame, and $V : \mathrm{PROP} \to \mathfrak{P}(M)$ is a function — the valuation function. It is required that, for each $i \in \mathrm{NOM}$, $|V(i)| = 1$.*

(3) *Given a frame $\mathcal{F} = (M, R)$ and a model $\mathcal{M} = (M, R, V)$, we say that $\mathcal{M}$ is based on $\mathcal{F}$ and $\mathcal{F}$ underlies $\mathcal{M}$.*

(4) *An assignment for a model $\mathcal{M} = (M, \mathcal{R}, V)$ is a function $g : \mathrm{SVAR} \to M$.*

(5) *Given an assignment g, a state variable x, and a state m, an x-variant g^x_m of g is defined by*

$$g^x_m(x') = \begin{cases} m & \text{if } x' = x, \\ g(x') & \text{otherwise.} \end{cases}$$

In order to define satisfaction and satisfiability, we use the meta-logical symbols &, or, $\Rightarrow$, $\Leftrightarrow$, ⓔ, and ⓐ in order to distinguish connectives and quantifiers of the meta-language (i.e., "and", "or", "implies", "if and only if", "*there exists* a state from M", and "*for all* states ...", respectively) from the operators in our logic. In addition, wherever space is short and misunderstandings are impossible, we will omit the addition "$\in M$" from quantified meta-variables.

Definition 2.3

(1) *For any atom a, let*

$$[V, g](a) = \begin{cases} \{g(a)\} & \text{if } a \in \mathrm{SVAR}, \\ V(a) & \text{otherwise.} \end{cases}$$

(2) *Given a hybrid model $\mathcal{M} = (M, R, V)$, an assignment g for $\mathcal{M}$, and a state $m \in M$, the satisfaction relation is as follows.*

$\mathcal{M}, g, m \Vdash a$	*iff*	$m \in [V, g](a)$, $a \in \mathrm{ATOM}$
$\mathcal{M}, g, m \Vdash \bot$	*never*	
$\mathcal{M}, g, m \Vdash \neg\varphi$	*iff*	$\mathcal{M}, g, m \nVdash \varphi$
$\mathcal{M}, g, m \Vdash \varphi \vee \psi$	*iff*	$\mathcal{M}, g, m \Vdash \varphi$ or $\mathcal{M}, g, m \Vdash \psi$
$\mathcal{M}, g, m \Vdash \mathsf{F}\varphi$	*iff*	ⓔ$n \in M(mRn \;\&\; \mathcal{M}, g, n \Vdash \varphi)$
$\mathcal{M}, g, m \Vdash \mathsf{P}\varphi$	*iff*	ⓔ$n \in M(nRm \;\&\; \mathcal{M}, g, n \Vdash \varphi)$
$\mathcal{M}, g, m \Vdash \varphi\mathsf{U}\psi$	*iff*	ⓔ$n\big(mRn \;\&\; \mathcal{M}, n \Vdash \psi \;\&\;$ ⓐ$s(mRsRn \Rightarrow \mathcal{M}, s \Vdash \varphi)\big)$
$\mathcal{M}, g, m \Vdash \varphi\mathsf{S}\psi$	*iff*	ⓔ$n\big(nRm \;\&\; \mathcal{M}, n \Vdash \psi \;\&\;$ ⓐ$s(nRsRm \Rightarrow \mathcal{M}, s \Vdash \varphi)\big)$

[1] We predominantly prefer "states" to "worlds" because this seems to be the most neutral, application-independent term.

$$\begin{array}{lll}
\mathcal{M}, g, m \Vdash {\downarrow} x.\varphi & \textit{iff} & \mathcal{M}, g^x_m, m \Vdash \varphi \\
\mathcal{M}, g, m \Vdash \exists x.\varphi & \textit{iff} & \textcircled{\scriptsize$\exists$} n \in M(\mathcal{M}, g^x_n, m \Vdash \varphi) \\
\mathcal{M}, g, m \Vdash @_t\varphi & \textit{iff} & \mathcal{M}, g, n \Vdash \varphi, \text{ where } [V, g](t) = \{n\} \\
\mathcal{M}, g, m \Vdash \mathsf{E}\varphi & \textit{iff} & \textcircled{\scriptsize$\exists$} n \in M(\mathcal{M}, g, n \Vdash \varphi)
\end{array}$$

(3) *If all states from $\mathcal{M}$ satisfy φ under g, we write $\mathcal{M}, g \Vdash \varphi$ and say that φ is globally satisfied by $\mathcal{M}$ under g.*

(4) *A formula φ is satisfiable if there exist a model $\mathcal{M} = (M, R, V)$, an assignment g for $\mathcal{M}$, and a state $m \in M$, such that $\mathcal{M}, g, m \Vdash \varphi$.*

(5) *A formula φ is globally satisfiable if there exist a model $\mathcal{M} = (M, R, V)$ and an assignment g for $\mathcal{M}$, such that $\mathcal{M}, g \Vdash \varphi$.*

2.1.3 Other operators of interest

Whenever the choice of temporal operators is restricted to F and G, it is common to write $\Diamond$ and $\Box$ instead of F and G, respectively. This reflects the fact that modal and hybrid logic are used for far more purposes than only to express temporal properties. We will often use these more general operator names when working with "non-temporal" languages.

Besides the full hybrid language and fragments thereof, two generalisations will be of interest for this thesis. The first is *multi-modal hybrid logic*. This is non-temporal hybrid logic with several operators $\Diamond_\ell, \Box_\ell, \ell = 1, \ldots, n$, for some positive integer n. In this case, the concepts of a frame and a model have to be extended. Instead of one accessibility relation, we have a tuple $(R_1, \ldots, R_n)$ of accessibility relations. Now, of course, the satisfaction relation for $\Diamond_\ell$-formulae refers to R_ℓ.

The second generalisation concerns the until/since operators. In [ABM00], a variant, U^+ and S^+, is introduced in order to "simulate" transitive accessibility relations syntactically. We will make use of a further modification which we call U^{++} and S^{++}. The resulting temporal language is an even closer simulation of transitivity, as we will see in Section 5.4.1.

Definition 2.4 *Given a binary relation R over some set M, we use*

(1) *R^+ to denote the transitive closure of R, that is,*

$$R^+ = \{(m, n) \in M \times M \mid \textcircled{\scriptsize$\exists$} p_0, \ldots, p_k \in M \text{ with } p_0 = m,\ p_k = n,\ p_0 R \ldots R p_n\};$$

(2) *R^* to denote the reflexive transitive closure of R, that is,*

$$R^* = R^+ \cup \{(m, m) \mid m \in M\}.$$

frame class	abbr.	properties of each member (M,R) of this class
arbitrary frames	—	—
trees	tree	acyclic, connected, each point has at most one R-predecessor
transitive frames	trans	R is transitive
transitive trees	tt	$R = S^+$, where (M,S) is a tree
linear orders	lin	R is transitive, irreflexive, and trichotomous — trichotomy: $\big(\text{Ⓐ}xy(xRy \text{ or } x = y \text{ or } yRx)\big)$
natural numbers	$(\mathbb{N}, >)$	$(M,R) = (\mathbb{N}, >)$
ER frames	ER	R is an equivalence relation
complete frames	compl	$R = M \times M$

Table 2.1: Relevant frame classes, their abbreviations and definitions

Definition 2.5 *Given a model $\mathcal{M} = (M,R,V)$ and a state $m \in M$, the satisfaction relation for the new until/since operators is defined as follows.*

$$\mathcal{M}, m \Vdash \varphi \mathsf{U}^+ \psi \;\; \textit{iff} \;\; \text{Ⓔ}n\big(mRn \;\&\; \mathcal{M}, n \Vdash \psi \;\&\; \text{Ⓐ}s(m\boldsymbol{R}^+ s\boldsymbol{R}^+ n \Rightarrow \mathcal{M}, s \Vdash \varphi)\big)$$

$$\mathcal{M}, m \Vdash \varphi \mathsf{S}^+ \psi \;\; \textit{iff} \;\; \text{Ⓔ}n\big(nRm \;\&\; \mathcal{M}, n \Vdash \psi \;\&\; \text{Ⓐ}s(n\boldsymbol{R}^+ s\boldsymbol{R}^+ m \Rightarrow \mathcal{M}, s \Vdash \varphi)\big)$$

$$\mathcal{M}, m \Vdash \varphi \mathsf{U}^{++} \psi \;\; \textit{iff} \;\; \text{Ⓔ}n\big(m\boldsymbol{R}^+ n \;\&\; \mathcal{M}, n \Vdash \psi \;\&\; \text{Ⓐ}s(mR^+ sR^+ n \Rightarrow \mathcal{M}, s \Vdash \varphi)\big)$$

$$\mathcal{M}, m \Vdash \varphi \mathsf{S}^{++} \psi \;\; \textit{iff} \;\; \text{Ⓔ}n\big(n\boldsymbol{R}^+ m \;\&\; \mathcal{M}, n \Vdash \psi \;\&\; \text{Ⓐ}s(nR^+ sR^+ m \Rightarrow \mathcal{M}, s \Vdash \varphi)\big)$$

2.1.4 Properties of models and frames

Definition 2.6 introduces the frame classes that are relevant for our considerations. Parts (1)–(4) carry over straightforwardly to the multi-modal case by replacing R by R_ℓ and requiring that the defined frame properties hold for *each* (M, R_ℓ).

Definition 2.6 *Let $\mathcal{M} = (M,R,V)$ be a hybrid model with the underlying frame $\mathcal{F} = (M,R)$.*

(1) For any subset $M' \subseteq M$, we write $R{\restriction}_{M'}$ and $V{\restriction}_{M'}$ for the restrictions of R and V to M'.

(2) The frame classes used in this thesis and their abbreviations are given in Table 2.1. The properties from Column 3 are used for frames as well as for models.

In addition, we will need two more basic concepts connected with models.

Definition 2.7 *Let $\mathcal{M} = (M, R, V)$ be a model.*

(1) A submodel of $\mathcal{M}$ is a model $\mathcal{M}' = (M', R'V')$ with $M' \subseteq M$, $R' = R\restriction_{M'}$, and $V' = V\restriction_{M'}$.

(2) A cluster of $\mathcal{M}$ is a maximal complete submodel, that is, a submodel $\mathcal{M}' = (M', R', V')$ such that $R\restriction_{M'} = M' \times M'$, and for each set M'' with $M' \subseteq M'' \subseteq M$, it holds that $R\restriction_{M''} \subset M'' \times M''$.

(3) Given a state $m_0 \in M$, the submodel of $\mathcal{M}$ generated by m_0 is the smallest submodel $\mathcal{M}' = (M', R'V')$ that contains m_0 and satisfies the condition that, for each $m, n \in M$, if $m \in M'$ and mRn, then $n \in M'$.

Such a submodel is also called point-generated, and m_0 is its root.

It is well-known [BdRV05] that the truth of formulae from $\mathcal{ML}(\Diamond)$ is preserved under taking point-generated submodels, that is, whenever $\mathcal{M}, m_0 \Vdash \varphi$, then $\mathcal{M}', m_0 \Vdash \varphi$, where $\mathcal{M}'$ is the submodel of $\mathcal{M}$ generated by φ. This property carries over to hybrid languages without Past, satisfaction operators and global modalities.

2.1.5 Hybrid languages and their decision problems

In this thesis we examine decision problems of hybrid languages over those classes of frames that were introduced in Subsection 2.1.4. "Languages" refers to fragments of the full hybrid language $\mathcal{HL}(\mathsf{F}, \mathsf{P}, \mathsf{U}, \mathsf{S}, \downarrow, \exists, @, \mathsf{E})$ as well as the generalisations introduced in Subsection 2.1.3. The following conventions ensure a uniform notation of all relevant languages, introduce frame properties and corresponding frame classes, and establish decision problems of hybrid logics.

We will denote fragments of the full hybrid language simply by omitting those operators that are not in the respective language. The fragment $\mathcal{HL}(\Diamond) = \mathcal{HL}(\mathsf{F})$ is referred to as the *minimal hybrid language*. Multi-modal languages are denoted by $\mathcal{HL}_n(\Diamond, \dots)$, where n is the number of modalities.

Whenever we have neither nominals, nor binders, nor satisfaction operators in our language, we use $\mathcal{ML}(\dots)$ to denote the respective (modal) language. The fragment $\mathcal{ML}(\Diamond)$ is called the *basic modal language*. Furthermore, we denote the *pure fragments* (i.e., without atomic propositions) of $\mathcal{ML}(\cdot)$ and $\mathcal{HL}(\cdot)$ by $\mathcal{PML}(\cdot)$ and $\mathcal{PHL}(\cdot)$, respectively.

Each hybrid language has decision problems over each class of frames. This thesis will mainly be concerned with satisfiability, but also with model checking. Definition 2.8 introduces these two problems, as well as the *global* satisfiability problem, which we will occasionally refer to, too.

Definition 2.8 *Let L be a hybrid language and $\mathfrak{F}$ be frame class.*

(1) *The satisfiability problem L-$\mathfrak{F}$-SAT is defined as follows: Given a formula $\varphi \in L$, is φ satisfiable in a model based on a frame from $\mathfrak{F}$?*
(2) *The global satisfiability problem L-$\mathfrak{F}$-GLOBSAT is this: Given a formula $\varphi \in L$, is φ globally satisfiable in a model based on a frame from $\mathfrak{F}$?*
(3) *The model-checking problem L-$\mathfrak{F}$-MC is defined as follows: Given a formula $\varphi \in L$, a finite model $\mathcal{M}$ based on a frame from $\mathfrak{F}$, and an assignment g for $\mathcal{M}$, is there a state m in $\mathcal{M}$ such that $\mathcal{M}, g, m \Vdash \varphi$?*
(4) *If $\mathfrak{F}$ is the class of all frames, then $\mathfrak{F}$ may be left out of either notation.*

This notation is demonstrated by the following examples.

- The satisfiability problem over linear orders for the hybrid until/since language with the @ operator is denoted by $\mathcal{HL}(\mathsf{U}, \mathsf{S}, @)$-lin-SAT.
- The satisfiability problem over transitive frames for the bi-modal hybrid $\downarrow$ language is denoted by $\mathcal{HL}_2(\Diamond, \downarrow)$-trans-SAT.
- The global satisfiability problem for the basic modal language is denoted by $\mathcal{ML}(\Diamond)$-GLOBSAT.
- The model-checking problem over arbitrary frames for the full hybrid language is denoted by $\mathcal{HL}(\mathsf{F}, \downarrow, \mathsf{E})$-MC — with the remark that $\mathcal{HL}(\mathsf{F}, \downarrow, \mathsf{E})$ is already as expressive as the full hybrid language, with an efficiently computable translation function, as will be shown in Chapter 3.

2.1.6 Bounded model properties

The following properties are helpful for establishing decidability or upper complexity bounds of satisfiability for logics.

Definition 2.9 *Let L be a hybrid language, $\mathfrak{F}$ be a class of frames, and $f : \mathbb{N} \to \mathbb{N}$ be a computable function.*

(1) *L has the f-size model property with respect to $\mathfrak{F}$ iff every formula $\varphi \in L$-$\mathfrak{F}$-SAT is satisfiable in a model from $\mathfrak{F}$ that has at most $f(|\varphi|)$ states, where $|\varphi|$ denotes the length of φ.*
(2) *L has the bounded model property with respect to $\mathfrak{F}$ iff it has the g-size model property with respect to $\mathfrak{F}$, for some computable function g.*
(3) *L has the finite model property with respect to $\mathfrak{F}$ iff every formula $\varphi \in L$-$\mathfrak{F}$-SAT is satisfiable in a model from $\mathfrak{F}$ that has finitely many states.*

When we speak of L having the $\mathcal{O}(f)$-size model property, we mean that it has the g-size model property for some computable $f \in \mathcal{O}(g)$.

2.2 First-order logic

2.2.1 Basic concepts

Modal and hybrid logic can be embedded into fragments of first-order logic (FOL). Since we will make use of such embeddings, we define the basic notions of FOL here following the standard notation as introduced, for example, in [EFT96, Fit96].

Definition 2.10 *Let* VAR *be a set of variables. For each $n \in \mathbb{N}$, let* FUNC_n *be a set of n-ary function symbols. For each $n \in \mathbb{N} - \{0\}$, let* REL_n *be a set of n-ary relation symbols.*

(1) A term has the form

$$t ::= x \mid f(t_1, \ldots, t_n),$$

where $x \in \mathrm{VAR}$, $n \in \mathbb{N}$, and $f \in \mathrm{FUNC}_n$.

(2) The first-order language $\mathcal{FOL}$ is the set of all formulae of the form

$$\alpha ::= \bot \mid t_1 = t_2 \mid R(t_1, \ldots, t_n) \mid \neg\alpha \mid \alpha \vee \alpha' \mid \exists x.\alpha,$$

where $n \in \mathbb{N} - \{0\}$, $R \in \mathrm{REL}_n$, $x \in \mathrm{VAR}$, and t_i are terms.[2]

(3) A formula α containing exactly the free variables $x_1, \ldots, x_n$ is denoted by $\alpha(x_1, \ldots, x_n)$.

We use the abbreviations $\top$, $\wedge$, $\rightarrow$, $\leftrightarrow$, and $\forall$ from Definition 2.1 (2). The terms *scope*, *bound/free variable*, and *sentence* are analogous to those from Definition 2.1 (3)–(4).

First-order logic is interpreted in terms of models and assignments.

Definition 2.11

(1) A model is a pair $\mathcal{M} = (D, I)$ with the following components.

- *D is a nonempty set, the domain.*
- *I, the interpretation, is a mapping that assigns*
  - *to every n-ary function symbol $f \in \mathrm{FUNC}_n$ some n-ary function $f^I : D^n \rightarrow D$, and*
  - *to every n-ary relation symbol $R \in \mathrm{REL}_n$ some n-ary relation $R^I \subseteq D^n$.*

[2] In the case of binary relations, we prefer the infix notation xRy to $R(x, y)$.

(2) *Given a model $\mathcal{M} = (D, I)$, an assignment for $\mathcal{M}$ is a function $g : \mathrm{VAR} \to D$.*

(3) *Given a model $\mathcal{M} = (D, I)$, two assignments g, h for $\mathcal{M}$, and a variable x; h is an x-variant of g iff h assigns the same values to every variable except possibly for x.*

The combination of an interpretation and an assignment makes it possible to define values for arbitrary terms as follows.

Definition 2.12 *Given a model $\mathcal{M} = (D, I)$, an assignment g for $\mathcal{M}$, and a term t, the value $t^{I,g}$ of t is given as follows.*

- *For a variable x, $x^{I,g} = g(x)$.*
- *For a function symbol $f \in \mathrm{FUNC}_n$, $\big(f(t_1, \dots, t_n)\big)^{I,g} = f^I\big(t_1^{I,g}, \dots, t_n^{I,g}\big)$.*

Definition 2.13 *Let $\mathcal{M} = (D, I)$ be a model and g an assignment for $\mathcal{M}$.*

(1) *The satisfaction relation is defined as follows.*

$$\begin{array}{ll}
\mathcal{M} \models \bot\,[g] & \textit{never} \\
\mathcal{M} \models t_1 = t_2\,[g] & \textit{iff } t_1^{I,g} \textit{ and } t_2^{I,g} \textit{ are equal} \\
\mathcal{M} \models R(t_1, \dots, t_n)\,[g] & \textit{iff } (t_1^{I,g}, \dots, t_n^{I,g}) \in R^I \\
\mathcal{M} \models \neg\alpha\,[g] & \textit{iff } \mathcal{M} \not\models \alpha\,[g] \\
\mathcal{M} \models \alpha \vee \beta\,[g] & \textit{iff } \mathcal{M} \models \alpha\,[g] \textit{ or } \mathcal{M} \models \beta\,[g] \\
\mathcal{M} \models \exists x.\alpha\,[g] & \textit{iff } \mathcal{M} \models \alpha\,[h] \textit{ for some } x\textit{-variant } h \textit{ of } g
\end{array}$$

(2) *Given a formula $\alpha(x_1, \dots, x_n)$ and elements $d_1, \dots, d_n \in D$, we also write $\mathcal{M} \models \alpha\,[d_1, \dots, d_n]$ instead of $\mathcal{M} \models \alpha\,[g]$, provided that $g(x_i) = d_i$ for each $i = 1, \dots, n$.*

(3) *In the case of (2), if the assignment g is denoted by $d_1, \dots, d_n$, then the expression $d_1, \dots, d_n, x \mapsto d$ stands for that x-variant of g which assigns d to x, where $x \in \mathrm{VAR}$ and $d \in D$.*

(4) *A formula α is satisfiable iff there exist a model $\mathcal{M}$ and an assignment g for $\mathcal{M}$ such that $\mathcal{M} \models \alpha\,[g]$.*

(5) *The satisfiability problem $\mathcal{FOL}$-SAT is the following: Given a formula α, is α satisfiable?*

Note that, for sentences, satisfaction is independent of assignments. This leads to the notion of truth.

Definition 2.14 *If a sentence α is satisfied in some model $\mathcal{M}$ (under any assignment), we write $\mathcal{M} \models \alpha$ and say that α is true in $\mathcal{M}$.*

2.2.2 Fragments of FOL and the standard translation

To embed hybrid logic into first-order logic, it is sufficient to restrict the vocabulary to nullary function symbols (constant symbols), unary and binary relation symbols and, if no binders are present, a constant number of variables. We do not discuss further restrictions here that are sufficient for such an embedding.

We will denote fragments of $\mathcal{FOL}$ in the style of [BGG97]. We only introduce that part of the terminology which is relevant for the following chapters of this thesis.

Definition 2.15 *Let $a, b, c \in \mathbb{N}$.*

(1) The fragment of $\mathcal{FOL}$ that permits no equality, no relation symbols other than a unary ones and b binary ones, and no function symbols other than c nullary ones, is denoted by $[\text{all}, (a,b), (c)]$.
If equality is permitted, we write $[\text{all}, (a,b), (c)]_=$.
We abbreviate $[\text{all}, (a,b), (0)]$ by $[\text{all}, (a,b)]$.

(2) The satisfiability problem $[\text{all}, (a,b), (c)]$-SAT is the following: Given a formula $\alpha \in [\text{all}, (a,b), (c)]$, is α satisfiable?

(3) The satisfiability problem $[\text{all}, (a,b), (c)]$-trans-SAT is the following: Given a formula $\alpha \in [\text{all}, (a,b), (c)]$, is α satisfiable in a model that interprets each binary relation symbol by a transitive relation?

Hybrid logic (and hence modal logic) can be embedded into first-order logic. Let us first consider the full hybrid language minus the until/since operators. This language can be embedded canonically into the fragment $[\text{all}, (\omega, 1), (\omega)]$, restricted to two variables, via the *Standard Translation* ST [tCF05b]. This translation consists of two functions ST_x and ST_y, defined recursively. Since ST_y is obtained from ST_x by exchanging x and y, we only give ST_x here.

$$
\begin{aligned}
\mathrm{ST}_x(p) &= P(x) & \mathrm{ST}_x(\mathsf{F}\varphi) &= \exists y.\big(xRy \wedge \mathrm{ST}_y(\varphi)\big)\\
\mathrm{ST}_x(t) &= t = x & \mathrm{ST}_x(\mathsf{P}\varphi) &= \exists y.\big(yRx \wedge \mathrm{ST}_y(\varphi)\big)\\
\mathrm{ST}_x(\bot) &= \bot & \mathrm{ST}_x({\downarrow} v.\varphi) &= \exists v.\big(x = v \wedge \mathrm{ST}_x(\varphi)\big)\\
\mathrm{ST}_x(\neg\varphi) &= \neg\,\mathrm{ST}_x(\varphi) & \mathrm{ST}_x(\exists v.\varphi) &= \exists v.\big(\mathrm{ST}_x(\varphi)\big)\\
\mathrm{ST}_x(\varphi \vee \psi) &= \mathrm{ST}_x(\varphi) \vee \mathrm{ST}_x(\psi) & \mathrm{ST}_x(@_t\varphi) &= \exists y.\big(y = t \wedge \mathrm{ST}_y(\varphi)\big)\\
& & \mathrm{ST}_x(\mathsf{E}\varphi) &= \exists y.\big(\mathrm{ST}_y(\varphi)\big)
\end{aligned}
$$

Here, $p \in \mathrm{PROP}$, $t \in \mathrm{NOM} \cup \mathrm{SVAR}$, and $v \in \mathrm{SVAR}$.

name of class	TM model	restriction
P	deterministic	polynomial time
NP	nondeterministic	polynomial time
PSPACE	deterministic	polynomial space
NPSPACE	nondeterministic	polynomial space
EXPTIME	deterministic	exponential time
NEXPTIME	nondeterministic	exponential time
2EXPTIME	deterministic	doubly exponential time
N2EXPTIME	nondeterministic	doubly exponential time

Table 2.2: An overview of complexity classes used in this thesis

The operators U and S, as well as their variants from Definition 2.5, require three variables for an embedding. Hence, for until/since languages, ST consists of three functions ST_x, ST_y, and ST_z, where ST_x is extended as follows. ST_y and ST_z are defined by exchanging x, y, z cyclically.

$$
\begin{aligned}
\mathrm{ST}_x(\varphi \mathsf{U} \psi) &= \exists y.\big[xRy \wedge \mathrm{ST}_y(\psi) \wedge \forall z.\big((xRz \wedge zRy) \rightarrow \mathrm{ST}_z(\varphi)\big)\big] \\
\mathrm{ST}_x(\varphi \mathsf{S} \psi) &= \exists y.\big[yRx \wedge \mathrm{ST}_y(\psi) \wedge \forall z.\big((yRz \wedge zRx) \rightarrow \mathrm{ST}_z(\varphi)\big)\big] \\
\mathrm{ST}_x(\varphi \mathsf{U}^{+} \psi) &= \exists y.\big[xRy \wedge \mathrm{ST}_y(\psi) \wedge \forall z.\big((xR^{+}z \wedge zR^{+}y) \rightarrow \mathrm{ST}_z(\varphi)\big)\big] \\
\mathrm{ST}_x(\varphi \mathsf{S}^{+} \psi) &= \exists y.\big[yRx \wedge \mathrm{ST}_y(\psi) \wedge \forall z.\big((yR^{+}z \wedge zR^{+}x) \rightarrow \mathrm{ST}_z(\varphi)\big)\big] \\
\mathrm{ST}_x(\varphi \mathsf{U}^{++} \psi) &= \exists y.\big[xR^{+}y \wedge \mathrm{ST}_y(\psi) \wedge \forall z.\big((xR^{+}z \wedge zR^{+}y) \rightarrow \mathrm{ST}_z(\varphi)\big)\big] \\
\mathrm{ST}_x(\varphi \mathsf{S}^{++} \psi) &= \exists y.\big[yR^{+}x \wedge \mathrm{ST}_y(\psi) \wedge \forall z.\big((yR^{+}z \wedge zR^{+}x) \rightarrow \mathrm{ST}_z(\varphi)\big)\big]
\end{aligned}
$$

2.3 Computational complexity

In order to establish how many resources are necessary to solve decision problems of hybrid logics, we use the common terminology from complexity theory, which is based on Turing machines and includes — amongst others — concepts on which we build our considerations, namely decision problems, complexity classes, and reductions. An introduction into this terminology can be found, for instance, in [Pap94].

In this section, we will not explain the classical concept of a Turing machine, but we will introduce complexity classes and reductions. We will always speak of decision problems, as opposed to computation problems, counting problems, etc.

Definition 2.16

(1) *Complexity classes are defined according to Table 2.2: For each row from the table, the class with the name from Column 1 is the set of all problems decidable by some Turing machine with the property from Column 2 whose runtime or tape length is restricted as given in Column 3.*

(2) *A problem is* elementary *iff it is decidable by some Turing machine whose runtime is restricted by $2^{2^{\cdot^{\cdot^{2^n}}}}$, for some finite number of exponents. A problem is* nonelementarily decidable *iff it is decidable and not elementary.*

(3) *A problem is in* coRE *iff it is the complement of a recursively enumerable problem.*

(4) *A problem $A \subseteq X$ is* polynomial-time reducible *to a problem $B \subseteq Y$, written $A \leqslant^{\mathrm{P}}_{\mathrm{m}} B$, iff there is a polynomial-time computable function $f : X \to Y$ such that for all $x \in X$ the following equivalence holds.*

$$x \in A \qquad \Leftrightarrow \qquad f(x) \in B$$

(5) *The problems A and B are* polynomial-time equivalent*, written $A \equiv^{\mathrm{P}}_{\mathrm{m}} B$, iff $A \leqslant^{\mathrm{P}}_{\mathrm{m}} B$ and $B \leqslant^{\mathrm{P}}_{\mathrm{m}} A$.*

(6) *Let $\mathcal{C}$ be a complexity class and A be a problem. A is called* $\mathcal{C}$-hard *iff, for each $C \in \mathcal{C}$, $C \leqslant^{\mathrm{P}}_{\mathrm{m}} A$. A is called* $\mathcal{C}$-complete *iff it is $\mathcal{C}$-hard and contained in $\mathcal{C}$.*

It is well-known that each complexity class $\mathcal{C}$ from Definition 2.16 is closed under polynomial-time reductions, that is, whenever $B \in \mathcal{C}$ and $A \leqslant^{\mathrm{P}}_{\mathrm{m}} B$, then $A \in \mathcal{C}$. This allows to show containment in $\mathcal{C}$ for some problem A by reducing it to a problem already known to be in $\mathcal{C}$.

Theorem 2.17 *The following inclusions between complexity classes hold.*

$$\begin{aligned} &\mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE} = \mathrm{NPSPACE} \subseteq \mathrm{EXPTIME} \subseteq \mathrm{NEXPTIME} \subseteq \\ &\subseteq \mathrm{2EXPTIME} \subseteq \mathrm{N2EXPTIME} \subseteq \mathrm{coRE} \end{aligned}$$

Furthermore, each decidable problem is in coRE.

2.4 Tools used for establishing complexity bounds

2.4.1 Quantified Boolean Formulae

A quantified Boolean formula (QBF) is a formula of the form

$$\alpha = \mathsf{Q}_1 x_1 \ldots \mathsf{Q}_n x_n . \beta,$$

where $\mathsf{Q}_i \in \{\exists, \forall\}$, for each $i = 1, \ldots, n$, and β is a Boolean formula with only $x_1, \ldots, x_n$ as propositional variables. We use the abbreviation QBF for the set of all QBF as well. The problem to decide whether a given QBF is valid is a well-known PSPACE-complete problem and is widely used for establishing complexity bounds.

Definition 2.18

(1) *The validity of a QBF is defined inductively as follows.*

- *The QBF* $\exists x_1.\beta$ *is valid iff* β *is satisfiable.*
- *The QBF* $\forall x_1.\beta$ *is valid iff* β *is a tautology.*
- *The QBF* $\exists x_1 \mathsf{Q}_2 x_2 \ldots \mathsf{Q}_n x_n.\beta$ *is valid iff* $\mathsf{Q}_2 x_2 \ldots \mathsf{Q}_n x_n.\beta[x_1/\bot]$ *or* $\mathsf{Q}_2 x_2 \ldots \mathsf{Q}_n x_n.\beta[x_1/\top]$ *is valid.*
- *The QBF* $\forall x_1 \mathsf{Q}_2 x_2 \ldots \mathsf{Q}_n x_n.\beta$ *is valid iff* $\mathsf{Q}_2 x_2 \ldots \mathsf{Q}_n x_n.\beta[x_1/\bot]$ *and* $\mathsf{Q}_2 x_2 \ldots \mathsf{Q}_n x_n.\beta[x_1/\top]$ *are valid.*

In these statements, the term $\beta[\gamma_1/\gamma_2]$ *denotes the formula obtained by* β *by substituting* γ_2 *for each occurrence of* γ_1*.*

(2) *The validity problem for QBF is the following.*

$$\mathrm{QSAT} = \{\alpha \in \mathrm{QBF} \mid \alpha \textit{ is valid}\}$$

Theorem 2.19 ([Sto77]) QSAT *is* PSPACE*-complete.*

2.4.2 Propositional dynamic logic for sibling-ordered trees

Kracht [Kra97] introduced a variant of propositional dynamic logic, which was referred to as "propositional dynamic logic for sibling-ordered trees" ($\mathcal{PDL}_{\mathrm{tree}}$) in [ABD+05], where its complexity was examined. We will use the latter result for showing an upper complexity bound of $\mathcal{HL}(\mathsf{U}, \mathsf{S}, \mathsf{E})$-tt-SAT in Section 5.5. This subsection defines syntax and semantics of $\mathcal{PDL}_{\mathrm{tree}}$.

$\mathcal{PDL}_{\mathrm{tree}}$ is the language of propositional dynamic logic with four atomic programmes left, right, up, and down that are associated with the relations "left sister", "right sister", "mother", and "daughter" in trees. Formulae and programmes are defined in a mutually inductive manner as follows.

Definition 2.20 *Let* PROP *be a countable set of propositional atoms.*

(1) *The language* $\mathcal{PDL}_{\mathrm{tree}}$ *is the set of all formulae of the form*

$$\varphi ::= p \mid \bot \mid \neg\varphi \mid \varphi \vee \varphi' \mid \langle\pi\rangle\, \varphi\,,$$

where $p \in$ PROP *and* π *is a programme.*

(2) *Programmes are given by*

$$\pi ::= \textsf{left} \mid \textsf{right} \mid \textsf{up} \mid \textsf{down} \mid \pi;\pi' \mid \pi \cup \pi' \mid \pi^* \mid \varphi?\,,$$

where φ is a formula.

(3) *We use the abbreviations $\top$, $\wedge$, $\rightarrow$, $\leftrightarrow$ from Definition 2.1 (2), as well as*

$$[\pi]\,\varphi = \neg\langle\pi\rangle\,\neg\varphi \qquad \text{and}$$
$$a^+ = a;a^*,$$

for atomic programmes a.

$\mathcal{PDL}_{\text{tree}}$ is interpreted over multi-modal models. Although there are infinitely many programmes and therefore infinitely many $\langle\cdot\rangle$ operators, accessibility relations for $\langle\textsf{down}\rangle$ and $\langle\textsf{right}\rangle$ suffice. They are extended inductively to arbitrary programmes. This induction and the induction for the satisfaction relation intertwine.

Definition 2.21

A $\mathcal{PDL}_{tree}$ model is a bi-modal model $\mathcal{M} = \big(T, (R_{down}, R_{right}), V\big)$, where

- *T is a finite, nonempty set;*
- *(T, R_{down}) is a tree (see Table 2.1) with an order relation on all immediate successors of any node;* and
- *R_{right} is the "next-sister" relation describing that order.*

The relations (which are meant to correspond to $\langle down\rangle$ and $\langle right\rangle$) are extended to arbitrary programmes as follows.

$$\begin{aligned} R_{up} &= R^-_{down} & R_{\pi\cup\pi'} &= R_\pi \cup R_{\pi'} \\ R_{left} &= R^-_{right} & R_{\pi^*} &= R^*_\pi \\ R_{\pi;\pi'} &= R_\pi \circ R_{\pi'} & R_{\varphi?} &= \{(m,m) \mid \mathcal{M}, m \Vdash \varphi\} \end{aligned}$$

(The notation R^- stands for the converse of R, and $R_1 \circ R_2$ denotes the set of all pairs (a,b) for which there is some c with aR_1cR_2b.)

Definition 2.22

(1) *Given a $\mathcal{PDL}_{\text{tree}}$ model $\mathcal{M} = \big(T, (R_{down}, R_{right}), V\big)$ and a state $m \in M$, the satisfaction relation is defined by*

$$\begin{array}{lll} \mathcal{M}, m \Vdash p & \textit{iff} \;\; m \in V(p), & p \in \text{PROP} \\ \mathcal{M}, m \Vdash \bot & \textit{never} & \\ \mathcal{M}, m \Vdash \neg\varphi & \textit{iff} \;\; \mathcal{M}, m \nVdash \varphi & \\ \mathcal{M}, m \Vdash \varphi \vee \psi & \textit{iff} \;\; \mathcal{M}, m \Vdash \varphi \text{ or } \mathcal{M}, m \Vdash \psi & \\ \mathcal{M}, m \Vdash \langle\pi\rangle\,\varphi & \textit{iff} \;\; \exists n \in T(mR_\pi n \;\&\; \mathcal{M}, n \Vdash \varphi) & \end{array}$$

(2) *A formula* $\varphi \in \mathcal{PDL}_{\text{tree}}$ *is satisfiable if there exists a* $\mathcal{PDL}_{\text{tree}}$ *model* $\mathcal{M} = \big(T, (R_{down}, R_{right}), V\big)$ *with* $\mathcal{M}, m \Vdash \varphi$*, where* m *is the root of* (T, R_{down})*.*
(3) *The satisfiability problem* $\mathcal{PDL}_{\text{tree}}$-SAT *is the following: Given a* $\mathcal{PDL}_{\text{tree}}$ *formula* φ*, is* φ *satisfiable?*

Theorem 2.23 ([ABD+05]) $\mathcal{PDL}_{\text{tree}}$-SAT *is in* EXPTIME*.*

2.4.3 First-order logic over strings

Let Σ be a non-empty set, called *alphabet*. The language of *first-order logic over strings* is the fragment of $\mathcal{FOL}$ with predicates restricted to two binary ones $<, =$, and unary ones P_σ for each $\sigma \in \Sigma$. It is interpreted over structures (D^n, I^s), where $n \in \mathbb{N}$ and $s = w_0 \dots w_{n-1}$ is a string with $w_i \in \Sigma$, for each $i = 0, \dots, n-1$. Furthermore, these structures satisfy the following conditions.

$$\begin{aligned} D^n &= \{0, \dots, n-1\} \\ I^s(<) &= \{(i,j) \in D \times D \mid i < j\} \\ I^s(P_\sigma) &= \{i \mid w_i = \sigma\} \end{aligned}$$

We will use $\mathcal{FOL}$-Strings-SAT to denote the satisfiability problem of first-order logic over strings, although this does not completely conform with the conventions under which we normally use this kind of notation.

Theorem 2.24 ([Sto74]) $\mathcal{FOL}$-Strings-SAT *is nonelementarily decidable.*

2.4.4 Tilings

Domino tiling problems are useful for establishing lower complexity bounds for logics. They have been proposed by Hao Wang [Wan61] and are defined in the following.

Definition 2.25 *Let C be a non-empty set whose members are called colours.*

(1) *A tile is a unit square, divided into four triangles by its diagonals.*
(2) *A tile type t is a quadruple* $t = \big(\text{left}(t), \text{right}(t), \text{top}(t), \text{bot}(t)\big) \in C^4$*.*
(3) *A tile is of type t iff its left, right, upper, and lower side have colours* $\text{left}(t)$, $\text{right}(t)$, $\text{top}(t)$, *and* $\text{bot}(t)$*, respectively.*

It is clear that, after a rotation, a tile of type t is not necessarily of type t again.

A tiling is a complete covering of a given subset of the $\mathbb{Z} \times \mathbb{Z}$ grid with tiles having certain types, such that each point (x, y) is covered by exactly one tile and adjacent tiles have the same colour at their common edges. This is put more formally in Definition 2.26.

Definition 2.26 *Let T be a set of tile types and $A \subseteq \mathbb{Z}\times\mathbb{Z}$.*
A T-tiling for A is a function $\tau : A \to T$ satisfying the following conditions for all $(x, y) \in A$.

$$\text{If } (x+1, y) \in A, \quad \text{then} \quad \mathrm{right}\left(\tau(x,y)\right) = \mathrm{left}\left(\tau(x+1,y)\right). \tag{2.1}$$
$$\text{If } (x, y+1) \in A, \quad \text{then} \quad \mathrm{top}\left(\tau(x,y)\right) = \mathrm{bot}\left(\tau(x,y+1)\right). \tag{2.2}$$

In order to easily refer to the set of tile types matching the type of a given tile in a tiling, we introduce the following notation.

