Algorithmic properties of modal logics with restricted languages

School of
Computer Science and Applied Mathematics
University of the Witwatersrand, Johannesburg

Mikhail Rybakov
1122321

Supervised by
Dr Dmitry Shkatov

May 23, 2019

A Thesis submitted to the Faculty of Science, University of the Witwatersrand, Johannesburg, in fulfillment of the requirements for the degree of Doctor of Philosophy

Preface

Abstract

Modal logics, both propositional and predicate, have been used in computer science since the late 1970s. One of the most important properties of modal logics of relevance to their applications in computer science is the complexity of their satisfiability problem. The complexity of satisfiability for modal logics is rather high: it ranges from NP-complete to undecidable for propositional logics and is undecidable for predicate logics. This has, for a long time, motivated research in drawing the borderline between tractable and intractable fragments of propositional modal logics as well as between decidable and undecidable fragments of predicate modal logics.

In the present thesis, we investigate some very natural restrictions on the languages of propositional and predicate modal logics and show that placing those restrictions does not decrease complexity of satisfiability. For propositional languages, we consider restricting the number of propositional variables allowed in the construction of formulas, while for predicate languages, we consider restricting the number of individual variables as well as the number and arity of predicate letters allowed in the construction of formulas. We develop original techniques, which build on and develop the techniques known from the literature, for proving that satisfiability for a finite-variable fragment of a propositional modal logic is as computationally hard as satisfiability for the logic in the full language and adapt those techniques to predicate modal logics and prove undecidability of fragments of such logics in the language with a finite number of unary predicate letters as well as restrictions on the number of individual variables.

The thesis is based on four articles published or accepted for publication. They concern propositional dynamic logics, propositional branching- and alternating-time temporal logics, propositional logics of symmetric rela-

tions, and first-order predicate modal and intuitionistic logics. In all cases, we identify the “minimal,” with regard to the criteria mentioned above, fragments whose satisfiability is as computationally hard as satisfiability for the entire logic.

Declaration

I declare that this thesis is my own, unaided work. It is being submitted for the Degree of Doctor of Philosophy at the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other University.

Mikhail Rybakov
May 23, 2019

Acknowledgements

I would like to thank the Faculty of Science, University of the Witwatersrand, Johannesburg, and the School of Computer Science and Applied Mathematics, University of the Witwatersrand, Johannesburg, for providing financial assistance that allowed me to complete this thesis.

I would also like to acknowledge the financial support provided by Russian Foundation for Basic Research, grants 16–07–01272, 17–03–00818, and 18–011–00869, while working on this thesis.

Contents

Chapter 1

Introduction

Modal logics, both propositional and predicate, have been used in computer science since the late 1970s (see, for example, [64, 24]), when it was realised that they provide languages for describing and reasoning about mathematical structures that very naturally model various phenomena of interest to computer scientists—transition systems corresponding to computational paths of terminating and non-terminating programs, abstract models of hardware such as CPUs, etc. Therefore, it was realised that these logics can be used in computer system design, particularly in those cases when empirical methods such as using a debugger to produce a correct parallel program, are not guaranteed to succeed.

Modal logics have two main applications to computer system design, corresponding to two different stages in the system design process, traditionally conceived of as having specification, implementation, and verification phases. First, the task of verifying that an implemented system conforms to a specification can be carried out by checking that a formula expressing the specification is satisfied in the structure modelling the system,—for program verification, this structure usually models execution paths of the program; this task corresponds to the model checking problem [14] for the logic. Second, the task of verifying that a specification of a system is satisfiable—and, thus, can be implemented by some system—corresponds to the satisfiability problem for the logic. Being able to check that a specification is satisfiable has the obvious advantage of avoiding wasted effort in trying to implement unsatisfiable systems. Moreover, an algorithm that checks for satisfiability of a formula expressing a specification builds, explicitly or implicitly, a model for the formula, thus supplying a formal model of a system conforming to the specification; this model can subsequently be used in the implementation phase. There is even hope that one day such models can be used as part of a "push-button" procedure producing an assuredly correct implementation from a specification model, avoiding the need for subsequent verification altogether. Satisfiability-checking algorithms, such as tableaux, developed for modal logics (see, for example, [20], [65], [35], [15]) all implicitly build a model for the formula whose satisfiability is being checked.

The main problem involved in using modal logics in formal specification of computer systems is that the computational complexity of checking satisfiability of modal formulas is usually quite high. For propositional languages, it ranges from NP-complete to undecidable, with any degree of undecidability, while for predicate languages it is always undecidable, since such languages extend the language of the classical first-order logic, which is known to be undecidable [12]. This has stimulated the search for tractable fragments of propositional modal logics (see, for example, [40]) as well as for decidable fragments of first-order modal logics (see, for example, [85]). On the other side, not less important has been work devoted to finding fragments of those logics that are as algorithmically hard as the entire logics (see, for example, [8], [40], [18], [10], [82]), since this delineates fragments that should be be definitely avoided in the search for fragments which desirable computational properties.

When looking for fragments of modal logics with both desirable and undesirable computational properties, a variety of criteria have been used in delineating such fragments. For propositional languages, the main criteria looked at have been the depth of nesting of modal connectives (see, e.g., [40]), the number of propositional variables used in the construction of formulas (see, e.g., [8, 40]), and the pattern of use of modal connectives (see, e.g., [56]), while for predicate languages it has mostly been the number of predicate letters and propositional variables used in the construction of formulas (see, e.g., [85, 47]).

1.1 Aim

The main aim of the present thesis is to develop techniques and methods for establishing that a fragment of a propositional or a predicate modal logic is as algorithmically hard as the entire logic. In particular, we will be looking at those fragments that are obtained by restricting vocabularies of the languages of logics in question. For propositional logics this means restricting the number of propositional variables allowed in the construction of formulas, while for predicate logics this means restricting the number and arity of predicate letters, as well as the number of individual variables, allowed in the construction of formulas.

The main purpose of developing these techniques is the improvement on the similar results known from the literature, as well as proving new results, most of which could not have been obtained with the use of the known techniques.

We are not, however, going to be describing techniques and methods in the abstract; rather, we will be applying them to the particular cases, obtaining new results. The technique used will be illustrated in action rather than described in the vacuum.

1.2 Hypotheses and questions

Our main hypothesis is that, for most propositional modal logics of interest in computer science, complexity of the satisfiability problem is as hard for a fragment that contains only a very small number of propositional variables—one or zero, in most cases—as it is for these logics in languages with an unlimited supply of propositional variables. This appears to be surprising, or unexpected, in many cases.

Another hypothesis is that, in predicate modal logic, very tight restrictions on the language generate fragments that are undecidable—in most cases, it suffices to have one monadic predicate letter and two individual variables in the language to generate an undecidable fragment (this is an improvement on a result from [47], which shows that two-variable fragments are undecidable, with much weaker restrictions on the number and arity of predicate letters).

The main question for our investigation is for how many formalisms our hypotheses hold. We are also angling towards the question, without addressing it in this thesis, of where exactly the borderline is between logics that are well-behaved in the sense that restricting their languages to finite-variable fragments, in the propositional case, produces computationally tractable fragments (see, for example, [59] and [40]) and logics that have the intractable satisfiability problem even in the presence of a very small number of propositional variables. Similarly, there arises a question of whether it is possible to draw a sharper divide between decidable (see [85]) and undecidable fragments of predicate modal logics, as there is a small, albeit a noticeable gap between the fragments described in [85] and those we investigate.

1.3 Methodology

We have developed our own methodology for proving undecidability and complexity results for propositional and predicate modal logics. Our methodology is built upon, and is a refinement of, earlier work done by Joseph Halpern in [40].

We now briefly explain the main limitations of the methodology used in [40] and how our techniques overcome those limitations.

The approach of [40] is to model propositional variables by (the so-called pp-like) formulas of a single variable; to establish the PSPACE-harness results presented in [40], a substitution is made of such pp-like formulas for propositional variables into formulas encoding a PSPACE-hard problem. In the case of logics containing modalities corresponding to transitive relations, such as the modal logic **S4**, for such a substitution to work, the formulas into which the substitution is made need to satisfy the property referred to in [40] as "evidence in a structure,"—a formula is evident in a structure if it has a model satisfying the following heredity condition: if a propositional variable is true at a state, it has to be true at all the states accessible from that state. In the case of PSPACE-complete logics, formulas satisfying the evidence condition can always be found, as the intuitionistic logic, which is PSPACE-complete, has the heredity condition built into its semantics. The situation is drastically different for logics that are EXPTIME-hard, which is the case for most propositional modal logics used in computer science; the examples include propositional dynamic logics, branching-time temporal logics, and alternating-time temporal logics. To show that a logic is EXPTIME-hard, one uses formulas that require for their satisfiability chains of states of the length exponential in the size of the formula,—it is not clear how to achieve this with formulas that are evident in a structure, as by varying the valuations of propositional variables that have to satisfy the heredity condition we can only describe chains whose length is linear in the size of the formula. Thus, the technique from [40] is not directly applicable to EXPTIME-hard logics with "transitive" modalities, as the formulas into which the substitution of pp-like formulas needs to be made do not satisfy the condition that has to be met for such a substitution to work. As most the logics of interest to computer science—i.e., propositional dynamic logics, branching-time temporal logics, and alternating-time temporal logics—do have a "transitive" modality—namely, the temporal connective "always in the future", which is interpreted by the reflexive, transitive closure of the relation corresponding to the temporal connective "at the next instance" as well as the iteration modality of propositional dynamic logics—this limitation prevents the technique from [40] from being directly applied to them.

We modify the approach of [40] by coming up with substitutions of single-variable formulas for propositional variables that can be made into arbitrary formulas, rather than formulas satisfying a particular property, such as evidence in a structure. This allows us to break away from the class PSPACE and to deal with such logics as **PDL**, **CTL**, **CTL***, **ATL**, and **ATL***, all of which are at least EXPTIME-hard.

The flexibility of our methodology allows it to be applied in the settings where other methodologies are more cumbersome to deal with. Thus, in the realm of PSPACE-hard logics, while logics describing structures with reflexive, transitive, serial, and Euclidean relations have been investigated, no study has been made of logics describing symmetric relations. We use our methodology for the study of such logics. Also, our methodology can be readily applied to various predicate modal logics.

1.4 The structure of the thesis

The present thesis comprises, in the form of distinct chapters, the content of four articles that have been either officially published or pre-published with a DOI number, awaiting publication in an issue of a journal. These articles are:

- Mikhail Rybakov and Dmitry Shkatov. Complexity and expressivity of propositional dynamic logics with finitely many variables. *Logic Journal of the IGPL*, 26 (5), 2018, pp. 539–547.
- Mikhail Rybakov and Dmitry Shkatov. Complexity and expressivity of branching- and alternating-time temporal logics with finitely many variables. In Fischer B., Uustalu T. (eds) *Theoretical Aspects of Com-*

puting - ICTAC 2018. Lecture Notes in Computer Science, Vol. 11187. Springer, 2018, pp. 396–414.

- Mikhail Rybakov and Dmitry Shkatov. Complexity of finite-variable fragments of propositional modal logics of symmetric frames. *Logic Journal of the IGPL*, 27 (1), 2019, pp. 60–68.

- Mikhail Rybakov and Dmitry Shkatov. Undecidability of first-order modal and intuitionistic logics with two variables and one monadic predicate letter. To appear in *Studia Logica*. DOI 10.1007/s11225-018-9815-7.

We have made no changes to the content of the articles, with two exceptions. First, the abstracts of all the articles have been removed and the bibliographies of all the articles have been combined to conform to the formal requirements of a PhD thesis. Secondly, for stylistic consistency, the word "paper" has, in every chapter, been replaced by "chapter". Finally, we have corrected a few typos noticed after the publication of the articles. Each of the original articles, thus, retains its own introduction and conclusion. On the downside, this makes the introductions repetitive in places; on the upside, this retains the integrity of the original articles. The order of the chapters, the same as in the list above, has been chosen so that our methods are seen in action from the simpler to the more complicated settings. The headings of the chapters coincide with the titles of the articles.

Chapter 2

Complexity and expressivity of propositional dynamic logics with finitely many variables

2.1 Introduction

The propositional dynamic logic, PDL, introduced in [24], has ever since been used for reasoning about the input-output behaviour of terminating programs. Over the years, it has been extended in various ways to deal with a wider variety of terminating programs [75, 43, 79, 62, 30]. Also, various formalisms closely linked to PDL have been developed for applications in areas other than reasoning about programs; among them are knowledge representation [28, 19, 55], querying semistructured data [1], data analysis [16], and linguistics [48].

Clearly, the complexity of satisfiability—equivalently, validity—problem for all of these variants of PDL is of crucial importance to their applications in the above-mentioned domains. Typically, for formulas containing an arbitrary number of propositional variables, the complexity of satisfiability problem for variants of PDL is rather high: it ranges from EXPTIME-complete [24] to undecidable [4].

It has, however, been observed that, in practice, one rarely uses formulas containing a large number of propositional variables—usually, this number is rather small. This raises the question of whether the complexity of satisfiability for PDL can be tamed by restricting the language to a finite number of propositional variables. Such an effect is not, after all, entirely unknown: for many logics, the complexity of satisfiability goes down from “intractable” to “tractable” once we place a limit on the number of propositional variables that can be used in the construction of formulas. For the classical propositional logic, as well as for the normal extensions of the modal logic **K5** [59], which include logics **K45**, **KD45**, and **S5** (see also [40]), the complexity of satisfiability goes down from NP-complete to polynomial-time computable once we restrict the number of propositional variables to any finite number. Similarly, as follows from [61], the complexity of satisfiability for intuitionistic propositional logic goes down from PSPACE-complete to polynomial-time computable if we consider only formulas of one variable.

The main contribution of the present chapter is to show that for propositional dynamic logics this route to reducing the complexity of satisfiability seems to be closed: even formulas built out of propositional constants, and thus containing no propositional variables at all, are as hard to test for satisfiability as formulas with an arbitrary number of propositional variables.

We suspect that this behaviour is representative of PDL-style logics. It would, however, be difficult to make an exhaustive case, given a wild proliferation of such formalisms. What we do instead is pick three examples that are representative in the sense of their satisfiability problems belonging to three representative complexity classes (broadly understood, i.e., treating “undecidable” as a complexity class); namely, we consider regular PDL, which has an EXPTIME-complete satisfiability problem [24], PDL with intersection, which has a 2EXPTIME-complete satisfiability problem [52], and PDL with parallel composition, which has an undecidable satisfiability problem [4]. We show that satisfiability problem for the variable-free fragment of each of these logics is as hard as for the entire logic. Moreover, we show that this is a consequence of the richness of the expressive power of variable-free fragments: for all the logics we consider, variable-free fragments are as semantically expressive as entire logics.

Similar results for other propositional modal logics have been obtained in [8], [40], [44], [18], [82], and [10]. The techniques used in those studies are not directly applicable to obtain the results presented in this chapter; we do, however, substantially draw on the ideas from [8] and [40].

The chapter is organised as follows. In section 2.2, we recall the syntax and semantics of the logics we consider. Then, in section 2.3, we present our results about complexity and expressivity of their variable-free fragments. We conclude in section 2.4.

2.2 Syntax and semantics

In this section, we recall the syntax and semantics of PDL with intersection (**IPDL**), regular PDL (**PDL**), and PDL with parallel composition (**PRSPDL**).

The language of **IPDL** contains a countable set $\mathit{Var} = \{p_1, p_2, \ldots\}$ of propositional variables, the propositional constant $\bot$ ("falsehood"), the Boolean connective $\to$, and modalities of the form $[\,\alpha\,]$, where α ranges over program terms built out of a countable set $AP = \{a_1, a_2, \ldots\}$ of atomic program terms as well as formulas, using the operations ? (test), ; (composition), $\cup$ (choice), $\cap$ (intersection), and * (iteration). The intended meaning of the formula $[\,\alpha\,]\,\varphi$ is that every execution of the program α at the current state results in a state where φ holds. Formulas φ and program terms α are simultaneously defined by the following BNF expressions:

$$\varphi := p \mid \bot \mid (\varphi \to \varphi) \mid [\,\alpha\,]\,\varphi,$$

$$\alpha := a \mid \varphi? \mid (\alpha\,;\,\alpha) \mid (\alpha \cup \alpha) \mid (\alpha \cap \alpha) \mid \alpha^*,$$

where p ranges over Var and a ranges over AP. The other connectives are defined as usual. Formulas are evaluated in Kripke models. A Kripke model is a tuple $\mathfrak{M} = (\mathcal{S}, \{\mathcal{R}_a\}_{a \in AP}, V)$, where $\mathcal{S}$ is a non-empty set (of states), $\mathcal{R}_a$ is a binary (accessibility) relation on $\mathcal{S}$, and V is a (valuation) function $V : \mathit{Var} \to 2^{\mathcal{S}}$. Accessibility relations for non-atomic program terms as well as the satisfaction relation between models, states, and formulas are defined by simultaneous induction as follows:

- $(s,t) \in \mathcal{R}_{\varphi?} \leftrightharpoons s = t$ and $\mathfrak{M}, s \models \varphi$;
- $(s,t) \in \mathcal{R}_{\alpha\,;\,\beta} \leftrightharpoons (s,u) \in \mathcal{R}_\alpha$ and $(u,t) \in \mathcal{R}_\beta$, for some $u \in \mathcal{S}$;
- $(s,t) \in \mathcal{R}_{\alpha\cup\beta} \leftrightharpoons (s,t) \in \mathcal{R}_\alpha$ or $(s,t) \in \mathcal{R}_\beta$;
- $(s,t) \in \mathcal{R}_{\alpha\cap\beta} \leftrightharpoons (s,t) \in \mathcal{R}_\alpha$ and $(s,t) \in \mathcal{R}_\beta$;
- $(s,t) \in \mathcal{R}_{\alpha^*} \leftrightharpoons (s,t) \in \mathcal{R}_\alpha^*$, where $\mathcal{R}_\alpha^*$ is the reflexive, transitive closure of $\mathcal{R}_\alpha$;
- $\mathfrak{M}, s \models p_i \leftrightharpoons s \in V(p_i)$;
- $\mathfrak{M}, s \models \bot$ never holds;

- $\mathfrak{M}, s \models \varphi \to \psi \leftrightharpoons \mathfrak{M}, s \models \varphi$ implies $\mathfrak{M}, s \models \psi$;
- $\mathfrak{M}, s \models [\,\alpha\,]\,\varphi \leftrightharpoons \mathfrak{M}, t \models \varphi$ whenever $(s, t) \in \mathcal{R}_\alpha$.

A formula is satisfiable if it is satisfied at some state of some model. A formula is valid if it is satisfied by every state of every model. Formally, by **IPDL**, we mean the set of all valid formulas in this language.

The language of **PDL** differs from that of **IPDL** in that it does not contain program operations $\cap$ and ?. The semantics is modified accordingly.

The language of **PRSPDL** is interpreted on models made up of states possessing inner structure: a state s is a composition $x * y$ of states x and y if s can be separated into components x and y; in general, there is no requirement that $*$ be a function on the set of states. The program terms are formed out of atomic program terms as well as four special program terms r_1, r_2 (recovery of the first and second $*$-components, respectively, of a state), s_1, and s_2 (storing a state as the first and second $*$-components, respectively, of a composite state), using the operations ? (test), * (iteration), and $||$ (parallel composition). Note that the language of **PRSPDL** does not contain the operation of union of program terms. A Kripke model is a tuple $\mathfrak{M} = (\mathcal{S}, \{\mathcal{R}_a\}_{a \in AP}, *, V)$, where $\mathcal{S}$, $\mathcal{R}_a$, and V have the same meaning as in Kripke models for **IPDL**, and $*$ is a function $\mathcal{S} \times \mathcal{S} \to 2^{\mathcal{S}}$. The meaning of $||$, r_1, r_2, s_1, and s_2 is given by the following clauses:

- $(s, t) \in \mathcal{R}_{\alpha\,||\,\beta} \leftrightharpoons$ there exist $x_1, y_1, x_2, y_2 \in \mathcal{S}$ such that $s \in x_1 * x_2$, $t \in y_1 * y_2$, $(x_1, y_1) \in \mathcal{R}_\alpha$, and $(x_2, y_2) \in \mathcal{R}_\beta$;
- $(s, t) \in \mathcal{R}_{r_1} \leftrightharpoons$ there exists $u \in \mathcal{S}$ such that $s \in t * u$;
- $(s, t) \in \mathcal{R}_{r_2} \leftrightharpoons$ there exists $u \in \mathcal{S}$ such that $s \in u * t$;
- $(s, t) \in \mathcal{R}_{s_1} \leftrightharpoons$ there exists $u \in \mathcal{S}$ such that $t \in s * u$;
- $(s, t) \in \mathcal{R}_{s_2} \leftrightharpoons$ there exists $u \in \mathcal{S}$ such that $t \in u * s$.

The models thus defined are referred to in [4] as "$*$-separated." The authors of [4] consider a number of logics in the same language, which differ in the conditions placed on the function $*$ in their semantics. For our purposes, it suffices to consider only one of the logics from [4],—the rest can be dealt with in a similar way.

The notions of satisfiability and validity are defined as for **IPDL** and **PDL**.

For each of the logics we consider, by a variable-free fragment we mean the subset of the logic containing only variable-free formulas—i.e., formulas not containing any propositional variables. Given formulas φ, ψ and a propositional variable p, we denote by $\varphi(p/\psi)$ the result of uniformly substituting ψ for p in φ.

2.3 Finite-variable fragments

In this section, we show that variable-free fragments of **IPDL**, **PDL**, and **PRSPDL** have the same expressive power and computational complexity as the entire logics, by embedding each logic into its variable-free fragment; in the case of **IPDL** and **PDL**, the embeddings are polynomial-time computable. We initially work with **IPDL** and subsequently point out how that work carries over to **PDL** and **PRSPDL**.

Let φ be an arbitrary **IPDL**-formula. Assume that φ only contains propositional variables $p_1, \ldots, p_n$ and atomic program terms $a_1, \ldots, a_l$. Let $\gamma = a_1 \cup \ldots \cup a_l$. First, recursively define translation $\cdot'$ as follows:

$$\begin{array}{lcl}
{a_j}' & = & a_j, \quad \text{where } j \in \{1, \ldots, l\}; \\
(\alpha\,;\,\beta)' & = & \alpha'\,;\,\beta'; \\
(\alpha \cup \beta)' & = & \alpha' \cup \beta'; \\
(\alpha \cap \beta)' & = & \alpha' \cap \beta'; \\
(\alpha^*)' & = & (\alpha')^*; \\
(\phi?)' & = & (\phi')?; \\
{p_i}' & = & p_i, \quad \text{where } i \in \{1, \ldots, n\}; \\
(\bot)' & = & \bot; \\
(\phi \to \psi)' & = & \phi' \to \psi'; \\
([\,\alpha\,]\,\phi)' & = & [\,\alpha'\,]\,(p_{n+1} \to \phi').
\end{array}$$

Second, define

$$\Theta = p_{n+1} \wedge [\,\gamma^*\,]\,(\langle\gamma\rangle\, p_{n+1} \to p_{n+1}).$$

Finally, let

$$\widehat{\varphi} = \Theta \wedge \varphi'.$$

Lemma 2.3.1 *Formula φ is satisfiable if, and only if, formula $\widehat{\varphi}$ is satisfiable.*

Proof. Suppose $\widehat{\varphi}$ is not satisfiable. Then, $\neg\widehat{\varphi} \in$ **IPDL** and, since **IPDL** is closed under substitution, $\neg\widehat{\varphi}(p_{n+1}/\top) \in$ **IPDL**. As $\widehat{\varphi}(p_{n+1}/\top) \leftrightarrow \varphi \in$ **IPDL**, we have $\neg\varphi \in$ **IPDL**; thus, φ is not satisfiable.

