COMPLEXITY
OF
MODAL LOGICS

Promotor: Prof. J.F.A.K. van Benthem
Faculteit Wiskunde en Informatica
Universiteit van Amsterdam
Plantage Muidergracht 24
1018 TV Amsterdam

Druk: Haveka B.V., Alblasserdam

COMPLEXITY
OF
MODAL LOGICS

ACADEMISCH PROEFSCHRIFT

Ter verkrijging van de graad van Doctor aan de
Universiteit van Amsterdam,
op Gezag van de Rector Magnificus
Prof. Dr. P.W.M. de Meijer
in het openbaar te verdedigen in de
Aula der Universiteit
(Oude Lutherse Kerk, ingang Singel 411, hoek Spui)
op dinsdag 23 maart 1993 te 11.30 uur.

door
Edith Spaan
geboren te Schagen

Contents

Chapter 1

Introduction

The actual "use" of modal logics in fields like distributed systems, computational linguistics, and program verification has given rise to new questions in the field of modal logic itself. For instance, though a logician might be satisfied by knowing that a logic is decidable, a typical "user" might want more precise information, e.g., *how* decidable that logic is, or, in other words, what the complexity of the provability problem for the logic is. The literature contains many results about the complexity of modal logics. However, all these results are on the complexity of specific, "fixed" logics, and, for each new logic, the complexity has to be discovered and proven anew. In this thesis, we develop a theory of the *sources* of complexity in modal logics—by identifying specific features that, when possessed by a logic, ensure or preclude a certain level of complexity for the logic's provability problem.

In "traditional" modal logic, there are a number of results showing decidability, completeness, and the finite model property for large classes of logics (see for instance [Fin85, Zak92]). Perhaps the easiest example of this phenomenon is Bull's theorem: all the (uncountably many) $S4.3$ extensions have the finite model property (and are decidable). In the next chapter we prove the following complexity analog of Bull's theorem: the satisfiability problems for all $S4.3$ extensions are NP-complete. This theorem is proven by a straightforward adaptation of Fine's proof of Bull's theorem [Fin72]. Its importance lies in the fact that it shows that general theorems on the complexity of modal logics exist. Looking at constructions used to prove decidability or the finite model property often leads to non-trivial upper bounds on the complexity of the satisfiability problems. These upper bound results, however, are restricted to uni-modal logics, though "useful" modal logics are usually multi-modal. Unfortunately, this situation is much more complex than the uni-modal case, and only recently has some progress been made towards proving general theorems for the multi-modal case. These theorems—called transfer theorems in [FS]—are of the following form. Suppose we have a collection of logics and we construct a new logic from these logics; we say that a property *transfers* if the resulting logic has this property whenever all logics in the collection have this property.

The simplest instance of this phenomenon is the following. Given two uni-modal logics with different modal operators, consider the smallest (bi-modal) logic that contains both uni-modal logics. Such a construction is called the (independent) join, also known as bimodal sum or fusion. Joins inherit many properties from their uni-modal fragments: decidability, the finite model property, and (strong) completeness all transfer [FS, KW91]. However, it is in general not the case that upper bounds transfer. A counterexample (under the assumption that NP $\neq$ PSPACE) is given by the join of two $S5$ logics, since $S5$ satisfiability is NP-complete [Lad77], though $S5 \oplus S5$-satisfiability is PSPACE-complete [HM85]. On the other hand, the join does not always increase the complexity, as the satisfiability problems for K and $K \oplus K$ are both PSPACE-complete ([Lad77, HM85]). In chapter 3, we show that under reasonable conditions upper bounds containing PSPACE transfer. In addition, we completely characterize what happens with the join of two sub-PSPACE logics.

Multiprocessor systems are often modeled in the literature by the independent join. However, if we want to make global statements about the system, we need more expressive power. One of the simplest ways to do this is by enriching the language with the universal modality $[u]$, with the semantics: $[u]\phi$ is true iff ϕ is true in every world of the model. Another auxiliary modality that occurs in various guises in the literature is the reflexive transitive closure. This modality occurs, for instance, in temporal logic, as the "always" operator is the reflexive transitive closure of the "nexttime" operator, and in logics of knowledge, where "common knowledge" is defined as the reflexive transitive closure of the $S5$ logics that model the processors.

Chapter 4 is devoted to the complexity of these two extensions. We show that, in contrast to the join, decidability does not transfer to the enriched versions, even if we add a number of extra restrictions. In fact, the complexity of the satisfiability problem can jump from NP-complete to highly undecidable. Fortunately, this is not always the case: Halpern and Vardi's multiprocessor system with common knowledge [HV89] is (only) EXPTIME-complete. In chapter 4, we show that it is impossible to do better than that: except in some very trivial cases, adding the universal modality or the reflexive transitive closure always forces EXPTIME-hardness.

It is clear that much work remains. The last two chapters give some idea of the tremendous task that lies ahead: we will look at the complexity of various complex logics that occur in computational linguistics (chapter 5) and distributed systems (chapter 6).

Computational Linguistics Attribute Value Structures are probably the most widely used means of representing linguistic structure in computational linguistics, and the process of unifying Attribute Value descriptions lies at the heart of many parsers. As a number of researchers have recently observed, the most common formalisms for describing AVSs are variants of languages of propositional modal logic. Furthermore, testing whether two Attribute Value descriptions unify amounts to testing for modal satisfiability. In chapter 5, which is based on joint work with Patrick Blackburn [BS], we put this observation to work. We study the complexity of the satisfiability problem for nine modal languages which mirror different aspects of AVS description formalisms, including the ability to express re-entrancy, the ability to express generalizations, and the ability to express recursive constraints.

Distributed Systems Recent work has shown that modal logics are extremely useful in formalizing the design and analysis of distributed protocols. Halpern and Vardi [HV89] unify current formalisms for reasoning about knowledge and time, and prove the complexity for all cases corresponding to different choices of knowledge operators and different assumptions made about the distributed system. In the cases of most interest to distributed systems, the validity problems for the logics modeling these systems are highly undecidable—-in fact, Π^1_1-complete. Since this is a situation we want to avoid, it is essential to determine what causes this complexity. The Π^1_1-hardness proofs of Halpern and Vardi [HV89] rely heavily upon the presence of various temporal operators, and in that paper it is conjectured that reducing the temporal expressive power decreases the complexity of the corresponding validity problem. However, in chapter 6, based on [Spa90], we show that this is *not* the case—even restricting the temporal operators to just the "always" operator is not enough to defy Π^1_1-completeness.

These results provide evidence that undecidability is caused not by having a fancy set of operators, but rather by the ability to express generalizations in combination with certain assumptions of the models. For example, in the logics for distributed systems mentioned above, undecidability is caused by assuming that processors never forget.

Chapter 2

Modal Logic and Complexity

2.1 A Bit of Modal Logic

Syntax The language $\mathcal{L}(I, \mathcal{P})$ is a language of propositional modal logic with an I indexed set of modal operators, and a set $\mathcal{P}$ of propositional variables. Unless explicitly stated otherwise, we assume that $\mathcal{P}$ is infinite. When I and/or $\mathcal{P}$ are clear from context, we write $\mathcal{L}(\mathcal{P})$, $\mathcal{L}(I)$ or $\mathcal{L}$. The set of $\mathcal{L}$ formulas is inductively defined as follows:

- p is an $\mathcal{L}$ formula for every $p \in \mathcal{P}$,
- if ϕ and ψ are $\mathcal{L}$ formulas, then so are $\neg\phi$ and $\phi \wedge \psi$,
- if ϕ is an $\mathcal{L}$ formula, and $a \in I$, then $[a]\phi$ is an $\mathcal{L}$ formula.

We define the other Boolean connectives $\vee, \rightarrow, \leftrightarrow, \top$ and $\perp$ in the usual way. In addition, we define $\langle a\rangle\phi := \neg[a]\neg\phi$ for each $a \in I$. (If $|I| = 1$, we sometimes use $\Box$ and $\Diamond$.) The choice for basic operators is always arbitrary, and no choice is best under all circumstances. In particular, in inductive proofs on models, it's often more convenient to view $\langle a\rangle$ as the basic modal operator instead of $[a]$.

For the remainder of this section, we assume that $\mathcal{L}(I, \mathcal{P})$ is fixed. For ϕ a formula, we define the *size* of ϕ as the length of ϕ as a string over alphabet $I \cup \mathcal{P} \cup \{(,), \wedge, \neg, [,]\}$. The *modal depth* of ϕ is the nesting depth of modal operators, and the *closure* of ϕ, denoted by $Cl(\phi)$, is the least set of formulas containing ϕ, and closed under subformulas and single negations, i.e. if $\psi \in Cl(\phi)$ and ψ is not of the form $\neg\xi$, then $\neg\psi \in Cl(\phi)$. Since the number of subformulas of ϕ is at most the size of ϕ (for every connective and proposition letter in ϕ corresponds to a subformula of ϕ and vice versa), the size of $Cl(\phi)$ is at most twice the size of ϕ. To ease notation, we often identify double negations, i.e. $\neg\neg\psi$ is identified with ψ.

Semantics An I *frame* is a tuple $F = \langle W, \{R_a\}_{a \in I}\rangle$ where W is a non-empty set of possible worlds, and for every $a \in I$, R_a is a binary relation on W. The class of all I frames is denoted by Fr_I. For σ a string over I, let R_σ stand for $R_{\sigma 1} R_{\sigma 2} \cdots R_{\sigma_{|\sigma|}}$; R_λ is the equality relation.

If $wR_\sigma w'$ for some string σ, we say that w' is reachable from w. A frame F is *rooted at* w_0 if every world w is reachable from w_0. We call w_0 the *root* of F.

An $\mathcal{L}$ *model* is of the form $M = \langle W, \{R_a\}_{a\in I}, \pi\rangle$ such that $\langle W, \{R_a\}_{a\in I}\rangle$ is an I frame (we say that M is *based on* this frame), and $\pi : \mathcal{P} \to Pow(W)$ is a valuation, i.e. $w \in \pi(p)$ means that p is true at w. For ϕ an $\mathcal{L}$ formula, we'll write $M, w \models \phi$ for ϕ is *true /satisfied at w in M*. The truth relation $\models$ is defined with induction on ϕ in the following way:

- $M, w \models p$ iff $w \in \pi(p)$ for $p \in \mathcal{P}$,
- $M, w \models \neg\phi$ iff not $(M, w \models \phi)$,
- $M, w \models \phi \wedge \psi$ iff $M, w \models \phi$ and $M, w \models \psi$,
- $M, w \models [a]\phi$ iff $\forall w' \in W(wR_aw' \Rightarrow M, w' \models \phi)$.

For Δ a set of formulas, we write $M, w \models \Delta$ if $M, w \models \phi$ for all $\phi \in \Delta$. The *size* of a model or a frame, denoted by $|\cdot|$ is the number of worlds in the model or frame.

The notion of truth can be extended to models and frames in the following way: ϕ is true in model M ($M \models \phi$) if $M, w \models \phi$ for every world w in M; ϕ is valid in frame F ($F \models \phi$) if $M \models \phi$ for every model M based on F. For Δ a set of $\mathcal{L}(I)$ formulas, $Fr(\Delta)$ is the set of all I frames F such that $F \models \Delta$.

In the same way, we can extend the notion of satisfiability: ϕ is satisfied in M if $M, w \models \phi$ for some world w in M, and ϕ is satisfiable in F (F satisfiable) if ϕ is satisfied in M for some model M based on F.

We usually look at satisfiability and validity with respect to a class of frames $\mathcal{F}$ instead of a single frame or model. ϕ is valid in $\mathcal{F}$ ($\mathcal{F} \models \phi$) if $F \models \phi$ for every frame $F \in \mathcal{F}$, and ϕ is satisfiable with respect to $\mathcal{F}/\mathcal{F}$ satisfiable if ϕ is satisfiable in some frame $F \in \mathcal{F}$.

The central problem of this thesis can then be formulated in the following way:

What is the complexity of $\mathcal{F}$ satisfiability?

Before we make precise what we mean by complexity, we first describe some useful constructions on frames and models.

Restrictions A technique that is often used in upper bound proofs is restricting the size of satisfying models and frames.

For $F = \langle W, \{R_a\}_{a\in I}\rangle$ a frame, and $\widehat{W} \subseteq W$, we define the frame $F|\widehat{W}$ in the obvious way, i.e. $F|\widehat{W} = \langle \widehat{W}, \{\widehat{R}_a\}_{a\in I}\rangle$ where $\widehat{R}_a = R_a|\widehat{W} = R_a \cap (\widehat{W} \times \widehat{W})$. If $\widehat{F} = F|\widehat{W}$ for some $\widehat{W}$, we call $\widehat{F}$ a *subframe* of F (and F a *superframe* of $\widehat{F}$). Similarly, for a model $M = \langle W, \{R_a\}_{a\in I}, \pi\rangle$ and $\widehat{W} \subseteq W$, let $M|\widehat{W} = \langle \widehat{W}, \{R_a|\widehat{W}\}_{a\in I}, \widehat{\pi}\rangle$ where $\widehat{\pi}(p) = \pi(p) \cap \widehat{W}$. We call $\widehat{M} = M|\widehat{W}$ a *submodel* of M.

It is obviously not the case that submodels preserve satisfiability. However, if $\widehat{W}$ is closed under all accessibility relations (i.e. for all $\widehat{w} \in \widehat{W} : \widehat{w}R_aw \Rightarrow w \in \widehat{W}$), then for all formulas ϕ and all worlds $w \in \widehat{W}$: $M, w \models \phi$ iff $\widehat{M}, w \models \phi$. We call subframes and submodels that satisfy this additional requirement *generated* subframes and submodels.

Generated submodels preserve satisfiability for all formulas. If we are only interested in preserving satisfiability of one specific formula in one specific world, we can restrict the set of worlds more drastically. For suppose for example that $M, w_0 \models \langle 1\rangle(\langle 2\rangle p \wedge \neg\langle 1\rangle q)$. Then we can safely remove all worlds that are *not* reachable by R_λ, R_1, R_{12} or R_{11}. We'll define *paths*(ϕ) as the set of relevant strings over I. In the example above, *paths*$(\phi) = \{\lambda, 1, 12, 11\}$. Formally, for ϕ a formula, define *paths*(ϕ) as follows:

- $\mathit{paths}(p) = \{\lambda\}$
- $\mathit{paths}(\neg\psi) = \mathit{paths}(\psi)$
- $\mathit{paths}(\psi \wedge \xi) = \mathit{paths}(\psi) \cup \mathit{paths}(\xi)$
- $\mathit{paths}([a]\psi) = \{\lambda\} \cup \{a\sigma | \sigma \in \mathit{paths}(\psi)\}$

It is easy to see that for any model $M = \langle W, \{R_a\}_{a\in I}, \pi\rangle$, world $w_0 \in W$ and formula ϕ the following holds:

$$M, w_0 \models \phi \Leftrightarrow M|\widehat{W}, w_0 \models \phi, \text{ where } \widehat{W} = \{w | w_0 R_\sigma w \text{ for some } \sigma \in \mathit{paths}(w)\}.$$

For uni-modal logics, $\widehat{W}$ has a particularly easy form:

$$\widehat{W} = \{w | w_0 R^i w \text{ for some } i \leq \text{ the modal depth of } \phi\}.$$

P-morphisms Given $\mathcal{L}$ models $M = \langle W, R_I, \pi\rangle$ and $M' = \langle W', R'_I, \pi'\rangle$, we say that f is a *p-morphism from M onto M'* if the following hold:

- f is a map from W onto W',
- if $wR_a w'$ then $f(w)R'_a f(w')$,
- if $f(w)R'_a w'$ then $wR_a u$ for some $u \in W$ such that $f(u) = w'$, and
- $\pi(p) = \{w | f(w) \in \pi'(p)\}$.

If the first three conditions are fulfilled, we say that f is a p-morphism from $\langle W, R_I\rangle$ onto $\langle W', R'_I\rangle$. What good are surjective p-morphisms? They provide an immediate equivalence between models, in the sense that for all formulas ϕ and all worlds $w \in W$: $M, w \models \phi$ iff $M', f(w) \models \phi$. It follows that surjective p-morphisms preserve validity: if $\langle W, R_I\rangle \models \phi$, then $\langle W', R'_I\rangle \models \phi$.

Logics The semantical approach given above is the usual way to define applied modal formalisms. An alternative definition is provided by the axiomatic approach. We will treat this approach here as well for the following reasons. First of all, an axiomatization can give extra information about the modal formalism. Furthermore, many theorems from modal logic that we will use are formulated in terms of logics rather than classes of frames.

A *normal modal logic* in $\mathcal{L}$ is a set L of $\mathcal{L}$ formulas such that:

- L contains all propositional tautologies,
- L is closed under substitution,
- $[a](p \rightarrow q) \rightarrow ([a]p \rightarrow [a]q) \in L$ for all $a \in I$,
- L is closed under modus ponens, i.e. if ϕ and $\phi \rightarrow \psi$ are in L, then so is ψ,
- L is closed under generalization, i.e. if $\phi \in L$, then so is $[a]\phi$.

ϕ is L *consistent* if $\neg\phi$ not in L. A set Δ is L consistent if for every finite $\Delta' \subseteq \Delta$, the formula $\bigwedge_{\phi\in\Delta'} \phi$ is L consistent.

The canonical model For L a logic, define the canonical model $M^c_L = \langle W^c, \{R^c_a\}_{a\in I}, \pi^c\rangle$ where:

- W^c consists of all maximal L consistent sets,
- $\langle \Gamma, \Delta\rangle \in R^c_a$ iff for all formulas ϕ, if $[a]\phi \in \Gamma$ then $\phi \in \Delta$,
- $\pi(p) = \{\Delta \in W^c | p \in \Delta\}$.

This definition ensures that $M^c_L, \Delta \models \phi$ iff $\phi \in \Delta$. It follows that every L consistent set Δ is satisfiable in a model for L. If M^c_L is based on an L frame, we call L a *canonical* logic.

Completeness For $\mathcal{L}$ a language, L an $\mathcal{L}$ logic and $\mathcal{F}$ a class of $\mathcal{L}$ frames:

- L is *complete* with respect to $\mathcal{F}$ if $F \models L$ for all $F \in \mathcal{F}$, and and if $\phi \notin L$ then $\neg\phi$ is $\mathcal{F}$ satisfiable. In this case, L consistency $=$ $\mathcal{F}$ satisfiability, and we often write L satisfiability for L consistency.
- L is *strongly complete* with respect to $\mathcal{F}$ if $F \models L$ for all $F \in \mathcal{F}$, and for all L consistent sets Δ: Δ is $\mathcal{F}$ satisfiable.

We say that L is (strongly) complete if L is (strongly) complete with respect to some class of frames (which holds iff L is (strongly) complete with respect to $\mathit{Fr}(L)$). Note that any canonical logic L is strongly complete. Not every logic is complete, but every class of frames $\mathcal{F}$ corresponds to the (complete!) logic $\{\phi | \mathcal{F} \models \phi\}$. We use the logic terminology on classes of frames: for instance, we will say that $\mathcal{F}$ is strongly complete if the associated logic is strongly complete.

Some Examples K is the minimal normal uni-modal logic, and, since any frame is a frame for K, K is canonical. Adding extra axioms gives us the following canonical logics:

Name	Axioms	Frames
T	$K + \Box p \to p$	R is reflexive
$S4$	$T + \Box p \to \Box\Box p$	R is reflexive and transitive
$S5$	$S4 + \Diamond\Box p \to p$	R is reflexive, transitive and symmetric
$S4.3$	$S4 + \Box(\Box p \to \Box q) \vee \Box(\Box q \to \Box p)$	R is reflexive, transitive and connected.

The finite model property A logic L has the *finite model property* (fmp) if every L consistent formula ϕ can be satisfied on a finite L model. It is well known that this implies that L is complete with respect to a class of finite frames, thus we will also refer to this notion as the finite frame property. The finite model property is useful in the context of complexity, for every finitely axiomatizable logic with the finite model property is decidable.

Since we are interested in refining the notion of decidability, it is natural to refine the notion of finite frame property: we say that L has the *$s(n)$-size frame property* if every L consistent formula ϕ is satisfiable on an L frame of size at most $s(n)$ where n is the size of the formula. (Note that this implies that L has the finite model property.) If L has the $p(n)$-size frame property for p a polynomial, we say that L has the *poly-size frame property*.

From [Urq81], we know that the finite model property does not imply decidability. We will use similar methods to show, as a first example of the complexity reasoning that will be used in this book, that even the poly-size frame property does not imply decidability.

Theorem 2.1.1 *There exists a uni-modal logic L such that L has the poly-size frame property, but L is undecidable.*

We show that there exist a continuum of uni-modal logics with the poly-size frame property. Since we work in a fixed countable language, only countably many of these logics can be decidable, and the theorem follows. Showing the existence of a continuum of logics with a certain property sounds more scary than it is, for we can translate the requirements to a countable number of logics (cf [Fin74]). In the sequel, we will define classes of uni-modal frames $\{\mathcal{F}_i\}_{i\in\mathbf{N}}$ such that the following two conditions are fulfilled.

- For each i there exists a formula ϕ_i such that ϕ_i is $\mathcal{F}_i$ satisfiable, but ϕ_i is *not* $\mathcal{F}_j$ satisfiable for any $j \neq i$, and
- there exists a *fixed* polynomial p such that for every i $\mathcal{F}_i$ has the $p(n)$-size frame property. (Note that this is stronger than requiring that every $\mathcal{F}_i$ has the poly-size frame property.)

If we can define $\{\mathcal{F}_i\}_{i\in\mathbf{N}}$ in this way, then for every $X \subseteq \mathbf{N}$, let $\mathcal{F}_X = \bigcup_{i\in X} \mathcal{F}_i$. All these classes define distinct logics. For suppose $X, Y \subseteq \mathbf{N}$ and $X \neq Y$. Assume that X is not a subset of Y, and let $i \in \mathbf{N}$ be such that $i \in X$, and $i \notin Y$. Then ϕ_i is $\mathcal{F}_X$ satisfiable, since $\mathcal{F}_i \subseteq \mathcal{F}_X$. But ϕ_i is not $\mathcal{F}_Y$ satisfiable, since ϕ_i is not $\mathcal{F}_j$ satisfiable for any $j \in Y$.

Furthermore, for all $X \subseteq \mathbf{N}$, $\mathcal{F}_X$ has the poly-size frame property. For suppose ϕ is $\mathcal{F}_X$ satisfiable. Then ϕ is $\mathcal{F}_i$ satisfiable for some $i \in X$. By the construction, ϕ is satisfiable on an $\mathcal{F}_i$ frame of size at most $p(n)$ for n the size of the formula.

It remains to construct the classes $\mathcal{F}_i$. All frames we use in the construction will be finite, rooted, linear and intransitive. Formally, let $\mathcal{F}_{lin}$ consist of all frames $\langle W, R\rangle$ such that $W = \{w_0, \ldots, w_k\}$, and for $i \neq j$, $w_i R w_j$ if and only if $j = i+1$. Thus, an $\mathcal{F}_{lin}$ frame is completely determined by its size and its set of reflexive worlds.

Though this is a very restricted class, it suffices for out purposes. For $i \geq 1$, define $\mathcal{F}_i$ as the class that consists of the following frames:

- All $\mathcal{F}_{lin}$ frames such that the last world is reflexive,
- all $\mathcal{F}_{lin}$ frames such that all worlds are irreflexive, and
- the frame F_i: the $\mathcal{F}_{lin}$ frame of size $i+1$ such that the root is the only reflexive world.

And let ϕ_i be the following formula:

$$\phi_i = p \wedge \Diamond p \wedge \Diamond(\neg p \wedge \Diamond^{i-1}\Box\bot).$$

We claim that these classes of frames and these formulas fulfill the two conditions given above. First we show that ϕ_i is satisfiable in F_i, and not satisfiable on any $\mathcal{F}_j$ frame for $j \neq i$. It is immediate that ϕ_i is satisfiable in the root of F_i, letting $\pi(p) = \{w_0\}$. Next, let F be a frame in $\mathcal{F}_j$ for $j \neq i$. We show that ϕ_i is not F satisfiable. There are three cases to consider:

- F is an $\mathcal{F}_{lin}$ frame that ends with a reflexive world. Then every world in F has a successor, and therefore no world can satisfy $\Box\bot$. But satisfiability of ϕ_i implies satisfiability of $\Box\bot$. It follows that ϕ_i is not satisfiable on F.

- F is an $\mathcal{F}_{lin}$ frame where all worlds are irreflexive. Then every world in F has at most one successor. But if ϕ_i is satisfied at some world w, then w satisfies $\Diamond p \wedge \Diamond \neg p$, which implies that w has at least two successors.

- Finally, suppose that $F = F_j$, and suppose that ϕ_i is F_j satisfiable. Let M_j be the model based on F_j, and w a world such that $M_j, w \models \phi_i$. Then w must have at least two successors. By the form of F_j, w is the root w_0 of F_j. Since $M_j, w_0 \models p$, and w_1 is the only other successor of w_0, it follows that $M_j, w_1 \models \Diamond^{i-1}\Box\bot$. But if $j < i$, no world is reachable in $i-1$ steps from w_1, and if $j > i$, then the only world reachable in $i-1$ steps from w_1 is w_i. But $w_i R w_{i+1}$, and thus $M_j, w_i \not\models \Box\bot$. It follows that $M_j, w_1 \not\models \Diamond^{i-1}\Box\bot$, which contradicts our assumption.

Finally, to show that the second condition is fulfilled, let ϕ be an $\mathcal{F}_i$ satisfiable formula. We will show that ϕ is satisfiable on a frame in $\mathcal{F}_i$ of size at most $m+2$ for m the modal depth of ϕ. First of all, it is easy to see that $\mathcal{F}_i$ is closed under generated subframes. Thus, we may assume that ϕ is satisfiable at the root w_0 of some frame $F \in \mathcal{F}_i$. If $|F| \leq m+2$, then we are done. So suppose that $w_0, w_1, \ldots, w_m, w_{m+1}$ and w_{m+2} are worlds in F. Since m is the modal depth of ϕ, ϕ is satisfiable in w_0 on the frame $F|\{w_0, \ldots, w_m\}$. Let G be the $\mathcal{F}_{lin}$ frame with $m+2$ worlds such that the last world is reflexive, and $G|\{w_0, \ldots, w_m\} = F|\{w_0, \ldots, w_m\}$. Then G satisfies ϕ at w_0, and $|G| = m+2$. Furthermore, G is a frame in $\mathcal{F}_i$, since the last world in G is reflexive. □

2.2 A Bit of Complexity

For a class of I frames $\mathcal{F}$, we investigate the complexity of $\mathcal{F}$ satisfiability, i.e. the complexity of the following problem:

Given a formula ϕ,

Question Is ϕ $\mathcal{F}$ satisfiable?

This might look like a proper definition of the problem, but it isn't. First of all, the problem depends on the choice of the set $\mathcal{P}$ of propositional variables. Secondly, ϕ is a string over the infinite alphabet $I \cup \mathcal{P} \cup \{[,], \wedge, \neg\}$. We can solve the second problem by requiring that $\mathcal{P}$ and I are given as strings over a finite alphabet. But this leads to strange behavior, for if $\mathcal{P}$ is given in unary, propositional satisfiability over this set is in P. Another possibility is to fix the set of propositional variables. Although this certainly solves many problems, we want to be free to use different sets of propositional variables in the constructions. The solution we take here is the following: we *encode* set $\mathcal{P}$ as $p_1, p_2, p_3, \ldots$ (where p_i is encoded as p followed by i as a binary string).

The problems with I are more complex. In this section, we will assume that I is finite. The problems encountered when I is infinite will be discussed in section 3.5.

Complexity is usually measured as a function of the length of the input. But what is the length of a formula? Usually, we mean the size of the formula, i.e. the length over the infinite alphabet. However, the actual length of the input is the length of the encoding. By the approach taken above, this is not a tremendous problem: if we replace the i-th variable in a formula ϕ by p_i, the length of the encoding of ϕ, denoted by $|\phi|$ is at most $n \log n$ for n the size of ϕ. Similarly, the length of the encoding of a model or frame is polynomial in the size (i.e. the number of worlds) in the structure. In the sequel, We will only consider complexity classes that are oblivious to these differences.

Let's move on to some complexity examples. We assume that the reader has at least a basic acquaintance with complexity classes like P, NP, PSPACE, EXPTIME etc. Completeness (not to be confused with the earlier defined notion of modal completeness!) and hardness are defined by means of polynomial time many-one reductions. For basic definitions see for example [BDG88]. In the proofs of the complexity of modal satisfiability problems, we often use that the following problems are in P. Checking if a formula is satisfied on a given model:

Given A finite model M, a world w in M, and a formula ϕ,

Question Does M satisfy ϕ (at w)?

And frame membership for a first order definable class of frames $\mathcal{F}$ (cf. [Imm87]):

Given A finite frame F,

Question $F \in \mathcal{F}$?

NP

Since all consistent normal modal logics are conservative extensions of propositional logic, it follows that all satisfiability problems considered are NP-hard. There do exist NP-complete satisfiability problems for non-trivial modal logics. As an example, we prove in detail that $S5$ satisfiability is in NP ([Lad77]). Recall that $S5$ satisfiability is satisfiability with respect to reflexive, symmetric and transitive frames. We show that every $S5$ satisfiable formula ϕ is satisfiable in an $S5$ frame of size at most $m+1$ where m is the number of modalities in ϕ. This implies that $S5$ satisfiability is in NP, since for any formula ϕ, ϕ is $S5$ satisfiable iff there exist a model M of size at most $m+1$, and a world w in M such that M is based on an $S5$ frame and $M, w \models \phi$. As the number of modalities in ϕ is obviously less than the length of ϕ, and verifying that M satisfies ϕ is in P, it follows that $S5$ satisfiability is in NP.