Definition 2.27 *Let T be a tiling and $t \in T$ a tile type.*

$$\mathrm{RI}(t,T) = \{t' \in T \mid \mathrm{right}(t) = \mathrm{left}(t')\}$$
$$\mathrm{UP}(t,T) = \{t' \in T \mid \mathrm{top}(t) = \mathrm{bot}(t')\}$$

Conditions (2.1) and (2.2), then, are equivalent to

$$\text{If } (x+1, y) \in A, \quad \text{then} \quad \tau(x+1,y) \in \mathrm{RI}\left(\tau(x,y)\right).$$
$$\text{If } (x+1, y) \in A, \quad \text{then} \quad \tau(x,y+1) \in \mathrm{UP}\left(\tau(x,y)\right).$$

Definition 2.28

(1) *Let $A \subseteq \mathbb{Z}\times\mathbb{Z}$.*
The A-tiling problem is the following question: Given a finite set T of tile types, is there a T-tiling of A ?

(2) *Let $n \in \mathbb{N}$.*
The square tiling problem is the following question: Given a finite set T of tile types and a string 1^n of n consecutive 1s, is there a T-tiling of $\{0, \ldots, n-1\} \times \{0, \ldots, n-1\}$?

In [Rob71] it was shown that the $\mathbb{N}\times\mathbb{N}$-tiling problem and the $\mathbb{Z}\times\mathbb{Z}$-tiling problem are equivalent. We will refer to this problem as the *unbounded tiling problem*, whereas we will call A-tiling problems for finite sets A — including the square tiling problem — *bounded*.

Theorem 2.29

(1) *The unbounded tiling problem is* coRE-*complete. [Ber66]*

(2) *The square tiling problem is* NP-*complete. [SvEB84]*

Reductions from the unbounded tiling problem have, for instance, been used to prove undecidability of satisfiability for modal logics [Spa93a], products of modal logics [Mar99, GKWZ05], hybrid logics [BS95, Gor96, ABM99], and description logics with binders [Mar99]. We will establish such reductions in Chapter 6.

The square tiling problem is only one of many examples of the great variety of bounded tiling problems. Examples can be found in [Chl86, vEB97]. Since there are bounded tiling problems of many complexity levels, they have widely been used as a convenient tool for establishing lower complexity bounds for logics, see [Chl86], or [GKWZ03] and the references therein.

Revisiting the proof of Theorem 2.29 (2) in [SvEB84] shows that the proof technique, which translates Turing machine computations into tilings, is very robust. Hence, simple variants of the square tiling problem can analogously be shown to be complete for larger classes. We will consider the following variant, which we will call the 2^{2^n}*-tiling problem*. Given a finite set T of tile types and a string 1^n, is there a T-tiling of the $2^{2^n} \times 2^{2^n}$ square?

Corollary 2.30 *The* 2^{2^n}*-tiling problem is* N2EXPTIME*-complete.*

Chapter 3

Expressivity

In this thesis, we consider hybrid logics with temporal operators F, P, S, U; hybrid binders $\downarrow$, $\exists$; satisfaction operators $@_t$; and the "somewhere" modality E. In order to classify the complexity of decision problems for all languages constructed by subsets of these operators, it is essential to ask what languages there are and what their inclusion structure is. Hierarchies of hybrid languages have been established, for example, in [BS95], [Gor96], and [FdRS03]. Building on the expressivity results from those papers, we will establish the inclusion structure of all fragments of the full hybrid language that are relevant for this thesis, with respect to several frame classes.

3.1 Towards a hierarchy of hybrid languages

From a naïve point of view, arbitrary combinations of these eight operators should yield $2^8 = 256$ hybrid languages. Fortunately enough, certain operators can be simulated using others, which decreases the number of different languages dramatically and leads to inclusions between hybrid languages. Furthermore, even more languages coincide over restricted frame classes.

The informal statement "an operator X can be simulated using (one or more) other operators Y" will be stated more precisely later on and makes use of the following definition of the relation $\equiv$.

Definition 3.1 *Let φ_1, φ_2 be formulae from two hybrid languages $\mathcal{HL}(X_1)$ and $\mathcal{HL}(X_2)$, respectively. The relation $\varphi_1 \equiv \varphi_2$ holds if and only if for any model $\mathcal{M} = (M, R, V)$, any assignment g for $\mathcal{M}$, and any state $m \in M$:*

$$\mathcal{M}, g, m \Vdash \varphi \qquad \Leftrightarrow \qquad \mathcal{M}, g, m \Vdash \varphi'.$$

The following simulations between temporal, modal, and hybrid operators are well-known, see, for instance, [FdRS03].

Fact 3.2 *Let φ, ψ be hybrid formulae, $t \in \mathrm{NOM} \cup \mathrm{SVAR}$, and $x, a, b \in \mathrm{SVAR}$, where neither a nor b occurs free in φ or ψ. Then the following propositions hold.*

(1) $@_t\varphi \equiv \mathsf{E}(t \wedge \varphi)$

(2) $\downarrow x.\varphi \equiv \exists x.(x \wedge \varphi)$

(3) $\exists x.\varphi \equiv \downarrow a.\,\mathsf{E}\downarrow x.\,\mathsf{E}(a \wedge \varphi)$

(4) $\mathsf{E}\varphi \equiv \exists a.(@_a\varphi)$

(5) $\mathsf{F}\varphi \equiv \top\mathsf{U}\varphi$

(6) $\mathsf{P}\varphi \equiv \top\mathsf{S}\varphi$

(7) $\varphi\mathsf{U}\psi \equiv \downarrow a.\,\mathsf{F}\downarrow b.\big(\psi \wedge @_a\mathsf{G}(\mathsf{F}b \to \varphi)\big)$

(8) $\varphi\mathsf{U}\psi \equiv \downarrow a.\,\mathsf{F}\big(\psi \wedge \mathsf{H}(\mathsf{P}a \to \varphi)\big)$

(9) $\varphi\mathsf{S}\psi \equiv \downarrow a.\,\mathsf{P}\big(\psi \wedge \mathsf{G}(\mathsf{F}a \to \varphi)\big)$

(10) $\mathsf{P}\varphi \equiv \downarrow a.\,\mathsf{E}(\varphi \wedge \mathsf{F}a)$

(11) $\varphi\mathsf{U}\psi \equiv \exists a.\big(\mathsf{F}(a \wedge \psi) \wedge \mathsf{G}(\mathsf{F}a \to \varphi)\big)$

We first examine the inclusion structure of all hybrid languages with no temporal operators other than F. We consider all languages that contain F and arbitrary combinations of the $\downarrow$, $\exists$, $@$, and E operators. Although there are $2^4 = 16$ such combinations, Fact 3.2 (1)–(4) causes some of them to coincide, where the term "coincidence" is formalised as follows.

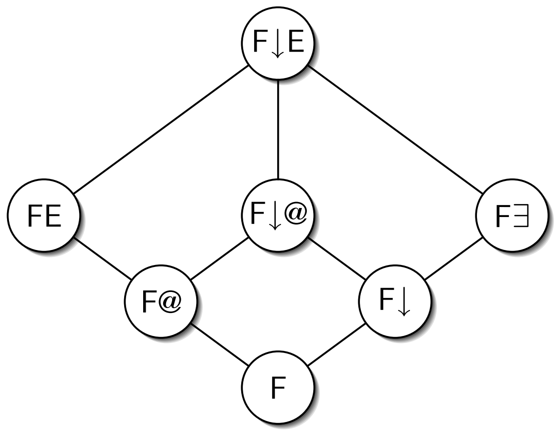
Definition 3.3 *Let L, L' be two hybrid languages. We write $L \sqsubseteq L'$ iff for each $\varphi \in L$, there is some $\varphi' \in L'$ such that $\varphi \equiv \varphi'$. If $L \sqsubseteq L'$ and $L' \sqsubseteq L$, we write $L \sim L'$.*

We have the following coincidences.

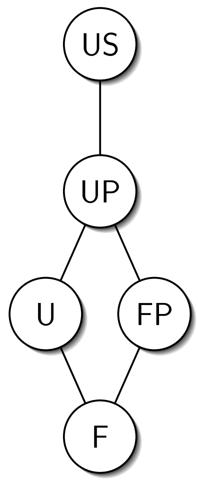
- $\mathcal{HL}(\mathsf{F},\downarrow,\exists,@,\mathsf{E}) \overset{(1),(3)}{\sim} \mathcal{HL}(\mathsf{F},\downarrow,\mathsf{E}) \overset{(3)}{\sim} \mathcal{HL}(\mathsf{F},\downarrow,\exists,\mathsf{E}) \overset{(1),(3)}{\sim} \mathcal{HL}(\mathsf{F},\downarrow,@,\mathsf{E})$
- $\mathcal{HL}(\mathsf{F},\downarrow,\mathsf{E}) \overset{(2),(3)}{\sim} \mathcal{HL}(\mathsf{F},\exists,\mathsf{E}) \overset{(1)}{\sim} \mathcal{HL}(\mathsf{F},\exists,@,\mathsf{E})$
- $\mathcal{HL}(\mathsf{F},\exists,@) \overset{(1),(3)}{\sqsubseteq} \mathcal{HL}(\mathsf{F},\downarrow,\mathsf{E}) \overset{(4)}{\sqsubseteq} \mathcal{HL}(\mathsf{F},\downarrow,\exists,@) \overset{(2)}{\sqsubseteq} \mathcal{HL}(\mathsf{F},\exists,@)$
- $\mathcal{HL}(\mathsf{F},@,\mathsf{E}) \overset{(1)}{\sim} \mathcal{HL}(\mathsf{F},\mathsf{E})$
- $\mathcal{HL}(\mathsf{F},\downarrow,\exists) \overset{(2)}{\sim} \mathcal{HL}(\mathsf{F},\exists)$

Hence there are only seven relevant languages. Fact 3.2 (1)–(4) also causes inclusions between them, as shown in Figure 3.1 (a), where the abbreviation "$\mathcal{HL}$" as well as parentheses and commas have been omitted.

In addition to non-temporal operators, we will not consider all possible combinations of temporal operators. Instead, we will restrict our attention to the languages $\mathcal{HL}(\mathsf{F},X)$, $\mathcal{HL}(\mathsf{F},\mathsf{P},X)$, $\mathcal{HL}(\mathsf{U},X)$, $\mathcal{HL}(\mathsf{U},\mathsf{P},X)$, and $\mathcal{HL}(\mathsf{U},\mathsf{S},X)$, where X stands for any of the above combinations of the four non-temporal operators. The reason for this restriction is that all other languages coincide with, or are mirror images of, one of these languages, where "mirror image"



(a) Without temporal operators



(b) With only temporal operators

Figure 3.1: Parts of a hierarchy of hybrid languages

means that F is replaced by P, U is replaced by S, and vice versa. For instance, $\mathcal{HL}(\mathsf{P},\mathsf{U})$ is the mirror image of $\mathcal{HL}(\mathsf{F},\mathsf{S})$, which coincides with $\mathcal{HL}(\mathsf{F},\mathsf{P},\mathsf{S})$. The mentioned coincidences are due to Fact 3.2 (5)–(6), which is also responsible for the inclusion structure of the five languages as shown in Figure 3.1 (b).

If we now combine each temporal language with each non-temporal language, this should yield $5 \cdot 7 = 35$ combinations. Fortunately, due to Fact 3.2, some of them coincide.

- $\mathcal{HL}(\mathsf{F},\mathsf{P},\downarrow) \overset{(5)}{\sqsubseteq} \mathcal{HL}(\mathsf{U},\mathsf{P},\downarrow) \overset{(6)}{\sqsubseteq} \mathcal{HL}(\mathsf{U},\mathsf{S},\downarrow) \overset{(8),(9)}{\sqsubseteq} \mathcal{HL}(\mathsf{F},\mathsf{P},\downarrow)$
- $\mathcal{HL}(\mathsf{F},\exists) \overset{(5),(11)}{\sim} \mathcal{HL}(\mathsf{U},\exists)$
- $\mathcal{HL}(\mathsf{F},\mathsf{P},\exists) \overset{(5)}{\sqsubseteq} \mathcal{HL}(\mathsf{U},\mathsf{P},\exists) \overset{(6)}{\sqsubseteq} \mathcal{HL}(\mathsf{U},\mathsf{S},\exists) \overset{(8),(9),(2)}{\sqsubseteq} \mathcal{HL}(\mathsf{F},\mathsf{P},\exists)$
- $\mathcal{HL}(\mathsf{F},\downarrow,@) \overset{(5),(7)}{\sim} \mathcal{HL}(\mathsf{U},\downarrow,@)$
- $\mathcal{HL}(\mathsf{F},\mathsf{P},\downarrow,@) \overset{(5)}{\sqsubseteq} \mathcal{HL}(\mathsf{U},\mathsf{P},\downarrow,@) \overset{(6)}{\sqsubseteq} \mathcal{HL}(\mathsf{U},\mathsf{S},\downarrow,@) \overset{(8),(9)}{\sqsubseteq} \mathcal{HL}(\mathsf{F},\mathsf{P},\downarrow,@)$
- $\mathcal{HL}(\mathsf{F},\downarrow,\mathsf{E}) \overset{(5)}{\sqsubseteq} \mathcal{HL}(\mathsf{U},\downarrow,\mathsf{E}) \sqsubseteq \mathcal{HL}(\mathsf{U},\mathsf{P},\downarrow,\mathsf{E}) \overset{(6)}{\sqsubseteq} \mathcal{HL}(\mathsf{U},\mathsf{S},\downarrow,\mathsf{E})$
  $\overset{(8),(9)}{\sqsubseteq} \mathcal{HL}(\mathsf{F},\mathsf{P},\downarrow,\mathsf{E}) \overset{(10)}{\sqsubseteq} \mathcal{HL}(\mathsf{F},\downarrow,\mathsf{E})$

Hence there are 23 languages of interest, whose inclusion structure is given in Figure 3.2. The picture from Figure 3.1 (b) appears as a substructure in Figure 3.2 several times. It is distorted there, which will allow for plotting the borders between complexity results more clearly in Section 5.2.

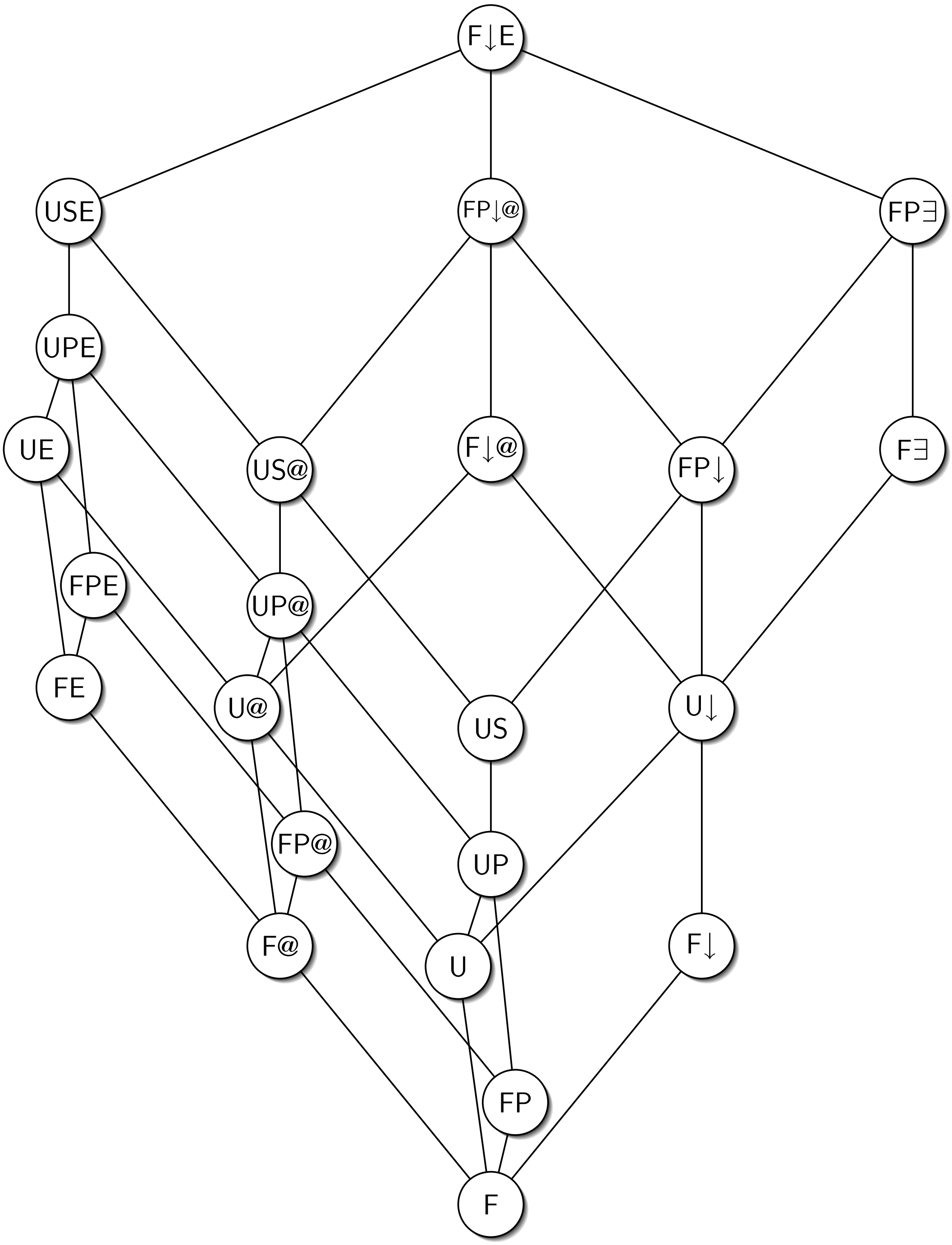


Figure 3.2: A hierarchy of hybrid languages over arbitrary frames

3.2 Hierarchies over restricted frame classes

Over transitive trees or linear frames, more languages coincide. This is so for several reasons. First, it is easily observed that $\downarrow$ is useless over any class of acyclic frames, as long as it is accompanied only by F or U. This is because it is not possible to reach a state to which some state variable has been bound, once this state has been left. Hence every formula φ from $\mathcal{HL}(\mathsf{F},\downarrow)$ (or $\mathcal{HL}(\mathsf{U},\downarrow)$) is $\equiv$-equivalent to φ' from $\mathcal{HL}(\mathsf{F})$ (or $\mathcal{HL}(\mathsf{U})$, respectively), where φ' is obtained from φ as follows [FdRS03].

- Replace each occurrence of any state variable x in the scope of some F (or U) operator which is in the scope of "$\downarrow x.$" by $\bot$.
- Replace each other occurrence of any state variable x by $\top$.
- Remove each occurrence of "$\downarrow x.$", for any state variable x.

Second, over transitive trees, the E operator can be simulated using F and P: $\mathsf{E}\varphi \equiv \varphi \vee \mathsf{P}\varphi \vee \mathsf{PF}\varphi$ [ABM00]. Over linear frames, $\mathsf{E}\varphi \equiv \mathsf{P}\varphi \vee \varphi \vee \mathsf{F}\varphi$ suffices. This causes the following equalities.

- $\mathcal{HL}(\mathsf{F},\mathsf{P}) \sim \mathcal{HL}(\mathsf{F},\mathsf{P},@) \sim \mathcal{HL}(\mathsf{F},\mathsf{P},\mathsf{E})$
- $\mathcal{HL}(\mathsf{U},\mathsf{P}) \sim \mathcal{HL}(\mathsf{U},\mathsf{P},@) \sim \mathcal{HL}(\mathsf{U},\mathsf{P},\mathsf{E})$
- $\mathcal{HL}(\mathsf{U},\mathsf{S}) \sim \mathcal{HL}(\mathsf{U},\mathsf{S},@) \sim \mathcal{HL}(\mathsf{U},\mathsf{S},\mathsf{E})$
- $\mathcal{HL}(\mathsf{F},\mathsf{P},\downarrow) \sim \mathcal{HL}(\mathsf{F},\downarrow,\mathsf{E}) \sim \mathcal{HL}(\mathsf{F},\mathsf{P},\downarrow,@) \sim \mathcal{HL}(\mathsf{F},\mathsf{P},\exists)$

In contrast to all previous ones, this simulation causes an exponential blowup. Therefore, for a classification of the complexity of decision problems, it is not helpful to merge entries in the hierarchy according to the above equalities. Instead, we have indicated these coincidences in Figure 3.3 by small halfcircles.

However, as for the satisfiability problem, it is helpful to know that, over these frame classes, the E operator can be simulated using F, P, and linearly many additional atomic propositions — provided that no binder is in the language. This simulation does not satisfy the relation $\equiv$, but preserves satisfiability and is computable in polynomial time. The details are given in [ABM00].

Over ER frames, temporal operators other than F do not add any expressive power. It is obvious that P and S are needless — they can be replaced by F and U because the accessibility relation is symmetric. Furthermore, $\varphi\mathsf{U}\psi \equiv \mathsf{F}\psi \wedge \mathsf{G}\varphi$. For these reasons, the languages over ER frames and their inclusion structure are the same as given in Figure 3.1 (a). We will often use $\Diamond$ instead of F in the absence of other temporal operators.

If only complete frames are considered, then $\mathsf{E}\varphi \equiv \mathsf{F}\varphi$. Hence, there are only the languages $\mathcal{HL}(\mathsf{F})$ and $\mathcal{HL}(\mathsf{F},\downarrow)$ with the inclusion $\mathcal{HL}(\mathsf{F}) \sqsubseteq \mathcal{HL}(\mathsf{F},\downarrow)$.

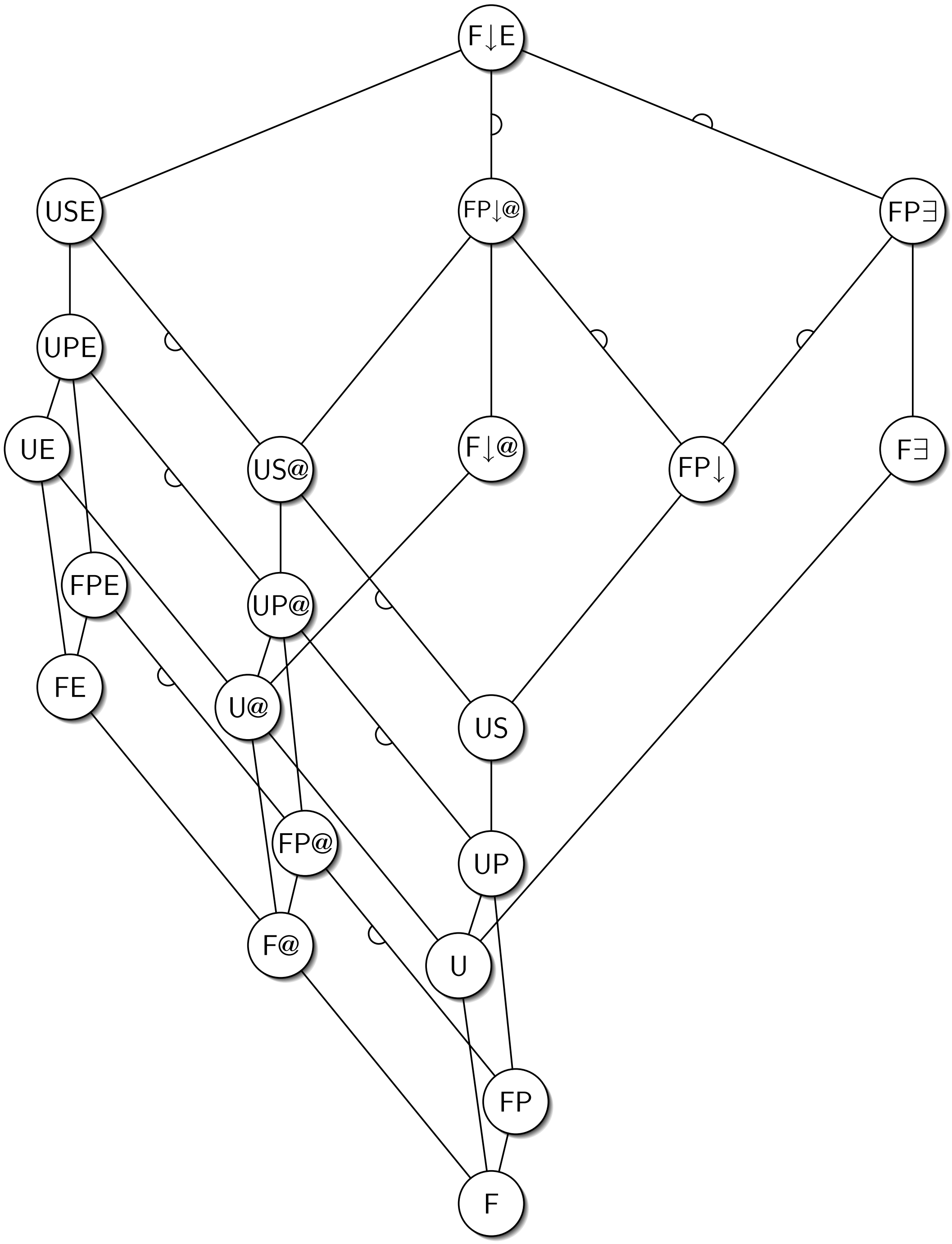


Figure 3.3: A hierarchy of hybrid languages over transitive trees and linear frames

Chapter 4

Model Checking

In this chapter, we classify the computational complexity of the model-checking problem of all fragments of the full hybrid language with respect to different frame classes. This problem has been exhaustively examined over arbitrary frames by Franceschet and de Rijke in [FdR06]. Their results can be summarised as follows. For binder-free languages, model checking is solvable in polynomial time, and for languages with binders, it is PSPACE-complete. Since the upper bounds of these problems carry over if we restrict the class of frames, there is only one interesting question left: Does model checking remain PSPACE-hard for binder languages over restricted frame classes? Our contribution is the answer "yes" for the frame classes we consider.

The hardness results by Franceschet and de Rijke hold for $\mathcal{PML}(\mathsf{F}, \downarrow)$-MC. With a slight modification of their proof technique, it is possible to establish the same lower bound over complete frames. If we consider acyclic frame classes, there are three minimal binder-free languages. Hence we have to establish the lower bounds for $\mathcal{PML}(\mathsf{F}, \mathsf{P}, \downarrow)$, $\mathcal{PML}(\mathsf{F}, \downarrow, @)$, and $\mathcal{PML}(\mathsf{F}, \exists)$ over $(\mathbb{N}, >)$.

4.1 Complete frames and above

Lemma 4.1 $\mathcal{PML}(\Diamond, \downarrow)$-compl-MC *is* PSPACE-*hard.*

Proof. We will give a polynomial-time reduction from QSAT, which is defined in Section 2.4.1. Consider an arbitrary instance $\alpha = \mathsf{Q}_1 x_1 \ldots \mathsf{Q}_n x_n.\beta$ of QSAT. Let $\mathcal{M} = (M, R, \emptyset)$ consist of two states that form an equivalence class, namely $M = \{1, 2\}$ and $R = M \times M$. Let g be an arbitrary assignment for $\mathcal{M}$. (The choice of g is irrelevant, since the reduction function will produce sentences only.)

The central idea is to bind a state variable t (denoting "true") to some state (say, 1) such that, when evaluating β, each variable x_i will be treated as true iff it is bound to 1. For this purpose, we define a translation function τ from unquantified and quantified Boolean formulae to formulae from $\mathcal{PML}(\Diamond, \downarrow)$ as follows.

$$\begin{aligned} \tau(\bot) &= \bot & \tau(x_i) &= \Diamond(x_i \wedge t) \\ \tau(\neg\gamma) &= \neg\tau(\gamma) & \tau(\exists x_i.\gamma) &= \Diamond{\downarrow} x_i.\tau(\gamma) \\ \tau(\gamma_1 \vee \gamma_2) &= \tau(\gamma_1) \vee \tau(\gamma_2) & \tau(\forall x_i.\gamma) &= \Box{\downarrow} x_i.\tau(\gamma) \end{aligned}$$

From α we construct a sentence $f(\alpha) = {\downarrow} t.\tau(\alpha)$, which ensures that t is bound to some state. Clearly, f is computable in polynomial time.

It remains to show that $\alpha \in$ QSAT if and only if $\mathcal{M}, g, 1 \Vdash f(\alpha)$. (Since the states 1 and 2 of $\mathcal{M}$ satisfy the same sentences, it is correct to fix the state 1.) This property is a consequence of the following claim.

Claim. For each assignment g for $\mathcal{M}$ with $g(t) = 1$ and each $\alpha \in$ QBF:

$$\alpha \in \text{QSAT} \qquad \text{if and only if} \qquad \mathcal{M}, g, 1 \Vdash \tau(\alpha)$$

Proof of Claim. We will proceed by induction on α.

For the base case, suppose $\alpha = \exists x.\beta$. The $\forall$ case is analogous. The line of proof is by the following chain of equivalent statements.

$$\begin{aligned} \alpha \in \text{QSAT} \quad &\overset{(1)}{\Leftrightarrow} \quad \beta \text{ is satisfiable} \\ &\overset{(2)}{\Leftrightarrow} \quad \beta \text{ is true under the assignment } \{x\} \text{ or } \{\neg x\} \\ &\overset{(3)}{\Leftrightarrow} \quad \mathcal{M}, g_1^x, 1 \Vdash \beta \ \text{ or } \ \mathcal{M}, g_2^x, 1 \Vdash \beta \\ &\overset{(4)}{\Leftrightarrow} \quad \mathcal{M}, g_1^x, 1 \Vdash \beta[x/\Diamond(x \wedge t)] \ \text{ or } \ \mathcal{M}, g_2^x, 2 \Vdash \beta[x/\Diamond(x \wedge t)] \\ &\overset{(5)}{\Leftrightarrow} \quad \mathcal{M}, g_1^x, 1 \Vdash \tau(\beta) \ \text{ or } \ \mathcal{M}, g_2^x, 2 \Vdash \tau(\beta) \\ &\overset{(6)}{\Leftrightarrow} \quad \mathcal{M}, g, 1 \Vdash {\downarrow} x.\tau(\beta) \ \text{ or } \ \mathcal{M}, g, 2 \Vdash {\downarrow} x.\tau(\beta) \\ &\overset{(7)}{\Leftrightarrow} \quad \mathcal{M}, g, 1 \Vdash \Diamond{\downarrow} x.\tau(\beta) \end{aligned}$$

These equivalences are justified as follows.

(1) Definition of QSAT.
(2) Definition of satisfiability for propositional formulae.
(3) Obvious.
(4) Since $\mathcal{M}$ is a complete model, the formula $\Diamond(x \wedge t)$ is true under g_1^x and false under g_2^x *everywhere* in $\mathcal{M}$.
(5) Since x is the only variable of β, $\tau(\beta) = \beta[x/\Diamond(x \wedge t)]$.
(6) Definition of satisfaction for $\downarrow$.
(7) Definition of satisfaction for $\Diamond$, together with the construction of $\mathcal{M}$.

For the induction step, suppose $\alpha = \exists y \mathrm{Q}_1 x_1 \dots \mathrm{Q}_n x_n.\beta$. Again, the $\forall$ case is analogous. We use the following chain of equivalent statements.

$$
\begin{aligned}
&\alpha \in \mathrm{QSAT} \\
&\overset{(1)}{\Leftrightarrow}\ \mathrm{Q}_1 x_1 \dots \mathrm{Q}_n x_n.\beta[y/\bot] \in \mathrm{QSAT} \text{ or } \mathrm{Q}_1 x_1 \dots \mathrm{Q}_n x_n.\beta[y/\top] \in \mathrm{QSAT} \\
&\overset{(2)}{\Leftrightarrow}\ \mathcal{M}, g, 1 \Vdash \tau(\mathrm{Q}_1 x_1 \dots \mathrm{Q}_n x_n.\beta[y/\bot]) \text{ or} \\
&\qquad \mathcal{M}, g, 1 \Vdash \tau(\mathrm{Q}_1 x_1 \dots \mathrm{Q}_n x_n.\beta[y/\top]) \\
&\overset{(3)}{\Leftrightarrow}\ \mathcal{M}, g, 1 \Vdash \tau(\mathrm{Q}_1 x_1 \dots \mathrm{Q}_n x_n.\beta)[\Diamond(y \wedge t)/\bot] \text{ or} \\
&\qquad \mathcal{M}, g, 1 \Vdash \tau(\mathrm{Q}_1 x_1 \dots \mathrm{Q}_n x_n.\beta)[\Diamond(y \wedge t)/\top] \\
&\overset{(4)}{\Leftrightarrow}\ \mathcal{M}, g_1^y, 1 \Vdash \tau(\mathrm{Q}_1 x_1 \dots \mathrm{Q}_n x_n.\beta) \text{ or } \mathcal{M}, g_2^y, 1 \Vdash \tau(\mathrm{Q}_1 x_1 \dots \mathrm{Q}_n x_n.\beta) \\
&\overset{(5)}{\Leftrightarrow}\ \mathcal{M}, g_1^y, 1 \Vdash \tau(\mathrm{Q}_1 x_1 \dots \mathrm{Q}_n x_n.\beta) \text{ or } \mathcal{M}, g_2^y, \mathbf{2} \Vdash \tau(\mathrm{Q}_1 x_1 \dots \mathrm{Q}_n x_n.\beta) \\
&\overset{(6)}{\Leftrightarrow}\ \mathcal{M}, g, 1 \Vdash {\downarrow} y.\tau(\mathrm{Q}_1 x_1 \dots \mathrm{Q}_n x_n.\beta) \text{ or } \mathcal{M}, g, 2 \Vdash {\downarrow} y.\tau(\mathrm{Q}_1 x_1 \dots \mathrm{Q}_n x_n.\beta) \\
&\overset{(7)}{\Leftrightarrow}\ \mathcal{M}, g, 1 \Vdash \Diamond {\downarrow} y.\tau(\mathrm{Q}_1 x_1 \dots \mathrm{Q}_n x_n.\beta)
\end{aligned}
$$

These equivalences are justified as follows.

(1) Definition of QSAT.

(2) Induction hypothesis.

(3) Construction of τ.

(4) Since $\mathcal{M}$ is a complete model, the formula $\Diamond(y \wedge t)$ is true under g_1^y and false under g_2^y *everywhere* in $\mathcal{M}$.

(5) Since $\tau(\dots)$ is a $\Diamond$- or $\Box$-formula and $\mathcal{M}$ is complete, $\tau(\dots)$ is true at 1 iff it is true at 2.

(6) Definition of satisfaction for $\downarrow$.

(7) Definition of satisfaction for $\Diamond$, together with the construction of $\mathcal{M}$.

❑

The following theorem is a consequence of Lemma 4.1 and [FdR06, Theo. 4.5].

Theorem 4.2 *Let* $X \in \{\{\Diamond,\downarrow\}, \{\Diamond,\downarrow,@\}, \{\Diamond,\exists\}, \{\Diamond,\downarrow,\mathsf{E}\}\}$ *and* $\mathfrak{F}$ *be a class of frames with* $\mathsf{compl} \subseteq \mathfrak{F}$.
Then $(\mathcal{P})\mathcal{ML}(X)$-$\mathfrak{F}$-MC *and* $(\mathcal{P})\mathcal{HL}(X)$-$\mathfrak{F}$-MC *are* PSPACE-*complete.*

Corollary 4.3

(1) Let $X \in \{\{\Diamond,\downarrow\}, \{\Diamond,\downarrow,@\}, \{\Diamond,\exists\}, \{\Diamond,\downarrow,\mathsf{E}\}\}$.
$(\mathcal{P})\mathcal{ML}(X)$-compl-MC *and* $(\mathcal{P})\mathcal{HL}(X)$-compl-MC *are* PSPACE-*complete.*

(2) Let $X \in \{\{\mathsf{F},\downarrow\}, \{\mathsf{U},\downarrow\}, \{\mathsf{F},\mathsf{P},\downarrow\}, \{\mathsf{F},\downarrow,@\}, \{\mathsf{F},\mathsf{P},\downarrow,@\}, \{\mathsf{F},\exists\}, \{\mathsf{F},\mathsf{P},\exists\}, \{\mathsf{F},\downarrow,\mathsf{E}\}\}$.
$(\mathcal{P})\mathcal{ML}(X)$-trans-MC *and* $(\mathcal{P})\mathcal{HL}(X)$-trans-MC *are* PSPACE-*complete.*

4.2 Natural numbers and above

Lemma 4.4

(1) $\mathcal{PML}(\mathsf{F},\mathsf{P},\downarrow)$-$(\mathbb{N},>)$-MC *is* PSPACE-*hard.*

(2) $\mathcal{PML}(\mathsf{F},\downarrow,@)$-$(\mathbb{N},>)$-MC *is* PSPACE-*hard.*

(3) $\mathcal{PML}(\mathsf{F},\exists)$-$(\mathbb{N},>)$-MC *is* PSPACE-*hard.*

Proof. We will give polynomial-time reductions from QSAT, analogously to the proof of Lemma 4.1. Consider an arbitrary instance $\alpha = \mathsf{Q}_1 x_1 \ldots \mathsf{Q}_n x_n.\beta$ of QSAT. Let $\mathcal{M} = (\mathbb{N}, >, \emptyset)$ and g be an arbitrary assignment. Again, we will bind a state variable t ("true") to a state different from 0, such that, when evaluating β, each variable x_i will be treated as true iff it is bound to this state. The overall technique for all three languages will essentially be the same. We will only give the polynomial-time computable translation and reduction functions. The remaining technical details are analogous to the proof of Lemma 4.1, with only slight modifications according to the frame class and the set of operators available.

(1). The reduction function is given by $f(\psi) = \downarrow s.\mathsf{F}\downarrow t.\mathsf{P}\big(s \wedge \tau(\psi)\big)$, and the translation $\tau(\cdot)$ is defined as follows.

$$\begin{aligned} \tau(\bot) &= \bot & \tau(x_i) &= \mathsf{F}(x_i \wedge t) \\ \tau(\neg\beta) &= \neg\tau(\beta) & \tau(\exists x_i.\beta) &= \mathsf{F}\downarrow x_i.\mathsf{P}\big(s \wedge \tau(\beta)\big) \\ \tau(\beta_1 \vee \beta_2) &= \tau(\beta_1) \vee \tau(\beta_2) & \tau(\forall x_i.\beta) &= \mathsf{G}\downarrow x_i.\mathsf{P}\big(s \wedge \tau(\beta)\big) \end{aligned}$$

(2). This case can be treated exactly as case (1) if we replace each occurrence of $\mathsf{P}(s \wedge \vartheta)$ in the definitions of $\tau(\cdot)$ and $f(\psi)$ by $@_s\vartheta$.

(3). The reduction function is given by $f(\psi) = \exists t.\big(\mathsf{F}t \wedge \tau(\psi)\big)$, and the translation $\tau(\cdot)$ is defined as follows.

$$\begin{aligned} \tau(\bot) &= \bot & \tau(x_i) &= \mathsf{F}(x_i \wedge t) \\ \tau(\neg\beta) &= \neg\tau(\beta) & \tau(\exists x_i.\beta) &= \exists x_i.\big(\mathsf{F}x_i \wedge \tau(\beta)\big) \\ \tau(\beta_1 \vee \beta_2) &= \tau(\beta_1) \vee \tau(\beta_2) & \tau(\forall x_i.\beta) &= \forall x_i.\big(\mathsf{F}x_i \rightarrow \tau(\beta)\big) \end{aligned}$$

❑

The following theorem is a consequence of Lemma 4.4 and [FdR06, Theo. 4.5].