Suppose that $\widehat{\varphi}$ is satisfiable. In particular, let $\mathfrak{M}, s_0 \models \widehat{\varphi}$ for some model $\mathfrak{M}$ and some s_0 in $\mathfrak{M}$. Define $\mathfrak{M}'$ to be the smallest submodel of $\mathfrak{M}$ such that

- s_0 is in $\mathfrak{M}'$;
- if x is in $\mathfrak{M}'$, $x\mathcal{R}_\gamma y$, and $\mathfrak{M}, y \models p_{n+1}$, then y is also in $\mathfrak{M}'$.

Notice that p_{n+1} is universally true in $\mathfrak{M}'$. It is straightforward to show that, for every subformula ψ of φ and every s in $\mathfrak{M}'$, we have $\mathfrak{M}, s \models \psi'$ if, and only if, $\mathfrak{M}', s \models \psi$. As $\mathfrak{M}, s_0 \models \varphi'$, this gives us $\mathfrak{M}', s_0 \models \varphi$; hence, φ is satisfiable. ❑

Remark 2.3.2 *It follows from the proof of Lemma 2.3.1 that, if $\widehat{\varphi}$ is satisfiable, then it is satisfiable in a model where p_{n+1} is universally true. Indeed, if $\widehat{\varphi}$ is satisfiable, then φ is satisfiable in a model where p_{n+1} is universally true. The claim follows from the fact that φ is equivalent to $\widehat{\varphi}(p_{n+1}/\top)$.*

Now, consider the following class M of finite models. Let b be the lexicographically first atomic program term of φ if φ contains such terms; otherwise, let b be a_1. For every $m \in \{1, \dots, n+1\}$, where $p_1, \dots, p_n$ are the variables in φ, class M contains a unique member $\mathfrak{M}_m$, defined as follows: $\mathfrak{M}_m = (\mathcal{S}_m, \{\mathcal{R}_a\}_{a \in AP}, V_m)$, where

- $\mathcal{S}_m = \{r_m, t^m, s_1^m, s_2^m, \dots, s_m^m\}$;
- $\mathcal{R}_b$ is the transitive closure of the relation $\{\langle r_m, t^m\rangle, \langle t^m, t^m\rangle, \langle r_m, s_1^m\rangle\} \cup \{\langle s_i^m, s_{i+1}^m\rangle : 1 \leqslant i \leqslant m-1\}$;
- $\mathcal{R}_a = \varnothing$ if $a \neq b$;
- $V_m(p) = \varnothing$ for every $p \in \mathit{Var}$.

The model $\mathfrak{M}_m$ is depicted in Figure 2.1, where arrows represent $\mathcal{R}_b$; to avoid clutter, arrows are omitted whenever the presence of $\mathcal{R}_b$ can be deduced from its transitivity; the circle represents a state related by $\mathcal{R}_b$ to itself, and solid dots represent states without such loops.

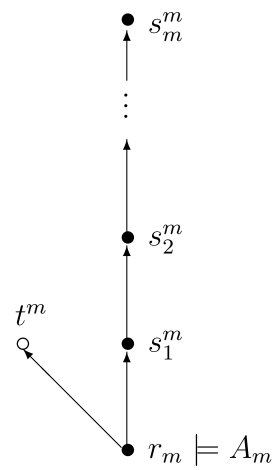

Figure 2.1: Model $\mathfrak{M}_m$

We now define formulas that will be true at the roots of models from M. For $j \geqslant 0$, inductively define the formula $\langle b\rangle^j\psi$ as follows: $\langle b\rangle^0\psi = \psi$; $\langle b\rangle^{k+1}\psi = \langle b\rangle\,\langle b\rangle^k\psi$. Next, for every $m \in \{1, \ldots, n+1\}$, define

$$A_m = \langle b\rangle^m[\,b\,]\perp \wedge \neg\langle b\rangle^{m+1}[\,b\,]\perp \wedge \langle b\rangle\,(\langle b\rangle\top \wedge [\,b\,]\,\langle b\rangle\top).$$

Lemma 2.3.3 *Let $\mathfrak{M}_k \in \mathsf{M}$ and let x be a state in $\mathfrak{M}_k$. Then, $\mathfrak{M}_k, x \models A_m$ if, and only if, $k = m$ and $x = r_m$.*

Proof. Straightforward. ❑

Now, define

$$B_m = \langle b\rangle\,A_m.$$

Let σ be a (substitution) function that, given an **IPDL**-formula ψ, replaces all occurrences of p_i in ψ by B_i, where $1 \leqslant i \leqslant n+1$. Finally, define

$$\varphi^* = \sigma(\widehat{\varphi})$$

to produce a variable-free formula φ^*.

Lemma 2.3.4 *Formula φ is satisfiable if, and only if, formula φ^* is satisfiable.*

Proof. Suppose that φ is not satisfiable. Then, by Lemma 2.3.1, $\widehat{\varphi}$ is not satisfiable, either, and hence $\neg\widehat{\varphi} \in$ **IPDL**. Since **IPDL** is closed under substitution, $\neg\varphi^* \in$ **IPDL** and, thus, φ^* is not satisfiable.

Suppose that φ is satisfiable. Then, in view of Lemma 2.3.1 and Remark 2.3.2, $\mathfrak{M}, s_0 \models \widehat{\varphi}$ for some $\mathfrak{M}$ such that p_{n+1} is true at every state of $\mathfrak{M}$ and some s_0 in $\mathfrak{M}$. Define model $\mathfrak{M}'$ as follows. Attach to $\mathfrak{M}$ all the models from M; then, for every x in $\mathfrak{M}$, put $x\mathcal{R}_b r_m$ (where r_m is the root of $\mathfrak{M}_m \in \mathsf{M}$) exactly when $\mathfrak{M}, x \models p_m$. Notice that r_{n+1} is accessible in $\mathfrak{M}'$ from every x in $\mathfrak{M}$.

To conclude the proof, it suffices to show that $\mathfrak{M}', s_0 \models \varphi^*$. It is easy to check that $\mathfrak{M}', s_0 \models \sigma(\Theta)$. It then remains to show that $\mathfrak{M}', s_0 \models \sigma(\varphi')$. To that end, it suffices to show that $\mathfrak{M}, x \models \psi'$ if, and only if, $\mathfrak{M}', x \models \sigma(\psi')$, for every subformula ψ of φ and every x in $\mathfrak{M}$. This can be done by induction on ψ; we only consider the base case, leaving the rest to the reader.

Let $\mathfrak{M}', x \models B_i$. Then, for some y in $\mathfrak{M}'$, we have $x\mathcal{R}'_b y$ and $\mathfrak{M}', y \models A_i$. This is only possible if y is not in $\mathfrak{M}$. Indeed, suppose otherwise. Then, $\mathfrak{M}', y \models p_{n+1}$, and therefore, $y\mathcal{R}'_b r_{n+1}$. Hence, $\mathfrak{M}', y \models \langle b \rangle^{i+1}[\, b\,]\,\bot$, and therefore, $\mathfrak{M}', y \not\models A_i$, resulting in a contradiction. Thus, y is in $\mathfrak{M}_m$, for some $m \in \{1, \ldots, n+1\}$. Then, by Lemma 2.3.3, $y = r_i$, and therefore, by definition of $\mathfrak{M}'$, we have $\mathfrak{M}, x \models p_i$. The other direction is straightforward. ❑

Theorem 2.3.5 *There exists an embedding that embeds* **IPDL** *into its variable-free fragment in polynomial time.*

We now look at the complexity-theoretic implications of Theorem 2.3.5. It has been shown in [52] that the fragment of **IPDL** containing a single atomic program term is 2EXPTIME-complete. This gives us the following:

Theorem 2.3.6 *The satisfiability problem for the fragment of* **IPDL** *containing variable-free formulas with a single atomic program term is* 2EXPTIME-*complete.*

We now point out how the work we have done so far for **IPDL** carries over to **PDL** and **PRSPDL**.

It is easy to check that the construction presented above works for **PDL**, as well, if we omit the details peculiar to **IPDL**. This gives us the following:

Theorem 2.3.7 *There exists an embedding that embeds* **PDL** *into its variable-free fragment in polynomial time.*

Since satisfiability problem for **PDL** with a single atomic program term is EXPTIME-complete [24], we have the following:

Theorem 2.3.8 *The satisfiability problem for the fragment of* **PDL** *containing variable-free formulas with a single atomic program term is* EXPTIME-*complete.*

We next show how to modify the above argument for **PRSPDL**. We remind the reader that we confine our attention to **PRSPDL** over $*$-separated models; other variants of this formalism considered in [4] can be treated in essentially the same way. We first need to construct the analogue of formula $\widehat{\varphi}$. It is straightforward to define the translation $\cdot'$:

$$
\begin{array}{lcl}
{a_i}' & = & a_i, \quad \text{where } i \in \{1, \ldots, l\}; \\
{r_i}' & = & r_i, \quad \text{where } i \in \{1, 2\}; \\
{s_i}' & = & s_i, \quad \text{where } i \in \{1, 2\}; \\
(\alpha\,;\,\beta)' & = & \alpha'\,;\,\beta'; \\
(\alpha\,||\,\beta)' & = & \alpha'\,||\,\beta' \\
(\alpha^*)' & = & (\alpha')^*; \\
(\phi?)' & = & (\phi')?; \\
{p_i}' & = & p_i, \quad \text{where } i \in \{1, \ldots, n\}; \\
(\bot)' & = & \bot'; \\
(\phi \to \psi)' & = & \phi' \to \psi'; \\
([\,\alpha\,]\,\phi)' & = & [\,\alpha'\,]\,(p_{n+1} \to \phi').
\end{array}
$$

We next define the analogue of formula Θ. As **PRSPDL** does not have the operation of choice on program terms, we proceed as follows. Let

$$
\begin{array}{c}
\alpha_1^1 \ldots \alpha_{n_1}^1 \\
\ldots \\
\alpha_1^k \ldots \alpha_{n_k}^k
\end{array}
$$

be all sequences of nested program terms in φ. Then,

$$\Theta = p_{n+1} \wedge \bigwedge_{i=1}^{k} \bigwedge_{j=1}^{n_k-1} [\,\alpha_1^i\,] \ldots [\,\alpha_j^i\,] (\langle \alpha_{j+1}^i \rangle\, p_{n+1} \to p_{n+1}).$$

Finally, let

$$\widehat{\varphi} = \Theta \wedge \varphi'.$$

From here on, we argue exactly as in the case of **IPDL** to obtain the following:

Theorem 2.3.9 *There exists an embedding that embeds* **PRSPDL** *over $*$-separated models into its variable-free fragment.*

Theorem 2.3.10 *The variable-free fragment of* **PRSPDL** *over $*$-separated models is undecidable.*

Remark 2.3.11 *It is well-known that the consequence relation for propositional dynamic logics is not compact, as the formula $[\,a^*\,]\,\varphi$ follows from the infinite set $\{[\,a\,]^n\varphi : n \geqslant 0\}$ of formulas but not from any of its finite subsets; thus, the consequence relation is not reducible to satisfiability for formulas. The technique presented above can be used to reduce the consequence relation for the logics we have considered to the consequence relation for their variable-free fragments. To that end, unless the number of propositional variables occurring in the premises is finite, we need to use an extra atomic program term corresponding to the accessibility relation connecting the roots of the models attached in the proof of Lemma 2.3.4 to the original model. This is necessary as in the proof of Lemma 2.3.4 we relied on the variable p_{n+1}, used as a marker of the worlds of the original model, having the maximal index of all the variables of the formula φ.*

2.4 Conclusion

We have shown that for three variants of propositional dynamic logic representative of various complexity classes, broadly understood, the complexity of satisfiability remains the same if we restrict the language to formulas built out of propositional constants, i.e., without the use of propositional variables.

This is a consequence of the richness of the expressive power of the variable-free fragments—as we have shown, they are as expressive as the logics with an infinite supply of propositional variables.

We suspect that these results are representative of how PDL-style formalisms behave. If this is indeed so, the important question for future research is to find out if there are ways to tame the complexity of satisfiability that might be applicable en masse to a wide range of PDL-style logics and that might be of relevance to how these formalisms are applied in practice.

Chapter 3

Complexity and expressivity of branching- and alternating-time temporal logics with finitely many variables

3.1 Introduction

The propositional Branching-time temporal logics **CTL** [13, 17] and **CTL*** [21, 17] have for a long time been used in formal specification and verification of (parallel) non-terminating computer programs [46, 17], such as (components of) operating systems, as well as in formal specification and verification of hardware. More recently, Alternating-time temporal logics **ATL** and **ATL*** [2, 17] have been used for formal specification and verification of multi-agent [72] and, more broadly, so-called open systems, i.e., systems whose correctness depends on the actions of external entities, such as the environment or other agents making up a multi-agent system.

Logics **CTL**, **CTL***, **ATL**, and **ATL*** have two main applications to computer system design, corresponding to two different stages in the system design process, traditionally conceived of as having specification, implementation, and verification phases. First, the task of verifying that an implemented system conforms to a specification can be carried out by checking that a formula expressing the specification is satisfied in the structure modelling the system,—for program verification, this structure usually models execution paths of the program; this task corresponds to the model checking problem [14] for the logic. Second, the task of verifying that a specification of a system is satisfiable—and, thus, can be implemented by some system—corresponds to the satisfiability problem for the logic. Being able to check that a specification is satisfiable has the obvious advantage of avoiding wasted effort in trying to implement unsatisfiable systems. Moreover, an algorithm that checks for satisfiability of a formula expressing a specification builds, explicitly or implicitly, a model for the formula, thus supplying a formal model of a system conforming to the specification; this model can subsequently be used in the implementation phase. There is hope that one day such models can be used as part of a "push-button" procedure producing an assuredly correct implementation from a specification model, avoiding the need for subsequent verification altogether. Tableaux-style satisfiability-checking algorithms developed for **CTL** in [20], for **CTL*** in [65], for **ATL** in [35], and for **ATL*** in [15] all implicitly build a model for the formula whose satisfiability is being checked.

In this chapter, we are concerned with the satisfiability problem for **CTL**, **CTL***, **ATL**, and **ATL***; clearly, the complexity of satisfiability for these logics is of crucial importance to their applications to formal specification. It is well-known that, for formulas that might contain contain an arbitrary number of propositional variables, the complexity of satisfiability for all of these logics is quite high: it is EXPTIME-complete for **CTL** [24, 20], 2EXPTIME-complete for **CTL*** [80], EXPTIME-complete for **ATL** [37, 84], and 2EXPTIME-complete for **ATL*** [71].

It has, however, been observed (see, for example, [18]) that, in practice, formulas expressing formal specifications, despite being quite long and containing deeply nested temporal operators, usually contain only a very small number of propositional variables,—typically, two or three. The question thus arises whether limiting the number of propositional variables allowed to be used in the construction of formulas we take as inputs can bring down the complexity of the satisfiability problem for **CTL**, **CTL***, **ATL**, and **ATL***. Such an effect is not, after all, unknown in logic: examples are known of logics whose satisfiability problem goes down from "intractable" to "tractable" once we place a limit on the number of propositional variables allowed in the language: thus, satisfiability for the classical propositional logic as well as the extensions of the modal logic **K5** [59], which include such logics as **K45**, **KD45**, and **S5** (see also [40]), goes down from NP-complete to polynomial-time decidable once we limit the number of propositional variables in the language to an (arbitrary) finite number.[1] Similarly, as follows from [61], satisfiability for the intuitionistic propositional logic goes down from PSPACE-complete to polynomial-time decidable if we allow only a single propositional variable in the language.

The question of whether the complexity of satisfiability for **CTL**, **CTL***, **ATL**, and **ATL***can be reduced by restricting the number of propositional variables allowed to be used in the formulas has not, however, been investigated in the literature. The present chapter is mostly meant to fill that gap.

A similar question has been answered in the negative for Linear-time temporal logic **LTL** in [18], where it was shown, using a proof technique peculiar to **LTL** (in particular, [18] relies on the fact that for **LTL** with a finite number of propositional variables satisfiability reduces to model-checking), that a single-variable fragment of **LTL** is PSPACE-complete, i.e., as computationally hard as the entire logic [73]. It should be noted that, in this respect, **LTL** behaves like most "natural" modal and temporal logics, for which the presence of even a single variable in the language is sufficient to generate a fragment whose satisfiability is as hard as satisfiability for the entire logic. The first results to this effect have been proven in [8] for logics for reasoning about linguistic structures and in [82] for provability logic. A general method of proving such results for PSPACE-complete logics has been proposed in [40]; even though [40] considers only a handful of logics, the method can be generalised to large classes of logics, often in the language without propositional variables [44, 10] (it is not, however, applicable to **LTL**, as it relies on unrestricted branching in the models of the logic, which runs contrary to the semantics of **LTL**,—hence the need for a different approach, as in [18]). In this chapter, we use a suitable modification of the technique from [40] (see [68, 69]) to show that single-variable fragments of **CTL**, **CTL***, **ATL**, and **ATL*** are as computationally hard as the entire logics; thus, for these logics, the complexity of satisfiability cannot be reduced by restricting the number of variables in the language.

Before doing so, a few words might be in order to explain why the technique from [40] is not directly applicable to the logics we are considering

[1] To avoid ambiguity, we emphasise that we use the standard complexity-theoretic convention of measuring the complexity of the input as its size; in our case, this is the length of the input formula. In other words, we do not measure the complexity of the input according to how many distinct variables it contains; limiting the number of variables simply provides a restriction on the languages we consider.

in this chapter. The approach of [40] is to model propositional variables by (the so-called pp-like) formulas of a single variable; to establish the PSPACE-harness results presented in [40], a substitution is made of such pp-like formulas for propositional variables into formulas encoding a PSPACE-hard problem. In the case of logics containing modalities corresponding to transitive relations, such as the modal logic **S4**, for such a substitution to work, the formulas into which the substitution is made need to satisfy the property referred to in [40] as "evidence in a structure,"—a formula is evident in a structure if it has a model satisfying the following heredity condition: if a propositional variable is true at a state, it has to be true at all the states accessible from that state. In the case of PSPACE-complete logics, formulas satisfying the evidence condition can always be found, as the intuitionistic logic, which is PSPACE-complete, has the heredity condition built into its semantics. The situation is drastically different for logics that are EXPTIME-hard, which is the case for all the logics considered in the present chapter: to show that a logic is EXPTIME-hard, one uses formulas that require for their satisfiability chains of states of the length exponential in the size of the formula,—this cannot be achieved with formulas that are evident in a structure, as by varying the valuations of propositional variables that have to satisfy the heredity condition we can only describe chains whose length is linear in the size of the formula. Thus, the technique from [40] is not directly applicable to EXPTIME-hard logics with "transitive" modalities, as the formulas into which the substitution of pp-like formulas needs to be made do not satisfy the condition that has to be met for such a substitution to work. As all the logics considered in this chapter do have a "transitive" modality—namely, the temporal connective "always in the future", which is interpreted by the reflexive, transitive closure of the relation corresponding to the temporal connective "at the next instance"—this limitation prevents the technique from [40] from being directly applied to them.

In the present chapter, we modify the approach of [40] by coming up with substitutions of single-variable formulas for propositional variables that can be made into arbitrary formulas, rather than formulas satisfying a particular property, such as evidence in a structure. This allows us to break away from the class PSPACE and to deal with **CTL**, **CTL***, **ATL**, and **ATL***, all of which are at least EXPTIME-hard. A similar approach has recently been used in [68] and [69] for some other propositional modal logics.

A by-product of our approach, and another contribution of this chapter, is that we establish that single-variable fragments of **CTL**, **CTL***, **ATL**, and

ATL* are as semantically expressive as the entire logic, i.e., all properties that can be specified with any formula of the logic can be specified with a formula containing only one variable—indeed, our complexity results follow from this. In this light, the observation cited above—that in practice most properties of interest are expressible in these logics using only a very small number of variables—is not at all surprising from a purely mathematical point of view, either.

The chapter is structured as follows. In Section 3.2, we introduce the syntax and semantics of **CTL** and **CTL***. Then, in Section 3.3, we show that **CTL** and **CTL*** can be polynomial-time embedded into their single-variable fragments. As a corollary, we obtain that satisfiability for the single variable fragment of **CTL** is EXPTIME-complete and satisfiability for the single variable of of **CTL*** is 2EXPTIME-complete. In Section 3.4, we introduce the syntax and semantics of **ATL** and **ATL***. Then, in Section 3.5, we prove results for **ATL** and **ATL*** that are analogous to those proven in Section 3.3 for **CTL** and **CTL***. We conclude in Section 3.6 by discussing other formalisms related to the logics considered in this chapter to which our proof technique can be applied to obtain similar results.

3.2 Branching-time temporal logics

We start by briefly recalling the syntax and semantics of **CTL** and **CTL***.

The language of **CTL*** contains a countable set $\mathit{Var} = \{p_1, p_2, \ldots\}$ of propositional variables, the propositional constant $\bot$ ("falsehood"), the Boolean connective $\rightarrow$ ("if ..., then ..."), the path quantifier $\forall$, and temporal connectives $\bigcirc$ ("next") and $\mathcal{U}$ ("until"). The language contains two kinds of formulas: state formulas and path formulas, so called because they are evaluated in the models at states and paths, respectively. State formulas φ and path formulas ϑ are simultaneously defined by the following BNF expressions:

$$\varphi ::= p \mid \bot \mid (\varphi \rightarrow \varphi) \mid \forall \vartheta,$$

$$\vartheta ::= \varphi \mid (\vartheta \rightarrow \vartheta) \mid (\vartheta \,\mathcal{U} \vartheta) \mid \bigcirc \vartheta,$$

where p ranges over Var. Other Boolean connectives are defined as follows: $\neg A := (A \rightarrow \bot)$, $(A \wedge B) := \neg(A \rightarrow \neg B)$, $(A \vee B) := (\neg A \rightarrow B)$, and $(A \leftrightarrow B) := (A \rightarrow B) \wedge (B \rightarrow A)$, where A and B can be either state or path formulas. We also define $\top := \bot \to \bot$, $\Diamond \vartheta := (\top \mathcal{U} \vartheta)$, $\Box \vartheta := \neg \Diamond \neg \vartheta$, and $\exists \vartheta := \neg \forall \neg \vartheta$.

Formulas are evaluated in Kripke models. A Kripke model is a tuple $\mathfrak{M} = (\mathcal{S}, \longmapsto, V)$, where $\mathcal{S}$ is a non-empty set (of states), $\longmapsto$ is a binary (transition) relation on $\mathcal{S}$ that is serial (i.e., for every $s \in \mathcal{S}$, there exists $s' \in \mathcal{S}$ such that $s \longmapsto s'$), and V is a (valuation) function $V : \mathit{Var} \to 2^{\mathcal{S}}$.

An infinite sequence $s_0, s_1, \ldots$ of states in $\mathfrak{M}$ such that $s_i \longmapsto s_{i+1}$, for every $i \geqslant 0$, is called a *path*. Given a path π and some $i \geqslant 0$, we denote by $\pi[i]$ the ith element of π and by $\pi[i, \infty]$ the suffix of π beginning at the ith element. If $s \in \mathcal{S}$, we denote by $\Pi(s)$ the set of all paths π such that $\pi[0] = s$.

The satisfaction relation between models $\mathfrak{M}$, states s, and state formulas φ, as well as between models $\mathfrak{M}$, paths π, and path formulas ϑ, is defined as follows:

- $\mathfrak{M}, s \models p_i \leftrightharpoons s \in V(p_i)$;
- $\mathfrak{M}, s \models \bot$ never holds;
- $\mathfrak{M}, s \models \varphi_1 \to \varphi_2 \leftrightharpoons \mathfrak{M}, s \models \varphi_1$ implies $\mathfrak{M}, s \models \varphi_2$;
- $\mathfrak{M}, s \models \forall \vartheta_1 \leftrightharpoons \mathfrak{M}, \pi \models \vartheta_1$ for every $\pi \in \Pi(s)$.
- $\mathfrak{M}, \pi \models \varphi_1 \leftrightharpoons \mathfrak{M}, \pi[0] \models \varphi_1$;
- $\mathfrak{M}, \pi \models \vartheta_1 \to \vartheta_2 \leftrightharpoons \mathfrak{M}, \pi \models \vartheta_1$ implies $\mathfrak{M}, \pi \models \vartheta_2$;
- $\mathfrak{M}, \pi \models \bigcirc \vartheta_1 \leftrightharpoons \mathfrak{M}, \pi[1, \infty] \models \vartheta_1$;
- $\mathfrak{M}, \pi \models \vartheta_1 \mathcal{U} \vartheta_2 \leftrightharpoons \mathfrak{M}, \pi[i, \infty] \models \vartheta_2$ for some $i \geqslant 0$ and $\mathfrak{M}, \pi[j, \infty] \models \vartheta_1$ for every j such that $0 \leqslant j < i$.