It remains to show that every $S5$ satisfiable formula ϕ is satisfiable in an $S5$ frame of size at most $m+1$, where m is the number of modalities in ϕ. Let $M = \langle W, R, \pi \rangle$ be a model based on an $S5$ frame, $w_0 \in W$ a world such that $M, w_0 \models \phi$. We suppose that M is generated. For every $\Diamond\psi$ subformula of ϕ, either $M, w \models \Diamond\psi$ for all $w \in W$, or $M, w \not\models \Diamond\psi$ for all $w \in W$. Let $\Diamond\psi_1, \Diamond\psi_2, \ldots, \Diamond\psi_k$ be all $\Diamond$ subformulas of ϕ that are satisfied in M. Note that k is at most m, the number of modalities in ϕ. For each i, let w_i be a world such that $M, w_i \models \psi_i$, and let $\widehat{M} = M|\{w_0, w_1, \ldots, w_k\}$. We claim that $\widehat{M}, w_0 \models \phi$. This proves that ϕ is satisfied on an $S5$ frame of size at most $m+1$ as required. To prove that $\widehat{M}, w_0 \models \phi$, we show with induction that for all $\psi \in Cl(\phi)$, and all $i \leq k$:

$$M, w_i \models \psi \Leftrightarrow \widehat{M}, w_i \models \psi$$

The only non-trivial step is for $\Diamond\psi$. First suppose that $M, w_i \models \Diamond\psi$. Then $\psi = \psi_j$ for some $j \leq k$. By definition, w_j was chosen in such a way that $M, w_j \models \psi_j$. By induction, $\widehat{M}, w_j \models \psi_j$, and thus $\widehat{M}, w_i \models \Diamond\psi_j$. On the other hand, if $\widehat{M}, w_i \models \Diamond\psi$, then $\widehat{M}, w_j \models \psi$ for some $j \leq k$, and therefore $M, w_j \models \psi$, which implies that $M, w_i \models \Diamond\psi$. This completes the proof of the NP upper bound on $S5$ satisfiability.

Many proofs of NP upper bounds for satisfiability problems proceed in a similar way, i.e. by showing that

- $\mathcal{F}$ has the poly-size frame property, and
- membership in $\mathcal{F}$ can be determined in polynomial time.

In section 2.4, we will use this proof method to obtain NP upper bounds for an infinite number of modal logics. Other examples of NP-complete modal satisfiability problems can be found in e.g. [ON80].

Though the two conditions given above imply that $\mathcal{F}$ satisfiability is in NP, it should be noted that the converse does not hold. For a trivial counter-example, let $\mathcal{F}$ consist of all frames $\langle W, W \times W\rangle$ such that $|W| \in A$ for A an undecidable subset of $\mathbf{N}$. Then $\mathcal{F}$ satisfiability = $S5$ satisfiability. But membership in $\mathcal{F}$ is undecidable, since A is reducible to this problem. On the other hand, we know from theorem 2.1.1 that the first condition is not sufficient.

PSPACE

PSPACE is the complexity class most associated with modal logic. First of all, for any index set $I \neq \emptyset$, satisfiability with respect to all I frames is PSPACE-complete, and so is satisfiability with respect to all tense frames (frames of the form $\langle W, R, R^{-1}\rangle$). In addition, the corresponding satisfiability problems with respect to all reflexive and/or transitive frames are PSPACE-complete as well. (see [Lad77] for the uni-modal case, [HM85] for the multi-modal case, and [Spa] for the tense case).

In this section, we review some results for uni-modal logics. The relation between multi-modal and uni-modal cases is the subject of chapter 3. We first review Ladner's proof of the PSPACE upper bound for K satisfiability. After this, we formulate some consequences of Ladner's PSPACE-hardness proof that we will use in the next chapters.

K satisfiability is in PSPACE To prove the PSPACE upper bound of K satisfiability, define a recursive function K-WORLD(Δ, Σ) (a slight variation of Ladner's procedure K-WORLD). For Δ and Σ sets of formulas, and Σ closed under subformulas and single negations, K-WORLD(Δ, Σ) will be true iff Δ is a maximal K satisfiable subset of Σ, i.e. iff there exists a model M and a world w such that for all $\psi \in \Sigma : (M, w \models \psi \Leftrightarrow \psi \in \Delta)$. This function can then be used to solve K satisfiability, since ϕ is K satisfiable iff there exists a set $\Delta \subseteq Cl(\phi)$ such that $\phi \in \Delta$ and K-WORLD$(\Delta, Cl(\phi))$ is true.

For Δ and Σ sets of formulas, Σ closed under subformulas and single negations,

$$K\text{-WORLD}(\Delta, \Sigma)$$
$$\text{iff}$$

- Δ is a maximally propositionally consistent subset of Σ, i.e.

  - $\psi \in \Delta \Rightarrow \psi \in \Sigma$,
  - $(\psi \in \Delta \Leftrightarrow \neg\psi \notin \Delta)$ for $\neg\psi \in \Sigma$
  - $(\psi_1 \wedge \psi_2 \in \Delta \Leftrightarrow \psi_1 \in \Delta$ and $\psi_2 \in \Delta)$ for $\psi_1 \wedge \psi_2 \in \Sigma$, and

- For each subformula $\Diamond\psi \in \Delta$ there exists a set Δ_ψ such that

  - $\psi \in \Delta_\psi$,
  - $\forall\xi(\Box\xi \in \Delta \Rightarrow \xi \in \Delta_\psi)$, and
  - K-WORLD(Δ_ψ, Σ'), where Σ' is the closure under subformulas and single negations of the set $\{\xi | \Box\xi \in \Sigma\}$.

That K-WORLD is correct can be proved by induction on the size of Σ. We won't go into the details of the proof here. Note that for the induction to go through, we need to show that $|\Sigma'| < |\Sigma|$. It is immediate that $\Sigma' \subseteq \Sigma$. Since the maximal modal depth of a formula in Σ' is less than the maximal modal depth of a formula in Σ, it follows that Σ' is a strict subset of Σ.

For any formula ϕ, ϕ is K satisfiable iff there exists a set $\Delta \subseteq Cl(\phi)$ such that $\phi \in \Delta$ and K-WORLD$(\Delta, Cl(\phi))$ is true. All subsets encountered in the execution of K-WORLD are subsets of $Cl(\phi)$, and each subset of $Cl(\phi)$ can be represented in space $O(|\phi|)$, by using pointers to a copy of the formula. (Note that every connective and proposition letter in ϕ corresponds to a subformula of ϕ and vice versa.) Therefore, at each level of recursion, we use space $O(|\phi|)$. After m recursive calls, for m the modal depth of ϕ, $\Sigma = \emptyset$. It follows that the recursion depth is bounded by m, and therefore certainly by $|\phi|$. The total amount of space used to determine K satisfiability of ϕ is therefore $O(|\phi|^2)$.

It might seem that K-WORLD is of a non-deterministic nature, since it contains the phrase "For each subformula $\Diamond\psi \in \Delta$ there *exists* a set Δ_ψ such that...."[1] Since PSPACE = NPSPACE [Sav70], this doesn't matter if we are just interested in proving that K satisfiability is in PSPACE. For a more precise bound however, note that this part can be executed by a deterministic machine using space $O(|\phi|)$: cycle through all subsets of $Cl(\phi)$ and for each set check if the set fulfills the conditions. If so, K-WORLD is true. If not, we try the next subset. Finally, if all sets have failed to fulfill our requirements, then K-WORLD is false. Cycling through all subsets of $Cl(\phi)$ can be be implemented by cycling through all binary strings of length $|\phi|$, since interpreting whether a string encodes a subset of $Cl(\phi)$ takes polynomial time.

A PSPACE-hardness criterion In [Lad77], Ladner proves that the provability problem for every logic between K and $S4$ is PSPACE-hard. This result is proven by reducing the PSPACE-complete set QBF of true quantified boolean formulas to modal satisfiability problems, by a polynomial time computable function f_{La} with the following properties:

- $A \in$ QBF $\Rightarrow f_{La}(A) \in S4$ satisfiability,

- $f_{La}(A) \in K$ satisfiability $\Rightarrow A \in$ QBF.

[1] In fact, K-WORLD is of an "alternating" nature [CKP81].

For L a logic between K and $S4$, f_{La} is a reduction from QBF to L consistency, since if $A \in$ QBF, then $f_{La}(A)$ is $S4$ satisfiable, and therefore $f_{La}(A)$ is L consistent, and if $f_{La}(A)$ is L consistent then $f_{La}(A)$ is K satisfiable, and hence $A \in$ QBF. It follows that L consistency is PSPACE-hard, and, since PSPACE is closed under complementation, L provability is PSPACE-hard.

We won't go into the details of the proof here. To prove PSPACE-hardness of other logics, we can of course try to adapt the reduction method given above. This often works, but leads to lots of duplications. What we look for in this section is a simple criterion that tells if for a class of frames $\mathcal{F}$, $\mathcal{F}$ satisfiability is PSPACE-hard by Ladner's method. We do not claim that we derive strongest criterion possible. Rather, we look for a good complexity/generality trade-off. We need the following observation: Ladner's method can be used if $\mathcal{F}$ can simulate binary trees. Formally, let the binary tree of depth n be a frame $\langle W_T, R_T\rangle$ such that $W_T = \{0, \ldots, 2^{n+1} - 2\}$ and $R_T = \{\langle i, 2i+1\rangle, \langle i, 2i+2\rangle\}$. (i.e. $\langle W_T, R_T\rangle$ is the full irreflexive, asymmetric, intransitive binary tree, with distance n from root to leaf). Define *depth*(i) as $\lfloor \log(i+1)\rfloor$, i.e. the distance from the root. Inspection of Ladner's reduction shows that the following hold:

- $f_{La}(A)$ is of the form $\phi_1 \wedge \bigwedge_{i=0}^{n} \Box^i \phi_2$, with ϕ_1, ϕ_2 of modal depth ≤ 1,
- if $A \in$ QBF then $f_{La}(A)$ is satisfiable in the root of the binary tree of depth n.
- if $f_{La}(A)$ is satisfiable then $A \in$ QBF.

This observation leads to the following PSPACE-hardness criterion.

Theorem 2.2.1 *Let $\mathcal{L}(I)$ be a language and $\mathcal{F}$ a class of I frames. If there exists a polynomial time computable function f such that for all uni-modal formulas ϕ of the form $\phi_1 \wedge \bigwedge_{i=0}^{n} \Box^i \phi_2$, with ϕ_1, ϕ_2 of modal depth ≤ 1 the following holds:*

- *$f(\phi)$ is an $\mathcal{L}$ formula,*
- *if ϕ is satisfiable in the root of the binary tree of depth n, then $f(\phi)$ is satisfiable in an $\mathcal{F}$ subframe,*
- *if $f(\phi)$ is satisfiable, then ϕ is satisfiable.*

Then every set of $\mathcal{L}$ formulas between $\mathcal{F}$ satisfiability and Fr_I satisfiability is PSPACE-hard.

Proof. First suppose that $\mathcal{F}$ is closed under subframes. Let C be a set of $\mathcal{L}$ formulas between $\mathcal{F}$ satisfiability and Fr_I satisfiability. We'll show that $f \cdot f_{La}$ reduces QBF to C, which implies the theorem. First suppose that $A \in$ QBF . Then $f_{La}(A)$ is of the form $\phi_1 \wedge \bigwedge_{i=0}^{n} \Box^i \phi_2$, with ϕ_1, ϕ_2 of modal depth ≤ 1, and $f_{La}(A)$ is satisfied in the root of the binary tree of depth n. It follows that $f(f_{La}(A)) \in \mathcal{F}$ satisfiability, and thus in C. For the converse, suppose that $f(f_{La}(A)) \in C$. Then $f(f_{La}(A)) \in Fr_I$ satisfiability. It follows that $f_{La}(A)$ in K satisfiability, and therefore $A \in$ QBF.

This proves the theorem for the special case that $\mathcal{F}$ is closed under subformulas. To prove the general case, let p_f be a new propositional variable and define function g as follows:

$$g(p) = p;\ \ g(\neg\phi) = \neg g(\phi);\ \ g(\phi \wedge \psi) = g(\phi) \wedge g(\psi);\ \ g([a]\phi) = [a](p_f \rightarrow g(\phi)).$$

It is immediate that g is polynomial time computable, and it follows by straightforward induction that for all models M and $\widehat{M} = M|\{w \in M | M, w \models p_f\}$

$$\widehat{M}, w \models \phi \text{ iff } M, w \models g(\phi).$$

This implies that $p_f \wedge g \cdot f \cdot f_{La}$ is a polynomial time comoputable reduction from QBF to C as required. □

We will use this theorem to prove PSPACE-hardness for two uni-modal examples. In the next chapter, we'll turn our attention to multi-modal logics. The theorem can roughly be stated as follows: if $\mathcal{F}$ can simulate binary trees, then $\mathcal{F}$ satisfiability is PSPACE-hard.

A PSPACE-hardness example Let $\mathcal{F}$ be the class of frames such that every world has at most two 2-step successors. We use theorem 2.2.1 to show that $\mathcal{F}$ satisfiability is PSPACE-hard. Let $M_T = \langle W_T, R_T, \pi_T \rangle$ be such that $\langle W_T, R_T \rangle$ is the binary tree of depth n, and $M_T, 0 \models \phi$. Binary trees can be simulated by $\mathcal{F}$ frames if we use R^2 as edges in the tree. Define the corresponding model $M = \langle W, R, \pi \rangle$ as follows:

- $W = W_T \cup \{i' | i \in W_T\}$
- $R = \{\langle i, i' \rangle, \langle i', j \rangle | i R_T j\}$
- $\pi(p) = \pi_T(p)$

It is immediate that for all $i, j \in W_T$, iR_Tj iff iR^2j. Furthermore, if iR^2w, then $w \in W_T$. Now it is clear how the reduction should be defined: just replace every occurrence of $\Box$ by $\Box\Box$. Formally, define f inductively as follows:

$$f(p) = p; f(\neg\psi) = \neg f(\psi); f(\psi_1 \wedge \psi_2) = f(\psi_1) \wedge f(\psi_2); f(\Box\psi) = \Box\Box f(\psi).$$

It is immediate that f is polynomial time computable. With induction, it is easy to prove that for all formulas ϕ, and for all $i \in W_T$,

$$M, i \models \phi \text{ iff } M, i \models f(\phi)$$

For propositional variables, this follows from the definition of π. The cases for $\neg\psi$ and $\psi_1 \wedge \psi_2$ follow immediately from the induction hypothesis. Finally, look at the case where $\phi = \Diamond\psi$. First suppose that $M_T, i \models \Diamond\psi$. Let j be such that iR_Tj and $M_T, j \models \psi$. Then, by definition, iR^2j, and by induction, $M, j \models f(\psi)$. Therefore, $M, i \models \Diamond\Diamond f(\psi)$ as required since $\Diamond\Diamond f(\psi) = f(\Diamond\psi)$.

For the converse, suppose that $M, i \models f(\Diamond\psi)$, i.e. $M, i \models \Diamond\Diamond f(\psi)$. Let w be such that iR^2w and $M, w \models f(\psi)$. By definition of R, $w \in W_T$ and iR_Tw. It follows that $M_T, w \models \psi$, and therefore, $M_T, i \models \Diamond\psi$.

This shows that f fulfills the second requirement of the theorem. For the third requirement, suppose that $f(\phi)$ is satisfiable. We need to show that ϕ is satisfiable as well. Let $M = \langle W, R, \pi \rangle$ be a model and w_0 a world such that $M, w_0 \models f(\phi)$. Define the corresponding model $\widehat{M} = \langle W, \widehat{R}, \pi \rangle$ such that $\widehat{R} = R^2$. It is immediate that for all $w \in W$, for all formulas ψ, $M, w \models f(\psi)$ iff $\widehat{M}, w \models \psi$. This proves that $\mathcal{F}$ satisfiability is PSPACE-hard. Note that in this case, we didn't use the special form of the formulas ϕ. □

PSPACE-hardness for $S4$ Let $\mathcal{F} = Fr(S4)$. (Based on Ladner) Let $\langle W_T, R_T\rangle$ be the binary tree of depth n. We want to simulate this tree by the binary reflexive transitive tree of depth n. ($\langle W_T, R\rangle$ where R is the reflexive transitive closure of R_T). The problem is how to simulate the edges of the tree. It is obvious that we can't use R itself, since R is transitive. However, it is certainly the case that for all $i, j \in W_T$, if $depth(j) = depth(i) + 1$, then iR_Tj iff iRj. If we label each $i \in W_T$ by its depth, using propositional vector $depth \in \{0, \ldots, n\}$ (that is, $depth$ consists of $\log n$ propositional variables and each assignment to these propositional variables is interpreted as an element in $\{0, \ldots, n\}$). $R_T = \{\langle i, j\rangle | iRj, M, i \models depth = d$ and $M, j \models depth = d + 1$ for some $d\}$. If we define g as follows:

$$g(\Box\psi) := \bigwedge_{0 \le d < n} (depth = d \rightarrow \Box(depth = d + 1 \rightarrow g(\psi)))$$

Then for all formulas ϕ and all i, $M_T, i \models \phi$ iff $M, i \models g(\phi)$. Unfortunately, g is not polynomial time computable for nested modalities. If we unpack the definition of $g(\Box\psi)$, we see that $g(\psi)$ occurs more than once at the right hand side of the definition, which leads to exponential blow-up. Now, we make use of the special form of formulas. Let $\phi = \phi_1 \wedge \bigwedge_{i=0}^n \Box^i\phi_2$, for ϕ_1 and ϕ_2 of modal depth ≤ 1. We define

$$f(\phi) = f(\phi_1 \wedge \bigwedge_{i=0}^{n} \Box^i\phi_2) = g(\phi_1) \wedge \bigwedge_{i=0}^{n} \Box^i g(\phi_2).$$

f is polynomial time computable and if $M_T, 0 \models \phi_1 \wedge \bigwedge_{i=0}^n \Box^i\phi_2$, then $M_T, 0 \models \phi_1$ and for all $i \in W_T$, $M_T, i \models \phi_2$. It follows that $M, 0 \models g(\phi_1)$, and $M, i \models g(\phi_2)$ for all $i \in W_T$. Therefore, $M, i \models g(\phi_1) \wedge \bigwedge_{i=0}^n \Box^i g(\phi_2)$.

For the third requirement, let $M = \langle W, R, \pi\rangle$ be a model and w_0 a world such that $M, w_0 \models f(\phi)$. Define the corresponding model $\widehat{M} = \langle W, \widehat{R}, \pi\rangle$ such that $i\widehat{R}j$ if and only if iRj, $M, i \models (depth = d)$ and $M, j \models (depth = d + 1)$ for some d. It is immediate that for all $w \in W$, for all formulas ψ, $M, w \models \psi$ iff $\widehat{M}, w \models g(\psi)$, and therefore, $\widehat{M}, w_0 \models \phi$. This proves that $S4$ satisfiability is PSPACE-hard. □

EXPTIME

What PSPACE is to (the join of) uni-modal logics, EXPTIME is to logics with more expressive power. For instance, the satisfiability problems for the following propositional logics are all EXPTIME-complete:

- PDL (Propositional Dynamic Logic): EXPTIME lower bound in [FL79], EXPTIME upper bound in [Pra79].

  In our notation, the language contains a modal operator for every regular expression over some set of atomic programs. We look at satisfiability with respect to frames with the additional requirement that: $R_{\alpha;\beta} = R_\alpha R_\beta$, $R_{\alpha\cup\beta} = R_\alpha \cup R_\beta$, and $R_{\alpha^*} = R_\alpha^* = \bigcup_{k\in\mathrm{N}} R_\alpha^k$.

- DPDL: Deterministic Propositional Dynamic Logic: lower bound in [Par80], upper bound in [BHP82]. The language for DPDL is the same as for PDL. Additional requirement on frames: for all atomic programs a, R_a is deterministic, i.e. for every world w, there is at most one world w' such that wR_aw'.

- Various logics for knowledge with an operator C for Common Knowledge [HM85]. The language contains $m+1$ modalities: $K_1, \ldots, K_m$ and C, where $K_i\psi$ stands for "processor i knows that ϕ," and $C\phi$ for "ϕ is common knowledge." All frames have the restriction that $R_C = (R_1 \cup \cdots \cup R_m)^*$. EXPTIME-completeness has been shown for:
    - $m \geq 1$ and all processors are K or T logics,
    - $m \geq 2$ and all processors are $S4$ or $S5$ logics. (Note that adding common knowledge for one transitive logic doesn't add expressive power.)
- Branching Time logics [EH85]. Interpreted on trees, with operators like $\forall\Box$: "at every branch in the tree," and $\exists\Box$: "at some branch in the tree."
- Various Attribute Value description formalisms with the ability to express generalizations and recursive constraints [BS].

What all these logics have in common is the ability to make universal statements, e.g. statements about what is true in all worlds in a model, or all reachable worlds, or in all worlds that are reachable by a certain set of edges.

As an example, look at the bimodal language with modal operators $\Box$ and $[u]$, where $[u]$ is the so-called universal modality, with semantics $M, w \models [u]\phi$ iff $M, w' \models \phi$ for all $w' \in M$. For a detailed discussion of the logical consequences of augmenting modal languages with the universal modality we refer to [GP92].

Let $Fr_{[u]}$ consist of all frames $\langle W, R, R_u \rangle$ such that $R_u = W \times W$ (We identify this frame with $\langle W, R \rangle$). We will show that $Fr_{[u]}$ satisfiability is EXPTIME-complete.

EXPTIME upper bound Using methods similar to [Pra79] and [HM85] we sketch a construction of a deterministic exponential time algorithm for $Fr_{[u]}$ satisfiability.

Let S be the set of all subsets Γ of $Cl(\phi)$ that are maximally propositionally consistent, and are closed under reflexivity of $[u]$; that is, if $[u]\psi \in \Gamma$ then ψ is also in Γ. Suppose ϕ is satisfiable in some model M. Let S_M be the set of subsets of $Cl(\phi)$ that actually occur in M, that is, $S_M = \{\Gamma \in S : M, w \models \Gamma,$ for some $w \in M\}$. Obviously, $S_M \subseteq S$, and every element of S_M contains the same $[u]$ formulas. Let $\Sigma \subseteq Pow(S)$, consisting of all $S' \subseteq S$ which are maximal with respect to:

$$\forall \Gamma, \Gamma' \in S', \forall [u]\psi \in Cl(\phi) : [u]\psi \in \Gamma \Leftrightarrow [u]\psi \in \Gamma'$$

If ϕ is satisfied in M, then there exists a set $S' \in \Sigma$ such that $S_M \subseteq S'$. What can we say about the size of Σ? Since $Cl(\phi)$ contains at most $2|\phi|$ elements, there exist at most $2^{2|\phi|}$ maximal sets $\widehat{S} \subseteq S$ fulfilling this condition. Since k is bounded by $|\phi|$, the size of Σ is exponential in the length of ϕ.

For every $S_1 \in \Sigma$, we will construct a sequence of sets $S_1 \supset S_2 \supset S_3 \supset \cdots$ such that: if ϕ is satisfiable in a model M, and $S_M \subseteq S_1$, then $S_M \subseteq S_i$ for all i.

Suppose we have defined S_i. Call a set $\Gamma \in S_i$ inconsistent if one of the following situations occurs:

1. $\langle u \rangle \psi \in \Gamma$, but for all $\Gamma' \in S_i$: $\psi \notin \Gamma'$, or
2. $\Diamond\psi \in \Gamma$, but there is *no* $\Gamma' \in S_i$ such that $\psi \in \Gamma'$ and $\forall \Box\xi \in Cl(\phi)(\Box\xi \in \Gamma \Rightarrow \xi \in \Gamma')$.

If there are inconsistent sets in S_i, then we let S_{i+1} consist of all sets of S_i that are *not* inconsistent, otherwise, we can stop the construction, since ϕ is satisfiable iff $\phi \in \Gamma$ for some set $\Gamma \in S_i$.

Since S_1 is of exponential size, and S_{i+1} is strictly included in S_i, the algorithm terminates after at most exponentially many cycles. Determining which sets in S_i are inconsistent takes polynomial time in the length of the representation of S_i. Thus, for every member of Σ, the algorithm takes at most deterministic exponential time. Since Σ is of exponential size, we can determine if ϕ is satisfiable in EXPTIME. □

EXPTIME lower bound: For the lower bound, we can use the same reduction as in the lower bound proof for PDL [FL79]. Again, we won't go into the details of the proof. We just state some properties of the reduction that immediately follow from the proof. Let A be some EXPTIME-complete set and let f_{FL} be the polynomial time reduction from A to PDL satisfiability from [FL79]. For all x, there are only two modalities occurring in $f_{FL}(x)$: $[\vdash]$ and $[\vdash^*]$, where $\vdash$ is an atomic program. We use $\Box$ and $[*]$ for these modalities. The following holds:

- $f_{FL}(x)$ is of the form $\psi_1 \wedge [*]\psi_2$, where ψ_1, ψ_2 are of modal depth ≤ 1 with only modality $\Box$,
- if $f_{FL}(x)$ is satisfiable, then $f_{FL}(x)$ is satisfied at the root of a finite binary tree.
- if $f_{FL}(x)$ is satisfiable, then $x \in A$.

This immediately implies that $Fr_{[u]}$ satisfiability is EXPTIME-hard, using this reduction with $[u]$ instead of $[*]$.

Note that this is all very similar to the PSPACE case described above. Using the same arguments, we can prove the following analog of theorem 2.2.1 that will be used in chapter 4.

Theorem 2.2.2 *Let $\mathcal{L}(I)$ be a language, and $\mathcal{F}$ be a class of I frames. If there exists a polynomial time computable function f such that for all formulas ϕ of the form $\phi_1 \wedge [*]\phi_2$, with $\phi_1, \phi_2 \in \mathcal{L}(\Box)$ of modal depth ≤ 1 the following holds:*

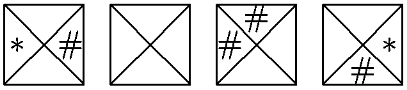
- *$f(\phi)$ is an $\mathcal{L}$ formula,*
- *if ϕ is satisfiable in the root of finite binary tree, then $f(\phi)$ is satisfiable on an $\mathcal{F}$ subframe,*
- *if $f(\phi)$ is satisfiable, then ϕ is satisfiable.*

Then every set of $\mathcal{L}$ formulas between $\mathcal{F}$ satisfiability and Fr_I satisfiability is EXPTIME-hard.

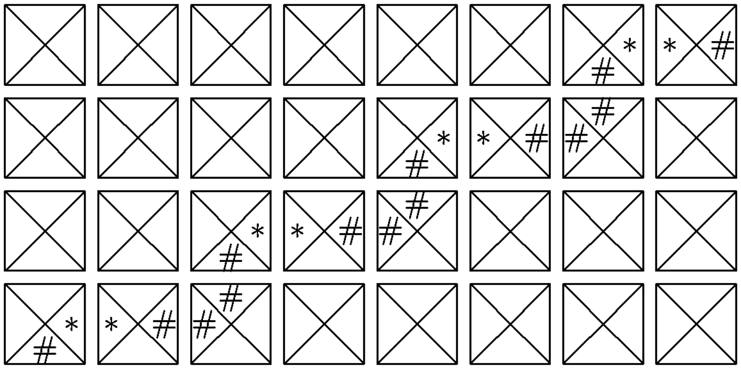
2.3 Tiling Problems

Moving up in the complexity hierarchy we now arrive at undecidable cases. As is shown in [Har83], *tiling problems* provide a particularly elegant method of proving lower bounds for modal logics, so we'll use such an approach here.

A tile T is a 1×1 square fixed in orientation with colored edges *right*(T), *left*(T), *up*(T), and *down*(T) taken from some denumerable set. A tiling problem takes the following form: given a finite set of $\mathcal{T}$ of tile types, can we cover a certain part of $\mathbf{Z}\times\mathbf{Z}$, using only tiles of this type, in such a way that adjacent tiles have the same color on the common edge, and such that the tiling obeys certain constraints? One of the attractive features of tiling problems is that they are very easy to visualize. As an example, consider the following puzzle. Suppose $\mathcal{T}$ consists of the following four types of tile:



Can an 8 by 4 rectangle be tiled with the fourth type placed in the left hand corner? Indeed it can:



There exist complete tiling problems for many complexity classes ([Emd83, Lew78]). For our destructive purposes we will make use of the following two tiling problems:

As shown in [Ber66, Rob71], the following problem is Π_1^0-complete[2]

N × N tiling: Given a finite set $\mathcal{T}$ of tiles, can $\mathcal{T}$ tile $\mathbf{N}\times\mathbf{N}$?

That is, does there exist a function t from $\mathbf{N}\times\mathbf{N}$ to $\mathcal{T}$ such that:

$$\begin{array}{lcl} \mathit{right}(t(n,m)) & = & \mathit{left}(t(n+1,m)),\ \text{and} \\ \mathit{up}(t(n,m)) & = & \mathit{down}(t(n,m+1))? \end{array}$$

We will encounter satisfiability problems that are not just undecidable, but *highly* undecidable, in fact, Σ_1^1-complete [Rog67]. The hardness part of such results is often proved by constructing a reduction from the following Σ_1^1-complete tiling problem [Har86]:

N × N recurrent tiling: Given a finite set $\mathcal{T}$ of tiles, and a tile $T_1\in\mathcal{T}$, can $\mathcal{T}$ tile $\mathbf{N}\times\mathbf{N}$ such that T_1 occurs in the tiling infinitely often on the first row?

That is, does there exist a function t from $\mathbf{N}\times\mathbf{N}$ to $\mathcal{T}$ such that: $\mathit{right}(t(n,m)) = \mathit{left}(t(n+1,m))$, $\mathit{up}(t(n,m)) = \mathit{down}(t(n,m+1))$, and the set $\{i : t(i,0) = T_1\}$ is infinite?

[2] A set is Π_1^0 if it is the complement of a recursively enumerable set. We'll also use coRE for this class.

2.4 Towards General Modal Complexity Results

To prove general complexity results, it makes sense to look at the form of general theorems in modal logic. Often, these theorems are of the form: every logic L with property X has property Y. In modal logic, property Y stands for instance for being complete, having the finite model property, or being decidable. In complexity theory, we are interested in properties like: being PSPACE-hard, being in EXPTIME, or being undecidable.

Perhaps the easiest example of such a general modal theorem is Bull's theorem [Bul66]: Every normal uni-modal logic L containing $S4.3$ has the finite model property. Recall that $S4.3$ frames are those frames that are reflexive, transitive, and connected.[3]

Following Fine's proof of Bull's theorem [Fin72], we show the following complexity theoretic analog:

Theorem 2.4.1 *The satisfiability problem for every logic containing S4.3 is NP-complete.*

For the proof, we fix a normal uni-modal logic L containing $S4.3$. As in the case of $S5$, we show that L has the poly-size frame property, and that membership in $Fr(L)$ is in P. Let's start with the poly-size frame property.