Theorem 4.5 *Let* $X \in \big\{\{\mathsf{F},\mathsf{P},\downarrow\},\ \{\mathsf{F},\downarrow,@\},\ \{\mathsf{F},\mathsf{P},\downarrow,@\},\ \{\mathsf{F},\exists\},\ \{\mathsf{F},\mathsf{P},\exists\},\ \{\mathsf{F},\downarrow,\mathsf{E}\}\big\}$ *and* $\mathfrak{F}$ *be a class of frames with* $(\mathbb{N},>) \in \mathfrak{F}$.
Then $(\mathcal{P})\mathcal{ML}(X)$-$\mathfrak{F}$-MC *and* $(\mathcal{P})\mathcal{HL}(X)$-$\mathfrak{F}$-MC *are* PSPACE-*complete.*

Chapter 5

Satisfiability

5.1 Introduction

In this chapter, we classify the computational complexity of the satisfiability problem of all fragments of the full hybrid language with respect to different frame classes. Our contribution mainly concerns all languages over ER frames as well as many binder and until/since languages over transitive frames and transitive trees.

It goes without saying that decision problems for richer logics (such as hybrid languages) require more resources than those for simpler ones (such as the basic modal language). Satisfiability for $\mathcal{ML}(\mathsf{F})$ and $\mathcal{ML}(\mathsf{F},\mathsf{P})$ over arbitrary as well as over transitive frames are PSPACE-complete [Lad77, Spa93b]. In contrast, $\mathcal{ML}(\mathsf{F},\mathsf{E})$-SAT is EXPTIME-complete [Spa93a]. Over more restricted frame classes, satisfiability for $\mathcal{ML}(\mathsf{F})$ and $\mathcal{ML}(\mathsf{F},\mathsf{P})$ is NP-complete [Lad77, ON80, SC85]. In contrast, the known part of the complexity spectrum of hybrid satisfiability reaches up to undecidability.

Many complexity results for hybrid logics have been established in [ABM99, ABM00]. It was proven in [ABM99] that $\mathcal{HL}(\mathsf{F},@)$-SAT is PSPACE-complete and $\mathcal{HL}(\mathsf{F},\mathsf{P})$-SAT is PSPACE-complete and remains so when @ or E is added. The same authors show that these problems have the same complexity (or drop to PSPACE-complete or NP-complete, respectively) if the class of frames is restricted to transitive frames (or transitive trees, or linear frames, respectively) [ABM00].

Undecidability results for hybrid languages containing the restricted binder $\downarrow$ originate from [BS95, Gor96]. The strongest such result — for the pure nominal-free fragment of $\mathcal{HL}(\mathsf{F},\downarrow)$ — is given in [ABM99].

In a recent paper [tCF05b], it has been demonstrated that the decidability of $\mathcal{HL}(\mathsf{F},\downarrow,@)$ over arbitrary frames can be regained under certain syntactic restrictions concerning the interaction of $\downarrow$ and the modal operator $\Box$. In the same paper, decidability has been recovered by restricting the frame class to frames of bounded width (i.e., frames where the number of successors of each state is bounded). Other semantic restrictions by means of temporally relevant frame classes have been shown to sustain decidability in the following contexts. Over transitive trees and linear orders, the $\downarrow$ operator on its own is useless [FdRS03]. The class of transitive frames and the class of complete frames are further frame classes over which the $\downarrow$ language is "tamed", as was shown in [MSSW05].

Since, apparently, we can "restore" decidability for $\mathcal{HL}(\mathsf{F},\downarrow)$ by restricting the frame class, it is natural to ask whether decidability persists if we enrich the language, which can be done in several ways. One possibility is to include other operators, such as $@$, P, or E. We will examine the complexity of satisfiability of the thus extended languages and prove, among other results, undecidability over transitive frames, nonelementary decidability over transitive trees, and decidability (completeness for NEXPTIME or N2EXPTIME, depending on the language) over ER frames.

Another enrichment is the multi-modal version of the $\downarrow$ language, which will be considered in Chapter 6. We will show undecidability even of $\mathcal{HL}_2(\Diamond,\downarrow)$ over a wide range of frame classes, of which the classes of transitive frames, transitive trees, linear frames, and ER frames are prominent examples.

5.2 A map of the results for satisfiability

The pictures in this section illustrate all results that will be cited and established in this chapter. Figures 5.1–5.6 show the complexity results for satisfiability of all fragments of the full hybrid language over arbitrary frames, transitive frames, transitive trees, linear frames, the natural numbers, ER frames, and complete frames.

In the pictures, each coloured region is labelled by a complexity class (in bold face type) and corresponds to completeness with respect to this class. The only exception to this rule is the label "nonelementarily decidable", which is in the sense of Definition 2.16 (2). Regions that are left white and have two labels point out that these two complexity bounds are not tight. Red coloured nodes mark results established in the following sections; the results for the remaining nodes will be cited.

The following sections of this chapter will systematically examine satisfiability for all languages, separated by frame classes and in the same order as in Figures 5.1–5.6. Note that there will not be a specific section on complete frames because the results for the two remaining languages $\mathcal{HL}(\mathsf{F})$ and $\mathcal{HL}(\mathsf{F},\downarrow)$ follow from Section 5.8 and [MSSW05], respectively.

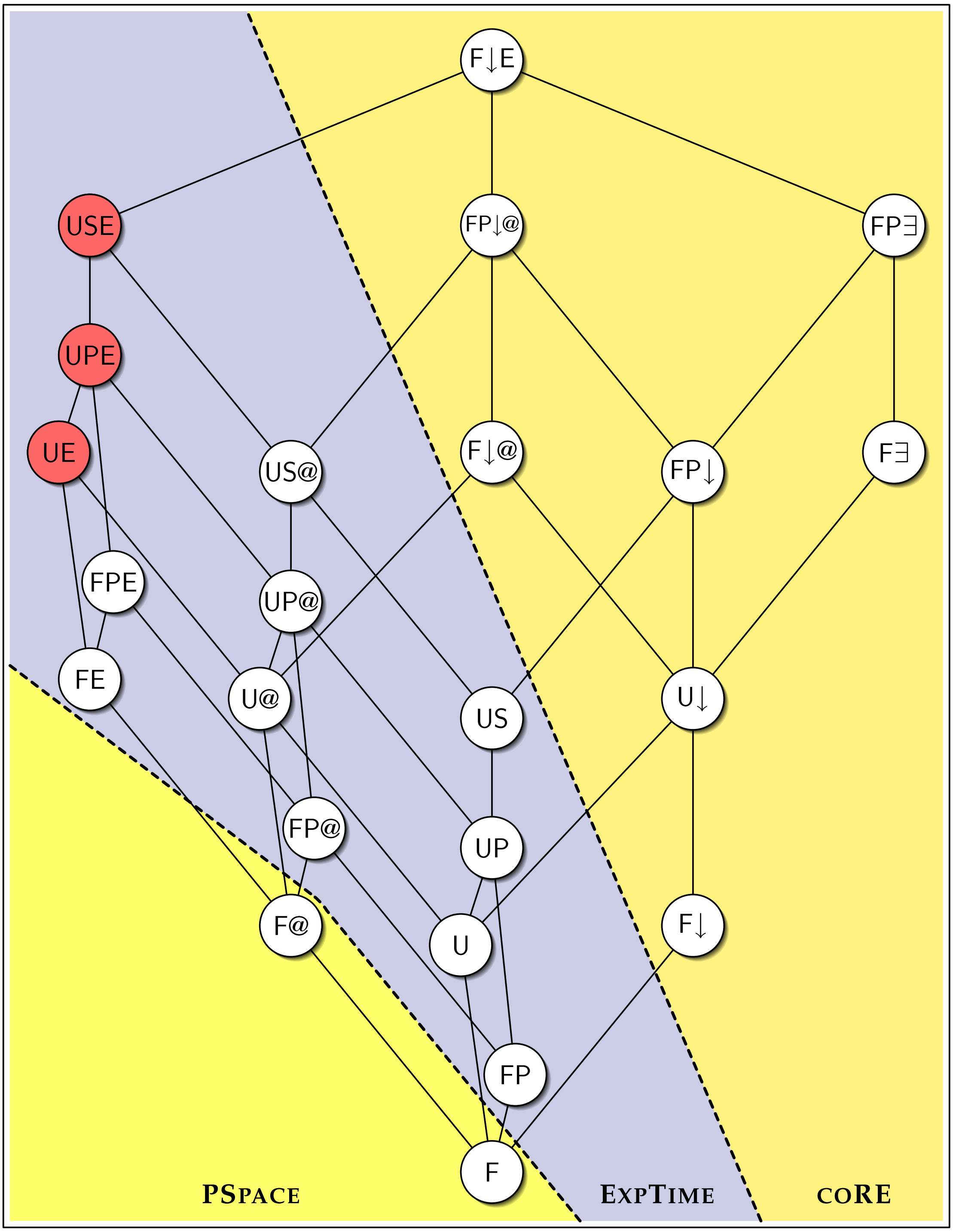


Figure 5.1: Complexity results for satisfiability over arbitrary frames

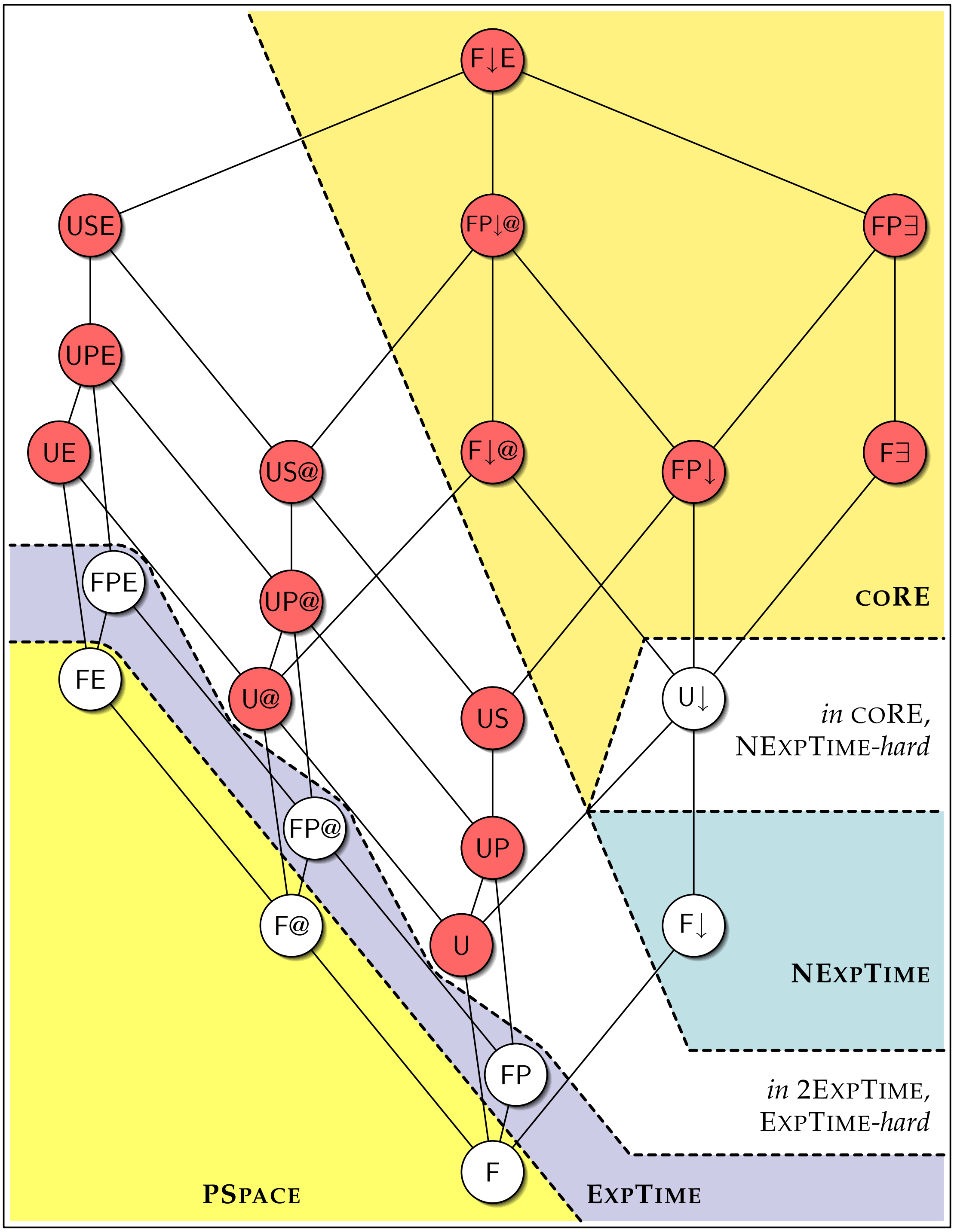


Figure 5.2: Complexity results for satisfiability over transitive frames

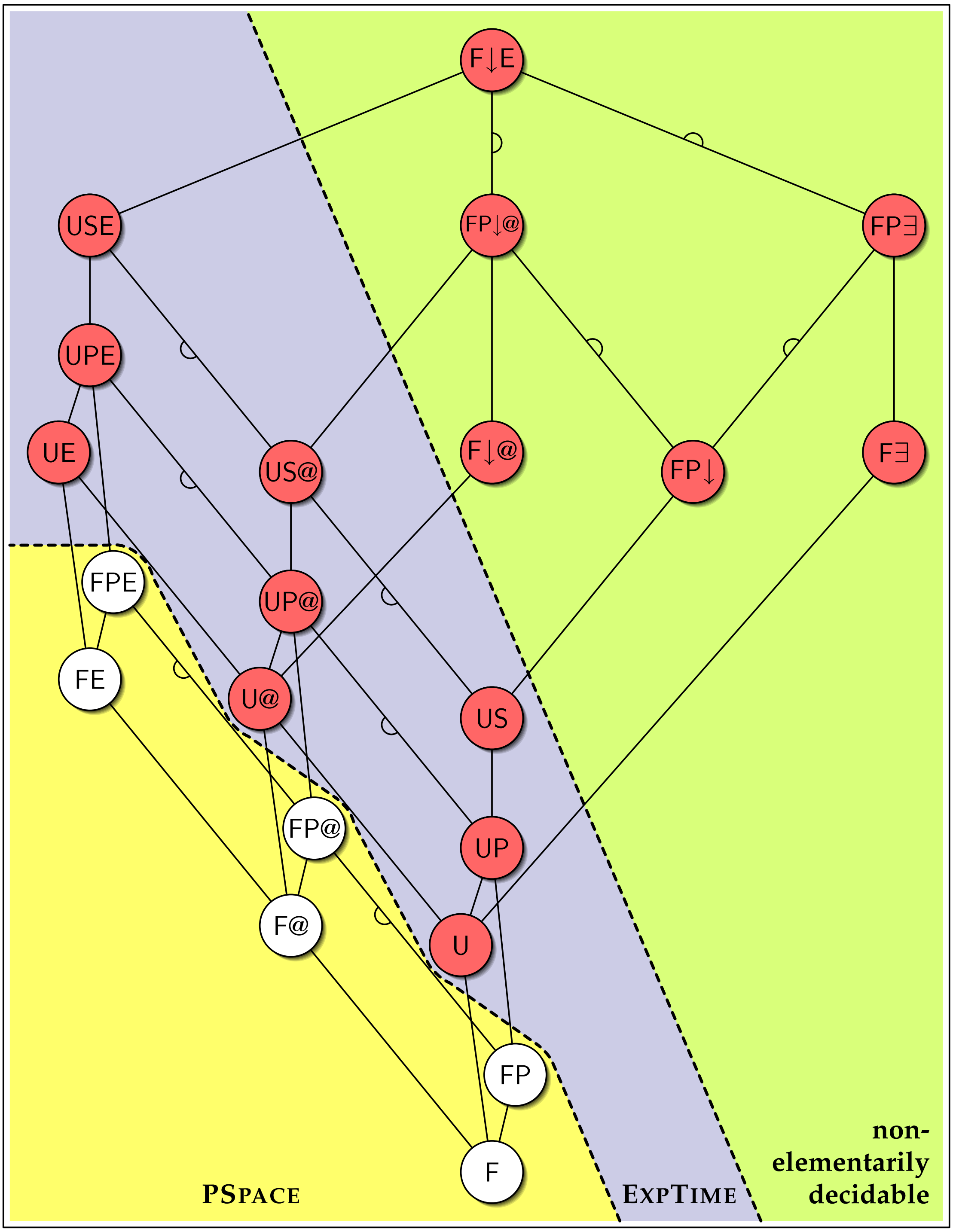


Figure 5.3: Complexity results for satisfiability over transitive trees

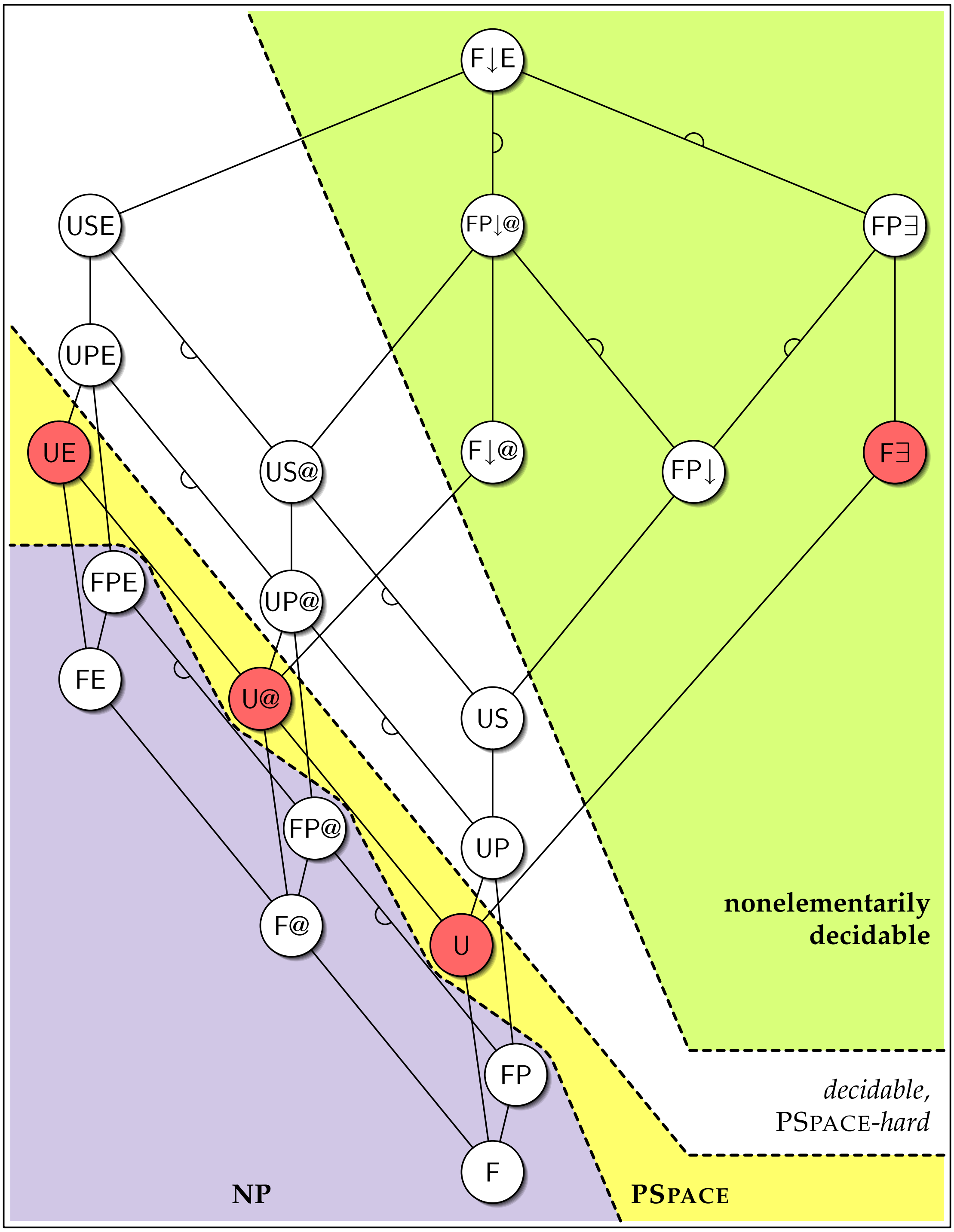


Figure 5.4: Complexity results for satisfiability over linear frames

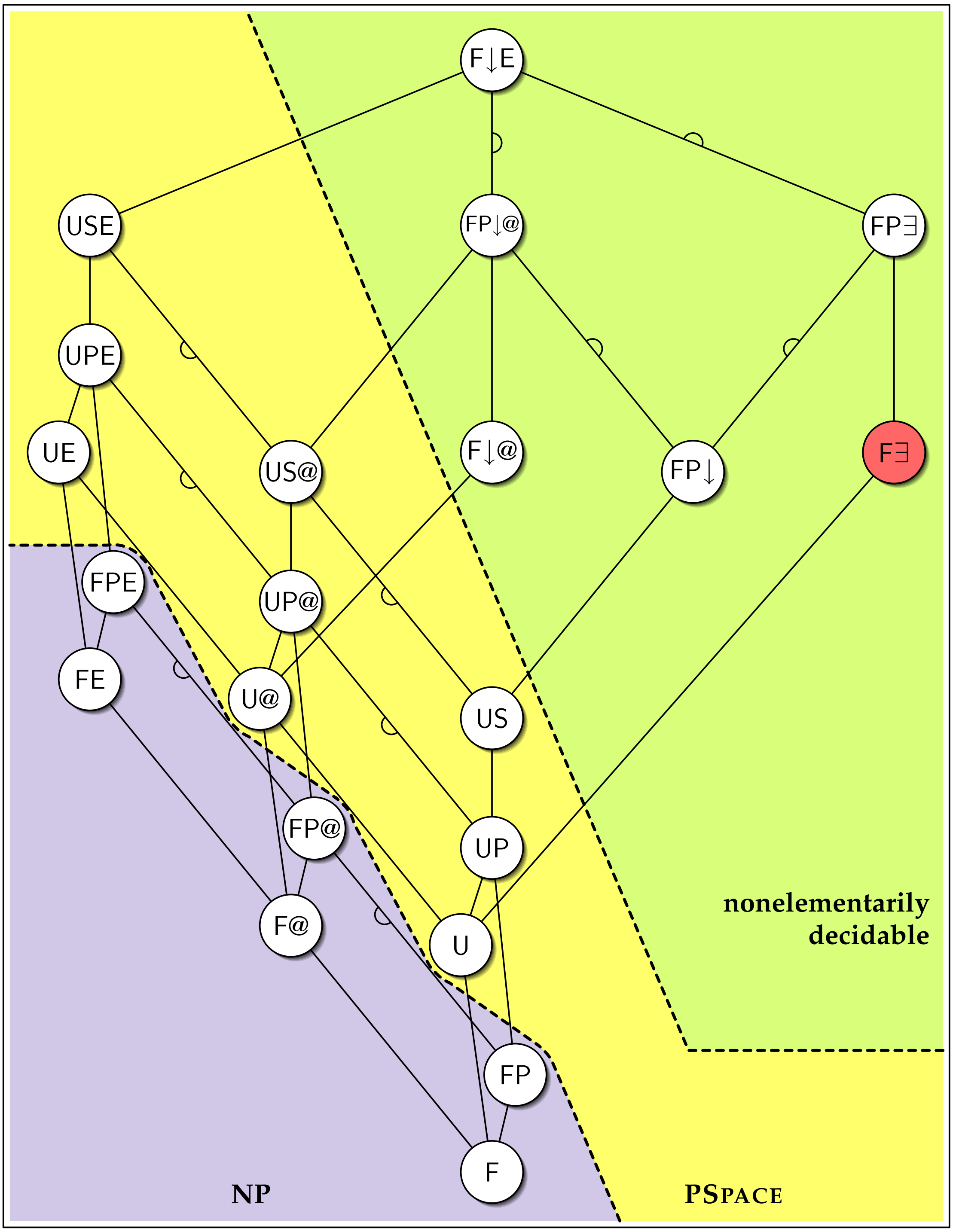
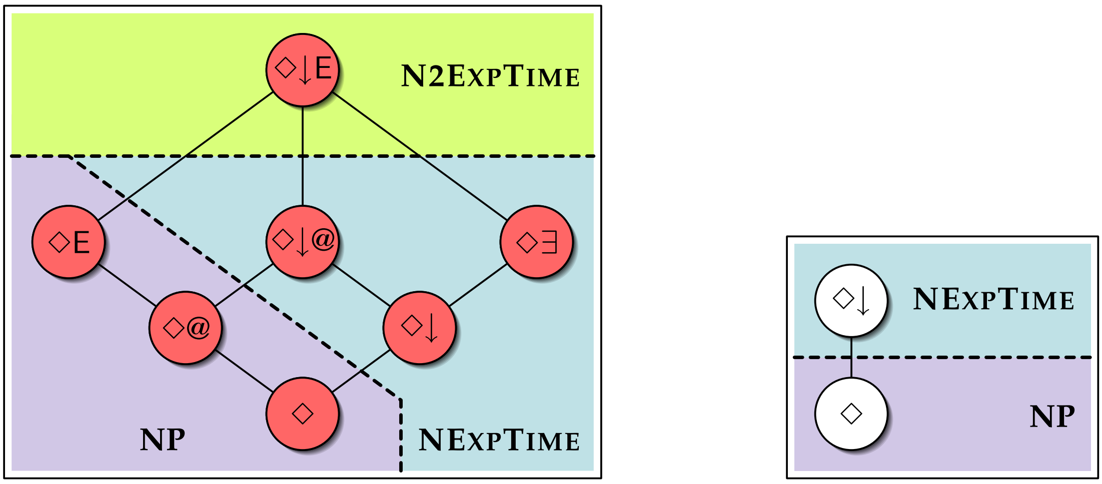


Figure 5.5: Complexity results for satisfiability over the natural numbers



(a) ER frames

(b) Complete frames

Figure 5.6: Complexity results for satisfiability over ER and complete frames

5.3 Arbitrary frames

Over arbitrary frames, the complexity of almost all fragments of the full hybrid language is known. The following theorem summarises all these cases.

Theorem 5.1

(1) $\mathcal{HL}(\mathsf{F})$-SAT *and* $\mathcal{HL}(\mathsf{F},@)$-SAT *are* PSPACE*-complete. [ABM99]*

(2) $\mathcal{HL}(\mathsf{F},\mathsf{P})$-SAT *and* $\mathcal{HL}(\mathsf{F},\mathsf{P},@)$-SAT *are* EXPTIME*-complete. [ABM99]*

(3) Let $X \in \{\{\mathsf{U}\}, \{\mathsf{U},@\}, \{\mathsf{U},\mathsf{P}\}, \{\mathsf{U},\mathsf{P},@\}, \{\mathsf{U},\mathsf{S}\}, \{\mathsf{U},\mathsf{S},@\}\}$.
Then $\mathcal{HL}(X)$-SAT *is* EXPTIME*-complete. [ABM00]*

(4) $\mathcal{HL}(\mathsf{F},\mathsf{E})$-SAT *and* $\mathcal{HL}(\mathsf{F},\mathsf{P},\mathsf{E})$-SAT *are* EXPTIME*-complete. [Spa93a, ABM00]*

(5) Let $X \in \{\{\mathsf{F},\downarrow\}, \{\mathsf{U},\downarrow\}, \{\mathsf{F},\mathsf{P},\downarrow\}, \{\mathsf{F},\exists\}, \{\mathsf{F},\mathsf{P},\exists\}, \{\mathsf{F},\downarrow,@\}, \{\mathsf{F},\mathsf{P},\downarrow,@\}, \{\mathsf{F},\downarrow,\mathsf{E}\}\}$.
Then $\mathcal{HL}(X)$-SAT *is* coRE*-complete. [ABM99]*

In the cases of Parts (2) and (3), EXPTIME-hardness — in the presence of one nominal only — has been shown in [ABM99] and [ABM00], respectively. For the corresponding upper bound, the authors refer to an embedding into the (loosely) guarded fragment with two or three variables, whose satisfiability problem is EXPTIME-complete [Grä99]. As pointed out by Balder ten Cate (personal communication), the canonical embedding via the Standard Translation does not map into this fragment in the presence of nominals. Since guarded fragments do not have constants, nominals have to be translated into existentially quantified variables. Hence the restriction in the number of variables is insufficient.

However, membership in EXPTIME follows from the fact that satisfiability for (loosely) guarded formulae with predicates of bounded arity is EXPTIME-complete as well [Grä99]. This fragment is appropriate for an embedding of $\mathcal{HL}(\mathsf{U},\mathsf{S},@)$ via the Standard Translation.

Regarding Part (4) of Theorem 5.1, the lower bound follows from EXPTIME-hardness of $\mathcal{ML}(\mathsf{F},\mathsf{E})$ [Spa93a], and the upper bound is due to [ABM00].

As for the remaining three languages with U and E, they cannot be straightforwardly embedded into any guarded fragment because the existential quantifier in the Standard Translation of the E operator is not guarded. However, we can establish EXPTIME-membership of their satisfiability problems via a reduction to $\mathcal{HL}(\mathsf{U},\mathsf{S},@)$-SAT using the spypoint technique [BS95, ABM99].

Theorem 5.2 *Let* $X \in \big\{\{\mathsf{U},\mathsf{E}\}, \{\mathsf{U},\mathsf{P},\mathsf{E}\}, \{\mathsf{U},\mathsf{S},\mathsf{E}\}\big\}$.
Then $\mathcal{HL}(X)$-SAT *is* EXPTIME-*complete.*

Proof. It suffices to show EXPTIME-membership of $\mathcal{HL}(\mathsf{U},\mathsf{S},\mathsf{E})$-SAT. In order to simulate the E operator using @, we use a spypoint. This is an additional state s named by the nominal i, from which all other states are accessible. The E operator can thus be replaced by a jump to s followed by an R-step, where R is the accessibility relation. The translations of the U and S operators have to ensure that only states that are visible from the spypoint are accessed.

More formally, we will define a translation function $(\cdot)^t : \mathcal{HL}(\mathsf{U},\mathsf{S},\mathsf{E}) \to \mathcal{HL}(\mathsf{U},\mathsf{S},@)$ inductively. It preserves atoms and Boolean operators. The operators U, S, and E are translated as follows (where $\mathsf{F}\varphi$ abbreviates $\top\mathsf{U}\varphi$).

$$\begin{aligned}(\varphi\mathsf{U}\psi)^t &= \varphi^t\mathsf{U}(\mathsf{P}i \wedge \psi^t)\\ (\varphi\mathsf{S}\psi)^t &= \varphi^t\mathsf{S}(\mathsf{P}i \wedge \psi^t)\\ (\mathsf{E}\varphi)^t &= @_i\mathsf{F}\varphi^t\end{aligned}$$

Now, a polynomial-time computable reduction function $f : \mathcal{HL}(\mathsf{U},\mathsf{S},\mathsf{E}) \to \mathcal{HL}(\mathsf{U},\mathsf{S},@)$ is

$$f(\varphi) = i \wedge \mathsf{GGP}i \wedge \mathsf{GH}(\neg i \to \mathsf{P}i) \wedge \mathsf{G}\neg\mathsf{F}i \wedge \mathsf{F}\varphi^t. \tag{5.1}$$

The first conjunct ensures the existence of the spypoint. Due to the second and third conjunct, for every state m accessible from s, every other state connected to s (via arbitrarily many, possibly reverse, R-edges) is accessible from s, too. The fourth conjunct ensures that s and every state accessible from s does not see s. Finally, the last conjunct enforces the existence of a state reachable from s in which φ^t is true. Clearly, f is computable in polynomial time. It remains to prove that for each formula $\varphi \in \mathcal{HL}(\mathsf{U},\mathsf{S},\mathsf{E})$,

$$\varphi \in \mathcal{HL}(\mathsf{U},\mathsf{S},\mathsf{E})\text{-}\textsc{Sat} \qquad \Leftrightarrow \qquad f(\varphi) \in \mathcal{HL}(\mathsf{U},\mathsf{S},@)\text{-}\textsc{Sat}.$$

"$\Rightarrow$". Suppose $\varphi \in \mathcal{HL}(\mathsf{U},\mathsf{S},\mathsf{E})$-SAT. Then there exist a model $\mathcal{M} = (M,R,V)$ and a state $m_0 \in M$, such that $\mathcal{M}, m_0 \Vdash \varphi$. From $\mathcal{M}$ we construct a model $\mathcal{M}^{\triangleright} = (M^{\triangleright}, R^{\triangleright}, V^{\triangleright})$, such that $M^{\triangleright} = M \uplus \{s\}$,[1] $R^{\triangleright} = R \cup \{(s,m) \mid m \in M\}$, and $V^{\triangleright} = V \cup \{(i,\{s\})\}$. It is immediately clear that $\mathcal{M}^{\triangleright}, s$ satisfies all conjuncts of $f(\varphi)$, except for the last one. Hence, it remains to show $\mathcal{M}^{\triangleright}, s \Vdash \mathsf{F}\varphi^t$, which is a consequence of the following claim and the fact that $sR^{\triangleright}m_0$.

Claim. For each subformula ψ of φ, each $m \in M$: $\mathcal{M}, m \Vdash \psi \Leftrightarrow \mathcal{M}^{\triangleright}, m \Vdash \psi^t$.

Proof of Claim. We use induction on ψ with the Boolean cases being straightforward and the U case being analogous to the S case. The S and E cases are treated by the following chains of equivalent statements:

$$
\begin{aligned}
&\mathcal{M}, m \Vdash \alpha\mathsf{S}\beta\\
&\overset{(1)}{\Leftrightarrow}\ \exists n \in M\big(nRm \;\&\; \mathcal{M}, n \Vdash \beta \wedge \forall \ell \in M(nR\ell Rm \Rightarrow \mathcal{M}, \ell \Vdash \alpha)\big)\\
&\overset{(2)}{\Leftrightarrow}\ \exists n \in M\big(nRm \;\&\; \mathcal{M}^{\triangleright}, n \Vdash \beta^t \wedge \forall \ell \in M(nR\ell Rm \Rightarrow \mathcal{M}^{\triangleright}, \ell \Vdash \alpha^t)\big)\\
&\overset{(3)}{\Leftrightarrow}\ \exists n \in M^{\triangleright}\big(sR^{\triangleright}n \;\&\; nR^{\triangleright}m \;\&\; \mathcal{M}^{\triangleright}, n \Vdash \beta^t \wedge \forall \ell \in M(nR\ell Rm \Rightarrow \mathcal{M}^{\triangleright}, \ell \Vdash \alpha^t)\big)\\
&\overset{(4)}{\Leftrightarrow}\ \exists n \in M^{\triangleright}\big(sR^{\triangleright}nR^{\triangleright}m \;\&\; \mathcal{M}^{\triangleright}, n \Vdash \beta^t \wedge \forall \ell \in M^{\triangleright}(nR^{\triangleright}\ell R^{\triangleright}m \Rightarrow \mathcal{M}^{\triangleright}, \ell \Vdash \alpha^t)\big)\\
&\overset{(5)}{\Leftrightarrow}\ \mathcal{M}^{\triangleright}, m \Vdash \alpha^t\mathsf{S}(\mathsf{P}i \wedge \beta^t)
\end{aligned}
$$

These equivalences are justified as follows.

(1) Definition of satisfaction for the S operator.
(2) Induction hypothesis.
(3) Construction of $M^{\triangleright}$ and $R^{\triangleright}$.
(4) Since $nR\ell$, but not nRs, we have that ($\ell \in M$ and $nR\ell Rm$) if and only if ($\ell \in M^{\triangleright}$ and $nR^{\triangleright}\ell R^{\triangleright}m$).
(5) Definition of satisfaction for the P, S, $\wedge$ operators, and for nominals.

$$
\begin{aligned}
&\mathcal{M}, m \Vdash \mathsf{E}\psi\\
&\quad\Leftrightarrow\ \exists n \in M(\mathcal{M}, n \Vdash \psi) && \text{(definition of satisfaction for E)}\\
&\quad\Leftrightarrow\ \exists n \in M(\mathcal{M}^{\triangleright}, n \Vdash \psi^t) && \text{(induction hypothesis)}\\
&\quad\Leftrightarrow\ \exists n \in M^{\triangleright}(sR^{\triangleright}n \;\&\; \mathcal{M}^{\triangleright}, n \Vdash \psi^t) && \text{(construction of } M^{\triangleright} \text{ and } R^{\triangleright}\text{)}\\
&\quad\Leftrightarrow\ \mathcal{M}^{\triangleright}, m \Vdash @_i\mathsf{F}\psi^t && \text{(definition of satisfaction for F and @)}
\end{aligned}
$$

[1] The symbol $\uplus$ denotes the disjoint union of two sets.

"$\Leftarrow$". Suppose $f(\varphi) \in \mathcal{HL}(\mathsf{U},\mathsf{S},@)$-SAT. Then there exist a model $\mathcal{M} = (M, R, V)$ and a state $s \in M$, such that $\mathcal{M}, s \Vdash \varphi$. Because of the first four conjuncts of $f(\varphi)$, s is named i, and all properties given after Equation (5.1) hold. The last conjunct enforces the existence of a state $m_0 \in M$ with $\mathcal{M}, m_0 \Vdash \varphi^t$.

From $\mathcal{M}$ we construct a model $\mathcal{M}^{\triangleleft} = (M^{\triangleleft}, R^{\triangleleft}, V^{\triangleleft})$, such that $M^{\triangleleft} = \{m \in M \mid sRm\}$, $R^{\triangleleft} = R\restriction_{M^{\triangleleft}}$, and $V^{\triangleleft} = V\restriction_{M^{\triangleleft}}$. It remains to show $\mathcal{M}^{\triangleleft}, m_0 \Vdash \varphi$, which is a consequence of the following claim.

Claim. For each subformula ψ of φ, each $m \in M^{\triangleleft}$: $\mathcal{M}, m \Vdash \psi^t \Leftrightarrow \mathcal{M}^{\triangleleft}, m \Vdash \psi$.

Proof of Claim. Again, we use induction on ψ and only demonstrate the S and E cases.

$$
\begin{aligned}
&\mathcal{M}, m \Vdash \alpha^t \mathsf{S}(\mathsf{P}i \wedge \beta^t)\\
\overset{(1)}{\Leftrightarrow}\ & \textcircled{\exists} n \in M\big(nRm \;\&\; sRn \;\&\; \mathcal{M}, n \Vdash \beta^t \wedge \textcircled{\forall} \ell \in M(nR\ell Rm \Rightarrow \mathcal{M}, \ell \Vdash \alpha^t)\big)\\
\overset{(2)}{\Leftrightarrow}\ & \textcircled{\exists} n \in M\big(nRm \;\&\; sRn \;\&\; \mathcal{M}, n \Vdash \beta^t \wedge \textcircled{\forall} \ell \in M^{\triangleleft}(nR^{\triangleleft}\ell R^{\triangleleft}m \Rightarrow \mathcal{M}, \ell \Vdash \alpha^t)\big)\\
\overset{(3)}{\Leftrightarrow}\ & \textcircled{\exists} n \in M^{\triangleleft}\big(nR^{\triangleleft}m \;\&\; \mathcal{M}, n \Vdash \beta^t \wedge \textcircled{\forall} \ell \in M^{\triangleleft}(nR^{\triangleleft}\ell R^{\triangleleft}m \Rightarrow \mathcal{M}, \ell \Vdash \alpha^t)\big)\\
\overset{(4)}{\Leftrightarrow}\ & \textcircled{\exists} n \in M^{\triangleleft}\big(nR^{\triangleleft}m \;\&\; \mathcal{M}^{\triangleleft}, n \Vdash \beta \wedge \textcircled{\forall} \ell \in M^{\triangleleft}(nR^{\triangleleft}\ell R^{\triangleleft}m \Rightarrow \mathcal{M}^{\triangleleft}, \ell \Vdash \alpha)\big)\\
\overset{(5)}{\Leftrightarrow}\ & \mathcal{M}^{\triangleleft}, m \Vdash \alpha \mathsf{S} \beta
\end{aligned}
$$

These equivalences are justified as follows.