A **CTL***-formula is a state formula in this language. A **CTL***-formula is satisfiable if it is satisfied by some state of some model, and valid if it is satisfied by every state of every model. Formally, by **CTL*** we mean the set of valid **CTL***-formulas. Notice that this set is closed under uniform substitution.

Logic **CTL** can be thought of as a fragment of **CTL*** containing only formulas where a path quantifier is always paired up with a temporal connective. This, in particular, disallows formulas whose main sign is a temporal connective and, thus, eliminates path-formulas. Such composite "modal" operators are $\forall\bigcirc$ (universal "next"), $\forall\,\mathcal{U}$ (universal "until"), and $\exists\,\mathcal{U}$ (existential "until"). Formulas are defined by the following BNF expression:

$$\varphi ::= p \mid \bot \mid (\varphi \to \varphi) \mid \forall\bigcirc\varphi \mid \forall\,(\varphi\,\mathcal{U}\varphi) \mid \exists\,(\varphi\,\mathcal{U}\varphi),$$

where p ranges over Var. We also define $\neg\varphi := (\varphi \to \bot)$, $(\varphi\wedge\psi) := \neg(\varphi \to \neg\psi)$, $(\varphi\vee\psi) := (\neg\varphi \to \psi)$, $\top = \bot \to \bot$, $\exists\bigcirc\varphi := \neg\forall\bigcirc\neg\varphi$, $\exists\Diamond\varphi := \exists(\top\,\mathcal{U}\varphi)$, and $\forall\Box\varphi := \neg\exists\Diamond\neg\varphi$.

The satisfaction relation between models $\mathfrak{M}$, states s, and formulas φ is inductively defined as follows (we only list the cases for the "new" modal operators):

- $\mathfrak{M}, s \models \forall\bigcirc\varphi_1 \leftrightharpoons \mathfrak{M}, s' \models \varphi_1$ whenever $s \longmapsto s'$;
- $\mathfrak{M}, s \models \forall(\varphi_1\mathcal{U}\varphi_2) \leftrightharpoons$ for every path $s_0 \longmapsto s_1 \longmapsto \dots$ with $s_0 = s$, $\mathfrak{M}, s_i \models \varphi_2$, for some $i \geqslant 0$, and $\mathfrak{M}, s_j \models \varphi_1$, for every $0 \leqslant j < i$;
- $\mathfrak{M}, s \models \exists(\varphi_1\mathcal{U}\varphi_2) \leftrightharpoons$ there exists a path $s_0 \longmapsto s_1 \longmapsto \dots$ with $s_0 = s$, such that $\mathfrak{M}, s_i \models \varphi_2$, for some $i \geqslant 0$, and $\mathfrak{M}, s_j \models \varphi_1$, for every $0 \leqslant j < i$.

Satisfiable and valid formulas are defined as for **CTL***. Formally, by **CTL** we mean the set of valid **CTL**-formulas; this set is closed under substitution.

For each of the logics described above, by a variable-free fragment we mean the subset of the logic containing only formulas without any propositional variables. Given formulas φ, ψ and a propositional variable p, we denote by $\varphi[p/\psi]$ the result of uniformly substituting ψ for p in φ.

3.3 Finite-variable fragments of CTL* and CTL

In this section, we consider the complexity of satisfiability for finite-variable fragments of **CTL** and **CTL***, as well as semantic expressivity of those fragments.

We start by noticing that for both **CTL** and **CTL*** satisfiability of the variable-free fragment is polynomial-time decidable. Indeed, it is easy to check that, for these logics, every variable-free formula is equivalent to either $\bot$ or $\top$. Thus, to check for satisfiability of a variable-free formula φ, all we need to do is to recursively replace each subformula of φ by either $\bot$ or $\top$, which gives us an algorithm that runs in time linear in the size of φ. Since both **CTL** and **CTL*** are at least EXPTIME-hard and P $\neq$ EXPTIME, variable-free fragments of these logics cannot be as expressive as the entire logic.

We next prove that the situation changes once we allow just one variable to be used in the construction of formulas. Then, we can express everything we can express in the full languages of **CTL** and **CTL***; as a consequence, the complexity of satisfiability becomes as hard as satisfiability for the full languages. In what follows, we first present the proof for **CTL***, and then point out how that work carries over to **CTL**.

Let φ be an arbitrary **CTL***-formula. Without a loss of generality we may assume that φ contains propositional variables $p_1, \dots p_n$. Let p_{n+1} be a variable not occurring in φ. First, inductively define the translation $\cdot'$ as follows:

$$
\begin{array}{lcl}
{p_i}' & = & p_i, \quad \text{where } i \in \{1, \dots, n\}; \\
\bot' & = & \bot; \\
(\phi \to \psi)' & = & \phi' \to \psi'; \\
(\forall \alpha)' & = & \forall(\Box p_{n+1} \to \alpha'); \\
(\bigcirc \alpha)' & = & \bigcirc \alpha'; \\
(\alpha \,\mathcal{U} \beta)' & = & \alpha' \,\mathcal{U} \beta'.
\end{array}
$$

Next, let

$$\Theta = p_{n+1} \wedge \forall \Box (\exists \bigcirc p_{n+1} \leftrightarrow p_{n+1}),$$

and define

$$\widehat{\varphi} = \Theta \wedge \varphi'.$$

Intuitively, the translation $\cdot'$ restricts evaluation of formulas to the paths where every state makes the variable p_{n+1} true, while Θ acts as a guard making sure that all paths in a model satisfy this property. Notice that φ is equivalent to $\widehat{\varphi}[p_{n+1}/\top]$.

Lemma 3.3.1 *Formula φ is satisfiable if, and only if, formula $\widehat{\varphi}$ is satisfiable.*

Proof. Suppose that $\widehat{\varphi}$ is not satisfiable. Then, $\neg\widehat{\varphi} \in$ **CTL*** and, since **CTL*** is closed under substitution, $\neg\widehat{\varphi}[p_{n+1}/\top] \in$ **CTL***. As $\widehat{\varphi}[p_{n+1}/\top] \leftrightarrow \varphi \in$ **CTL***, so $\neg\varphi \in$ **CTL***; thus, φ is not satisfiable.

Suppose that $\widehat{\varphi}$ is satisfiable. In particular, let $\mathfrak{M}, s_0 \models \widehat{\varphi}$ for some model $\mathfrak{M}$ and some s_0 in $\mathfrak{M}$. Define $\mathfrak{M}'$ to be the smallest submodel of $\mathfrak{M}$ such that

- s_0 is in $\mathfrak{M}'$;

- if x is in $\mathfrak{M}'$, $x \longmapsto y$, and $\mathfrak{M}, y \models p_{n+1}$, then y is also in $\mathfrak{M}'$.

Notice that, since $\mathfrak{M}, s_0 \models p_{n+1} \wedge \forall\Box(\exists\bigcirc p_{n+1} \leftrightarrow p_{n+1})$, the model $\mathfrak{M}'$ is serial, as required, and that p_{n+1} is true at every state of $\mathfrak{M}'$.

We now show that $\mathfrak{M}', s_0 \models \varphi$. Since $\mathfrak{M}, s_0 \models \varphi'$, it suffices to prove that, for every state x in $\mathfrak{M}'$ and every state subformula ψ of φ, we have $\mathfrak{M}, x \models \psi'$ if, and only if, $\mathfrak{M}', x \models \psi$; and that, for every path π in $\mathfrak{M}'$ and every path subformula α of φ, we have $\mathfrak{M}, \pi \models \alpha'$ if, and only if, $\mathfrak{M}', \pi \models \alpha$. This can be done by simultaneous induction on ψ and α.

The base case as well as Boolean cases are straightforward.

Let $\psi = \forall\alpha$, so $\psi' = \forall(\Box p_{n+1} \to \alpha')$. Assume that $\mathfrak{M}, x \not\models \forall(\Box p_{n+1} \to \alpha')$. Then, $\mathfrak{M}, \pi \not\models \alpha'$, for some $\pi \in \Pi(x)$ such that $\mathfrak{M}, \pi[i] \models p_{n+1}$, for every $i \geqslant 0$. By construction of $\mathfrak{M}'$, π is a path is $\mathfrak{M}'$; thus, we can apply the inductive hypothesis to conclude that $\mathfrak{M}', \pi \not\models \alpha$. Therefore, $\mathfrak{M}', x \not\models \forall\alpha$, as required. Conversely, assume that $\mathfrak{M}', x \not\models \forall\alpha$. Then, $\mathfrak{M}', \pi \not\models \alpha$, for some $\pi \in \Pi(x)$. Clearly, π is a path in $\mathfrak{M}$. Since p_{n+1} is true at every state in $\mathfrak{M}'$, and thus, at every state in π, using the inductive hypothesis, we conclude that $\mathfrak{M}, x \not\models \forall(\Box p_{n+1} \to \alpha')$.

The cases for the temporal connectives are straightforward. ❑

Lemma 3.3.2 *If $\widehat{\varphi}$ is satisfiable, then it is satisfied in a model where p_{n+1} is true at every state.*

Proof. If $\widehat{\varphi}$ is satisfiable, then, as has been shown in the proof of Lemma 3.3.1, φ is satisfied in a model where p_{n+1} is true at every state; i.e., $\mathfrak{M}, s \models \varphi$ for some $\mathfrak{M} = (\mathcal{S}, \longmapsto, V)$ such that p_{n+1} is true at every state in $\mathcal{S}$ and some $s \in S$. Since φ is equivalent to $\widehat{\varphi}[p_{n+1}/\top]$, clearly $\mathfrak{M}, s \models \widehat{\varphi}$. ❑

Next, we model all the variables of $\widehat{\varphi}$ by single-variable formulas $A_1, \ldots, A_{n+1}$. This is done in the following way. Consider the class M of models that, for each $m \in \{1, \ldots, n+1\}$, contains a model $\mathfrak{M}_m = (\mathcal{S}_m, \longmapsto, V_m)$ defined as follows:

- $\mathcal{S}_m = \{r_m, b^m, a_1^m, a_2^m, \ldots, a_{2m}^m\}$;

- $\longmapsto = \{\langle r_m, b^m\rangle, \langle r_m, a_1^m\rangle\} \cup \{\langle a_i^m, a_{i+1}^m\rangle : 1 \leqslant m \leqslant 2m-1\} \cup \{\langle s, s\rangle : s \in \mathcal{S}_m\}$;

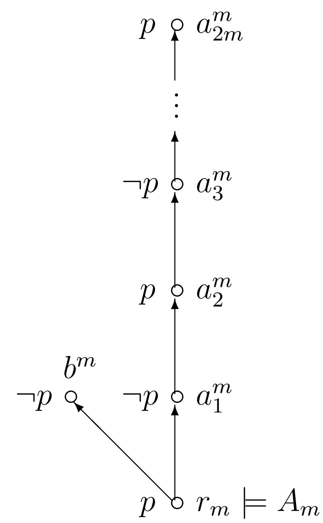

Figure 3.1: Model $\mathfrak{M}_m$

- $s \in V_m(p)$ if, and only if, $s = r_m$ or $s = a^m_{2k}$, for some $k \in \{1, \dots, m\}$.

The model $\mathfrak{M}_m$ is depicted in Figure 3.1, where circles represent states with loops. With every such $\mathfrak{M}_m$, we associate a formula A_m, in the following way. First, inductively define the sequence of formulas

$$\begin{array}{lcl} \chi_0 & = & \forall \Box p; \\ \chi_{k+1} & = & p \wedge \exists \bigcirc (\neg p \wedge \exists \bigcirc \chi_k). \end{array}$$

Next, for every $m \in \{1, \dots, n+1\}$, let

$$A_m = \chi_m \wedge \exists \bigcirc \forall \Box \neg p.$$

Lemma 3.3.3 *Let $\mathfrak{M}_k \in \mathsf{M}$ and let x be a state in $\mathfrak{M}_k$. Then, $\mathfrak{M}_k, x \models A_m$ if, and only if, $k = m$ and $x = r_m$.*

Proof. Straightforward. ❑

Now, for every $m \in \{1, \dots, n+1\}$, define

$$B_m = \exists \bigcirc A_m.$$

Finally, let σ be a (substitution) function that, for every $i \in \{1, \ldots, n+1\}$, replaces p_i by B_i, and let

$$\varphi^* = \sigma(\widehat{\varphi}).$$

Notice that the formula φ^* contains only a single variable, p.

Lemma 3.3.4 *Formula φ is satisfiable if, and only if, formula φ^* is satisfiable.*

Proof. Suppose that φ is not satisfiable. Then, in view of Lemma 3.3.1, $\widehat{\varphi}$ is not satisfiable. Then, $\neg\widehat{\varphi} \in \mathbf{CTL}^*$ and, since $\mathbf{CTL}^*$ is closed under substitution, $\neg\varphi^* \in \mathbf{CTL}^*$. Thus, φ^* is not satisfiable.

Suppose that φ is satisfiable. Then, in view of Lemmas 3.3.1 and 3.3.2, $\widehat{\varphi}$ is satisfiable in a model $\mathfrak{M} = (\mathcal{S}, \longmapsto, V)$ where p_{n+1} is true at every state. We can assume without a loss of generality that every $x \in \mathcal{S}$ is connected by some path to s. Define model $\mathfrak{M}'$ as follows. Append to $\mathfrak{M}$ all the models from M (i.e., take their disjoint union), and for every $x \in \mathcal{S}$, make r_m, the root of $\mathfrak{M}_m$, accessible from x in $\mathfrak{M}'$ exactly when $\mathfrak{M}, x \models p_m$. The evaluation of p is defined as follows: for states from each $\mathfrak{M}_m \in \mathsf{M}$, the evaluation is the same as in $\mathfrak{M}_m$, and for every $x \in \mathcal{S}$, let $x \notin V'(p)$.

We now show that $\mathfrak{M}', s \models \varphi^*$. It is easy to check that $\mathfrak{M}', s \models \sigma(\Theta)$. It thus remains to show that $\mathfrak{M}', s \models \sigma(\varphi')$. Since $\mathfrak{M}, s \models \varphi'$, it suffices to prove that $\mathfrak{M}, x \models \psi'$ if, and only if, $\mathfrak{M}', x \models \sigma(\psi')$, for every state x in $\mathfrak{M}$ and every state subformula ψ of φ; and that $\mathfrak{M}, \pi \models \alpha'$ if, and only if, $\mathfrak{M}', \pi \models \sigma(\alpha')$, for every path π in $\mathfrak{M}$ and every path subformula α of φ. This can be done by simultaneous induction on ψ and α.

Let $\psi = p_i$, so $\psi' = p_i$ and $\sigma(\psi') = B_i$. Assume that $\mathfrak{M}, x \models p_i$. Then, by construction of $\mathfrak{M}'$, we have $\mathfrak{M}', x \models B_i$. Conversely, assume that $\mathfrak{M}', x \models B_i$. As $\mathfrak{M}', x \models B_i$ implies $\mathfrak{M}', x \models \exists\bigcirc p$ and since $\mathfrak{M}, y \not\models p$, for every $y \in \mathcal{S}$, this can only happen if $x \longmapsto^{\mathfrak{M}'} r_m$, for some $m \in \{1, \ldots, n+1\}$. Since, then, $r_m \models A_i$, in view of Lemma 3.3.3, $m = i$, and thus, by construction of $\mathfrak{M}'$, we have $\mathfrak{M}, x \models p_i$.

The Boolean cases are straightforward.

Let $\psi = \forall\alpha$, so $\psi' = \forall(\Box p_{n+1} \to \alpha')$ and $\sigma(\psi') = \forall(\Box B_{n+1} \to \sigma(\alpha'))$. Assume that $\mathfrak{M}, x \not\models \forall(\Box p_{n+1} \to \alpha')$. Then, for some $\pi \in \Pi(x)$ such that $\mathfrak{M}, \pi[i] \models p_{n+1}$ for every $i \geqslant 0$, we have $\mathfrak{M}, \pi \not\models \alpha'$. Clearly, π is a path in $\mathfrak{M}'$, and thus, by inductive hypothesis, $\mathfrak{M}', \pi[i] \models B_{n+1}$, for every $i \geqslant 0$, and $\mathfrak{M}', \pi \not\models \sigma(\alpha')$. Hence, $\mathfrak{M}', x \not\models \forall(\Box B_{n+1} \to \sigma(\alpha'))$, as required. Conversely, assume that $\mathfrak{M}', x \not\models \forall(\Box B_{n+1} \to \sigma(\alpha'))$. Then, for some $\pi \in \Pi(x)$ such that $\mathfrak{M}', \pi[i] \models B_{n+1}$ for every $i \geqslant 0$, we have $\mathfrak{M}', \pi \not\models \sigma(\alpha')$. Since by construction of $\mathfrak{M}'$, no state outside of $\mathcal{S}$ satisfies B_{n+1}, we know that π is a path in $\mathfrak{M}$. Thus, we can use the inductive hypothesis to conclude that $\mathfrak{M}, x \not\models \forall(\Box p_{n+1} \rightarrow \alpha')$.

The cases for the temporal connectives are straightforward. ❑

Lemma 3.3.4, together with the observation that the formula φ^* is polynomial-time computable from φ, give us the following:

Theorem 3.3.5 *There exists a polynomial-time computable function e assigning to every* **CTL****-formula φ a single-variable formula $e(\varphi)$ such that $e(\varphi)$ is satisfiable if, and only if, φ is satisfiable.*

Theorem 3.3.6 *The satisfiability problem for the single-variable fragment of* **CTL*** *is* 2EXPTIME*-complete.*

Proof. The lower bound immediately follows from Theorem 3.3.5 and 2EXPTIME-hardness of satisfiability for **CTL*** [80]. The upper bound follows from the 2EXPTIME upper bound for satisfiability for **CTL*** [80]. ❑

We now show how the argument presented above for **CTL*** can be adapted to **CTL**. First, we notice that if our sole purpose were to prove that satisfiability for the single-variable fragment of **CTL** is EXPTIME-complete, we would not need to work with the entire set of connectives present in the language of **CTL**,—it would suffice to work with a relatively simple fragment of **CTL** containing the modal operators $\forall\bigcirc$ and $\forall\Box$, whose satisfiability, as follows from [24], is EXPTIME-hard. We do, however, also want to establish that the single-variable fragment of **CTL** is as expressive the entire logic; therefore, we embed the entire **CTL** into its single-variable fragment. To that end, we can carry out an argument similar to the one presented above for **CTL***.

First, we define the translation $\cdot'$ as follows:

$$
\begin{array}{lcl}
p_i' & = & p_i \quad \text{where } i \in \{1, \ldots, n\}; \\
(\bot)' & = & \bot; \\
(\phi \rightarrow \psi)' & = & \phi' \rightarrow \psi'; \\
(\forall\bigcirc\phi)' & = & \forall\bigcirc(p_{n+1} \rightarrow \phi'); \\
(\forall(\phi\,\mathcal{U}\psi))' & = & \forall(\phi'\,\mathcal{U}(p_{n+1} \wedge \psi')); \\
(\exists(\phi\,\mathcal{U}\psi))' & = & \exists(\phi'\,\mathcal{U}(p_{n+1} \wedge \psi')).
\end{array}
$$

Next, let

$$\Theta = p_{n+1} \wedge \forall\Box(\exists\bigcirc p_{n+1} \leftrightarrow p_{n+1}).$$

and define

$$\widehat{\varphi} = \Theta \wedge \varphi'.$$

Intuitively, the translation $\cdot'$ restricts the evaluation of formulas to the states where p_{n+1} is true. Formula Θ acts as a guard making sure that all states in a model satisfy this property. We can then prove the analogues of Lemmas 3.3.1 and 3.3.2.

Lemma 3.3.7 *Formula φ is satisfiable if, and only if, formula $\widehat{\varphi}$ is satisfiable.*

Proof. Analogous to the proof of Lemma 3.3.1. In the right-to-left direction, inductive steps for modal connectives rely on the fact that in a submodel we constructed every state makes the variable p_{n+1} true. ❑

Lemma 3.3.8 *If $\widehat{\varphi}$ is satisfiable, then it is satisfied in a model where p_{n+1} is true at every state.*

Proof. Analogous to the proof of Lemma 3.3.2. ❑

Next, we model propositional variables $p_1, \ldots, p_{n+1}$ in the formula $\widehat{\varphi}$ exactly as in the argument for **CTL***, i.e., we use formulas A_m and their associated models $\mathfrak{M}_m$, where $m \in \{1, \ldots, n+1\}$. This can be done since formulas A_m are, in fact, **CTL**-formulas. Lemma 3.3.3 can, thus, be reused for **CTL**, as well.

We then define a single-variable **CTL**-formula φ^* analogously to the way it had been done for **CTL***:

$$\varphi^* = \sigma(\widehat{\varphi}),$$

where σ is a (substitution) function that, for every $i \in \{1, \ldots, n+1\}$, replaces p_i by $B_i = \exists\bigcirc A_i$. We can then prove the analogue of Lemma 3.3.4.

Lemma 3.3.9 *Formula φ is satisfiable if, and only if, formula φ^* is satisfiable.*

Proof. Analogous to the proof of Lemma 3.3.4. In the left-to-right direction, the inductive steps for the modal connectives rely on the fact that the formula B_{n+1} is true precisely at the states of the model that satisfies φ. ❑

We, thus, obtain the following:

Theorem 3.3.10 *There exists a polynomial-time computable function e assigning to every* **CTL***-formula φ a single-variable formula $e(\varphi)$ such that $e(\varphi)$ is satisfiable if, and only if, φ is satisfiable.*

Theorem 3.3.11 *The satisfiability problem for the single-variable fragment of* **CTL** *is* EXPTIME*-complete.*

Proof. The lower bound immediately follows from Theorem 3.3.10 and EXPTIME-hardness of satisfiability for **CTL** [24]. The upper bound follows from the EXPTIME upper bound for satisfiability for **CTL** [20]. ❑

3.4 Alternating-time temporal logics

Alternating-time temporal logics **ATL*** and **ATL** can be conceived of as generalisations of **CTL*** and **CTL**, respectively. Their models incorporate transitions occasioned by simultaneous actions of the agents in the system rather than abstract transitions, as in **CTL*** and **CTL**, and we now reason about paths that can be forced by cooperative actions of coalitions of agents, rather than just about all ($\forall$) and some ($\exists$) paths. We do not lose the ability to reason about all and some paths in **ATL*** and **ATL**, however, so these logics are generalisations of **CTL*** and **CTL**, respectively.

The language of **ATL*** contains a non-empty, finite set $\mathbb{AG}$ of names of agents (subsets of $\mathbb{AG}$ are called coalitions); a countable set $\mathit{Var} = \{p_1, p_2, \ldots\}$ of propositional variables; the propositional constant $\bot$; the Boolean connective $\rightarrow$; coalition quantifiers $\langle\!\langle C \rangle\!\rangle$, for every $C \subseteq \mathbb{AG}$; and temporal connectives $\bigcirc$ ("next"), $\mathcal{U}$ ("until"), and $\Box$ ("always in the future"). The language contains two kinds of formulas: state formulas and path formulas. State formulas φ and path formulas α are simultaneously defined by the following BNF expressions:

$$\varphi ::= p \mid \bot \mid (\varphi \rightarrow \varphi) \mid \langle\!\langle C \rangle\!\rangle \vartheta,$$

$$\vartheta ::= \varphi \mid (\vartheta \to \vartheta) \mid (\vartheta \mathcal{U} \vartheta) \mid \bigcirc \vartheta \mid \Box \vartheta,$$

where C ranges over subsets of $\mathbb{AG}$ and p ranges over Var. Other Boolean and temporal connectives are defined as for $\mathbf{CTL}^*$.