Lemma 2.4.2 *Any L consistent formula ϕ is satisfiable on an L frame of size at most $m+2$, for m the number of modalities in ϕ.*

Let $M = \langle W, R, \pi \rangle$ be a finite model based on an L frame and $w_0 \in W$ a world such that $M, w_0 \models \phi$ (such a model exists by Bull's theorem). We assume that M is rooted at w_0. Let $\Diamond\psi_1, \Diamond\psi_2, \ldots, \Diamond\psi_k$ be all $\Diamond$ subformulas of ϕ that are satisfied at w_0 in M. Note that k is at most m, the number of modalities in ϕ. For each i, let w_i be a last world satisfying ψ_i, i.e. a world such that $M, w_i \models \psi_i$ and if not wRw_i, then $M, w \not\models \psi_i$ (such a world exists since M is based on a finite rooted $S4.3$ frame).

The worlds $\{w_0, w_1, \ldots, w_k\}$ are needed to keep ϕ satisfied at w_0. However, if we restrict M to these worlds, it is possible that the underlying frame is not an L frame. (This problem didn't occur when we considered $S5$, since any $S5$ subframe is an $S5$ frame). To ensure that the restricted frame remains an L frame, let w_{k+1} be a last world in M, i.e. for all $w \in M$, wRw_{k+1}. Let $\widehat{W} = \{w_0, w_1, \ldots, w_{k+1}\}$, and let $\widehat{M} = M|\widehat{W}$. We show that $\widehat{M}$ satisfies ϕ at w_0, and that the underlying frame $\widehat{M}$ is an L frame, from which lemma 2.4.2 follows.

To prove that $\widehat{M}, w_0 \models \phi$, we show with induction that for all $\psi \in Cl(\phi)$, and all $i \leq k+1$:

$$M, w_i \models \psi \Leftrightarrow \widehat{M}, w_i \models \psi$$

The only non-trivial step is for $\Diamond\psi$. First suppose that $M, w_i \models \Diamond\psi$. Since M is rooted at w_0 and R is reflexive, $w_0 R w_i$. It follows that $M, w_0 \models \Diamond\psi$, and thus $\psi = \psi_j$ for some $j \leq k$. By definition, w_j is a last world satisfying ψ_j. Therefore, $w_i R w_j$, and by induction $\widehat{M}, w_j \models \psi_j$, which implies that $\widehat{M}, w_i \models \Diamond\psi_j$. On the other hand, if $\widehat{M}, w_i \models \Diamond\psi$, then $\widehat{M}, w_j \models \psi$ for some $j \leq k$ (by the maximality of w_j), and therefore $M, w_j \models \psi$, which implies that $M, w_i \models \Diamond\psi$.

To prove lemma 2.4.2, it remains to show that the $\widehat{M}$ is based on an L frame. We will show that there exists a surjective p-morphism f from $\langle W, R \rangle$ onto $\langle \widehat{W}, R|\widehat{W} \rangle$. Since

[3] "Connectedness" is equivalent to the first order property: $\forall xyz(xRy \wedge xRz \rightarrow yRz \vee zRy)$.

$\langle \widehat{W}, R\rangle$ is an L frame, and surjective p-morphisms preserve validity, this implies that $\langle \widehat{W}, R|\widehat{W}\rangle$ is an L frame as well. The existence of such a p-morphism follows from the following more general lemma:

Lemma 2.4.3 *Let F and G be two finite, rooted $S4.3$ frames. The following two statements are equivalent:*

- *There exists a surjective p-morphism from F onto G.*
- *G is isomorphic to a subframe of F that contains a last world of F.*

First suppose that there exists a surjective p-morphism from F onto G. Let w_l be a last world in F, and let $\widehat{W}$ consist of w_l and exactly one world in $f^{-1}[v]$ for every world $v \in G$ such that $v \neq f(w)$. It is immediate that taking $\widehat{F} = F|\widehat{W}$ is the subframe we're after.

For the converse, suppose that $\widehat{W}$ is a subset of the worlds in F, and $\widehat{W}$ contains a last world w_l of F. We show that there exists a surjective p-morphism from F onto $F|\widehat{W}$. Define f as follows: $f(w) = w$ for $w \in \widehat{W}$, and if $w \notin \widehat{W}$, let $f(w)$ be a first world $\widehat{w} \in \widehat{W}$ such that $wR\widehat{w}$ (where first means that for all $w' \in \widehat{W}, wRw' \Rightarrow \widehat{w}Rw'$). Note that f is well-defined, since for all $w \in W$, wRw_l and $w_l \in \widehat{W}$. It remains to show that f is a surjective p-morphism. That f is surjective is immediate. Next suppose that wRw'. Since $w'Rf(w')$ and R is transitive, it follows that $wRf(w')$. By definition, $f(w)$ is a first element in $\widehat{W}$ such that $wRf(w)$, which implies that $f(w)Rf(w')$. Finally, suppose that $f(w)Rf(w')$. Then $wRf(w)Rf(w')$, which implies that $wRf(w')$, since $f(w)$ is a first world $\widehat{w}$ in $\widehat{W}$ such that $wR\widehat{w}$. Since $f(f(w')) = f(w')$, the last condition for p-morphisms is fulfilled. □

To complete the proof of theorem 2.4.1, it remains to show that membership in *Fr*(L) can be determined in polynomial time. From Fine [Fin72], we know that there exists a finite set X of finite $S4.3$ frames that characterizes *Fr*(L) in the following way:

> $F \models L$ iff F is an $S4.3$ frame, and there does *not* exist a surjective p-morphism from F onto any frame in X.

Since verifying that a finite frame is an $S4.3$ frame is obviously in P (we check a first order condition), and since X is a fixed finite set, we only have to prove that determining if there exists a surjective p-morphism from a finite $S4.3$ frame F onto a *fixed* finite $S4.3$ frame G is in P. We can of course look at all functions from F to G, and see if one of them is a surjective p-morphism. However, there exist exponentially many such functions, so this would take too much time. However, if we apply lemma 2.4.3, we see that we only have to verify that there exists a set $\widehat{W}$ of worlds in F such that $F|\widehat{W}$ is isomorphic to G, and $\widehat{W}$ contains a last world of F. Since $|\widehat{W}| = |G|$, we need to investigate less than $|F|^{|G|}$ embeddings, and this amount is polynomial in the size of F, since G is fixed. □

Chapter 3

The Complexity of the Join

3.1 Introduction

In this chapter, we look at the complexity of the satisfiability problems for the simplest kind of multi-modal logics. For two normal uni-modal logics, L_1 and L_2 with operators [1] and [2] respectively, the join of L_1 and L_2, denoted by $L_1 \oplus L_2$ is the minimal normal bimodal logic that contains L_1 and L_2. As recently shown in [FS, KW91], joins inherit many properties from their uni-modal fragments. Following [FS], we say that a property *transfers* if the join of two normal uni-modal logics with this property has the property as well.

Theorem 3.1.1 ([FS, KW91]) *Completeness, strong completeness and the finite model property transfer. Decidability transfers under the condition of completeness.*

A natural question is: does complexity transfer? Note that lower bounds *do* transfer for consistent logics, since then $L_1 \oplus L_2$ is a conservative extension of L_1 and L_2. However, it is in general not the case that upper bounds transfer. A counterexample (under the assumption that NP $\neq$ PSPACE) is given by the join of two $S5$ logics, since $S5$ satisfiability is NP-complete [Lad77], while $S5 \oplus S5$ satisfiability is PSPACE-complete [HM85]. On the other hand, the join does not always add to the complexity, as the satisfiability problems for K and $K \oplus K$ are both PSPACE-complete ([Lad77, HM85]). In this chapter, we investigate in what way the the complexity of the satisfiability problem of the join is related to the complexity of the satisfiability problems of its uni-modal fragments.

As in the previous chapter, we will look at the satisfiability problems with respect classes of frames. For $\mathcal{F}_1$ and $\mathcal{F}_2$ two classes of uni-modal frames, the join of $\mathcal{F}_1$ and $\mathcal{F}_2$, denoted by $\mathcal{F}_1 \oplus \mathcal{F}_2$, consists of the frames $\langle W, R_1, R_2 \rangle$ such that $\langle W, R_1 \rangle \in \mathcal{F}_1$ and $\langle W, R_2 \rangle \in \mathcal{F}_2$. Joins of logics and joins of frames are related in the following way:

Theorem 3.1.2 ([FS]) *If $\mathcal{F}_1$ and $\mathcal{F}_2$ are closed under disjoint union, L_1 is complete with respect to $\mathcal{F}_1$ and L_2 is complete with respect to $\mathcal{F}_2$, then $L_1 \oplus L_2$ is complete with respect to $\mathcal{F}_1 \oplus \mathcal{F}_2$.*

Note that closure under disjoint union is essential, since for instance $\{\bullet\!\!\longrightarrow\!\!\bullet\} \oplus \{\bullet\} = \emptyset$. Since the logic $\{\phi \mid \mathcal{F} \models \phi\}$ is complete with respect to $\mathcal{F}$, we obtain the following frame version of this theorem.

Corollary 3.1.3 *If $\mathcal{F}_1, \widehat{\mathcal{F}}_1, \mathcal{F}_2$, and $\widehat{\mathcal{F}}_2$ are closed under disjoint union, $\mathcal{F}_1$ satisfiability $= \widehat{\mathcal{F}}_1$ satisfiability and $\mathcal{F}_2$ satisfiability $= \widehat{\mathcal{F}}_2$ satisfiability, then $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability $= \widehat{\mathcal{F}}_1 \oplus \widehat{\mathcal{F}}_2$ satisfiability.*

The frame version of the problem to be considered in this section can be stated as follows: given two classes of frames $\mathcal{F}_1$ and $\mathcal{F}_2$ that are closed under disjoint union, in what way is the complexity of the satisfiability problem with respect to $\mathcal{F}_1 \oplus \mathcal{F}_2$ related to the complexity of the satisfiability problems with respect to $\mathcal{F}_1$ and $\mathcal{F}_2$. Note that solving the frame version of the problem solves the problem for complete logics as well, since if L_1 and L_2 are complete, then these logics are complete with respect to classes of frames closed under disjoint union, and, by theorem 3.1.2, $L_1 \oplus L_2$ is complete with respect to the join of these two classes.

The goal of this chapter is to answer the following questions:

- Under what conditions (on the classes of frames and the complexity class) do upper bounds transfer?
- When do upper bounds *not* transfer? And what can we say about lower bounds on the join in that case?

Section 3.2 is devoted to the first question. We show that under reasonable conditions, upper bounds containing PSPACE transfer. In addition, we derive a criterion for NP upper bound transfer. In section 3.3 we investigate the second question. In section 3.4, we show that the results of the previous two sections are optimal in the sense that given two classes of frames $\mathcal{F}_1$ and $\mathcal{F}_2$ that are closed under disjoint union, we are in one of the following three cases:

I One of the classes of frames, say $\mathcal{F}_2$, is trivial, in which case the satisfiability problem for the join is polynomial time reducible to the satisfiability problem for $\mathcal{F}_1$,

II $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability is PSPACE-hard by section 3.3, or

III $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability is in NP by section 3.2.

Finally, in section 3.5, we discuss to what extent the results in the three previous sections go through for the join of arbitrarily many multi-modal logics.

3.2 Upper Bound Transfer

In this section, we investigate in what way upper bounds on the complexity of $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability are related to upper bounds on the the complexity of satisfiability problems of the uni-modal fragments. In particular, we want to determine under what conditions upper bounds transfer, i.e. when does the fact that $\mathcal{F}_1$ and $\mathcal{F}_2$ satisfiability are in $\mathcal{C}$, for some complexity class $\mathcal{C}$ and $\mathcal{F}_1, \mathcal{F}_2$ closed under disjoint union, imply that $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability is in $\mathcal{C}$?

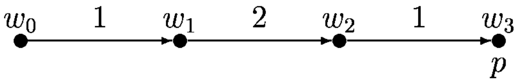
From Fine [FS], we know that this is certainly the case for $\mathcal{C}$ is REC, since decidability transfers. Furthermore, in [HM85], it is shown that the satisfiability problems for $K \oplus K$, $T \oplus T$, $S4 \oplus S4$ and $S5 \oplus S5$ are in PSPACE, from which we might conjecture that PSPACE upper bounds transfer. We prove this result under an extra "uniformity" condition, which is possessed by many modal logics. This result is not restricted to the PSPACE case; we show that for $\mathcal{C}$ a nice complexity class that is large enough, $\mathcal{C}$ upper bounds transfer under this extra condition.

For classes not containing PSPACE, this result obviously doesn't go through. However, we will give a condition under which NP upper bounds *do* transfer. In section 3.4, we will show that this criterion is optimal, in the sense that it completely characterizes when the join of two non-trivial classes of frames is in NP (under the assumption that NP $\neq$ PSPACE).

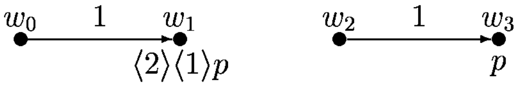
Satisfying Models for the Join

We will show how to build satisfying models based on $\mathcal{F}_1 \oplus \mathcal{F}_2$ frames from models based on $\mathcal{F}_1$ and $\mathcal{F}_2$ frames. The construction is similar to the construction used in the proof of completeness transfer in [FS, KW91].

As an example, look at the following model for $\langle 1\rangle\langle 2\rangle\langle 1\rangle p$:



That this model satisfies $\langle 1\rangle\langle 2\rangle\langle 1\rangle p$, follows from the fact that p is satisfied in w_3. Now look at the corresponding 1-model:



We want this model to really correspond to the previous model. But here, the assignment to the propositional variable p is not enough to conclude that this model satisfies $\langle 1\rangle\langle 2\rangle\langle 1\rangle p$; we need the information that w_1 satisfies $\langle 2\rangle\langle 1\rangle p$. That is, in this model $\langle 2\rangle\langle 1\rangle p$ should be viewed as a propositional variable.

We can make this change of views explicit by introducing a new propositional variable $p_{\langle 2\rangle\langle 1\rangle p}$, and use functions that translate formulas from one language into another. This is the approach taken in [KW91]. Another approach is taken in [FS], where using replacement functions is avoided by making the language ambiguous: if the language considered is $\mathcal{L}_1$, we take $\langle 2\rangle\langle 1\rangle p$ to be a propositional variable. We look at the following three languages:

- $\mathcal{L}_{12} = \mathcal{L}(\{1,2\}, \mathcal{P})$
- $\mathcal{L}_1 = \mathcal{L}(\{1\}, \mathcal{P} \cup \{[2]\phi | \phi \text{ an } \mathcal{L} \text{ formula}\})$
- $\mathcal{L}_2 = \mathcal{L}(\{2\}, \mathcal{P} \cup \{[1]\phi | \phi \text{ an } \mathcal{L} \text{ formula}\})$

We'll use this approach, because it leads to more elegant formulations of the construction.

Construction To determine whether a formula is $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiable, define a recursive predicate $\mathcal{F}_1 \oplus \mathcal{F}_2$-WORLD as follows, for Γ and Σ sets of formulas, and $a \in \{1,2\}$:

$$\mathcal{F}_1 \oplus \mathcal{F}_2\text{-WORLD}(a, \Gamma, \Sigma)$$
$$\text{iff}$$
$$\Sigma = \emptyset \text{ or there exist a model } M = \langle W, R_a, \pi\rangle, \text{ and a world } w_0 \in W \text{ such that:}$$

- $\langle W, R_a \rangle \in \mathcal{F}_a$,
- $Form_M(w_0) \cap \Sigma = \Gamma$ ($Form_M(w)$ is the set of formulas true in M at w),
- $\forall w \neq w_0 : \mathcal{F}_1 \oplus \mathcal{F}_2$-WORLD$(\overline{a}, Form_M(w) \cap \Sigma(w), \Sigma(w))$, where $\overline{1} = 2$ and $\overline{2} = 1$, and where $\Sigma(w)$ (the relevant formulas at w) is defined smallest family of sets satisfying:
  - $\Sigma(w_0) = \Sigma$,
  - if wR_aw' and $[a]\psi \in \Sigma(w)$ then $\psi \in \Sigma(w')$,
  - $\Sigma(w)$ is closed under subformulas and single negations.

This construction is a simplified version of the construction used in the proof of completeness transfer in [FS, KW91]. From these proofs, it follows that ϕ is $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiable iff there exists a set $\Gamma \subseteq Cl(\phi)$ such that $\phi \in \Gamma$, $\mathcal{F}_1 \oplus \mathcal{F}_2$-WORLD$(1, \Gamma, Cl(\phi))$ and $\mathcal{F}_1 \oplus \mathcal{F}_2$-WORLD$(2, \Gamma, Cl(\phi))$.

Can this construction be used to prove upper bound transfer? Let's first look at a specific instance of this question: can this construction be used to show that $K \oplus K$ satisfiability is in PSPACE? The answer is yes, but it seems that we can't draw the conclusion just from the fact that K satisfiability is in PSPACE. This is caused by the fact that the uni-modal models used in the construction are not just any model, they are models with the extra restriction that we only allow certain sets of relevant formulas. For instance, the formula $\langle 1 \rangle [2] p \wedge \langle 1 \rangle [2] \neg p \wedge \langle 2 \rangle \top$ viewed as an $\mathcal{L}_1$ formula is satisfiable on the reflexive singleton, but $\{[2]p, [2]\neg p, \langle 2 \rangle \top\}$ viewed as a set of $\mathcal{L}_2$ formulas is not satisfiable. This example shows that we can't just take any model, and that we can't take a model of minimal size.

Making this precise, we introduce the following general notion. For a language $\mathcal{L} = \mathcal{L}(I)$, $\mathcal{F}$ a class of I frames, and $\mathcal{R} \subseteq Pow(\mathcal{L})$, we define *$\mathcal{F}$ satisfiability under restriction $\mathcal{R}$* as follows:

Given An $\mathcal{L}$ formula ϕ,

Question Does there exist an $\mathcal{L}$ model $M = \langle W, \{R_a\}_{a \in I}, \pi \rangle$ and a world $w_0 \in W$ such that:

- M is based on an $\mathcal{F}$ frame,
- $M, w_0 \models \phi$,
- for every world w, $\Sigma(w) \cap Form_M(w) \in \mathcal{R}$, where $\Sigma(w)$ is the earlier introduced set of formulas relevant at w.

Note that $\mathcal{F}$ satisfiability is $\mathcal{F}$ satisfiability with restriction $Pow(\mathcal{L})$. How should we define the complexity of such problems? It seems reasonable not to count the complexity of computing $\mathcal{R}$ membership. To be more precise, on input ϕ we do not count the complexity of computing $\mathcal{R}$ membership for subsets of $Cl(\phi)$ of modal depth less than ϕ.[1]

We say that $\mathcal{F}$ satisfiability under restrictions is in $\mathcal{C}$ if there exists an oracle Turing machine M in $\mathcal{C}$ such that:

[1] This is similar to the notion of self-reducibility (cf [Sel88].

- On input ϕ, M queries only subsets of $Cl(\phi)$ of modal depth less than the modal depth of ϕ,
- For all $\mathcal{R} \subseteq Pow(\mathcal{L})$: ϕ is $\mathcal{F}$ satisfiable under restriction $\mathcal{R}$ iff $M^{\mathcal{R}}$ accepts ϕ.

As a first example, note that $S5$ satisfiability under restrictions is in NP. For suppose M witnesses that ϕ is $S5$ satisfiable under restriction $\mathcal{R}$. In the proof of the NP upper bound for $S5$ satisfiability, it was shown that M can be restricted to at most $m+1$ worlds, where m is the modal depth of ϕ, such that the restricted model still satisfies ϕ and every world in the restricted model satisfies the same set of relevant formulas as in the original model. It is immediate that the restricted model satisfies ϕ under restriction $\mathcal{R}$, and thus $S5$ satisfiability under restrictions is in NP. The same argument can be used to show that L satisfiability under restrictions is in NP for all $S4.3$ extensions L, by inspection of the proof of theorem 2.4.1.

For a non-NP example, we show that K satisfiability under restrictions is in PSPACE, using the following variation of recursive predicate K-WORLD from the introduction: for Δ and Σ sets of formulas, and Σ closed under subformulas and single negations, $K^{\mathcal{R}}$-WORLD(Δ, Σ) will be true iff Δ is a maximal subset of Σ that is K satisfiable under restriction $\mathcal{R}$.

$K^{\mathcal{R}}$-WORLD(Δ, Σ)
iff

- Δ is a maximal propositionally consistent subset of Σ, i.e.
  - $\psi \in \Delta \Rightarrow \psi \in \Sigma$,
  - $(\neg\psi \in \Delta \Leftrightarrow \psi \notin \Delta)$ for $\neg\psi \in \Sigma$
  - $(\psi_1 \wedge \psi_2 \in \Delta \Leftrightarrow \psi_1 \in \Delta$ and $\psi_2 \in \Delta)$ for $\psi_1 \wedge \psi_2 \in \Sigma$, and
- For each subformula $\Diamond\psi \in \Delta$ there exists a set Δ_ψ such that
  - $\psi \in \Delta_\psi$,
  - $\forall\xi(\Box\xi \in \Delta \Rightarrow \xi \in \Delta_\psi)$,
  - $K^{\mathcal{R}}$-WORLD(Δ_ψ, Σ'), where Σ' is the closure under subformulas and single negations of the set $\{\xi | \Box\xi \in \Sigma\}$,
  - $\Delta_\psi \in \mathcal{R}$.

In the same way, the PSPACE upper bound proofs for T, $K4$ and $S4$ satisfiability imply PSPACE upper bounds for the corresponding satisfiability under restrictions problems.

Now we return to the question of upper bound transfer:

Theorem 3.2.1

- *If $\mathcal{C}$ is a space class that is closed under polynomial time reductions and that contains PSPACE, then $\mathcal{C}$ upper bounds for satisfiability under restrictions transfer.*
- *If $\mathcal{C}$ is a time class that is closed under polynomial time reductions and exponentiation, then $\mathcal{C}$ upper bounds for satisfiability under restrictions transfer.*

Proof. Suppose that for $a \in \{1,2\}$, $\mathcal{F}_a$ satisfiability under restrictions is in $\mathcal{C}$, as witnessed by oracle machine M_a. Define $\mathcal{R}_a$ such that $\Gamma \in \mathcal{R}_a$ iff $\mathcal{F}_1 \oplus \mathcal{F}_2$-WORLD$(a, \Gamma, Cl(\Gamma))$. From the construction of $\mathcal{F}_1 \oplus \mathcal{F}_2$-WORLD we immediately obtain the following:

- $\mathcal{F}_1 \oplus \mathcal{F}_2$-WORLD$(a, \Gamma, \Sigma)$ iff
- $\bigwedge \Gamma \wedge \neg \bigvee(\Sigma \setminus \Gamma)$ is $\mathcal{F}_a$ satisfiable under restriction $\mathcal{R}_{\overline{a}}$ iff
- M_a with oracle $\mathcal{R}_{\overline{a}}$ accepts $\bigwedge \Gamma \wedge \neg \bigvee(\Sigma \setminus \Gamma)$.

Suppose that $\mathcal{F}_1$ and $\mathcal{F}_2$ satisfiability are in DSPACE$(s(n))$ and DTIME$(t(n))$. We prove that for $\Gamma \subseteq \Sigma \subseteq Cl(\phi)$, $a \in \{1,2\}, n = |\phi|$ and m the modal depth of Σ it holds that $\mathcal{F}_1 \oplus \mathcal{F}_2$-WORLD$(a, \Gamma, \Sigma)$ in DSPACE$(ms(n^2))$ and DTIME$(t(n^2)^m)$.

The proof is with induction on m:

- If $m = 0$ then M_a doesn't query any strings. The length of the formula $\bigwedge \Gamma \wedge \neg \bigvee(\Sigma \setminus \Gamma)$ is less than $|\phi|^2$. It follows that M_a uses space at most $s(n^2)$, and time at most $t(n^2)$.
- Next suppose that $m > 0$. We know that $\mathcal{F}_1 \oplus \mathcal{F}_2$-WORLD$(a, \Gamma, \Sigma)$ iff M_a with oracle $\mathcal{R}_{\overline{a}}$ accepts $\bigwedge \Gamma \wedge \neg \bigvee(\Sigma \setminus \Gamma)$.

  M_a on this input can only query subsets of $Cl(\phi)$ of modal depth less than m. It follows from the induction hypothesis that every query to $\mathcal{R}$ can be computed in DSPACE$((m-1)s(n^2))$, and DTIME$(t(n^2)^{m-1})$.

  Thus, $\mathcal{F}_1 \oplus \mathcal{F}_2$-WORLD$(a, \Gamma, \Sigma)$ can be computed as follows: simulate M_a, and for each query, compute the answer to this query (i.e. whether the query is a member of $\mathcal{R}_{\overline{a}}$). Since queries can be handled one at a time, we only need extra space to compute one query. It follows that we need DSPACE$(s(n^2) + (m-1)s(n^2)) =$ DSPACE$(ms(n^2))$. On the other hand, the time that is needed is proportional to the number of queries multiplied by the time per query. It follows that we need DTIME$(t(n^2)t(n^2)^{m-1}) =$ DTIME$(t(n^2)^m)$.

For nondeterministic time classes, this proof does not go through, since in general these classes are not known to be closed under complementation. However, a slight variation of the construction above gives the wanted transfer result. Instead of computing whether a query belongs to $\mathcal{R}_{\overline{a}}$, we guess the set of queries in $\mathcal{R}_{\overline{a}}$ and verify that all these queries are indeed in $\mathcal{R}_{\overline{a}}$. This gives us an NTIME$(t(n^2)^m)$ algorithm if $\mathcal{F}_1$ and $\mathcal{F}_2$ satisfiability are in NTIME$(t(n))$. That this construction is correct follows from the fact that M works for every restriction $\mathcal{R}$. For this implies that M is positive, i.e. if $\mathcal{R} \subseteq \mathcal{R}'$ and $M^{\mathcal{R}}$ accepts, then so does $M^{\mathcal{R}'}$. □

The NP case The previous theorem showed transfer of upper bounds for classes like PSPACE and EXPTIME. As we have mentioned before, NP upper bounds do not transfer in general (under the assumption that NP $\neq$ PSPACE). However, as the following theorem shows, under the extra restriction that every $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiable formula is satisfiable on an $\mathcal{F}_1 \oplus \mathcal{F}_2$ frame of which the relevant part is of polynomial size, NP upper bounds do transfer. Although this is an obvious restriction, we will show in the sequel that this theorem is optimal, in the sense that – under the assumption that NP $\neq$ PSPACE – satisfiability for the join of two non-trivial classes of frames can only be in NP by this theorem.

Theorem 3.2.2 *Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be closed under disjoint union. If $\mathcal{F}_1$ and $\mathcal{F}_2$ satisfiability are in NP, and there exists a polynomial p such that for every $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiable formula ϕ there exists a model $M = \langle W, R_1, R_2, \pi\rangle$ such that:*

- $\langle W, R_1, R_2\rangle \in \mathcal{F}_1 \oplus \mathcal{F}_2$,
- $M, w_0 \models \phi$,
- $|\{w | w_0 R_{paths(\phi)} w\}| \leq p(|\phi|)$,

Then $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability is in NP.

The obvious way to approach this is by looking at the following algorithm. On input ϕ:

1. Guess a model M of size $\leq p(n)$, and a world $w_0 \in W$,
2. verify that M satisfies ϕ,
3. verify that M is based on $F|\{w|w_0 R_{paths(\phi)} w\}$ for some $F \in \mathcal{F}_1 \oplus \mathcal{F}_2$.

The problem is the last step, for even under the assumption that $\mathcal{F}_1$ and $\mathcal{F}_2$ satisfiability are in NP, step 3 can be undecidable. For let A be an undecidable subset of $\mathbf{N}$, and let F_k be defined as $\langle W, R\rangle$ such that $W = \{w_i\}_{i \leq k}$, and $R = \{\langle w_i, w_{i+1}\rangle | i < k\} \cup \{\langle w_0, w_0\rangle\} \cup \{\langle w_k, w_k\rangle\}$. Let $\mathcal{F}$ consist of the closure under disjoint union of the frame $F = \langle W, R\rangle$ such that $W = \{w_i\}_{i \in \mathbf{N}}$, $R = \{\langle w_i, w_{i+1}\rangle | i \in \mathbf{N}\} \cup \{w_0, w_0\}$, and all frames F_k for $k \in A$. It is immediate that A is reducible to the problem of step 3 given above, by mapping k to the pair $\langle F_k, \phi\rangle$ for some ϕ of modal depth k. But $\mathcal{F}$ satisfiability is in NP, since any $\mathcal{F}$ satisfiable formula is satisfiable on F.

To prove the theorem, we will relate $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability to $\mathcal{F}_1$ and $\mathcal{F}_2$ satisfiability. This is not very difficult, just rather awkward. Let's first define some abbreviations that will help in making the construction more readable:

- Define the formula $mss(\Gamma, \Sigma)$ which reflects the fact that Γ is a maximal satisfiable subset of Σ:
$$mss(\Gamma, \Sigma) := \bigwedge \Gamma \wedge \neg \bigvee (\Sigma \setminus \Gamma)$$
- For Σ a set of formulas, $a \in \{1, 2\}$, $(\Sigma)_a$ is the closure under subformulas and single negations of the set $\{\psi | [a]\psi \in \Sigma\}$. This definition can be extended to strings $\sigma \in \{1, 2\}^*$ in the obvious way: $(\Sigma)_\lambda = \Sigma$; $(\Sigma)_{a\sigma} = ((\Sigma)_a)_\sigma$.

Now prove the following equivalence:

ϕ is $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiable
iff

There exist some $k < p(n)$ and subsets of $Cl(\phi)$: $\Gamma_0, \Gamma_1, \ldots, \Gamma_k$, $\Sigma_0, \Sigma_1, \ldots, \Sigma_k$ such that

- $\phi \in \Gamma_0$, Σ_i closed under subformulas and single negations,
- for all $i < k$, $a \in \{1, 2\}$ the formula $\phi_{i,a}$ is $\mathcal{F}_a$ satisfiable:
$$\phi_{i,a} := mss(\Gamma_i, \Sigma_i) \wedge \bigwedge_{\ell \leq m} [a^\ell](\bigvee_j mss(\Gamma_j \cap (\Sigma_i)_{a^\ell}, (\Sigma_i)_{a^\ell})$$

Note that this claim implies that $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability is in NP, for on input ϕ we only have to guess $k < p(n)$, and $2(k+1)$ subsets of $Cl(\phi)$, and verify that $\phi \in \Gamma_0$, $\Sigma_0 = Cl(\phi)$, Σ_i closed under subformulas and single negations, and for all $i < k$, $a \in \{1,2\}$ the formula $\phi_{i,a}$ is $\mathcal{F}_a$ satisfiable. Since $\mathcal{F}_a$ satisfiability is in NP and $\phi_{i,a}$ can be constructed in polynomial time, it follows that $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability is in NP.