(1) Definition of satisfaction for the P, S, $\wedge$ operators, and for nominals.
(2) Because of sRn and the conjunct $\mathsf{GGP}i$ of $f(\varphi)$, we have that ($\ell \in M$ and $nR\ell$) if and only if ($\ell \in M^{\triangleleft}$ and $nR^{\triangleleft}\ell$).
(3) Due to the construction of $M^{\triangleleft}$ and $R^{\triangleleft}$, we have that (sRn and nRm and $n \in M$) is equivalent to $nR^{\triangleleft}m$ and $n \in M^{\triangleleft}$).
(4) Induction hypothesis.
(5) Definition of satisfaction for the S operator.

$$
\begin{aligned}
&\mathcal{M}, m \Vdash @_i \mathsf{F}\psi^t\\
&\quad \Leftrightarrow\ \textcircled{\exists} n \in M(sRn \;\&\; \mathcal{M}, n \Vdash \psi^t) && \text{(definition of satisfaction for F and @)}\\
&\quad \Leftrightarrow\ \textcircled{\exists} n \in M^{\triangleleft}(\mathcal{M}, n \Vdash \psi^t) && \text{(construction of } M^{\triangleleft})\\
&\quad \Leftrightarrow\ \textcircled{\exists} n \in M^{\triangleleft}(\mathcal{M}^{\triangleleft}, n \Vdash \psi) && \text{(induction hypothesis)}\\
&\quad \Leftrightarrow\ \mathcal{M}^{\triangleleft}, m \Vdash \mathsf{E}\psi && \text{(definition of satisfaction for E)}
\end{aligned}
$$

❑

This completes the analysis of hybrid languages over arbitrary frames.

5.4 Transitive frames

Over transitive frames, there have been fewer languages for which the complexity of satisfiability has already been known. These are summarised in the following theorem.

Theorem 5.3

(1) Let $X \in \{\{\mathsf{F}\}, \{\mathsf{F}, @\}, \{\mathsf{F}, \mathsf{E}\}\}$.
Then $\mathcal{HL}(X)$-trans-SAT *is* PSPACE-*complete. [ABM00]*

(2) Let $X \in \{\{\mathsf{F}, \mathsf{P}\}, \{\mathsf{F}, \mathsf{P}, @\}, \{\mathsf{F}, \mathsf{P}, \mathsf{E}\}\}$.
Then $\mathcal{HL}(X)$-trans-SAT *is* EXPTIME-*complete, even in the presence of one nominal. [ABM00]*

(3) $\mathcal{HL}(\mathsf{F}, \downarrow)$-trans-SAT *is* NEXPTIME-*complete. [MSSW05]*

We examine the remaining cases in two groups.

5.4.1 Binder-free until/since languages

As for the lower bound, we establish a result as general as possible, namely EXPTIME-hardness of $\mathcal{ML}(\mathsf{U})$-trans-SAT and $\mathcal{ML}(\mathsf{U})$-tt-SAT.

Lemma 5.4 $\mathcal{ML}(\mathsf{U})$-trans-SAT *and* $\mathcal{ML}(\mathsf{U})$-tt-SAT *are* EXPTIME-*hard.*

Proof. We will reduce $\mathcal{ML}(\Diamond)$-GLOBSAT to both problems using the same reduction function. $\mathcal{ML}(\Diamond)$-GLOBSAT is EXPTIME-hard, which follows from the proof of EXPTIME-hardness for $\mathcal{ML}(\Diamond, \mathsf{E})$-SAT [Spa93a].

It may seem difficult to try to reduce this problem over *arbitrary* frames to our satisfiability problem over *transitive* frames. The critical point lies in making a non-transitive model transitive: taking the transitive closure of its relation would lead to new accessibilities that would disturb satisfaction of $\neg\Diamond$-formulae. Fortunately though, the U operator enables us to distinguish the accessibilities in the original model from those that have been added to make the relation transitive. Hence, a translation of $\Diamond\varphi$ should demand: "Make sure that the current state sees a state in which the translation of φ holds, and that there is no state in between." This translates as $\bot\mathsf{U}(\varphi^t)$ into the modal language.

In order to construct the required reduction, we define a translation function $(\cdot)^t : \mathcal{ML}(\Diamond) \to \mathcal{ML}(\mathsf{U})$ as follows.

$$\begin{aligned} \bot^t &= \bot & (\varphi \vee \psi)^t &= \varphi^t \vee \psi^t \\ p^t &= p, \qquad p \in \mathrm{PROP} & (\Diamond\varphi)^t &= \bot\mathsf{U}(\varphi^t) \\ (\neg\varphi)^t &= \neg(\varphi^t) \end{aligned}$$

Using $(\cdot)^t$, we construct a polynomial-time computable reduction function $f : \mathcal{ML}(\Diamond) \to \mathcal{ML}(\mathsf{U})$ via $f(\varphi) = \varphi^t \wedge \mathsf{G}\varphi^t$. (Note that G is expressible using U, see Fact 3.2 (5).) It is straightforward to prove the following two claims for each $\varphi \in \mathcal{ML}(\Diamond)$.

(1) If $\varphi \in \mathcal{ML}(\Diamond)$-GLOBSAT, then $f(\varphi) \in \mathcal{ML}(\mathsf{U})$-tt-SAT.

(2) If $f(\varphi) \in \mathcal{ML}(\mathsf{U})$-trans-SAT, then $\varphi \in \mathcal{ML}(\Diamond)$-GLOBSAT.

Since each transitive tree is a transitive model, (1) and (2) imply the claim of this theorem.

(1). Suppose φ is satisfied in all states of some Kripke model $\mathcal{M} = (M, R, V)$. By considering the submodel generated by some arbitrary state, we can assume w.l.o.g. that $\mathcal{M}$ has a root m_0.

Due to the *tree model property* [BdRV05] there exists a tree-like model (a model whose underlying frame is a tree) that satisfies φ at all states. Hence we can assume $\mathcal{M}$ itself to be tree-like. From this model, we construct $\mathcal{M}^{\triangleright} = (M, R^+, V)$, which is clearly a transitive tree.

Because of the tree-likeness of $\mathcal{M}$, we observe that for each pair $(m, n) \in R$, there exists no $\ell \in M$ *between* m and n in terms of R^+, that is, no ℓ such that $mR^+\ell$ and $\ell R^+ n$. By means of this observation, we show that for all states $m \in M$ and all formulae $\psi \in \mathcal{ML}(\Diamond)$: $\mathcal{M}, m \Vdash \psi$ iff $\mathcal{M}^{\triangleright}, m \Vdash \psi^t$. This claim implies that $\mathcal{M}^{\triangleright}, m_0 \Vdash \varphi^t \wedge \mathsf{G}\varphi^t$. It is proven by induction on the structure of ψ. The only interesting case is $\psi = \Diamond\vartheta$, and the necessary argument can be summarised as follows.

$$
\begin{aligned}
\mathcal{M}, m \Vdash \Diamond\vartheta \quad &\overset{(1)}{\Leftrightarrow} \quad \exists n \in M(mRn \;\&\; \mathcal{M}, n \Vdash \vartheta) \\
&\overset{(2)}{\Leftrightarrow} \quad \exists n \in M(mRn \;\&\; \mathcal{M}^{\triangleright}, n \Vdash \vartheta^t) \\
&\overset{(3)}{\Leftrightarrow} \quad \exists n \in M\big(mR^+n \;\&\; \mathcal{M}^{\triangleright}, n \Vdash \vartheta^t \;\&\; \neg\exists \ell \in M(mR^+\ell R^+ n)\big) \\
&\overset{(4)}{\Leftrightarrow} \quad \mathcal{M}^{\triangleright}, m \Vdash \bot\mathsf{U}(\vartheta^t)
\end{aligned}
$$

These equivalences are justified as follows.

(1) Definition of satisfaction for the $\Diamond$ operator.

(2) Induction hypothesis.

(3) See the above observation.

(4) Definition of satisfaction for $\bot$ and U.

(2). Let $\mathcal{M} = (M, R, V)$ be a transitive model and $m_0 \in M$ such that $\mathcal{M}, m_0 \Vdash f(\varphi)$. We restrict ourselves to the submodel generated by m_0. Hence all states of $\mathcal{M}$ are accessible from m_0.

Define a new Kripke model $\mathcal{M}^{\triangleleft} = (M, R^{\triangleleft}, V)$ with $R^{\triangleleft} = \{(m,n) \in R \mid \neg \textcircled{\exists} \ell \in M(mR\ell Rn)\}$. We show that for all states $m \in M$ and all formulae $\psi \in \mathcal{ML}(\Diamond)$: $\mathcal{M}^{\triangleleft}, m \Vdash \psi$ iff $\mathcal{M}, m \Vdash \psi^t$. Again, we use induction on the structure of ψ with the only interesting case $\psi = \Diamond\vartheta$ and the following argument.

$$
\begin{aligned}
\mathcal{M}^{\triangleleft}, m \Vdash \Diamond\vartheta \quad &\overset{(1)}{\Leftrightarrow} \quad \textcircled{\exists} n \in M(mR^{\triangleleft}n \;\&\; \mathcal{M}^{\triangleleft}, n \Vdash \vartheta) \\
&\overset{(2)}{\Leftrightarrow} \quad \textcircled{\exists} n \in M\big(mRn \;\&\; \neg\textcircled{\exists}\ell(mR\ell Rn) \;\&\; \mathcal{M}^{\triangleleft}, n \Vdash \vartheta^t\big) \\
&\overset{(3)}{\Leftrightarrow} \quad \textcircled{\exists} n \in M\big(mRn \;\&\; \neg\textcircled{\exists}\ell(mR\ell Rn) \;\&\; \mathcal{M}, n \Vdash \vartheta^t\big) \\
&\overset{(4)}{\Leftrightarrow} \quad \mathcal{M}, m \Vdash \bot\mathsf{U}(\vartheta^t)
\end{aligned}
$$

These equivalences are justified as follows.

(1) Definition of satisfaction for the $\Diamond$ operator.
(2) Construction of $R^{\triangleleft}$.
(3) Induction hypothesis.
(4) Definition of satisfaction for $\bot$ and U.

Since $\mathcal{M}, m_0 \Vdash \varphi^t \wedge \mathsf{G}\varphi^t$, we conclude that for all states $m \in M$, $\mathcal{M}, m \Vdash \varphi^t$. The previous claim implies that $\mathcal{M}^{\triangleleft}$ satisfies φ at all states. ❑

In order to establish upper bounds for the languages with U (and S), we aim for an embedding into an appropriate fragment of first-order logic — if possible, the guarded fragment or an extension thereof [Grä99, GW99]. Since the usual first-order formula enforcing transitivity is not guarded, it is necessary to make a "detour" around transitivity by syntactic means, namely using the operators U^{++} and S^{++} defined in Section 2.1.3.

Lemma 5.5 *For every $X \subseteq \{@, \mathsf{E}\}$, the problems $\mathcal{HL}(\mathsf{U}, \mathsf{S}, X)$-trans-SAT and $\mathcal{HL}(\mathsf{U}^{++}, \mathsf{S}^{++}, X)$-SAT are polynomially reducible to each other.*

Proof. Either problem can be reduced to the other via a simple bijection $f : \mathcal{HL}(\mathsf{U}, \mathsf{S}, X) \to \mathcal{HL}(\mathsf{U}^{++}, \mathsf{S}^{++}, X)$ or its inverse, respectively. This function merely replaces every occurrence of U (or S, respectively) in the input formula by U^{++} (or S^{++}, respectively). Obviously, f and f^{-1} can be computed in polynomial time. It is straightforward to inductively verify the following two propositions.

(1) For every $\varphi \in \mathcal{HL}(\mathsf{U}, \mathsf{S}, X)$: If φ is satisfied in a state m of some transitive model $\mathcal{M}$, then $\mathcal{M}, m \Vdash f(\varphi)$.

(2) For all $\varphi \in \mathcal{HL}(\mathsf{U}^{++}, \mathsf{S}^{++}, X)$: If φ is satisfied in a state m of some model $\mathcal{M} = (M, R, V)$, then the transitive model $\mathcal{M}' = (M, R^+, V)$ satisfies $f^{-1}(\varphi)$ at m.

❑

Now we have reduced the original problems to the goal of embedding languages containing U^{++} (and S^{++}) over *arbitrary* frames into an appropriate fragment of FOL. Unfortunately, U^{++} and S^{++} rely on the transitive closure R^+ of the accessibility relation R. In order to express xR^+y, we have to use fixpoint operators, which lead us out of FOL. A natural fragment for our purpose is the loosely guarded fragment of first-order logic enhanced by fixpoint operators, μLGF. Its satisfiability problem is 2EXPTIME-complete in general and EXPTIME-complete if the number variables is bounded [GW99]. An embedding of our languages cannot satisfy the latter condition, as already observed in Section 5.3. Since there is currently no EXPTIME-completeness result for satisfiability of μLGF with predicates of bounded arity, our embedding leaves a gap between EXPTIME-hardness and 2EXPTIME-membership.

It is straightforward to embed even $\mathcal{HL}(\mathsf{U}^{++}, \mathsf{S}^{++}, @)$ into μLGF. The languages with the E operator require a more careful analysis that will be similar to that in Section 5.3.

Lemma 5.6 $\mathcal{HL}(\mathsf{U}, \mathsf{S}, @)$-trans-SAT *is in* 2EXPTIME.

Proof. For the embedding of $\mathcal{HL}(\mathsf{U}^{++}, \mathsf{S}^{++}, @)$ into μLGF, we recall from Section 2.2.2 that the Standard Translation ST consists of the rule

$$\mathrm{ST}_x(\varphi \mathsf{U}^{++} \psi) = \exists y. \big[xR^+y \wedge \mathrm{ST}_y(\psi) \wedge \forall z. \big((xR^+z \wedge zR^+y) \to \mathrm{ST}_z(\varphi) \big) \big]$$

for the U^{++} operator and an analogous rule for S^{++}. Since this expression is “almost” loosely guarded, it remains to take care of the R^+ expressions. But xR^+y can be expressed by

$$\big[\, \mathrm{LFP}\ W(x,y). \big(xRy \vee \exists z. (zRy \wedge xWz) \big) \big] xy,$$

yielding a μLGF-sentence with three variables. (If U^{++} operators are nested, variables can be “recycled”.) The constants from the translations of nominals can be eliminated introducing new, existentially quantified variables as usual. The whole translation only requires time polynomial in the length of the input formula. ❑

Lemma 5.7

(1) $\mathcal{HL}(\mathsf{U},\mathsf{E})$-trans-SAT $\leqslant^{\mathrm{P}}_{\mathrm{m}}$ $\mathcal{HL}(\mathsf{U},@)$-trans-SAT.

(2) $\mathcal{HL}(\mathsf{U},\mathsf{P},\mathsf{E})$-trans-SAT $\leqslant^{\mathrm{P}}_{\mathrm{m}}$ $\mathcal{HL}(\mathsf{U},\mathsf{P})$-trans-SAT.

(3) $\mathcal{HL}(\mathsf{U},\mathsf{S},\mathsf{E})$-trans-SAT $\leqslant^{\mathrm{P}}_{\mathrm{m}}$ $\mathcal{HL}(\mathsf{U},\mathsf{S})$-trans-SAT.

Proof. We proceed similarly to the proof of Theorem 5.2, invoking a spypoint argument. We only give the changes here; the technical details are analogous.

(1) Here we use the same translation function $(\cdot)^t$ (without the S case, of course) and the reduction function $f(\varphi) = i \wedge \neg\mathsf{F}i \wedge \mathsf{F}\varphi^t$.

(2) Here we use the translation function $(\cdot)^t$ that preserves atoms, Boolean operators, and the U operator. The operators P and E are translated as follows.

$$(\mathsf{P}\varphi)^t = \mathsf{P}(\mathsf{P}i \wedge \varphi^t)$$
$$(\mathsf{E}\varphi)^t = \mathsf{P}(i \wedge \mathsf{F}\varphi^t)$$

The reduction function is $f(\varphi) = i \wedge \neg\mathsf{F}i \wedge \mathsf{GH}(\neg i \rightarrow \mathsf{P}i) \wedge \mathsf{F}\varphi^t$.

(3) We use the translation function from (2), minus the rule for P plus the rule $(\varphi\mathsf{S}\psi)^t = \varphi^t\mathsf{S}(\mathsf{P}i \wedge \psi^t)$, and the same reduction function.

❏

Lemmata 5.4–5.7 yield the following result.

Theorem 5.8

(1) Let $X \in \{\{\mathsf{U}\},\{\mathsf{U},\mathsf{P}\},\{\mathsf{U},\mathsf{S}\},\{\mathsf{U},@\},\{\mathsf{U},\mathsf{P},@\},\{\mathsf{U},\mathsf{S},@\},\{\mathsf{U},\mathsf{E}\},\{\mathsf{U},\mathsf{P},\mathsf{E}\},\{\mathsf{U},\mathsf{S},\mathsf{E}\}\}$.
Then $\mathcal{HL}(X)$-trans-SAT *is* EXPTIME*-hard and in* 2EXPTIME.

(2) With only a bounded number of nominals, all these problems are EXPTIME*-complete.*

It is not known whether in the case of a bounded number of variables, but an arbitrary number of constants, satisfiability for μLGF-sentences also decreases from 2EXPTIME to EXPTIME, as is the case for the fragment without the μ operator [tCF05a]. If there were a positive answer to this question, EXPTIME-completeness of all satisfiability problems from Theorem 5.8 (1) would follow.

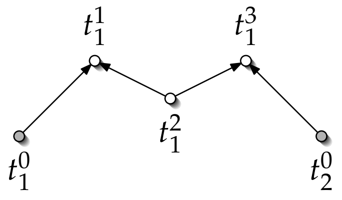


Figure 5.7: A zig-zag transition

5.4.2 Languages with binders

If we consider satisfiability over transitive frames, we cannot sustain decidability when enriching $\mathcal{HL}(\mathsf{F},\downarrow)$ with @ or P, or when proceeding to $\mathcal{HL}(\mathsf{F},\exists)$. We prove CORE-completeness for all languages strictly above $\mathcal{HL}(\mathsf{F},\downarrow)$ in our hierarchy — except for $\mathcal{HL}(\mathsf{U},\downarrow)$ —, making a detour via an undecidable fragment of first-order logic. The notation of such fragments is given in Section 2.2.2.

We proceed in two steps. First, we will show that $[\text{all},(4,1)]$-trans-SAT is undecidable. This will be accomplished by a reduction from $[\text{all},(0,1)]$-SAT. The undecidability of the latter is a consequence of the undecidability of contained traditional standard classes [BGG97]. The second step will consist of reductions from $[\text{all},(4,1)]$-trans-SAT to $\mathcal{HL}(\mathsf{F},\downarrow,@)$-trans-SAT; $\mathcal{HL}(\mathsf{F},\mathsf{P},\downarrow)$-trans-SAT; and $\mathcal{HL}(\mathsf{F},\exists)$-trans-SAT; respectively. To be more precise, the ranges of these reductions will be the fragments of the respective hybrid languages consisting of all nominal-free sentences.

Lemma 5.9 $[\text{all},(4,1)]$-trans-SAT *is* CORE*-complete.*

Proof. The upper bound follows from CORE-completeness of $\mathcal{FOL}$-SAT. In order to obtain the required reduction from $[\text{all},(0,1)]$-SAT, we will transform a (not necessarily transitive) model satisfying α into a transitive one. Simply taking the transitive closure adds new pairs to the interpretation of the relation in general and is not sufficient for keeping the information which pairs were in the "old" relation and which pairs were not. This problem does not arise if we instead use a variation of the *zig-zag technique* successfully applied in [ABM00] in a reduction from $\mathcal{ML}(\Diamond)$-GLOBSAT to $\mathcal{HL}(\mathsf{F},\mathsf{P})$-trans-SAT. The core idea of this technique is shown in Figure 5.7. The shown construction simulates an R-step $t_1 R t_2$ in the original model $\mathcal{M} = (D, I)$ by a zig-zag transition in a model $\mathcal{M}^{\triangleright} = (D^{\triangleright}, I^{\triangleright})$, where $I^{\triangleright}(R)$ is transitive.

We define a reduction function $f : [\text{all},(0,1)] \to [\text{all},(4,1)]$ using extra predicate symbols $0, 1, 2, 3$ as follows.

$$\begin{aligned}
f(\bot) &= \bot \\
f(xRy) &= \exists abc.\big(xRa \wedge bRa \wedge bRc \wedge yRc \wedge 0(x) \wedge 1(a) \wedge 2(b) \wedge 3(c) \wedge 0(y)\big) \\
f(\neg\alpha) &= \neg f(\alpha) \\
f(\alpha \vee \beta) &= f(\alpha) \vee f(\beta) \\
f(\exists x.\alpha) &= \exists x.\big(0(x) \wedge f(\alpha)\big)
\end{aligned}$$

The translation of the xRy-atoms exactly reflects the shown zig-zag transition. It is now straightforward to prove the following claim: *For each formula α, α is satisfiable iff $f(\alpha)$ is satisfiable in some model that interprets R by a transitive relation.*

Without loss of generality, we may assume that α has no free variables and that each variable is quantified exactly once. This can always be achieved by additional existential quantification and renaming, respectively.

"$\Rightarrow$". Suppose α is satisfied by some model $\mathcal{M} = (D, I)$. We construct a new model $\mathcal{M}^{\triangleright} = (D^{\triangleright}, I^{\triangleright})$, where $D^{\triangleright} = D^0 \cup \cdots \cup D^3$ with $D^i = \{d^i \mid d \in D\}$, for $i = 0, 1, 2, 3$. The interpretation $I^{\triangleright}$ is defined by

$$\begin{aligned}
&I^{\triangleright}(R) = \{(x^0, x^1),\ (x^2, x^1),\ (x^2, x^3),\ (y^0, x^3) \mid (x, y) \in I(R)\} \quad \text{and} \\
&I^{\triangleright}(P) = D^P, \quad P = 0, 1, 2, 3.
\end{aligned}$$

$I^{\triangleright}(R)$ codes an $I(R)$-transition from element x to y in the domain of $\mathcal{M}$ as a sequence of backward and forward transitions from x_0 to y_0 via x_1, x_2, x_3 as shown in Figure 5.7. It is easy to see that $I^{\triangleright}(R)$ is transitive, since there is no domain element with incoming *and* outgoing $I^{\triangleright}(R)$-edges.

We will now show that for all subformulae $\beta(x_1, \ldots, x_m)$ of α and for all $d_1, \ldots, d_m \in D$: $\mathcal{M} \models \beta\,[d_1, \ldots, d_m]$ iff $\mathcal{M}^{\triangleright} \models f(\beta)\,[d_1^0, \ldots, d_m^0]$. This immediately implies that $\mathcal{M}^{\triangleright}$ satisfies $f(\alpha)$. We will proceed by induction on β. The base case, $\beta = xRy$, is clear from the construction of $I^{\triangleright}(R)$. The Boolean cases are obvious. The case $\beta = \exists x.\gamma$ is treated by the following chain of equivalent statements.

$$\begin{aligned}
&\mathcal{M} \models\ \exists x.\gamma\,[d_1, \ldots, d_m] \\
&\qquad\qquad \overset{(1)}{\Leftrightarrow}\ \exists d \in D\ \big(\mathcal{M} \models \gamma\,[d_1, \ldots, d_m, x \mapsto d]\big) \\
&\qquad\qquad \overset{(2)}{\Leftrightarrow}\ \exists d \in D\ \big(\mathcal{M}^{\triangleright} \models f(\gamma)\,[d_1^0, \ldots, d_m^0, x \mapsto d^0]\big) \\
&\qquad\qquad \overset{(3)}{\Leftrightarrow}\ \exists d \in D^{\triangleright}\ \big(\mathcal{M}^{\triangleright} \models \big(0(x) \wedge f(\gamma)\big)\,[d_1^0, \ldots, d_m^0, x \mapsto d]\big) \\
&\qquad\qquad \overset{(4)}{\Leftrightarrow}\ \mathcal{M}^{\triangleright} \models \exists x.\big(0(x) \wedge f(\gamma)\big)\,[d_1^0, \ldots, d_m^0]
\end{aligned}$$

These equivalences are justified as follows.

(1) Definition of satisfaction for the $\exists$ quantifier.

(2) Induction hypothesis.

(3) Construction of $R^{\triangleright}$. ("$\Leftarrow$": Since x is interpreted by d' and $0(x)$ is satisfied, d is indeed some d^0.)

(4) Definition of satisfaction for $\wedge$ and $\exists$.

"$\Leftarrow$". Let $\mathcal{M} = (D, I)$ be a model satisfying $f(\alpha)$, where $I(R)$ is transitive. We construct a new model $\mathcal{M}^{\triangleleft} = (D^{\triangleleft}, I^{\triangleleft})$, where $D^{\triangleleft} = I(0)$ and

$$I^{\triangleleft}(R) = \big\{(d,e) \in (D^{\triangleleft})^2 \mid \exists abc \in D\big((d,a),(b,a),(b,c),(e,c) \in I(R) \\ \& \ a \in I(1) \ \& \ b \in I(2) \ \& \ c \in I(3)\big)\big\}.$$

We will now show that for all subformulae $\beta(x_1, \ldots, x_m)$ of α and for all $d_1, \ldots, d_m \in D$: $\mathcal{M}^{\triangleleft} \models \beta\,[d_1, \ldots, d_m]$ iff $\mathcal{M} \models f(\beta)\,[d_1, \ldots, d_m]$. This immediately implies that $\mathcal{M}^{\triangleleft}$ satisfies α. Again, the proof is via induction on β. The base case, $\beta = xRy$, is clear from the construction of $I^{\triangleleft}(R)$ and the fact that the translation of xRy requires $0(x)$ and $0(y)$. The Boolean cases are obvious. The case $\beta = \exists x.\gamma$ is treated by the following chain of equivalent statements.

$$\begin{aligned}
&\mathcal{M}^{\triangleleft} \models \exists x.\gamma\,[d_1, \ldots, d_m] \\
&\quad \overset{(1)}{\Leftrightarrow} \exists d \in D^{\triangleleft}\,\big(\mathcal{M}^{\triangleleft} \models \gamma\,[d_1, \ldots, d_m, x \mapsto d]\big) \\
&\quad \overset{(2)}{\Leftrightarrow} \exists d \in I(0)\,\big(\mathcal{M} \models f(\gamma)\,[d_1, \ldots, d_m, x \mapsto d]\big) \\
&\quad \overset{(3)}{\Leftrightarrow} \exists d \in D\,\big(\mathcal{M} \models \big(0(x) \wedge f(\gamma)\big)\,[d_1, \ldots, d_m, x \mapsto d]\big) \\
&\quad \overset{(4)}{\Leftrightarrow} \mathcal{M} \models \exists x.\big(0(x) \wedge f(\gamma)\big)\,[d_1, \ldots, d_m]
\end{aligned}$$

These equivalences are justified as follows.

(1) Definition of satisfaction for the $\exists$ quantifier.

(2) Induction hypothesis.

(3) "$\Rightarrow$": Obvious.
"$\Leftarrow$": Since x is interpreted by d and $0(x)$ is satisfied, $d \in I(0)$.

(4) Definition of satisfaction for $\wedge$ and $\exists$.

This proves the above claim. Since f is a polynomial-time computable reduction function, we have established undecidability for $[\text{all}, (4,1)]$-trans-SAT. ❑

Lemma 5.10

(1) $\mathcal{HL}(\mathsf{F}, \downarrow, @)$-trans-SAT *is* CORE-*hard.*

(2) $\mathcal{HL}(\mathsf{F}, \mathsf{P}, \downarrow)$-trans-SAT *is* CORE-*hard.*

(3) $\mathcal{HL}(\mathsf{F}, \exists)$-trans-SAT *is* CORE-*hard.*

Proof. We reduce from [all, (4, 1)]-trans-SAT, invoking the spypoint technique (see also proofs of Theorem 5.2 and Lemma 5.7). Since our reduction will not make use of any nominals other than the nominal i denoting the spypoint s, our undecidability result will even hold for the nominal-free fragments of the hybrid languages in question. We will simply treat i as a state variable and bind it to s.

(1). We define a translation function $(\cdot)^t$ from [all, (4, 1)] to $\mathcal{HL}(\mathsf{F}, \downarrow, @)$ as follows.

$$\begin{aligned} \bot^t &= \bot & (\neg\alpha)^t &= \neg(\alpha^t) \\ (xRy)^t &= @_x \mathsf{F} y & (\alpha \vee \beta)^t &= \alpha^t \vee \beta^t \\ \big(P(x)\big)^t &= @_x\, p & (\exists x.\alpha)^t &= @_i \mathsf{F} \downarrow x.\alpha^t \end{aligned}$$

The polynomial-time computable reduction function f is defined by

$$f(\alpha) = \downarrow i.\big(\neg\mathsf{F} i \wedge \alpha^t\big).$$

In order to argue that each formula α is satisfiable iff $f(\alpha)$ is satisfiable, we assume w.l.o.g. that α is a sentence (see proof of Lemma 5.9). For the "$\Rightarrow$" direction, suppose α is satisfied by a model $\mathcal{M} = (D, I)$. By adding the spypoint s to D, we obtain the hybrid model $\mathcal{M}^{\triangleright} = (M^{\triangleright}, R^{\triangleright}, V^{\triangleright})$, where $M^{\triangleright} = D \cup \{s\}$, $R^{\triangleright} = I(R) \cup \{(s, d) \mid d \in D\}$, and $V^{\triangleright}(p) = I(P)$. It can be shown via straightforward induction that $\mathcal{M}^{\triangleright}$ satisfies $f(\alpha)$ at s — under any assignment, since $f(\alpha)$ is a sentence.

For the "$\Leftarrow$" direction, suppose $f(\alpha)$ is satisfied at state s of some hybrid model $\mathcal{M} = (M, R, V)$. The composition of $f(\alpha)$ enforces s to behave as the spypoint. It is straightforward to show that $\mathcal{M}^{\triangleleft} = (M - \{s\}, I)$, where $I(R) = R{\restriction}_{M-\{s\}}$ and $I(P) = V(p)$, satisfies α.

(2). We modify the reduction from (1) simulating the @ operator by means of P, which is possible in the presence of a spypoint and transitivity. We simply re-define $(\cdot)^t$ as follows.

$$\begin{aligned} \bot^t &= \bot & (\neg\alpha)^t &= \neg(\alpha^t) \\ (xRy)^t &= \mathsf{P}\big(i \wedge \mathsf{F}(x \wedge \mathsf{F} y)\big) & (\alpha \vee \beta)^t &= \alpha^t \vee \beta^t \\ \big(P(x)\big)^t &= \mathsf{P}\big(i \wedge \mathsf{F}(x \wedge p)\big) & (\exists x.\alpha)^t &= \mathsf{P}(i \wedge \mathsf{F} \downarrow x.\alpha^t) \end{aligned}$$

The rest of the proof is the same as for (1).

(3). Here we assume w.l.o.g. that any formula $\alpha \in [\text{all}, (4,1)]$ is in prefix notation, that is, it is of the form $\alpha = \mathsf{Q}_1 x_1 \ldots \mathsf{Q}_n x_n.\beta$, where each Q_i is either $\exists$ or $\forall$, and β is a quantifier-free formula. The idea is to simulate the first-order quantifiers by means of the hybrid ones without leaving the spypoint. Then it remains to simulate predicates by F-steps.

We define a translation function $(\cdot)^t$ from $[\text{all}, (4,1)]$ to $\mathcal{HL}(\mathsf{F}, \exists)$ as follows.

$$
\begin{aligned}
\bot^t &= \bot & (\neg\alpha)^t &= \neg(\alpha^t) \\
(xRy)^t &= \mathsf{F}(x \wedge \mathsf{F}y) & (\alpha \vee \beta)^t &= \alpha^t \vee \beta^t \\
\big(P(x)\big)^t &= \mathsf{F}(x \wedge p) & (\exists x.\alpha)^t &= \exists x.(\mathsf{F}x \wedge \alpha^t)
\end{aligned}
$$

The polynomial-time computable reduction function f is defined by

$$f(\alpha) = \downarrow i.\big(\neg \mathsf{F}i \wedge \alpha^t\big).$$

❑

The following result is a consequence of Lemma 5.10 and the coRE-completeness of $\mathcal{FOL}$-SAT.

Theorem 5.11 *Let* $X \in \big\{\{\mathsf{F},\mathsf{P},\downarrow\}, \{\mathsf{F},\downarrow,@\}, \{\mathsf{F},\mathsf{P},\downarrow,@\}, \{\mathsf{F},\exists\}, \{\mathsf{F},\mathsf{P},\exists\}, \{\mathsf{F},\downarrow,\mathsf{E}\}\big\}$. *Then* $\mathcal{HL}(X)$-trans-SAT *is* coRE*-complete.*

The complexity of $\mathcal{HL}(\mathsf{U},\downarrow)$-trans-SAT is still open. Theorem 5.3 (3) implies NEXPTIME-hardness of this problem, while coRE-membership is obvious.

5.5 Transitive trees

The situation over transitive trees is similar to that over transitive frames insofar as for the same languages, the complexity of satisfiability has been known. (Remember from Section 3.2 that $\mathcal{HL}(\mathsf{F},\downarrow)$ and $\mathcal{HL}(\mathsf{U},\downarrow)$ coincide with $\mathcal{HL}(\mathsf{F})$ and $\mathcal{HL}(\mathsf{U})$, respectively.)

Theorem 5.12 *Let* $X \in \big\{\{\mathsf{F}\}, \{\mathsf{F},@\}, \{\mathsf{F},\mathsf{E}\}, \{\mathsf{F},\mathsf{P}\}, \{\mathsf{F},\mathsf{P},@\}, \{\mathsf{F},\mathsf{P},\mathsf{E}\}\big\}$.
Then $\mathcal{HL}(X)$-trans-SAT *is* PSPACE*-complete. [ABM00]*

As in the previous section, we examine the remaining cases in two groups.

5.5.1 Binder-free until/since languages

We will now examine satisfiability over transitive trees for all binder-free until/since languages. EXPTIME-hardness follows from Lemma 5.4. For EXPTIME-membership, we use an embedding of $\mathcal{HL}(\mathsf{U},\mathsf{S},\mathsf{E})$ into $\mathcal{PDL}_{\text{tree}}$, which is introduced in Section 2.4.2.

Lemma 5.13 *$\mathcal{HL}(\mathsf{U},\mathsf{S},\mathsf{E})$-tt-SAT is in* EXPTIME.

Proof. We reduce $\mathcal{HL}(\mathsf{U},\mathsf{S},\mathsf{E})$-tt-SAT to $\mathcal{PDL}_{\text{tree}}$-SAT and define a translation $(\cdot)^t : \mathcal{HL}(\mathsf{U},\mathsf{S},\mathsf{E}) \to \mathcal{PDL}_{\text{tree}}$ as follows (where nominals are translated into atomic propositions).

$$\begin{aligned} p^t &= p, \quad p \in \text{ATOM} & & \\ \bot^t &= \bot & (\mathsf{E}\varphi)^t &= \langle \mathsf{up}^*; \mathsf{down}^* \rangle\, \varphi^t \\ (\neg\varphi)^t &= \neg(\varphi^t) & (\varphi \mathsf{U} \psi)^t &= \big\langle (\mathsf{down}; \varphi^t?)^*; \mathsf{down} \big\rangle\, \psi^t \\ (\varphi \vee \psi)^t &= \varphi^t \vee \psi^t & (\varphi \mathsf{S} \psi)^t &= \big\langle (\mathsf{up}; \varphi^t?)^*; \mathsf{up} \big\rangle\, \psi^t \end{aligned}$$

Since $\mathcal{PDL}_{\text{tree}}$ has no nominals, we must enforce that the translation of each nominal is true at exactly one point by requiring

$$\begin{aligned} \nu(i) = \langle \mathsf{down}^* \rangle\, i \ \wedge\ [\mathsf{down}^*] \Big(i \to \big([\mathsf{down}^+]\, \neg i \wedge [\mathsf{up}^+]\, \neg i \\ \wedge [\mathsf{up}^*; \mathsf{left}^+; \mathsf{down}^*]\, \neg i \wedge [\mathsf{up}^*; \mathsf{right}^+; \mathsf{down}^*]\, \neg i \big) \Big) \end{aligned}$$

to hold for each nominal i. As a reduction function, we have

$$f(\varphi) = \langle \mathsf{down}^* \rangle\, \varphi^t \wedge \bigwedge_{i \in \text{NOM}(\varphi)} \nu(i).$$

It is clear that f is computable in polynomial time and straightforward to show that f is indeed a reduction function: Suppose φ is satisfiable in some finite transitive tree model $\mathcal{M} = (M, R, V)$ based on the tree (M, R') with root m. Then $f(\varphi)$ is satisfiable in m of the $\mathcal{PDL}_{\text{tree}}$ model based on the tree (M, R'), equipped with the valuation V. For the converse, if $f(\varphi)$ is satisfied at the root of some $\mathcal{PDL}_{\text{tree}}$ model $\mathcal{M} = (M, R_{\mathsf{down}}, R_{\mathsf{right}}, V)$, then φ^t is true at some point m, and each nominal is true at exactly one point of $\mathcal{M}$. Hence $(M, R^+_{\mathsf{down}}, V)$ — where R^+_{down} is the transitive closure of R_{down} — is a hybrid transitive tree model satisfying φ at m.

Now there is one drawback in the reduction via f. According to our definition of a tree, it is not necessary that a (transitive) tree is finite or has a root. A node can have infinitely many successors, or there may be an infinitely long forward or backward path from some point. For most practical applications these cases are certainly hardly of interest, but we strive for a more general result. If we do allow for infinite depth or width, the above embedding into $\mathcal{PDL}_{\text{tree}}$ — which is interpreted over finite, rooted trees — is not sufficient.

To overcome finiteness, it suffices to re-examine the proof for the EXPTIME upper bound of $\mathcal{PDL}_{\text{tree}}$-satisfiability in [ABD+05]. This proof in fact covers

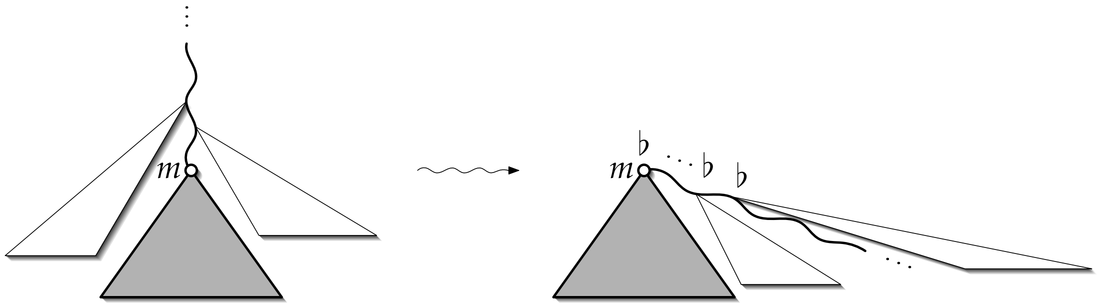


Figure 5.8: Making predecessors successors

a more general result, too, namely that satisfiability of $\mathcal{PDL}_{\text{tree}}$ formulae over (not necessarily finite) trees is in EXPTIME.

To incorporate the fact that "our" trees do not need to have roots, we first observe that satisfiability over *rooted* transitive trees is reducible to satisfiability over (arbitrary) transitive trees, because a root is expressible by $\mathsf{PH}\bot$ in our language. (Note that the lower bound from Lemma 5.4 holds with respect to rooted transitive trees, hence it holds for arbitrary ones, too.)