Formulas are evaluated in concurrent game models. A concurrent game model is a tuple $\mathfrak{M} = (\mathbb{AG}, \mathcal{S}, Act, act, \delta, V)$, where

- $\mathbb{AG} = \{1, \ldots, k\}$ is a finite, non-empty set of agents;
- $\mathcal{S}$ is a non-empty set of states;
- Act is a non-empty set of actions;
- $act : \mathbb{AG} \times \mathcal{S} \mapsto 2^{Act}$ is an action manager function assigning a non-empty set of "available" actions to an agent at a state;
- δ is a transition function assigning to every state $s \in \mathcal{S}$ and every action profile $\alpha = (\alpha_1, \ldots, \alpha_k)$, where $\alpha_a \in act(a, s)$, for every $a \in \mathbb{AG}$, an outcome state $\delta(s, \alpha)$;
- V is a (valuation) function $V : \mathit{Var} \to 2^{\mathcal{S}}$.

A few auxiliary notions need to be introduced for the definition of the satisfaction relation.

A *path* is an infinite sequence $s_0, s_1, \ldots$ of states in $\mathfrak{M}$ such that, for every $i \geqslant 0$, the following holds: $s_{i+1} \in \delta(s_i, \alpha)$, for some action profile α. The set of all such sequences is denoted by $\mathcal{S}^\omega$. The notation $\pi[i]$ and $\pi[i, \infty]$ is used as for $\mathbf{CTL}^*$. Initial segments $\pi[0, i]$ of paths are called *histories*; a typical history is denoted by h, and its last state, $\pi[i]$, is denoted by $last(h)$. Note that histories are non-empty sequences of states in $\mathcal{S}$; we denote the set of all such sequences by $\mathcal{S}^+$.

Given $s \in \mathcal{S}$ and $C \subseteq \mathbb{AG}$, a C-action at s is a tuple α_C such that $\alpha_C(a) \in act(a, s)$, for every $a \in C$, and $\alpha_C(a')$, for every $a' \notin C$, is an unspecified action of agent a' at s (technically, a C-action might be thought of as an equivalence class on action profiles determined by a vector of chosen actions for every $a \in C$); we denote by $act(C, s)$ the set of C-actions at s. An action profile α extends a C-action α_C, symbolically $\alpha_C \sqsubseteq \alpha$, if $\alpha(a) = \alpha_C(a)$, for every $a \in C$. The *outcome set* of the C-action α_C at s is the set of states $out(s, \alpha_C) = \{\delta(s, \alpha) \mid \alpha \in act(\mathbb{AG}, s) \text{ and } \alpha_C \sqsubseteq \alpha\}$.

A *strategy* for an agent a is a function $str_a(h) : \mathcal{S}^+ \mapsto act(a, last(h))$ assigning to every history an action available to a at the last state of the history.

A C-strategy is a tuple of strategies for every $a \in C$. The function $out(s, \alpha_C)$ can be naturally extended to the functions $out(s, str_C)$ and $out(h, str_C)$ assigning to a given state s, or more generally a given history h, and a given C-strategy the set of states that can result from applying str_C at s or h, respectively. The set of all paths that can result when the agents in C follow the strategy str_C from a given state s is denoted by $\Pi(s, str_C)$ and defined as $\{\pi \in \mathcal{S}^\omega \mid \pi[0] = s \text{ and} \pi[j+1] \in out(\pi[0,j], str_C), \text{ for every } j \geqslant 0\}$.

The satisfaction relation between models $\mathfrak{M}$, states s, and state formulas φ, as well as between models $\mathfrak{M}$, paths π, and path formulas ϑ, is defined as follows:

- $\mathfrak{M}, s \models p_i \leftrightharpoons s \in V(p_i)$;
- $\mathfrak{M}, s \models \bot$ never holds;
- $\mathfrak{M}, s \models \varphi_1 \to \varphi_2 \leftrightharpoons \mathfrak{M}, s \models \varphi_1$ implies $\mathfrak{M}, s \models \varphi_2$;
- $\mathfrak{M}, s \models \langle\!\langle C \rangle\!\rangle \vartheta_1 \leftrightharpoons$ there exists a C-strategy str_C such that $\mathfrak{M}, \pi \models \vartheta_1$ holds for every $\pi \in \Pi(s, str_C)$;
- $\mathfrak{M}, \pi \models \varphi_1 \leftrightharpoons \mathfrak{M}, \pi[0] \models \varphi_1$;
- $\mathfrak{M}, \pi \models \vartheta_1 \to \vartheta_2 \leftrightharpoons \mathfrak{M}, \pi \models \vartheta_1$ implies $\mathfrak{M}, \pi \models \vartheta_2$;
- $\mathfrak{M}, \pi \models \bigcirc \vartheta_1 \leftrightharpoons \mathfrak{M}, \pi[1, \infty] \models \vartheta_1$;
- $\mathfrak{M}, \pi \models \Box \vartheta_1 \leftrightharpoons \mathfrak{M}, \pi[i, \infty] \models \vartheta_1$, for every $i \geqslant 0$;
- $\mathfrak{M}, \pi \models \vartheta_1 \mathcal{U} \vartheta_2 \leftrightharpoons \mathfrak{M}, \pi[i, \infty] \models \vartheta_2$ for some $i \geqslant 0$ and $\mathfrak{M}, \pi[j, \infty] \models \vartheta_1$ for every j such that $0 \leqslant j < i$.

An **ATL***-formula is a state formula in this language. An **ATL***-formula is satisfiable if it is satisfied by some state of some model, and valid if it is satisfied by every state of every model. Formally, by **ATL*** we mean the set of all valid **ATL***-formulas; notice that this set is closed under uniform substitution.

Logic **ATL** can be thought of as a fragment of **ATL*** containing only formulas where a coalition quantifier is always paired up with a temporal connective. This, as in the case of **CTL**, eliminates path-formulas. Such composite "modal" operators are $\langle\langle C\rangle\rangle\bigcirc$, $\langle\langle C\rangle\rangle\Box$, and $\langle\langle C\rangle\rangle\mathcal{U}$. Formulas are defined by the following BNF expression:

$$\varphi ::= p \mid \bot \mid (\varphi \to \varphi) \mid \langle\langle C\rangle\rangle\bigcirc\varphi \mid \langle\langle C\rangle\rangle\Box\varphi \mid \langle\langle C\rangle\rangle(\varphi\mathcal{U}\varphi),$$

where C ranges over subsets of $\mathbb{AG}$ and p ranges over Var. The other Boolean connectives and the constant $\top$ are defined as for **CTL**.

The satisfaction relation between concurrent game models $\mathfrak{M}$, states s, and formulas φ is inductively defined as follows (we only list the cases for the "new" modal operators):

- $\mathfrak{M}, s \models \langle\langle C\rangle\rangle\bigcirc\varphi_1 \leftrightharpoons$ there exists a C-action α_C such that $\mathfrak{M}, s' \models \varphi_1$ whenever $s' \in out(s, act_C)$;

- $\mathfrak{M}, s \models \langle\langle C\rangle\rangle\Box\varphi_1 \leftrightharpoons$ there exists a C-strategy str_C such that $\mathfrak{M}, \pi[i] \models \varphi_1$ holds for all $\pi \in out(s, str_C)$ and all $i \geqslant 0$;

- $\mathfrak{M}, s \models \langle\langle C\rangle\rangle(\varphi_1\mathcal{U}\varphi_2) \leftrightharpoons$ there exists a C-strategy str_C such that, for all $\pi \in out(s, str_C)$, there exists $i \geqslant 0$ with $\mathfrak{M}, \pi[i] \models \varphi$ and $\mathfrak{M}, \pi[j] \models \varphi$ holds for every j such that $0 \leqslant j < i$.

Satisfiable and valid formulas are defined as for **ATL***. Formally, by **ATL** we mean the set of all valid **ATL***-formulas; this set is closed under substitution.

Remark 3.4.1 *We have given definitions of satisfiability and validity for* **ATL*** *and* **ATL** *that assume that the set of all agents* $\mathbb{AG}$ *present in the language is "fixed in advance". At least two other notions of satisfiability (and, thus, validity) for these logics have been discussed in the literature (see, e.g., [84])—i.e., satisfiability of a formula in a model where the set of all agents coincides with the set of agents named in the formula and satisfiability of a formula in a model where the set of agents is any set including the agents named in the formula (in this case, it suffices to consider all the agents named in the formula plus one extra agent). In what follows, we explicitly consider only the notion of satisfiability for a fixed set of agents; other notions of satisfiability can be handled in a similar way.*

3.5 Finite-variable fragments of ATL* and ATL

We start by noticing that satisfiability for variable-free fragments of both **ATL*** and **ATL** is polynomial-time decidable, using the algorithm similar to the one outlined for $\mathbf{CTL}^*$ and **CTL**. It follows that variable-free fragments of $\mathbf{ATL}^*$ and **ATL** cannot be as expressive as entire logics.

We also notice that, as is well-known, satisfiability for $\mathbf{CTL}^*$ is polynomial-time reducible to satisfiability for $\mathbf{ATL}^*$ and satisfiability for **CTL** is polynomial-time reducible to satisfiability for **ATL**, using the translation that replaces all occurrences of $\forall$ by $\langle\!\langle \varnothing \rangle\!\rangle$ and all occurrences of $\exists$ by $\langle\!\langle \mathbb{AG} \rangle\!\rangle$. Thus, Theorems 3.3.6 and 3.3.11, together with the known upper bounds [37, 77, 71], immediately give us the following:

Theorem 3.5.1 *The satisfiability problem for the single-variable fragment of* $\mathbf{ATL}^*$ *is* 2EXPTIME*-complete.*

Theorem 3.5.2 *The satisfiability problem for the single-variable fragment of* **ATL** *is* EXPTIME*-complete.*

In the rest of this section, we show that single-variable fragments of $\mathbf{ATL}^*$ and **ATL** are as expressive as the entire logics by embedding both $\mathbf{ATL}^*$ and **ATL** into their single-variable fragments. The arguments closely resemble the ones for $\mathbf{CTL}^*$ and **CTL**, so we only provide enough detail for the reader to be able to easily fill in the rest.

First, consider $\mathbf{ATL}^*$. The translation $\cdot'$ is defined as for $\mathbf{CTL}^*$, except that the clause for $\forall$ is replaced by the following:

$$(\langle\!\langle C \rangle\!\rangle \alpha)' \quad = \quad \langle\!\langle C \rangle\!\rangle (\Box p_{n+1} \wedge \alpha').$$

Next, we define

$$\Theta = p_{n+1} \wedge \langle\!\langle \varnothing \rangle\!\rangle \Box (\langle\!\langle \mathbb{AG} \rangle\!\rangle \bigcirc p_{n+1} \leftrightarrow p_{n+1})$$

and

$$\widehat{\varphi} = \Theta \wedge \varphi'.$$

Then, we can prove the analogues of Lemmas 3.3.1 and 3.3.2.

We next model all the variables of $\widehat{\varphi}$ by single-variable formulas $A'_1, \dots, A'_{n+1}$. To that end, we use the class of concurrent game models $\mathsf{M} = \{\mathfrak{M}'_1, \dots, \mathfrak{M}'_{\mathsf{n+1}}\}$ that closely resemble models $\mathfrak{M}_1, \dots, \mathfrak{M}_{n+1}$ used in the argument for $\mathbf{CTL}^*$. For every $\mathfrak{M}'_i$, with $i \in \{1, \dots, n+1\}$, the set of states and the valuation V are the same as for $\mathfrak{M}_i$; in addition, whenever $s \longmapsto s'$ holds in $\mathfrak{M}_i$, we set $\delta(s, \alpha) = s'$, for every action profile α. The actions available to an agent a at each state of $\mathfrak{M}_i$ are all the actions available to a at any of the states of the model $\mathfrak{M}$ to which we are going to attach models $\mathfrak{M}'_i$ when proving the analogue of Lemma 3.3.4, as well as an extra action d_a that we need to set up transitions from the states of $\mathfrak{M}$ to the roots of $\mathfrak{M}'_i$s.

With every $\mathfrak{M}'_i$ we associate the formula A'_i. First, inductively define the sequence of formulas

$$\begin{array}{lcl} \chi'_0 & = & \langle\!\langle \varnothing \rangle\!\rangle \Box\, p; \\ \chi'_{k+1} & = & p \wedge \langle\!\langle \mathbb{AG} \rangle\!\rangle \bigcirc (\neg p \wedge \langle\!\langle \mathbb{AG} \rangle\!\rangle \bigcirc \chi_k). \end{array}$$

Next, for every $m \in \{1, \ldots, n+1\}$, let

$$A'_m = \chi'_m \wedge \langle\!\langle \mathbb{AG} \rangle\!\rangle \bigcirc \langle\!\langle \varnothing \rangle\!\rangle \Box\, \neg p.$$

Lemma 3.5.3 *Let $\mathfrak{M}'_k \in \mathsf{M}$ and let x be a state in $\mathfrak{M}'_k$. Then, $\mathfrak{M}'_k, x \models A'_m$ if, and only if, $k = m$ and $x = r_m$.*

Proof. Straightforward. ❑

Now, for every $m \in \{1, \ldots, n+1\}$, define

$$B'_m = \langle\!\langle \mathbb{AG} \rangle\!\rangle \bigcirc A'_m.$$

Finally, let σ be a (substitution) function that, for every $i \in \{1, \ldots, n+1\}$, replaces p_i by B'_i, and let

$$\varphi^* = \sigma(\widehat{\varphi}).$$

This allows us to prove the analogue of Lemma 3.3.4.

Lemma 3.5.4 *Formula φ is satisfiable if, and only if, formula φ^* is satisfiable.*

Proof. Analogous to the proof of Lemma 3.3.4. When constructing the model $\mathfrak{M}'$, whenever we need to connect a state s in $\mathfrak{M}$ to the root r_i of $\mathfrak{M}'_i$, we make an extra action, d_a, available to every agent a, and define $\delta(s, \langle d_a \rangle_{a \in \mathbb{AG}}) = r_i$. ❑

Thus, we have the following:

Theorem 3.5.5 *There exists a polynomial-time computable function e assigning to every* **ATL****-formula φ a single-variable formula $e(\varphi)$ such that $e(\varphi)$ is satisfiable if, and only if, φ is satisfiable.*

We then can adapt the argument for **ATL** form the one just presented in the same way we adapted the argument for **CTL** from the one for **CTL***, obtaining the following:

Theorem 3.5.6 *There exists a polynomial-time computable function e assigning to every* **ATL**-*formula* φ *a single-variable formula* $e(\varphi)$ *such that* $e(\varphi)$ *is satisfiable if, and only if,* φ *is satisfiable.*

3.6 Discussion

We have shown that logics **CTL***, **CTL**, **ATL***, and **ATL** can be polynomial-time embedded into their single-variable fragments; i.e., their single-variable fragments are as expressive as the entire logics. Consequently, for these logics, satisfiability is as computationally hard when one considers only formulas of one variable as when one considers arbitrary formulas. Thus, the complexity of satisfiability for these logics cannot be reduced by restricting the number of variables allowed in the construction of formulas.

The technique presented in this chapter can be applied to many other modal and temporal logics of computation considered in the literature. We will not here attempt a comprehensive list, but rather mention a few examples.

The proofs presented in this chapter can be extended in a rather straightforward way to Branching- and Alternating-time temporal-epistemic logics [42, 77, 83, 34], i.e., logics that enrich the logics considered in this chapter with the epistemic operators of individual, distributed, and common knowledge for the agents. Our approach can be used to show that single-variable fragments of those logics are as expressive as the entire logics and that, consequently, the complexity of satisfiability for them is as hard (EXPTIME-hard or 2EXPTIME-hard) as for the entire logics. Clearly, the same approach can be applied to epistemic logics [22, 32, 36], i.e., logics containing epistemic, but not temporal, operators—such logics are widely used for reasoning about distributed computation. Our argument also applies to logics with the so-called universal modality [31] to obtain EXPTIME-completeness of their variable-free fragments. The technique presented here has also been recently used [68] to show that propositional dynamic logics are as expressive in the language without propositional variables as in the language with an infinite supply of propositional variables. Since our method is modular in the way it tackles modalities present in the language, it naturally lends itself to modal languages combining various modalities—a trend that has been gaining prominence for some time now.

The technique presented in this chapter can also be lifted to first-order languages to prove undecidability results about fragments of first-order modal and related logics,—see [67].

We conclude by noticing that, while we have been able to overcome the limitations of the technique from [40] described in the introduction, our modification thereof has limitations of its own. It is not applicable to logics whose semantics forbids branching, such as **LTL** or temporal-epistemic logics of linear time [42, 33]. Our technique cannot be used, either, to show that finite-variable fragments of logical systems that are not closed under uniform substitution—such as public announcement logic **PAL** [63, 78]—have the same expressive power as the entire system. This does not preclude it from being used in establishing complexity results for finite-variable fragments of such systems provided they contain fragments, as is the case with **PAL** [53], that are closed under substitution and have the same complexity as the entire system.

Chapter 4

Complexity of finite-variable fragments of propositional modal logics of symmetric frames

4.1 Introduction

While the propositional modal logic **S5**, which has Kripke-style semantics in terms of reflexive, transitive, and symmetric frames, is "computationally intractable"—namely, its satisfiability problem is NP-complete—its n-variable fragments, for every $n \in \mathbb{N}$, are polynomial-time decidable. In contrast, as originally shown in [40] and further elaborated in [10] (see also [44] and [82]), most "natural" modal logics whose Kripke frames are reflexive, transitive, or both, remain intractable even if the number of propositional variables in their languages is restricted to one (for all logics in the intervals [**K**, **GL**] and [**K**, **Grz**]) or zero (for all logics in the interval [**K**, **K4**]). It is, thus, interesting to see if symmetry plays any role in making **S5** stand apart from other propositional modal logics in this regard. More generally, does the addition of the axiom of symmetry to a logic make its finite-variable fragments easier to decide than the entire logic? The role of symmetry in this context has not been investigated in the literature.

In this chapter, we answer this question in the negative by showing that all logics in the interval [**K**, **KTB**], where **KTB** is the propositional modal logic of reflexive and symmetric frames, have PSPACE-hard single-variable fragments. As a by-product, we prove that logics **KTB** and **KB**, which is the propositional modal logic of symmetric frames, can be embedded into their single-variable fragments, which are, thus, as semantically expressive—from the point of view of validity and (local) satisfiability—as the entire logics.

The chapter is organized as follows. In section 4.2, we briefly recall the syntax and semantics of the logics we consider in the present chapter and establish that all the logics in the interval [**K**, **KTB**] are PSPACE-hard. Then, in section 4.3, we present our main results concerning single-variable fragments of logics in [**K**, **KTB**]. We conclude in section 4.4.

4.2 Preliminaries

The propositional modal language contains countably many propositional variables $p_1, p_2, \ldots$, the Boolean constant $\bot$ ("falsehood"), the Boolean connective $\to$, and the modal connective $\Box$. Other connectives, as well as formulas, are defined as usual. We also use the following abbreviations:

$$\begin{array}{rclcrcl} \Box^0\varphi & = & \varphi, & & \Box^{\leqslant 0}\varphi & = & \varphi, \\ \Box^{n+1}\varphi & = & \Box\Box^n\varphi, & & \Box^{\leqslant n+1}\varphi & = & \Box^{\leqslant n}\varphi \wedge \Box^{n+1}\varphi, \\ \Box^{+}\varphi & = & \varphi \wedge \Box\varphi, & & \Diamond^n\varphi & = & \neg\Box^n\neg\varphi. \end{array}$$

A (Kripke) frame is a pair $\mathfrak{F} = \langle W, R\rangle$, where W is a non-empty set (of worlds) and R is a binary (accessibility) relation on W. A (Kripke) model is a pair $\mathfrak{M} = \langle \mathfrak{F}, V\rangle$, where $\mathfrak{F}$ is a frame and V is a valuation function assigning to every propositional variable a subset of W; if $\mathfrak{M}$ has the form $\langle \mathfrak{F}, V\rangle$, we say that it is based on $\mathfrak{F}$. The satisfaction relation between models $\mathfrak{M}$, worlds w, and formulas φ is defined as follows:

- $\mathfrak{M}, w \models p_i \leftrightharpoons w \in V(p_i)$;
- $\mathfrak{M}, w \models \bot$ never holds;
- $\mathfrak{M}, w \models \varphi_1 \to \varphi_2 \leftrightharpoons \mathfrak{M}, w \models \varphi_1$ implies $\mathfrak{M}, w \models \varphi_2$;
- $\mathfrak{M}, w \models \Box\varphi_1 \leftrightharpoons \mathfrak{M}, w' \models \varphi_1$ whenever wRw'.

Let $\mathfrak{C}$ be a class of frames. A formula is valid on $\mathfrak{C}$ if it is satisfied at every world of every model based on a frame from $\mathfrak{C}$. A formula is satisfiable in $\mathfrak{C}$ if it is satisfied at some world of some model based on a frame from $\mathfrak{C}$.

A propositional modal logic is a set of formulas containing all classical tautologies as well as the formula $\Box(p \to q) \to (\Box p \to \Box q)$ and closed under uniform substitution, modus ponens, and necessitation. Of particular interest to us are the logics **K**, which is the set of formulas valid on all frames; **KB**, which is the set of formulas valid on all frames whose accessibility relation is symmetric; and **KTB**, which is the set of formulas valid on all frames whose accessibility relation is reflexive and symmetric. More generally, we will be concerned with the interval [**K, KTB**] of logics L such that $\mathbf{K} \subseteq L \subseteq \mathbf{KTB}$. This interval contains a number of logics that have been, for various reasons, of interest to logicians; examples include **T**, which is the set of formulas valid on reflexive frames; **KB**; **KDB**, which is the set of formulas valid on symmetric and serial frames; and Hughes's logic [45]. If a logic L is Kripke-complete, i.e., coincides with the set of formulas valid on some class $\mathfrak{C}$ of frames, we say that a formula φ is L-satisfiable if φ is satisfiable in $\mathfrak{C}$.

We say that a logic is PSPACE-hard (PSPACE-complete) if the problem of membership in it is PSPACE-hard (PSPACE-complete); analogously for fragments of logics. In what follows, we rely on the statement as well as the proof of the following:

Theorem 4.2.1 *Let L be a logic such that $\mathbf{K} \subseteq L \subseteq \mathbf{KTB}$. Then, L is* PSPACE-*hard.*

Proof. The proof is a slight modification of Ladner's proof for logics between **K** and **S4** (Theorem 3.1 in [51]; see also [7], Section 6.7), which proceeds by reduction from the set TQBF of true quantified Boolean formulas, known to be PSPACE-hard [74]. Note that as PSPACE is closed under complementation, the complement of TQBF is also PSPACE-hard. Since every quantified Boolean formula can be reduced to one in the prenex normal form, we may assume without a loss of generality that both TQBF and its complement only contain formulas in the prenex normal form.

First, we define a polynomial-time computable translation f from the set of quantified Boolean formulas in the prenex normal form to the set of modal formulas such that

- if $\theta \in$ TQBF, then $f(\theta)$ is **KTB**-satisfiable;
- if $\theta \notin$ TQBF, then $f(\theta)$ is not **K**-satisfiable.

Let $\theta = \mathsf{Q}_1 p_1 \ldots \mathsf{Q}_m p_m \, \varphi(p_1, \ldots, p_m)$, where $\mathsf{Q}_1, \ldots, \mathsf{Q}_m \in \{\exists, \forall\}$ and $\varphi(p_1, \ldots, p_m)$ is a propositional formula containing no variables other than

$p_1, \dots, p_m$. Let $q_0, q_1, \dots, q_m$ be propositional variables not in θ. Then, $f(\theta)$ is a conjunction of the following formulas:

- q_0;
- $\Box^{\leqslant m} \bigwedge_{i=0}^{m} (q_i \to \bigwedge_{j \neq i} \neg q_j)$;
- $\Box^{\leqslant m-1} \bigwedge_{\{i \,:\, \mathsf{Q}_i = \exists\}} (q_{i-1} \to \Diamond q_i)$;
- $\Box^{\leqslant m-1} \bigwedge_{\{i \,:\, \mathsf{Q}_i = \forall\}} (q_{i-1} \to \Diamond (q_i \wedge p_i) \wedge \Diamond (q_i \wedge \neg p_i))$;
- $\Box^{\leqslant m-1} \bigwedge_{i=1}^{m-1} (q_i \to \bigwedge_{j \leqslant i} (p_j \to \Box (q_{i+1} \to p_j)) \wedge \bigwedge_{j \leqslant i} (\neg p_j \to \Box (q_{i+1} \to \neg p_j)))$;
- $\Box^m (q_m \to \varphi)$.