To prove the claim, we argue as follows: For the right implication, suppose $M = \langle W, R_1, R_2, \pi\rangle$, $w_0 \in W$ and $M, w_0 \models \phi$. Let $\widehat{W} = \{w \in W | w_0 R_{paths(\phi)} w\}$. Then $|\widehat{W}| \leq p(n)$, by assumption. Let $\widehat{W} = \{w_i | i < k\}$. For every i, let Σ_i be the set of relevant formulas at w_i, and let $\Gamma_i = Form_M(w) \cap \Sigma_i$. It is easy to verify that these sets fulfill the requirements.

We prove the converse by using the definition of $\mathcal{F}_1 \oplus \mathcal{F}_2$-WORLD. With induction on the size of Σ, we prove the following: if Σ is closed under subformulas and single negations and $\Sigma \subseteq \Sigma_i$, then $\langle a, \Gamma_i \cap \Sigma, \Sigma\rangle \in \mathcal{F}_1 \oplus \mathcal{F}_2$-WORLD. This proves the theorem, since $\phi \in \Gamma_0$.

- Suppose $|\Sigma| = 0$. By definition, $\langle a, \emptyset, \emptyset\rangle \in \mathcal{F}_1 \oplus \mathcal{F}_2$-WORLD.

- Suppose the claim holds for all sets of size $< |\Sigma|$. Suppose $\Sigma \subseteq \Sigma_i$. We show that $\langle a, \Gamma_i \cap \Sigma, \Sigma\rangle \in \mathcal{F}_1 \oplus \mathcal{F}_2$-WORLD.

  Since $\phi_{i,a}$ is $\mathcal{F}_a$ satisfiable, there exists a model $M = \langle W, R_a, \pi\rangle$ such that

  $$\langle W, R_a\rangle \in \mathcal{F}_a \text{ and } M, w_0 \models \phi_{i,a}$$

  From the definition of $\phi_{i,a}$, it follows immediately that $Form_M(w_0) \cap \Sigma_i = \Gamma_i$, and therefore, since $\Sigma \subseteq \Sigma_i$, $Form_M(w_0) \cap \Sigma = \Gamma_i \cap \Sigma$

  To prove that $\langle a, \Gamma_i \cap \Sigma, \Sigma\rangle \in \mathcal{F}_1 \oplus \mathcal{F}_2$-WORLD, it remains to show that for all $w \neq w_0$, $\langle \overline{a}, Form_M(w) \cap \Sigma(w), \Sigma(w)\rangle \in \mathcal{F}_1 \oplus \mathcal{F}_2$-WORLD. (recall that $\Sigma(w)$ was defined as follows: $\Sigma(w_0) = \Sigma$; if $wR_a w'$ and $[a]\psi \in \Sigma(w)$ then $\psi \in \Sigma(w)$; $\Sigma(w)$ closed under subformulas and single negations).

  If $\Sigma(w) = \emptyset$, we are done. So, suppose that $\Sigma(w) \neq \emptyset$. Then $w_0 R_a^\ell w$ for some $\ell > 0$ ($w \neq w_0$!) and $\Sigma(w) = (\Sigma)_{a^\ell}$. Since $M, w_0 \models \phi_{i,a}$, it follows that for some j, $M, w \models mss(\Gamma_j \cap (\Sigma_i)_{a^\ell}, (\Sigma_i)_{a^\ell})$, which implies that $Form_M(w) \cap (\Sigma_i)_{a^\ell} = \Gamma_j \cap (\Sigma_i)_{a^\ell}$.

  Since $\Sigma \subseteq \Sigma_i$, it is immediate that $(\Sigma)_{a^\ell} \subseteq (\Sigma_i)_{a^\ell}$. We may therefore conclude that $Form_M(w) \cap (\Sigma)_{a^\ell} = \Gamma_j \cap (\Sigma)_{a^\ell}$. Since $(\Sigma)_{a^\ell} = \Sigma(w)$, it holds that $Form_M(w) \cap \Sigma(w) = \Gamma_j \cap \Sigma(w)$.

  Since $\Sigma(w)$ is a strict subset of Σ, and $\Sigma \subseteq \Sigma_i$, it follows by the inductive hypothesis that $\langle \overline{a}, Form_M(w) \cap \Sigma(w), \Sigma(w)\rangle \in \mathcal{F}_1 \oplus \mathcal{F}_2$-WORLD. □

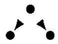
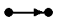
3.3 The Power of the Join

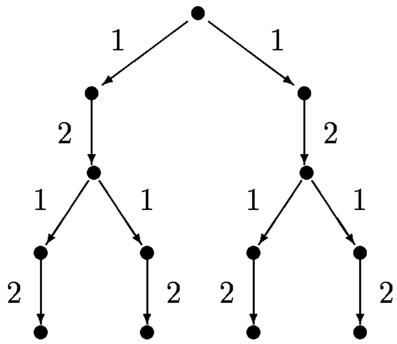
Whereas the previous section was devoted to determining when upper bounds transfer, this section addresses the dual problem: when do upper bounds *not* transfer, or, in other words, when does the join contribute to the complexity?

Given the results of the previous section and the fact that $S5 \oplus S5$ satisfiability is PSPACE complete, while $S5$ satisfiability is in NP, it makes sense to focus on determining when the join causes PSPACE hardness. We derive a criterion that is applicable in many cases; in fact, we show in section 3.4, that this criterion completely characterizes when the join can contribute to the complexity.

How do we prove PSPACE hardness for $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability? Recall from theorem 2.2.1 that it suffices to construct a polynomial time computable function f such that for all uni-modal formulas ϕ of the form $\phi_1 \wedge \bigwedge_{i=0}^{n} \Box^i \phi_2$, with ϕ_1, ϕ_2 of modal depth ≤ 1, the following holds:

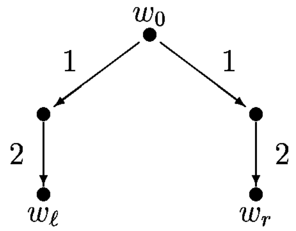
- $f(\phi)$ is an $\mathcal{L}$ formula,
- if ϕ is satisfiable in the root of the binary tree of depth n, then $f(\phi)$ is satisfiable in an $\mathcal{F}_1 \oplus \mathcal{F}_2$ subframe,
- if $f(\phi)$ is satisfiable, then ϕ is satisfiable.

As an example, look at the case where $\mathcal{F}_1$ is the closure under disjoint union and generated subframes of the frame  and $\mathcal{F}_2$ the closure under disjoint union and generated subframes of . The satisfiability problems of $\mathcal{F}_1$ and $\mathcal{F}_2$ are obviously in NP, since both problems amount to satisfiability with respect to a fixed finite frame. However, it turns out that even in this simple case, $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability is PSPACE complete. The upper bound follows from the previous section. For the hardness part, we construct a function f as given above. This boils down to the ability of $\mathcal{F}_1 \oplus \mathcal{F}_2$ subframes to simulate binary trees. In this specific case, it is easy to see how to do this: look at the following $\mathcal{F}_1 \oplus \mathcal{F}_2$ frame:



This certainly looks like a binary tree. Furthermore, trees of arbitrary depth can be encoded in this way. Indeed, it is readily seen that defining $f(\phi)$ as the result of substituting [1][2] for $\Box$ in ϕ fulfills the conditions of theorem 2.2.1 as repeated above, which proves PSPACE hardness for this simple case.

Note that all the information of this tree simulation is contained in the following $\mathcal{F}_1 \oplus \mathcal{F}_2$ frame $\widehat{F}$:



Binary tree simulations can be viewed as being constructed from $\widehat{F}$ frames by replacing each node i by a copy of $\widehat{F}$, and identifying the w_ℓ (w_r) world of the frame belonging to i with the w_0 world of the frame belonging to j for j the left (right) child of i.

In general, assume that $\mathcal{F}_1$ and $\mathcal{F}_2$ are closed under disjoint union, and let $\widehat{F}$ be an $\mathcal{F}_1 \oplus \mathcal{F}_2$ subframe, rooted at w_0 and containing worlds w_ℓ and w_r. The goal of this section is to determine what the requirements on $\widehat{F}$ are to let the construction as sketched in the example go through, and to conclude PSPACE hardness for $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability. We first derive these requirements intuitively; the formal proofs can be found after the statement of the theorem.

In the sequel, we need the following lemma, which gives the (obvious) correspondence between $\mathcal{F}_1 \oplus \mathcal{F}_2$ subframes and $\mathcal{F}_1$ and $\mathcal{F}_2$ subframes. The proof of this lemma can be found in the last section of this chapter.

Lemma 3.3.1 *Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two classes of uni-modal frames closed under disjoint union. $\langle W, R_1, R_2\rangle$ is a subframe of $\mathcal{F}_1 \oplus \mathcal{F}_2$ iff $\langle W, R_1\rangle$ is a subframe of $\mathcal{F}_1$ and $\langle W, R_2\rangle$ is a subframe of $\mathcal{F}_2$.*

First of all, note that if we build tree simulations from $\widehat{F}$ frames in the way as described above, we identify w_0 worlds with w_ℓ and w_r worlds. It seems reasonable to require that w_0, w_ℓ and w_r have the same reflexive behavior, i.e. for $a \in \{1,2\}$, either all three worlds are a reflexive, or all three worlds are a irreflexive. Next, we want the constructed tree to be an $\mathcal{F}_1 \oplus \mathcal{F}_2$ subframe. This is not true for every frame $\widehat{F}$: Suppose for instance that $w_0 \widehat{R}_1 w_\ell$, and that $\mathcal{F}_1$ consists of the closure under disjoint union of the frame $\bullet\!\rightarrow\!\bullet$. Then the tree constructed from $\widehat{F}$ is not an $\mathcal{F}_1 \oplus \mathcal{F}_2$ subframe, as identification of w_0 and w_ℓ worlds leads to arbitrarily long R_1 paths. This problem can be avoided by requiring that w_0 has no non-reflexive $\widehat{R}_1$ edges, and w_ℓ and w_r have no non-reflexive $\widehat{R}_2$ edges or vice versa. This ensures that any R_a connected set in the tree is a subset of one of the copies of $\widehat{F}$. Since we assume that $\widehat{F}$ is an $\mathcal{F}_1 \oplus \mathcal{F}_2$ subframe, it then follows that every R_a connected set in the tree is an $\mathcal{F}_a$ subframe, and thus, by lemma 3.3.1, the tree itself is an $\mathcal{F}_1 \oplus \mathcal{F}_2$ subframe.

These two requirements are enough to simulate binary trees by $\mathcal{F}_1 \oplus \mathcal{F}_2$ frames. However, this does *not* yet imply PSPACE hardness. Look for instance at the case where $\mathcal{F}_1$ and $\mathcal{F}_2$ consist of the closure under disjoint union of all linear intransitive irreflexive frames. Then $\mathcal{F}_1 \oplus \mathcal{F}_2$ frames can simulate binary trees by using R_1 to simulate the left successor, and R_2 to simulate the right successor. However, $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability is in NP. The crucial point is that for any formula ϕ, only polynomially many worlds are relevant to determine satisfiability of ϕ. In other words, if we want PSPACE hardness then, apart from being able to encode binary trees, we also have to be able to reach all worlds in the tree.

Reachability can be forced by putting the following condition on $\widehat{F}$: there exists a string $\sigma \in \{1,2\}^+$ such that w_ℓ and w_r are reachable from w_0 by $\widehat{R}_\sigma$. This forces that for a binary tree of depth n, all (exponentially many) nodes are reachable by the (polynomial size) relation: $\bigcup_{i=0}^{n} R_\sigma^i$. Making the above precise, we obtain the following theorem:

Theorem 3.3.2 *Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be closed under disjoint union. If there exists an $\mathcal{F}_1 \oplus \mathcal{F}_2$ subframe $\widehat{F} = \langle \widehat{W}, \widehat{R}_1, \widehat{R}_2\rangle$ such that:*

1. *$\widehat{W} \supseteq \{w_0, w_\ell, w_r\}$, and for some $\sigma \in \{1,2\}^+$: $w_0 \widehat{R}_\sigma w_\ell$ and $w_0 \widehat{R}_\sigma w_r$,*

2. *w_0, w_ℓ and w_r have the same reflexive behavior, i.e. for $a = 1, 2$:*

$$w_0 \widehat{R}_a w_0 \Leftrightarrow w_\ell \widehat{R}_a w_\ell \Leftrightarrow w_r \widehat{R}_a w_r,$$

3. For some $a \in \{1, 2\}$:

- *w_0 has no non-reflexive $\widehat{R}_a$ edges, i.e. if $w_0 \widehat{R}_a w$ or $w \widehat{R}_a w_0$, then $w_0 = w$.*
- *w_ℓ and w_r have no non-reflexive $\widehat{R}_{\overline{a}}$ edges, i.e. if $w_\ell \widehat{R}_{\overline{a}} w$ or $w \widehat{R}_{\overline{a}} w_\ell$ then $w = w_\ell$, and if $w_r \widehat{R}_{\overline{a}} w$ or $w \widehat{R}_{\overline{a}} w_r$ then $w = w_r$.*

Then $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability is PSPACE-hard.

Proof. We will construct a polynomial time computable function f such that for all uni-modal formulas ϕ of the form $\phi_1 \wedge \bigwedge_{i=0}^{n} \Box^i \phi_2$, with ϕ_1, ϕ_2 of modal depth ≤ 1:

- ϕ is an $\mathcal{L}$ formula,
- if ϕ is satisfiable in the root of the binary tree of depth n, then $f(\phi)$ is satisfiable in an $\mathcal{F}_1 \oplus \mathcal{F}_2$ subframe,
- if $f(\phi)$ is satisfiable, then so is ϕ.

Suppose $\phi = \phi_1 \wedge \bigwedge_{i=0}^{n} \Box^i \phi_2$, where ϕ_1 and ϕ_2 of modal depth ≤ 1 and suppose ϕ is satisfiable at the root of the binary tree of depth n. Let $M_T = \langle W_T, R_T, \pi_T \rangle$ be the model witnessing this fact, i.e. $\langle W_T, R_T \rangle$ is the binary tree of depth n with root $0 \in W_T$, and $M_T, 0 \models \phi$. $\langle W_T, R_T \rangle$ will be simulated by the $\mathcal{F}_1 \oplus \mathcal{F}_2$ subframe built from copies of $\widehat{F}$ in the way described above, i.e. by replacing each node i by a copy of $\widehat{F}$, and identifying the w_ℓ (w_r) world of the frame belonging to i with the w_0 world of the frame belonging to j for j the left (right) child of i.

Formally, let $F = \langle W, R_1, R_2 \rangle$ where $W = W_T \times (\widehat{W} \setminus \{w_\ell, w_r\})$, and R_1 and R_2 are inherited from $\widehat{R}_1$ and $\widehat{R}_2$ in the obvious way. That is, let g_T be the mapping function from $W_T \times \widehat{W}$ to W defined by: $g_T(w) = w$ for $w \in W$, and $g_T(\langle i, w_\ell \rangle) = \langle j, w_0 \rangle$ for j the left child of i, and $g_T(\langle i, w_r \rangle) = \langle j, w_0 \rangle$ for j the right child of i, and let R_b be defined by: $w R_b w'$ iff for some $i \in W_T$ and $\widehat{w}, \widehat{w}' \in \widehat{W}$ it is the case that $\widehat{w} \widehat{R}_b \widehat{w}'$, $g_T(\langle i, \widehat{w} \rangle) = w$, and $g_T(\langle i, \widehat{w}' \rangle) = w'$.

We first show that F is an $\mathcal{F}_1 \oplus \mathcal{F}_2$ subframe. Since w_0, w_ℓ and w_r have the same reflexivity behavior, it is easy to see that F consists of copies of $\widehat{F}$ in the sense that for every i in W_T, $F | g_T[\{i\} \times \widehat{W}]$ is isomorphic to $\widehat{F}$ by associating $g_T(\langle i, w \rangle)$ with w.

To prove that F is an $\mathcal{F}_1 \oplus \mathcal{F}_2$ subframe, it suffices to show by lemma 3.3.1, that for every $b \in \{1, 2\}$ and for every R_b connected set $W' \subseteq W$, $\langle W', R_b | W' \rangle$ is an $\mathcal{F}_b$ subframe. We will show that every world in an R_b connected set W' belongs to the same copy of $\widehat{F}$, i.e. there exists a node i such that $W' \subseteq g_T[\{i\} \times \widehat{W}]$. Since $F | g_T[\{i\} \times \widehat{W}]$ is isomorphic to $\widehat{F}$, and $\widehat{F}$ is an $\mathcal{F}_1 \oplus \mathcal{F}_2$ subframe, this immediately implies that $\langle W', R_b | W' \rangle$ is an $\mathcal{F}_b$ subframe as required. Now suppose for a contradiction that the worlds in W' do not belong to the same copy of $\widehat{F}$. Since W' is R_b connected, there must exist worlds w_1, w_2 and w_3 in W' such that $w_1 (R_b \cup R_b^{-1}) w_2$ and $w_1 (R_b \cup R_b^{-1}) w_3$, and w_2 and w_3 do not belong to the same copy of $\widehat{F}$. By definition of R_b, this can only be the case if w_1 is of the form $\langle i, w_0 \rangle$, for only worlds of this form can belong to more than one copy of $\widehat{F}$. Let j be the parent of i, and suppose that i is a left child. It follows that (modulo permutation) $w_2 \in g_T[\{j\} \times \widehat{W}]$ and $w_3 \in g_T[\{i\} \times \widehat{W}]$. Furthermore, $g_T^{-1}(\langle i, w_0 \rangle) = \{\langle i, w_0 \rangle, \langle j, w_\ell \rangle\}$. By definition of R_b, it follows that both w_0 and w_ℓ have non-reflexive R_b edges, which contradicts the third condition of the theorem. This proves that F is an $\mathcal{F}_1 \oplus \mathcal{F}_2$ subframe.

Node i of the binary tree is simulated by world $\langle i, w_0\rangle$ in F. But how do we simulate the edges of the tree? Since w_ℓ and w_r are $\widehat{R}_\sigma$ reachable from w_0 in $\widehat{F}$, it follows that $\langle i, w_0\rangle R_\sigma \langle j, w_0\rangle$ in F if j is a child of i. Thus, R_σ looks like a promising candidate for simulation of the edges of the tree. However, it is quite possible that $\langle i, w_0\rangle R_\sigma \langle j, w_0\rangle$ for j not a child of i. To simulate the edges of the tree, we have to avoid this situation.

It is easy to see that j is a child of i iff $\mathit{depth}(j) = \mathit{depth}(i) + 1$ and there exists an R_σ path from $\langle i, w_0\rangle$ to $\langle j, w_0\rangle$ that does not contain $\langle k, w_0\rangle$ for k the parent of i or the sibling of i. From this observation follows the following rather awkward formulation, which can be translated directly into modal formulas: Suppose i is a node at depth d, then j is a child of i iff $\mathit{depth}(j) = \mathit{depth}(i)+1$ and there exists an R_σ path from $\langle i, w_0\rangle$ to $\langle j, w_0\rangle$ such that for every world of the form $\langle k, w_0\rangle$ on this path, $\mathit{depth}(k) = (d+1)$ or $\mathit{depth}(k) = d$ and k is a left child iff i is a left child.

Now we can define the reduction f, and the satisfying model for $f(\phi)$. Let $M = \langle W, R_1, R_2, \pi\rangle$ be such that:

- $M, w \models p_t \Leftrightarrow w \in W_T \times \{w_0\}$,
- $M, \langle i, w_0\rangle \models (\mathit{depth} = d) \Leftrightarrow \mathit{depth}(i) = d$, where *depth* is a propositional vector as on page 16,
- $M, \langle i, w_0\rangle \models p_{\mathit{left}} \Leftrightarrow i$ is a left child,
- $M, \langle i, w_0\rangle \models p \Leftrightarrow M_T, i \models p$ for p in ϕ.

And define R_{succ} to simulate the edges of the tree in the obvious way: if $M, w \models \mathit{depth} = d$ and $M, w \models p_{\mathit{left}} \leftrightarrow \ell$ for $d \in \{0, \ldots, n\}$ and $\ell \in \{\top, \bot\}$, then $wR_{\mathit{succ}}w'$ iff

- $M, w' \models p_t \wedge \mathit{depth}(j) = d + 1$,
- there exists an R_σ path from w to w' such that for every world w'' on this path: $M, w'' \models \neg p_t$ or $M, w'' \models (\mathit{depth} = d+1)$ or $M, w'' \models (\mathit{depth} = d) \wedge (p_{\mathit{left}} \leftrightarrow \ell)$

It is immediate that j is a child of i iff $\langle i, w_0\rangle R_{\mathit{succ}} \langle j, w_0\rangle$. We define a modality corresponding to R_{succ} in the following way: for ψ a propositional formula, let $[\mathit{succ}]\psi$ be defined as follows:

$$\begin{array}{l}\displaystyle\bigwedge_{0\le d\le n,\ \ell\in\{\top,\bot\}} (\mathit{depth} = d) \wedge (p_{\mathit{left}} \leftrightarrow \ell) \rightarrow \\ \qquad [\sigma_1](\neg p_t \vee (\mathit{depth} = d+1) \vee ((\mathit{depth} = d) \wedge (p_{\mathit{left}} \leftrightarrow \ell)) \rightarrow \\ \qquad\quad [\sigma_2](\neg p_t \vee (\mathit{depth} = d+1) \vee ((\mathit{depth} = d) \wedge (p_{\mathit{left}} \leftrightarrow \ell)) \rightarrow \\ \qquad\qquad \cdots \\ \qquad\qquad\quad [\sigma_k](p_t \wedge (\mathit{depth} = d+1) \rightarrow \psi)) \cdots)\end{array}$$

From the definition of R_{succ}, it follows that $M, w \models [\mathit{succ}]\psi$ iff $\forall w'(wR_{\mathit{succ}}w' \Rightarrow \psi)$. Let $g(\psi)$ be the result of replacing every $\Box$ by $[\mathit{succ}]$. Then for all $i \in W_T$, and for all ψ of modal depth ≤ 1:

$$M_T, i \models \psi \Leftrightarrow M, \langle i, w_0\rangle \models g(\psi)$$

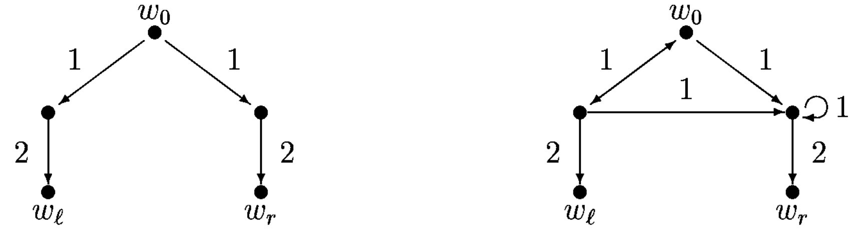
Finally, define the reduction f as follows:

$$f(\phi) = f(\phi_1 \wedge \bigwedge_{i=0}^{n} \Box^i \phi_2) = p_t \wedge g(\phi_1) \wedge \bigwedge_{i=0}^{n} [\sigma]^i (p_t \to g(\phi_2)).$$

It is immediate that if $M_T, 0 \models \phi$, then $M, \langle 0, w_0 \rangle \models f(\phi)$. Furthermore, f is polynomial time computable, since ϕ_1 and ϕ_2 are of modal depth at most 1.

It remains to show that if $f(\phi)$ is satisfiable, then so is ϕ. Suppose $M = \langle W, R_1, R_2, \pi \rangle$ is a model, $w_0 \in W$ such that $M, w_0 \models f(\phi)$. Define the corresponding uni-modal model as $\widehat{M} = \langle W, \widehat{R}, \pi \rangle$ such that $w\widehat{R}w'$ if and only if $wR_{succ}w'$. It is immediate that for all $w \in W$, for all formulas ψ of modal depth ≤ 1, $M, w \models \psi$ iff $\widehat{M}, w \models g(\psi)$, and therefore, $\widehat{M}, w_0 \models \phi$. □

In the next section, we will give a complete description of all different cases where theorem 3.3.2 is applicable. To avoid counting similar cases twice, note that the following two subframes behave in the same way according to theorem 3.3.2:



Note that the first frame can be obtained from the second frame by removing edges. Define the following notion:

Definition 3.3.3 *Let $F = \langle W, \{R_a\}_{a \in I} \rangle$ and $F' = \langle W', \{R_a\}_{a \in I} \rangle$ be two frames. We say that F is a* skeleton subframe *of F' if $W \subseteq W'$ and $R_a \subseteq R'_a$ for all $a \in I$.*

We will show that theorem 3.3.2 holds as well if we only require that $\widehat{F}$ is a skeleton subframe instead of a subframe. This theorem follows immediately from theorem 3.3.2 and the correspondence between skeleton subframes and subframes of $\mathcal{F}_1 \oplus \mathcal{F}_2$, $\mathcal{F}_1$ and $\mathcal{F}_2$. As in the case of subframes, if $\mathcal{F}_1$ and $\mathcal{F}_2$ are closed under disjoint union, then skeleton subframes of the join correspond to skeleton subframes of the uni-modal fragments in the obvious way: $\langle W, R_1, R_2 \rangle$ is a skeleton subframe of $\mathcal{F}_1 \oplus \mathcal{F}_2$ iff $\langle W, R_1 \rangle$ is a skeleton subframe of $\mathcal{F}_1$ and $\langle W, R_2 \rangle$ is a skeleton subframe of $\mathcal{F}_2$. The following lemma gives more correspondence between skeleton subframes and subframes. Again, we defer the proof to the last section.

Lemma 3.3.4 *Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two classes of uni-modal frames closed under disjoint union.*

- *If $\langle \widehat{W}, \widehat{R}_1, \widehat{R}_2 \rangle$ is a skeleton subframe of $\mathcal{F}_1 \oplus \mathcal{F}_2$ then $\langle \widehat{W}, \widehat{R}_1 \rangle$ is a skeleton subframe of $\mathcal{F}_1$ and $\langle \widehat{W}, \widehat{R}_2 \rangle$ is a skeleton subframe of $\mathcal{F}_2$.*
- *If $\langle \widehat{W}, \widehat{R}_1 \rangle$ is a skeleton subframe of $\mathcal{F}_1$ and $\langle \widehat{W}, \widehat{R}_2 \rangle$ is a skeleton subframe of $\mathcal{F}_2$, then there exists a subframe $F = \langle \widehat{W}, R_1, R_2 \rangle$ of $\mathcal{F}_1 \oplus \mathcal{F}_2$ such that $\widehat{R}_1 \subseteq R_1$, $\widehat{R}_2 \subseteq R_2$ and for all $w \in \widehat{W} : wR_a w' \Rightarrow w(\widehat{R}_a \cup \widehat{R}_a^{-1})^* w'$.*

Combining this lemma with theorem 3.3.2 gives the skeleton version of this theorem:

Theorem 3.3.5 *Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two classes of uni-modal frames closed under disjoint union. If there exists an $\mathcal{F}_1 \oplus \mathcal{F}_2$ skeleton subframe $\widehat{F} = \langle \widehat{W}, \widehat{R}_1, \widehat{R}_2 \rangle$ such that:*

1. *$\widehat{W} \supseteq \{w_0, w_\ell, w_r\}$, and for some $\sigma \in \{1,2\}^+$: $w_0 \widehat{R}_\sigma w_\ell$ and $w_0 \widehat{R}_\sigma w_r$,*
2. *w_0, w_ℓ and w_r have the same reflexive behavior, i.e. for $a = 1, 2$:*
$$w_0 \widehat{R}_a w_0 \Leftrightarrow w_\ell \widehat{R}_a w_\ell \Leftrightarrow w_r \widehat{R}_a w_r,$$
3. *For some $a \in \{1,2\}$:*
    - *w_0 has no non-reflexive $\widehat{R}_a$ edges, i.e. if $w_0 \widehat{R}_a w$ or $w \widehat{R}_a w_0$, then $w_0 = w$, and*
    - *w_ℓ and w_r have no non-reflexive $\widehat{R}_{\overline{a}}$ edges: if $w_\ell \widehat{R}_{\overline{a}} w$ or $w \widehat{R}_{\overline{a}} w_\ell$ then $w = w_\ell$, and if $w_r \widehat{R}_{\overline{a}} w$ or $w \widehat{R}_{\overline{a}} w_r$ then $w = w_r$.*

Then $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability is PSPACE-hard.

3.4 Classification

In section 3.2 we have given a criterion for the transfer of NP upper bounds, and in section 3.3 a criterion for PSPACE hardness of the join. In this section we prove that both these theorems are optimal in the following sense:

Theorem 3.4.1 *Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two classes of uni-modal frames closed under disjoint union. Then we are in one of the following three cases:*

I *For some $a \in \{1,2\}$, $\mathcal{F}_a$ is trivial. (That is, every frame in $\mathcal{F}_a$ consists of the disjoint union of singletons, or, in other words, •→• is not a skeleton subframe of $\mathcal{F}_a$). In this case, $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability is polynomial time reducible to $\mathcal{F}_{\overline{a}}$ satisfiability.*

II *$\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability is PSPACE-hard by theorem 3.3.5.*

III *$\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability is in NP by theorem 3.2.2.*

It follows that for two classes of non trivial frames, theorems 3.3.5 and 3.2.2 are optimal under the assumption that NP $\neq$ PSPACE. Furthermore, case II is the only case where the join can contribute to the complexity. The proof of the theorem is rather lengthy. We start with the simplest part, i.e. with case I.

Case I: Singletons

We will prove that if •→• is not a skeleton subframe of $\mathcal{F}_2$, then $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability is polynomial time reducible to $\mathcal{F}_1$ satisfiability.