In order to obtain the upper bound with respect to arbitrary transitive trees, we propose a modification of the above reduction via f. The basic idea is to turn the backward path from the node m (that is to satisfy φ) into a forward path, such that m becomes the root of the transformed model. Thus all predecessors of m (and *their* predecessors etc.) become successors and must be marked by a fresh proposition $\flat$. (See Figure 5.8.)

As a first step, we construct a new translation $(\cdot)^{t\flat}$ from $(\cdot)^{t}$ retaining all but the U/S-cases. For U/S, we replace all occurrences of the programmes down and up by programmes that incorporate the new structure and the fact that for $\flat$-nodes, their predecessors used to be their successors, and their $\flat$-successors used to be their predecessors. Hence, we define

$$
\begin{aligned}
(\varphi \mathsf{U} \psi)^{t\flat} &= \big\langle (\mathsf{dn}'; \varphi^{t\flat}?)^*; \mathsf{dn}' \big\rangle\, \psi^{t\flat}, \quad \text{where} \quad \mathsf{dn}' = (\mathsf{down}; \neg\flat?) \cup (\flat?; \mathsf{up})\,, \\
(\varphi \mathsf{S} \psi)^{t\flat} &= \big\langle (\mathsf{up}'; \varphi^{t\flat}?)^*; \mathsf{up}' \big\rangle\, \psi^{t\flat}, \quad \text{where} \quad \mathsf{up}' = (\neg\flat?; \mathsf{up}) \cup (\flat?; \mathsf{down}; \flat?)\,.
\end{aligned}
$$

Note that we do not change the translation of $\mathsf{E}\varphi$. It only remains to enforce that there is exactly one path at whose every node $\flat$ is true. This means that $\flat$ must be true at the root node and at exactly one successor of each node satisfying $\flat$. This can be expressed by

$$
\begin{aligned}
\beta = \flat \wedge [\mathsf{down}^*] \Big(\flat \to \big([\mathsf{left}^+]\,\neg\flat \wedge [\mathsf{right}^+]\,\neg\flat \wedge \langle \mathsf{down}\rangle\, \flat\big)\Big) \\
\wedge [\mathsf{down}^*] \big(\neg\flat \to [\mathsf{down}]\neg\flat\big)\,.
\end{aligned}
$$

It is now straightforward to show that $f^\flat$, given by

$$f^\flat(\varphi) = \varphi^{t\flat} \wedge \beta \wedge \bigwedge_{i \in \mathrm{NOM}(\varphi)} \nu(i)\,,$$

is indeed a polynomial-time computable reduction function. (Note that $\varphi^{t\flat}$ replaces $\langle \mathsf{down}^* \rangle\, \varphi^{t\flat}$, because we have turned m into the new root node.) ❑

The following theorem is a consequence of Lemmata 5.4 and 5.13.

Theorem 5.14 *Let* $X \in \big\{\{\mathsf{U}\}, \{\mathsf{U},\mathsf{P}\}, \{\mathsf{U},\mathsf{S}\}, \{\mathsf{U},@\}, \{\mathsf{U},\mathsf{P},@\}, \{\mathsf{U},\mathsf{S},@\}, \{\mathsf{U},\mathsf{E}\}, \{\mathsf{U},\mathsf{P},\mathsf{E}\}, \{\mathsf{U},\mathsf{S},\mathsf{E}\}\big\}$.
Then $\mathcal{HL}(X)$-tt-SAT *is* EXPTIME-*complete.*

5.5.2 Languages with binders

Decidability of satisfiability over transitive trees for even the full hybrid language is an immediate consequence of the decidability of the monadic second-order theory of the countably branching tree, SωS, [BGG97]. However, we have to face a nonelementary lower bound for the smallest binder languages $\mathcal{HL}(\mathsf{F},\downarrow,@)$, $\mathcal{HL}(\mathsf{F},\mathsf{P},\downarrow)$, and $\mathcal{HL}(\mathsf{F},\exists)$. For the two $\downarrow$-languages, we will obtain this by a reduction from $\mathcal{HL}(\mathsf{F},\mathsf{P},\downarrow)$-$(\mathbb{N},>)$-SAT, which is nonelementarily decidable [FdRS03]. For the $\exists$-language, we will show a more general result, covering a whole range of frame classes. This will be accomplished by a reduction from $\mathcal{FOL}$-Strings-SAT (see Section 2.4.3), modifying a reduction from the latter to $\mathcal{HL}(\mathsf{F},\downarrow,@)$-lin-SAT given in [MSSW07].

Lemma 5.15

(1) $\mathcal{HL}(\mathsf{F},\mathsf{P},\downarrow)$-$(\mathbb{N},>)$-SAT $\leqslant^{\mathrm{P}}_{\mathrm{m}}$ $\mathcal{HL}(\mathsf{F},\mathsf{P},\downarrow)$-tt-SAT.
(2) $\mathcal{HL}(\mathsf{F},\mathsf{P},\downarrow)$-$(\mathbb{N},>)$-SAT $\leqslant^{\mathrm{P}}_{\mathrm{m}}$ $\mathcal{HL}(\mathsf{F},\downarrow,@)$-tt-SAT.

Proof.
(1). The frame $(\mathbb{N},>)$ is a special case of a transitive tree. Our language is strong enough to enforce that any given transitive tree model is based on $(\mathbb{N},>)$. All we have to do is require two properties:

(i) Every point has at most one direct successor.
(ii) The underlying frame is rooted.

Property (ii) is expressed by $\mathsf{PH}\bot$. Property (i) can be formulated as follows. For any state x, whenever x has some successor, then we name one of the *direct* successors y and ensure that *all* direct successors of x satisfy y. This translates as

$$\lambda = \mathsf{F}\top \rightarrow \mathsf{F}^1{\downarrow} y.\mathsf{P}^1\mathsf{G}^1 y,$$

where F^1, P^1, and G^1 can be expressed by means of U and S, for example $\mathsf{F}^1\varphi \equiv \bot\mathsf{U}\varphi$. But $\varphi\mathsf{U}\psi$ and $\varphi\mathsf{S}\psi$ can be simulated in $\mathcal{HL}(\mathsf{F},\mathsf{P},\downarrow)$, as is shown in Fact 3.2 (8)–(9).

Hence λ is expressible in our language and of constant length. A polynomial-time computable reduction function f is given by $f(\varphi) = \varphi \wedge \lambda \wedge \mathsf{H}\lambda \wedge \mathsf{HG}\lambda \wedge \mathsf{PH}\bot$. It is straightforward to show that φ is satisfiable in some model based on $(\mathbb{N},>)$ iff $f(\varphi)$ is satisfiable in some transitive tree.

(2). In the case of $\mathcal{HL}(\mathsf{F},\downarrow,@)$-tt-SAT, we first have to simulate the P operator. This achieved via a variation of the spypoint technique. We simply label one point in the transitive tree by a fresh nominal i and simulate P using $\downarrow$, a fresh state variable v, and i. This is done in the following translation function $(\cdot)^t : \mathcal{HL}(\mathsf{F},\mathsf{P},\downarrow) \rightarrow \mathcal{HL}(\mathsf{F},\downarrow,@)$.

$$\begin{aligned}
\bot^t &= \bot & (\mathsf{F}\psi)^t &= \mathsf{F}(\psi^t)\\
a^t &= a, \qquad a \in \text{ATOM} & (\mathsf{P}\psi)^t &= {\downarrow} v.@_i\mathsf{F}(\mathsf{F}v \wedge \psi^t)\\
(\neg\psi)^t &= \neg(\psi^t) & ({\downarrow} x.\psi)^t &= {\downarrow} x.(\psi^t)\\
(\psi_1 \vee \psi_2)^t &= \psi_1^t \vee \psi_2^t
\end{aligned}$$

It is easy to see that for each model $\mathcal{M}$ based on $(\mathbb{N},>)$, for each point $x \in \mathbb{N}$, and for each formula $\varphi \in \mathcal{HL}(\mathsf{F},\mathsf{P},\downarrow)$: whenever $\mathcal{M},0 \Vdash i$, then $\mathcal{M},x \Vdash \varphi \;\Leftrightarrow\; \mathcal{M},x \Vdash \varphi'$.

The point s labelled i represents the initial state of the frame $(\mathbb{N},>)$. In the language $\mathcal{HL}(\mathsf{F},\downarrow,@)$, it is not possible to express Property (ii). This is in fact not necessary because the translation function $(\cdot)^t$ never refers to the past of s. This is in particular ensured by the definition of $(\mathsf{P}\psi)^t$. It remains to ensure Property (i). This is done by replacing λ by

$$\lambda' = \mathsf{F}\top \rightarrow {\downarrow} x.\mathsf{F}^1{\downarrow} y.@_x\mathsf{G}^1 y$$

and again expressing F^1 and G^1 by means of U (as shown for F^1 and G^1 above), and simulating U in $\mathcal{HL}(\mathsf{F},\downarrow,@)$ as given in Fact 3.2 (7). Now, a polynomial-time computable reduction function is f', where $f'(\varphi) = {\downarrow} i.(\mathsf{F}\varphi^t \wedge \mu \wedge \lambda' \wedge \mathsf{G}\lambda')$. ❑

Lemma 5.16 *For each class $\mathfrak{F}$ of frames with $(\mathbb{N},>) \in \mathfrak{F} \subseteq \mathsf{trans}$, it holds that $\mathcal{FOL}$-Strings-SAT $\leqslant^{\mathrm{P}}_{\mathrm{m}}$ $\mathcal{HL}(\mathsf{F},\exists)$-$\mathfrak{F}$-SAT.*

Proof. Let α be an arbitrary formula from the language of first-order logic over strings. Without loss of generality, we assume that α is a sentence in prefix notation, that is, it is of the form $\alpha = \mathsf{Q}_1 x_1 \ldots \mathsf{Q}_m x_m.\beta$, where each Q_i is either $\exists$ or $\forall$, and β is a quantifier-free formula whose free variables are among $x_1, \ldots, x_m$. Now it is straightforward to encode α in $\mathcal{HL}(\mathsf{F}, \exists)$ — provided that each satisfying model has a discrete linear submodel that consists of states $0, \ldots, n$, where the states $1, \ldots, n$ correspond to the positions of letters in the string, and 0 acts as a spypoint without name. These properties of the satisfying model will be enforced by the reduction function.

The encoding of α is given by the following translation function $(\cdot)^t$ from the first-order fragment to $\mathcal{HL}(\mathsf{F}, \exists)$.

$$
\begin{aligned}
\bot^t &= \bot & (\neg\gamma)^t &= \neg(\gamma^t)\\
\big(P_\sigma(x)\big)^t &= \mathsf{F}(x \wedge p_\sigma) & (\gamma_1 \vee \gamma_2)^t &= \gamma_1^t \vee \gamma_2^t\\
(x = y)^t &= \mathsf{F}(x \wedge y) & (\exists x.\gamma)^t &= \exists x.(\mathsf{F}x \wedge \gamma^t)\\
(x < y)^t &= \mathsf{F}(x \wedge \mathsf{F}y) & &
\end{aligned}
$$

Now the polynomial-time computable reduction function is given by

$$f(\alpha) = \alpha^t \wedge \mathsf{ENDPOINT} \wedge \mathsf{SUCC} \wedge \mathsf{PRED} \wedge \mathsf{NOBRANCH} \wedge \mathsf{UNIQUE},$$

where the five additional conjuncts enforce that each satisfying transitive model has a submodel whose states behave exactly as the positions in a string. In particular, the conjuncts require (in this order)

- the existence of an endpoint (expressing a finite length of the string);
- each state to have at least one direct successor;
- each state to have at least one direct predecessor;
- each state to have at most one direct successor; and
- the occurrence of exactly one alphabet symbol per position.

(Note that, while we need to complement the second requirement by the third, a "mirror image" of the fourth one will not be necessary.) To express these requirements, we introduce a fresh nominal e that marks the last position of the string, and define the five conjuncts as follows (where $\mathsf{G}^*\varphi = \varphi \wedge \mathsf{G}\varphi$).

$$
\begin{aligned}
\mathsf{ENDPOINT} &= \mathsf{F}e\\
\mathsf{SUCC} &= \mathsf{G}^*\big(\mathsf{F}e \to \exists x.(\mathsf{F}(x \wedge \mathsf{F}e) \wedge \neg\mathsf{F}\mathsf{F}x)\big)\\
\mathsf{PRED} &= \forall x.\big(\mathsf{F}(x \wedge \mathsf{F}e) \to \mathsf{F}(\mathsf{F}x \wedge \neg\mathsf{F}\mathsf{F}x)\big)\\
\mathsf{NOBRANCH} &= \mathsf{G}^*\forall x.\forall y.\big((\mathsf{F}x \wedge \neg\mathsf{F}\mathsf{F}x \wedge \mathsf{F}y \wedge \neg\mathsf{F}\mathsf{F}y) \to \mathsf{F}(x \wedge y)\big)\\
\mathsf{UNIQUE} &= \mathsf{G} \bigvee_{\sigma \in \Sigma} \big(\sigma \wedge \bigwedge_{\sigma' \neq \sigma} \neg\sigma'\big)
\end{aligned}
$$

Let α be a sentence of first-order logic over strings. We will show two propositions, which immediately imply the statement of this lemma.

(1) If α is satisfiable over a string, then $f(\alpha)$ is satisfiable in some model based on $(\mathbb{N}, >)$.

(2) If $f(\alpha)$ is satisfiable a transitive model, then α is satisfiable over a string.

(1). Suppose α is satisfied by $\mathcal{M} = (D^n, I^s)$ with $n \in \mathbb{N}$ and $s = w_0 \dots w_{n-1} \in \Sigma^n$. From $\mathcal{M}$ we construct the hybrid model $\mathcal{M}^{\triangleright} = (\mathbb{N}, >, V^{\triangleright})$ with $V^{\triangleright}(p_\sigma) = \{k+1 \mid k \in I^s(P_\sigma)\}$ and $V^{\triangleright}(e) = n$. It is easy to see from this construction that for *any* assignment g for $\mathcal{M}$, it holds that $\mathcal{M}^{\triangleright}, g, 0 \Vdash \mathsf{ENDPOINT} \wedge \mathsf{SUCC} \wedge \mathsf{PRED} \wedge \mathsf{NOBRANCH} \wedge \mathsf{UNIQUE}$. The missing fact $\mathcal{M}^{\triangleright}, g, 0 \Vdash \alpha^t$ can be shown via a straightforward induction.

(2). Suppose there exist a transitive hybrid model $\mathcal{M} = (M, R, V)$, an assignment g for $\mathcal{M}$, and a state $m \in M$ such that $\mathcal{M}, g, m \Vdash f(\alpha)$. The conjuncts ENDPOINT, SUCC, PRED, NOBRANCH of $f(\alpha)$ cause m to have finitely many successors $m_0, \dots, m_{n-1}$ forming a discrete linear order $mRm_0R \dots Rm_{n-1}$ with $V(e) = \{m_{n-1}\}$. Now the first-order structure $\mathcal{M}^{\triangleleft} = (D^n, I)$, where $I(P_\sigma) = V(p_\sigma)$, is an appropriate structure, due to UNIQUE. By a straightforward induction, it can be shown that $\mathcal{M}^{\triangleleft}$ satisfies α. ❑

The following theorem is a consequence of Lemmata 5.15 and 5.16, and the remarks at the begin of this subsection.

Theorem 5.17
Let $X \in \big\{\{\mathsf{F}, \mathsf{P}, \downarrow\}, \{\mathsf{F}, \downarrow, @\}, \{\mathsf{F}, \mathsf{P}, \downarrow, @\}, \{\mathsf{F}, \exists\}, \{\mathsf{F}, \mathsf{P}, \exists\}, \{\mathsf{F}, \downarrow, \mathsf{E}\}\big\}$.
Then $\mathcal{HL}(X)$-tt-SAT *is nonelementarily decidable.*

5.6 Linear frames

Known complexity results for satisfiability over linear frames are the following.

Theorem 5.18

(1) Let $X \in \big\{\{\mathsf{F}\}, \{\mathsf{F}, @\}, \{\mathsf{F}, \mathsf{E}\}, \{\mathsf{F}, \mathsf{P}\}, \{\mathsf{F}, \mathsf{P}, @\}, \{\mathsf{F}, \mathsf{P}, \mathsf{E}\}\big\}$.
Then $\mathcal{HL}(X)$-lin-SAT *is* NP*-complete. [ABM00]*

(2) Let $X \in \big\{\{\mathsf{F}, \mathsf{P}, \downarrow\}, \{\mathsf{F}, \mathsf{P}, \downarrow, @\}, \{\mathsf{F}, \mathsf{P}, \exists\}, \{\mathsf{F}, \downarrow, \mathsf{E}\}\big\}$.
Then $\mathcal{HL}(X)$-lin-SAT *is nonelementarily decidable. [FdRS03]*

(3) $\mathcal{HL}(\mathsf{F}, \downarrow, @)$-lin-SAT *is nonelementarily decidable. [MSSW07]*

The complexity of the remaining language with binders, namely $\mathcal{HL}(\mathsf{F}, \exists)$, is an immediate consequence of Lemma 5.16 and Theorem 5.18 (2).

Theorem 5.19 $\mathcal{HL}(\mathsf{F}, \exists)$-lin-SAT *is nonelementarily decidable.*

It remains to examine binder-free until/since languages. Their satisfiability problems are all PSPACE-hard, which is implied by the following result.

Theorem 5.20 ([Rey03]) $\mathcal{ML}(\mathsf{U})$-lin-SAT *is* PSPACE*-complete.*

We will now show how to carry over the upper bound to $\mathcal{HL}(\mathsf{U}, \mathsf{E})$.

Lemma 5.21 $\mathcal{HL}(\mathsf{U}, \mathsf{E})$-lin-SAT $\leqslant_{\mathrm{m}}^{\mathrm{P}}$ $\mathcal{ML}(\mathsf{U})$-lin-SAT.

Proof. In order to find an appropriate reduction, we have to simulate nominals and the E operator. It has been shown in [ABM00] how to transform $\mathcal{HL}(\mathsf{F}, \mathsf{P}, \mathsf{E})$-formulae into equisatisfiable $\mathcal{HL}(\mathsf{F}, \mathsf{P})$-formulae of length quadratic in the size of the original formula. We extend this technique such that it works in the absence of past operators and permits to express nominals as well. The underlying idea is not difficult, but the exact proof of the correctness of the reduction will require some technical details.

The first step is the introduction of a spypoint. Every linear structure remains so when adding a spypoint s. Since it will not be necessary to jump to s, we do not need a nominal for it.

Once we have the spypoint, it is straightforward to replace each nominal i by a new atomic proposition i'. Let $\varphi \in \mathcal{HL}(\mathsf{U}, \mathsf{E})$. Then the following formula enforces that, for each $i \in \mathrm{NOM}(\varphi)$, i' behaves as a nominal.

$$\mathsf{NOMINALS} = \bigwedge_{i \in \mathrm{NOM}(\varphi)} \big(\mathsf{F}i' \wedge \mathsf{G}(i' \rightarrow \mathsf{G}\neg i')\big)$$

Let φ' be the formula obtained from φ by replacing each occurrence of any nominal i by i'. Then $\mathsf{F}\varphi' \wedge \mathsf{NOMINALS} \in \mathcal{ML}(\mathsf{U}, \mathsf{E})$, and it is easy to see that $\varphi \in \mathcal{HL}(\mathsf{U}, \mathsf{E})$-lin-SAT if and only if $\mathsf{F}\varphi' \wedge \mathsf{NOMINALS} \in \mathcal{ML}(\mathsf{U}, \mathsf{E})$-lin-SAT.

In order to simulate the E operator, let $\mathsf{E}\psi_1, \ldots, \mathsf{E}\psi_n$ be all E-subformulae of φ'. Now again we can successively replace each $\mathsf{E}\psi_k$ by a fresh atomic proposition e_k, provided that we enforce e_k to behave exactly as $\mathsf{E}\psi_k$.

For each subformula ϑ of φ, let ϑ'' be the formula obtained from ϑ by first replacing nominals as above and then substituting e_k for each maximal occurrence of some $\mathsf{E}\psi_k$. (We call an occurrence of $\mathsf{E}\psi_k$ in ϑ maximal if it is not a subformula of any other $\mathsf{E}\psi_\ell$.) Furthermore, let

$$\mathsf{E\text{-}OP} = \bigwedge_{k=1}^{n} \big((\mathsf{F}\psi_k'' \rightarrow \mathsf{G}e_k) \wedge (\neg\mathsf{F}\psi_k'' \rightarrow \mathsf{G}\neg e_k)\big).$$

Clearly, neither φ'' nor E-OP contain any occurrence of the E operator.

Let $f(\varphi) = \mathsf{F}\varphi'' \wedge \mathsf{NOMINALS} \wedge \mathsf{E}\text{-}\mathsf{OP}$. From this construction it is clear that $f(\varphi) \in \mathcal{ML}(\mathsf{U})$, and that the size of $f(\varphi)$ is quadratic in the size of φ. It remains to show that $\varphi \in \mathcal{HL}(\mathsf{U},\mathsf{E})$-lin-SAT if and only if $f(\varphi) \in \mathcal{ML}(\mathsf{U})$-lin-SAT.

"$\Rightarrow$". Suppose $\varphi \in \mathcal{HL}(\mathsf{U},\mathsf{E})$-lin-SAT. Then there exist a hybrid model $\mathcal{M} = (M, R, V)$ (based on a linear frame) and a state $m_0 \in \mathcal{M}$ such that $\mathcal{M}, m \Vdash \varphi$. From $\mathcal{M}$ we construct a model $\mathcal{M}^{\triangleright} = (M^{\triangleright}, R^{\triangleright}, V^{\triangleright})$, where

$$
\begin{aligned}
M^{\triangleright} &= M \uplus \{s\} \\
R^{\triangleright} &= R \cup \{(s,m) \mid m \in M\} \\
V^{\triangleright}(p) &= V(p), && \text{for any } p \in \text{PROP} \\
V^{\triangleright}(i') &= V(i), && \text{for any } i \in \text{NOM} \\
V^{\triangleright}(e_k) &= \begin{cases} M & \text{if } \exists m \in M(\mathcal{M}, m \Vdash \psi_k) \\ \emptyset & \text{otherwise} \end{cases} && \text{for } k = 1, \dots, n
\end{aligned}
$$

This definition ensures that $\mathcal{M}^{\triangleright}$ is based on a linear frame, too. Clearly, $\mathcal{M}^{\triangleright}, s \Vdash \mathsf{NOMINALS}$. It is also the case that $\mathcal{M}^{\triangleright}, s \Vdash \mathsf{F}\varphi''$. This is a consequence of sRm_0, $\mathcal{M}, m_0 \Vdash \varphi$, and the following claim.

Claim. For each subformula ϑ of φ and each state $m \in M$: $\mathcal{M}, m \Vdash \vartheta$ if and only if $\mathcal{M}^{\triangleright}, m \Vdash \vartheta''$.

Proof of Claim. We use induction on ϑ. The cases for $\bot$, atomic propositions, nominals, $\neg$, and $\vee$ are obvious, where latter two simply require the easy observation that, for instance, $\neg(\xi'') = (\neg\xi)''$.

- $\vartheta = \mathsf{E}\psi_k$.

$$
\begin{aligned}
\mathcal{M}, m \Vdash \mathsf{E}\psi_k &\Leftrightarrow \exists n \in M(\mathcal{M}, n \Vdash \psi_k) && \text{(satisfaction for E)} \\
&\Leftrightarrow V^{\triangleright}(e_k) = M && \text{(construction of } V^{\triangleright}) \\
&\Leftrightarrow \mathcal{M}^{\triangleright}, m \Vdash e_k && (\text{``}\Leftarrow\text{'': construction of } V^{\triangleright})
\end{aligned}
$$

- $\vartheta = \xi_1 \mathsf{U} \xi_2$.

$$
\begin{aligned}
&\mathcal{M}, m \Vdash \xi_1 \mathsf{U} \xi_2 \\
&\overset{(1)}{\Leftrightarrow} \exists n \in M\big(mRn \;\&\; \mathcal{M}, n \Vdash \xi_2 \;\&\; \forall \ell \in M(mR\ell Rn \Rightarrow \mathcal{M}, \ell \Vdash \xi_1)\big) \\
&\overset{(2)}{\Leftrightarrow} \exists n \in M\big(mRn \;\&\; \mathcal{M}^{\triangleright}, n \Vdash \xi_2'' \;\&\; \forall \ell \in M(mR\ell Rn \Rightarrow \mathcal{M}^{\triangleright}, \ell \Vdash \xi_1'')\big) \\
&\overset{(3)}{\Leftrightarrow} \exists n \in M^{\triangleright}\big(mR^{\triangleright}n \;\&\; \mathcal{M}^{\triangleright}, n \Vdash \xi_2'' \;\&\; \forall \ell \in M^{\triangleright}(mR^{\triangleright}\ell R^{\triangleright}n \Rightarrow \mathcal{M}^{\triangleright}, \ell \Vdash \xi_1'')\big) \\
&\overset{(4)}{\Leftrightarrow} \mathcal{M}^{\triangleright}, m \Vdash \xi_1'' \mathsf{U} \xi_2'' \\
&\overset{(5)}{\Leftrightarrow} \mathcal{M}^{\triangleright}, m \Vdash (\xi_1 \mathsf{U} \xi_2)''
\end{aligned}
$$

These equivalences are justified as follows.

(1) Definition of satisfaction for the U operator.

(2) Induction hypothesis.

(3) Due to the construction of $M^{\triangleright}$, $R^{\triangleright}$, and because of $m \in M$, the property ($n \in M$ and mRn) is equivalent to ($n \in M^{\triangleright}$ and $mR^{\triangleright}n$). The same holds for ℓ in place of n.

(4) Definition of satisfaction for the U operator.

(5) Obvious.

This finishes the proof of the claim.

It remains to show that $\mathcal{M}^{\triangleright}, s \Vdash \mathsf{E\text{-}OP}$. We will prove $\mathcal{M}^{\triangleright}, s \Vdash \mathsf{F}\psi_k'' \to \mathsf{G}e_k$, for $k = 1, \ldots, n$. The remaining conjuncts of $\mathsf{E\text{-}OP}$ are treated analogously.

$$\begin{aligned}
\mathcal{M}^{\triangleright}, s \Vdash \mathsf{F}\psi_k'' &\Rightarrow \circledexists m \in M(\mathcal{M}^{\triangleright}, m \Vdash \psi_k'') && \text{(construction of } M^{\triangleright}, R^{\triangleright}\text{)}\\
&\Rightarrow \circledexists m \in M(\mathcal{M}, m \Vdash \psi_k) && \text{(previous claim)}\\
&\Rightarrow V^{\triangleright}(e_k) = M && \text{(construction of } V^{\triangleright}\text{)}\\
&\Rightarrow \circledforall m \in M(\mathcal{M}^{\triangleright}, m \Vdash e_k) && \text{(definition of satisfaction for } e_k\text{)}\\
&\Rightarrow \mathcal{M}^{\triangleright}, s \Vdash \mathsf{G}e_k && \text{(construction of } M^{\triangleright}, R^{\triangleright}\text{)}
\end{aligned}$$

"$\Leftarrow$". Suppose $f(\varphi) \in \mathcal{ML}(\mathsf{U})\text{-lin-}\textsc{Sat}$. Then there exist a Kripke model $\mathcal{M} = (M, R, V)$ (based on a linear frame) and a state $s \in \mathcal{M}$ such that $\mathcal{M}, s \Vdash \mathsf{F}\varphi'' \wedge \mathsf{NOMINALS} \wedge \mathsf{E\text{-}OP}$. Due to the first conjunct, there is a state $m_0 \in M$ with sRm_0 and $\mathcal{M}, m_0 \Vdash \varphi''$. We construct a hybrid model $\mathcal{M}^{\triangleleft} = (M^{\triangleleft}, R^{\triangleleft}, V^{\triangleleft})$ as follows.

$$\begin{aligned}
M^{\triangleleft} &= \{m \in M \mid sRm\} & V^{\triangleleft}(p) &= V(p) \cap M^{\triangleleft}, && \text{for any } p \in \mathrm{PROP}\\
R^{\triangleleft} &= R\restriction_{M^{\triangleleft}} & V^{\triangleleft}(i) &= V(i') \cap M^{\triangleleft}, && \text{for any } i \in \mathrm{NOM}
\end{aligned}$$

The correctness of the definition of $V^{\triangleleft}(i)$ is ensured by $\mathsf{NOMINALS}$. It remains to show that $\mathcal{M}^{\triangleleft}, m_0 \Vdash \varphi$, which is a consequence of $\mathcal{M}, m_0 \Vdash \varphi''$ and the following claim.

Claim. For each subformula ϑ of φ and each state $m \in M^{\triangleleft}$: $\mathcal{M}, m \Vdash \vartheta''$ if and only if $\mathcal{M}^{\triangleleft}, m \Vdash \vartheta$.

Proof of Claim. Again, we use induction on ϑ. All cases except for $\vartheta = \mathsf{E}\psi_k$ are obvious or analogous to the proof of the previous claim. For $\vartheta = \mathsf{E}\psi_k$, we only show the "$\Rightarrow$" direction using the $2k$-th of the $2n$ conjuncts of $\mathsf{E\text{-}OP}$. The opposite direction is shown analogously, starting from $\mathcal{M}, m \nVdash e_k$ and applying conjunct $2k - 1$ of $\mathsf{E\text{-}OP}$.

$$\begin{aligned}
\mathcal{M}, m \Vdash e_k &\Rightarrow \mathcal{M}, s \nVdash \mathsf{G}\neg e_k && \text{(since } m \in M^{\triangleleft}\text{)}\\
&\Rightarrow \mathcal{M}, s \nVdash \neg\mathsf{F}\psi_k'' && \text{(conjunct } 2k \text{ of } \mathsf{E\text{-}OP}\text{)}\\
&\Rightarrow \mathcal{M}, s \Vdash \mathsf{F}\psi_k'' && \text{(satisfaction for } \neg\text{)}
\end{aligned}$$

$$
\begin{aligned}
&\Rightarrow\ \circledexists n \in M^{\lhd}(\mathcal{M}, n \Vdash \psi_k'') && \text{(construction of } M^{\lhd}\text{)}\\
&\Rightarrow\ \circledexists n \in M^{\lhd}(\mathcal{M}^{\lhd}, n \Vdash \psi_k) && \text{(induction hypothesis)}\\
&\Rightarrow\ \mathcal{M}^{\lhd}, m \Vdash \mathsf{E}\psi_k && \text{(satisfaction for } \mathsf{E}\text{)}
\end{aligned}
$$

This finishes the proof of the claim. ❑

The following theorem is a consequence of Theorem 5.20 and Lemma 5.21.

Theorem 5.22 *Let* $X \in \{\{\mathsf{U}\}, \{\mathsf{U}, @\}, \{\mathsf{U}, \mathsf{E}\}\}$.
Then $\mathcal{HL}(X)$-lin-SAT *is* PSPACE-*complete.*

Unfortunately, the cases with U and Past operators are unsolved. Hardness for PSPACE, of course, follows from Theorem 5.20, and decidability is due to Theorem 5.18 (2).

5.7 Natural numbers

If we consider satisfiability over the frame $(\mathbb{N}, >)$, the complexity of all languages but $\mathcal{HL}(\mathsf{F}, \exists)$ has been known. We can contribute a result for the latter. All results are summarised in the following theorem.

Theorem 5.23

(1) Let $X \in \{\{\mathsf{F}\}, \{\mathsf{F}, @\}, \{\mathsf{F}, \mathsf{E}\}, \{\mathsf{F}, \mathsf{P}\}, \{\mathsf{F}, \mathsf{P}, @\}, \{\mathsf{F}, \mathsf{P}, \mathsf{E}\}\}$.
Then $\mathcal{HL}(X)$-$(\mathbb{N}, >)$-SAT *is* NP-*complete. [Mar04, ABM00]*

(2) Let $X \in \{\{\mathsf{U}\}, \{\mathsf{U}, \mathsf{P}\}, \{\mathsf{U}, \mathsf{S}\}, \{\mathsf{U}, @\}, \{\mathsf{U}, \mathsf{P}, @\}, \{\mathsf{U}, \mathsf{S}, @\}, \{\mathsf{U}, \mathsf{E}\}, \{\mathsf{U}, \mathsf{P}, \mathsf{E}\}, \{\mathsf{U}, \mathsf{S}, \mathsf{E}\}\}$.
Then $\mathcal{HL}(X)$-$(\mathbb{N}, >)$-SAT *is* PSPACE-*complete. [SC85, ABM00]*

(3) Let $X \in \{\{\mathsf{F}, \mathsf{P}, \downarrow\}, \{\mathsf{F}, \mathsf{P}, \downarrow, @\}, \{\mathsf{F}, \mathsf{P}, \exists\}, \{\mathsf{F}, \downarrow, \mathsf{E}\}\}$.
Then $\mathcal{HL}(X)$-$(\mathbb{N}, >)$-SAT *is nonelementarily decidable. [FdRS03]*

(4) $\mathcal{HL}(\mathsf{F}, \downarrow, @)$-$(\mathbb{N}, >)$-SAT *is nonelementarily decidable. [MSSW07]*

(5) $\mathcal{HL}(\mathsf{F}, \exists)$-$(\mathbb{N}, >)$-SAT *is nonelementarily decidable.*

As for Part (1), the lower bound is clear. The upper bound is due to NP-membership of $\mathcal{ML}(\mathsf{F}, \mathsf{P})$-$(\mathbb{N}, >)$-SAT [Mar04] plus the fact that, over each class of linear frames, nominals and the E operator can be simulated by F and P without a blowup in formula size [ABM00].

The lower bound for Part (2) is an immediate consequence of PSPACE-hardness of $\mathcal{ML}(\mathsf{U})$-$(\mathbb{N}, >)$-SAT [SC85]. The upper bound follows from PSPACE-membership of $\mathcal{ML}(\mathsf{U}, \mathsf{S})$-$(\mathbb{N}, >)$-SAT [SC85] plus the same simulation result as in the previous case.

Finally, Part (5) is an immediate consequence of Lemma 5.16 and Part (3).

5.8 Frames with equivalence relations

As already observed in Section 3.2, there exist only seven hybrid languages over the class of all ER frames. In this section, we provide complexity results for satisfiability for all of them. In order to obtain results as general as possible, we also consider the pure fragment of each language $\mathcal{HL}(X)$, denoted by $\mathcal{PHL}(X)$.

5.8.1 Languages without binders

We show NP-completeness of satisfiability for all pure and non-pure languages without binders, which is the same complexity as for modal logic over ER frames [Lad77]. The lower bound is almost trivial, and the upper bound is due to the $\mathcal{O}(n^2)$-size model property, which is established by a generalisation of the selection procedure given in [Lad77].

Theorem 5.24 *Let* $X \in \{\{\Diamond\}, \{\Diamond, @\}, \{\Diamond, \mathsf{E}\}\}$.
Then $\mathcal{HL}(X)$-ER-SAT *and* $\mathcal{PHL}(X)$-ER-SAT *are* NP*-complete.*

Proof. For the lower bound, we reduce from the satisfiability problem SAT for propositional logic to $\mathcal{PHL}(\Diamond)$-ER-SAT. Let φ be a propositional formula with atomic propositions $p_1, \ldots, p_n$. The polynomial-time computable reduction function simply replaces each p_k by a nominal i_k. We call the resulting hybrid formula φ'. Clearly, if φ is satisfiable, then there exists a satisfying assignment β for all atomic propositions. Then a satisfying hybrid ER model for φ' consists of states $M = \{0,1\}$, the relation $R = M \times M$, and the valuation function defined by $V(i_k) = \{\beta(p_k)\}$.

Conversely, if φ' is satisfiable in a state m of some hybrid ER model $\mathcal{M} = (M, R, V)$, then a satisfying assignment β for φ is obtained by setting $\beta(p_k) = 1$ iff $V(i_k) = \{m\}$.

For the upper bound, we first prove that $\mathcal{HL}(\Diamond, \mathsf{E})$ has the $\mathcal{O}(n^2)$-size model property with respect to ER frames.

Let $\varphi \in \mathcal{HL}(\Diamond, \mathsf{E})$-ER-SAT. Then there exists a hybrid model $\mathcal{M} = (M, R, V)$ and a state $m_{0,0} \in M$ such that $\mathcal{M}, m_{0,0} \Vdash \varphi$. Let $\mathsf{E}\psi_1, \ldots, \mathsf{E}\psi_k$ and $\Diamond\vartheta_1, \ldots, \Diamond\vartheta_\ell$ be all E- and $\Diamond$-subformulae of φ. Now, for each $\mathsf{E}\psi_i$ that is satisfied at $m_{0,0}$, there is a state $m_{i,0}$ satisfying ψ_i. For every other $\mathsf{E}\psi_i$ choose $m_{i,0} = m_{0,0}$. Furthermore, for each of these $m_{i,0}$ and each $\Diamond\vartheta_j$ that is satisfied at $m_{i,0}$, there is a state $m_{i,j}$ in the cluster of $m_{i,0}$ satisfying ϑ_j. For every other $\Diamond\vartheta_j$, choose $m_{i,j} = m_{i,0}$.

Now let $\mathcal{M}'$ be the restriction of $\mathcal{M}$ to all $m_{i,j}$ with $i, j = 0, \ldots, n$. This model clearly has at most $(n+1)^2$ states and contains $m_{0,0}$. The crucial fact

$\mathcal{M}', m_{0,0} \Vdash \varphi$ follows from the claim that for each subformula ψ of φ and each $m_{i,j}$: $\mathcal{M}, m_{i,j} \Vdash \psi$ iff $\mathcal{M}', m_{i,j} \Vdash \psi$. This claim can be proven by a straightforward induction on ψ.

Let φ be a formula from $\mathcal{HL}(\Diamond, \mathsf{E})$ of length n. Due to the $\mathcal{O}(n^2)$-size model property, it suffices to guess a model of size $\mathcal{O}(n^2)$ and verify whether it satisfies φ. The last step can be done in time polynomial in n, due to [FdR06, Theorem 4.3]. ❑

5.8.2 Languages with binders and without E

We consider the languages $\mathcal{HL}(\Diamond, \downarrow)$, $\mathcal{HL}(\Diamond, \downarrow, @)$, and $\mathcal{HL}(\Diamond, \exists)$ and show that satisfiability is NEXPTIME-complete (Theorem 5.28). Using the hierarchy of the languages, it suffices to prove that $\mathcal{HL}(\Diamond, \downarrow)$-ER-SAT is NEXPTIME-hard (Lemma 5.25), and that $\mathcal{HL}(\Diamond, \downarrow, @)$-ER-SAT and $\mathcal{HL}(\Diamond, \exists)$-ER-SAT are in NEXPTIME (see Lemmata 5.26 and 5.27).

Lemma 5.25 *$\mathcal{HL}(\Diamond, \downarrow)$-ER-SAT is NEXPTIME-hard.*

Proof. It was shown in [MSSW05] that $\mathcal{HL}(\Diamond, \downarrow)$-compl-SAT is NEXPTIME-complete. A complete frame is an ER frame with one cluster only. It is straightforward to reduce $\mathcal{HL}(\Diamond, \downarrow)$-compl-SAT to $\mathcal{HL}(\Diamond, \downarrow)$-ER-SAT. The polynomial-time computable reduction function defined by $f(\varphi) = \varphi \wedge \bigwedge_{i \in \mathrm{NOM}(\varphi)} \Diamond i$ maps φ to a formula that enforces that a satisfying ER model can be restricted to one cluster. ❑

Lemma 5.26 *$\mathcal{HL}(\Diamond, \downarrow, @)$-ER-SAT is in NEXPTIME.*

Proof. It suffices to reduce $\mathcal{HL}(\Diamond, \downarrow, @)$-ER-SAT to $\mathcal{HL}(\Diamond, \downarrow)$-compl-SAT, because the latter is known to be NEXPTIME-complete [MSSW05]. This reduction will rely on two basic observations. First, it suffices to consider sentences only, because free state variables can be replaced by nominals without affecting satisfiability. Second, a satisfying ER model for an $\mathcal{HL}(\Diamond, \downarrow, @)$ sentence φ consists w.l.o.g. of not more clusters than there are nominals in φ plus one.