Note that only the second-to-last formula is substantively different from the formulas used in [51]. Suppose that θ is true, and thus, there exists a quantifier tree T witnessing its truth. We use T to define a **KTB**-model satisfying $f(\theta)$. Let W be the set of nodes of T and R be the symmetric and reflexive closure of the "daughter-of" relation of T. Thus, $\langle W, R \rangle$ is a **KTB**-frame. It remains to define the valuation. Let q_i be true precisely at the nodes of level i (where the root is a node of level 0), let p_i be true at a node of level $j \geqslant i$ if, and only if, the substitution of truth values for a variable of θ connected to that node, or to the node of level i on the same branch of T, returns "true" for p_i, and let p_i be false at all nodes of levels $j < i$. It is then straightforward to check that $f(\theta)$ is satisfied at the root of T. That falsehood of θ implies that $f(\theta)$ is not **K**-satisfiable is argued exactly as in Ladner's proof [51].

Now, let L be a logic such that $\mathbf{K} \subseteq L \subseteq \mathbf{KTB}$. If $\theta \notin$ TQBF, then $\neg f(\theta) \in \mathbf{K}$ and, hence, $\neg f(\theta) \in L$. Conversely, if $\theta \in$ TQBF, then $\neg f(\theta) \notin \mathbf{KTB}$ and, hence, $\neg f(\theta) \notin L$. Thus, the translation $t(\theta) = \neg f(\theta)$ reduces the complement of TQBF, which is PSPACE-hard, to L. Therefore, L is PSPACE-hard. ❑

As there exist polynomial-space algorithms for deciding satisfiability, and thus validity, for **KB**, **KDB**, and **KTB** (see, e.g., [38]), these logics are PSPACE-complete.

4.3 Complexity of finite-variable fragments

We now show, using a suitable modification of Halpern's technique [40], that single-variable fragments of all logics in the interval [**K, KTB**] are PSPACE-hard. In the course of the proof we establish that logics **KB** and **KTB** can be effectively embedded into their single-variable fragments.

Let φ be an arbitrary modal formula. Assume that φ only contains propositional variables $p_1, \dots, p_n$. First, recursively define the translation $\cdot'$ as follows:

$$\begin{array}{lcl} {p_i}' & = & p_i, \quad \text{where } i \in \{1, \dots, n\}; \\ \bot' & = & \bot; \\ (\phi \to \psi)' & = & \phi' \to \psi'; \\ (\Box\phi)' & = & \Box(p_{n+1} \to \phi'). \end{array}$$

Second, put

$$\widehat{\varphi} = p_{n+1} \wedge \varphi'.$$

Notice that φ is equivalent to $\widehat{\varphi}(p_{n+1}/\top)$ in **K** and, hence, in **KTB**.

Lemma 4.3.1 *Let $L \in \{\mathbf{K}, \mathbf{KTB}\}$. If $\widehat{\varphi}$ is L-satisfiable, then it is satisfiable in a model based on a frame for L where p_{n+1} is true at every world.*

Proof. Suppose that $\mathfrak{M}, w_0 \models \widehat{\varphi}$ for some model $\mathfrak{M}$ and some world w_0. Consider the submodel $\mathfrak{M}'$ of $\mathfrak{M}$ that consists of worlds where p_{n+1} is true. As $\mathfrak{M}, w_0 \models p_{n+1}$, the set of worlds of $\mathfrak{M}'$ is non-empty. It is straightforward to check both that $\mathfrak{M}'$ is based on a frame for L and that $\mathfrak{M}', w_0 \models \widehat{\varphi}$. ❑

Lemma 4.3.2 *Let $L \in \{\mathbf{K}, \mathbf{KTB}\}$. Then, φ is L-satisfiable if, and only if, $\widehat{\varphi}$ is L-satisfiable.*

Proof. Suppose that $\mathfrak{M}, w_0 \models \varphi$. To obtain a model satisfying $\widehat{\varphi}$, make p_{n+1} true at every world of $\mathfrak{M}$. Conversely, suppose that $\mathfrak{M}, w_0 \models \widehat{\varphi}$. In view of Lemma 4.3.1, we may assume that p_{n+1} is universally true in $\mathfrak{M}$. As φ is equivalent to $\widehat{\varphi}(p_{n+1}/\top)$, it follows that $\mathfrak{M}, w_0 \models \varphi$. ❑

Now, consider the following class M of finite models. For every $k \in \{1, \dots, n+1\}$, the class M contains a model $\mathfrak{M}_k$, depicted in Figure 4.1, that looks as follows. For brevity, we call some worlds p-worlds; if a world is

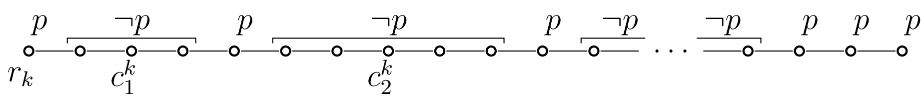

Figure 4.1: Model $\mathfrak{M}_k$

not a p-world, we call it a $\bar{p}$-world. The model $\mathfrak{M}_k$ is a chain of worlds whose root, r_k, is a p-world. The root is part of a pattern of worlds, described below, which is succeeded by three final p-worlds. The pattern looks as follows: a single p-world is followed by $2i+1$ $\bar{p}$-worlds, for $1 \leqslant i \leqslant k$. Thus, the chain looks as follows: the root (a p-world), then three $\bar{p}$-worlds, then a p-world, then five $\bar{p}$-worlds, then a p-world, ..., then a p-world, then $2k+1$ $\bar{p}$-worlds, then three p-worlds. The accessibility relation R_k between the worlds of $\mathfrak{M}_k$ is both reflexive and symmetric. To complete the definition of $\mathfrak{M}_k$, we define the propositional variable p to be true at exactly the p-worlds.

Before proceeding, we prove a lemma about the models in M. Given a model $\mathfrak{M}_k$, denote by c_i^k, for $i \in \{1, \ldots, k\}$, the "middle" world of the chain of $2i+1$ $\bar{p}$-worlds preceded and succeeded by p-worlds; see Figure 4.1. Also, let

$$\varepsilon_i \quad = \quad \Box^{\leqslant i}\neg p \wedge \Diamond^{i+1}p, \text{ where } i \in \mathbb{N}.$$

Lemma 4.3.3 *Let x be a world of $\mathfrak{M}_k$ that lies between r_k and c_i^k, for some $i \leqslant k$ (i.e., c_i^k cannot be reached from r_k by consecutive steps along R_k without passing through x). Then, $\mathfrak{M}_k, x \models \varepsilon_i$ if, and only if, $x = c_i^k$.*

Proof. Straightforward. ❑

We now define formulas we use to simulate the propositional variables of $\widehat{\varphi}$. First, inductively define, for every $k \in \{1, \ldots, n+1\}$, the following sequence of formulas:

$$\begin{aligned} \delta &= \Box^{+}p; \\ \delta_k^k &= \varepsilon_k \wedge \Diamond^{k+2}\delta; \\ \delta_i^k &= \varepsilon_i \wedge \Diamond^{2i+3}\delta_{i+1}^k, \text{ where } 1 \leqslant i < k. \end{aligned}$$

Next, let, for every $k \in \{1, \ldots, n+1\}$,

$$\alpha_k = p \wedge \Diamond^2 \delta_1^k$$

and

$$\beta_k = \neg p \wedge \Diamond \alpha_k.$$

Let σ be a (substitution) function that, given a formula ψ, replaces all occurrences of p_i in ψ by β_i, where $1 \leqslant i \leqslant n+1$. Finally, define

$$\varphi^* = \sigma(\widehat{\varphi})$$

to produce a single-variable formula φ^*.

Lemma 4.3.4 *Let $L \in \{\mathbf{K}, \mathbf{KTB}\}$. Then, φ is L-satisfiable if, and only if, φ^* is L-satisfiable.*

Proof. Suppose that φ is not L-satisfiable. Then, in view of Lemma 4.3.2, $\widehat{\varphi}$ is not L-satisfiable; hence, $\neg\widehat{\varphi} \in L$. Since L is closed under substitution, $\neg\varphi^* \in L$, and so φ^* is not L-satisfiable.

Suppose that φ is L-satisfiable. Then, in view of Lemmas 4.3.1 and 4.3.2, $\mathfrak{M}, w_0 \models \widehat{\varphi}$ for some $\mathfrak{M} = \langle W, R, V \rangle$, such that $\langle W, R \rangle$ is a frame for L and p_{n+1} is true at every $w \in W$, and for some $w_0 \in W$. (Recall that $\widehat{\varphi}$ only contains variables $p_1, \ldots, p_{n+1}$.) Define model $\mathfrak{M}'$ as follows. Attach to $\mathfrak{M}$ all the models from M; then, for every x in $\mathfrak{M}$, put $xR'r_m$ and $r_mR'x$, where r_m is the root of $\mathfrak{M}_m \in \mathsf{M}$, exactly when $\mathfrak{M}, x \models p_m$. Notice that r_{n+1} is accessible in $\mathfrak{M}'$ from every $x \in W$. Finally, make p true at exactly those worlds of the attached models where it was true, and make it false at every world in W. Notice that $\mathfrak{M}'$ is based on a frame for L.

To conclude the proof, it suffices to show that $\mathfrak{M}', w_0 \models \varphi^*$. To that end, we first prove two auxiliary Sublemmas:

Sublemma 4.3.5 *Let x be a world of $\mathfrak{M}'$ that lies between the root r_k of the attached model $\mathfrak{M}_k$ and the world c_i^k of $\mathfrak{M}_k$, for some $i \leqslant k$ (i.e., c_i^k cannot be reached from r_k by consecutive steps along R' without passing through x). Then, $\mathfrak{M}', x \models \varepsilon_i$ if, and only if, $x = c_i^k$.*

Proof. Straightforward, using Lemma 4.3.3. ❑

Sublemma 4.3.6 *Let $x \in W$ and let $\mathfrak{M}', x \models \Diamond\alpha_k$. Then, $xR'r_k$.*

Proof. Since $\mathfrak{M}', x \models \Diamond\alpha_k$, so $xR'y$ and $\mathfrak{M}', y \models \alpha_k$, for some y in $\mathfrak{M}'$. We show that $y = r_k$. Since $\mathfrak{M}', y \models p$, clearly $y \notin W$, and thus y is the root r_m

of some $\mathfrak{M}_m$. As $\mathfrak{M}', y \models \Diamond^2 \delta_1^k$, we can reach from y in two R'-steps a world y_1 such that $\mathfrak{M}', y_1 \models \varepsilon_1$. Since $wR'r_{n+1}$ holds for every $w \in W$, and thus $\mathfrak{M}', w \not\models \Box\neg p$ for every $w \in W$, we know that $y_1 \notin W$, so y_1 belongs to one of the attached models $\mathfrak{M}_j$. In two R'-steps, we cannot go past c_1^j for any j and can only reach c_1^j if $j = m$; hence, due to Sublemma 4.3.5, $y_1 = c_1^m$. Since $\mathfrak{M}', c_1^m \models \Diamond^5(\varepsilon_2 \wedge \delta_2^k)$, we can reach from c_1^m in five R'-steps a world y_2 such that $\mathfrak{M}', y_2 \models \varepsilon_2$. As $\mathfrak{M}', w \not\models \Box\neg p$ for every $w \in W$, we know that $y_2 \notin W$. In five R'-steps, we cannot go past c_2^j for any j and can only reach c_2^j if $j = m$; hence, due to Sublemma 4.3.5, $y_2 = c_2^m$, and so $\mathfrak{M}', c_2^m \models \Diamond^7(\varepsilon_3 \wedge \delta_3^k)$. We can now repeat the argument without worrying about the possibility of satisfying further formulas due to the presence in $\mathfrak{M}'$ of the worlds outside of $\mathfrak{M}_m$, as we cannot step outside of $\mathfrak{M}_m$, starting from c_2^m, in seven steps. By inductively repeating the argument m times, we arrive at the world c_m^m such that $\mathfrak{M}', c_m^m \models \Diamond^{k+2}\delta$, which can only happen if $m = k$. Thus, all along we have been evaluating the formulas in $\mathfrak{M}_k$, and thus $y = r_k$, as required. ❑

Now, we proceed with the proof of the main Lemma.

Recall that $\varphi^* = \sigma(\widehat{\varphi}) = \sigma(p_{n+1} \wedge \varphi') = \beta_{n+1} \wedge \sigma(\varphi')$. It is easy to check that $\mathfrak{M}', w_0 \models \beta_{n+1}$. It then suffices to show that $\mathfrak{M}, x \models \psi'$ if, and only if, $\mathfrak{M}', x \models \sigma(\psi')$, for every subformula ψ of φ and every $x \in W$. This can be done by induction on ψ.

For the base case, assume that $\mathfrak{M}', x \models \beta_i$; in particular, $\mathfrak{M}', x \models \Diamond\alpha_i$. Then, due to Sublemma 4.3.6, $xR'r_i$, and therefore $\mathfrak{M}, x \models p_i$ by definition of $\mathfrak{M}'$. The other direction is straightforward. The Boolean cases are also straightforward.

Let $\psi = \Box\chi$. Assume that $\mathfrak{M}', x \not\models \Box(\beta_{n+1} \to \sigma(\chi'))$. Then, $xR'y$, as well as $\mathfrak{M}', y \models \beta_{n+1}$ and $\mathfrak{M}', y \not\models \sigma(\chi')$, for some y in $\mathfrak{M}'$. In particular, $\mathfrak{M}', y \models \neg p$; thus, y cannot be the root of any of the attached models. Therefore, $y \in W$ and the inductive hypothesis is applicable; this gives us $\mathfrak{M}, x \not\models \Box(p_{n+1} \to \chi')$, as desired. The other direction is straightforward, using the converse of Sublemma 4.3.6. ❑

Given a formula φ, let

$$e(\varphi) = \neg((\neg\varphi)^*).$$

Theorem 4.3.7 *Let $L \in \{\mathbf{K}, \mathbf{KTB}\}$. Then, there exists a polynomial-time mapping that embeds L into its single-variable fragment.*

Proof. Take the mapping e defined above. ❑

Remark 4.3.8 *Notice that Lemma 4.3.4 and Theorem 4.3.7 apply to the logic* **KB***, as well. We did not mention* **KB** *in their statements as this is not required for the proof of our main result, Theorem 4.3.9.*

Theorem 4.3.9 *Let L be a logic in the interval* $[\mathbf{K}, \mathbf{KTB}]$*. Then, the single-variable fragment of L is* PSPACE*-hard.*

Proof. We reduce the PSPACE-hard complement of the set TQBF of true quantified Boolean formulas to the single-variable fragment of L. Let $\theta \notin$ TQBF; then, $t(\theta) \in \mathbf{K}$, where t is the translation defined in the proof of Theorem 4.2.1; hence, in view of Theorem 4.3.7, $e(t(\theta)) \in \mathbf{K}$, and thus $e(t(\theta)) \in L$. Let, on the other hand, $\theta \in$ TQBF; then, as shown in the proof of Theorem 4.2.1, $t(\theta) \notin \mathbf{KTB}$; hence, in view of Theorem 4.3.7, $e(t(\theta)) \notin \mathbf{KTB}$, and thus $e(t(\theta)) \notin L$. Thus, the polynomial-time computable translation $g(\theta) = e(t(\theta))$ reduces the complement of TQBF to the single-variable fragment of L; the statement of the Theorem follows. ❑

Theorem 4.3.9 implies that the single-variable fragments of all PSPACE-complete logics in $[\mathbf{K}, \mathbf{KTB}]$ are PSPACE-complete; in particular, we have the following:

Corollary 4.3.10 *The single-variable fragments of* **T***,* **KB***,* **KDB***, and* **KTB** *are* PSPACE*-complete.*

Note that the PSPACE-completeness of the single-variable fragment of **T** has been established in [40].

4.4 Conclusion

We have shown that when it comes to their computational properties, the modal logics of symmetric, as well as of reflexive and symmetric frames, behave in the same way as the logics of transitive, as well as of reflexive and transitive, frames—they remain intractable, namely, PSPACE-hard, when their languages are restricted to only one propositional variable.

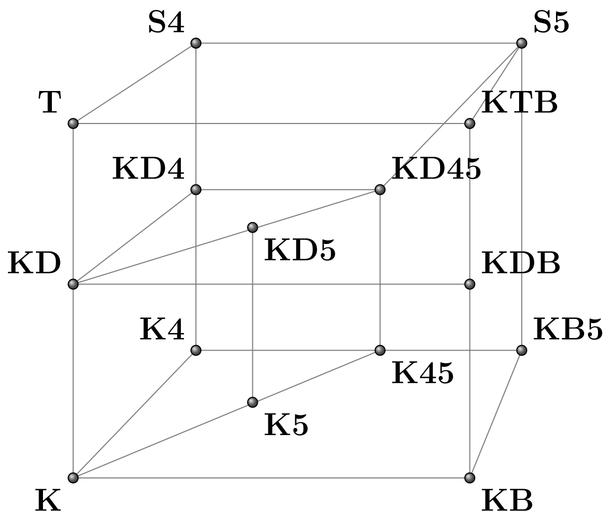

Figure 4.2: Cube of modal logics

Adding the axiom of symmetry to the logic of reflexive and transitive frames, i.e., **S4**, is not the only way of arriving at **S5**,—it can also be obtained, iter alia, by adding the axiom of Euclideanness, $\neg\Box p \to \Box\neg\Box p$, to **T**. The role of Euclideanness is well understood in the context of the present inquiry,—it has been shown in [59] that every extension of the logic of Euclidean frames, **K5**, is locally tabular; therefore, any finite-variable fragment of such a logic is polynomial-time decidable.

Thus, we have a good understanding of the role played by various properties of Kripke frames of most interest to "traditional" logicians (reflexivity, seriality, symmetry, transitivity, and Euclideanness)—represented by logics included in the "cube of modal logics" [27], see Figure 4.2—in the computational behaviour of the finite-variable fragments of the corresponding logics: while Euclideanness—as well as symmetry combined with transitivity, which imply Euclideanness—make such fragments "tractable", reflexivity, symmetry, transitivity, and seriality by themselves—as well as transitivity and symmetry combined either with reflexivity or seriality—do not have this effect (seriality, along with reflexivity and transitivity, has been considered in [10]).

This raises the more general question of where the borderline lies between, on the one hand, logics that behave like those described in [40], [44], [82], [10], and in this chapter, i.e., whose finite-variable fragments remain intractable, and on the other, those that behave like **S5**, i.e., whose finite-variable fragments are simpler than entire logics (in all the cases known in the literature, this amounts to having polynomial-time decidable finite-variable fragments). It is clear that the answer is not directly linked to the complexity of the logic in question—as shown in [10], satisfiability for single- and two-variable fragments of such logics as **S4.3**, **GL.3**, and **Grz.3**, whose satisfiability problem is NP-complete, is also NP-complete. While the borderline between the NP-hard and the PSPACE-hard in modal logic has received attention in the literature (see, for example, [41]), this question has not, as far as we know, been so far addressed.

Chapter 5

Undecidability of first-order modal and intuitionistic logics with two variables and one monadic predicate letter

5.1 Introduction

While the (first-order) quantified classical logic **QCl** is undecidable [12], it contains a number of rather expressive decidable fragments [9]. This has long stimulated interest in drawing the borderline between decidable and undecidable fragments of **QCl** using a variety of criteria, in isolation or in combination, imposed on the language. One such criterion is the number and arity of predicate letters allowed in the language: while the monadic fragment is decidable [5], the fragment containing a single binary letter is not, as follows from [29]. Another criterion is the number of individual variables allowed in the language: while the two-variable fragment is decidable [58, 39], the three-variable fragment is not [76].

Similar questions have long been of interest in (first-order) quantified intuitionistic and modal logics. For languages without restrictions on the number of individual variables, Kripke [50] has shown that all "natural" quantified modal logics with two monadic predicate letters are undecidable, while Maslov, Mints, and Orevkov [54] and, independently, Gabbay [26] have shown that quantified intuitionistic logic with a single monadic predicate letter is undecidable.

The question of where the borderline lies in the intuitionistic and modal case when it comes to the number of individual variables allowed in the language has recently been investigated by Kontchakov, Kurucz, and Zakharyschev in [47]. It is shown in [47] that two-variable fragments of quantified intuitionistic and all "natural" modal logics are undecidable. Moreover, it is established in [47] that, to obtain undecidability of two-variable fragments, in the intuitionistic case, it suffices to use two binary and infinitely many monadic predicate letters, while in the modal case, it suffices to use only (infinitely many) monadic predicate letters.

Two questions were raised in [47] concerning the languages combining restrictions on the number of individual variables and predicate letters: first, how many monadic predicate letters are needed to obtain undecidability of the two-variable fragments in the modal case, and second, whether it suffices to use monadic predicate letters to obtain undecidability of the two-variable fragment in the intuitionistic case.

In the present chapter, we address both of the aforementioned questions. First, we show that for two-variable fragments of most modal logics considered in [47], it suffices to use a single monadic predicate letter to obtain undecidability. Second, we show that the positive fragment of quantified intuitionistic logic **QInt** is undecidable in the language with two variables and a single monadic predicate letter. We also show that the latter result holds true for all logics in intervals [**QBL**, **QKC**] and [**QBL**, **QFL**], where **QKC** is the logic of the weak law of the excluded middle and **QBL** and **QFL** are first-order counterparts of Visser's basic and formal logics, respectively.

The chapter is structured as follows. In section 5.2, we prove undecidability results about modal logics. In section 5.3, we do likewise for the intuitionistic and related logics. We conclude, in section 5.4, by discussing how our results can be applied in settings not considered in this chapter and pointing out some open questions following from our work.

5.2 Modal logics

In this section, we prove undecidability results about two-variable fragments of quantified modal logics with a single monadic predicate letter. This is essentially achieved by adapting to the first-order language of Halpern's technique [40] for establishing complexity results for single-variable fragments of propositional modal logics.

5.2.1 Syntax and semantics

A (first-order) quantified modal language contains countably many individual variables; countably many predicate letters of every arity; Boolean connectives $\wedge$ and $\neg$; modal connective $\Box$; and a quantifier $\forall$. Formulas as well as the symbols $\vee$, $\rightarrow$, $\exists$, and $\Diamond$ are defined in the usual way. We also use the following abbreviations: $\Box^+\varphi = \varphi \wedge \Box\varphi$ and $\Diamond^+\varphi = \varphi \vee \Diamond\varphi$.

A Kripke frame is a tuple $\mathfrak{F} = \langle W, R \rangle$, where W is a non-empty set (of worlds) and R is a binary (accessibility) relation on W. A predicate Kripke frame is a tuple $\mathfrak{F}_D = \langle W, R, D \rangle$, where $\langle W, R \rangle$ is a Kripke frame and D is a function from W into a set of non-empty subsets of some set (the domain of $\mathfrak{F}_D$), satisfying the condition that wRw' implies $D(w) \subseteq D(w')$. We call the set $D(w)$ the domain of w. We will also be interested in predicate frames satisfying the condition that wRw' implies $D(w) = D(w')$; we refer to such frames as frames with constant domains.

A Kripke model is a tuple $\mathfrak{M} = \langle W, R, D, I \rangle$, where $\langle W, R, D \rangle$ is a predicate Kripke frame and I is a function assigning to a world $w \in W$ and an n-ary predicate letter P an n-ary relation $I(w, P)$ on $D(w)$. We refer to I as the interpretation of predicate letters with respect to worlds in W.