If •→• is not a skeleton subframe of $\mathcal{F}_2$, then no world in an $\mathcal{F}_2$ frame has a successor different from itself, i.e. every frame in $\mathcal{F}_2$ consists of the disjoint union of singletons. Note that it might be the case that $\mathcal{F}_2$ is empty. If that is the case, then $\mathcal{F}_1 \oplus \mathcal{F}_2 = \emptyset$, and therefore no formula is satisfiable in $\mathcal{F}_1 \oplus \mathcal{F}_2$. Then $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability is trivially reducible to $\mathcal{F}_1$ satisfiability by the reduction $\lambda\phi.\bot$.

Now suppose that $\mathcal{F}_1, \mathcal{F}_2 \neq \emptyset$. There exist many different classes of frames that are closed under disjoint union and contain only singletons. For instance, $\mathcal{F}_2$ might consist of all frames that contain at least 5 reflexive singletons, and an even number of irreflexive singletons. However, it is easy to see that there are only three different cases for $\mathcal{F}_2$ satisfiability.

1. If all worlds in $\mathcal{F}_2$ are reflexive, then ϕ is $\mathcal{F}_2$ satisfiable iff ϕ is satisfiable on the reflexive singleton. (Logic: $K_2 + [2]p \leftrightarrow p$.)

2. If all worlds in $\mathcal{F}_2$ are irreflexive, then ϕ is $\mathcal{F}_2$ satisfiable iff ϕ is satisfiable on the irreflexive singleton. (Logic: $K_2 + [2]\bot$.)

3. If $\mathcal{F}_2$ contains reflexive and irreflexive worlds, then ϕ is $\mathcal{F}_2$ satisfiable iff ϕ is satisfiable on some singleton. (Logic: $K_2 + p \to [2]p$).

Recall that corollary 3.1.3 states that if $\mathcal{F}_2$ and $\widehat{\mathcal{F}}_2$ are closed under disjoint union and $\mathcal{F}_2$ satisfiability $=$ $\widehat{\mathcal{F}}_2$ satisfiability, then $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability $=$ $\mathcal{F}_1 \oplus \widehat{\mathcal{F}}_2$ satisfiability. From this observation, it follows that:

1. If all worlds in $\mathcal{F}_2$ are reflexive, then ϕ is satisfiable in $\mathcal{F}_1 \oplus \mathcal{F}_2$ iff ϕ is satisfiable on a frame $F = \langle W, R_1, R_2 \rangle$ such that $\langle W, R_1 \rangle \in \mathcal{F}_1$ and $R_2 = \{\langle w, w\rangle | w \in W\}$.

2. If all worlds in $\mathcal{F}_2$ are irreflexive, then ϕ is satisfiable in $\mathcal{F}_1 \oplus \mathcal{F}_2$ iff ϕ is satisfiable on a frame $F = \langle W, R_1, R_2 \rangle$ such that $\langle W, R_1 \rangle \in \mathcal{F}_1$ and $R_2 = \emptyset$.

3. If $\mathcal{F}_2$ contains reflexive and irreflexive worlds, then ϕ is satisfiable in $\mathcal{F}_1 \oplus \mathcal{F}_2$ iff ϕ is satisfiable on a frame $F = \langle W, R_1, R_2 \rangle$ such that $\langle W, R_1 \rangle \in \mathcal{F}_1$ and $R_2 \subseteq \{\langle w, w\rangle | w \in W\}$.

For each of these cases, we define a reduction from $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability to $\mathcal{F}_1$ satisfiability. First suppose that every world in $\mathcal{F}_2$ is reflexive. Define the following reduction f_r:

$$\begin{array}{lll} f_r(p) = p; & f_r(\neg\psi) = \neg f_r(\psi); & f_r(\psi_1 \wedge \psi_2) = f_r(\psi_1) \wedge f_r(\psi_2); \\ f_r([1]\psi) = [1]f_r(\psi); & f_r([2]\psi) = f_r(\psi). & \end{array}$$

It is easy to verify that ϕ is satisfiable on a frame $\langle W, R_1, \{\langle w, w\rangle | w \in W\}\rangle$ iff $f_r(\phi)$ is satisfiable on the frame $\langle W, R_1 \rangle$.

Now suppose that every world in $\mathcal{F}_2$ is irreflexive. Define the following reduction f_{ir}:

$$\begin{array}{lll} f_{ir}(p) = p; & f_{ir}(\neg\psi) = \neg f_{ir}(\psi); & f_{ir}(\psi_1 \wedge \psi_2) = f_{ir}(\psi_1) \wedge f_{ir}(\psi_2); \\ f_{ir}([1]\psi) = [1]f_{ir}(\psi); & f_{ir}([2]\psi) = \top. & \end{array}$$

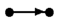
It is easy to verify that ϕ is satisfiable on a frame $\langle W, R_1, \emptyset\rangle$ iff $f_{ir}(\phi)$ is satisfiable on the frame $\langle W, R_1 \rangle$.

Finally, suppose that $\mathcal{F}_2$ contains reflexive and irreflexive worlds. In this case, we use a new propositional variable p_r to denote that a world is R_2 reflexive. Define the reduction f_{rir} as follows:

$$\begin{array}{lll} f_{rir}(p) = p; & f_{rir}(\neg\psi) = \neg f_{rir}(\psi); & f_{rir}(\psi_1 \wedge \psi_2) = f_{rir}(\psi_1) \wedge f_{rir}(\psi_2); \\ f_{rir}([1]\psi) = [1]f_{rir}(\psi); & & f_{rir}([2]\psi) = p_r \to f_{rir}(\psi). \end{array}$$

It is easy to verify that ϕ is satisfiable on a frame $\langle W, R_1, R_2 \rangle$ such that $R_2 \subseteq \{\langle w, w\rangle | w \in W\}$ iff $f_{rir}(\phi)$ is satisfiable in a model $\langle W, R_1, \pi\rangle$ such that $\pi(p_r) = \{w \in W | wR_2w\}$. □

Case II: PSPACE Hardness

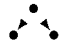
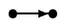
For the remainder of the proof of theorem 3.4.1, we have to show that if $\mathcal{F}_1$ and $\mathcal{F}_2$ are two classes of uni-modal frames that both contain  as skeleton subframe, then we are in one of the following two cases:

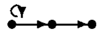
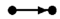
II $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability is PSPACE-hard by theorem 3.3.5.

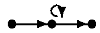
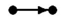
III $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability is in NP by theorem 3.2.2.

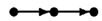
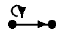
During the proof, we also obtain a complete classification of the $\mathcal{F}_1$ and $\mathcal{F}_2$ subframes that occur in case II.

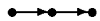
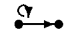
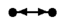
Theorem 3.4.2 *Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two classes of uni-modal frames closed under disjoint union. Then theorem 3.3.5 can be applied (i.e. we are in case II of theorem 3.4.1) iff we are in one of the following six cases for some $a \in \{1, 2\}$:*

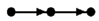
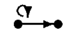
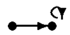
A  *is a skeleton subframe of $\mathcal{F}_a$ and*  *is a skeleton subframe of $\mathcal{F}_{\overline{a}}$.*

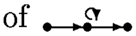
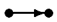
B  *is a skeleton subframe of $\mathcal{F}_a$ and*  *is a skeleton subframe of $\mathcal{F}_{\overline{a}}$.*

C  *is a skeleton subframe of $\mathcal{F}_a$ and*  *is a skeleton subframe of $\mathcal{F}_{\overline{a}}$.*

D  *is a skeleton subframe of $\mathcal{F}_a$ and*  *is a skeleton subframe of $\mathcal{F}_{\overline{a}}$.*

E  *and*  *are skeleton subframes of $\mathcal{F}_a$ and*  *is a skeleton subframe of $\mathcal{F}_{\overline{a}}$.*

F  *and*  *are skeleton subframes of $\mathcal{F}_a$ and*  *is a skeleton subframe of $\mathcal{F}_{\overline{a}}$.*

Note that none of the six cases is included in the five other cases, even if we look only at logics with satisfiability problems in NP (i.e. cases where the join really contributes to the complexity). For instance, if $\mathcal{F}_1$ consists of the closure under disjoint union of  and $\mathcal{F}_2$ of the closure under disjoint union of , then we are in case C, but not in any of the other cases, and both satisfiability problems are in NP.

Our two theorems will be proved simultaneously in the following way. First we show that theorem 3.3.5 can be applied in all six cases of theorem 3.4.2. Then we show that if $\mathcal{F}_1$ and $\mathcal{F}_2$ are two non-trivial classes of frames, and we are not in case A through F as given above, then $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability is in NP, by theorem 3.2.2. This proves both theorems, since theorem 3.3.5 and theorem 3.2.2 can never be applied at the same time.

We first show that if we are in case A through F, then the satisfiability problem for the join is PSPACE-hard by theorem 3.3.5. For each of the six cases, we give an $\mathcal{F}_1 \oplus \mathcal{F}_2$ skeleton subframe $\widehat{F} = \langle \widehat{W}, \widehat{R}_1, \widehat{R}_2 \rangle$ and a string $\sigma \in \{1, 2\}^+$ such that the conditions of theorem 3.3.5 are fulfilled, that is:

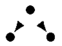
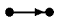
1. $\widehat{W} \supseteq \{w_0, w_\ell, w_r\}$, and for some $\sigma \in \{1, 2\}^+$: $w_0 \widehat{R}_\sigma w_\ell$ and $w_0 \widehat{R}_\sigma w_r$,

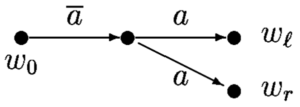
2. w_0, w_ℓ and w_r have the same reflexive behavior, i.e. for $a = 1, 2$:

$$w_0 \widehat{R}_a w_0 \Leftrightarrow w_\ell \widehat{R}_a w_\ell \Leftrightarrow w_r \widehat{R}_a w_r,$$

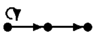
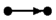
3. For some $a \in \{1, 2\}$:

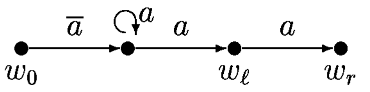
   - w_0 has no non-reflexive $\widehat{R}_a$ edges, i.e. if $w_0 \widehat{R}_a w$ or $w \widehat{R}_a w_0$, then $w_0 = w$, and
   - w_ℓ and w_r have no non-reflexive $\widehat{R}_{\overline{a}}$ edges: if $w_\ell \widehat{R}_{\overline{a}} w$ or $w \widehat{R}_{\overline{a}} w_\ell$ then $w = w_\ell$, and if $w_r \widehat{R}_{\overline{a}} w$ or $w \widehat{R}_{\overline{a}} w_r$ then $w = w_r$.

A  is a skeleton subframe of $\mathcal{F}_a$ and  is a skeleton subframe of $\mathcal{F}_{\overline{a}}$.

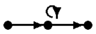
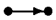


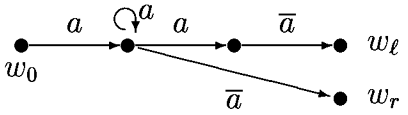
$\sigma = \overline{a}a$

B  is a skeleton subframe of $\mathcal{F}_a$ and  is a skeleton subframe of $\mathcal{F}_{\overline{a}}$.

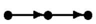
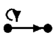


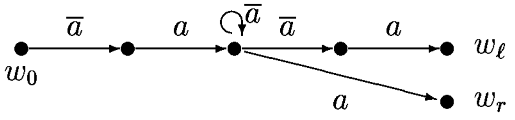
$\sigma = \overline{a}aa$

C  is a skeleton subframe of $\mathcal{F}_a$ and  is a skeleton subframe of $\mathcal{F}_{\overline{a}}$.

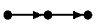
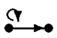
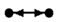


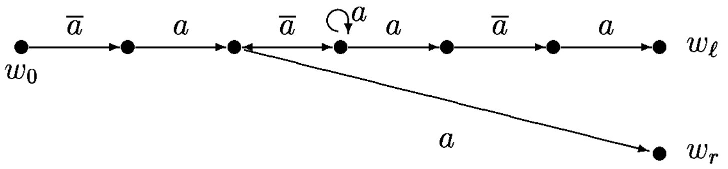
$\sigma = aa\overline{a}$

D  is a skeleton subframe of $\mathcal{F}_a$ and  is a skeleton subframe of $\mathcal{F}_{\overline{a}}$.

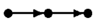
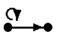
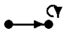


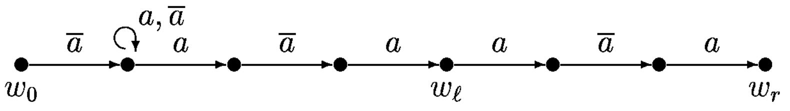
$\sigma = \overline{a}a\overline{a}a$

E  and  are skeleton subframes of $\mathcal{F}_a$ and  is a skeleton subframe of $\mathcal{F}_{\overline{a}}$.



$\sigma = \overline{a}a\overline{a}a\overline{a}a$

F  and  are skeleton subframes of $\mathcal{F}_a$ and  is a skeleton subframe of $\mathcal{F}_{\overline{a}}$.



$\sigma = \overline{a}a\overline{a}aa\overline{a}a$

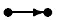
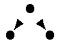
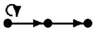
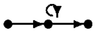
□

Case III: NP Upper Bounds

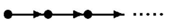
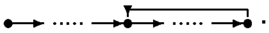
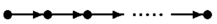
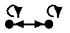
Finally, we show that if $\mathcal{F}_1$ and $\mathcal{F}_2$ are two non-trivial classes of frames, and we are not in case A through F of theorem 3.4.2, then $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability is in NP, by theorem 3.2.2. This completes the proofs of theorems 3.4.1 and 3.4.2. To conclude that $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiability is in NP by theorem 3.2.2, we need to show the following two requirements:

- $\mathcal{F}_1$ and $\mathcal{F}_2$ satisfiability are in NP, and

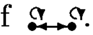
- there exists a polynomial p such that for every $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiable formula ϕ, there exists a model $M = \langle W, R_1, R_2, \pi\rangle$ such that:
  - $\langle W, R_1, R_2\rangle \in \mathcal{F}_1 \oplus \mathcal{F}_2$,
  - $M, w_0 \models \phi$,
  - $|\{w | w_0 R_{paths(\phi)} w\}| \leq p(|\phi|)$.

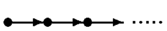
The first requirement Since $\mathcal{F}_1$ and $\mathcal{F}_2$ both contain  as a skeleton subframe, and we are not in cases A,B and C of theorem 3.4.2, it follows that , , and  are not skeleton subframes of $\mathcal{F}_1$ and $\mathcal{F}_2$. Let $\mathcal{F}$ be a class of frames that does not contain these three frames as skeleton subframes. We will show that $\mathcal{F}$ satisfiability is in NP.

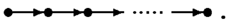
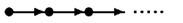
Let *Rt*($\mathcal{F}$) be the class of rooted generated subframes of $\mathcal{F}$. It is easy to see that every frame in *Rt*($\mathcal{F}$) is of one of the following forms:

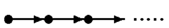
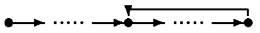
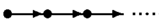
1. The infinite frame of the form:  ,
2. A finite frame of the form: . (The length of the cycle may be one, in which case the frame ends with a reflexive world).
3. A finite frame of the form: , or
4. A skeleton subframe of the frame: 

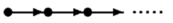
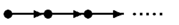
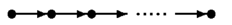
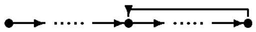
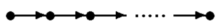
We'd like to use the following algorithm for $\mathcal{F}$ satisfiability: guess a model $M = \langle W, R, \pi\rangle$ of size at most $|\phi|$ and a world w_0, verify that $M, w_0 \models \phi$ and that there exists a frame $F \in \mathcal{F}$ such that $F|\{w|w_0 R_{paths(\phi)} w\} = \langle W, R\rangle$. Unfortunately, verifying the last condition can be of arbitrary complexity. For suppose A is a subset of the natural numbers. Define $\mathcal{F}$ in such a way that a cyclic frame F is a member of $\mathcal{F}$ iff the length of the cycle is in A. Then A is reducible to verification of the last condition for $\mathcal{F}$.

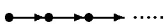
We will show that the satisfiability problem for $\mathcal{F}$ is in NP, by carefully analyzing the different cases that can occur. First of all, let Y be the set of frames in *Rt*($\mathcal{F}$) of type 4., i.e. skeleton subframes of . Since Y is a finite set of finite frames, Y satisfiability is in NP. It follows that *Rt*($\mathcal{F}$) satisfiability is in NP iff *Rt*($\mathcal{F}$) $\setminus Y$ satisfiability is in NP. We will therefore assume that *Rt*($\mathcal{F}$) contains only frames of types 1, 2 and 3. We distinguish the following cases:

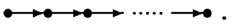
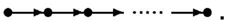
- *Rt*($\mathcal{F}$) is finite. In this case, either all frames are finite, in which case the corresponding satisfiability problem is in NP, or *Rt*($\mathcal{F}$) consists of a finite class of finite frames and the only infinite frame  . But in this case, the satisfiability problem is in NP as well.

- *Rt*($\mathcal{F}$) is infinite.
  - *Rt*($\mathcal{F}$) contains a finite class X of frames of the form . Then *Rt*($\mathcal{F}$) satisfiability $= X \cup \{$  $\}$ satisfiability.

    First suppose that ϕ is satisfiable in *Rt*($\mathcal{F}$). Then either ϕ is satisfiable in X, in which case we are done, or ϕ is satisfiable on a frame of the form  or  . But then ϕ is also satisfiable on the corresponding unraveled frame. In all cases, the unraveled frame is  .

For the converse, suppose that ϕ is satisfiable on $X \cup \{$  $\}$. Then either ϕ is satisfiable in X, in which case we are done, or ϕ is satisfiable on  , and therefore satisfiable on the frame  of $|\phi|$ worlds. Since $Rt(\mathcal{F})$ is infinite and X is finite, $Rt(\mathcal{F})$ contains an infinite number of frames of the form . But then there exists there exists a frame $G \in Rt(\mathcal{F})$ and world w_0 such that $G|\{w|w_0R^iw$ for $i \leq |\phi| - 1\}$ is equal the frame  of $|\phi|$ worlds. It follows that G satisfies ϕ at w_0.

Since X is a finite class of finite frames, the satisfiability problem for $X \cup \{$  $\}$ is in NP.

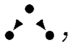
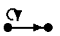
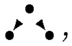
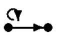
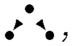
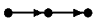
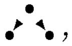
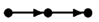
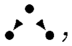
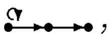
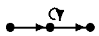
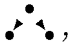
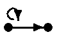
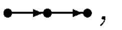
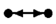
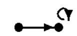
- $Rt(\mathcal{F})$ contains an infinite class of frames of the form . Using a similar argument as in the previous case, it can be shown that $Rt(\mathcal{F})$ satisfiability = satisfiability with respect to the class of all frames of the form . The satisfiability problem for this class of frames is in NP.

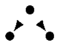
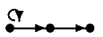
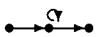
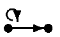
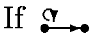
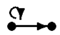
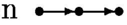
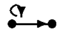
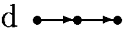
The second requirement Finally, we show that if $\mathcal{F}_1$ and $\mathcal{F}_2$ are two non-trivial classes of frames, and we are not in case A through F of theorem 3.4.2, then the second requirement of theorem 3.2.2 is fulfilled, i.e. we show that there exists a polynomial p such that for every $\mathcal{F}_1 \oplus \mathcal{F}_2$ satisfiable formula ϕ, there exists a model $M = \langle W, R_1, R_2, \pi\rangle$ such that:

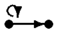
- $\langle W, R_1, R_2\rangle \in \mathcal{F}_1 \oplus \mathcal{F}_2$,
- $M, w_0 \models \phi$,
- $|\{w|w_0R_{paths(\phi)}w\}| \leq p(|\phi|)$.

We start by describing the form of the frames in $\mathcal{F}_1$ and $\mathcal{F}_2$. It turns out that there are three cases to consider:

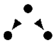
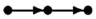
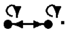
Lemma 3.4.3 *Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two classes of uni-modal frames closed under disjoint union. If we are not in case I and A through F, then for some $a \in \{1, 2\}$, we are in one of the following three cases:*

	not skeleton subframe of $\mathcal{F}_a$	*not skeleton subframe of $\mathcal{F}_{\overline{a}}$*
G	 , 	 , 
H	 , 	 , 
J	 ,  , 	 ,  ,  ,  , 

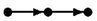
In the proof of the first requirement, we have seen that  ,  , and  are not skeleton subframes of $\mathcal{F}_1$ nor of $\mathcal{F}_2$. The remaining cases depend on the occurrence of  as a skeleton subframe. If  is not a skeleton subframe of $\mathcal{F}_1$ and not of $\mathcal{F}_2$, then we are in case G. If both $\mathcal{F}_1$ and $\mathcal{F}_2$ contain  as a skeleton subframe, then  is not a skeleton subframe of $\mathcal{F}_1$ nor of $\mathcal{F}_2$, and we are in case H. Finally, if  and  are skeleton subframes of $\mathcal{F}_1$, then we are in case J.

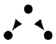
Case G Suppose and  are not skeleton subframes of $\mathcal{F}_1$ and $\mathcal{F}_2$. We will show that for any frame $F = \langle W, R_1, R_2\rangle \in \mathcal{F}_1 \oplus \mathcal{F}_2$, world $w_0 \in W$, and formula ϕ, $|\{w|w_0 R_{paths(\phi)} w\}| \leq |\phi|$.

Let $F = \langle W, R_1, R_2\rangle$ be a frame in $\mathcal{F}_1 \oplus \mathcal{F}_2$, and let $w_0 \in W$. For $\sigma \in \{1,2\}^*$, let *worlds*(σ) be the number of worlds w such that $w_0 R_\sigma w$. Then *worlds*$(\sigma 1) \leq 1$ and *worlds*$(\sigma 2) \leq 1$, since every world in F has at most one R_1 successor and at most one R_2 successor. From this it follows immediately that for every σ, *worlds*$(\sigma) \leq 1$, and therefore, for any formula ϕ, the set $\{w|w_0 R_{paths(\phi)} w\}$ is of size at most $|\phi|$.

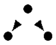
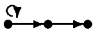
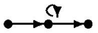
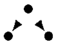
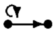
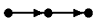
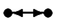
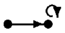
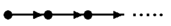
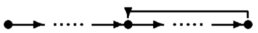
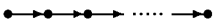
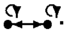
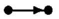
Case H Suppose  and  are not skeleton subframes of $\mathcal{F}_1$ and $\mathcal{F}_2$. Then all rooted generated subframes of $\mathcal{F}_1$ and $\mathcal{F}_2$ are skeleton subframes of .

Let $F = \langle W, R_1, R_2\rangle$ be a frame in $\mathcal{F}_1 \oplus \mathcal{F}_2$, let w_0 be a world in W, and let ϕ be a formula. We will show that $|\{w|w_0 R_{paths(\phi)} w\}| \leq |\phi|$. For strings σ, τ, we say that τ is contained in σ iff there exist $i_1, \ldots, i_k$ such that $1 \leq i_1 < i_2 < \cdots < i_k \leq |\sigma|$ and $\tau = \sigma_{i_1}\sigma_{i_2}\cdots\sigma_{i_k}$.

- If $w_0 R_\sigma w$ then there exists a τ contained in σ such that w is reachable from w_0 by an acyclic τ path. Since  is not a skeleton subframe of $\mathcal{F}_1$ nor of $\mathcal{F}_2$, it follows that τ is alternating.

- If τ is alternating and w and w' are reachable from w_0 by an acyclic τ path, then $w = w'$, since  is not a skeleton subframe of $\mathcal{F}_1$ nor of $\mathcal{F}_2$.

From these two observations, it follows that $|\{w|w_0 R_{paths(\phi)} w\}| \leq |\{\tau|\tau$ is an alternating sequence contained in some $\sigma \in paths(\phi)\}| \leq |\phi|$. □

Case J Suppose ,  and  are not skeleton subframes of $\mathcal{F}_1$, and , , ,  and are not skeleton subframes of $\mathcal{F}_2$. As we have seen in the proof of the first criterion, any frame in $Rt(\mathcal{F}_1)$ is of the form:  ,   or a skeleton subframe of . It is easy to see that any frame in $Rt(\mathcal{F}_2)$ is a singleton, or the frame .

Let $F = \langle W, R_1, R_2\rangle$ be a frame in $\mathcal{F}_1 \oplus \mathcal{F}_2$, and let $w_0 \in W$. We will show that $|\{w|w_0 R_{paths(\phi)} w\}| \leq 2|\phi|^2$.[2] For v a proper successor of w_0, let W_v consist of all the worlds in W that are reachable from v by a path that does not contain w_0. If $w_0 R_\sigma w$ and $w \neq w_0$, then for some real w_0 successor v, $w \in W_v$ and $v R_{\sigma'} w$ for σ' a suffix of σ. The following lemma shows that there exist at most two worlds in W_v that are reachable from v by σ'. Since w_0 has at most two real successors, it follows that $|\{w|w_0 R_{paths(\phi)} w\}| \leq 2|\{\sigma'|\sigma'$ is a suffix of some $\sigma \in paths(\phi)\}| + 1 \leq 2|\phi|^2$.

Lemma 3.4.4 *Let v be real successor of w_0. For every $w \in W_v$, $\sigma \in \{1,2\}^+$, $|\{w|v R_\sigma w\}| \leq 2$. Furthermore, if $v R_\sigma w_1$ and $v R_\sigma w_2$ and $w_1 \neq w_2$ then $w_1 R_1 w_1 R_1 w_2$ or $w_2 R_1 w_2 R_1 w_1$.*

Use induction on the length of σ. The claim trivially holds for $\sigma = \lambda$, since if $v R_\lambda w$ then $w = v$. Now assume that the claim holds for σ. We will prove the claim for $\sigma 1$ and $\sigma 2$. If no world is σ reachable from v, then there are certainly no worlds reachable by σa

[2] With a bit of extra work, we can prove a linear upper bound.

from v. Next assume that there is exactly one world w such that $vR_\sigma w$. Since w has at most one 2 successor, v has at most one $\sigma 2$ successor. Now look at the 1 successors of w. w can't have two 2 successors different from itself. Therefore, there exist at most two worlds reachable by $\sigma 1$ from v. Furthermore, if there exist two different worlds w_1 and w_2 reachable from v, then wR_1w_1 and wR_1w_2. It follows that $w = w_1$ or $w = w_2$ and therefore either $w_1R_1w_1R_1w_2$ or $w_2R_1w_2R_1w_1$.

Finally, suppose there exist exactly two worlds w_1 and w_2 that are reachable from v by R_σ. By the induction hypothesis, assume that $w_1R_1w_1R_1w_2$. It is easy to see that w_1 and w_2 do not have an R_1 successor different from w_1 and w_2, which proves the claim for $\sigma 1$.

For $\sigma 2$, we will prove that at least one of the worlds w_1 and w_2 has a real R_2 predecessor. From this, it follows that at most one of the worlds has an R_2 successor, and this successor is unique. Since $vR_\sigma w_1$, and v has by definition a real predecessor, w_1 has a real predecessor, say w_1'. If this is a 2 predecessor, then we are done, so suppose that $w_1'R_1w_1$. Then, $w_1'R_1w_1R_1w_1R_1w_2$. It follows that $w_1' = w_2$, and $w_1R_1w_2R_1w_1$. If both w_1 and w_2 are not equal to v, then there exists a worlds $v'' \neq w_1, w_2$ such that $v''R_aw_1$ or $v''R_aw_2$. But then $a = 2$, and at least one of the worlds w_1 and w_2 has a real 2 successor as required. Finally, if one of the worlds, say w_1, is equal to v, then either w_0R_2v, in which case we are done, or w_0R_1v has a real 1 successor v' such that not vR_1v'. Since $w_1, w_2 \in W_v$, it follows that $w_1, w_2 \neq w_0$. But this contradicts the form of $\mathcal{F}_1$ frames. □

3.5 The Complexity of the General Join

In the previous sections, we have investigated the relationship between the complexity of the satisfiability problem for the join of two uni-modal logics and the complexity of the satisfiability problems for these uni-modal fragments. The use of the join in the literature however, is not restricted to this simple case. In particular, we see the occurrence of infinite joins, for instance in PDL which is a conservative extension of the infinite join of K logics, and the join of multi-modal logics, as in logics for distributed systems which can be viewed as the join of a logic modeling discrete time with a logic modeling a multi-processor system.

In general, we consider joins of an arbitrary number of arbitrary multi-modal logics. As shown in [FS], these joins inherit many properties from their fragments as well. Following [FS], let Ω be a set of pairwise disjoint nonempty sets of indices $I, J, K, \ldots$. We assume that $\cup\Omega$ is countable. For normal logics $\{L_I\}_{I\in\Omega}$, let the join $\bigoplus_{I\in\Omega} L_I$ be the minimal normal logic containing L_I for all $I \in \Omega$. We say that a property *generally transfers* if $\{L_I\}_{I\in\Omega}$ has this property whenever L_I has this property for all $I \in \Omega$.

Theorem 3.5.1 ([FS]) *Weak completeness and strong completeness generally transfer.*

As before, we will look at the satisfiability problem with respect to a set of frames. For $\{\mathcal{F}_I\}_{I\in\Omega}$ sets of frames, the join of $\mathcal{F}_I$, denoted by $\bigoplus_{I\in\Omega} \mathcal{F}_I$, consists of the frames $\langle W, \{R_I\}_{I\in\Omega}\rangle$ such that for all $I \in \Omega$, $\langle W, R_I\rangle = \langle W, \{R_a\}_{a\in I}\rangle \in \mathcal{F}_I$. General joins of logics and general joins of frames are related as would be expected:

Theorem 3.5.2 ([FS]) *If for all $I \in \Omega$, $\mathcal{F}_I$ is closed under disjoint union, and L_I is complete with respect to $\mathcal{F}_I$, then $\bigoplus_{I\in\Omega} L_I$ is complete with respect to $\bigoplus_{I\in\Omega} \mathcal{F}_I$.*

In this section, we investigate what effect the general join has on the complexity of the satisfiability problem. In the previous sections, we measured the effect of the join of two uni-modal logics by comparing the complexity of the satisfiability problem of the join to the complexity of the satisfiability problems of its two uni-modal fragments. In particular, we said that the join contributed to the complexity if the satisfiability problem for the join was more complex than the satisfiability problems of its uni-modal fragments. However, when we consider the join of infinitely many logics, this definition does not capture our intuition. The problem is that the choice of Ω has an impact on the complexity. In fact, as pointed out in [FS], decidability does not generally transfer. Consider for instance the following example: Let A be an arbitrary subset of $\mathbf{N}$, let $\Omega = \{A, \mathbf{N} \setminus A\}$, and let $\mathcal{F}_A$ be the closure under disjoint union of the reflexive singleton and $\mathcal{F}_{\mathbf{N}\setminus A}$ the closure under disjoint union of the irreflexive singleton. Then the corresponding satisfiability problems are in NP, but A is reducible to $\mathcal{F}_A \oplus \mathcal{F}_{\mathbf{N}\setminus A}$, by $\lambda i.\langle i \rangle \top$.