To put the last observation more formally, let φ be an $\mathcal{HL}(\Diamond, \downarrow, @)$ sentence with nominals $i_1, \ldots, i_n$. If φ is satisfied in a state m of a model $\mathcal{M}$, then φ is satisfied in the restriction of $\mathcal{M}$ to the clusters that contain m and all $V(i_k)$. This is so because other clusters are not accessible by means of $\Diamond$ or @.

Hence we can assume w.l.o.g. that a satisfying model for φ consists of at most $n+1$ clusters, where $n \leqslant |\varphi|$. Such a model can be transformed into a model consisting of one "new" cluster being the union of the "old" clusters. The latter can be distinguished by fresh atomic propositions $c_0, \dots, c_n$, which help simulate $\Diamond$ and @ using only $\Diamond$. This simulation is captured by the following translation from $\mathcal{HL}(\Diamond,\downarrow,@)$ to $\mathcal{HL}(\Diamond,\downarrow)$ using a fresh state variable x.

$$
\begin{aligned}
\bot^t &= \bot \\
a^t &= a \quad (a \in \text{ATOM}) & (\Diamond\varphi)^t &= \downarrow x.\Diamond\Big(\textstyle\bigwedge_{k=0}^{n}\big(c_k \leftrightarrow \Box(x \to c_k)\big) \wedge \varphi^t\Big) \\
(\neg\varphi)^t &= \neg\varphi^t & (@_v\varphi)^t &= \Diamond(v \wedge \varphi^t) \\
(\varphi \wedge \psi)^t &= \varphi^t \wedge \psi^t & (\downarrow v.\varphi)^t &= \downarrow v.\varphi^t
\end{aligned}
$$

Using $(\cdot)^t$, we define the polynomial-time computable reduction function $f : \mathcal{HL}(\Diamond,\downarrow,@) \to \mathcal{HL}(\Diamond,\downarrow)$ by

$$
\begin{aligned}
f(\varphi) = \varphi^t \ \wedge\ c_0 \ \wedge\ \Box \bigvee_{k=0}^{n} c_k \ \wedge\ \Box(i_k \to c_k) \\
\wedge \bigwedge_{\substack{k,\ell=0,\dots,n \\ k\neq\ell}} \Big(\Box(c_k \leftrightarrow c_\ell) \vee \Box\big((c_k \to \neg c_\ell) \wedge (c_\ell \to \neg c_k)\big)\Big),
\end{aligned}
$$

where the conjuncts following φ^t express that φ is satisfied in cluster 0; each state of the new cluster belongs to some old cluster; nominal i_k is true in cluster k; and two clusters k, ℓ are either equal or disjoint. It remains to prove

$$\varphi \in \mathcal{HL}(\Diamond,\downarrow,@)\text{-ER-S}\text{AT} \qquad \text{iff} \qquad f(\varphi) \in \mathcal{HL}(\Diamond,\downarrow)\text{-compl-S}\text{AT}.$$

"$\Rightarrow$". Suppose $\varphi \in \mathcal{HL}(\Diamond,\downarrow,@)$-ER-SAT. This means that there exist a model $\mathcal{M} = (M, R, V)$, some state $m_0 \in M$, and an assignment g_0 for $\mathcal{M}$ such that $\mathcal{M}, g_0, m_0 \Vdash \varphi$. Without loss of generality, $\mathcal{M}$ has only those clusters that are determined by m and all $V(i_k)$. Let $V(i_k) = m_k$, for $k = 1, \dots, n$. We construct a model $\mathcal{M}^\triangleright = (M^\triangleright, R^\triangleright, V^\triangleright)$, where $M^\triangleright = M$, $R = M^\triangleright \times M^\triangleright$, and define $V^\triangleright$ by $V^\triangleright(a) = V(a)$ for $a \in \text{PROP} \cup \text{NOM}$, and $V^\triangleright(c_k) = \{m \in M \mid mRm_k\}$ for $k = 0, \dots, n$. Furthermore, for each assignment g for $\mathcal{M}$, define the assignment $g^\triangleright$ for $\mathcal{M}^\triangleright$ by $g^\triangleright(y) = y$ for each $y \neq x$, and $g^\triangleright(x) = m_0$.

We have to show that $\mathcal{M}^\triangleright, g_0^\triangleright, m_0 \Vdash f(\varphi)$. It is immediately clear from the construction that the conjuncts following φ^t in $f(\varphi)$ are satisfied in m_0 of $\mathcal{M}^\triangleright$ under $g_0^\triangleright$. The fact that $\mathcal{M}^\triangleright, g_0^\triangleright, m_0 \Vdash \varphi^t$ is a consequence of the following claim.

Claim. For each subformula ψ of φ, for each state $m \in M$, and for each assignment g for $\mathcal{M}$:

$$\mathcal{M}, g, m \Vdash \psi \qquad \text{if and only if} \qquad \mathcal{M}^\triangleright, g^\triangleright, m \Vdash \psi^t.$$

Proof of Claim. We proceed by induction on the structure of ψ. The atomic and Boolean cases follow immediately from the construction. The cases for @ and $\downarrow$ are straightforward. It remains to discuss the only interesting case $\psi = \Diamond\vartheta$, which is done via the following chain of equivalent statements.

$$
\begin{aligned}
&\mathcal{M}, g, m \Vdash \Diamond\vartheta \\
\overset{(1)}{\Leftrightarrow}\ & \exists \ell \in M[mR\ell \ \&\ \mathcal{M}, g, \ell \Vdash \vartheta] \\
\overset{(2)}{\Leftrightarrow}\ & \exists \ell \in M[mR\ell \ \&\ \mathcal{M}^{\triangleright}, g^{\triangleright}, \ell \Vdash \vartheta^t] \\
\overset{(3)}{\Leftrightarrow}\ & \exists \ell \in M[mR\ell \ \&\ \mathcal{M}^{\triangleright}, (g^{\triangleright})^x_m, \ell \Vdash \vartheta^t] \\
\overset{(4)}{\Leftrightarrow}\ & \exists \ell \in M^{\triangleright}\big[\forall k \leqslant n\big(\ell \in V^{\triangleright}(c_k) \Leftrightarrow m \in V^{\triangleright}(c_k)\big) \ \&\ \mathcal{M}^{\triangleright}, (g^{\triangleright})^x_m, \ell \Vdash \vartheta^t\big] \\
\overset{(5)}{\Leftrightarrow}\ & \exists \ell \in M^{\triangleright}\big[\mathcal{M}^{\triangleright}, (g^{\triangleright})^x_m, \ell \Vdash \textstyle\bigwedge_{k=0}^n \big(c_k \leftrightarrow \Box(x \to c_k)\big) \ \&\ \mathcal{M}^{\triangleright}, (g^{\triangleright})^x_m, \ell \Vdash \vartheta^t\big] \\
\overset{(6)}{\Leftrightarrow}\ & \mathcal{M}^{\triangleright}, (g^{\triangleright})^x_m, m \Vdash \Diamond\Big(\textstyle\bigwedge_{k=0}^n \big(c_k \leftrightarrow \Box(x \to c_k)\big) \wedge \vartheta^t\Big) \\
\overset{(7)}{\Leftrightarrow}\ & \mathcal{M}^{\triangleright}, g^{\triangleright}, m \Vdash \downarrow x.\Diamond\Big(\textstyle\bigwedge_{k=0}^n \big(c_k \leftrightarrow \Box(x \to c_k)\big) \wedge \vartheta^t\Big) \\
\overset{(8)}{\Leftrightarrow}\ & \mathcal{M}^{\triangleright}, g^{\triangleright}, m \Vdash (\Diamond\vartheta)^t
\end{aligned}
$$

These equivalences are justified as follows.

(1) Definition of satisfaction for $\Diamond$.
(2) Induction hypothesis.
(3) Since x is bound in ϑ^t.
(4) Construction of $M^{\triangleright}$, $V^{\triangleright}$.
(5) Definition of satisfaction for atoms, Boolean operators, and $\Box$.
(6) Definition of satisfaction for $\wedge$ and $\Diamond$.
(7) Definition of satisfaction for $\downarrow$.
(8) Definition of $(\cdot)^t$.

"$\Leftarrow$". Suppose $\varphi \in \mathcal{HL}(\Diamond, \downarrow)$-compl-SAT. This means that there exist a model $\mathcal{M} = (M, R, V)$, some state $m_0 \in M$, and an assignment g_0 for $\mathcal{M}$ such that $\mathcal{M}, g_0, m_0 \Vdash f(\varphi)$. Due to the conjuncts after φ^t in $f(\varphi)$, the variables c_k "almost partition" M in the following sense. Let $\mathrm{Cl}_k = V(c_k)$. Then $m_0 \in \mathrm{Cl}_0$; for each state $m \in M$ there is some $k \leqslant n$ with $m \in \mathrm{Cl}_k$; $V(i_k) \subseteq \mathrm{Cl}_k$; and for two disjoint $k, \ell \leqslant n$, either $\mathrm{Cl}_k = \mathrm{Cl}_\ell$ or $\mathrm{Cl}_k \cap \mathrm{Cl}_\ell = \emptyset$.

Hence the following construction of a model $\mathcal{M}^{\triangleleft} = (M^{\triangleleft}, R^{\triangleleft}, V^{\triangleleft})$ is correct. Let $M^{\triangleleft} = M$, $R^{\triangleleft} = \{(m, \ell) \mid \forall k \leqslant n(m \in \mathrm{Cl}_k \Leftrightarrow \ell \in \mathrm{Cl}_k)\}$, and $V^{\triangleleft}$ be the restriction of V to $(\mathrm{NOM} \cup \mathrm{PROP}) - \bigcup\{c_k\}$. Furthermore, for each assignment g for $\mathcal{M}$, let $g^{\triangleleft} = g$, which is an assignment for $\mathcal{M}^{\triangleleft}$.

It remains to show $\mathcal{M}^{\triangleleft}, g_0^{\triangleleft}, m_0 \Vdash \varphi$, which is a consequence of $\mathcal{M}, g_0, m_0 \Vdash \varphi^t$ and the following claim.

Claim. For each subformula ψ of φ, for each state $m \in M$, and for each assignment g for $\mathcal{M}$:

$$\mathcal{M}, g, m \Vdash \psi^t \qquad \text{if and only if} \qquad \mathcal{M}^{\triangleleft}, g^{\triangleleft}, m \Vdash \psi.$$

Proof of Claim. We proceed by induction on the structure of ψ. Again, the atomic and Boolean cases follow immediately from the construction, and the cases for @ and $\downarrow$ are straightforward. It remains to discuss the only interesting case $\psi = \Diamond\vartheta$, which is done via the following chain of equivalent statements.

$$\begin{aligned}
&\mathcal{M}, g, m \Vdash (\Diamond\vartheta)^t \\
&\overset{(1)}{\Leftrightarrow}\ \mathcal{M}, g, m \Vdash {\downarrow} x.\Diamond\Big(\textstyle\bigwedge_{k=0}^{n} \big(c_k \leftrightarrow \Box(x \to c_k)\big) \wedge \vartheta^t \Big) \\
&\overset{(2)}{\Leftrightarrow}\ \mathcal{M}, g_m^x, m \Vdash \Diamond\Big(\textstyle\bigwedge_{k=0}^{n} \big(c_k \leftrightarrow \Box(x \to c_k)\big) \wedge \vartheta^t \Big) \\
&\overset{(3)}{\Leftrightarrow}\ \exists \ell \in M\big[\mathcal{M}, g_m^x, \ell \Vdash \textstyle\bigwedge_{k=0}^{n} \big(c_k \leftrightarrow \Box(x \to c_k)\big) \;\&\; \mathcal{M}, g_m^x, \ell \Vdash \vartheta^t\big] \\
&\overset{(4)}{\Leftrightarrow}\ \exists \ell \in M\big[\forall k \leqslant n\big(\ell \in \mathrm{Cl}_k \Leftrightarrow m \in \mathrm{Cl}_k\big) \;\&\; \mathcal{M}, g_m^x, \ell \Vdash \vartheta^t\big] \\
&\overset{(5)}{\Leftrightarrow}\ \exists \ell \in M^{\triangleleft}[mR^{\triangleleft}\ell \;\&\; \mathcal{M}, g_m^x, \ell \Vdash \vartheta^t] \\
&\overset{(6)}{\Leftrightarrow}\ \exists \ell \in M^{\triangleleft}[mR^{\triangleleft}\ell \;\&\; \mathcal{M}, g, \ell \Vdash \vartheta^t] \\
&\overset{(7)}{\Leftrightarrow}\ \exists \ell \in M^{\triangleleft}[mR^{\triangleleft}\ell \;\&\; \mathcal{M}^{\triangleleft}, g^{\triangleleft}, \ell \Vdash \vartheta] \\
&\overset{(8)}{\Leftrightarrow}\ \mathcal{M}^{\triangleleft}, g^{\triangleleft}, m \Vdash \Diamond\vartheta
\end{aligned}$$

These equivalences are justified as follows.

(1) Definition of $(\cdot)^t$.
(2) Definition of satisfaction for $\downarrow$.
(3) Definition of satisfaction for $\wedge$ and $\Diamond$.
(4) Definition of satisfaction for atoms, Boolean operators, and $\Box$.
(5) Construction of $M^{\triangleleft}$, $V^{\triangleleft}$.
(6) Since x is bound in ϑ^t.
(7) Induction hypothesis.
(8) Definition of satisfaction for $\Diamond$.

This ends the proof. ❏

Lemma 5.27 $\mathcal{HL}(\Diamond, \exists)$-ER-SAT *is in* NEXPTIME.

Proof. The $\exists$ binder can bind state variables to states that are not accessible using $\Diamond$. In this case, the bound variable evaluates to false. Therefore, if $\varphi \in \mathcal{HL}(\Diamond, \exists)$-ER-SAT and $\mathcal{M}, g_0, m_0 \Vdash \varphi$, we can modify $\mathcal{M} = (M, R, V)$ into a model $\mathcal{M}' = (M', R', V')$ as follows. Let C be the cluster that contains m_0, and let s be a new state outside of M.

$$
\begin{aligned}
M' &= C \uplus \{s\} \\
R' &= R{\restriction_{M'}} \\
V'(p) &= V(p) \cap M', && \text{for } p \in \text{PROP} \\
V'(i) &= \begin{cases} V(i) & \text{if } V(i) \subseteq M', \\ \{s\} & \text{otherwise,} \end{cases} && \text{for } i \in \text{NOM}
\end{aligned}
$$

For each assignment g for $\mathcal{M}$, the corresponding assignment g' for $\mathcal{M}$ is obtained from g by binding all those variables to s that are bound to states outside of C by g. It is straightforward that $\mathcal{M}, g, m \Vdash \psi$ if and only if $\mathcal{M}', g', m \Vdash \psi$, for any state $m \in C$, any assignment g for $\mathcal{M}$, and any subformula ψ of φ (proof by induction). This implies $\mathcal{M}', (g_0)', m_0 \Vdash \varphi$.

Now, $\mathcal{M}'$ is a model with two clusters only. We can proceed as in the proof of Lemma 5.26 to construct an appropriate complete model. Thereby we obtain a reduction from $\mathcal{HL}(\Diamond, \exists)$-ER-SAT to $\mathcal{HL}(\Diamond, \downarrow)$-compl-SAT, where the latter is in NEXPTIME [MSSW05]. ❑

From Lemmata 5.25, 5.26, and 5.27 we obtain the complete characterisation of the satisfiability problems for hybrid logics with binders and without E.

Theorem 5.28 *Let* $X \in \big\{\{\Diamond, \downarrow\}, \{\Diamond, \exists\}, \{\Diamond, \downarrow, @\}\big\}$.
Then $\mathcal{HL}(X)$-ER-SAT *is* NEXPTIME*-complete.*

5.8.3 The full language

In contrast to the other binder languages, the complexity of $\mathcal{HL}(\Diamond, \downarrow, \mathsf{E})$-ER-SAT is one exponential level higher. The main reason for this property is the fact that small formulae can enforce satisfying models of doubly exponential size. We will show that it is possible, but not quite straightforward, to enforce a tiling in such big models, which establishes N2EXPTIME-hardness. On the other hand, we will prove that each satisfying model for an $\mathcal{HL}(\Diamond, \downarrow, \mathsf{E})$-formula φ can be restricted to a submodel of doubly exponential size that still satisfies φ. This will allow a guess-and-check procedure running in N2EXPTIME.

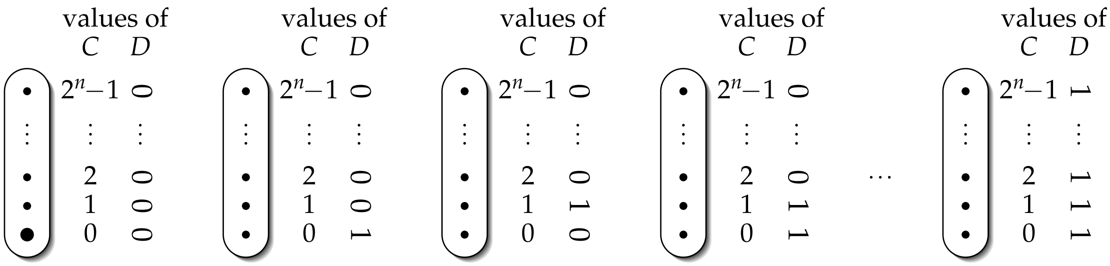


Figure 5.9: The behaviour of the counters C and D in an ER model

Lemma 5.29 *For each $n \in \mathbb{N}$ there is a formula $\varphi_n \in \mathcal{HL}(\Diamond, \downarrow, \mathsf{E})$ with the following properties.*

(i) $|\varphi_n| \in \mathcal{O}(n^2)$

(ii) $\varphi_n \in \mathcal{HL}(\Diamond, \downarrow, \mathsf{E})$-ER-SAT

(iii) Each satisfying ER model for φ_n has at least 2^{2^n} clusters with 2^n states each.

Proof. In order to enforce a model of the required size, we will proceed in two steps. In the first step, we will implement a counter C to take on values $0, \ldots, 2^n - 1$ within each cluster. This will make it possible, for each cluster, to distinguish 2^n states. The counter C will be realised by atomic propositions $c_{n-1}, \ldots, c_0$ whose truth values, in this order, constitute the binary representation of the value of C at the respective state. (The "truth value" of c_i at the state m is 1 if $m \in V(c_i)$, and 0 otherwise, as usual.)

In the second step we will implement a counter D that ranges over the values $0, \ldots, 2^{2^n} - 1$ and distinguishes 2^{2^n} clusters (not states). It will be realised by one atomic proposition d. Given a cluster X, the binary representation of the value of D at X will be determined by the truth values of d at the states in X, in the order given by their C-values. Such a doubly exponential counter has been used in [GLW06] to establish lower bounds on the size of certain concepts in description logics.

The required behaviour of C and D in a satisfying model for φ_n is visualised in Figure 5.9, where points and "sausages" represent states and clusters, respectively. The values of C and D in each state are displayed next to it. In the case of C, the shown number determines the truth values of all c_i as described above, and in case of D the given number is the truth value of d. The respective value of the whole counter D becomes readable after turning the D column counterclockwise by 90 degrees. The state with $C = 0$ in the cluster with $D = 0$ shall be the state that satisfies φ_n. It is marked by a larger point.

All these enforcements, of course, will make heavy use of the $\downarrow$ operator combined with E. We will now show how to achieve the required behaviour of C and D. This will be via several formulae whose conjunction results in φ_n. We will start with the conjuncts enforcing that each cluster has exactly 2^n states among which every value of C between 0 and $2^n - 1$ occurs once. In order to keep notation short, we will introduce some abbreviations. First, we would like to refer to specific C-values directly, as follows.

$$(C = 0) \;=\; \neg c_0 \wedge \ldots \wedge \neg c_{n-1} \qquad\qquad (C \neq 2^n - 1) \;=\; \neg c_0 \vee \ldots \vee \neg c_{n-1}$$

Second, it will be necessary to express that, for some $x \in \mathrm{SVAR}$, the C-value at the current state equals one plus the C-value of the state to which x is bound. (Recall that $@_x\psi$ abbreviates $\mathsf{E}(x \wedge \psi)$.)

$$(C = C_x + 1) \;=\; \bigvee_{k=0}^{n-1} \Big[c_k \wedge @_x \neg c_k \;\wedge\; \bigwedge_{\ell=0}^{k-1} (\neg c_\ell \wedge @_x c_\ell) \;\wedge\; \bigwedge_{\ell=k+1}^{n-1} (c_\ell \leftrightarrow @_x c_\ell) \Big]$$

In addition, we will use analogous shortcuts $C \gtreqless C_x$ expressing that the C-value at the current state is less than, equals, or is greater than the C-value of the state to which x is bound. The following conjuncts enforce the required behaviour of each cluster with respect to C.

- At the state satisfying φ_n, $C = 0$ holds.

  $\mathsf{CZERO}_1 \;=\; (C = 0)$

- In each cluster there is a state with $C = 0$.

  $\mathsf{CZERO}_2 \;=\; \mathsf{A}\Diamond(C = 0)$

- Each cluster has at most one state of each C-value.

  $\mathsf{CUNIQUE} \;=\; \mathsf{A}{\downarrow}x.\Box\big((C = C_x) \rightarrow x\big)$

- For each state of C-value $c < 2^n - 1$, there is a state of C-value $c + 1$ in the same cluster.

  $\mathsf{CSUCC} \;=\; \mathsf{A}\big[(C \neq 2^n - 1) \rightarrow {\downarrow}x.\Diamond(C = C_x + 1)\big]$

We will now construct the part of φ_n that implements the counter D. This requires expressing that the value of D in the cluster of the current state equals one plus the value of D in the cluster of the state assigned to some state variable x. The appropriate macro proceeds as follows (see also Figure 5.10).

Name the current state y. Name the state in the x-Cluster with $\neg d$ and lowest possible C-value z. For the state in the y-Cluster with the same C-value as

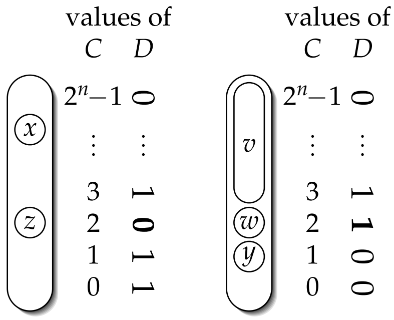


Figure 5.10: Incrementation of the D counter

z (which we call w only in this description and in the picture), require three things:

(a) d must hold at w;

(b) $\neg d$ must hold at all states of the y-Cluster with C-value less than the C-value of w;

(c) every state v of the y-Cluster with C-value greater than the C-value of w must agree in d with the states of the x-Cluster that have the same C-value as v.

$$
\begin{aligned}
(D = D_x + 1) \;=\; & {\downarrow} y.@_x\Box {\downarrow} z.\Big[\big(\neg d \wedge \Box((C < C_z) \to d)\big) \to \\
& \Big[@_y\Box\Big((C = C_z) \to \big[d \wedge \Box\big((C < C_z) \to \neg d\big) \\
& \qquad \wedge \Box\big((C > C_z) \to {\downarrow} v.@_x\Box((C = C_v) \to (d \leftrightarrow @_v d))\big)\big]\Big)\Big]\Big]
\end{aligned}
$$

We easily obtain the two remaining conjuncts for φ_n.

- The state satisfying φ_n belongs to a cluster with $D = 0$.

  $\mathsf{DZERO} \;=\; \Box\neg d$

- For each cluster X of D-value $d < 2^{2^n} - 1$, there is a cluster Y of D-value $d + 1$.

  $\mathsf{DSUCC} \;=\; \mathsf{A}{\downarrow} x.\big(\Diamond\neg d \to \mathsf{E}(D = D_x + 1)\big)$

Now let $\varphi_n \;=\; \mathsf{CZERO}_1 \wedge \mathsf{CZERO}_2 \wedge \mathsf{CUNIQUE} \wedge \mathsf{CSUCC} \wedge \mathsf{DZERO} \wedge \mathsf{DSUCC}$. Since each of the above abbreviations is of at most quadratic size and they do not occur nested in φ_n, Part (i) of the theorem is satisfied. For (ii), it is easy to see that the following model satisfies φ_n at the state $(0, 0)$ under any assignment.

Construct $\mathcal{M} = (M, R, V)$ as follows.

$$\begin{aligned} M &= \{(x,y) \mid x,y \in \mathbb{N};\ 0 \leqslant x < 2^{2^n};\ 0 \leqslant y < 2^n\} \\ R &= \{\big((x_1,y_1),(x_2,y_2)\big) \mid x_1 = x_2\} \\ V(c_i) &= \{(x,y) \mid \text{the } i\text{-th bit in the binary representation of } y \text{ is } 1\} \\ V(d) &= \{(x,y) \mid \text{the } y\text{-th bit in the binary representation of } x \text{ is } 1\} \end{aligned}$$

In order to show (iii), let $\mathcal{M} = (M, R, V)$ be an ER model with $m_{0,0} \in M$ and g be an assignment for $\mathcal{M}$ such that $\mathcal{M}, g, m_{0,0} \Vdash \varphi_n$. Now the four C-conjuncts enforce that $C = 0$ at $m_{0,0}$, and that each cluster of $\mathcal{M}$ contains exactly one state of C-value c for each $c = 0, \ldots, 2^n - 1$. Due to DZERO, the D-value of $m_{0,0}$'s cluster equals 0, and DSUCC successively enforces the existence of a cluster of D-value d for each $d = 0, \ldots, 2^{2^n} - 1$. (Note that the value of D in each cluster is uniquely determined by $V(d)$ and the order of the states of the cluster induced by their C-values.) Hence $\mathcal{M}$ has at least 2^{2^n} clusters with 2^n states each. ❑

Corollary 5.30 $\mathcal{HL}(\Diamond, \downarrow, \mathsf{E})$ *does not have the* $2^{\mathrm{poly}(n)}$*-size model property with respect to ER frames.*

Lemma 5.31 $\mathcal{HL}(\Diamond, \downarrow, \mathsf{E})$-ER-SAT *is* N2EXPTIME*-hard.*

Proof. We reduce the 2^{2^n}-tiling problem to $\mathcal{HL}(\Diamond, \downarrow, \mathsf{E})$-ER-SAT. The reduction uses the techniques enforcing doubly exponentially large satisfying models from the proof of Lemma 5.29. In order to encode a tiling for the $2^{2^n} \times 2^{2^n}$-square in an ER model $\mathcal{M}$, we will first enforce that $\mathcal{M}$ has $2^{2^{n+1}}$ clusters with 2^{n+1} states each, using the same construction of counters C and D, but with parameter $n+1$. The tiled square itself will be encoded in the states of C-value 0 of all clusters. Hence row 0 of the square will be in the clusters of D-values $0, \ldots, 2^{2^n} - 1$; row 1 will be in the clusters of D-values $2^{2^n}, \ldots, 2 \cdot 2^{2^n} - 1$; etc.; see Figure 5.11. The horizontal adjacencies in the original square can be expressed referring to pairs of clusters with successive D-values. In contrast, for the vertical adjacencies, pairs of clusters whose D-values differ by 2^{2^n} will have to be compared.[2]

For the required reduction, we will show how to transform an instance (T, n) of the tiling problem into a formula $\psi_{T,n}$ such that there is a T-tiling of the

[2] Note that this is not the standard way to encode tilings in models, insofar as not every state of the model corresponds to a position in the grid. However, this modification of the standard way is not new, since it relies on ideas developed by Chlebus in [Chl86] to encode the rectangle tiling problem with exponential parameter into a more intricate version of a bounded tiling problem that he called "High Tiling".

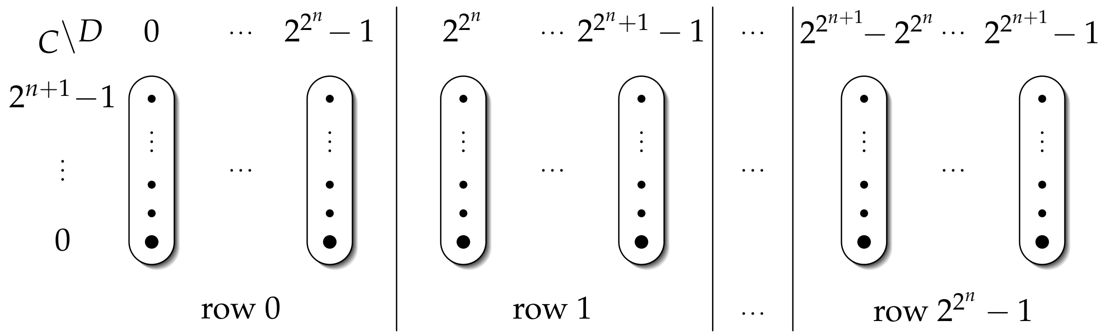


Figure 5.11: Enforcing a tiling in an ER model of doubly exponential size

$2^{2^n} \times 2^{2^n}$-square if and only if $\psi_{T,n}$ is satisfiable. As in the proof of Lemma 5.29, this formula will consist of several conjuncts. The first of them will be the formula φ_{n+1} from that proof, enforcing the required structure of the model. In order to keep the remaining conjuncts short, we will use the same abbreviations again, but with $n+1$ instead of n. Furthermore, $D = D_x + 2^{2^n}$ denotes that the D-value of the current state's cluster equals 2^{2^n} plus the D-value of the cluster containing the state to which x is bound. This abbreviation is defined analogously to the shortcut $D = D_x + 1$.

Now we are ready to give the conjuncts that enforce the tiling. Let T be a set of tile types. For each $t \in T$ we will use an atomic proposition t to denote that a tile of type t lies at the respective position.

- At each state with C-value 0 lies exactly one tile.

$$\mathsf{TILE} = \mathsf{A}\Big((C=0) \to \bigvee_{t \in T} \big(t \wedge \bigwedge_{\substack{t' \in T \\ t' \neq t}} \neg t\big)\Big)$$

- Tiles match horizontally.

$$\begin{aligned}\mathsf{HOR} = \mathsf{A}\Big[&\big((C=0) \wedge \Diamond(\neg c_n \wedge d)\big) \to \\ &\downarrow x.\Big(\bigwedge_{t \in T} \to \mathsf{A}\big(((C=0) \wedge (D = D_x + 1)) \to \bigvee_{t' \in \mathrm{RI}(t,T)} t'\big)\Big)\Big]\end{aligned}$$

(The $\Diamond$-subformula requires that the corresponding position of the current state does not belong to the last column of the square.)

- Tiles match vertically.

$$\begin{aligned}\mathsf{VER} = \mathsf{A}\Big[&\big((C=0) \wedge \Diamond(c_n \wedge \neg d)\big) \to \\ &\downarrow x.\Big(\bigwedge_{t \in T} \to \mathsf{A}\big(((C=0) \wedge (D = D_x + 2^{2^n})) \to \bigvee_{t' \in \mathrm{UP}(t,T)} t'\big)\Big)\Big]\end{aligned}$$

- The borders of the square are white.

$$\mathsf{WHITE} \;=\; \mathsf{A}\Big[\Big(\Box(c_n \to \neg d) \to \bigvee_{\substack{t\in T\\ \mathrm{bot}(t)=\mathrm{white}}} t\Big) \wedge \Big(\Box(c_n \to d) \to \bigvee_{\substack{t\in T\\ \mathrm{top}(t)=\mathrm{white}}} t\Big)$$
$$\wedge \Big(\Box(\neg c_n \to \neg d) \to \bigvee_{\substack{t\in T\\ \mathrm{left}(t)=\mathrm{white}}} t\Big) \wedge \Big(\Box(\neg c_n \to d) \to \bigvee_{\substack{t\in T\\ \mathrm{right}(t)=\mathrm{white}}} t\Big)\Big]$$

Now let $\psi_{T,n} = \varphi_{n+1} \wedge \mathsf{TILE} \wedge \mathsf{HOR} \wedge \mathsf{VER} \wedge \mathsf{WHITE}$. Each conjunct is of size at most $\mathcal{O}(n^2 + |T|^2)$. From their definitions it is clear that $\psi_{T,n}$ can be computed in time polynomial in $n + |T|$. It remains to show that there is a T-tiling of the $2^{2^n} \times 2^{2^n}$-square if and only if $\psi_{T,n} \in \mathcal{HL}(\Diamond, \downarrow, \mathsf{E})$-ER-SAT.

"$\Rightarrow$". Suppose there is a tiling τ for the $2^{2^n} \times 2^{2^n}$ square. We construct a model $\mathcal{M} = (M, R, V)$ for $\psi_{T,n}$ as follows.

$$\begin{aligned}
M &= \{(x,y) \mid x,y \in \mathbb{N};\ 0 \leqslant x < 2^{2^{n+1}};\ 0 \leqslant y < 2^{n+1}\}\\
R &= \{\big((x_1,y_1),(x_2,y_2)\big) \mid x_1 = x_2\}\\
V(c_i) &= \{(x,y) \mid \text{the } i\text{-th bit in the binary representation of } y \text{ is } 1\}\\
V(d) &= \{(x,y) \mid \text{the } y\text{-th bit in the binary representation of } x \text{ is } 1\}\\
V(t) &= \{(2^{2^n}\!\cdot i + j, 0) \mid 0 \leqslant i,j < 2^{2^n};\ \tau(i,j) = t\}, \qquad \text{for } t \in T
\end{aligned}$$

Now it is easy to see that $\mathcal{M}, g, (0,0) \Vdash \psi_{T,n}$ for any assignment g: The first conjunct, φ_{n+1}, is treated in the proof of Lemma 5.29 (ii). The remaining conjuncts hold at $(0,0)$ due to the definition of V, the fact that τ is a function, and the tiling conditions.

"$\Leftarrow$". Suppose $\psi_{T,n} \in \mathcal{HL}(\Diamond, \downarrow, \mathsf{E})$-ER-SAT. Then there exist a model $\mathcal{M} = (M, R, V)$, an assignment g for $\mathcal{M}$, and a state $m_{0,0} \in M$ such that $\mathcal{M}, g, m_{0,0} \Vdash \psi_{T,n}$. Due to the conjunct φ_{n+1} of $\psi_{T,n}$, consulting the proof of Lemma 5.29 (iii) shows that for every $x < 2^{2^{n+1}}$ and every $y < 2^{n+1}$, there are clusters Cl_x with states $m_{x,y} \in \mathrm{Cl}_x$ such that C has value y in each $m_{x,y}$, and D has value x in each Cl_x. This allows for constructing a tiling τ from the states $m_{x,0}$ via

$$\tau(i,j) = t \;\Leftrightarrow\; m_{x,0} \in V(t) \quad (\text{for } x = 2^{2^n} \cdot i + j).$$

The correctness of this definition is ensured by the conjunct TILE. Due to the remaining conjuncts, τ defines a permissible tiling. ❑

We will now establish the corresponding upper bound, showing that the full hybrid language has a doubly exponential size model property over ER frames. This will make it possible to decide satisfiability using a straightforward guess-and-check procedure and involving results for model checking.

Lemma 5.32 *$\mathcal{HL}(\Diamond, \downarrow, \mathsf{E})$ has the $2^{2^{2n+2}}$-size model property with respect to ER frames.*

Proof. Intuitively, the proof relies on the following considerations: Call the set of propositional variables and nominals that hold at a given state of a model the *type* of this state. Let the *C-type* of a cluster be the set of types of all points of this cluster. If there was no $\downarrow$ in our language, then two states of the same type that belong to the same cluster would not be distinguishable, that is, they would satisfy the same formulae. Even two states of the same type that belong to two different clusters of the same C-type would not be distinguishable. This would enable us to restrict clusters to at most one state per possible type and to restrict a whole satisfying model for some formula φ to at most one cluster per possible C-type without affecting satisfiability of φ.

In the presence of the $\downarrow$ binder, this argumentation must be refined and requires a certain amount of technical details. Let φ be a formula of size n and $\mathcal{M} = (M, R, V)$ be a satisfying model for φ. First, there are at most 2^n possible types of states. Since an assignment for $\mathcal{M}$ might bind all state variables occurring in φ to different states of the same type, only up to $n+1$ states of the same type belonging to the same cluster are distinguishable. Hence, it is legitimate to restrict each cluster of $\mathcal{M}$ to at most $n+1$ states of each type in the first step, which leads to an exponential bound in the size of clusters.

In the second step, we modify the notion of a C-type of a cluster X to be the multiset containing as many copies of each type as there are states of this type in X, but not more than $n+1$. It is legitimate, too, to restrict the whole model to at most $n+1$ clusters of each C-type. Since there are at most $(n+2)^{2^n}$ many different C-types, the number of clusters — and, hence, states — of the restricted model is bounded by $2^{2^{\mathcal{O}(n)}}$.

The formal proof of the $2^{2^{2n+2}}$-size model property requires quite some notation. Let $\varphi \in \mathcal{HL}(\Diamond, \downarrow, \mathsf{E})$-ER-SAT be of size n. Then there exist an ER model $\mathcal{M} = (M, R, V)$, an assignment g_0 for $\mathcal{M}$, and a state $m_0 \in \mathcal{M}$ such that $\mathcal{M}, g_0, m_0 \Vdash \varphi$. Let $C_i \subseteq M$, $i \in I$, be all clusters of $\mathcal{M}$, for an appropriate index set I that contains 0, such that $m_0 \in C_0$. Let $x_1, \ldots, x_s$ be all state variables occurring in φ. Analogously, let $a_1, \ldots, a_t$ be all other atoms in φ. Clearly $s, t \leqslant n$. A *φ-type* is a subset of $\{a_1, \ldots, a_t\}$. Let $A_1, \ldots, A_{2^t}$ be an enumeration of all φ-types, such that m_0 is of type A_1. (A state m is of type A_ℓ iff for each $j = 1, \ldots, t$: $(m \in V(a_j) \Leftrightarrow a_j \in A_\ell)$. Furthermore, we will deliberately speak of "(C-)types" instead of "φ-(C-)types" whenever no confusion may arise.) Given a cluster C, we divide it into 2^t "type layers" $C_i^\ell = \{m \in C_i \mid m \text{ is of type } A_\ell\}$, as shown in Figure 5.12.

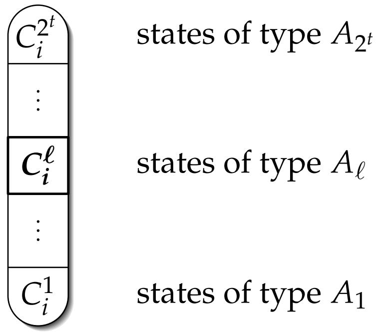


Figure 5.12: Dividing a cluster into "type layers"

We define a function $f : I \times \{1, \ldots, 2^t\} \to \mathfrak{P}(M)$ that assigns a set of states to each pair (i, ℓ) of a cluster number i and a type number ℓ, such that $f(i, \ell)$ is a subset of C_i. The union of all possible $f(i, \ell)$ will constitute the first restriction of $\mathcal{M}$. The function f is defined as follows, where $\#C_i^\ell$ denotes the number of states in C_i^ℓ. If $\#C_i^\ell \leqslant s + 1$, then $f(i, \ell) = C_i^\ell$. Otherwise, $f(i, \ell)$ is some subset of C_i^ℓ of size at most $s + 1$ that satisfies the following conditions.

(i) For each $j = 1, \ldots, s$: if $g_0(x_j) \in C_i^\ell$, then $g_0(x_j) \in f(i, \ell)$.

(ii) $m_0 \in f(0, 1)$.

Such a subset always exists. For any cluster C_i, let $f(C_i)$ denote the union of all $f(i, \ell)$. Due to the definition of f, $f(C_i) \subseteq C_i$, and $f(C_i)$ has at most $(s + 1) \cdot 2^t$ states. We denote the union of all $f(C_i)$ by M'.