An assignment in a model is a function g associating with every individual variable x an element of the domain of the underlying frame.

The truth of a formula φ in a world w of a model $\mathfrak{M}$ under an assignment g is inductively defined as follows:

- $\mathfrak{M}, w \models^g P(x_1, \ldots, x_n)$ if $\langle g(x_1), \ldots, g(x_n) \rangle \in I(w, P)$;
- $\mathfrak{M}, w \models^g \varphi_1 \wedge \varphi_2$ if $\mathfrak{M}, w \models^g \varphi_1$ and $\mathfrak{M}, w \models^g \varphi_2$;
- $\mathfrak{M}, w \models^g \neg\varphi_1$ if $\mathfrak{M}, w \not\models^g \varphi_1$;
- $\mathfrak{M}, w \models^g \Box\varphi_1$ if wRw' implies $\mathfrak{M}, w' \models^g \varphi_1$, for every $w' \in W$;
- $\mathfrak{M}, w \models^g \forall x\, \varphi_1$ if $\mathfrak{M}, w \models^{g'} \varphi_1$, for every assignment g' such that g' differs from g in at most the value of x and such that $g'(x) \in D(w)$.

Note that, given a Kripke model $\mathfrak{M} = \langle W, R, D, I \rangle$ and $w \in W$, the tuple $\mathfrak{M}_w = \langle D_w, I_w \rangle$, where $D_w = D(w)$ and $I_w(P) = I(w, P)$, is a classical predicate model.

We say that φ is true at world w of model $\mathfrak{M}$ and write $\mathfrak{M}, w \models \varphi$ if $\mathfrak{M}, w \models^g \varphi$ holds for every g assigning to free variables of φ elements of $D(w)$. We say that φ is true in $\mathfrak{M}$ and write $\mathfrak{M} \models \varphi$ if $\mathfrak{M}, w \models \varphi$ holds for every world w of $\mathfrak{M}$. We say that φ is true in predicate frame $\mathfrak{F}_D$ and write $\mathfrak{F}_D \models \varphi$ if φ is true in every model based on $\mathfrak{F}_D$. We say that φ is true in frame $\mathfrak{F}$ and write $\mathfrak{F} \models \varphi$ if φ is true in every predicate frame of the form $\mathfrak{F}_D$. Finally, we say that a formula is true in a class of frames if it is true in every frame from the class.

Let $\mathfrak{M} = \langle W, R, D, I \rangle$ be a model, $w \in W$, and $a_1, \dots, a_n \in D(w)$. Let $\varphi(x_1, \dots, x_n)$ be a formula whose free variables are among $x_1, \dots, x_n$. We write $\mathfrak{M}, w \models \varphi[a_1, \dots, a_n]$ to mean $\mathfrak{M}, w \models^g \varphi(x_1, \dots, x_n)$, where $g(x_1) = a_1, \dots, g(x_n) = a_n$.

Given a propositional normal modal logic L, let $\mathbf{Q}L$ be $\mathbf{QCl} \oplus L$ where $\oplus$ is the operation of closure under (predicate) substitution, modus ponens, generalization, and necessitation. Of particular interest to us are the quantified counterparts **QGL**, **QGrz**, and **QKTB** of propositinal logics **GL**, **QGrz**, and **KTB**. We recall that **GL** is the logic of Kripke frames whose accessibility relation is irreflexive, transitive, and contains no infinite ascending chains, while **Grz** is the logic of frames whose accessibility relation is reflexive, transitive, antisymmetric, and does not contain infinite ascending chains of pairwise distinct worlds (in other words, the accessibility relation on the frames for **Grz** is the reflexive closure of the one on the frames for **GL**). We also recall that **QGL** and **QGrz** are Kripke-incomplete [57, 70], but are valid on all the frames for **GL** and **Grz**, respectively. Thus, for technical reasons—namely, to avoid being distracted with Kripke-completeness—we define logics $\mathbf{QGL}^{sem}$ and $\mathbf{QGrz}^{sem}$ as the sets of quantified formulas true in all the frames of **GL** and **Grz**, respectively. What is important for us is that $\mathbf{QGL} \subseteq \mathbf{QGL}^{sem}$ and $\mathbf{QGrz} \subseteq \mathbf{QGrz}^{sem}$. Lastly, we recall that **KTB** is the logic of Kripke frames whose accessibility relation is reflexive and symmetric and that **QKTB** is complete with respect to this class of frames.

Given a logic L and a closed formula φ in the language of L, we say that φ is L-satisfiable if $\neg\varphi \notin L$. If L is complete with respect to a class $\mathfrak{C}$ of frames, L-satisfiability of φ amounts to φ being true at a world of a model based on a frame in $\mathfrak{C}$.

We now turn to addressing the question, raised in [47], of how many monadic predicate letters are needed in the language of quantified modal logics to obtain undecidability of their two-variable fragments. Using suitable adaptations of a technique originally proposed in [40], and further refined in [10], [68], and [69], for propositinal languages, we show that all sublogics of **QGL**, **QGrz**, and **QKTB** are undecidable in the language with a single monadic predicate letter.

5.2.2 Sublogics of QGL and QGrz

It is established in [47], Theorem 3, that two-variable fragments of a wide variety of quantified modal logics in the language with infinitely many monadic predicate letters are undecidable.

To that end, it is shown in [47] how, given an instance T of an undecidable tiling problem [6], one can effectively compute a formula ξ_T containing only monadic predicate letters and two individual variables such that T tiles $\mathbb{N}\times\mathbb{N}$ if, and only if, ξ_T is satisfiable in a logic L such that L is the set of quantified formulas valid on all the frames of a propositional logic valid on a frame containing a world that can see all worlds from an infinite set V_1, each of which can in its turn see infinitely many worlds from an infinite set V_2 disjoint from V_1. Note that the logics **QK**, $\mathbf{QGL}^{sem}$, and $\mathbf{QGrz}^{sem}$ satisfy the above condition; indeed, for **GL** and **Grz**, we can take V_1 and V_2 to be infinite antichains of irreflexive and reflexive worlds, respectively. As formulas ξ_T are computed in the same way for all of **QK**, $\mathbf{QGL}^{sem}$, and $\mathbf{QGrz}^{sem}$, it follows that if T does not tile $\mathbb{N}\times\mathbb{N}$, then ξ_T is not satisfiable in any of them, and if T tiles $\mathbb{N}\times\mathbb{N}$, then ξ_T is satisfiable in each of them. Let L be a logic such that $\mathbf{QK}\subseteq L\subseteq\mathbf{QGL}^{sem}$. If T does not tile $\mathbb{N}\times\mathbb{N}$, then $\neg\xi_T\in\mathbf{QK}$, and thus $\neg\xi_T\in L$. If, on the other hand, T tiles $\mathbb{N}\times\mathbb{N}$, then $\neg\xi_T\notin\mathbf{QGL}^{sem}$, and thus $\neg\xi_T\notin L$. This gives us a reduction of (the complement of) the undecidable tiling problem to L using formulas with only monadic predicate letters and two individual variables. Thus, every logic in $[\mathbf{QK},\mathbf{QGL}]$ is undecidable in the language with a single monadic letter and two individual variables; a similar argument can be made for logics in $[\mathbf{QK},\mathbf{QGrz}]$.

As formulas ξ_T are computed in the same way for all the logics in $[\mathbf{QK},\mathbf{QGL}]$ and $[\mathbf{QK},\mathbf{QGrz}]$, all formulas $\neg\xi_T$ corresponding to "bad" instances of the tiling problem [6] make up an undecidable fragment, F, that belongs to every logic in $[\mathbf{QK},\mathbf{QGL}]$ and $[\mathbf{QK},\mathbf{QGrz}]$. In the rest of this section, we effectively embed F, using an embedding e that does not increase the number of individual variables in a formula, into a fragment, F^e, containing a single monadic predicate letter and belonging to every logic in $[\mathbf{QK},\mathbf{QGL}]$ and $[\mathbf{QK},\mathbf{QGrz}]$. To that end, given a formula φ, we effectively construct, using e, the formula φ^e, such that $\varphi\in F$ if, and only if, $\varphi^e\in F^e$;

as φ^e contains the same number of individual variables as φ, our main result in this section immediately follows.

Let φ be a (closed) formula containing monadic predicate letters $P_1, \ldots, P_n$. Let P_{n+1} be a monadic predicate letter distinct from $P_1, \ldots, P_n$ and let $B = \forall x\, P_{n+1}(x)$. Define an embedding $\cdot'$ as follows:

$$
\begin{array}{lcl}
P_i(x)' & = & P_i(x), \quad \text{where } i \in \{1, \ldots, n\}; \\
(\neg\phi)' & = & \neg\phi'; \\
(\phi \wedge \psi)' & = & \phi' \wedge \psi'; \\
(\forall x\, \phi)' & = & \forall x\, \phi'; \\
(\Box\phi)' & = & \Box(B \to \phi').
\end{array}
$$

Lemma 5.2.1 *Let $L \in \{\mathbf{QK}, \mathbf{QGL}^{sem}, \mathbf{QGrz}^{sem}\}$. Then, φ is L-satisfiable if, and only if, $B \wedge \varphi'$ is L-satisfiable.*

Proof. Assume that $\mathfrak{M}, w_0 \models \varphi$, for some $\mathfrak{M}$ based on a frame for L and some w_0. Let $\mathfrak{M}'$ be a model that extends $\mathfrak{M}$ by setting $I(w, P_{n+1}) = D(w)$, for every $w \in W$. Then, $\mathfrak{M}', w_0 \models B \wedge \varphi'$. Conversely, assume that $\mathfrak{M}, w_0 \models B \wedge \varphi'$, for some $\mathfrak{M}$ based on a frame for L. Let $\mathfrak{M}'$ be a submodel of $\mathfrak{M}$ with $W' = \{w : \mathfrak{M}, w \models B\}$. Then, $\mathfrak{M}', w_0 \models \varphi$. Note that, for every logic L in the statement of the Lemma, $\mathfrak{M}'$ is based on a frame for L. ❑

Remark 5.2.2 *In view of the proof of Lemma 5.2.1, if $B \wedge \varphi'$ is satisfied in a model $\mathfrak{M}$, we can assume, without a loss of generality, that B is true in $\mathfrak{M}$.*

Now, given a monadic predicate letter P, we inductively define the following sequence of formulas:

$$
\begin{array}{lcl}
\delta_1(x) & = & P(x) \wedge \Diamond(\neg P(x) \wedge \Diamond\Box^+ P(x)); \\
\delta_{m+1}(x) & = & P(x) \wedge \Diamond(\neg P(x) \wedge \Diamond\delta_m(x)).
\end{array}
$$

Using formulas from this sequence, define, for every $k \in \{1, \ldots, n+1\}$, the formula

$$\alpha_k(x) = \delta_k(x) \wedge \neg\delta_{k+1}(x) \wedge \Diamond\Box^+\neg P(x).$$

We now define models associated with formulas $\alpha_k(x)$. For every $k \in \{1, \ldots, n+1\}$, let $\mathfrak{F}_k = \langle W_k, R_k \rangle$ be a Kripke frame where $W_k = \{w_k^0, \ldots, w_k^{2k}\} \cup \{w_k^*\}$ and R_k is the transitive closure of the relation $\{\langle w_k^i, w_k^{i+1} \rangle : 0 \leqslant i < 2k\} \cup \{\langle w_k^0, w_k^* \rangle\}$. For every such k, let

$\mathfrak{M}_k = \langle W_k, R_k, D, I\rangle$ be a model with constant domains and let a be an individual in the domain of every $\mathfrak{M}_k$ (other than that, the relationship between the domains of $\mathfrak{M}_k$s is immaterial at this point). We say that $\mathfrak{M}_k$ is a-suitable if

$$\mathfrak{M}_k, w \models P[a] \iff w = w_k^{2i},\ \text{for } i \in \{0, \dots, k\}.$$

Lemma 5.2.3 *Let a be an individual in the domain of the models $\mathfrak{M}_1, \dots, \mathfrak{M}_{n+1}$ and let $\mathfrak{M}_1, \dots, \mathfrak{M}_{n+1}$ be a-suitable. Then,*

$$\mathfrak{M}_k, w \models \alpha_m[a] \iff k = m \text{ and } w = w_k^0.$$

Proof. Straightforward. ❑

Remark 5.2.4 *Notice that the statement of Lemma 5.2.3 holds true if we replace the accessibility relations in $\mathfrak{M}_1, \dots, \mathfrak{M}_{n+1}$ with their reflexive closures.*

Now, for every $\alpha_k(x)$, where $k \in \{1, \dots, n+1\}$, define

$$\beta_k(x) = \neg P(x) \wedge \Diamond \alpha_k(x).$$

Let φ^* be the result of replacing in φ' of $P_k(x)$ with $\beta_k(x)$, for every $k \in \{1, \dots, n+1\}$.

Lemma 5.2.5 *Let $L \in \{\mathbf{QK}, \mathbf{QGL}^{sem}, \mathbf{QGrz}^{sem}\}$. Then, $B \wedge \varphi'$ is L-satisfiable if, and only if, $\forall x\, \beta_{n+1}(x) \wedge \varphi^*$ is L-satisfiable.*

Proof. The right-to-left direction follows from the closure of L under predicate substitution. For the other direction, suppose that $B \wedge \varphi'$ is **QK**-satisfiable. Let $\mathfrak{M} = \langle W, R, D, I\rangle$ be a model such that $\mathfrak{M}, w_0 \models B \wedge \varphi'$, for some $w_0 \in W$. In view of Remark 5.2.2, we may assume, without a loss of generality, that $\mathfrak{M} \models B$.

For every $w \in W$ and every frame $\mathfrak{F}_k$ ($1 \leqslant k \leqslant n+1$), let $\mathfrak{F}_k^w = \langle \{w\} \times W_k, R_k^w\rangle$ be an isomorphic copy of $\mathfrak{F}_k$. For every $w \in W$ and $k \in \{1, \dots, n+1\}$, add $\{w\} \times W_k$ to W to obtain the set W^*. Define the relation R^* on W^* as follows:

$$R^* = R \cup \bigcup \big\{ R_k^w \cup \{\langle w, (w, w_k^0)\rangle\} : w \in W, 1 \leqslant k \leqslant n+1 \big\}.$$

Thus, for every $w \in W$, we make the roots of frames $\mathfrak{F}_1^w, \ldots, \mathfrak{F}_{n+1}^w$ accessible from w. Next, for every $u \in W^*$ let

$$D^*(u) = \begin{cases} D(u), & \text{if } u \in W, \\ D(w), & \text{if } u \in \{w\} \times W_k. \end{cases}$$

Finally, for every $u \in W^*$ and every $a \in D^*(u)$, let

$$\begin{array}{lcl} \langle a \rangle \in I^*(u, P) & \leftrightharpoons & u = (w, w_k^{2i}), \text{ for some } w \in W,\ k \in \{1, \ldots, n+1\}, \\ & & \text{and } i \in \{0, \ldots, k\}; \text{ and } \mathfrak{M}, w \models P_k[a]. \end{array}$$

Let $\mathfrak{M}^* = \langle W^*, R^*, D^*, I^* \rangle$. It immediately follows from Lemma 5.2.3 that, for every $w \in W$, every $a \in D(w)$, and every $k \in \{1, \ldots, n+1\}$,

$$\mathfrak{M}, w \models P_k[a] \iff \mathfrak{M}^*, w \models \beta_k[a].$$

We can then show that, for every $w \in W$, every subformula $\psi(x_1, \ldots, x_m)$ of φ, and every $a_1, \ldots, a_m \in D(w)$,

$$\mathfrak{M}, w \models \psi'[a_1, \ldots, a_m] \iff \mathfrak{M}^*, w \models \psi^*[a_1, \ldots, a_m],$$

where $\psi^*(x_1, \ldots, x_m)$ is obtained by substituting $\beta_1(x), \ldots, \beta_{n+1}(x)$ for $P_1(x), \ldots, P_{n+1}(x)$ in $\psi'(x_1, \ldots, x_m)$.

The proof proceeds by induction. We only consider the modal case, leaving the rest to the reader. In this case, $\psi'(x_1, \ldots, x_m) = \Box(\forall x\, P_{n+1}(x) \to \chi'(x_1, \ldots, x_m))$ and $\psi^*(x_1, \ldots, x_m) = \Box(\forall x\, \beta_{n+1}(x) \to \chi^*(x_1, \ldots, x_m))$. If $\mathfrak{M}^*, w \not\models \psi^*[a_1, \ldots, a_m]$, then there exists $w' \in W^*$ with wR^*w' such that $\mathfrak{M}^*, w' \models \forall x\, \beta_{n+1}(x)$ and $\mathfrak{M}^*, w' \not\models \chi^*[a_1, \ldots, a_m]$. The condition $\mathfrak{M}^*, w' \models \forall x\, \beta_{n+1}(x)$ guarantees that $w' \in W$; therefore, we may apply the inductive hypothesis to conclude that $\mathfrak{M}, w' \not\models \chi'[a_1, \ldots, a_m]$. The other direction is straightforward.

Thus, $\mathfrak{M}^*, w_0 \models \forall x\, \beta_{n+1}(x) \wedge \varphi^*$, i. e., $\forall x\, \beta_{n+1}(x) \wedge \varphi^*$ is **QK**-satisfiable.

For $\mathbf{QGL}^{sem}$ and $\mathbf{QGrz}^{sem}$, the proof is similar. The only difference is that, when defining the model $\mathfrak{M}^*$, instead of R^* mentioned above, we take as the accessibility relations its transitive, and its reflexive and transitive, closure, respectively. ❑

We can now prove our main result in this section.

Theorem 5.2.6 *Let L be a logic such that* $\mathbf{QK} \subseteq L \subseteq \mathbf{QGL}$ *or* $\mathbf{QK} \subseteq L \subseteq \mathbf{QGrz}$*. Then, L is undecidable in the language with two individual variables and a single monadic predicate letter.*

Proof. Given a formula φ with two individual variables and only monadic predicate letters, let $e(\varphi) = \forall x\, \beta_{n+1}(x) \wedge \varphi^*$. Let $\neg\xi_T$ be a formula corresponding to a "bad" instance T of the tiling problem [6]. Due to Lemmas 5.2.1 and 5.2.5, $\neg\xi_T \in \mathbf{QK}$ if, and only if, $\neg e(\xi_T) \in \mathbf{QK}$; likewise, $\neg\xi_T \in \mathbf{QGL}^{sem}$ if, and only if, $\neg e(\xi_T) \in \mathbf{QGL}^{sem}$. As noticed at the beginning of this section, all such $\neg\xi_T$ make up an undecidable fragment, F, belonging to every logic in $[\mathbf{QK}, \mathbf{QGL}^{sem}]$ and $[\mathbf{QK}, \mathbf{QGrz}^{sem}]$. Therefore, every such logic contains an undecidable fragment $F^e = \{\neg e(\xi_T) : \neg\xi_T \in F\}$ made up of formulas with two individual variables and a single monadic predicate letter. The statement of the Theorem follows. ❑

Corollary 5.2.7 **QK**, **QT**, **QD**, **QK4**, **QS4**, **QGL**, *and* **QGrz** *are undecidable in the language with two individual variables and a single monadic predicate letter.*

Remark 5.2.8 *Theorem 5.2.6 and Corollary 5.2.7 hold true if we replace every logic L mentioned in their statements with $L \oplus bf$, where $bf = \forall x\, \Box P(x) \to \Box \forall x\, P(x)$; adding bf to L forces us to consider only predicate frames for L with constant domains.*

We conclude this section by noticing that the results obtained herein are quite tight. In has been shown in [85], Theorem 5.1, that for logics **QK**, **QT**, **QK4**, and **QS4**, adding—on top of the restriction to at most two individual variables and a single monadic predicate letter—the very slight restriction that modal operators apply only to formulas with at most one free individual variable results in decidable fragments. As noticed in [85], the same holds true for the other logics mentioned in Corollary 5.2.7.

5.2.3 Sublogics of QKTB

We now prove results similar to those established in the preceding section for logics in the interval $[\mathbf{QK}, \mathbf{QKTB}]$, where **QKTB** is the predicate logic of reflexive and symmetric frames. In so doing, we use an adaptation of a technique used in [69] for proving results about computational complexity of finite-variable fragments of sublogics of the propositional logic **KTB**.

We proceed as in the previous section right up to the point where formulas α_k and models $\mathfrak{M}_k$ are defined. Then, we define the formulas α_k as follows. First, let

$$\begin{array}{rclcrcl} \Box^0\varphi & = & \varphi, & \qquad & \Box^{\leqslant 0}\varphi & = & \varphi, \\ \Box^{n+1}\varphi & = & \Box\Box^n\varphi, & & \Box^{\leqslant n+1}\varphi & = & \Box^{\leqslant n}\varphi \wedge \Box^{n+1}\varphi, \\ \Diamond^n\varphi & = & \neg\Box^n\neg\varphi, & & \Diamond^{\leqslant n}\varphi & = & \neg\Box^{\leqslant n}\neg\varphi. \end{array}$$

Next, inductively define, for every $k \in \{1, \ldots, n+1\}$, the following sequence of formulas:

$$\begin{array}{rcl} \delta(x) & = & \Box^+ P(x); \\ \delta_k^k(x) & = & \Box^{\leqslant k}\neg P(x) \wedge \Diamond^{k+1} P(x) \wedge \Diamond^{k+2}\delta(x); \\ \delta_i^k(x) & = & \Box^{\leqslant i}\neg P(x) \wedge \Diamond^{i+1} P(x) \wedge \Diamond^{2i+3}\delta_{i+1}^k(x), \text{ where } 1 \leqslant i < k, \end{array}$$

and let, for every $k \in \{1, \ldots, n+1\}$,

$$\alpha_k(x) = P(x) \wedge \Diamond^2\delta_1^k(x).$$

Now we define models $\mathfrak{M}_k$ associated with formulas α_k. Given an individual a and $k \in \{1, \ldots, n+1\}$, a model $\mathfrak{M}_k$, whose domain contains a, looks as follows. For brevity, we call some worlds a-worlds; if a world is not an a-world, we call it an $\bar{a}$-world. The model is a chain of worlds whose root, r_k, is an a-world. The root is part of a pattern of worlds, described below, which is in turn succeeded by three final a-worlds. The pattern looks as follows: a single a-world is followed by $2i+1$ $\bar{a}$-worlds, for $1 \leqslant i \leqslant k$. Thus the chain looks as follows: the root (an a-world), then three $\bar{a}$-worlds, then an a-world, then five $\bar{a}$-worlds, then an a-world, ..., then an a-world, then $2k+1$ $\bar{a}$-worlds, then three a-worlds. The accessibility relation between the worlds of $\mathfrak{M}_k$ is both reflexive and symmetric.

We say that $\mathfrak{M}_k$ is a-suitable if

$$\mathfrak{M}_k, w \models P[a] \iff w \text{ is an } a\text{-world}.$$

We can, then, prove the following analogue of Lemma 5.2.3.

Lemma 5.2.9 *Let a be an individual in the domain of the models $\mathfrak{M}_1, \ldots, \mathfrak{M}_{n+1}$ and let $\mathfrak{M}_1, \ldots, \mathfrak{M}_{n+1}$ be a-suitable. Then,*

$$\mathfrak{M}_k, w \models \alpha_m[a] \iff k = m \text{ and } w = r_k.$$

Proof. Straightforard. ❑

As before, let

$$\beta_k(x) = \neg P(x) \wedge \Diamond \alpha_k(x),$$

and let φ^* be the result of replacing in φ' of $P_k(x)$ with $\beta_k(x)$, for every $k \in \{1, \ldots, n+1\}$.