But this is not the only problem. Even when it is easy to detect to which sublanguage a formula belongs, we can still get unlimited boosts in the complexity. Again, let A be an arbitrary subset of $\mathbf{N}$, and let $\Omega = \{\{i\} | i \in \mathbf{N}\}$. For all $i \in \mathbf{N}$, let $\mathcal{F}_i$ consist of the closure under disjoint union of the reflexive singleton if $i \in A$, and of the closure under disjoint union of the irreflexive singleton if $i \notin A$. Obviously, for all $i \in \mathbf{N}$, $\mathcal{F}_i$ satisfiability is in NP. Furthermore, every frame in $\bigoplus_{i \in \mathbf{N}} \mathcal{F}_i$ consists of the disjoint union of singletons. In this sense, the join is trivial, but again, A is reducible to $\bigoplus_{i \in \mathbf{N}} \mathcal{F}_i$ satisfiability, by $\lambda i.\langle i \rangle \top$.

There are two ways to avoid the problems mentioned above. If we don't want to restrict the choice of Ω, these two examples make it clear that it is not fair to measure the effect of the join by comparing the satisfiability problem for the join with the satisfiability problems of the joinees. The problem is how to separate the effect of the choice of Ω from the effect of the join. Note that in both examples above, the boost in complexity as caused by the choice of Ω is already apparent in the complexity of the union of the satisfiability problems of the joinees, in the sense that in both cases the constructed set A is reducible to the set $\bigcup_{I \in \Omega}(\mathcal{F}_I$ satisfiability). We can therefore say that the join contributes to the complexity if the complexity of the join is higher than the complexity of the union of the satisfiability problems. Note that in the case that we consider the join of finitely many sets, this definition corresponds to taking the supremum of the complexity of the joinees, and thus this definition is consistent with the one used in the previous section.

This solution has the advantage that it is very general. However, it leads to rather awkward formulations of the theorems, since the results are relative to the set $\bigcup_{I \in \Omega}(\mathcal{F}_I$ satisfiability). The approach we will take is the following: we will restrict the choice of Ω and the classes of frames $\{\mathcal{F}_I\}_{I \in \Omega}$ in such a way that this choice does not contribute to the complexity. We want these restrictions to be reasonable, in the sense that the logics encountered in the literature still fit this framework. Our first restriction on Ω ensures that we don't have problems with recognizing the sublanguages involved, thereby avoiding the first problem mentioned above. This is straightforward; we will only look at sets Ω such that for all $I \in \Omega$, $I \in$ P. The second problem sketched above can informally be stated as follows: given I, determining $\mathcal{F}_I$ does not contribute to the complexity. Note that this problem only occurs when Ω is infinite. We will ensure that there exists a finite number of classes of frames such that for every $I \in \Omega$, $\mathcal{F}_I$ satisfiability is isomorphic to the satisfiability problem with respect to one of these classes, and that these isomorphisms can be computed in polynomial time. An additional advantage of this requirement is that it ensures that the infinite union of for instance NP logics is itself NP.

Finally, we exclude undesirable behavior of the joinees. For it is still possible that $\mathbf{N} \in \Omega$, $\mathcal{F}_{\mathbf{N}}$ consists of the disjoint union of singletons, but $\mathcal{F}_{\mathbf{N}}$ satisfiability is undecidable: just let $\mathcal{F}_{\mathbf{N}} = \langle \{w\}, \{R_i\}_{i \in \mathbf{N}} \rangle$ such that $R_i = \emptyset$ iff $i \in A$ for some undecidable set A. This problem can be avoided by ensuring that trivial logics contain only a finite number of relations.

Formalizing the above, we obtain the following:

Definition 3.5.3 *Let Ω be a set of pairwise disjoint sets of indices, and for every $I \in \Omega$, let $\mathcal{F}_I$ be a class of frames. We call $\{\mathcal{F}_I\}_{I \in \Omega}$* well-behaved *if*

- *for all I, $\mathcal{F}_I$ is non-empty and closed under disjoint union,*
- *for all $I \in \Omega$: $I \in P$,*
- *there exist $I_1, \ldots I_k \in \Omega$ and a polynomial time computable function f from $\bigcup \Omega$ to $I_1 \cup \ldots \cup I_k$ such that for all $I \in \Omega$, $\mathcal{F}_I$ satisfiability is isomorphic to $\mathcal{F}_{I_j}$ satisfiability by f, and*
- *if there exists an upper bound on the size of rooted subframes of $\mathcal{F}_I$ then I is finite.*

Under these conditions, most of the theorems of the previous sections go through. In particular, the upper bound transfer theorems from section 3.2 still hold if we replace "transfer" by "generally transfer." The PSPACE hardness criterion from section 3.3 has a direct analog for the general join:

Theorem 3.5.4 *Let Ω be a set of pairwise disjoint sets of indices, and for every $I \in \Omega$, let $\mathcal{F}_I$ be a non-empty class of frames, closed under disjoint union. If there exist $I, J \in \Omega$, and a $\mathcal{F}_1 \oplus \mathcal{F}_2$ skeleton subframe $\widehat{F} = \langle \widehat{W}, \{\widehat{R}_I\}_{I \in \Omega} \rangle$ such that:*

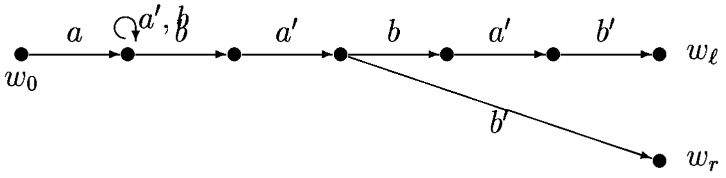
1. *$\widehat{W} \supseteq \{w_0, w_\ell, w_r\}$, and for some $\sigma \in (\cup\Omega)^+$: $w_0 \widehat{R}_\sigma w_\ell$ and $w_0 \widehat{R}_\sigma w_r$,*
2. *w_0, w_ℓ and w_r have the same reflexive behavior, i.e. for $a = \cup\Omega$:*
$$w_0 \widehat{R}_a w_0 \Leftrightarrow w_\ell \widehat{R}_a w_\ell \Leftrightarrow w_r \widehat{R}_a w_r,$$
3. *w_0 has no non-reflexive $\widehat{R}_I$ edges, i.e. if $w_0 \widehat{R}_a w$ or $w \widehat{R}_a w_0$ for some $a \in I$, then $w_0 = w$, and*
4. *w_ℓ and w_r have no non-reflexive $\widehat{R}_J$ edges: if $w_\ell \widehat{R}_a w$ or $w \widehat{R}_a w_\ell$ for some $a \in J$, then $w = w_\ell$, and if $w_r \widehat{R}_a w$ or $w \widehat{R}_a w_r$ for some $a \in J$, then $w = w_r$.*

Then $\bigoplus_{I \in \Omega} \mathcal{F}_I$ satisfiability is PSPACE-hard.

Classification

We now turn to the analog of theorem 3.4.1. We will prove the analog for the general join of uni-modal logics. There is no reason why there shouldn't exist such a theorem for multi-modal logics. However, there are very many cases which force PSPACE hardness, and we haven't managed to obtain a complete characterization. Recall that such a complete characterization was essential in the proof of theorem 3.4.1. To see why the situation is more complex if we consider the join of multi-modal logics, look at the following example:

$\Omega = \{\{a, a'\}, \{b, b'\}\}$, and $\bullet\xrightarrow{a}\bullet\circlearrowleft^{a'}$and $\bullet\xrightarrow{a'}\bullet$ are skeleton subframes of $\mathcal{F}_{\{a,a'\}}$ and $\circlearrowleft^{b}\bullet\longrightarrow\bullet$ and $\bullet\xrightarrow{b}\bullet\xleftarrow{b'}\bullet$ are skeleton subframes of $\mathcal{F}_{\{b,b'\}}$. Then $\mathcal{F}_{\{a,a'\}} \oplus \mathcal{F}_{\{b,b'\}}$ is PSPACE-hard by theorem 3.5.4, using the following skeleton subframe.



$\sigma = aba'ba'b'$

However, we have been able to prove the analog of theorem 3.4.1 for the arbitrary join of uni-modal logics:

Theorem 3.5.5 *Let Ω be a countable set of indices and for all $a \in \Omega$ let $\mathcal{F}_a$ be a set of frames such that $\{\mathcal{F}_a\}_{a\in\Omega}$ is well-behaved in the sense of definition 3.5.3. Then we are in one of the following three cases:*

I *There exists an index $a \in \Omega$ such that $\bullet\!\!\longrightarrow\!\!\bullet$ is not a skeleton subframe of $\mathcal{F}_b$ for all $b \neq a$. In this case, $\bigoplus_{a\in\Omega} \mathcal{F}_a$ satisfiability is polynomial time reducible to $\mathcal{F}_a$ satisfiability.*

II *$\bigoplus_{a\in\Omega} \mathcal{F}_a$ satisfiability is PSPACE-hard, by theorem 3.5.4.*

III *$\bigoplus_{a\in\Omega} \mathcal{F}_a$ satisfiability is in NP, by the general analog of theorem 3.2.2.*

Case I We first show how to get rid of uni-modal fragments that consist of the disjoint union of singletons. In the previous section, we saw that one set of frames that did not contain $\bullet\!\!\longrightarrow\!\!\bullet$ as a skeleton subframe didn't have any impact on the complexity of the join. The following lemma shows that this observation also holds for an infinite number of sets of frames with this property.

Lemma 3.5.6 *Let Ω be a countable set of indices, and let Ω' be a finite subset of Ω such that for all $a \in \Omega \setminus \Omega'$: $\bullet\!\!\longrightarrow\!\!\bullet$ is not a skeleton subframe of $\mathcal{F}_a$. Then $\bigoplus_{a\in\Omega} \mathcal{F}_a$ satisfiability is polynomial time reducible to $\bigoplus_{a\in\Omega'} \mathcal{F}_a$ satisfiability.*

The proof is a generalization of the proof of case G in the previous section. Let $b \in \Omega \setminus \Omega'$. Then $\mathcal{F}_b$ does not contain $\bullet\!\!\longrightarrow\!\!\bullet$ as a skeleton subframe, and therefore $\mathcal{F}_b$ consists of the disjoint union of singletons. It follows that there are only three cases for $\mathcal{F}_b$ satisfiability, depending on the occurrence of reflexive and irreflexive worlds. Using corollary 3.1.3, it follows that ϕ is $\bigoplus_{a\in\Omega} \mathcal{F}_a$ satisfiable, iff ϕ is satisfiable on a frame $F = \langle W, \{R_a\}_{a\in\Omega}\rangle$ such that $\langle W, \{R_a\}_{a\in\Omega'}\rangle \in \bigoplus_{a\in\Omega'} \mathcal{F}_a$, and for all $a \in \Omega \setminus \Omega'$:

$$\begin{array}{lcl} R_a = & \{\langle w, w\rangle | w \in W\} & \text{if every world in } \mathcal{F}_a \text{ is reflexive} \\ R_a = & \emptyset & \text{if every world in } \mathcal{F}_a \text{ is irreflexive} \\ R_a \subseteq & \{\langle w, w\rangle | w \in W\} & \text{if } \mathcal{F}_a \text{ contains both reflexive and irreflexive worlds} \end{array}$$

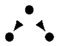
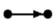
Now define the reduction f as follows, using a propositional variable r_a to denote that a world is a reflexive:

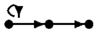
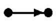
$$f(p) = p; f(\neg\psi) = \neg f(\psi); f(\psi_1 \wedge \psi_2) = f(\psi_1) \wedge f(\psi_2);$$
$$f([a]\psi) = \begin{cases} [a]f(\psi) & \text{if } a \in \Omega' \\ f(\psi) & \text{if } a \notin \Omega' \text{ and reflexive} \\ \top & \text{if } a \notin \Omega' \text{ and irreflexive} \\ r_a \to f(\psi) & \text{if } a \notin \Omega' \text{ and both reflexive and irreflexive} \end{cases}$$

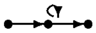
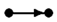
This lemma immediately implies case I. □

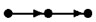
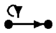
Case II For the remainder of the proof, we use the following complete classification of situations where theorem 3.5.4 can be applied.

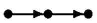
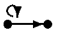
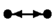
Theorem 3.5.7 *Let Ω be a countable set of indices and for all $a \in \Omega$ let $\mathcal{F}_a$ be a set of frames such that $\{\mathcal{F}_a\}_{a\in\Omega}$ is well-behaved in the sense of definition 3.5.3. Then $\bigoplus_{a\in\Omega} \mathcal{F}_a$ satisfiability is PSPACE-hard by theorem 3.5.4 iff there exist $a, b \in \Omega$ such that $a \neq b$ and $\mathcal{F}_a \oplus \mathcal{F}_b$ is in one of the six cases of theorem 3.4.2, i.e.*

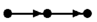
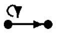
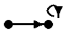
A  *is a skeleton subframe of $\mathcal{F}_a$ and*  *is a skeleton subframe of $\mathcal{F}_{\overline{a}}$.*

B  *is a skeleton subframe of $\mathcal{F}_a$ and*  *is a skeleton subframe of $\mathcal{F}_{\overline{a}}$.*

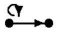
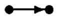
C  *is a skeleton subframe of $\mathcal{F}_a$ and*  *is a skeleton subframe of $\mathcal{F}_{\overline{a}}$.*

D  *is a skeleton subframe of $\mathcal{F}_a$ and*  *is a skeleton subframe of $\mathcal{F}_{\overline{a}}$.*

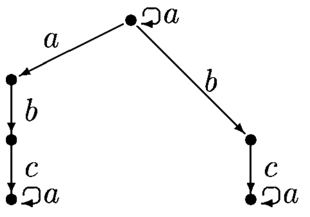
E  *and*  *are skeleton subframes of $\mathcal{F}_a$ and*  *is a skeleton subframe of $\mathcal{F}_{\overline{a}}$.*

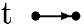
F  *and*  *are skeleton subframes of $\mathcal{F}_a$ and*  *is a skeleton subframe of $\mathcal{F}_{\overline{a}}$.*

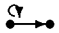
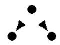
OR

N *there exist three different elements $a, b, c \in \Omega$ such that*  *is a skeleton subframe of $\mathcal{F}_a$, and*  *is a skeleton subframe of $\mathcal{F}_b$ and $\mathcal{F}_c$.*

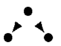
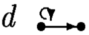
That theorem 3.5.4 can be applied in cases A through F follows from the previous section. For case N, we can apply theorem 3.5.4 with $\sigma = abc$, and $\widehat{F}$ the following frame:



Case III Finally, we show that if we are not in in case I, and not in case A through F, and N, then $\bigoplus_{a\in\Omega}\mathcal{F}_a$ satisfiability is in NP by the analog of theorem 3.2.2. Let Ω' be the set of indices a such that  is a skeleton subframe of $\mathcal{F}_a$. Since we are not in case I, it follows that $|\Omega'| \geq 2$. First suppose that Ω' consists of two elements, say a and b. By lemma 3.5.6, $\bigoplus_{a\in\Omega}\mathcal{F}_a$ satisfiability is polynomial time reducible to $\mathcal{F}_a \oplus \mathcal{F}_b$ satisfiability. By the previous section, $\mathcal{F}_a \oplus \mathcal{F}_b$ satisfiability is in NP, and thus $\bigoplus_{a\in\Omega}\mathcal{F}_a$ satisfiability is in NP as well.

Finally, suppose that Ω' contains at least three elements. Since we are not in case N, it follows that for all $a \in \Omega$,  is not a skeleton subframe of $\mathcal{F}_a$. Furthermore, since we are not in case A, we know that  is not a skeleton subframe of $\mathcal{F}_a$. This situation is the multi-modal analog of case H of the previous section.

By the general analog of theorem 3.2.2, to prove that $\bigoplus_{a\in\Omega}\mathcal{F}_a$ satisfiability is in NP, it suffices to prove that for all a, $\mathcal{F}_a$ satisfiability is in NP and that every satisfiable formula can be satisfied on a frame with polynomially many relevant worlds. From the previous section it follows that for every $a \in \Omega$, $\mathcal{F}_a$ satisfiability is in NP. It remains to show that there exists a polynomial size bound on the relevant parts of $\bigoplus_{a\in\Omega}\mathcal{F}_a$ frames:

Lemma 3.5.8 *If*  *and*  *are not skeleton subframes of $\mathcal{F}_a$, then for any frame $F = \langle W, \{R_a\}_{a\in\Omega}\rangle$ and world $w_0 \in W$: $|\{w|w_0R_{paths(\phi)}w\}| \leq |\phi|$.*

Let $F = \langle W, \{R_a\}_{a\in\Omega}\rangle$ be a frame in $\bigoplus_{a\in\Omega}\mathcal{F}_a$, and let $w_0 \in W$. For $\sigma \in \Omega^*$, let *worlds*(σ) be the number of worlds w such that $w_0R_\sigma w$. Then *worlds*$(\sigma\alpha) \leq 1$, since every world in F has at most one R_α successor, and therefore, for any formula ϕ, the set $\{w|w_0R_{paths(\phi)}w\}$ is of size at most $|\phi|$. □

3.6 The Structure of the Join of Frames

In this section we prove the relationships between (sub)structures of $\mathcal{F}_1$, $\mathcal{F}_2$ and $\mathcal{F}_1 \oplus \mathcal{F}_2$ that were used in the previous sections.

Lemma 3.6.1 *Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two classes of uni-modal frames closed under disjoint union. If $F_1 = \langle W_1, R_1\rangle \in \mathcal{F}_1$ and $F_2 = \langle W_2, R_2\rangle \in \mathcal{F}_2$, then there exists a frame $F = \langle W, S_1, S_2\rangle \in \mathcal{F}_1 \oplus \mathcal{F}_2$ such that F_a is a disjoint subframe of $\langle W, S_a\rangle$ (i.e. $\langle W, S_a\rangle = F_a \oplus (\langle W, S_a\rangle|(W \setminus W_a))$.*

Proof. Let $F_1' = \langle W_1', R_1'\rangle$ consist of the disjoint union of F_1 and $2|W_2| - 1$ frames isomorphic to F_1, and let $F_2' = \langle W_2', R_2'\rangle$ consist of the disjoint union of F_2 and $2|W_1| - 1$ frames isomorphic to F_2. Since $\mathcal{F}_a$ is closed under disjoint union, $F_a' \in \mathcal{F}_a$. Define $F = \langle W, S_1, S_2\rangle$ in such a way that there exist isomorphisms f_a from F_a' to $\langle W, S_a\rangle$ and for all $w \in W_a : f_a(w) = w$. It is obvious that if F can be constructed, then F satisfies the requirements of the lemma.

It remains to show that these isomorphisms can be constructed. But this is easy: it suffices to show that $|W| \geq |W_1 \cup W_2|$. Since $|W| = |W_a| = 2|W_1||W_2|$, and W_a is not empty, it follows that $|W| \geq |W_1| + |W_2| \geq |W_1 \cup W_2|$. □

Proof of lemma 3.3.1 Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two classes of uni-modal frames closed under disjoint union. We have to prove that $\langle W, R_1, R_2\rangle$ is a subframe of $\mathcal{F}_1 \oplus \mathcal{F}_2$ iff $\langle W, R_1\rangle$ is a subframe of $\mathcal{F}_1$ and $\langle W, R_2\rangle$ is a subframe of $\mathcal{F}_2$.

- If $\langle \widehat{W}, \widehat{R}_1, \widehat{R}_2 \rangle$ is a subframe of $\mathcal{F}_1 \oplus \mathcal{F}_2$ then there exists a frame $F = \langle W, R_1, R_2 \rangle \in \mathcal{F}_1 \oplus \mathcal{F}_2$ such that $\widehat{W} \subseteq W$ and $\widehat{R}_a = R_a | (\widehat{W} \times \widehat{W})$. This implies that $\langle \widehat{W}, \widehat{R}_a \rangle$ is a subframe of $\langle W, R_a \rangle$ and $\langle W, R_a \rangle \in \mathcal{F}_a$ by definition of the join.

- Suppose $\langle \widehat{W}, \widehat{R}_a \rangle$ is a subframe of $\mathcal{F}_a$. Let $\langle W_a, R'_a \rangle \in \mathcal{F}_a$ be a superframe of $\langle \widehat{W}, \widehat{R}_a \rangle$. By lemma 3.6.1, there exists a frame $F = \langle W, R_1, R_2 \rangle \in \mathcal{F}_1 \oplus \mathcal{F}_2$ such that $R'_a = R_a | (W_a \times W_a)$. But then $\widehat{R}_a = R_a | (\widehat{W} \times \widehat{W})$, and thus $\langle \widehat{W}, \widehat{R}_1, \widehat{R}_2 \rangle$ is a subframe of F. □

Proof of lemma 3.3.4 Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two classes of uni-modal frames closed under disjoint union. We have to show that: If $\langle \widehat{W}, \widehat{R}_1, \widehat{R}_2 \rangle$ is a skeleton subframe of $\mathcal{F}_1 \oplus \mathcal{F}_2$ then $\langle \widehat{W}, \widehat{R}_1 \rangle$ is a skeleton subframe of $\mathcal{F}_1$ and $\langle \widehat{W}, \widehat{R}_2 \rangle$ is a skeleton subframe of $\mathcal{F}_2$, and if $\langle \widehat{W}, \widehat{R}_1 \rangle$ is a skeleton subframe of $\mathcal{F}_1$ and $\langle \widehat{W}, \widehat{R}_2 \rangle$ is a skeleton subframe of $\mathcal{F}_2$, then there exists a subframe $F = \langle \widehat{W}, R_1, R_2 \rangle$ of $\mathcal{F}_1 \oplus \mathcal{F}_2$ such that $\widehat{R}_1 \subseteq R_1$, $\widehat{R}_2 \subseteq R_2$ and for all $w \in \widehat{W} : wR_aw' \Rightarrow w(\widehat{R}_a \cup \widehat{R}_a^{-1})^*w'$.

- If $\langle \widehat{W}, \widehat{R}_1, \widehat{R}_2 \rangle$ is a skeleton subframe of $\mathcal{F}_1 \oplus \mathcal{F}_2$ then there exists a frame $F = \langle W, R_1, R_2 \rangle \in \mathcal{F}_1 \oplus \mathcal{F}_2$ such that $\widehat{W} \subseteq W$, $\widehat{R}_1 \subseteq R_1$, and $\widehat{R}_2 \subseteq R_2$. This implies that $\langle \widehat{W}, \widehat{R}_a \rangle$ is a skeleton subframe of $\langle W, R_a \rangle$, and $\langle W, R_a \rangle \in \mathcal{F}_a$ by definition of the join.

- Suppose $\langle \widehat{W}, \widehat{R}_a \rangle$ is a skeleton subframe of $\mathcal{F}_a$. Let $\mathcal{W}_a \subseteq Pow(\widehat{W})$ be the set of maximal $\widehat{R}_a \cup \widehat{R}_a^{-1}$ connected subsets of $\widehat{W}$. Then $\widehat{W}$ is the disjoint union of the sets in $\mathcal{W}_a$. For every $V \in \mathcal{W}_a$, $\langle V, \widehat{R}_a | (V \times V) \rangle$ is a subframe of $\langle \widehat{W}, \widehat{R}_a \rangle$ and therefore a skeleton subframe of $\mathcal{F}_a$. Let $R_{V,a}$ be such that $\langle V, R_{V,a} \rangle$ is a subframe of $\mathcal{F}_a$, and $R_{V,a}$ contains the restriction of $\widehat{R}_a$ to the elements of V. Let $F_a = \langle \widehat{W}, \bigcup_{v \in \mathcal{W}_a} R_{V,a} \rangle$. Since $\widehat{W}$ is the disjoint union of the sets in $\mathcal{W}_a$, F_a is the disjoint union of the frames $\langle V, R_{V,a} \rangle$ for $V \in \mathcal{W}_a$. Since all these frames are subframes of $\mathcal{F}_a$, and $\mathcal{F}_a$ is closed under disjoint union, F_a is a subframe of $\mathcal{F}_a$. By lemma 3.3.1, $F = \langle \widehat{W}, \bigcup_{V \in \mathcal{W}_1} R_{V,1}, \bigcup_{V \in \mathcal{W}_2} R_{V,2} \rangle$ is a subframe of $\mathcal{F}_1 \oplus \mathcal{F}_2$. Furthermore, if wR_aw' then there exists a set $V \in \mathcal{W}_a$ such that $w, w' \in V$. Since V is $\widehat{R}_a \cup \widehat{R}_a^{-1}$ connected, it follows that $w(\widehat{R}_a \cup \widehat{R}_a^{-1})^*w'$. □

Chapter 4

Enriching the Language

4.1 Introduction

The independent join occurs in the literature to model for instance multiprocessor systems. However, if we want to make global statements about the system, we need more expressive power.

One of the simplest ways to do this is by enriching the language with the universal modality $[u]$, with semantics $[u]\phi$ is true iff ϕ is true in every world of the model. Another auxiliary modality which occurs in various guises in the literature is the reflexive transitive closure, which we will denote by $[*]$. This modality occurs for instance in temporal logic, as the "always" operator is the reflexive transitive closure of the "nexttime" operator, and in logics of knowledge, where "common knowledge" is defined as the reflexive transitive closure of the $S5$ logics that model the processors.

In this chapter, we investigate the complexity of the satisfiability problems for languages enriched with $[u]$ and $[*]$. Formally, for $F = \langle W, \{R_a\}_{a \in I}\rangle$ define $F_{[u]}$ as $\langle W, \{R_a\}_{a \in I}, R_u\rangle$ such that $R_u = W \times W$, and $F_{[*]}$ as $\langle W, \{R_a\}_{a \in I}, R_*\rangle$ such that $R_* = (\cup_{a \in I} R_a)^*$. When no confusion arises, we'll identify $F_{[u]}$ and $F_{[*]}$ with F. For $\mathcal{F}$ a class of frames, we define $\mathcal{F}_{[u]}$ as the class of all frames $F_{[u]}$ such that $F \in \mathcal{F}$, and $\mathcal{F}_{[*]}$ as the class of all frames $F_{[*]}$ such that $F \in \mathcal{F}$.

In section 4.2, we investigate the transfer of upper bounds. We show that in contrast to the join, decidability does not transfer to the enriched version, even if we add a number of extra restrictions. In particular, we can assume that $\mathcal{F} = Fr(L)$ for L a uni-modal, finitely axiomatizable, canonical and universal first order logic, thereby refuting a conjecture from [GP92].

In sections 4.3 and 4.4, we look at enriched versions of the join. In section 4.3, we show that the satisfiability problems for the enriched versions of the join of two non-trivial classes of frames are always PSPACE-hard, and in almost all cases even EXPTIME-hard. The criterion for EXPTIME-hardness has a particularly simple form: it depends only on the size of rooted subframes of the joinees. We can conclude EXPTIME-hardness if there exist a joinee that has a rooted subframe of size three, and a different joinee with a rooted subframe of size two, or if there exist three different joinees that have rooted subframes of size two. Note that these subframes don't have to be generated subframes of the join. In particular, the presence of a rooted subframe of size k implies the existence of a rooted subframe of size m for all $m \leq k$.

Finally, in section 4.4, we show that this criterion is optimal. We obtain the following analog of theorem 3.4.1: There are three possibilities for the join enriched with $[u]$ or $[*]$:

- All but one of the joinees is trivial, in which case the satisfiability problem is polynomial time reducible to the satisfiability problem of the enriched non-trivial joinee, or
- the satisfiability problem is EXPTIME-hard, by theorem 4.3.3, or
- The satisfiability problem is PSPACE-complete.

4.2 Upper Bounds

Universal Box

In this section, we look at the following problem: given $\mathcal{F}$ and an upper bound on the complexity of $\mathcal{F}$ satisfiability, what can we say about $\mathcal{F}_{[u]}$ satisfiability? The answer is: "not much," as shown by the next theorem:

Theorem 4.2.1 *There exists a class of uni-modal frames $\mathcal{F}$ such that:*

- *$\mathcal{F}$ satisfiability is decidable, and $\mathcal{F}_{[u]}$ satisfiability is undecidable.*
- *$\mathcal{F}$ is first order universal,*
- *$\mathcal{F} = Fr(L)$ for L a uni-modal, finitely axiomatizable, and canonical logic.*

In the proofs that follow, we show undecidability for $\mathcal{F}_{[u]}$ satisfiability by constructing a reduction from the following coRE-complete tiling problem:

$\mathbf{N} \times \mathbf{N}$ tiling: Given a finite set $\mathcal{T}$ of tiles, can $\mathcal{T}$ tile $\mathbf{N} \times \mathbf{N}$?

That is, does there exist a function t from $\mathbf{N} \times \mathbf{N}$ to $\mathcal{T}$ such that:

$$\begin{array}{rcl} \mathit{right}(t(n,m)) & = & \mathit{left}(t(n+1,m)), \text{ and} \\ \mathit{up}(t(n,m)) & = & \mathit{down}(t(n,m+1))? \end{array}$$

An example As an example, we first look at the satisfiability problem with respect to $\mathbf{N} \times \mathbf{N}$; that is, we look $F = \langle \mathbf{N} \times \mathbf{N}, S \rangle$, where S is the successor relation in the grid, i.e. $S = \{\langle \langle n,m \rangle, \langle n+1,m \rangle \rangle, \langle \langle n,m \rangle, \langle n,m+1 \rangle \rangle | n,m \in \mathbf{N}\}$. We will show that F satisfiability is NP-complete, while $F_{[u]}$ satisfiability is coRE-hard.

To prove that F satisfiability is in NP, suppose that ϕ is satisfied in $\langle \mathbf{N} \times \mathbf{N}, S \rangle$. We may assume that ϕ is satisfied at the origin. Now let k be the modal depth of ϕ. Then all relevant worlds $\langle n,m \rangle$ can be reached from the origin in at most k steps. Thus, satisfiability of ϕ can be verified by looking at the frame $\langle \{\langle n,m \rangle | n+m \leq k\}, S \rangle$, which is obviously of polynomial size in the length of ϕ.