After restricting the cluster size, we will restrict the number of the clusters. Let $\mathcal{A}$ be the multiset containing $s + 1$ copies of each type A_ℓ. Call each subset of $\mathcal{A}$ a *φ-C-type*. The power set $\mathfrak{P}(\mathcal{A})$ contains $(s + 2)^{2^t}$ elements. Let $\mathcal{A}_1, \ldots, \mathcal{A}_{(s+2)^{2^t}}$ be an enumeration of all φ-C-types, such that $f(C_0)$ is of C-type $\mathcal{A}_1$. (The C-type of a cluster C_i is determined by the number of states of each type in its restriction $f(C_i)$.) We divide M' into $(s + 2)^{2^t}$ "C-type layers" $\mathcal{C}_\ell$ being the union of $f(C_i)$ for all C_i of C-type $\mathcal{A}_\ell$.

Now define a second choice function $f' : \{1, \ldots, (s + 2)^{2^t}\} \to \mathfrak{P}(M')$ that assigns a set of states to each C-type number such that $f'(\ell)$ is a union of (restricted) clusters. The union of all possible $f'(\ell)$ will constitute the second restriction of $\mathcal{M}$. The function f' is defined as follows. If there are not more than $s + 1$ clusters of C-type $\mathcal{A}_\ell$, then $f'(\ell) = \mathcal{C}_\ell$. Otherwise, $f'(\ell)$ is the union of $s + 1$ restricted clusters of type $\mathcal{A}_\ell$ satisfying

(i) For each $j = 1, \ldots, s$:
if $g_0(x_j) \in f(C_i)$ for some C_i of type $\mathcal{A}_\ell$, then $f(C_i) \subseteq f'(\ell)$.

(ii) $f(C_0) \subseteq f'(1)$.

Such a set always exists. Due to the definition of f', each $f'(\ell)$ contains at most $s+1$ restricted clusters and, hence, $(s+1)^2 \cdot 2^t$ states. We now construct a new model $\mathcal{M}'' = (M'', R'', V'')$ from $\mathcal{M}$, where M'' is the union of $f'(\ell)$ for all C-types $\mathcal{A}_\ell$, and R'' and V'' are the restrictions of R and V to M''. Now the following facts about $\mathcal{M}''$ are obvious. It is still an ER model, whose clusters are restrictions of clusters of $\mathcal{M}$. It contains m_0, because $m_0 \in f(C_0) \subseteq f'(1)$. The assignment g_0 is an assignment for $\mathcal{M}''$. Since there are $(s+2)^{2^t}$ C-types, M'' contains $(s+2)^{2^t} \cdot (s+1)^2 \cdot 2^t$ states. This number is limited by $2^{2^{2n+2}}$ because $s, t \leqslant n$.

It remains to show that $\mathcal{M}'', g_0, m_0 \Vdash \varphi$. For this purpose, we make use of an auxiliary statement. This statement uses the concept of agreement in a pair of assignments. We say that two states m and m' from $\mathcal{M}$ agree in two assignments g/g' for $\mathcal{M}$ iff $\{x_k \mid g(x_k) = m\} = \{x_k \mid g'(x_k) = m'\}$. Two clusters C_i and $C_{i'}$ agree in g/g' iff they are of the same C-type, and for each A_ℓ, each $m \in C_i^\ell$, there is some $m' \in C_{i'}^\ell$ that agrees with m in g/g'.

Claim 1. For each subformula ψ of φ; for each two assignments g, g' for $\mathcal{M}$; for each C-type $\mathcal{A}_\ell$; for each two clusters C_i and $C_{i'}$ that agree in g/g'; for each type A_ℓ; and for each $m \in C_i^\ell$ and $m' \in C_{i'}^\ell$ that agree in g/g'; it holds that $\mathcal{M}, g, m \Vdash \psi$ iff $\mathcal{M}, g', m' \Vdash \psi$.

Proof of Claim 1. By induction on ψ. Direction "$\Rightarrow$" suffices because of the symmetry of the conditions on m and m'. The atomic and Boolean cases of the induction are immediate and easy, respectively. The E case is trivial, and the $\downarrow$ case is straightforward if one considers the fact that, since m and m' agree in g/g', they also agree in $g^x_m/(g')^x_{m'}$ for any state variable x. The only interesting case is the $\Diamond$ case, with the following argumentation. Suppose $\mathcal{M}, g, m \Vdash \Diamond\vartheta$. Then there exists some $\overline{m} \in C_i$ with $\mathcal{M}, g, \overline{m} \Vdash \vartheta$. Let $A_{\ell'}$ be the φ-type of $\overline{m}$. Then $C_i^{\ell'}$ and, hence, $C_{i'}^{\ell'}$ is not empty. Because C_i and $C_{i'}$ agree in g/g', there is some $\overline{m}' \in C_{i'}^{\ell'}$ that agrees with $\overline{m}$ in g'/g. Due to the induction hypothesis, $\mathcal{M}, g', \overline{m}' \Vdash \vartheta$. Hence, $\mathcal{M}, g', m' \Vdash \Diamond\vartheta$.

Now the required fact $\mathcal{M}'', g_0, m_0 \Vdash \varphi$ is a consequence of the following claim.

Claim 2. For each subformula ψ of φ, for each $m \in M''$, for each assignment g for $\mathcal{M}''$, it holds that $\mathcal{M}, g, m \Vdash \psi$ iff $\mathcal{M}'', g, m \Vdash \psi$.

Proof of Claim 2. Since $m \in M''$, there is some $i \in I$ such that $m \in f(C_i) \subseteq M''$. Let $\mathcal{A}_\ell$ be the C-type of C_i. We prove the claim by induction. The atomic cases follow from the facts that $\mathcal{M}''$ is a restriction of $\mathcal{M}$ and that g is an assignment for both $\mathcal{M}$ and $\mathcal{M}''$. The Boolean cases are straightforward. So is the $\downarrow$ case if one considers the fact that g^x_m is still an assignment for $\mathcal{M}''$. For the remaining cases for $\Diamond$ and E, the "$\Leftarrow$" direction is trivial. We will only prove the "$\Rightarrow$" direction.

Case $\psi = \Diamond\vartheta$. Suppose $\mathcal{M}, g, m \Vdash \Diamond\vartheta$. Then there exists some $m' \in C_i$ with $\mathcal{M}, g, m' \Vdash \vartheta$. Let the type of m' be A_k. There are three cases to distinguish.

(1) $\#C_i^k \leqslant s+1$. Then m' belongs to $f(i,k)$ and, hence, to $f(C_i)$. Hence $m' \in M''$ and $mR''m'$. Together with the induction hypothesis, this immediately yields $\mathcal{M}'', g, m \Vdash \Diamond\vartheta$.

(2) $\#C_i^k > s+1$ and, for some $j = 1, \ldots, s$, $g(x_j) = m'$. Since g is *for* $\mathcal{M}''$, we obtain $m' \in M''$ and $mR''m'$, which yields $\mathcal{M}'', g, m \Vdash \Diamond\vartheta$ as in case (1).

(3) $\#C_i^k > s+1$ and, for no $j = 1, \ldots, s$, $g(x_j) = m'$. Due to the size of C_i^k and the construction of f, there is some $m'' \in f(i,k)$ not affected by g either. Since m' and m'' are of the same type and agree in g/g, Claim 1 implies that $\mathcal{M}, g, m'' \Vdash \vartheta$. The remaining argumentation is the same as in case (1), with m'' instead of m'.

Case $\psi = \mathsf{E}\vartheta$. Suppose $\mathcal{M}, g, m \Vdash \mathsf{E}\vartheta$. Then there exists some $m' \in M$ with $\mathcal{M}, g, m' \Vdash \vartheta$. Let the type of m' be A_k, and let m' be from $C_{i'}$, the latter being of C-type $\mathcal{A}_\ell$. As in the $\Diamond$ case, there are three subcases to distinguish.

(1) There are at most $s+1$ clusters of C-type $\mathcal{A}_\ell$. Then C_i and, hence, m' belong to $\mathcal{M}''$. Together with the induction hypothesis, this immediately yields $\mathcal{M}'', g, m \Vdash \mathsf{E}\vartheta$.

(2) There are more than $s+1$ clusters of C-type $\mathcal{A}_\ell$ and, for some $j = 1, \ldots, s$, $g(x_j) \in f(C_{i'})$. Since g is *for* $\mathcal{M}''$, we obtain $f(C_{i'}) \subseteq M''$, which yields $\mathcal{M}'', g, m \Vdash \mathsf{E}\vartheta$ as in case (1).

(3) There are more than $s+1$ clusters of C-type $\mathcal{A}_\ell$ and, for no $j = 1, \ldots, s$, $g(x_j) \in f(C_{i'})$. Due to the "large enough" number of clusters of C-type $\mathcal{A}_\ell$ and the construction of f', there is some cluster $C_{i''} \subseteq f'(\ell)$ not affected by g either. Since $C_{i'}$ and $C_{i''}$ agree in g/g, there is some $m'' \in f(C_{i''})$, having the same type as m' and agreeing with m' in g/g. Hence, due to Claim 1, $\mathcal{M}, g, m'' \Vdash \vartheta$. Here we have to distinguish the same three subcases as in the $\Diamond$ case. The argumentation is analogous and leads to the required result $\mathcal{M}'', g, m \Vdash \mathsf{E}\vartheta$. ❑

Theorem 5.33 *$\mathcal{HL}(\Diamond, \downarrow, \mathsf{E})$-ER-SAT is* N2EXPTIME*-complete.*

Proof. The lower bound follows from Lemma 5.31. For the upper bound, let φ be an arbitrary instance of $\mathcal{HL}(\Diamond, \downarrow, \mathsf{E})$-ER-SAT. In order to determine whether $\varphi \in \mathcal{HL}(\Diamond, \downarrow, \mathsf{E})$-ER-SAT, we guess a model $\mathcal{M} = (M, R, V)$, an assignment g, and a state $m \in M$, and check whether $\mathcal{M}, g, m \Vdash \varphi$. Let $n = |\varphi|$. If $\varphi \in \mathcal{HL}(\Diamond, \downarrow, \mathsf{E})$-ER-SAT, then, due to Lemma 5.32, it has a satisfying model with state space M of size at most $2^{2^{2n+2}}$. Hence, in time $\mathcal{O}(2^{2^{2n+2}})$ we can guess a model $\mathcal{M} = (M, R, V)$ of size at most $2^{2^{2n+2}}$ and check whether R is an equivalence relation. An assignment g can be guessed in time $\mathcal{O}(2^{3n+2})$. All the guesses together take time $\mathcal{O}(2^{2^{k' \cdot n}})$ for a constant k'.

Finally, checking whether there exists a state m such that $\mathcal{M}, g, m \Vdash \varphi$ can be accomplished using the procedure `MCFULL` from [FdR06]. By [FdR06, Theorem 4.5] this takes time $\mathcal{O}(|\varphi| \cdot (|M| + |R|) \cdot |M|^k) = \mathcal{O}(n \cdot (2^{2^{2n+2}} + (2^{2^{2n+2}})^2) \cdot (2^{2^{2n+2}})^k) = \mathcal{O}(2^{2^{k'' \cdot n}})$ for an appropriate constant k''. Altogether, we have a nondeterministic algorithm that runs in doubly exponential time. ❑

5.8.4 Pure languages with binders

Satisfiability for all pure languages with binders is PSPACE-complete. Hardness is due to an easy reduction from QSAT similar to that for the model checking problem in Lemma 4.1. The upper bound will use a polynomial-size model property obtained in a similar manner as the $2^{2^{2n+2}}$-size model property for $\mathcal{HL}(\Diamond, \downarrow, \mathsf{E})$ in Lemma 5.32. Note the following subtle difference in reasoning. While the $2^{2^{2n+2}}$-size model property of $\mathcal{HL}(\Diamond, \downarrow, \mathsf{E})$ implies an N2EXPTIME upper bound for satisfiability, the polynomial-size model property of any binder language does not imply an NP upper bound for satisfiability. The reason becomes clear if we recall the complexity results for model checking over arbitrary frames from [FdR06]: In the presence of binders, this problem is PSPACE-complete, but an upper time bound is $\mathcal{O}(|\varphi| \cdot |M|^{2|\varphi|})$. If the model is large compared to the formula, as in the case of $\mathcal{HL}(\Diamond, \downarrow, \mathsf{E})$, then the factor $|\varphi|$ in the exponent is unimportant. In the case of a polynomial-size model property, however, the upper time bound for model checking only yields an exponential time bound for the whole guess-and-check algorithm deciding satisfiability.

Theorem 5.34 *Let* $X \in \{\{\Diamond, \downarrow\}, \{\Diamond, \downarrow, @\}, \{\Diamond, \exists\}, \{\Diamond, \downarrow, \mathsf{E}\}\}$.
Then $\mathcal{PML}(X)$-ER-SAT *and* $\mathcal{PHL}(X)$-ER-SAT *are* PSPACE*-complete.*

Proof. For PSPACE-hardness, we reduce $\mathcal{PML}(\Diamond, \downarrow)$-compl-MC (see Lemma 4.1) to $\mathcal{PML}(\Diamond, \downarrow)$-compl-SAT. (Note that the problems $\mathcal{PML}(\Diamond, \downarrow)$-compl-SAT and $\mathcal{PML}(\Diamond, \downarrow)$-ER-SAT are identical, because the truth of $\Diamond$-$\downarrow$-formulae is preserved under taking generated submodels.) Let $(\mathcal{M}, g, \varphi)$ be an instance of the model-checking problem, consisting of a complete model $\mathcal{M}$ having n states, an assignment g, and a formula φ. Since the reduction in the proof of Lemma 4.1 produces sentences only, we may assume w.l.o.g. that φ is a sentence. From this instance, we construct a formula $\varphi' = f(\mathcal{M}, g, \varphi)$ that uses fresh state variables $x_1, \ldots, x_n$ to enforce that each satisfying model has exactly n states (i.e., equals $\mathcal{M}$). This is achieved by the following polynomial-time computable reduction function.

$$f(\mathcal{M}, g, \varphi) = \downarrow x_1.\Diamond \downarrow x_2. \cdots \Diamond \downarrow x_n.\Big(\bigwedge_{\substack{i,j=1,\ldots,n \\ i \neq j}} \Box(x_i \to \neg x_j) \wedge \Box \bigvee_{i=1,\ldots,n} x_i \Big) \wedge \varphi$$

Now it is straightforward to show that $(\mathcal{M}, g, \varphi) \in \mathcal{PML}(\Diamond, \downarrow)$-compl-MC if and only if $\varphi' \in \mathcal{PML}(\Diamond, \downarrow)$-compl-SAT. Suppose $\mathcal{M}, g, m \Vdash \varphi$, for some state m. Then $\mathcal{M}, g, m \Vdash \varphi'$. For the converse, suppose $\mathcal{M}', g', m \Vdash \varphi'$, for some model $\mathcal{M}'$ and some assignment g' for $\mathcal{M}'$. Since the first part of φ' enforces $\mathcal{M}'$ to have n states, $\mathcal{M}$ and $\mathcal{M}'$ are identical. Since φ is a sentence, it follows that $\mathcal{M}, g, m \Vdash \varphi$.

Membership in PSPACE follows from the $\mathcal{O}(n^2)$-size model property of the logic $\mathcal{PHL}(\Diamond, \downarrow, \mathsf{E})$ with respect to ER frames. The proof of this property is analogous to the proof of Lemma 5.32, but with one fundamental difference. Since our language is pure, the number of types decreases to one. Hence, in each cluster, at most $n+1$ different states can be distinguished by means of state variables. This means that there are only $n+1$ C-types (representing clusters with $1, 2, \ldots, n+1$ states), and, again, only $n+1$ clusters of each C-type can be distinguished. This leads to a $(n+1)^2$-size model property. The technical details are essentially the same as in the proof of Lemma 5.32.

Now a model can be guessed in polynomial time and checked in polynomial space (Lemma 4.1). Since NP $\subseteq$ PSPACE, the upper bound follows. ❑

Chapter 6

Satisfiability of Multi-Modal Downarrow Logic

6.1 Introduction

The quintessence of this chapter can be informally expressed by the following warning.

Warning 6.1 *When hybridising a multi-modal logic $\mathcal{L}_n$, expect it to become undecidable — even if only frame classes are considered over which the uni-modal hybridised $\mathcal{L}$ is decidable.*

We will begin with an explanation and a more precise formulation of this statement.

This chapter examines the effects of the interaction between the hybrid $\downarrow$ operator and multiple modalities on the decidability of the satisfiability problem of hybrid languages. In the previous chapters, we have seen that $\downarrow$ is a very powerful means of expression, which is dangerous in terms of computational costs. In general, $\mathcal{HL}(\Diamond, \downarrow)$ is undecidable [ABM99]. However, over restricted frame classes, such as transitive frames, transitive trees, linear orders, ER frames, or complete frames, $\downarrow$ is either of no use at all, or the expressive power added does not lead to undecidability (see Chapter 5). We will show that for these and other frame classes, satisfiability becomes undecidable in the bi-modal case.

Table 6.1 summarises complexity results for satisfiability of the smallest binder language, namely $\mathcal{HL}(\Diamond, \downarrow)$, over different frame classes. It contains the frame classes examined in Chapter 5 and classes of frames of bounded width (see [tCF05b]). All names of complexity classes stand for completeness results.

The contribution of this chapter is to consider extensions of $\mathcal{ML}$ along two axes — allowing for multiple modalities and adding the $\downarrow$ operator — , which by themselves are benign in terms of decidability over restricted frame classes,

arbitrary frames	transitive frames	transitive trees	linear orders	equivalence relations	width 0, 1	finite width $\geqslant 2$
coRE [1]	NExpTime [4]	PSpace [2, c]	NP [2]	NExpTime [4]	NP [3]	NExpTime [3]

Legend.	[1]	[ABM99]	[3]	[tCF05b]	c.	conclusion
	[2]	[FdRS03]	[4]	[MSSW05]		

Table 6.1: Complexity results for the hybrid downarrow language with respect to different frame classes

and to point out that their combination does not necessarily behave benignly, too. This is the intended meaning of Warning 6.1.

Precisely speaking, we prove the following results.

(1) For each frame class containing one particular linear frame, satisfiability of the bi-modal $\downarrow$-language is undecidable.

(2) For each frame class containing one particular ER frame, satisfiability of the bi-modal $\downarrow$-language is undecidable.

These results are given in Table 6.2 (in bold face type), together with known results for hybrid languages extended along only one of the above mentioned axes.

It is worth noting that each of these two statements involves a wide range of frame classes, including temporally (in the first case) and epistemically (in the second case) relevant ones. This is remarkable because it is not always possible that techniques used to establish complexity results for modal or hybrid logics apply (or are transferable) to different frame classes. For instance, complexity results for hybrid languages have been proven in the literature either over single frame classes [ABM99, FdRS03, MS07a] or separately by frame classes [ABM00, MSSW05]. However, many positive examples of complexity or (un)decidability results for satisfiability of modal logics involving more than one frame class can be found, for instance, in [Lad77, Spa93a, Wol96, Wol97, HHK02, GKWZ05, LW05, HR07].

In the presence of $\downarrow$, it is possible to achieve general results because of the enormous expressive power of this operator: We will see that there are $\downarrow$-formulae enforcing that each satisfying model has very specific properties — it is based on $\mathcal{F}$, where $\mathcal{F}$ is one of the frames mentioned above in (1) and (2). Since each $\mathcal{F}$ belongs to a very restricted frame class, a single reduction from the classical tiling problem to $\mathcal{HL}(\Diamond,\downarrow)$-$\{\mathcal{F}\}$-SAT can easily be extended to cover $\mathcal{HL}(\Diamond,\downarrow)$-$\mathfrak{F}$-SAT for each class $\mathfrak{F}$ of frames containing $\mathcal{F}$.

hybrid lang.	arbitrary frames	transitive frames	linear orders	ER frames
$\mathcal{HL}(\Diamond)$	PSPACE [1]	PSPACE [2]	NP [2]	NP [5, c.]
$\mathcal{HL}_n(\Diamond)$	PSPACE [1]	PSPACE [2]	NP-hard [7]	PSPACE [4, c.]
$\mathcal{HL}(\Diamond,\downarrow)$	coRE [1]	NEXPTIME [6]	NP [3]	NEXPTIME [6]
$\mathcal{HL}_n(\Diamond,\downarrow)$	coRE [1, c.]	**coRE** (6.2)	**coRE** (6.2)	**coRE** (6.4)

Legend.						
	[1]	[ABM99]	[5]	[Lad77]	c.	conclusion from surrounding results
	[2]	[ABM00]	[6]	[MSSW05]		
	[3]	[FdRS03]	[7]	[ON80]	(a.b.c)	Theorem (a.b.c)
	[4]	[HM92]				

Table 6.2: An overview of complexity results for multi-modal hybrid logics

Result (2) is of relevance for epistemic applications of hybrid logic, because the class of all multi-modal ER frames and its superclasses are important for modelling knowledge and belief of multi-agent scenarios, see also Section 1.2.

In contrast, we cannot accredit such a practical relevance to Result (1). Although some of the covered classes of frames are important for temporal applications in the uni-modal setting, it is hard to see the use of frames with several accessibility relations that are, say, transitive trees or linear orders. However, Result (1) is of the same *theoretical* interest as Result (2) in that they both demonstrate the "dangerous" behaviour of the combination of $\downarrow$ and multiple modalities described above.

Before we prove the announced results, we will try to provide an intuition why the two extensions of the minimal hybrid language lead to undecidability when combined. For this purpose, it is helpful to understand what makes $\mathcal{HL}(\Diamond,\downarrow)$ behave "benignly" over restricted frame classes.

- In models based on acyclic frames (linear orders or transitive trees), states named by $\downarrow$ can never be reached again. Hence $\downarrow$ is useless. In contrast, in a frame with two acyclic accessibility relations, cycles are possible.
- Over transitive frames, the proof of NEXPTIME-membership for satisfiability of $\mathcal{HL}(\Diamond,\downarrow)$ relies on the fact that each cycle is a cluster, that is, a complete subframe. But in a transitive frame for a bi-modal language, there can be cycles consisting of edges of different accessibility relations which are not necessarily clusters.
- Over ER frames, $\mathcal{HL}(\Diamond,\downarrow)$ is equivalent to a fragment of first-order logic (the monadic class). This equivalence cannot be established for the bi-modal language.

Hence, while for some frame classes proof techniques for the uni-modal minimal hybrid language can straightforwardly be extended to its multi-modal version, this is not possible in the presence of $\downarrow$.

Furthermore, the natural reduction from multi-modal to uni-modal logic, which implies that, in general, satisfiability for the former cannot be harder than for the latter, does not work on transitive frames. This reduction is described in [KW97] and goes back to S. K. Thomason [Tho74]. For each modality $\Diamond_i$, it introduces a new atomic proposition p_i and replaces each edge mR_in by a sequence $mR\ell^i_{m,n}Rn$, where $\ell^i_{m,n}$ is a new state in which p_i is true. This replacement transforms every multi-modal model into a uni-modal one. Hence, replacing each occurrence of $\Diamond_i\psi$ in a given formula φ by $\Diamond(p_i \wedge \Diamond\psi)$ preserves satisfiability of φ. This technique fails in the case of transitive frames or a subclass thereof, because the addition of the new states destroys transitivity. If we require the new model to be based on a transitive frame again, we cannot just take the transitive closure. This would add extra accessibilities that were not present in the original model.

The bi-modal language with $\downarrow$ is in fact strong enough to encode tilings (see Section 2.4.4) on any of the frame classes covered by Results (1) and (2). Tilings have been used in the literature to establish undecidability for satisfiability of different hybrid $\downarrow$ languages. Blackburn and Seligman [BS95] showed undecidability of $\mathcal{HL}_4(\Diamond,\downarrow)$-SAT involving the spypoint technique. The results of this chapter are a generalisation of that result, decreasing the number of modalities to two and restricting the class of frames. Furthermore, Goranko [Gor96] showed undecidability of $\mathcal{HL}(\Diamond,\downarrow,\mathsf{E})$-SAT. This and the previously mentioned result were generalised by Areces, Blackburn, and Marx [ABM99], who reduced the global satisfiability problem of a certain modal logic (whose undecidability was shown via a reduction from the tiling problem in [Spa93a]) to $\mathcal{HL}(\Diamond,\downarrow)$-SAT. Reviewing these results, ten Cate and Franceschet showed undecidability of the fragment of $\mathcal{HL}(\mathsf{F},\mathsf{P},\downarrow,@)$ without nested occurrences of $\downarrow$ via a reduction from the tiling problem in [tCF05b].

All these reductions have the same standard procedure in common, which consists of two basic steps: to enforce a satisfying model to behave like the $\mathbb{N}\times\mathbb{N}$ grid, and to encode the tiling in the states of this model. The first step consists in forcing the states of the model to behave like the nodes of the grid and mimicking the upper and right successor function by the accessibility relation(s). The second step must ensure that atomic propositions corresponding to tiles are assigned to the states of the model in such a way that the tiling conditions are satisfied. Deviations from this "standard procedure" can, of course, lead to appropriate reductions as well, see, for instance, [GKWZ05] — or Section 5.8.3 of this thesis, which contains a technically more involved reduction from a bounded tiling problem.

We will use the "standard procedure" (although the encoding of the grid will not be trivial), and combine it with a modification of the spypoint technique [BS95, ABM99], which we will modify into a spypoint-sinkpoint argument for Result (1).

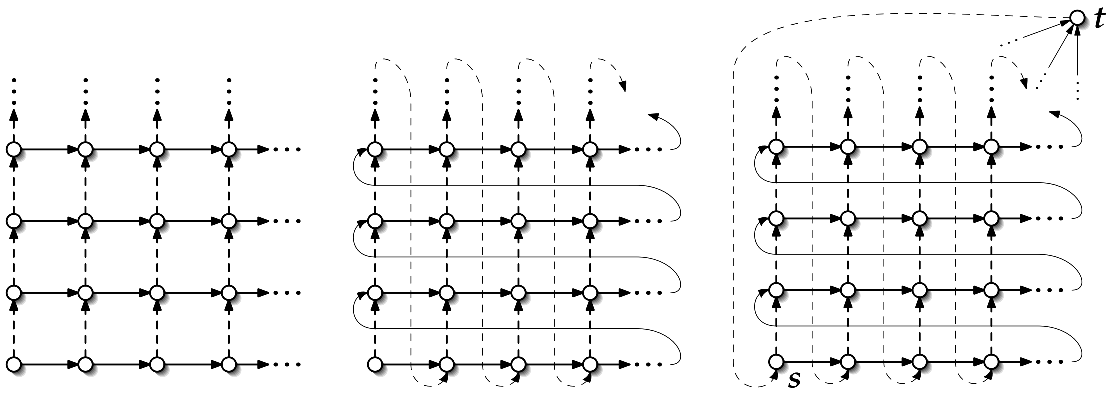
6.2 Linear orders and above

In this section, we show that $\mathcal{HL}_2(\Diamond, \downarrow)$ is able to encode tilings of $\mathbb{N}\times\mathbb{N}$ on any frame class containing one particular linear frame, which we will call Grid in the following. This ability is not too surprising if one considers the fact that $\downarrow$ is powerful enough to force the two accessibility relations to behave as the "right neighbour" and "upper neighbour" relations in the $\mathbb{N}\times\mathbb{N}$ grid. Since we are interested in a result as general as possible, we will have to insist on Grid having two linear (i.e., transitive, irreflexive, and trichotomous) accessibility relations when constructing this frame. As remarked in Section 6.1, such a frame condition is a bit artificial, but by considering it we will be able to cover a wide range of frame classes.

In order to construct Grid, we start with two accessibility relations R_h ("horizontal") and R_v ("vertical"). The frame will consist of points $(x, y) \in \mathbb{N}^2$, where $(x, y)R_h(x', y')$ whenever $x < x'$ and $y = y'$, and $(x, y)R_v(x', y')$ whenever $x = x'$ and $y < y'$. This situation is shown in Figure 6.1 (a), where a full line denotes an R_h edge, and a dashed line stands for an R_v edge. Note that the transitive closure of both relations is implicit. Clearly, R_h and R_v are irreflexive. For reasons just stated, we will make them trichotomous by adding extra edges as given in Figure 6.1 (b) and taking the transitive closure again. More precisely speaking, we make each point on the nth row see each point on the mth row via R_h, for each $m > n$; and we make each point on the nth column see each point on the mth column via R_v, for each $m > n$.

We will need to refer to the lower left point (the "origin" of the grid) several times. For this purpose, we introduce a variant of the spypoint technique. Apart from the fact that the "origin" behaves almost as a spypoint — that is, all other points in Grid are accessible from it via some R_h-R_v-path —, we will add a sinkpoint to the model that is accessible from all other points via R_h and that sees the spypoint via R_v, cf. Figure 6.1 (c). Note that the spypoint-sinkpoint construction does not destroy irreflexivity or trichotomy.

Let ∞ denote the sinkpoint. We formally define Grid $= \big(N, (R_h, R_v)\big)$, where $N = (\mathbb{N}\times\mathbb{N}) \cup \{\infty\}$, and the accessibility relations are given as follows, using



(a) Transitive and irreflexive
(b) In addition, trichotomous
(c) The frame Grid with spy-/sinkpoint

Figure 6.1: Simulating the $\mathbb{N}\times\mathbb{N}$ grid with two relations. The transitive closures are not drawn.

the abbreviation $\mathbb{N}^2 = \mathbb{N}\times\mathbb{N}$.

$$R_h = \big\{\big((x,y),(x',y')\big) \in (\mathbb{N}^2)^2 \mid (y = y' \text{ and } x < x') \text{ or } y < y'\big\} \cup \big(\mathbb{N}^2 \times \{\infty\}\big)$$
$$R_v = \big\{\big((x,y),(x',y')\big) \in (\mathbb{N}^2)^2 \mid (x = x' \text{ and } y < y') \text{ or } x < x'\big\} \cup \big(\{\infty\} \times \mathbb{N}^2\big)$$

Clearly, Grid is a linear frame. Whenever we construct a model based on Grid, we will name the spypoint s and the sinkpoint t, where s and t are nominals. This is reflected in Figure 6.1 (c), too. We now formulate our result such that it covers as many frame classes as possible.

Theorem 6.2 *Let $\mathfrak{F}$ be a bi-modal frame class with* Grid $\in \mathfrak{F}$.
Then $\mathcal{HL}_2(\Diamond,\downarrow)$-$\mathfrak{F}$-SAT is coRE*-complete.*

Proof. It suffices to show coRE-hardness because coRE-membership follows from the embedding of $\mathcal{HL}_2(\Diamond,\downarrow)$ into $\mathcal{FOL}$, see Section 2.2.2. Let T be a set of tile types. We define a formula φ_T that enforces each satisfying model to behave like the $\mathbb{N}\times\mathbb{N}$ grid, and that encodes the tiling. This formula has to be equipped with two properties. First, it must be satisfied in some model based on Grid, given a T-tiling. Second, φ_T has to enforce that each satisfying *arbitrary* model behaves as the T-tiled $\mathbb{N}\times\mathbb{N}$ grid. Hence, when constructing φ_T, we will have to enforce properties like for example transitivity or convergence that hold naturally in Grid, while we do not need to enforce, for instance, trichotomy.

We start with the conjuncts of φ_T responsible for the grid.

► The spypoint and sinkpoint are as given in Figure 6.1 (c).

$$\mathsf{SPY} = s \wedge \Diamond_h\big(t \wedge \Diamond_v s\big)$$

Before we proceed, we define a useful abbreviation that allows us to refer only to points that are not the sinkpoint.

$$\Diamond_h^{\neg t}\psi = \Diamond_h(\neg t \wedge \psi) \qquad\qquad \Box_h^{\neg t}\psi = \neg\Diamond_h^{\neg t}\neg\psi$$

Another shortcut is used for the "reflexive closure" of the modal operators.

$$\Diamond_v^{*}\psi = \psi \vee \Diamond_v\psi \qquad\qquad \Box_v^{*}\psi = \neg\Diamond_v^{*}\neg\psi$$
$$\Diamond_h^{*}\psi = \psi \vee \Diamond_h^{\neg t}\psi \qquad\qquad \Box_h^{*}\psi = \neg\Diamond_h^{*}\neg\psi$$

Note that $\Diamond_h^{*}$ is defined to contain $\Diamond_h^{\neg t}$ as a disjunct, hence we do not need to state "$\neg t$" explicitly whenever we use $\Diamond_h^{*}$ or $\Box_h^{*}$.

From now on, we will call all points other than the sinkpoint that are accessible from s via a sequence consisting of at most one R_v edge and at most one R_h edge *R_v-R_h-reachable*. Within the set of all R_v-R_h-reachable points, we can simulate the @ operator. Suppose x is bound to such a point, then we can assert $@_x\psi$ at *any* other point by going directly to the sinkpoint, from there to the spypoint and then to the point to which x is bound. This idea is captured by the following definition.

$$@_x\psi = \Diamond_h\Big(t \wedge \Diamond_v\big(s \wedge \Diamond_v^{*}\Diamond_h^{*}(x \wedge \psi)\big)\Big)$$

Note that $@_x\psi$ only works if the point to which x is bound is R_v-R_h-reachable. On the other hand, the point y at which $@_x\psi$ is satisfied, is enforced to see the sinkpoint horizontally. (As an aside, we could even simulate the "somewhere" modality E if we left out x on the right-hand side of the above definition.)

For the @ operator and subsequent conjuncts to function properly even on arbitrary frames, it will be necessary to require that every point accessible from R_v-R_h-reachable points is R_v-R_h-reachable again. This is ensured by the following formula enforcing that both relations are transitive within the grid.

► For every R_v-R_h-reachable point x, each point accessible from x via two R_v (or R_h) edges is accessible from x in one R_v (or R_h) step.

$$\mathsf{TRANS} = \Box_v^{*}\Box_h^{*}{\downarrow}x.\big(\Box_h^{\neg t}\Box_h^{\neg t}{\downarrow}y.@_x\Diamond_h y \wedge \Box_v\Box_v{\downarrow}y.@_x\Diamond_v y\big)$$

At first glance, the fact that TRANS uses the @ operator, while the @ operator seems to act on the assumption that the relations are transitive, appears to expose a cyclic definition. This is not the case because TRANS operates in an inductive manner, which will become clear further below when the tiling is constructed from a model satisfying φ_T.

We will need to refer to *neighbours* of points. A point y is a *right neighbour* of x if xR_hy and there is no z such that xR_hzR_hy. *Upper neighbours* are defined analogously. In order to refer to neighbours, we define "next" operators to be the following abbreviations.

$$\bigcirc_h\psi = {\downarrow} a.\Diamond_h^{\neg t} {\downarrow} b.(@_a\neg\Diamond_h\Diamond_h b \wedge \psi)$$
$$\bigcirc_v\psi = {\downarrow} a.\Diamond_v {\downarrow} b.(@_a\neg\Diamond_v\Diamond_v b \wedge \psi)$$

Whenever $\bigcirc_h$ and $\bigcirc_v$ are employed in the following, a and b must be substituted by fresh state variables. Note that these operators are diamond-style. We will not introduce an abbreviation for their duals. After we have required every R_v-R_h-reachable point to have exactly one right and one upper neighbour, the new next operators can be used box-style, as well.

- Every R_v-R_h-reachable point has exactly one right and exactly one upper neighbour.

$$\mathsf{NEIGH} = \Box_v^*\Box_h^* {\downarrow} x.(\bigcirc_h {\downarrow} y.@_x\neg\bigcirc_h\neg y \ \wedge\ \bigcirc_v {\downarrow} y.@_x\neg\bigcirc_v\neg y)$$

- For every R_v-R_h-reachable point x, the unique point y that is the right neighbour of the upper neighbour of x coincides with the upper neighbour of the right neighbour of x.

$$\mathsf{CONV} = \Box_v^*\Box_h^* {\downarrow} x.\bigcirc_v\bigcirc_h {\downarrow} y.@_x\bigcirc_h\bigcirc_v y$$

Having enforced the grid, it is straightforward to encode the tiling on it. For each tile type, we will use an atomic proposition to denote that a tile of that type lies at a given point. For the sake of short notation, we will deliberately confuse tile types with their associated atoms.

- At each point in the grid lies exactly one tile.

$$\mathsf{TILE} = \Box_v^*\Box_h^* \bigvee_{t\in T}\Big(t \wedge \bigwedge_{t'\neq t} \neg t'\Big)$$

- The tiling conditions are met.

$$\mathsf{MATCH} = \Box_v^*\Box_h^* \bigwedge_{t\in T}\Big(t \rightarrow (\bigvee_{t'\in \mathrm{UP}(t,T)} \bigcirc_v t' \ \wedge \bigvee_{t'\in \mathrm{RI}(t,T)} \bigcirc_h t')\Big)$$

Let $\varphi_T = \mathsf{SPY} \wedge \mathsf{TRANS} \wedge \mathsf{NEIGH} \wedge \mathsf{CONV} \wedge \mathsf{TILE} \wedge \mathsf{MATCH}$. In order to prove the statement of this theorem, it is sufficient to show that the following two propositions hold:

(1) If T admits a tiling, then φ_T is satisfiable in Grid.

(2) If φ_T is satisfiable in an arbitrary model, then T admits a tiling.

Proof of (1). Suppose T is given and admits a tiling of $\mathbb{N}^2$. Then there exists a function $\tau : \mathbb{N}^2 \to T$ such that for all $(x, y) \in \mathbb{N}^2$, the tiling condition holds. We construct a model $\mathcal{M} = \big(N, (R_h, R_v), V\big)$ based on Grid, where V is defined by $V(s) = \{(0,0)\}$, $V(t) = \{\infty\}$, and $V(t) = \{(x,y) \mid \tau(x,y) = t\}$ for each $t \in T$.

We claim that $\mathcal{M}, (0,0) \models \varphi_T$ and show that each conjunct of φ_T is satisfied at $(0,0)$ in $\mathcal{M}$. Conjunct SPY follows directly from the definitions of R_h and R_v of Grid. Since both relations are transitive, TRANS holds. Conjuncts NEIGH and CONV are satisfied because they express basic properties of R_h and R_v that are based on $\mathbb{N}^2$. TILE and MATCH hold due to the tiling.

Proof of (2). Let $\mathcal{M} = \big(M, (R_h, R_v), V\big)$ be an arbitrary model satisfying φ_T. Since s, t are nominals, there exist points $m_0, m_\infty \in M$ such that $V(s) = \{m_0\}$ and $V(t) = \{m_\infty\}$. Conjunct SPY implies $m_0 R_h m_\infty$ and $m_\infty R_v m_0$. We now define a mapping $f : \mathbb{N}^2 \to M - \{m_\infty\}$ that satisfies the following conditions for all $(x, y) \in \mathbb{N}^2$.

(3) If $x \geqslant 1$, then $f(x, y)$ is the right neighbour of $f(x-1, y)$.

(4) If $y \geqslant 1$, then $f(x, y)$ is the upper neighbour of $f(x, y-1)$.