We can then prove the following analogue of Lemma 5.2.5:

Lemma 5.2.10 *Let* $L \in \{\mathbf{QK}, \mathbf{QKTB}\}$*. Then,* $B \wedge \varphi'$ *is* L*-satisfiable if, and only if,* $\forall x\, \beta_{n+1}(x) \wedge \varphi^*$ *is* L*-satisfiable.*

Proof. Analogous to the proof of Lemma 5.2.5, with the observation that the truth status of formulas α_k is not changed at the worlds of the models $\mathfrak{M}_k$ once they get attached to the model $\mathfrak{M}$ satisfying the formula $B \wedge \varphi'$ to obtain the model $\mathfrak{M}^*$ satisfying the formula $\forall x\, \beta_{n+1}(x) \wedge \varphi^*$, even though their roots can now see the worlds of $\mathfrak{M}$ due to the symmetry of the accessibility relation of $\mathfrak{M}^*$. For a detailed argument showing that the truth status of formulas α_k in $\mathfrak{M}^*$ at worlds from $\mathfrak{M}_k$ is not affected, we refer the reader to the proof of Lemma 3.9 in [69]. ❑

Then, using an argument analogous to the one used in the proof of Theorem 5.2.6, we obtain the following:

Theorem 5.2.11 *Let* L *be a logic such that* $\mathbf{QK} \subseteq L \subseteq \mathbf{QKTB}$*. Then,* L *is undecidable in the language with two individual variables and a single monadic predicate letter.*

Corollary 5.2.12 **QKB** *and* **QKTB** *are undecidable in the language with two individual variables and a single monadic predicate letter.*

5.3 Intuitionistic and related logics

We now consider logics closely related to the quantified intuitionistic logic **QInt**.

5.3.1 Syntax and semantics

The (first-order) quantified intuitionistic language contains countably many individual variables; countably many predicate letters of every arity; propositional constants $\bot$ ("falsehood") and $\top$ ("truth"); propositional connectives $\wedge$, $\vee$, and $\to$; and quantifiers $\exists$ and $\forall$. Formulas are defined in the usual way; when parentheses are left out, $\wedge$ and $\vee$ are understood to bind tighter than $\to$. We also use the following abbreviations: $\Box\varphi = \top \to \varphi$, $\Box^0\varphi = \varphi$, and $\Box^{n+1}\varphi = \Box\Box^n\varphi$.

A Kripke frame is a tuple $\mathfrak{F} = \langle W, R\rangle$, where W is a non-empty set (of worlds) and R is a binary (accessibility) relation on W that is reflexive, anti-symmetric, and transitive.

A Kripke model $\mathfrak{M} = \langle W, R, D, I\rangle$ is defined as in the modal case, except that the interpretation function I satisfies the additional condition that wRw' implies $I(w, P) \subseteq I(w', P)$. An assignment is defined as in the modal case.

The truth of a formula φ in a world w of a model $\mathfrak{M}$ under an assignment g is inductively defined as follows:

- $\mathfrak{M}, w \not\models^g \bot$;
- $\mathfrak{M}, w \models^g \top$;
- $\mathfrak{M}, w \models^g P(x_1, \ldots, x_n)$ if $\langle g(x_1), \ldots, g(x_n)\rangle \in I(w, P)$;
- $\mathfrak{M}, w \models^g \varphi_1 \wedge \varphi_2$ if $\mathfrak{M}, w \models^g \varphi_1$ and $\mathfrak{M}, w \models^g \varphi_2$;
- $\mathfrak{M}, w \models^g \varphi_1 \vee \varphi_2$ if $\mathfrak{M}, w \models^g \varphi_1$ or $\mathfrak{M}, w \models^g \varphi_2$;
- $\mathfrak{M}, w \models^g \varphi_1 \to \varphi_2$ if wRw' and $\mathfrak{M}, w' \models^g \varphi_1$ imply $\mathfrak{M}, w' \models^g \varphi_2$, for every $w' \in W$;
- $\mathfrak{M}, w \models^g \exists x\, \varphi_1$ if $\mathfrak{M}, w \models^{g'} \varphi_1$, for some assignment g' that differs from g at most in the value of x and such that $g'(x) \in D(w)$;
- $\mathfrak{M}, w \models^g \forall x\, \varphi_1$ if $\mathfrak{M}, w' \models^{g'} \varphi_1$, for every $w' \in W$ such that wRw' and every assignment g' such that g' differs from g in at most the value of x and such that $g'(x) \in D(w')$.

Truth in models, frames, and classes of frames is defined as in the modal case. **QInt** is the set of formulas true in all frames.

We also consider some logics closely related to **QInt**. First, **QKC** is the quantified counterpart of the propositional logic $\mathbf{KC} = \mathbf{Int} + \neg p \vee \neg\neg p$. Semantically, **QKC** is characterized by the frames that satisfy the (convergence) condition that wRv_1 and wRv_2 imply the existence of a world u such that v_1Ru and v_2Ru.

Second, we consider quantified counterparts of Visser's basic propositional logic **BPL** and formal propositional logic **FPL** [81]: **BPL** and **FPL** are logics in the intuitionistic language whose modal companions are **K4** and **GL**—that is, given the Gödel's translation t of the intuitionistic language into the modal one (see, for example, [11], § 3.9), $\mathbf{BPL} = t^{-1}(\mathbf{K4})$ and $\mathbf{FPL} = t^{-1}(\mathbf{GL})$. Therefore, we define their quantified counterparts as logics $\mathbf{QBL} = T^{-1}(\mathbf{QK4})$ and $\mathbf{QFL} = T^{-1}(\mathbf{QGL})$, where T is the extension of t with the following clauses: $T(\exists x\, \varphi) = \exists x\, T(\varphi)$; and $T(\forall x_1 \ldots \forall x_n\, \varphi) = \Box\forall x_1 \ldots \forall x_n\, T(\varphi)$, where φ does not begin with a universal quantifier. To give the semantic account of **QBL** and **QFL**, we use Kripke frames and models as defined for **QInt**, except that the accessibility relation is now only required to be anti-symmetric and transitive. The relation $\mathfrak{M}, w \models^g \varphi$ is defined as in the intuitionistic case, with the following modification for the universal quantifiers:

- $\mathfrak{M}, w \models^g \forall x_1 \ldots \forall x_n\, \varphi_1$, where φ_1 does not begin with a universal quantifier, if $\mathfrak{M}, w' \models^{g'} \varphi_1$, for every $w' \in W$ such that wRw' and every assignment g' such that g' differs from g in at most the values of $x_1, \ldots, x_n$ and such that $g'(x_1), \ldots, g'(x_n) \in D(w')$.

This clause is required to make, in the absence of reflexivity of the accessibility relation, the formula $\forall x \forall y\, \varphi$ equivalent to the formula $\forall y \forall x\, \varphi$. Then, **QBL** is sound (and complete) with respect to all such frames, while **QFL** is sound (but not complete) with respect to the subclass where the converse of the accessibility relation is well-founded (i. e., with respect to the frames of the logic **GL**). For technical reasons, namely to avoid being distracted with Kripke-completeness, we define the logic $\mathbf{QFL}^{sem}$ as the set of formulas valid in all frames where the converse of the accessibility relation is well-founded; all that matters to us is that $\mathbf{QFL} \subseteq \mathbf{QFL}^{sem}$.

5.3.2 Undecidability results

We now address the question, raised in [47], of whether it suffices to use only monadic predicate letters to obtain undecidability of the two-variable fragment **QInt**(2) of **QInt**. We show that, in fact, it suffices to use a *single* monadic predicate letter to obtain undecidability of **QInt**(2). We do so by suitably adapting the technique used in [66] to (polynomially) reduce satisfiability in propositional intuitionistic logic **Int** to satisfiability in the fragment of **Int** with only two propositional variables. As the technique from [66] requires that we work with positive formulas, we first show that the *positive* monadic fragment of **QInt**(2) is undecidable. We note here that transitioning from the propositional language to the first-order one, we "strengthen" the result from [66] in the following sense: while in the propositional case, (the positive fragment of) **Int** is polynomially reducible to its two-variable subfragment, in the the first-order case, we (polynomially) reduce (the positive fragment of) **QInt**(2) to its subfragment containing a single predicate letter.[1] Working with the positive fragment of **QInt** also allows us to extend our results to the interval [**QInt**, **QKC**], as all logics in this interval share the positive fragment. Moreover, a modification of this construction allows us to obtain analogous results for logics in [**QBL**, **QFL**].

It is proven in [47] that **QInt**(2) is undecidable by reducing the following undecidable tiling problem [6] to the complement of **QInt**(2): given a finite set T of tile types that are tuples of colours $t = \langle left(t), right(t), up(t), down(t)\rangle$, decide whether T tiles the grid $\mathbb{N} \times \mathbb{N}$ in the sense that there exists a function $\tau : \mathbb{N} \times \mathbb{N} \to T$ such that, for every $i, j \in \mathbb{N}$, we have $up(\tau(i, j)) = down(\tau(i, j + 1))$ and $right(\tau(i, j)) = left(\tau(i + 1, j))$. The results in this section build on this proof.

We start off by proving that the positive fragment of **QInt**(2) containing two binary and an unlimited number of unary predicate letters, as well as two propositional variables, is undecidable. This is achieved by eliminating the constant $\bot$ from the formulas used in the proof of undecidability of **QInt**(2) from [47]. For most formulas from [47], all we do is replace $\bot$ with a propositional variable q. The resultant formulas are listed below for the reader's convenience; for ease of reference, we preserve the numbering from [47]:

$$\forall x \bigvee_{t \in T} (P_t(x) \wedge \bigwedge_{t' \neq t} (P_{t'}(x) \to q)), \tag{5.1}$$

[1] In light of [61], the reduction of **Int** to its single-variable fragment would imply that the complexity classes **P** and **PSPACE** are equivalent.

$$\bigwedge_{right(t) \neq left(t')} \forall x \, \forall y \, (H(x, y) \wedge P_t(x) \wedge P_{t'}(y) \to q), \tag{5.2}$$

$$\bigwedge_{up(t) \neq down(t')} \forall x \, \forall y \, (V(x, y) \wedge P_t(x) \wedge P_{t'}(y) \to q), \tag{5.3}$$

$$\forall x \, \exists y \, H(x, y) \wedge \forall x \, \exists y \, V(x, y), \tag{5.4}$$

$$\forall x \, \forall y \, (V(x, y) \vee (V(x, y) \to q)), \tag{5.5}$$

$$\begin{aligned} &\forall x \, \forall y \, [V(x, y) \wedge \exists x \, (D(x) \wedge H(y, x)) \to \\ &\qquad\qquad \forall y \, (H(x, y) \to \forall x \, (D(x) \to V(y, x)))]. \end{aligned} \tag{5.6}$$

Let ψ_T^+ be the conjunction of formulas (1) through (6). Then, define

$$\varphi_T^+ = \psi_T^+ \to ((\exists x \, (D(x) \to q) \to p) \to p),$$

where p is a propositional variable distinct from q.

Lemma 5.3.1 $\varphi_T^+ \notin \mathbf{QInt}(2)$ *if, and only if,* T *tiles* $\mathbb{N} \times \mathbb{N}$.

Proof. The proof is a minor modification of the proof of Theorem 1 from [47], with q essentially playing the role that "falsehood" plays in [47].

For the left to right direction, we observe that, given a model $\mathfrak{M}$ and a world w such that $\mathfrak{M}, w \not\models \varphi_T^+$, as well as an arbitrary $d \in D(w)$, there exists a world u in $\mathfrak{M}$ with wRu such that $\mathfrak{M}, u \models D[d]$ and $\mathfrak{M}, u \not\models q$. This is a straightforward consequence of the fact that $\mathfrak{M}, w \not\models (\exists x \, (D(x) \to q) \to p) \to p$. Given this, the argument from [47] applies.

For the other direction, the model falsifying φ_T^+ is different from the one used in [47] only in the evaluation of p and q. Thus, we use the same frame and interpretation of predicate letters as in [47], and additionally make q false at every world of the model and make p false at w_0 and true at every other world. ❑

Since φ_T^+ is a positive formula, this immediately gives us the following:

Corollary 5.3.2 *The positive fragment of* **QInt** *with two individual variables, two binary predicate letters, an unlimited number of unary predicate letters, and two propositional variables is undecidable.*

We now show how, drawing on an idea of Kripke's for modal logics [50], one can, in the positive fragment of **QInt**, simulate binary predicate letters using monadic predicate letters and propositional variables. As this does not increase the number of individual variables in a formula, it will allow us to eliminate binary predicate letters from the formula φ_T^+.

Lemma 5.3.3 *Let χ be a positive formula in* **QInt** *containing an occurrence of a binary predicate letter Q, and let Q_1 and Q_2 be unary predicate letters, and r and s be propositional variables, not occurring in χ. Let χ' be the result of uniformly replacing every subformula of χ of the form $Q(x,y)$ with $(Q_1(x) \wedge Q_2(y) \to r) \vee s$. Then, $\chi \in$* **QInt** *if, and only if, $\chi' \in$* **QInt**.

Proof. The left-to-right direction follows from the closure of **QInt** under substitution. For the other direction, assume that there exist $\mathfrak{M} = \langle W, R, D, I \rangle$ and $w_0 \in W$ such that $\mathfrak{M}, w_0 \not\models \chi$. We modify $\mathfrak{M}$ to obtain a model $\mathfrak{M}'$ falsifying χ' as follows. For every $w \in W$ and every $a, b \in D(w)$ such that $\mathfrak{M}, w \not\models Q[a,b]$, add to W a world $w_{a,b}$ with $wR'w_{a,b}$ and let

$$\begin{array}{lcl} \mathfrak{M}', w_{a,b} \not\models r; & & \\ \mathfrak{M}', w_{a,b} \models s; & & \\ \mathfrak{M}', w_{a,b} \models Q_1[d] & \leftrightharpoons & d = a; \\ \mathfrak{M}', w_{a,b} \models Q_2[d] & \leftrightharpoons & d = b; \end{array}$$

and let all the predicate letters different from Q_1 and Q_2 and occurring in χ' be universally true at every such world; likewise for propositional variables different from r and s. Also, let $\mathfrak{M}', w \not\models s$.

Then we can show that $\mathfrak{M}, w \models \theta[a_1, \ldots, a_m]$ if, and only if, $\mathfrak{M}', w \models \theta'[a_1, \ldots, a_m]$, for every subformula θ of χ, every $w \in W$, and every $a_1, \ldots, a_m \in D(w)$, where θ' is the result of substituting in θ every occurrence of $Q(x,y)$ with $(Q_1(x) \wedge Q_2(y) \to r) \vee s$. The proof is by induction on θ.

For the base case, first note that if $\mathfrak{M}, w \not\models Q[a,b]$, then the presence in $\mathfrak{M}'$ of the world $w_{a,b}$ guarantees that $\mathfrak{M}', w \not\models (Q_1[a] \wedge Q_2[b] \to r) \vee s$; on the other hand, if $\mathfrak{M}, w \models Q[a,b]$, then $\mathfrak{M}', w \models (Q_1[a] \wedge Q_2[b] \to r) \vee s$, as $\mathfrak{M}, u \not\models Q_1[a]$ or $\mathfrak{M}, u \not\models Q_2[b]$, for every u with $wR'u$.

The cases for $\theta = \theta_1 \vee \theta_2$, $\theta = \theta_1 \wedge \theta_2$, and $\theta = \exists x\, \theta_1$ are straightforward.

Let $\theta = \theta_1 \to \theta_2$. Assume that $\mathfrak{M}', w \not\models \theta'[a_1, \dots, a_m]$. Then, $\mathfrak{M}', u \models \theta'_1[a_1, \dots, a_m]$ and $\mathfrak{M}', u \not\models \theta'_2[a_1, \dots, a_m]$, for some $u \in W'$ with $wR'u$. If we could apply the inductive hypothesis to u, we would be done. To see that we can, notice that θ'_2 is build out of atomic formulas and the formula $(Q_1(x) \wedge Q_2(y) \to r) \vee s$, all of which are true under every assignment in every $w' \in W' - W$, using only $\wedge$, $\vee$, $\to$, $\exists$, and $\forall$. Therefore, θ'_2 is true in every $w' \in W' - W$ under every assignment; hence, $u \in W$ and the inductive hypothesis is, therefore, applicable. Thus, $\mathfrak{M}, w \not\models \theta[a_1, \dots, a_m]$. The other direction is straightforward.

The case $\theta = \forall x\, \theta_1$ is similarly argued. ❑

Now, let ξ_T^+ be the result of replacing in φ_T^+ of

$$
\begin{array}{lll}
H(x,y) & \text{with} & (H_1(x) \wedge H_2(y) \to r_1) \vee s_1; \\
V(x,y) & \text{with} & (V_1(x) \wedge V_2(y) \to r_2) \vee s_2.
\end{array}
$$

In view of Lemma 5.3.3, $\xi_T^+ \notin \mathbf{QInt}(2)$ if, and only if, T tiles $\mathbb{N} \times \mathbb{N}$. As we can replace in ξ_T^+ a propositinal variable such as q with, say, $\exists x\, Q(x)$, we immediately obtain the following:

Theorem 5.3.4 *The positive monadic fragment of* **QInt** *with two individual variables is undecidable.*

We now embed the positive monadic fragment of $\mathbf{QInt}(2)$ into its subfragment containing formulas with only one monadic predicate letter, suitably adapting the technique from [66]. As this embedding does not introduce any fresh variables, our main result in this section immediately follows.

We start by defining the frame $\mathfrak{F} = \langle W, R \rangle$. This frame, depicted in Figure 5.1, is made up of "levels" of worlds. The three top-most levels are depicted at the top of Figure 5.1: the top-most level contains worlds d_1, d_2, and d_3; level 0, worlds a_1^0, a_2^0, b_1^0, and b_2^0; level 1, worlds a_1^1, a_2^1, a_3^1, b_1^1, b_2^1, and b_3^1. The successive levels are defined inductively. Assume that level k has been defined and that it contains worlds $a_1^k, \dots, a_n^k, b_1^k, \dots, b_n^k$. For every $i, j \in \{2, \dots, n\}$, the level $k+1$ contains the world a_m^{k+1} such that $a_m^{k+1} R b_1^k$, $a_m^{k+1} R a_i^k$, and $a_m^{k+1} R b_j^k$, as well as the world b_m^{k+1} such that $b_m^{k+1} R a_1^k$, $b_m^{k+1} R a_i^k$, and $b_m^{k+1} R b_j^k$. Let $\mathfrak{M}$ be a model with constant domains, say Z, based on $\mathfrak{F}$ (without a loss of generality, we can assume that Z contains at least three individuals) and let $a \in Z$. We say that $\mathfrak{M}$ is a-suitable if, for some $b \in Z$

such that $b \neq a$, the following hold: $I(d_2, P) = \{\langle c \rangle : c \in Z \text{ and } c \neq a\}$; $I(d_3, P) = \{\langle a \rangle, \langle b \rangle\}$; $I(b_0^1, P) = \{\langle b \rangle\}$; $I(w, P) = \varnothing$, for $w \notin \{d_2, d_3, b_1^0\}$.

We now define formulas, of one free variable x, that correspond to the worlds of an a-suitable model in the sense that each formula fails at a world w of the model, with a assigned to x, exactly when w can see the world corresponding to the formula. For these formulas, we use notation that makes clear which worlds they correspond to; thus, formula D_i corresponds to world d_i, A_i^k to a_i^k, and B_i^k to b_i^k. First, we define formulas for the three top-most levels:

$$\begin{array}{lcl}
D_1 & = & \exists x\, P(x), \\
D_2(x) & = & \exists x\, P(x) \to P(x), \\
D_3(x) & = & P(x) \to \forall x\, P(x), \\
A_1^0(x) & = & D_2(x) \to D_1 \vee D_3(x), \\
A_2^0(x) & = & D_3(x) \to D_1 \vee D_2(x), \\
B_1^0(x) & = & D_1 \to D_2(x) \vee D_3(x), \\
B_2^0(x) & = & A_1^0(x) \wedge A_2^0(x) \wedge B_1^0(x) \to D_1 \vee D_2(x) \vee D_3(x), \\
A_1^1(x) & = & A_1^0(x) \wedge A_2^0(x) \to B_1^0(x) \vee B_2^0(x), \\
A_2^1(x) & = & A_1^0(x) \wedge B_1^0(x) \to A_2^0(x) \vee B_2^0(x), \\
A_3^1(x) & = & A_1^0(x) \wedge B_2^0(x) \to A_2^0(x) \vee B_1^0(x), \\
B_1^1(x) & = & A_2^0(x) \wedge B_1^0(x) \to A_1^0(x) \vee B_2^0(x), \\
B_2^1(x) & = & A_2^0(x) \wedge B_2^0(x) \to A_1^0(x) \vee B_1^0(x), \\
B_3^1(x) & = & B_1^0(x) \wedge B_2^0(x) \to A_1^0(x) \vee A_2^0(x).
\end{array}$$

Now, assume that the formulas for level k have been defined and define

$$\begin{array}{lcl}
A_m^{k+1}(x) & = & A_1^k(x) \to B_1^k(x) \vee A_i^k(x) \vee B_j^k(x), \\
B_m^{k+1}(x) & = & B_1^k(x) \to A_1^k(x) \vee A_i^k(x) \vee B_j^k(x),
\end{array}$$

where m is uniquely determined for every pair $i, j \in \{2, \ldots, n_k\}$, where n_k is the maximal index for formulas of level k.

Lemma 5.3.5 *Let $\mathfrak{M} = \langle W, R, D, I \rangle$ be an a-suitable model and let $w \in W$. Then,*

$$\mathfrak{M}, w \not\models A_m^k[a] \iff wRa_m^k, \quad \text{and} \quad \mathfrak{M}, w \not\models B_m^k[a] \iff wRb_m^k.$$

Proof. Induction on k. ❑

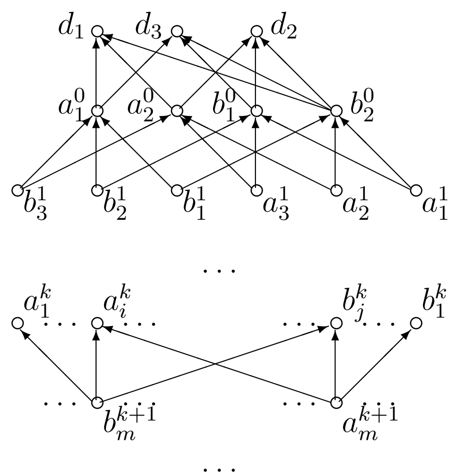

Figure 5.1: Frame $\mathfrak{F}$

Now, let φ be a positive formula containing monadic predicate letters $P_1, \ldots, P_n$ (we may assume $n \geqslant 2$). For each $i \in \{1, \ldots, n\}$, define

$$\alpha_i(x) = A_i^n(x) \vee B_i^n(x).$$

Finally, let φ^* be the result of substituting, for every $i \in \{1, \ldots, n\}$, of $\alpha_i(x)$ for $P_i(x)$ into φ.

Lemma 5.3.6 $\varphi \in$ **QInt** *if, and only if,* $\varphi^* \in$ **QInt**.

Proof. The right-to-left direction follows from the closure of **QInt** under predicate substitution. For the other direction, assume that $\mathfrak{M}_\varphi, w_0 \not\models \varphi$ for some $\mathfrak{M}_\varphi = \langle W_\varphi, R_\varphi, D_\varphi, I_\varphi \rangle$ and $w_0 \in W_\varphi$. (We may assume without a loss of generality that the domain of $\mathfrak{M}_\varphi$ contains at least three individuals; we use this fact in the construction of $\mathfrak{M}^*$ below.) We need to construct a model $\mathfrak{M}^*$ falsifying φ^* at some world.