Next we give a reduction from $\mathbf{N} \times \mathbf{N}$ tiling to $F_{[u]}$ satisfiability, which implies coRE-hardness for $\mathcal{F}_{[u]}$ satisfiability. Let $\mathcal{T} = \{T_1, \ldots, T_k\}$ be a set of tiles. We construct a formula $\phi_{\mathcal{T}}$ such that:

$$\mathcal{T} \text{ tiles } \mathbf{N} \times \mathbf{N} \text{ iff } \phi_{\mathcal{T}} \text{ is } F_{[u]} \text{ satisfiable.}$$

To encode the tiling, we use a propositional vector *tile* $\in \{1,\dots,k\}$. We need to ensure that adjacent tiles have the same color on their common edges. In order to force this, we have to be able to differentiate between upward and right successors. This would be easy if we knew the coordinates at each world, but as the relevant part of the frame can be infinite, this would take too much space. Let S_x and S_y stand for the right and up successor relations respectively. Then we want the following to hold:

- $S = S_x \cup S_y$;
- S_x and S_y are deterministic;
- $S_xS_y = S_yS_x$.

If S_x and S_y fulfill these conditions, then it is easy to see that one of the relations is the upward successor relation on $\mathbf{N} \times \mathbf{N}$, and the other the right successor relation on $\mathbf{N} \times \mathbf{N}$, which is what we were after. The last requirement seems the most difficult, for how can we force this?

This becomes clear if we look at the two step successors of a world w. Suppose that every world has an S_x and an S_y successor. Let $wS_xS_xw_{xx}$, $wS_xS_yw_{xy}$, $wS_yS_xw_{yx}$, and $wS_yS_yw_{yy}$. Since every world has exactly three 2-step successors, we know that two of these worlds must be equal. We will ensure that the only worlds that can be equal are w_{xy} and w_{yx}, which implies that $S_xS_y = S_yS_x$. We use propositional vector $w3 \in \{0,1,2\}$ and ensure that the values of $w3$ in w_{xy} and w_{yx} are the same, while the values of $w3$ in w_{xx}, w_{xy} and w_{yy} are all different. This is easy: intuitively, we let taking an S_x step correspond to adding 2 (mod3) to the value of $w3$, and taking an S_y step to addition of 1 mod3. Then it is immediate that, for a the value of $w3$ at w, the value of $w3$ is $a+1$ mod 3 at w_{xx}, $a+2$ mod 3 at w_{yy}, and a at w_{xy} and w_{yx}.

Formally, define

$$S_x := \bigcup_{0\le a\le 2} \{\langle w,w'\rangle | M,w \models (w3 = a) \text{ and } M,w' \models (w3 = (a+2) \bmod 3)\}$$

$$S_y := \{\langle w,w'\rangle | M,w \models (w3 = a) \text{ and } M,w' \models (w3 = (a+1) \bmod 3)\}$$

And define the corresponding modalities:

$$[x]\psi := \bigwedge_{a=0}^{2} ((w3 = a) \to \Box((w3 = (a+2) \bmod 3) \to \psi)$$

$$[y]\psi := \bigwedge_{a=0}^{2} ((w3 = a) \to \Box((w3 = (a+1) \bmod 3) \to \psi)$$

Recall that we need to force that $S = S_x \cup S_y$, S_x and S_y are deterministic, and $S_xS_y = S_yS_x$. It suffices to force the first two requirements, since these imply that every world has an S_x and an S_y successor, which in turn implies, by the argument given above, that $S_xS_y = S_yS_x$.

Thus we only have to force that $S = S_x \cup S_y$ and S_x and S_y are deterministic. Note that by definition, S_x and S_y are contained in S. Now look at the following formula, which states that every world has an S_x and an S_y successor:

$$\phi_{succ} = [u](\langle x\rangle\top \wedge \langle y\rangle\top)$$

Since S_x and S_y are by definition disjoint, and every world has exactly two S successors, this formula forces that $S = S_x \cup S_y$ and S_x and S_y are deterministic. We conclude that if ϕ_{succ} is satisfied on a model based on $F_{[u]}$, then one of S_x, S_y is the upward successor relation on $\mathbf{N} \times \mathbf{N}$, and the other the right successor relation on $\mathbf{N} \times \mathbf{N}$. Forcing a tiling is now trivial:

$$\phi_x = [u]((\mathit{tile} = i) \to \bigvee_{\mathit{right}(T_i)=\mathit{left}(T_j))} [x](\mathit{tile} = j))$$

$$\phi_y = [u]((\mathit{tile} = i) \to \bigvee_{\mathit{up}(T_i)=\mathit{down}(T_j))} [y](\mathit{tile} = j))$$

Putting all this together, we define $\phi_{\mathcal{T}}$ to be $\phi_{succ} \wedge \phi_x \wedge \phi_y$. We will prove that $\mathcal{T}$ tiles $\mathbf{N} \times \mathbf{N}$ iff $\phi_{\mathcal{T}}$ is $F_{[u]}$ satisfiable. The left to right direction follows from the arguments given above.

For the converse, suppose $t : \mathbf{N} \times \mathbf{N} \to \mathcal{T}$ is a tiling of $\mathbf{N} \times \mathbf{N}$. We construct the satisfying model for $\phi_{\mathcal{T}}$ as follows: $M = \langle \mathbf{N} \times \mathbf{N}, S, \pi \rangle$ such that:

$$\begin{aligned}&\langle n, m \rangle \models (\mathit{tile} = t(n, m))\\ &\langle n, m \rangle \models (w3 = (2n + m) \bmod 3)\end{aligned}$$

Clearly, $\phi_{\mathcal{T}}$ holds at any node $\langle n, m \rangle$ of M. This proves that $F_{[u]}$ satisfiability is coRE-hard. □

Proof of theorem 4.2.1: We need to construct a class $\mathcal{F}$ of uni-modal frames such that $\mathcal{F}$ is universal first order, $\mathcal{F} = Fr(L)$ for L a uni-modal, finitely axiomatizable, and canonical logic, and $\mathcal{F}$ satisfiability is decidable, while $\mathcal{F}_{[u]}$ is undecidable. The undecidability will be proved using the reduction constructed in the example, that is, we will construct $\mathcal{F}$ in such a way that $\mathcal{T}$ tiles $\mathbf{N} \times \mathbf{N} \Leftrightarrow \phi_{\mathcal{T}}$ is $\mathcal{F}_{[u]}$ satisfiable. The most difficult restriction on $\mathcal{F}$ is the first order definability, for how can such a class of frames be forced to behave like $\mathbf{N} \times \mathbf{N}$? We do need some kind of diamond property, for instance $\forall xyy'\exists z(xRy \wedge xRy' \to yRz \wedge y'Rz)$. But diamond properties are certainly not universal first order.

However, $F_{[u]}$ only has to behave like $\mathbf{N} \times \mathbf{N}$ if $\phi_{\mathcal{T}}$ is $\mathcal{F}_{[u]}$ satisfiable. What does ϕ_{succ} force? That every world has an x and a y successor. Recall from the previous proof that we used the fact that every world in $\mathbf{N} \times \mathbf{N}$ has two successors, and three 2-step successors. Let $\mathcal{F}$ be the class of frames such that every world has at most two successors, and at most three 2 step successors. Then $\mathcal{F}$ is defined by the following universal first order sentence:

$$\begin{aligned}\varphi_\forall = \quad & \forall x\overline{y} \quad (\bigwedge_{1\le i\le 3} xRy_i \to \bigvee_{1\le i<j\le 3} y_i = y_j) \quad \wedge\\ & \forall x\overline{yz} \quad (\bigwedge_{1\le i\le 4} xRy_iRz_i \to \bigvee_{1\le i<j\le 4} z_i = z_j)\end{aligned}$$

The claim is that $\mathcal{F}$ defined this way satisfies the requirements of the theorem. We start by proving that the reduction for coRE-hardness still works, i.e. $\mathcal{T}$ tiles $\mathbf{N} \times \mathbf{N}$ iff $\phi_{\mathcal{T}}$ is $\mathcal{F}_{[u]}$ satisfiable.

The right implication follows from the previous proof; if $\mathcal{T}$ tiles $\mathbf{N} \times \mathbf{N}$ then $\phi_{\mathcal{T}}$ is satisfiable on $\langle \mathbf{N} \times \mathbf{N}, S \rangle$ as defined above, and it is obvious that $\varphi_{\forall}$ holds on this frame, and thus $\phi_{\mathcal{T}}$ is $\mathcal{F}_{[u]}$ satisfiable.

To see that the converse also holds, suppose that $M = \langle W, R, \pi \rangle$ is a model such that $\langle W, R \rangle \models \varphi_{\forall}$ and M satisfies $\phi_{\mathcal{T}}$, say at $w_0 \in W$. We reason in a similar way as in the example: let R_x and R_y correspond to modalities $[x]$ and $[y]$, i.e.

$$R_x := \bigcup_{0 \leq a \leq 2} \{\langle w, w' \rangle | M, w \models (w3 = a) \text{ and } M, w' \models (w3 = (a+2) \bmod 3)\}$$

$$R_y := \bigcup_{0 \leq a \leq 2} \{\langle w, w' \rangle | M, w \models (w3 = a) \text{ and } M, w' \models (w3 = (a+1) \bmod 3)\}$$

By definition, R_x and R_y are disjoint. By ϕ_{succ}, every world has an R_x and an R_y successor. Thus, by $\varphi_{\forall}$, it follows that every world has exactly one R_x and exactly one R_y successor. Since the second conjunct of $\varphi_{\forall}$ forces that every world has at most three 2-step successors, it follows in the same way as in the example that $R_x R_y = R_y R_x$. Now define the tiling as follows:

$$t(n, m) = T_i \text{ iff } M, w \models (\mathit{tile} = i) \text{ where } w_0 R_x^n R_y^m w$$

Since w exists and is unique, t is well-defined. To show that t is indeed a tiling, suppose $t(n, m) = T_i$ and $t(n+1, m) = T_j$. Let w, w' be the corresponding worlds, i.e. $w_0 R_x^n R_y^m w$ and $w_0 R_x^{n+1} R_y^m w'$. Then, by definition, $M, w \models (\mathit{tile} = i)$ and $M, w' \models \mathit{tile} = j$. That these tiles match follows from ϕ_x if we can show that $w R_x w'$. Since $R_x R_y = R_y R_x$, it follows that $R_x^{n+1} R_y^m = R_x^n R_y^m R_x$, and therefore, $w R_x w'$ as required. That $t(n, m)$ and $t(n, m+1)$ match is immediate from the definition and ϕ_y. This proves that $\mathcal{F}_{[u]}$ satisfiability is coRE-hard, and thus undecidable.

Before we prove that $\mathcal{F}$ fulfills the other restrictions of the theorem, we first show that $\mathcal{F}$ is indeed a counter example to decidability transfer to enriched languages. That is, we need to show that $\mathcal{F}$ satisfiability is decidable. Let $M = \langle W, R, \pi \rangle$, $w_0 \in W$ be such that $M, w_0 \models \phi$ and $\langle W, R \rangle \in \mathcal{F}$, i.e. $\langle W, R \rangle \models \varphi_{\forall}$. For k the modal depth of φ, let $\widehat{W}$ be the set of worlds w in W such that $w_0 R^{\leq k} w$. Then $M | \widehat{W}, w_0 \models \phi$, and $\langle W, R \rangle | \widehat{W} \models \varphi_{\forall}$, since $\varphi_{\forall}$ is universal. It is easy to find an upper bound on $\widehat{W}$: since each world has at most two successors, the size of $\widehat{W}$ is certainly less than 2^{k+2}. It follows that ϕ is $\mathcal{F}$ satisfiable iff ϕ is satisfiable on an $\mathcal{F}$ frame of size at most 2^{k+2}. Since $\mathcal{F}$ is first order definable, verifying that a frame is in $\mathcal{F}$ takes polynomial time (in the size of the frame). It is immediate that $\mathcal{F}$ satisfiability is in NEXP.

To complete the proof of theorem 4.2.1, we need to show that $\mathcal{F} = \mathit{Fr}(L)$ for L finitely axiomatizable and canonical. This is easy to prove, for L is defined by the following axioms:

$$\Diamond p_1 \wedge \Diamond p_2 \wedge \Diamond p_3 \rightarrow \Diamond(p_1 \wedge p_2) \vee \Diamond(p_1 \wedge p_3) \vee \Diamond(p_2 \wedge p_3)$$
$$\bigwedge_{1 \leq i \leq 4} \Diamond\Diamond p_i \rightarrow \bigvee_{1 \leq i < j \leq 4} \Diamond\Diamond(p_i \wedge p_j)$$

The claim follows directly from Sahlqvist's theorem [Sah75], but can easily be proven directly. To prove that $\mathcal{F} = \mathit{Fr}(L)$, we need to show that for all frames F, $F \models \varphi_{\forall}$ iff $F \models L$. We prove an equivalence between the second conjunct of $\varphi_{\forall}$ $(\forall x \overline{yz} (\bigwedge_{1 \leq i \leq 4} x R y_i R z_i) \rightarrow$

$(\bigvee_{1\leq i<j\leq 4} z_i = z_j))$ and the second axiom of L. Proving an equivalence between the first conjunct of $\varphi_\forall$ and the first axiom of L can be done by similar arguments, from which $L = Fr(L)$ follows.

First suppose that $M = \langle W, R, \pi\rangle$ and $\langle W, R\rangle \models \varphi_{\forall,2}$. Suppose $M, w \models \Diamond\Diamond p_1 \wedge \Diamond\Diamond p_2 \wedge \Diamond\Diamond p_3 \wedge \Diamond\Diamond p_4$. Let w_1, w_2, w_3 and w_4 be such that $M, w_i \models p_i$ and $wR^2 w_i$. By $\varphi_{\forall,2}$, it holds that $w_i = w_j$ for some i, j with $1 \leq i < j \leq 4$. It follows that $M, w \models \Diamond\Diamond(p_i \wedge p_j)$ as required. For the converse, suppose that $\langle W, R\rangle$ is not an $\varphi_{\forall,2}$ frame. Let $w, w_1, \ldots, w_4$ be such that $wR^2 w_i$ and $w_i \neq w_j$ for $i \neq j$. Define valuation π in such a way that $\pi(p_i) = \{w_i\}$. Then $M, w \models \bigwedge_{1\leq i\leq 4} \Diamond\Diamond p_i$ but $M, w \not\models \Diamond\Diamond(p_i \wedge p_j)$ for all $1 \leq i < j \leq 4$. It follows that $\langle W, R\rangle$ is not an L frame.

Finally, we show that the canonical model for L has an underlying $\mathcal{F}$ frame. For suppose it doesn't, and suppose we violate the second conjunct of $\varphi_\forall$. Then there exist maximal consistent sets $\Gamma, \Gamma_1, \ldots, \Gamma_4$ such that $\Box\Box\psi \in \Gamma \Rightarrow \psi \in \Gamma_i$, and all Γ_i are different. Since all Γ_i are different, there exist formulas ψ_i such that $\psi_i \in \Gamma_i$ and $\psi_i \notin \Gamma_j$ for all $j \neq i$. It follows that:

$$\bigwedge_{1\leq i\leq 4} \Diamond\Diamond(\psi_i \wedge \bigwedge_{j\neq i} \neg\psi_j) \in \Gamma$$

By the second axiom of L, it follows that for some i, j such that $1 \leq i < j \leq 4$:

$$\Diamond\Diamond(\psi_i \wedge \bigwedge_{k\neq i} \neg\psi_k \wedge \psi_j \wedge \bigwedge_{k\neq j} \neg\psi_k) \in \Gamma$$

But then $\Diamond\Diamond\bot \in \Gamma$, which contradicts the consistency of Γ. It follows that L is canonical. This completes the proof of theorem 4.2.1. □

From frames to logics In [GP92], Goranko and Passy also investigate enriching the modal language with a universal modality. They use an axiomatic approach: given a uni-modal logic L, let $L_{[u]}$ consist of the following axioms:

- all L axioms,
- $S5$ axioms for the universal box,
- interaction axiom (containment): $[u]p \to \Box p$

Amongst other things, they investigate what properties transfer from L to $L_{[u]}$. For instance, it is shown that if L is strongly complete, then so is $L_{[u]}$. They also conjecture that decidability transfers. However, the logic L defined above provides a counter example: for, since L is canonical, it follows that L is strongly complete. By the above mentioned transfer result, $L_{[u]}$ is strongly complete as well. Since $Fr(L) = \mathcal{F}$, it follows that L provability is decidable, being the the complement of $\mathcal{F}$ satisfiability, and $L_{[u]}$ is undecidable, being the complement of $\mathcal{F}_{[u]}$ satisfiability.

Transitive Closure

In this section, we investigate what happens to upper bounds on satisfiability if we add $[*]$ to the language. Intuitively, $[*]$ is at least as strong as $[u]$, and thus we would expect the situation to be as least as bad as in the previous section. This is indeed the case:

we will show that theorem 4.2.1 also holds if we replace $[u]$ by $[*]$. But there is more to be said: whereas $\mathcal{F}_{[u]}$ satisfiability is coRE-complete, we'll show that the presence of the transitive closure operator boosts the complexity to Σ_1^1 complete.

Theorem 4.2.2 *There exists a class of uni-modal frames $\mathcal{F}$ such that:*

- *$\mathcal{F}$ satisfiability is decidable, and $\mathcal{F}_{[*]}$ satisfiability is Σ_1^1 complete,*
- *$\mathcal{F}$ is first order universal,*
- *$\mathcal{F} = Fr(L)$ for L a uni-modal, finitely axiomatizable, and canonical logic,*

Let $\mathcal{F}$ and L be as defined in theorem 4.2.1. This immediately implies the second and third clauses of the theorem. It remains to prove that $\mathcal{F}_{[*]}$ satisfiability is Σ_1^1 complete. The Σ_1^1 upper bound is immediate, since any $\mathcal{F}_{[*]}$ satisfiable formula is satisfiable on a countable $\mathcal{F}_{[*]}$ frame. For the corresponding lower bound, we construct a reduction from the following Σ_1^1-complete tiling problem:

> $\mathbf{N} \times \mathbf{N}$ **recurrent tiling**: Given a finite set $\mathcal{T}$ of tiles, and a tile $T_1 \in \mathcal{T}$, can $\mathcal{T}$ tile $\mathbf{N} \times \mathbf{N}$ such that T_1 occurs in the tiling infinitely often on the first row.

That is, does there exist a function t from $\mathbf{N} \times \mathbf{N}$ to $\mathcal{T}$ such that: $\mathit{right}(t(n,m)) = \mathit{left}(t(n+1,m))$, $\mathit{up}(t(n,m)) = \mathit{down}(t(n,m+1))$, and the set $\{i : t(i,0) = T_1\}$ is infinite?

Let $\mathcal{T} = \{T_1, \ldots, T_k\}$ be a set of tiles. We construct a formula ϕ_{rt} such that:

$$\langle \mathcal{T}, T_1 \rangle \in \mathbf{N} \times \mathbf{N} \text{ recurrent tiling} \quad \text{iff} \quad \phi_{rt} \text{ is satisfiable.}$$

To ensure that ϕ_{rt} forces a tiling of $\mathbf{N} \times \mathbf{N}$, we use the formula $\phi_{\mathcal{T}}$ constructed in the proof of theorem 4.2.1. Let $\phi'_{\mathcal{T}}$ be the result of replacing every occurrence of $[u]$ by $[*]$ in $\phi_{\mathcal{T}}$. Then, as in the proof of theorem 4.2.1, the following holds:

- If $\phi'_{\mathcal{T}}$ is not satisfiable, then $\mathcal{T}$ does not tile $\mathbf{N} \times \mathbf{N}$.
- If $M, w_0 \models \phi'_{\mathcal{T}}$, then there exists a tiling t defined as follows:

$$t(n,m) = T_i \text{ iff } M, w \models (\mathit{tile} = i) \text{ where } w_0 R_x^n R_y^m w$$

Now we force the recurrence: we will use a new propositional variable row_0, which can only be true at worlds of the form $t(n,0)$, and we will ensure that there exist an infinite number of worlds where row_0 holds and tile T_1 is placed. Define:

$$\phi_{rec} = [*][y][*]\neg row_0 \wedge row_0 \wedge [*](row_0 \rightarrow \langle x \rangle \langle * \rangle (row_0 \wedge (\mathit{tile} = 1))).$$

Let ϕ_{rt} be the conjunction of $\phi'_{\mathcal{T}}$ and ϕ_{rec}. It is easy to prove that $\langle \mathcal{T}, T_1 \rangle \in \mathbf{N} \times \mathbf{N}$ recurrent tiling iff ϕ_{rt} is $\mathcal{F}_{[*]}$ satisfiable. This proves theorem 4.2.2. □

4.3 Lower Bounds

For the rest of this chapter, let Ω be a set of pairwise disjoint sets of indices, and for $I \in \Omega$, let $\mathcal{F}_I$ be a non-empty class of frames such that $\{\mathcal{F}_I\}_{I\in\Omega}$ is well-behaved in the sense of definition 3.5.3. We look at the way in which the complexity of the satisfiability problems $[\bigoplus_{I\in\Omega}\mathcal{F}_I]_{[*]}$ and $[\bigoplus_{I\in\Omega}\mathcal{F}_I]_{[u]}$ are related to the complexity of the satisfiability problems for $\mathcal{F}_I$ for $I \in \Omega$.

First of all, we show that the enriched versions of the join of non-trivial classes of frames are always PSPACE-hard.

Theorem 4.3.1 *Let Ω be a set of pairwise disjoint sets of indices, and for every $I \in \Omega$, let $\mathcal{F}_I$ be a non empty class of frames closed under disjoint union. If there exist $I, J \in \Omega$ such that $\mathcal{F}_I$ and $\mathcal{F}_J$ contain $\bullet\!\!\rightarrow\!\!\bullet$ as skeleton subframe, then $\bigoplus_{I\in\Omega}\mathcal{F}_I$ satisfiability is PSPACE-hard.*

We construct a reduction from linear temporal logic with operators $\odot$ ("nexttime"), and $\Box$ ("always"), the satisfiability problem of which is in PSPACE-complete [SC85]. Reformulating this result in our notation:

Theorem 4.3.2 ([SC85]) *Let LIN be the closure under disjoint union of finite or infinite uni-modal frames of the form $0R1R2R3R4R\cdots$, i.e. frames $\langle\{i|i<\gamma\},\{\langle i,i+1\rangle|i+1<\gamma\}\rangle$, for $\gamma \in \mathbf{N}\cup\{\omega\}$. Then LIN$_{[*]}$ satisfiability is PSPACE-complete, even if we only look at formulas of the form $\phi_1 \wedge [*]\phi_2$, with $\phi_1, \phi_2 \in \mathcal{L}(\Box)$ of modal depth ≤ 1.*

Let $a \in I$, $b \in J$ be such that $\bullet\xrightarrow{a}\bullet$ is a skeleton subframe of $\mathcal{F}_I$ and $\bullet\xrightarrow{b}\bullet$ a skeleton subframe of $\mathcal{F}_J$. Then all frames of the form $0R_a1R_b2R_a3R_b\cdots$ are skeleton frames of $\bigoplus_{I\in\Omega}\mathcal{F}_I$. These frames are very close to LIN frames. The only problem is that they are only skeleton subframes of $\bigoplus_{I\in\Omega}\mathcal{F}_I$. In the reduction, we have to take care that extra worlds and extra edges are ignored. We use p_s to denote that a world is part of the skeleton subframe, and f_a for worlds that have an R_a successor in the skeleton subframe. Define g as follows:

$$g(p)=p;\ \ g(\neg\phi)=\neg g(\phi);\ \ g(\phi\wedge\psi)=g(\phi)\wedge g(\psi);\ \ g([*]\phi)=[*](p_s\rightarrow g(\phi));$$

$$g(\Box\phi)=(f_a\rightarrow[a](p_s\rightarrow g(\phi)))\wedge(\neg f_a\rightarrow[b](p_s\rightarrow g(\phi))).$$

Note that g is polynomial time computable for formulas of modal depth ≤ 2. g is almost a reduction from LIN satisfiability to $\bigoplus_{I\in\Omega}\mathcal{F}_I$ satisfiability: we only have to force that p_s doesn't behave too strangely. Define reduction f from LIN satisfiability for formulas of the form $\phi=\phi_1\wedge[*]\phi_2$, with $\phi_1,\phi_2\in\mathcal{L}(\Box)$ of modal depth ≤ 1 to $\bigoplus_{I\in\Omega}\mathcal{F}_I$ satisfiability as follows:

$$f(\phi)=p_s\wedge g(\phi)\wedge[*](\neg p_s\rightarrow[*](\neg p_s)).$$

□

EXPTIME-hardness Recall from the introduction, that EXPTIME is for logics with more expressive power, what PSPACE is for uni-modal logics and the join of uni-modal logics. In this section, we prove the analog of the PSPACE-hardness criterion for the join of the previous chapter. Not surprisingly, this analog has the form of an EXPTIME-hardness criterion. Later, we will show that this criterion is optimal, in the sense that it completely characterizes when enriching the join with $[u]$ or $[*]$ causes EXPTIME-hardness, and when the enriched join of two non-trivial classes of frames is EXPTIME-hard.

Recall from theorem 2.2.2 that EXPTIME-hardness for $[\bigoplus_{I\in\Omega}\mathcal{F}_I]_{[u]}$ and $[\bigoplus_{I\in\Omega}\mathcal{F}_I]_{[*]}$ satisfiability follows from the existence of a polynomial time computable function f such that for all formulas ϕ of the form $\phi_1 \wedge [*]\phi_2$, with $\phi_1, \phi_2 \in \mathcal{L}(\Box)$ of modal depth ≤ 1 the following hold: (letting x stand for u and $*$).

- $f(\phi)$ is an $\mathcal{L}_{[x]}$ formula,
- if ϕ is satisfiable in the root of finite binary tree, then $f(\phi)$ is $\mathcal{F}_{[x]}$ satisfiable,
- if $f(\phi)$ is satisfiable, then ϕ is satisfiable.

As in section 3.3, this boils down to the ability of $\bigoplus_{I\in\Omega}\mathcal{F}_I$ subframes to simulate binary trees. We obtain the following analog of theorem 3.5.4:

Theorem 4.3.3 *Let Ω be a set of pairwise disjoint sets of indices, and for every $I \in \Omega$, let $\mathcal{F}_I$ be a non empty class of frames closed under disjoint union. If there exist $I, J \in \Omega$, and a $\mathcal{F}_1 \oplus \mathcal{F}_2$ skeleton subframe $\widehat{F} = \langle \widehat{W}, \{\widehat{R}_I\}_{I\in\Omega}\rangle$ such that:*

1. *$\widehat{W} \supseteq \{w_0, w_\ell, w_r\}$, w_0 is the root of $\widehat{F}$,*
2. *w_0, w_ℓ and w_r have the same reflexive behavior, i.e. for $a = \cup\Omega$:*
$$w_0\widehat{R}_a w_0 \Leftrightarrow w_\ell \widehat{R}_a w_\ell \Leftrightarrow w_r \widehat{R}_a w_r,$$
3. *w_0 has no non-reflexive $\widehat{R}_I$ edges, i.e. if $w_0\widehat{R}_a w$ or $w\widehat{R}_a w_0$ for some $a \in I$, then $w_0 = w$, and*
4. *w_ℓ and w_r have no non-reflexive $\widehat{R}_J$ edges: if $w_\ell\widehat{R}_a w$ or $w\widehat{R}_a w_\ell$ for some $a \in J$, then $w = w_\ell$, and if $w_r\widehat{R}_a w$ or $w\widehat{R}_a w_r$ for some $a \in J$, then $w = w_r$.*

Then $\bigoplus_{I\in\Omega}\mathcal{F}_I$ satisfiability is PSPACE-hard.

Note that this is theorem 3.5.4 without the requirement that for some $\sigma \in (\cup\Omega)^+$: $w_0\widehat{R}_\sigma w_\ell$ and $w_0\widehat{R}_\sigma w_r$. The proof is completely similar to the proof of theorem 3.3.5, i.e. we build tree simulations from copies of $\widehat{F}$. Since w_0 is the root of $\widehat{F}$, there exist $\sigma_\ell, \sigma_r \in (\cup\Omega)^+$ such that $w_0\widehat{R}_{\sigma_\ell}w_\ell$ and $w_0\widehat{R}_{\sigma_r}w_r$. From σ_ℓ and σ_r, we can define "modalities" $[succ_\ell]$ and $[succ_r]$ that play the role of $\Box$ in the binary tree. For ψ in $\mathcal{L}(\Box)$ and of modal depth ≤ 1, let $g(\psi)$ be the result of replacing every occurrence of $\Box\xi$ by $[succ_\ell]\xi \wedge [succ_r]\xi$.