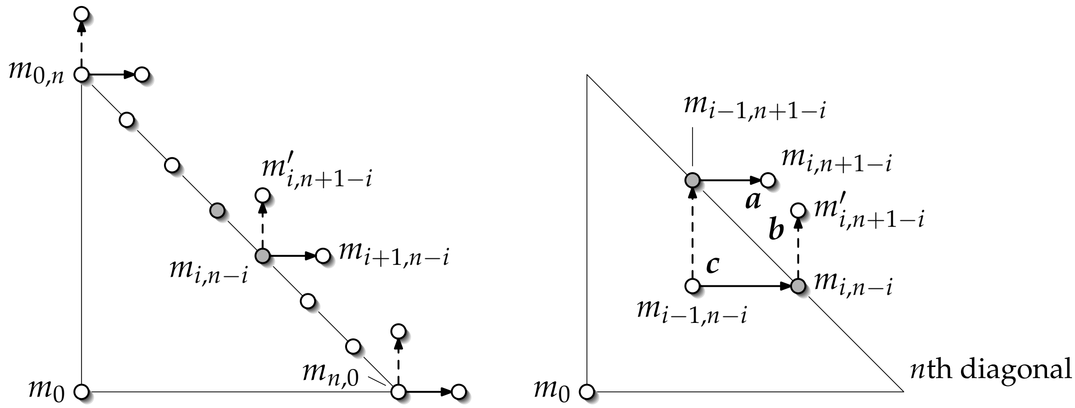
(5) If $x = 0$ and $y \geqslant 1$, then $m_0 R_v f(0, y)$.

(6) $f(x, y)$ is R_v-R_h-reachable.

(7) $f(x, y) R_h m_\infty$.

We construct f by induction on $n = x + y$, that is, diagonal-wise with respect to $\mathbb{N}^2$. The base case consists of $n = 0, 1$. For $n = 0$, we set $f(0,0) = m_0$. Since m_0 is R_v-R_h-reachable, NEIGH together with @ implies that m_0 has a unique right neighbour $m_{1,0}$ and a unique upper neighbour $m_{0,1}$. Due to the definition of @, they both see the sinkpoint via R_h. Set $f(1,0) = m_{1,0}$ and $f(0,1) = m_{0,1}$. Now Conditions (3)–(7) are satisfied up to the first diagonal.

For the induction step, suppose that $f(x, y)$ has already been defined for all (x, y) with $x + y \leqslant n$ (i.e., from the 0th to the nth diagonal), $n \geqslant 1$, and Conditions (3)–(7) hold up to here. Consider the points on the nth diagonal, namely $m_{i,n-i} = f(i, n-i)$ for $i = 0, \ldots, n$. Because of (6), NEIGH applies and implies that each $m_{i,n-i}$ has a unique horizontal successor $m_{i+1,n-i}$ and a unique vertical



(a) Points on the nth diagonal and their enforced successors

(b) Coincidence of $m_{i-1,n+1-i}$ and $m'_{i-1,n+1-i}$

Figure 6.2: The diagonal-wise construction of the grid

successor $m'_{i,n+1-i}$, see Figure 6.2 (a). Note that the @ operator works because each $m_{i,n-i}$ satisfies (6).

Now for each $i = 1, \ldots, n-1$, the points $a = m_{i,n+1-i}$ and $b = m'_{i,n+1-i}$ coincide. To justify this claim, let $c = f(i-1, n-i)$ (lying on the $(n-1)$st diagonal). Since c has the horizontal successor $m_{i,n-i}$ which has the vertical successor b, and c has the vertical successor $m_{i-1,n+1-i}$ which has the horizontal successor a, and (6) holds for c, CONV implies $a = b$. See also Figure 6.2 (b).

Let $f(0, n+1) = m'_{0,n+1}$ and $f(i, n+1-i) = m_{i,n+1-i}$, for all $i = 1, \ldots, n+1$. It follows from this construction that Conditions (3), (4), and (7) are satisfied for the "new" (x, y) from the $(n+1)$st diagonal. To end the inductive construction, we have to show that the "new" (x, y) also satisfy (5) and (6).

Condition (5) has to be shown for $(0, n+1)$. Since according to the induction hypothesis, $m_0 R_v f(0, n)$, TRANS applied to m_0 yields $m_0 R_v f(0, n+1)$.

Condition (6) for $(0, n+1)$ follows from (5). For the remaining $(i, n+1-i)$, we argue as follows. Due to the induction hypothesis, $m_{i-1,n+1-i}$ is R_v-R_h-reachable. Hence there is some point a which is accessible from m_0 in at most one R_v step and from which $m_{i-1,n+1-i}$ is accessible in at most one R_h step. If the last "at most one" is in fact 0, then we are done. If it is 1, then $m_{i,n+1-i}$ is accessible from a in two R_h steps. Since a is R_v-R_h-reachable, too, TRANS applied to a yields $a R_h m_{i,n+1-i}$, hence $m_{i,n+1-i} = f(i, n+1-i)$ is R_v-R_h-reachable.

With f at our disposal, we can easily define a function $\tau : \mathbb{N}^2 \to T$ as follows. Let $\tau(x, y) = t$ if and only if $f(x, y) \in V(t)$, for each $(x, y) \in \mathbb{N}^2$ and each $t \in T$. The correctness of this definition is ensured by the construction of f and TILE. Because of MATCH, τ satisfies the tiling conditions. ❑

Please observe that the formula φ_T occurring in the proof uses only two nominals s and t. Those can in fact be replaced by two more bound state variables. Furthermore, φ_T does not contain any free state variables. Hence, the statement of Theorem 6.2 does in fact hold for the nominal-free fragment of all sentences of $\mathcal{HL}_2(\diamond, \downarrow)$. Furthermore, Theorem 6.2 has the following implication.

Corollary 6.3 *$\mathcal{HL}_2(\diamond, \downarrow)$-trans-SAT and $\mathcal{HL}_2(\diamond, \downarrow)$-lin-SAT are CORE-complete.*

6.3 Frames with equivalence relations and above

In this section, we show that $\mathcal{HL}_2(\diamond, \downarrow)$ is able to encode tilings on *any* frame class containing one particular ER frame, which we will call Grid2 in the following. For the sake of an easy definition of the accessibility relations, we will consider tilings of the whole $\mathbb{Z}\times\mathbb{Z}$ grid here.

Before we again state a result that covers a wide range of frame classes, we give a construction of Grid2 and formally define this bi-modal frame to be $\mathsf{Grid2} = \big(N, (R_1, R_2)\big)$, whose components are given as follows.

- $N = (\mathbb{Z}\times\mathbb{Z}) \cup (\mathbb{Z}_2 \times \mathbb{Z}_2 \times \{+\}) \cup \{s\}$, where $\mathbb{Z}_2 = \{2z \mid z \in \mathbb{Z}\}$.
  Let N' denote $N - (\mathbb{Z}\times\mathbb{Z})$.
- $R_2 = \bigcup_{k,l\in\mathbb{Z}} \text{minicluster}(2k+1, 2l+1) \cup (N' \times N')$,
  where $\text{minicluster}(i,j) = \{(i,j), (i+1,j), (i,j+1), (i+1,j+1)\}^2$.
- $R_1 = \bigcup_{k,l\in\mathbb{Z}} \text{Minicluster}(2k, 2l)$,
  where $\text{Minicluster}(i,j) = \{(i,j), (i+1,j), (i,j+1), (i+1,j+1), (i,j,+)\}^2$.

These definitions are visualised in Figure 6.3, where a full line denotes an R_1 edge, and a dashed line stands for an R_2 edge. Note that due to symmetry, no arrowheads appear. Furthermore, many edges implied by transitivity have not been drawn for the sake of clarity.

The idea behind the construction of N needs a detailed explanation. First, each pair $(x, y) \in \mathbb{Z}\times\mathbb{Z}$ is a state and represents the point (x, y) from the $\mathbb{Z}\times\mathbb{Z}$ grid. These points are drawn in Figure 6.3 (a), where R_1 and R_2, each restricted to $\mathbb{Z}\times\mathbb{Z}$, are shown as well. We call every unit square in the same drawing a *minicluster*. Second, for every R_1-minicluster, with $(2i, 2j)$ as its lower left point, there is an additional point $(2i, 2j, +)$ that belongs to the same R_1-cluster. We call these new points *local spypoints* and collect them in $\mathbb{Z}_2 \times \mathbb{Z}_2 \times \{+\}$. Finally, there is a spypoint s that sees all local spypoints via R_2. This situation is shown

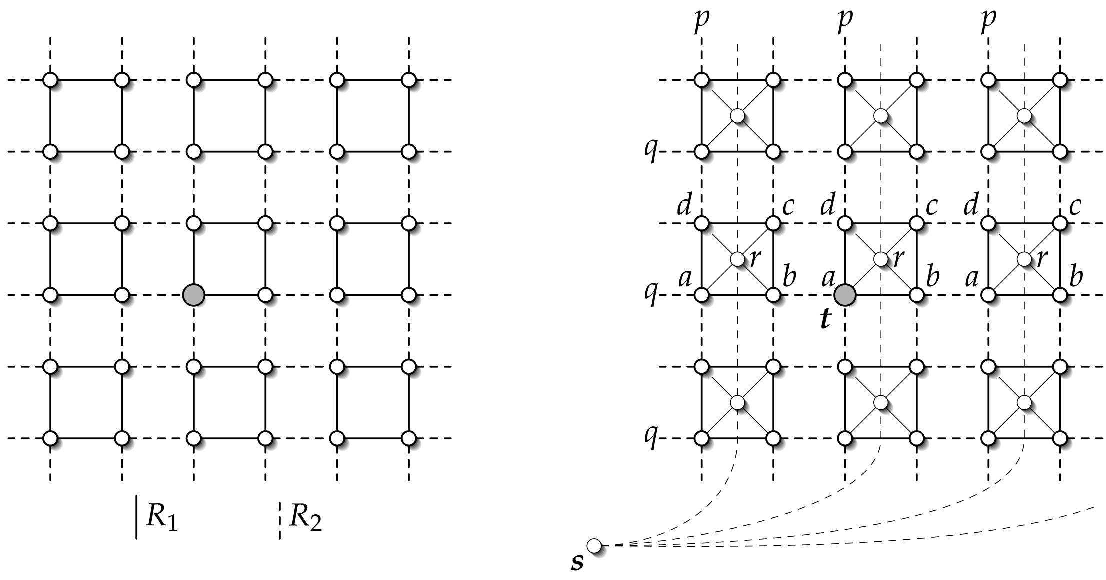


Figure 6.3: Simulating the $\mathbb{Z}\times\mathbb{Z}$ grid with two equivalence relations. Each line represents a bidirectional arrow. The transitive (and hence, reflexive) closures are not drawn.

in Figure 6.3 (b). Note that this somewhat intricate construction is necessary in order to distinguish directions, to distinguish direct neighbours from indirect ones, and to have access from s to each point in the grid at once— all in the presence of equivalence relations.

Whenever we will construct a model based on Grid2, we will name the spypoint s, where s is a nominal. The point $(0,0)$ is named by the nominal t, and the local spypoints are labelled by the atomic proposition r. Furthermore, we will use the atomic propositions p and q to label those points that lie on an even column or row, respectively. This will enable us to distinguish between four directions. For this purpose, we define the following abbreviations.

$a =$	$p \wedge q \wedge \neg r$	even row and even column
$b =$	$\neg p \wedge q \wedge \neg r$	even row and odd column
$c =$	$\neg p \wedge \neg q \wedge \neg r$	odd row and odd column
$d =$	$p \wedge \neg q \wedge \neg r$	odd row and even column

All these settings are reflected in Figure 6.3 (b), too. Again, we formulate our result such that it covers as many frame classes as possible, namely each class of frames containing Grid2. This includes the class of ER frames.

Theorem 6.4 *Let $\mathfrak{F}$ be a bi-modal frame class with* Grid2 $\in \mathfrak{F}$. *Then $\mathcal{HL}_2(\Diamond, \downarrow)$-$\mathfrak{F}$-*SAT *is* CORE*-complete.*

Proof. Again, it suffices to show CORE-hardness. Let T be a set of tile types. We define a formula φ_T that enforces each satisfying model to behave like the $\mathbb{N}\times\mathbb{N}$ grid, and that encodes the tiling — again using an atomic propositions t for each $t \in T$. In order to keep every part of φ_T short, we define two kinds of abbreviation.

First, we will have to refer to R_1-successors that are *not* local spypoints, and to R_2-successors that *are* local spypoints. This is done via new modal operators $\Diamond_i^l, \Box_i^l$ and $\Diamond_i^r, \Box_i^r$, where $i = 1, 2$, which are defined as follows.

$$\Diamond_i^l\psi = \Diamond_i(\neg r \wedge \psi) \qquad \Diamond_i^r\psi = \Diamond_i(r \wedge \psi)$$
$$\Box_i^l\psi = \neg\Diamond_i^l\neg\psi \qquad \Box_i^r\psi = \neg\Diamond_i^r\neg\psi$$

Second we define abbreviations that give us direct access to the left, right, upper, and lower neighbour of a given point.

$$\begin{aligned}
\Diamond_l\psi &= \big(a \wedge \Diamond_2(b\wedge\psi)\big) \vee \big(b \wedge \Diamond_1(a\wedge\psi)\big) \vee \big(c \wedge \Diamond_1(d\wedge\psi)\big) \vee \big(d \wedge \Diamond_2(c\wedge\psi)\big)\\
\Diamond_r\psi &= \big(a \wedge \Diamond_1(b\wedge\psi)\big) \vee \big(b \wedge \Diamond_2(a\wedge\psi)\big) \vee \big(c \wedge \Diamond_2(d\wedge\psi)\big) \vee \big(d \wedge \Diamond_1(c\wedge\psi)\big)\\
\Diamond_u\psi &= \big(a \wedge \Diamond_2(d\wedge\psi)\big) \vee \big(b \wedge \Diamond_2(c\wedge\psi)\big) \vee \big(c \wedge \Diamond_1(b\wedge\psi)\big) \vee \big(d \wedge \Diamond_1(a\wedge\psi)\big)\\
\Diamond_d\psi &= \big(a \wedge \Diamond_1(d\wedge\psi)\big) \vee \big(b \wedge \Diamond_1(c\wedge\psi)\big) \vee \big(c \wedge \Diamond_2(b\wedge\psi)\big) \vee \big(d \wedge \Diamond_2(a\wedge\psi)\big)
\end{aligned}$$

As usual, the duals are defined by $\Box_x\psi = \neg\Diamond_x\neg\psi$, $x \in \{l, r, u, d\}$. Note that we are not forced to use $\Diamond_1^l$ instead of $\Diamond_1$ in these definitions, because $\neg r$ is already required in a, b, c, d.

From now on, we call a point *accessible* if it is reachable from the spypoint using an R_2 edge to a local spypoint and from there using an R_1 edge, where both edges are bidirectional. Hence, ψ is satisfied at every accessible point if $\Box_2^r\Box_1^l\psi$ is satisfied at the spypoint. Now, the formula φ_T consists of the following conjuncts.

- The spypoint is named s. It does not satisfy r and sees itself via R_2. The origin is accessible. It is named t and satisfies a. Every point R_2-accessible from the spypoint is labelled r.

$$\mathsf{SPY} = s \wedge \neg r \wedge \Diamond_2 s \wedge \Diamond_2^r\Diamond_1^l(t \wedge a \wedge \Diamond_1^r\Diamond_2^l s) \wedge \Box_2(\neg s \rightarrow r)$$

- Each accessible point has a unique left, right, upper, and lower neighbour, respectively. Each of these four neighbours is accessible again.

$$\begin{aligned}\mathsf{NEIGH} = \Box_2^r \Box_1^l \downarrow x.\Big[\quad & \Diamond_l \downarrow y.\Diamond_1^r \Diamond_2^l \Big(s \wedge \Diamond_2^r \Diamond_1^l (y \wedge \Diamond_r (x \wedge \Box_l y))\Big) \\ \wedge\, & \Diamond_r \downarrow y.\Diamond_1^r \Diamond_2^l \Big(s \wedge \Diamond_2^r \Diamond_1^l (y \wedge \Diamond_l (x \wedge \Box_l y))\Big) \\ \wedge\, & \Diamond_u \downarrow y.\Diamond_1^r \Diamond_2^l \Big(s \wedge \Diamond_2^r \Diamond_1^l (y \wedge \Diamond_d (x \wedge \Box_l y))\Big) \\ \wedge\, & \Diamond_d \downarrow y.\Diamond_1^r \Diamond_2^l \Big(s \wedge \Diamond_2^r \Diamond_1^l (y \wedge \Diamond_u (x \wedge \Box_l y))\Big)\Big]\end{aligned}$$

- Convergence holds, that is, for each accessible point x, the (unique) point that is the right neighbour of the upper neighbour of x coincides with the upper neighbour of the right neighbour of x.

  $$\mathsf{CONV} = \Box_2^r \Box_1^l \Box_u \Box_r \downarrow x. \Box_l \Box_d \Box_r \Box_u x$$

  (Note that it suffices to replace the prefix $\Box_2^r \Box_1^l \Box_u \Box_r$ by $\Box_2^r \Box_1^l$, but the given definition of CONV simplifies the considerations at the end of this proof.)

- At each point in the grid lies exactly one tile.

  $$\mathsf{TILE} = \Box_2^r \Box_1^l \bigvee_{t \in T} \Big(t \wedge \bigwedge_{t' \neq t} \neg t' \Big)$$

- The tiling conditions are met.

  $$\mathsf{MATCH} = \Box_2^r \Box_1^l \bigwedge_{t \in T} \Big(t \to \big(\bigvee_{t' \in \mathrm{UP}(t,T)} \Box_u t' \ \wedge \bigvee_{t' \in \mathrm{RI}(t,T)} \Box_r t' \big)\Big)$$

Let $\varphi_T = \mathsf{SPY} \wedge \mathsf{NEIGH} \wedge \mathsf{CONV} \wedge \mathsf{TILE} \wedge \mathsf{MATCH}$. Note that we only have to require certain properties of Grid2, but not all of them. For example, it is not necessary to enforce that the R_i are equivalence relations or that each four points that correspond to an R_1-minicluster from Grid2 have a *common* local spypoint. The properties enforced by φ_T are chosen such that they are satisfied by Grid2 on the one hand, and sufficient for a satisfying model to encode a tiling on the other hand. More precisely, it remains to prove the following two propositions.

(1) If T admits a tiling, then φ_T is satisfiable in Grid2.

(2) If φ_T is satisfiable in an arbitrary model, then T admits a tiling.

Proof of (1). Proposition (1) is shown as in the proof of Theorem 6.2.

Proof of (2). Let $\mathcal{M} = \big(M, (R_1, R_2), V\big)$ be an arbitrary model satisfying φ_T at m_0. Because of SPY, $V(s) = \{m_0\}$. SPY also implies that there is an accessible point $m_{0,0}$ satisfying t and a. We define a mapping $f : \mathbb{Z}\times\mathbb{Z} \to M$ satisfying the following conditions for all $(x, y) \in \mathbb{Z}\times\mathbb{Z}$.

(3) $f(x, y)$ is accessible

(4) (i) $2 \mid x \;\Leftrightarrow\; \mathcal{M}, f(x, y) \Vdash p$

(ii) $2 \mid y \;\Leftrightarrow\; \mathcal{M}, f(x, y) \Vdash q$

(5) (i) $x \geqslant 1 \;\Rightarrow\; \big(\mathcal{M}, f(x-1, y) \Vdash p \;\Rightarrow\; f(x-1, y) R_1 f(x, y) R_1 f(x-1, y)\big)$

(ii) $x \geqslant 1 \;\Rightarrow\; \big(\mathcal{M}, f(x-1, y) \Vdash \neg p \;\Rightarrow\; f(x-1, y) R_2 f(x, y) R_2 f(x-1, y)\big)$

(iii) $y \geqslant 1 \;\Rightarrow\; \big(\mathcal{M}, f(x, y-1) \Vdash q \;\Rightarrow\; f(x, y-1) R_1 f(x, y) R_1 f(x, y-1)\big)$

(iv) $y \geqslant 1 \;\Rightarrow\; \big(\mathcal{M}, f(x, y-1) \Vdash \neg q \;\Rightarrow\; f(x, y-1) R_2 f(x, y) R_2 f(x, y-1)\big)$

We construct f by induction on $n = |x| + |y|$. For a given $n \in \mathbb{N}$, all points (x, y) satisfying $|x| + |y| = n$ lie on a square that is rotated by 45 degrees and whose corners are $(n, 0)$, $(-n, 0)$, $(0, n)$, and $(0, -n)$. In the considerations to follow, we shall restrict ourselves to the first quadrant, that is, $\mathbb{N}\times\mathbb{N}$. The arguments for the other three quadrants are analogous. (This might seem to suggest that it would have been easier to encode tilings of the $\mathbb{N}\times\mathbb{N}$ grid in the first place. However, it is not, because we would have to treat the boundaries of the grid separately when defining R_1, R_2, and NEIGH.)

The base case consists of $n = 0, 1$. Set $f(0, 0) = m_{0,0}$. Now NEIGH implies that there exist accessible $m_{1,0}, m_{0,1} \in M$ such that $\mathcal{M}, m_{1,0} \Vdash b$; $\mathcal{M}, m_{0,1} \Vdash d$; and there exist R_1-edges in both directions between $m_{0,0}$ and each of these two new points. Set $f(1, 0) = m_{1,0}$ and $f(0, 1) = m_{0,1}$. Clearly, Conditions (3)–(5) hold for all x, y with $x + y \leqslant 1$.

For the induction step, suppose that $f(x, y)$ has already been defined and satisfies Conditions (3)–(5) for all (x, y) with $x + y \leqslant n$. Consider the points on the nth diagonal, namely $m_{i,n-i} = f(i, n-i)$ for $i = 0, \ldots, n$. Because of (3), NEIGH applies, hence each $m_{i,n-i}$ has a unique right neighbour $m_{i+1,n-i}$ and a unique upper neighbour $m'_{i,n+1-i}$, see Figure 6.4.

With the same justifications as in the proof of Theorem 6.2, we conclude from CONV that $m_{i,n+1-i}$ and $m'_{i,n+1-i}$ coincide for each $i = 1, \ldots, n$. Set $f(0, n+1) = m'_{0,n+1}$ and $f(i, n+1-i) = m_{i,n+1-i}$, $i = 1, \ldots, n+1$. Now this construction and NEIGH imply (3)–(5) for all x, y with $x + y \leqslant n + 1$.

Now we define $\tau : \mathbb{Z}\times\mathbb{Z} \to T$ as follows. Let $\tau(x, y) = t$ if and only if $f(x, y) \in V(t)$, for each $(x, y) \in \mathbb{Z}\times\mathbb{Z}$ and each $t \in T$. The correctness of this definition is ensured by the construction of f and TILE. Because of MATCH, τ satisfies the tiling conditions. ❑

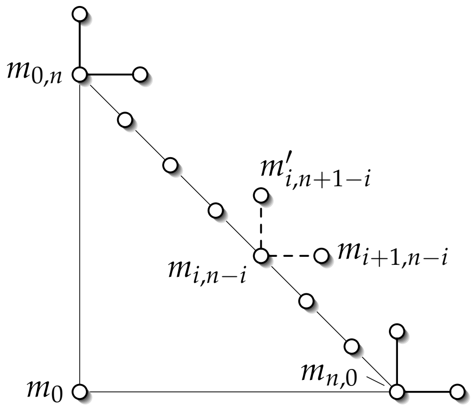


Figure 6.4: Points on the nth diagonal and their enforced successors

The note from the end of Section 6.2 applies here as well. Furthermore, Theorem 6.4 has the following implication.

Corollary 6.5 *$\mathcal{HL}_2(\Diamond, \downarrow)$-ER-SAT is coRE-complete.*

Chapter 7

Conclusion

In this thesis, we have systematically and almost exhaustively examined the computational complexity of model checking and the satisfiability problem for a collection of hybrid languages over several frame classes relevant for temporal and epistemic applications. Our "collection of languages" comprises all relevant combinations of those modal, temporal, and hybrid operators that were defined in Section 2.1 — namely F, P, U, S, ↓, ∃, @, and E. The frame classes we have considered are the classes of arbitrary frames, transitive frames, transitive trees, linear orders, $\{(\mathbb{N}, >)\}$, and the class of ER frames. We have collected results from the literature and complemented them by new results for almost all combinations that have been open.

For model checking, we have contributed the result that PSPACE-hardness for binder languages does not only hold over arbitrary frames [FdR06], but also over restricted frame classes. For satisfiability, we have completely classified the complexity over ER frames by our own results and have shown that complexity over transitive frames and transitive trees increases dramatically if the ↓ language is extended by other operators. Furthermore, we have established the complexity of satisfiability for hybrid until/since languages over the last-mentioned frame classes. All these results are visualised in Figures 5.1–5.6.

For establishing our results on model checking and the satisfiability problem, we have applied many well-known techniques such as reductions from tiling problems, the spypoint technique, encoding counters, and reductions from or to decision problems in modal logic, hybrid logic, fragments of first-order logic, and fixpoint logic. With the exception of satisfiability of until/since languages over transitive frames, all complexity bounds established in our results are tight. In addition to this gap, there remain two open questions, namely the complexity of satisfiability for $\mathcal{HL}(\mathsf{U}, \downarrow)$ over transitive frames and of until/past and until/since languages over linear frames.

Furthermore, we have examined satisfiability for multi-modal hybrid binder languages over the frame classes given above, except for transitive trees and $(\mathbb{N}, >)$. Our undecidability results even include all classes of frames containing one particular bi-modal linear or ER frame.

The classification we have provided gives rise to several further questions worth examining. For instance, languages with other hybrid operators such as the binders $\Downarrow$, Σ [BS95] could be incorporated into the hierarchies, and the complexity of their decision problems could be solved. It is also worthwhile considering other frame classes, such as epistemically relevant superclasses of the class of ER frames. Both these extensions could be examined in a more general way by considering combinations of first-order formulae in place of combinations of (first-order definable) hybrid operators and frame classes. By means of this abstraction, a classification of complexity for all such possible combinations could be approached.

In case of the very expressive binder languages, which have been shown to have very high complexity, restrictions other than via the class of frames could be considered. A promising approach is to systematically restrict the set of permitted Boolean operators in the language, as was done for classical propositional satisfiability in [Lew79], or, more recently, for modal satisfiability in [BHSS06] and satisfiability for linear temporal logic [BSS$^+$07].

For our results on multi-modal hybrid logics, it is interesting to ask whether they carry over to other classes of frames. In particular, multi-modal frame classes with compatibility conditions between the accessibility relations can be considered. Many-dimensional modal logics are multi-modal logics with special cases of such conditions. Much is known about their properties and their complexity [GKWZ03], but there are many open problems in this field. It is certainly worthwhile to extend such logics by hybrid operators and find out how this affects the complexity of decision problems.

Finally, other decision problems, such as logical implication, the linear-time model checking problem, or problems connected with conservative extensions (see, for instance, [GLW06] in connection with description logics) could be examined.

Bibliography

[ABD$^+$05] L. Afanasiev, P. Blackburn, I. Dimitriou, B. Gaiffe, E. Goris, M. Marx, and M. de Rijke. PDL for ordered trees. *Journal of Applied Non-Classical Logics*, 15(2):115–135, 2005.

[ABM99] C. Areces, P. Blackburn, and M. Marx. A road-map on complexity for hybrid logics. In *Proc. of the 13th CSL, 1999*, volume 1683 of *LNCS*, pages 307–321. Springer, 1999.

[ABM00] C. Areces, P. Blackburn, and M. Marx. The computational complexity of hybrid temporal logics. *Logic Journal of the IGPL*, 8(5):653–679, 2000.

[ABM01] C. Areces, P. Blackburn, and M. Marx. Hybrid logics: Characterization, interpolation and complexity. *Journal of Symbolic Logic*, 66(3):977–1010, 2001.

[AtC06] C. Areces and B. ten Cate. *Hybrid Logics*, pages 821–868. In [BvBW06], 2006.

[BdRV05] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2005. Reprint.

[Ber66] R. Berger. The undecidability of the domino problem. *Memoirs of the American Mathematical Society*, 66:1–72, 1966.

[BGG97] E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem*. Perspectives of Mathematical Logic. Springer, 1997.

[BHSS06] M. Bauland, E. Hemaspaandra, H. Schnoor, and I. Schnoor. Generalized modal satisfiability. In B. Durand and W. Thomas, editors, *STACS*, volume 3884 of *Lecture Notes in Computer Science*, pages 500–511. Springer, 2006.

[Bla93] P. Blackburn. Nominal tense logic. *Notre Dame Journal of Formal Logic*, 34(1):56–83, 1993.

[Bla00a] P. Blackburn. Internalizing labeled deuction. *Journal of Logic, Language and Information*, 10:137–168, 2000.

[Bla00b] P. Blackburn. Representation, reasoning, and relational structures: a hybrid logic manifesto. *Logic Journal of the IGPL*, 8(3):339–365, 2000.

[BS95] P. Blackburn and J. Seligman. Hybrid languages. *Journal of Logic, Language and Information*, 4(3):251–272, 1995.

[BS98] P. Blackburn and J. Seligman. What are hybrid languages? In M. Kracht, M. de Rijke, H. Wansing, and M. Zakharyaschev, editors, *Advances in Modal Logic*, volume 1 of *CSLI Publications*, pages 41–62. Stanford University, 1998.

[BSS$^+$07] M. Bauland, T. Schneider, H. Schnoor, I. Schnoor, and H. Vollmer. The complexity of generalized satisfiability for linear temporal logic. In H. Seidl, editor, *FoSSaCS*, volume 4423 of *Lecture Notes in Computer Science*, pages 48–62. Springer, 2007.

[BT99] P. Blackburn and M. Tzakova. Hybrid languages and temporal logic. *Logic Journal of the IGPL*, 7(1):27–54, 1999.

[Bul70] R. Bull. An approach to tense logic. *Theoria*, 36:282–300, 1970.

[BvBW06] P. Blackburn, J. van Benthem, and F. Wolter. *Handbook of Modal Logic*. Elsevier, 2006.

[CGP01] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. MIT, 2001. 3rd print.

[Chl86] B. Chlebus. Domino-tiling games. *Journal of Computer and System Sciences*, 32:374–392, 1986.

[DO05] A. Dawar and M. Otto. Modal characterisation theorems over special classes of frames. In *Proc. of the 20th LICS, 2005*, pages 21–30. IEEE Computer Society, 2005.

[EFT96] H. Ebbinghaus, J. Flum, and W. Thomas. *Einführung in die mathematische Logik*. Spektrum Akademischer Verlag, 4th edition, 1996.

[FdR06] M. Franceschet and M. de Rijke. Model checking for hybrid logics (with an application to semistructured data). *Journal of Applied Logic*, 4(3):279–304, 2006.

[FdRS03] M. Franceschet, M. de Rijke, and B. Schlingloff. Hybrid logics on linear structures: Expressivity and complexity. In *Proc. of the 10th TIME, 2003*, pages 166–173. IEEE Computer Society, 2003.

[FHMV95] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*. The MIT Press, 1995.

[Fit96] M. Fitting. *First-order logic and automated theorem proving*. Graduate texts in computer science. Springer, 2nd edition, 1996.

[Gab96] D. Gabbay. *Labeled deductive systems*. Oxford University Press, 1996.

[GG93] G. Gargov and V. Goranko. Modal logic with names. *Journal of Philosophical Logic*, 22:607–636, 1993.

[GKWZ03] D. Gabbay, Á. Kurucz, F. Wolter, and M. Zakharyaschev. *Many-dimensional modal logics: Theory and applications*, volume 148 of *Studies in Logic*. Elsevier Science, 2003.

[GKWZ05] D. Gabelaia, Á. Kurucz, F. Wolter, and M. Zakharyaschev. Products of transitive modal logics. *Journal of Symbolic Logic*, 70(3):993–1021, 2005.

[GLW06] S. Ghilardi, C. Lutz, and F. Wolter. Did I damage my ontology? A case for conservative extensions in description logics. In P. Doherty, J. Mylopoulos, and C. A. Welty, editors, *KR*, pages 187–197. AAAI Press, 2006.

[GMV99] H. Ganzinger, C. Meyer, and M. Veanes. The two-variable guarded fragment with transitive relations. In *Proc. of the 14th LICS, 1999*, pages 24–34. IEEE Computer Society, 1999.

[Gor96] V. Goranko. Hierarchies of modal and temporal logics with reference pointers. *Journal of Logic, Language and Information*, 5(1):1–24, 1996.

[Grä99] E. Grädel. On the restraining power of guards. *Journal of Symbolic Logic*, 64:1719–1742, 1999.

[GW99] E. Grädel and I. Walukiewicz. Guarded fixed point logic. In *Proc. of the 14th LICS, 1999*, pages 45–54. IEEE Computer Society, 1999.

[HHK02] R. Hirsch, I. M. Hodkinson, and Á. Kurucz. On modal logics between $\mathbf{K} \times \mathbf{K} \times \mathbf{K}$ and $\mathbf{S5} \times \mathbf{S5} \times \mathbf{S5}$. *Journal of Symbolic Logic*, 67(1):221–234, 2002.

[HM92] J. Y. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54(3):319–379, 1992.

[HR07] J. Y. Halpern and L. C. Rêgo. Characterizing the NP-PSPACE gap in the satisfiability problem for modal logic. In M. M. Veloso, editor, *IJCAI*, pages 2306–2311, 2007.

[IRR$^+$04] N. Immerman, A. M. Rabinovich, T. W. Reps, S. Sagiv, and G. Yorsh. The boundary between decidability and undecidability for transitive-closure logics. In *Proc. of the 13th CSL, 2004*, volume 3210 of *LNCS*, pages 160–174. Springer, 2004.

[Kie02] E. Kieronski. EXPSPACE-complete variant of guarded fragment with transitivity. In *Proc. of the 19th STACS, 2002*, volume 2285 of *LNCS*, pages 608–619. Springer, 2002.

[Kie03] E. Kieronski. The two-variable guarded fragment with transitive guards is 2EXPTIME-hard. In *Proc. of the 6th FoSSaCS, 2003*, volume 2620 of *LNCS*, pages 299–312. Springer, 2003.

[Kra97] M. Kracht. Inessential features. In *Selected Papers of 1st LACL, 1996*, volume 1328 of *LNCS*, pages 43–62. Springer, 1997.

[KW97] M. Kracht and F. Wolter. Simulation and transfer results in modal logic – a survey. *Studia Logica*, 59(1):149–177, 1997.

[Lad77] R. E. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal on Computing*, 6(3):467–480, 1977.

[Lew79] H. Lewis. Satisfiability problems for propositional calculi. *Mathematical Systems Theory*, 13:45–53, 1979.

[LW05] T. Litak and F. Wolter. All finitely axiomatizable tense logics of linear time flows are CONP-complete. *Studia Logica*, 78:1–13, 2005.

[Mar99] Maarten Marx. Complexity of products of modal logics. *J. Log. Comput.*, 9(2):197–214, 1999.

[Mar04] N. Markey. Past is for free: On the complexity of verifying linear temporal properties with past. *Acta Informatica*, 40(6-7):431–458, May 2004.

[MS07a] M. Mundhenk and T. Schneider. The complexity of hybrid logics over equivalence relations. In T. Braüner and J. Villadsen, editors, *HyLo*, 2007. To appear.

[MS07b] Martin Mundhenk and Thomas Schneider. Undecidability of multi-modal hybrid logics. *Electronic Notes in Theoretical Computer Science*, 174(6):29–43, 2007.

[MSSW05] M. Mundhenk, T. Schneider, T. Schwentick, and V. Weber. Complexity of hybrid logics over transitive frames. In H. Schlingloff, editor, *M4M*, volume 194 of *Informatik-Berichte*, pages 62–78. Humboldt-Universität zu Berlin, 2005.

[MSSW07] M. Mundhenk, T. Schneider, T. Schwentick, and V. Weber. Complexity of hybrid logics over transitive frames. *Journal of Logic, Language and Information*, 2007. To appear.

[ON80] H. Ono and A. Nakamura. On the size of refutation Kripke models for some linear modal and tense logics. *Studia Logica*, 34:325–333, 1980.

[Pap94] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[Pri67] A. Prior. *Past, Present and Future*. Oxford University Press, 1967.

[PT91] S. Passy and T. Tinchev. An essay in combinatory dynamic logic. *Information and Computation*, 93:263–332, 1991.

[Rey92] Mark Reynolds. An axiomatization for Until and Since over the reals without the irr rule. *Studia Logica*, 51(2):165–193, 1992.

[Rey03] M. Reynolds. The complexity of the temporal logic with "until" over general linear time. *Journal of Computer and System Sciences*, 66(2):393–426, 2003.

[Rob71] R. M. Robinson. Undecidability and nonperiodicity for tilings of the plane. *Inventiones mathematicae*, 12:177–209, 1971.

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

[Spa93a] E. Spaan. *Complexity of modal logics*. PhD thesis, ILLC, University of Amsterdam, 1993.

[Spa93b] E. Spaan. The complexity of propositional tense logics. In M. de Rijke, editor, *Diamonds and Defaults*, pages 287–307. Kluwer, 1993.

[ST01] W. Szwast and L. Tendera. On the decision problem for the guarded fragment with transitivity. In *Proc. of the 16th LICS, 2001*, pages 147–156. IEEE Computer Society, 2001.

[Sto74] L. Stockmeyer. *The Complexity of Decision Problems in Automata and Logic*. PhD thesis, MIT, 1974.

[Sto77] L. Stockmeyer. The polynomial time hierarchy. *Theoretical Computer Science*, 3:1–22, 1977.

[SvEB84] M. Savelsbergh and P. van Emde Boas. Bounded tiling, an alternative to satisfiability. In G. Wechsung, editor, *2nd Frege Conference*, volume 20 of *Mathematische Forschung*, pages 354–363. Akademie Verlag, Berlin, 1984.

[tCF05a] B. ten Cate and M. Franceschet. Guarded fragments with constants. *Journal of Logic, Language and Information*, 14(3):281–288, 2005.

[tCF05b] B. ten Cate and M. Franceschet. On the complexity of hybrid logics with binders. In *Proc. of the 19th CSL, 2005*, volume 3634 of *LNCS*, pages 339–354. Springer, 2005.

[Tho74] S. K. Thomason. Reduction of tense logic to modal logic I. *Journal of Symbolic Logic*, 39(3):549–551, 1974.

[vEB97] P. van Emde Boas. The convenience of tilings. In A. Sorbi, editor, *Complexity, Logic, and Recursion Theory*, Lecture Notes in Pure and Applied Mathematics, pages 331–363. Marcel Dekker Inc., 1997.

[Wan61] H. Wang. Proving theorems by pattern recognition II. *Bell Systems Technical Journal*, 40:1–42, 1961.

[Wol96] F. Wolter. Tense logic without tense operators. *Mathematical Logic Quarterly*, 42:145–171, 1996.

[Wol97] F. Wolter. Completeness and decidability of tense logics closely related to logics above **K4**. *Journal of Symbolic Logic*, 62:131–158, 1997.

[Wos85] L. Wos. What is automated reasoning? *Journal of Automated Reasoning*, 1(1):6–9, 1985.

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Jena, den 14. 05. 2007

Thomas Schneider

Lebenslauf

Persönliche Daten

Name	Thomas Schneider
Geburt	15. 06. 1976 in Leipzig

Schulausbildung

1983–1991	Polytechnische Oberschule, Erfurt
1991–1995	Spezialschulteil mathematisch-naturwissenschaftlich-technischer Richtung des Albert-Schweitzer-Gymnasiums Erfurt
06/1995	Abitur

Studium

10/1997–09/2002	Studium der Mathematik an der Friedrich-Schiller-Universität (FSU) Jena
06/1999–09/2002	Stipendium des Studienförderwerks Klaus Murmann
02/2001–08/2001	Auslandssemester an der *University of Cape Town*, Südafrika
09/2002	Abschluss des Studiums mit dem Grad Diplom-Mathematiker Thema der Diplomarbeit: Komplexität modaler Logiken
10/2003	Examenspreis der Fakultät für Mathematik und Informatik der FSU

Akademische Laufbahn

seit 10/2002	wissenschaftlicher Mitarbeiter am Institut für Informatik der FSU

Außerberufliches

seit 11/1998	Mitgliedschaft im „Wurzel-Verein zur Förderung der Mathematik an Schulen und Universitäten e. V.“
20. 03. 2003	Heirat mit Renate Stein, Ärztin