First, for every $w \in W_\varphi$ and $a \in D_\varphi(w)$, consider an a-suitable model $\mathfrak{M}_a^w = \langle W_a^w, R_a^w, D_a^w, I_a^w \rangle$, based on a copy of the frame $\mathfrak{F}$ defined above, where $D_a^w(u) = D_\varphi(w)$, for every $u \in W_a^w$. To obtain the frame $\mathfrak{F}^*$, first, append to $\mathfrak{F}_\varphi = \langle W_\varphi, R_\varphi \rangle$, for every $w \in W_\varphi$ and $a \in D_\varphi(w)$, frames of all such models $\mathfrak{M}_a^w$; in addition, let $wR^*a_i^n$ and $wR^*b_i^n$, for a_i^n and b_i^n belonging to

$\mathfrak{M}_a^w$, exactly when $\mathfrak{M}_\varphi, w \not\models P_i[a]$, for $i \in \{1, \dots, n\}$. Define D^* to agree with D_φ on W_φ and to agree with D_a^w on W_a^w, for every $w \in W_\varphi$ and $a \in D_\varphi(w)$. To finish off the definition of the model $\mathfrak{M}^* = \langle W^*, R^*, D^*, I^* \rangle$, define $I^*(u, P)$ to agree with $I_a^w(u, P)$ at the worlds in the appended models and to be $\varnothing$ at the worlds from W_φ. We can now show that $\mathfrak{M}_\varphi, w \models \psi[a_1, \dots, a_m]$, if and only if, $\mathfrak{M}^*, w \models \psi^*[a_1, \dots, a_m]$, for every $w \in W_\varphi$ and every subformula ψ of φ.

The proof proceeds by induction on ψ. The only case we explicitly consider here is $\psi = \psi_1 \to \psi_2$, leaving the rest to the reader. Assume $\mathfrak{M}^*, w \not\models \psi^*[a_1, \dots, a_m]$. Then, $\mathfrak{M}^*, u \models \psi_1^*[a_1, \dots, a_m]$ and $\mathfrak{M}^*, u \not\models \psi_2^*[a_1, \dots, a_m]$, for some $u \in W^*$ with wR^*u. If we could apply the inductive hypothesis to u, we would be done. To see that we can, notice that ψ_2^* is build out of formulas of the form $A_i^n(x) \vee B_i^n(x)$ using only $\wedge$, $\vee$, $\to$, $\exists$, and $\forall$. As, in view of Lemma 5.3.5, formulas $A_i^n(x) \vee B_i^n(x)$ are true at every world of $\mathfrak{M}^*$ that lies outside of W_φ and is accessible from W_φ, we conclude that $u \in W_\varphi$, and the inductive hypothesis is, therefore, applicable. Thus, $\mathfrak{M}_\varphi, w \not\models \psi[a_1, \dots, a_m]$. The other direction is straightforward.

We conclude that $\mathfrak{M}^*, w_0 \not\models \varphi^*$ and, thus, $\varphi^* \notin$ **QInt**. ❑

As the construction of φ^* from φ did not introduce any fresh individual variables, we have the following:

Theorem 5.3.7 *The positive fragment of* **QInt** *with two individual variables and a single predicate letter is undecidable.*

We now extend the argument presented above to the logics in the intervals [**QBL**, **QKC**] and [**QBL**, **QFL**].

First, to establish the undecidability of the two-variable fragments of logics whose semantics might contain irreflexive worlds, we need to slightly modify formulas (1) through (6) listed above. Therefore, we define ψ_T^* to be the conjunction of ψ_T^+ and following formula:

$$\forall x \, \forall y \, (H(x, y) \vee (H(x, y) \to q)), \tag{5a}$$

and define

$$\varphi_T^* = \psi_T^* \to [(\exists x \, (D(x) \to \Box^5 q) \to p) \to \Box p].$$

This enables us to prove, using the tiling problem described above, that T tiles $\mathbb{N} \times \mathbb{N}$ if and only if $\varphi_T^* \notin L(2)$, where $L \in \{\mathbf{QBL}, \mathbf{QFL}^{sem}\}$. We leave the details of the proof to the reader. As the construction of φ^*_T is uniform for both logics, it follows that the claim holds for every $L \in [\mathbf{QBL}, \mathbf{QFL}^{sem}]$. Notice that the same proof also works for logics in $[\mathbf{QBL}, \mathbf{QKC}]$. We simulate binary predicate letters by monadic ones as for **QInt**. We now show how to simulate all monadic predicate letters with a single one.

For the interval $[\mathbf{QBL}, \mathbf{QKC}]$, notice that if we add to the model $\mathfrak{M}^*$ built in the proof of Lemma 5.3.6 a world d accessible from every element of W^* and such that $I^*(d, P) = D(d)$, the resultant model is a model of every logic in the interval $[\mathbf{QBL}, \mathbf{QKC}]$. Thus, we have the following:

Theorem 5.3.8 *Let L be a logic in the interval $[\mathbf{QBL}, \mathbf{QKC}]$. Then, the positive fragment of L with two individual variables and a single predicate letter is undecidable.*

We next consider the interval $[\mathbf{QBL}, \mathbf{QFL}^{sem}]$. In this case, we need to make a more substantial modification to the frame $\mathfrak{F}$, as the semantics of $\mathbf{QFL}^{sem}$ prohibits the existence of reflexive worlds. We then proceed as follows. First, add to W worlds $\bar{d}_2$ and $\bar{d}_3$ with $d_2 R \bar{d}_2$ and $d_3 R \bar{d}_3$. Second, for every $k \geqslant 0$, do the following: for every world a^k_i, add to W the world $\bar{a}^k_i$ and, for every world b^k_i, add to W the world $\bar{b}^k_i$; also, let $a^k_i R \bar{a}^k_i$ and $b^k_i R \bar{b}^k_i$, for every k and i. Lastly, whenever in $\mathfrak{F}$ we had $a^{k+1}_i R a^k_j$ or $a^{k+1}_i R b^k_j$, let $\bar{a}^{k+1}_i R a^k_j$ and $\bar{a}^{k+1}_i R b^k_j$; also, whenever we had $b^{k+1}_i R a^k_j$ or $b^{k+1}_i R b^k_j$, let $\bar{b}^{k+1}_i R a^k_j$ and $\bar{b}^{k+1}_i R b^k_j$. We then define a-suitable models so that $I(\bar{d}_2, P) = I(d_2, P)$, $I(\bar{d}_3, P) = I(d_3, P)$, and for every k and i, $I(\bar{a}^i_k, P) = I(a^i_k, P)$ and $I(\bar{b}^i_k, P) = I(b^i_k, P)$. In essence, we created "doubles" for the worlds d_2, d_3, a^i_k, and b^i_k, which serve to evaluate formulas whose main connective is $\to$ or $\forall$ at the worlds whose doubles they are. Then, a-suitable models satisfy the condition in the statement of Lemma 5.3.5, and the model $\mathfrak{M}^*$ built in the proof of Lemma 5.3.6 becomes a model of every logic in $[\mathbf{QBL}, \mathbf{QFL}^{sem}]$. As $\mathbf{QFL} \subseteq \mathbf{QFL}^{sem}$, we have the following:

Theorem 5.3.9 *Let L be a logic in the interval $[\mathbf{QBL}, \mathbf{QFL}]$. Then, the positive fragment of L with two individual variables and a single predicate letter is undecidable.*

Remark 5.3.10 *Note that the results of this section hold true if we only consider frames with constant domains.*

5.4 Discussion

As already noticed, the results presented in the present chapter concerning sublogics of **QGL** and **QGrz** are quite tight: as shown in [85], for all "natural" sublogics of **QGL** and **QGrz**—including **QK**, **QT**, **QD**, **QK4**, **QS4**, **QGL**, and **QGrz**—adding to the restriction to two individual variables and a single monadic predicate letter considered in section 5.2 a minor restriction that the modal operators only apply to formulas with at most one free variable, results in decidable fragments of those logics. It is not difficult to notice that the results analogous to those obtained in section 5.2 can be obtained for quasi-normal logics such as **QS** (Solovay's logic) and Lewis's **QS1**, **QS2**, and **QS3** [23].

A notable exception in our consideration of modal logics is **QS5**, whose two-variable monadic fragment was shown to be undecidable in [47]. While it is not difficult to extend our results to the multimodal version of **QS5**—we need to modify the construction used for sublogics **QKTB** by substituting a succession of two steps along distinct accessibility relations for a single step along a single aceessibility relation in the frames of a-suitable models—nor is it difficult to show, by encoding the tiling problem used in [47], that the two-variable fragment of **QS5** with two monadic predicate letters and infinitely many propositional symbols is undecidable, the case of **QS5** remains elusive. We conjecture that the fragment of **QS5** with two variables and a single monadic predicate letter is decidable.

On the other hand, it is relatively straightforward to show that the two-variable fragment of **QS5** with a single monadic predicate letter and an infinite supply of individual variables is undecidable. Indeed, let **SIB** be the first-order theory of a symmetric irreflexive binary relation S; it is well-known that **SIB** is undecidable [60, 49]. We can then simulate $S(x, y)$ as $\Box(\neg P(x) \vee \neg P(y))$ and show that, if a quantified modal logic L is valid on a frame containing a world that can see infinitely many worlds, then L is undecidable in the language with a single monadic predicate letter (and infinitely many individual variables). This observation covers all modal logics considered in [47], but not covered by the results of section 5.2, including **QS5**, **QGL.3**, and **QGrz.3**.

By contrast, we can say nothing about superintuitionistic logics not included in the interval $[\mathbf{QInt}, \mathbf{QKC}]$, as our proof relies on the fact that we are working with the positive fragment of those logics. It is not essential to our proof that formulas $A_i^k(x)$ and $B_i^k(x)$ be positive; however, by discarding their positivity we would weaken, rather than strengthen, our results.

Chapter 6

Conclusion

We have presented a number of results based on techniques for embedding propositional modal logics into their fragments containing only a small number of propositional variables. We have also presented similar results for predicate modal and intuitionistic logics, based on techniques for embedding such logics into their fragments with a small number of predicate letters and individual variables.

Our techniques seem to be widely applicable, and the main direction of future research is exploring how they can be applied to the study of computational complexity of logical formalisms similar to, but also sufficiently different from, those studied in this thesis. These include non-normal modal logics, multidimensional modal logics [25], and description logics [3].

Bibliography

[1] Natasha Alechina, Stéphane Demri, and Maarten de Rijke. A modal perspective on path constraints. *Journal of Logic and Computaion*, 13(6):939–956, 2003.

[2] Rajeev Alur, Thomas A. Henzinger, and Orna Kuperman. Alternating-Time temporal logic. *Journal of ACM*, 49(5):672–713, 2002.

[3] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *Description Logic Handbook*. Cambridge University Press, 2nd edition, 2010.

[4] Philippe Balbiani and Tinko Tinchev. Definability and computability for PRSPDL. In *Advances in Modal Logic*, volume 10, pages 16–33, 2014.

[5] H. Behmann. Beiträige zür algebra der logik, inbesondere zum entscheidungsproblem. *Mathematische Annalen*, 86:163–229, 1922.

[6] Robert Berger. *The Undecidability of the Domino Problem*, volume 66 of *Memoirs of AMS*. AMS, 1966.

[7] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. Springer, 2001.

[8] Patrick Blackburn and Edith Spaan. A modal perspective on the computational complexity of attribute value grammar. *Journal of Logic, Language, and Information*, 2:129–169, 1993.

[9] Egon Börger, Erich Grädel, and Yuri Gurevich. *The Classical Decision Problem*. Springer, 1997.

[10] Alexander Chagrov and Mikhail Rybakov. How many variables does one need to prove PSPACE-hardness of modal logics? In *Advances in Modal Logic*, volume 4, pages 71–82, 2003.

[11] Alexander Chagrov and Michael Zakharyaschev. *Modal Logic*. Oxford University Press, 1997.

[12] Alonzo Church. A note on the "Entscheidungsproblem". *The Journal of Symbolic Logic*, 1:40–41, 1936.

[13] Edmund M. Clarke and E. Allen Emerson. Design and synthesis of synchronization skeletons using Branching Time Temporal Logic. In *Proceedings of Workshop on Logics of Programs*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71. Springer, 1981.

[14] Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. *Model Checking*. MIT Press, 2000.

[15] Am'elie David. Deciding ATL* by tableaux. In Amy P. Felty and Aart Middeldorp, editors, *Automated Deduction–CADE-25*, volume 9195 of *Lecture Notes in Computer Science*, pages 214–228, 2015.

[16] Farinas L. del Cerro and Eva Orlowska. DAL—a logic for data analysis. *Theoretical Computer Science*, 36:251–264, 1985.

[17] Stéphane Demri, Valentin Goranko, and Martin Lange. *Temporal Logics in Computer Science*. Cambridge University Press, 2016.

[18] Stéphane Demri and Philippe Schnoebelen. The complexity of propositional linear temporal logics in simple cases. *Information and Computation*, 174:84–103, 2002.

[19] Francesco M Donini, Maurizio Lenzerini, Daniele Nardi, and Werner Nutt. The complexity of concept languages. *Information and Computation*, 134(1):1–58, 1997.

[20] E. Allen Emerson and Joseph Halpern. Decision procedures and expressiveness in temporal logic of branching time. *Journal of Computer and System Sciences*, 30(1):1–24, 1985.

[21] E. Allen Emerson and Joseph Y. Halpern. "Sometimes and Not Never" revisited: on Branching versus Linear Time Temporal Logic. *Journal of ACM*, 33(1):151–178, 1986.

[22] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning About Knowledge*. MIT Press, 1995.

[23] Robert Feys. *Modal Logics*. E. Nauwelaerts, 1965.

[24] Michael J. Fischer and Richard E. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18:194–211, 1979.

[25] Dov Gabbay, Agi Kurucz, Frank Wolter, and Michael Zakharyaschev. *Many-dimensional Modal Logics: Theory and Applications*. Elsevier, 2003.

[26] Dov V. Gabbay. *Semantical Investigations in Heyting's Intuitionistic Logic*. D. Reidel, 1981.

[27] James Garson. Modal logic. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, 2016. Available at `https://plato.stanford.edu/`.

[28] Giuseppe De Giacomo and Maurizio Lenzerini. Boosting the correspondence between description logics and propositional dynamic logics. In *Proceeding of the Twelfth National Conference on Artificial Intelligence (AAI'94)*, volume 1, pages 205–212. AAI Press, 1994.

[29] Kurt Gödel. Zum Entsheidungsproblem des logischen Funktionenkalküls. *Monatschefte für Mathematische Physika*, 40:433–443, 1933.

[30] Robert Goldblatt. Parallel action: concurrent dynamic logic with independent modalities. *Studia Logica*, 51:551–578, 1992.

[31] Valentin Goranko and Solomon Passy. Using the universal modality: Gains and questions. *Journal of Logic and Computation*, 2:5–30, 1989.

[32] Valentin Goranko and Dmitry Shkatov. Tableau-based decision procedure for multi-agent epistemic logic with operators of commond and distributed knowledge. In A. Cerone and S. Gruner, editors, *Proceedings*

of the 6th IEEE International Conference on Software Engineering and Formal Methods (SEFM2008), pages 237–246. IEEE Computer Society Press, 2008.

[33] Valentin Goranko and Dmitry Shkatov. Tableau-based decision procedure for full coalitional multiagent temporal-epistemic logic of linear time. In C. Sierra, C. Castelfranchi, K. S. Decker, and J.Sichman, editors, *Proceedings of 8th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 09)*, pages 969–976, 2009.

[34] Valentin Goranko and Dmitry Shkatov. Tableau-based decision procedure for the full coalitional multiagent temporal-epistemic logic of branching time. In *Proceedings of Formal Approaches to Multi-Agent Systems 2009*, 2009.

[35] Valentin Goranko and Dmitry Shkatov. Tableau-based decision procedures for logics of strategic ability in multiagent systems. *ACM Transactions on Computational Logic*, 11(1):3–51, 2009.

[36] Valentin Goranko and Dmitry Shkatov. Tableu-based procedure for deciding satisfiability in the full coalitional multiagent epistemic logic. In S. Artemov, editor, *Logical Foundations of Computer Science 2009*, volume 5407 of *Lecture Notes in Computer Science*, pages 197–213. Springer, 2009.

[37] Valentin Goranko and Govert van Drimmelen. Complete axiomatization and decidability of the alternating-time temporal logic. *Theoretical Computer Science*, 353(1–3):93–117, 2006.

[38] Rajeev Goré. Tableau methods for modal and temporal logics. In Marcello D'Agostino, Dov M. Gabbay, Reiner Hähnle, and Joachim Posegga, editors, *Handbook of Tableau Methods*, pages 297–396. Kluwer, 1999.

[39] Erich Grädel, Phokion G. Kolaitis, and Moshe Y. Vardi. On the decision problem for two-variable first-order logic. *Bulletin of Symbolic Logic*, 3(1):53–69, 1997.

[40] Joseph Y. Halpern. The effect of bounding the number of primitive propositions and the depth of nesting on the complexity of modal logic. *Aftificial Intelligence*, 75(2):361–372, 1995.

[41] Joseph Y. Halpern and Leandro Chaves Rêgo. Characterizing the NP-PSPACE gap in the satisfiability problem for modal logic. *Journal of Logic and Computation*, 17(4):795—-806, 2007.

[42] Joseph Y. Halpern and Moshe Y. Vardi. The complexity of reasoning about knowledge and time I: Lower bounds. *Journal of Computer and System Sciences*, 38(1):195–237, 1989.

[43] David Harel and R. Sherman. Looping versus repeating in dynamic logic. *Information and Control*, 55:175–192, 1982.

[44] Edith Hemaspaandra. The complexity of poor man's logic. *Journal of Logic and Computation*, 11(4):609–622, 2001.

[45] George E. Hughes. Every world can see a reflexive world. *Studia Logica*, 49(2):175–181, 1990.

[46] Michael Huth and Mark Ryan. *Logic in Computer Science: Modelling and Reasoning about Systems*. Cambridge University Press, 2nd edition, 2004.

[47] Roman Kontchakov, Agi Kurucz, and Michael Zakharyaschev. Undecidability of first-order intuitionistic and modal logics with two variables. *Bulletin of Symbolic Logic*, 11(3):428–438, 2005.

[48] Marcus Kracht. Syntactic codes and grammar refinement. *Journal of Logic, Language, and Information*, 4:41–60, 1995.

[49] Philip Kremer. On the complexity of propositional quantification in intuitionistic logic. *The Journal of Symbolic Logic*, 62(2):529–544, 1997.

[50] Saul Kripke. The undecidability of monadic modal quantification theory. *Zeitschrift für Matematische Logik und Grundlagen der Mathematik*, 8:113–116, 1962.

[51] Richard E. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal on Computing*, 6(3):467–480, 1977.

[52] Martin Lange and Carsten Lutz. 2-EXPTIME lower bound for propositional dynamic logic with intersection. *The Journal of Symbolic Logic*, 70(4):1072–1086, 2005.

[53] Carsten Lutz. Complexity and succinctness of public announcement logic. In H. Nakashima, M. P. Wellman, G. Weiss, and P. Stone, editors, *Proceedings of 5th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 137–143, 2006.

[54] Sergei Maslov, Gregory Mints, and Vladimir Orevkov. Unsolvability in the constructive predicate calculus of certain classes of formulas containing only monadic predicate variables. *Soviet Mathematics Doklady*, 6:918–920, 1965.

[55] Fabio Massacci. Decision procedures for expressive description logics with intersection, composition, converse of roles and role identity. In B. Nebel, editor, *Proceedings of the seventeenth International Conference on Artificial Iintelligence*, pages 193–198. Morgan Kaufmann, 2001.

[56] Arne Meier, Martin Mundhenk, MichaelThomas, and Heribert Vollmer. The complexity of satisfiability for fragments of ctl and ctl*. *Electronic Notes in Theoretical Computer Science*, 223:201–213, 2008.

[57] Franco Montagna. The predicate modal logic of provability. *Notre Dame Journal of Formal Logic*, 25(2):179–189, 1984.

[58] Michael Mortimer. On languages with two variables. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, pages 135–140, 1975.

[59] Michael C. Nagle and S. K. Thomason. The extensions of the modal logic K5. *The Journal of Symbolic Logic*, 50(1):102–109, 1975.

[60] Anil Nerode and Richard A. Shore. Second order logic and first order theories of reducibility ordering. In J. Barwise, H. J. Keisler, and K. Kunen, editors, *The Kleene Symposium*, pages 181–200. North-Nolland, 1980.

[61] Iwao Nishimura. On formulas of one variable in intuitionistic propositional calculus. *The Journal of Symbolic Logic*, 25(4):327–331, 1960.

[62] David Peleg. Concurrent dynamic logic. *Journal of the ACM*, 34(2):450–479, 1987.

[63] Jan A. Plaza. Logics of public communications. In M. L. Emrich, M. S. Pfeifer, M. Hadzikadic, and Z. W. Ras, editors, *Proceedings of the fourth international symposium on methodologies for intelligent systems: Poster session program*, pages 201–216, 1989.

[64] Amir Pnueli. The temporal logic of programs. In *Proceedings of the 18th IEEE Symposium on Foundations of Computer Science*, pages 46–67, 1977.

[65] Mark Reynolds. A tableau for CTL*. In Ana Cavalcanti and Dennis Dams, editors, *FM 2009: Formal Methods*, Lecture Notes in Computer Science, pages 403–418. Springer, 2009.

[66] Mikhail Rybakov. Complexity of intuitionistic propositional logic and its fragments. *Journal of Applied Non-Classical Logics*, 18(2–3):267–292, 2008.

[67] Mikhail Rybakov and Dmitry Shkatov. Undecidability of first-order modal and intuitionistic logics with two variables and one monadic predicate letter. To appear in *Studia Logica*.

[68] Mikhail Rybakov and Dmitry Shkatov. Complexity and expressivity of propositional dynamic logics with finitely many variables. *Logic Journal of the IGPL*, 26(5):539–547, 2018.

[69] Mikhail Rybakov and Dmitry Shkatov. Complexity of finite-variable fragments of propositional modal logics of symmetric frames. *Logic Journal of the IGPL*, 27(1):60–68, 2019.

[70] Михаил Рыбаков. Перечислимость модальных предикатных логик и условия обрыва возрастающих цепей. *Логические исследования*, 8:155–167, 2001. (Mikhail Rybakov. Enumerability of predicate modal logics and conditions of non-existence of infinite ascending chains. *Logicheskiye Issledovaniya*, 8:155–167, 2001. In Russian.).

[71] Sven Schewe. ATL* satisfiability is 2EXPTIME-complete. In *Automata, Languages and Programming. ICALP 2008*, volume 5126 of *Lecture Notes in Computer Science*, pages 373–385, 2008.

[72] Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2008.

[73] A. Prasad Sistla and Edmund M. Clarke. The complexity of propositional linear temporal logics. *Journal of ACM*, 32(3):733–749, 1985.

[74] Larry J. Stockmeyer and Albert R. Meyer. Word problems requiring exponential time: Preliminary report. In *Proceedings of the 5th Annual ACM Symposium on Theory of Computing*, pages 1–9, 1973.

[75] Robert S. Streett. Propositional dynamic logic of looping and converse is elementarily decidable. *Information and Control*, 54, 1982.

[76] J. Surányi. Zur Reduktion des Entscheidungsproblems des logischen Funktioskalküls. *Mathematikai és Fizikai Lapok*, 50:51–74, 1943.

[77] Wiebe van der Hoek and Michael Wooldridge. Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications. *Studia Logica*, 75(1):125–157, 2003.

[78] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. *Dynamic Epistemic Logic*. Springer, 2008.

[79] Moshe Y. Vardi. The taming of converse: Reasoning about two-way computations. In R. Parikh, editor, *Proceedings of the Conference on Logic of Programs*, volume 193 of *Lecture Notes in Computer Science*, pages 413–424, 1985.

[80] Moshe Y. Vardi and Larry Stockmeyer. Improved upper and lower bounds for modal logics of programs. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, pages 240–251, 1985.

[81] Albert Visser. A propositional logic with explicit fixed points. *Studia Logica*, 40:155–175, 1981.

[82] Vítěslav Švejdar. The decision problem of provability logic with only one atom. *Archive for Mathematical Logic*, 42(8):763–768, 2003.

[83] Dirk Walther. ATEL with common and distributed knowledge is ExpTime-complete. In *Methods for Modalities 4*, pages 173–186, 2005.

[84] Dirk Walther, Carsten Lutz, Frank Wolter, and Michael Wooldridge. ATL satisfiability is indeed ExpTime-complete. *Journal of Logic and Computation*, 16(6):765–787, 2006.

[85] Frank Wolter and Michael Zakharyaschev. Decidable fragments of first-order modal logics. *The Journal of Symbolic Logic*, 66:1415–1438, 2001.