Recall that in the case of theorem 3.3.5, the clause $w_0\widehat{R}_\sigma w_\ell$ and $w_0\widehat{R}_\sigma w_r$ for some $\sigma \in (\cup\Omega)^+$ was used to force formulas to hold in every node of the tree. But if we have $[u]$ or $[*]$ in the language, we can express this right away. The reductions are now obvious: for $\phi = \phi_1 \wedge [*]\phi_2$, with $\phi_1, \phi_2 \in \mathcal{L}(\Box)$ of modal depth ≤ 1, and $x \in \{u, *\}$, let $f(\phi) = g(\phi_1) \wedge [x](p_t \rightarrow g(\phi_2))$. □

4.4 Classification

In this section, we show that the EXPTIME-hardness criterion of the previous section is optimal, in the following sense:

Theorem 4.4.1 *Let Ω be a set of pairwise disjoint sets of indices, and for all $I \in \Omega$, let $\mathcal{F}_I$ be a set of frames such that $\{\mathcal{F}_I\}_{I\in\Omega}$ is well-behaved in the sense of definition 3.5.3. Then we are in one of the following three cases:*

I *There exists a set $I \in \Omega$ such that for all $J \in \Omega$ with $J \neq I$, every frame in $\mathcal{F}_J$ consists of the disjoint union of singletons. In that case, $[\bigoplus_{I\in\Omega} \mathcal{F}_I]_{[u]}$ satisfiability is polynomial time reducible to $[\mathcal{F}_I]_{[u]}$ satisfiability, and $[\bigoplus_{I\in\Omega} \mathcal{F}_I]_{[*]}$ satisfiability is polynomial time reducible to $[\mathcal{F}_I]_{[*]}$ satisfiability.*

II *$[\bigoplus_{I\in\Omega} \mathcal{F}_I]_{[u]}$ satisfiability and $[\bigoplus_{I\in\Omega} \mathcal{F}_I]_{[*]}$ satisfiability are EXPTIME-hard by theorem 4.3.3, or*

III *$[\bigoplus_{I\in\Omega} \mathcal{F}_I]_{[u]}$ satisfiability and $[\bigoplus_{I\in\Omega} \mathcal{F}_I]_{[*]}$ satisfiability are PSPACE-complete.*

Case I: Singletons Let a $I \in \Omega$ be such that for all $J \in \Omega$, $J \neq I$, J consists of the disjoint union of singletons. The reduction is similar to lemma 3.5.6. We use propositional variables r_a for $a \in \cup\Omega \setminus I$ to denote that a world is a reflexive. First, define g as follows:

$$g(p) = p;\ \ g(\neg\phi) = \neg g(\phi);\ \ g(\phi \wedge \psi) = g(\phi) \wedge g(\psi);\ \ g([u]\phi) = [u]g(\psi);\ \ g([*]\phi) = [*]g(\psi);$$

$$g([a]\phi)) = [a]g(\phi) \text{ for } a \in I;\ \ g([a]\phi)) = r_a \rightarrow g(\phi) \text{ for } a \notin I$$

g can be computed in polynomial time, since all index sets $J \in \Omega$ are in P. The reduction $f(\phi)$ is defined as the conjunction of $g(\phi)$ and a formula which forces that the values of the r_a variables can occur, i.e. for all $J \in \Omega$ such that $J \neq I$ and J occurs in ϕ, let J' be the set of J modalities occurring in ϕ, and add the following formula:

$$\bigwedge_{\sigma\in paths(g(\phi))} [\sigma] \bigvee_{\langle\{w\},R_J\rangle\in\mathcal{F}_J} \Big(\bigwedge_{a\in J',R_a\neq\emptyset} r_a \wedge \bigwedge_{a\in J',R_a=\emptyset} \neg r_a \Big)$$

In general, this formula is not polynomial time computable, since the number of different J' singletons can be exponential in the size of J'. However, since we have assumed that Ω and the classes of formulas are well-behaved, every $J \neq I$ is finite. In addition, we know that there occur only finitely many different sizes amongst the index sets in Ω. It follows that there exists a fixed upper bound on the size of J for $J \neq I$, and thus g is polynomial time computable.

Case II: PSPACE-hardness For the remainder of the proof of theorem 3.4.1, we have to show that if there exist at least two sets of indices in Ω such that both contain $\bullet\!\!\longrightarrow\!\!\bullet$ as skeleton subframe, then we are in one of the following two cases:

II $[\bigoplus_{I\in\Omega} \mathcal{F}_I]_{[u]}$ satisfiability and $[\bigoplus_{I\in\Omega} \mathcal{F}_I]_{[*]}$ satisfiability are EXPTIME-hard by theorem 4.3.3, or

III $[\bigoplus_{I\in\Omega} \mathcal{F}_I]_{[u]}$ satisfiability and $[\bigoplus_{I\in\Omega} \mathcal{F}_I]_{[u]}$ satisfiability are PSPACE-complete.

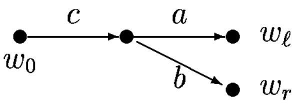
During the proof, we also obtain a complete classification on $\mathcal{F}_I$ subframes such that we are in case II.

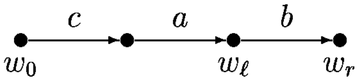
Theorem 4.4.2 *Theorem 4.3.3 can be applied (i.e. we are in case II of theorem 4.4) iff we are in one of the following two cases:*

A *There exist $I, J \in \Omega$ such that $I \neq J$ and $\mathcal{F}_I$ has a rooted subframe of size three, and $\mathcal{F}_J$ has a rooted subframe of size two, or*

B *There exist three sets of indices $I, J, K \in \Omega$ such that $\mathcal{F}_I$, $\mathcal{F}_J$ and $\mathcal{F}_K$ have rooted subframes of size two.*

We first show that if we are in case A and B, then the satisfiability problem for the enriched versions of the join is EXPTIME-hard by theorem 4.3.3. For case A, note that there exist a and b in I such that $\bullet \xleftarrow{a} \bullet \xrightarrow{b} \bullet$ or $\bullet \xrightarrow{a} \bullet \xrightarrow{b} \bullet$ are skeleton subframes of $\mathcal{F}_I$. In addition, let c be such that $\bullet \xrightarrow{c} \bullet$ is a skeleton subframe of $\mathcal{F}_J$. It follows that one of the two following frames is a skeleton subframe of $\mathcal{F}_I \oplus \mathcal{F}_J$, which proves case A.





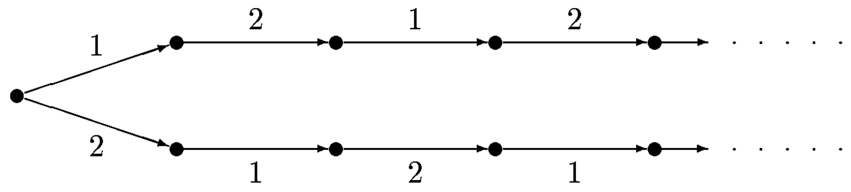
For case B, let a and b be such that $\bullet \xrightarrow{a} \bullet$ is a skeleton subframe of $\mathcal{F}_I$ and $\bullet \xrightarrow{b} \bullet$ a skeleton subframe of $\mathcal{F}_J$. Then $\bullet \xleftarrow{a} \bullet \xrightarrow{b} \bullet$ is a skeleton subframe of $\mathcal{F}_I \oplus \mathcal{F}_J$, and the claim follows from case A. □

Case III: PSPACE upper bounds It remains to show the last part of the theorem, i.e. if we are not in case I, A, B or C, then $[\bigoplus_{I\in\Omega} \mathcal{F}_I]_{[*]}$ satisfiability is PSPACE-complete. First note that, since we are not in case B and C, there exist exactly two sets of indices I and J in Ω such that $\mathcal{F}_I$ and $\mathcal{F}_J$ have 2-world rooted subframes, and for all $K \neq I, J$, $\mathcal{F}_K$ consists of the disjoint union of singletons. By case C, $[\bigoplus_{I\in\Omega} \mathcal{F}_I]_{[u]}$ satisfiability is polynomial time reducible to $[\mathcal{F}_I \oplus \mathcal{F}_J]_{[u]}$ satisfiability, and $[\bigoplus_{I\in\Omega} \mathcal{F}_I]_{[*]}$ satisfiability is polynomial time reducible to $[\mathcal{F}_I \oplus \mathcal{F}_J]_{[*]}$ satisfiability. It therefore suffices to prove the following:

Theorem 4.4.3 *If I and J are finite, $\mathcal{F}_I$ and $\mathcal{F}_J$ are closed under disjoint union, have rooted subframe of size two, but not of size three, then $[\mathcal{F}_I \oplus \mathcal{F}_J]_{[u]}$ and $[\mathcal{F}_I \oplus \mathcal{F}_J]_{[*]}$ satisfiability are PSPACE-complete.*

Hardness follows immediately from the previous section. It remains to prove the upper bound. As the proof is quite involved, we first look at a relatively simple instance. Let $\mathcal{F}_1$ and $\mathcal{F}_2$ consist of the closure under disjoint union of the frame $\bullet \rightarrow \bullet$. We prove that $[\mathcal{F}_1 \oplus \mathcal{F}_2]_{[*]}$ satisfiability is in PSPACE.

Let's first look at the form of a rooted $\mathcal{F}_1 \oplus \mathcal{F}_2$ frame F. Every world is R_1 irreflexive and R_2 irreflexive, and has at most one R_1 successor, and at most one R_2-successor. Furthermore, since F is generated, every world except the root has a predecessor. If w has an a predecessor, then w doesn't have any a successors. From these observations, it follows that F is of the following form where both branches can be finite or infinite:



Apart from the root, F is a linear frame. This situation is very close to $\mathrm{LIN}_{[*]}$ satisfiability as described in the previous section, the satisfiability problem of which is in PSPACE [SC85]. We can use similar methods to prove PSPACE upper bounds in this case, but that leads to duplication of work. What we will do here is to use the fact that $\mathrm{LIN}_{[*]}$ satisfiability is in PSPACE. Recall that LIN is the closure under disjoint union of finite or infinite uni-modal frames of the form $0R1R2R3R4R\ldots$, i.e. frames $\langle\{i|i<\gamma\},\{\langle i,i+1\rangle|i+1<\gamma\}\rangle$, for $\gamma \in \mathbf{N}\cup\{\omega\}$.

As a first step, we show how to simulate linear $\mathcal{F}_1 \oplus \mathcal{F}_2$ frames by LIN frames. This is simple: suppose $F=\langle W, R_1, R_2\rangle$ such that $W=\{i|i<\gamma\}$ and $i+1$ is the only successor of i. Then the corresponding LIN frame is defined on the same set of worlds. To encode R_1 and R_2, we use propositional variables f_1 and f_2 which will be true if a world has a R_1 or R_2 successor. Not all valuations of a LIN frame correspond to a linear $\mathcal{F}_1 \oplus \mathcal{F}_2$ frame. By definition of $\mathcal{F}_1$ and $\mathcal{F}_2$, two consecutive R_1 or R_2 edges do not occur:

$$[*](\neg(f_1 \wedge \Diamond f_1) \wedge \neg(f_2 \wedge \Diamond f_2))$$

Furthermore, a world has a successor if and only if this world has an R_1 or an R_2 successor:

$$[*](f_1 \vee f_2 \leftrightarrow \Diamond\top)$$

It is easy to verify that these formulas ensure that a LIN frame corresponds to a linear $\mathcal{F}_1 \oplus \mathcal{F}_2$ frame. Using this correspondence between frames, we can construct a polynomial time reduction from linear $[\mathcal{F}_1 \oplus \mathcal{F}_2]_{[*]}$ satisfiability to $[\mathrm{LIN}]_{[*]}$ satisfiability. The reduction constructed here will not be the simplest one for this particular example, but it will be easy to generalize. First of all, let $g(\phi)$ be the propositional version of ϕ:

$$g(p)=p;\ \ g(\neg\phi)=\neg g(\phi);\ \ g(\phi\wedge\psi)=g(\phi)\wedge g(\psi);\ \ g([a]\phi)=p_{[a]\phi};\ \ g([*]\phi)=p_{[*]\phi}.$$

Now define $f(\phi)$ as the conjunction of $g(\phi)$, the frame formulas given above, and the following formula which forces proper behavior of the new propositional variables. It is immediate that $\Box$ plays the role of [1] in worlds where f_1 holds, and the role of [2] in worlds where f_2 holds. Furthermore, the transitive closures of corresponding frames coincide. These observations lead to the following formulas:

$$[*](f_1 \to (p_{[1]\psi} \leftrightarrow \Box g(\psi)));\ \ [*](f_2 \to (p_{[2]\psi} \leftrightarrow \Box g(\psi)));\ \ [*](p_{[*]\psi} \leftrightarrow [*]g(\psi))).$$

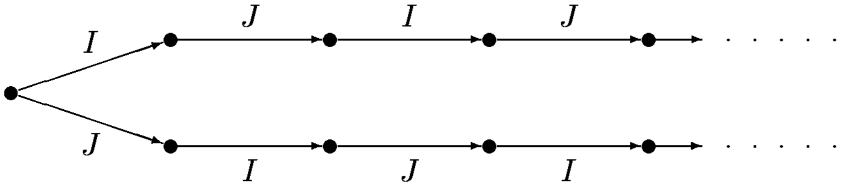
It is easy to verify that ϕ is satisfiable on a linear $\mathcal{F}_1 \oplus \mathcal{F}_2$ frame iff $f(\phi)$ is satisfiable on the corresponding LIN frame. Since f is obviously polynomial time computable, this proves that linear $[\mathcal{F}_1 \oplus \mathcal{F}_2]_{[*]}$ satisfiability is in PSPACE. From this, it is easy to derive a PSPACE upper bound for $[\mathcal{F}_1 \oplus \mathcal{F}_2]_{[*]}$ satisfiability: suppose ϕ is $[\mathcal{F}_1 \oplus \mathcal{F}_2]_{[*]}$ satisfiable. Let M be the model and w the world that witness this. Let Γ be the set of $Cl(\phi)$ formulas satisfied in M at w, and let Γ_1 (Γ_2) be the set of $Cl(\phi)$ formulas satisfied at w if we remove the R_2 (R_1) edge with origin w. Note that $\Gamma_1 \wedge [2]\bot$ and $\Gamma_2 \wedge [1]\bot$ are satisfiable, and therefore satisfiable in linear $\mathcal{F}_1 \oplus \mathcal{F}_2$ frames. Furthermore, since Γ, Γ_1, and Γ_2 all belong to the same world w, all three sets contain the same propositional formulas. In addition, Γ and Γ_1 agree on [1] formulas and Γ and Γ_2 on [2] formulas, and $[*]\psi \in \Gamma$ if and only if $[*]\psi \in \Gamma_1$ and $[*]\psi \in \Gamma_2$. It follows that ϕ is $[\mathcal{F}_1 \oplus \mathcal{F}_2]_{[*]}$ satisfiable iff there exist sets Γ, Γ_1 and $\Gamma_2 \subseteq Cl(\phi)$ such that:

- $\phi \in \Gamma$,
- $\psi \in \Gamma \Leftrightarrow \psi \in \Gamma_a$ for ψ propositional, or $\psi = [a]\xi$,
- $[*]\psi \in \Gamma \Leftrightarrow [*]\psi \in \Gamma_1$ and $[*]\psi \in \Gamma_2$,
- $\Gamma_1 \wedge [2]\bot$ and $\Gamma_2 \wedge [1]\bot$ are maximally satisfiable in linear $\mathcal{F}_1 \oplus \mathcal{F}_2$ frames.

Since subsets of $Cl(\phi)$ can be represented in space polynomial in the length of ϕ, and linear $[\mathcal{F}_1 \oplus \mathcal{F}_2]_{[*]}$ is in PSPACE, it follows that $[\mathcal{F}_1 \oplus \mathcal{F}_2]_{[*]}$ is in PSPACE.

Proof of theorem 4.4.3: Now we turn to the general case, i.e. if I and J are finite, $\mathcal{F}_I$ and $\mathcal{F}_J$ are closed under disjoint union, and do not have rooted subframes of size three, then $[\mathcal{F}_I \oplus \mathcal{F}_J]_{[*]}$ satisfiability is in PSPACE.

Again, we first look at the form of rooted generated $\mathcal{F}_I \oplus \mathcal{F}_J$ frame F. Since $\mathcal{F}_I$ and $\mathcal{F}_J$ only have rooted subframes of size at most two, it follows that for a *subset* of the edges, F is of the following form, where both branches can be finite or infinite, and an edge labeled I denotes some edge in I.



Again, we first look at the linear case, i.e. we look at frames $\langle\{i|i < \gamma\}, R_I, R_J\rangle$ in $\mathcal{F}_I \oplus \mathcal{F}_J$ such that $i+1$ is a successor of i, and for all $a \in I \cup J$, if iR_aj then $|i-j| \leq 1$. As in the simple case, the corresponding LIN frame is defined on the same set of worlds. To encode R_a, we use propositional variables f_a, b_a and r_a to denote that a world has a forward, backward or reflexive a edge. To ensure that a LIN frame indeed encodes an $\mathcal{F}_I \oplus \mathcal{F}_J$ frame, we first of all ensure that there are no I or J connected sets of worlds of size three, i.e. for all a, b such that a and b are both in I or both in J, it is *not* the case that $iR_a(i{+}1)R_b(i{+}2)$, or $(i{+}2)R_a(i{+}1)R_bi$, or $i{+}1R_ai{+}2$ and $i{+}1R_bi$. This can be forced by the following formula:

$$[*](\neg(f_a \wedge \Diamond f_b) \wedge \neg(b_a \wedge \Diamond b_b) \wedge \neg(f_a \wedge b_b))$$

This formula ensures that for $K = I, J$ every maximal K connected set is of the form $\{i, i+1\}$ or $\{i\}$. To ensure that the LIN frame encodes a linear $\mathcal{F}_I \oplus \mathcal{F}_J$ frame, it remains to force that these sets are generated $\mathcal{F}_K$ subframes. For $F = \langle\{w\}, R_K\rangle$ a generated $\mathcal{F}_K$ frame, let ϕ_F be the formula encoding the situation at w:

$$\bigwedge_{a \in K} \neg b_a \wedge \neg f_a \wedge \bigwedge_{wR_aw} r_a \wedge \bigwedge_{\neg wR_aw} \neg r_a$$

And for $F = \langle\{w, w'\}, R_K\rangle$ a rooted generated $\mathcal{F}_K$ frame, with root w, let ϕ_F be the formula encoding the situation at w:

$$\bigwedge_{a\in K} \bigwedge_{wR_aw'} f_a \wedge \bigwedge_{\neg wR_aw'} \neg f_a \wedge \bigwedge_{w'R_aw} \Diamond b_a \wedge \bigwedge_{\neg w'R_aw} \neg\Diamond b_a$$

$$\wedge \bigwedge_{wR_aw} r_a \wedge \bigwedge_{\neg wR_aw} \neg r_a \wedge \bigwedge_{w'R_aw'} \Diamond r_a \wedge \bigwedge_{\neg w'R_aw'} \neg\Diamond r_a$$

Recall that $Rt(\mathcal{F})$ is the class of rooted generated subframes of $\mathcal{F}$. Now add the following formula for $K = I, J$,

$$[*](\bigwedge_{a\in K} \neg b_a \rightarrow (\bigvee_{F\in Rt(\mathcal{F}_K)} \phi_F))$$

It is easy to verify that these formulas ensure that a LIN frame corresponds to a linear $\mathcal{F}_I \oplus \mathcal{F}_J$ frame. Using this correspondence between frames, we can construct a polynomial time reduction from linear $[\mathcal{F}_I \oplus \mathcal{F}_J]_{[*]}$ satisfiability to $[\text{LIN}]_{[*]}$ satisfiability. Again, let $g(\phi)$ be the propositional version of ϕ:

$$g(p) = p;\ g(\neg\phi) = \neg g(\phi);\ g(\phi\wedge\psi) = g(\phi)\wedge g(\psi);\ g([a]\phi) = p_{[a]\phi};\ g([*]\phi) = p_{[*]\phi}.$$

Now define $f(\phi)$ as the conjunction of $g(\phi)$, the frame formulas given above, and the following formulas which force proper behavior of the new propositional variables. We first treat the case for $p_{[a]\psi}$ for $a \in I \cup J$, and $[a]\psi \in Cl(\phi)$. This is relatively straightforward, as all successors are given by variables f_a, b_a and r_a. We treat all occurring combinations. First of all, suppose that $iR_ai{+}1$. Then f_a is true at i, and either i is R_a reflexive, in which case r_a is true at i, and i and $i+1$ are the R_a successors of i, or i is R_a irreflexive, in which case r_a is false at i, and $i+1$ is the only R_a successor of i:

$$[*](f_a \wedge r_a \rightarrow (p_{[a]\psi} \leftrightarrow g(\psi)\wedge\Box g(\psi)) \wedge (f_a \wedge \neg r_a \rightarrow (p_{[a]\psi} \leftrightarrow \Box g(\psi)))$$

We argue in a similar way in the case that $i{+}1R_ai$, i.e. b_a true at $i+1$:

$$[*](\Diamond b_a \wedge \Diamond r_a \rightarrow (\Diamond p_{[a]\psi} \leftrightarrow g(\psi)\wedge\Diamond g(\psi)) \wedge (\Diamond b_a \wedge \neg\Diamond r_a \rightarrow (\Diamond p_{[a]\psi} \leftrightarrow g(\psi)))$$

And if i does not have forward or backward R_a successors:

$$[*](\neg f_a \wedge \neg b_a \rightarrow (p_{[a]\psi} \leftrightarrow (r_a \rightarrow g(\psi)))$$

Finally, we ensure the proper behavior of $p_{[*]\psi}$ for $[*]\psi \in Cl(\phi)$. In the proof of the simple case this was easy, since there $i(R_1 \cup R_2)^*j$ if and only if $j \geq i$, i.e. the transitive closure in linear $\mathcal{F}_1 \oplus \mathcal{F}_2$ frames coincided with the transitive closure in LIN frames. In the general case we treat here, this only goes through for worlds without back edges:

$$[*]((\bigwedge_{a\in I\cup J} \neg b_a) \rightarrow (p_{[*]\psi} \leftrightarrow [*]g(\psi)))$$

On the other hand, if $i{+}1R_ai$ for some a, $[*]\psi$ holds at i iff it holds at $i+1$:

$$[*]((\Diamond \bigvee_{a\in I\cup J} b_a) \rightarrow (p_{[*]\psi} \leftrightarrow \Diamond p_{[*]\psi})))$$

It is easy to verify that ϕ is satisfiable in the root of a linear $\mathcal{F}_I \oplus \mathcal{F}_J$ frame iff $f(\phi) \wedge \bigwedge_{a \in I \cup J} \neg b_a$ is satisfiable on the corresponding LIN frame. Since f is obviously polynomial time computable, this proves that linear $[\mathcal{F}_I \oplus \mathcal{F}_J]_{[*]}$ satisfiability is in PSPACE.

As we have seen, $\mathcal{F}_I \oplus \mathcal{F}_J$ frames do not have to be linear. However, an $\mathcal{F}_I \oplus \mathcal{F}_J$ frame can be viewed as two linear $\mathcal{F}_I \oplus \mathcal{F}_J$ frames that coincide at w_0. Let M_I be the branch that starts with an I edge, i.e. M_I is constructed from M by removing all non-reflexive J edges that originate from w_0, and restricting the set of worlds to those that are still reachable from Now we have split M into two linear $\mathcal{F}_I \oplus \mathcal{F}_J$ models that together contain all the information of M.

Let $\Gamma = \mathit{Form}_M(w_0) \cap \mathit{Cl}(\phi)$, $\Gamma_I = \mathit{Form}_{M_I}(w_0) \cap \mathit{Cl}(\phi)$, and $\Gamma_J = \mathit{Form}_{M_J}(w_0) \cap \mathit{Cl}(\phi)$. Along the line of section 3.2, we can determine the relationship between these sets, thereby reducing $[\mathcal{F}_I \oplus \mathcal{F}_J]_{[*]}$ satisfiability to linear $[\mathcal{F}_I \oplus \mathcal{F}_J]_{[*]}$ satisfiability. This proves theorem 4.4.3 for $[*]$.

For $[\mathcal{F}_I \oplus \mathcal{F}_J]_{[u]}$, we use similar methods. First note that we can't assume that an $[\mathcal{F}_I \oplus \mathcal{F}_J]_{[u]}$ satisfiable formula is satisfiable on a rooted frame. Look for instance at the following formula:

$$p \wedge \langle u \rangle \neg p \wedge [u](p \to \Box p)$$

How many disjoint frames do we need? This depends only on the behavior of $[u]$. Since $[u]$ is an $S5$ modality, it follows from the introduction that $m+1$ worlds are sufficient to satisfy ϕ as far as the $[u]$ part is concerned. (m is the number of $[u]$ occurrences in ϕ.) This leads to the following equivalence:

ϕ is $[\mathcal{F}_I \oplus \mathcal{F}_J]_{[u]}$ satisfiable
iff
There exist $k \leq m$ and $\mathit{Cl}(\phi)$ subsets $\Gamma_0, \Gamma_1 \ldots, \Gamma_k$ such that:

- $[u]\psi \in \Gamma_i$ iff $[u]\psi \in \Gamma_j$ for all ψ and all $i, j \leq m$,
- If $\langle u \rangle \psi \in \Gamma_i$ then $\psi \in \Gamma_j$ for some $j \leq m$,
- Γ_i is maximally satisfiable in a rooted $[\mathcal{F}_I \oplus \mathcal{F}_J]_{[u]}$ frame.

Obviously, the main contribution to the complexity comes from the last condition, and just as obviously, this step is in PSPACE by the proof for $[*]$ given above. □

Chapter 5

The Complexity of Attribute Value Logics

5.1 Introduction

Attribute Value Structures (AVSs) are probably the most widely used means of representing linguistic structure in current computational linguistics, and the process of unifying descriptions of AVSs lies at the heart of many parsers. As a number of people have recently observed (see for example Kracht [Kra89], Blackburn [Bla2], Moss [Mos], Reape [Rea91] and Schild [Sch90]) the most common formalisms for describing AVSs are notational variants of propositional modal languages, AVSs themselves are Kripke models, and unification amounts to looking for a satisfying model for $\phi \wedge \psi$ given two (modal) wffs ϕ and ψ. The purpose of this chapter is to make use of this connection with modal logic to investigate the complexity of various unification tasks of interest in computational linguistics.

The chapter is structured as follows. The next section begins with an introduction to such topics as "attributes," "values," and "unification" and why they are of interest in computational linguistics. It then goes on to explain the link with modal logic, and gives the syntax and semantics of three modal languages — L, L^N and L^{KR} — which correspond to three common unification formalisms. In the third section we examine the satisfiability problems for these languages and show, using a very simple "small model" argument, that all three are NP-complete. In the fourth section we introduce three stronger languages, $L^{\Box}$, $L^{N\Box}$ and $L^{KR\Box}$. These are L, L^N and L^{KR} respectively augmented by the universal modality $\Box$. Adding this modality allows general constraints on linguistic structure to be expressed. As we will show, however, there is a price to pay: the satisfiability problem for $L^{KR\Box}$ is Π^0_1-complete. We then go on to show that dropping the ability to enforce generalizations involving re-entrancy results in decidable systems. In fact we show that the satisfiability problems for both $L^{\Box}$ and $L^{N\Box}$ are EXPTIME-complete. In the fifth section we examine modal languages in which recursive constraints on linguistic structure can be expressed, namely systems built using the master modality $[*]$ of Gazdar, Pullum, Carpenter, Klein, Hukari and Levine [GPC+88] and Kracht [Kra89]. We augment our base languages L, L^N and L^{KR} with $[*]$, forming $L^{[*]}$, $L^{N[*]}$ and $L^{KR[*]}$ respectively, and investigate the complexity of their satisfiability problems. We show that many of the proof methods and results from our discussion of the the universal modality transfer to the new setting, though in the case of most interest the satisfiability problem for $L^{KR[*]}$

turns out to be highly undecidable, in fact, Σ^1_1-complete. We conclude the chapter with a table summarizing our results and a discussion of more general issues arising from this work.

The chapter is relatively self contained; in particular, all the necessary concepts from unification based grammar and modal logic are presented. However we do assume that the reader understands what is meant by such complexity classes as NP, EXPTIME and so on; such definitions may be found in Balcázar, Díaz, and Gabarró [BDG88], for example. Further, later in the chapter some ideas from Propositional Dynamic Logic (PDL) are used. While these are explained, some readers may find the additional background provided by Harel [Har84] helpful. For further information on modal logic the reader is referred to Hughes and Cresswell [HC84]; and for more on unification based grammar, Shieber [Shi86] and Carpenter [Car92] are useful. Finally, it's worth remarking that there is a hidden agenda: although we emphasize the use of modal logic as tool for grammar specification, it is our belief that modal techniques have a wider role to play in computational linguistics and some possibilities are noted in the course of the chapter.

5.2 Attribute Value Logic

Even the most cursory examination of recent proceedings of computational linguistics conferences reveals that there is a substantial level of interest in such topics as "attributes," "values," and "unification." This section presents a brief introduction to these topics, and explains what they have to do with modal logic. The basic point it makes is that the most common machinery underlying Attribute Value grammar formalisms is simply that of propositional modal logic, and that testing whether unification is possible amounts to testing for modal satisfiability. This correspondence provides the *raison d'être* of the chapter: by examining the complexity of the satisfiability problem for the modal languages involved, we learn — often very straightforwardly — about the complexity of various tasks of interest to computational linguistics.

Perhaps the best way of approaching these topics is via Attribute Value Matrices (AVMs), or Feature Value Matrices as they are sometimes called. A (rather simple) AVM might look something like this:

$$\begin{bmatrix} \textsc{case} & \textit{nominative} \\ \textsc{agreement} & [\textsc{person}\ \textit{1st}] \end{bmatrix}$$

Such an AVM is taken to be a partial description of some piece of linguistic structure. In this case we are describing a piece of linguistic structure that has two attributes, namely CASE and AGREEMENT. The CASE attribute takes as value the atomic value *nominative*, while the AGREEMENT attribute takes as value the complex entity [PERSON*1st*]. This complex entity consists of an attribute PERSON that takes as value the atomic value *1st*. The particular atomic values (or constants) and attributes (or features) that may occur in AVMs varies widely from theory to theory, but typical choices of atomic entities a syntactician might make are *singular, plural, 3rd, 2nd, 1st, genitive* and *accusative*; and when it comes to a choice of attributes the selection might include TENSE, NUMBER, PERSON, AGREEMENT, and CASE. But although the different theories differ on the particular choices made, and indeed in the uses they put this machinery to, they are united in agreeing that at least a part of our descriptions of linguistic structure should embody the idea of attributes taking (possibly complex) values.

The information expressed by AVMs can be considerably more complex than in the above example. The above AVM is purely conjunctive, but many linguists feel it is necessary to be able to express both disjunctive and negative information in their Attribute Value grammars. To give two well known examples due to Kartunen [Kar84], one might write

$$\begin{bmatrix} \textsc{number} & \textit{plural} \\ \textsc{case} & \{\textit{nominative, genitive, accusative}\} \end{bmatrix}$$

an AVM which states that the attribute CASE takes one of the values *nominative*, *genitive*, or *accusative*, but doesn't say which; or one might write

$$\begin{bmatrix} \textsc{number} & \textit{plural} \\ \textsc{case} & [\neg\ \textit{dative}] \end{bmatrix}$$

an AVM which specifies that CASE doesn't take the value *dative*.

It's worth making a short historical remark here. We'll shortly be introducing Attribute Value Structures (AVSs) and treating them as semantic structures for AVMs. That is, we're going to be adopting the now standard distinction between description languages (for example AVMs) and linguistic structure (the AVSs). Historically, the impetus for making this distinction was motivated by the difficulties involved in giving a precise account of AVMs that employed disjunction or negation. The distinction was first introduced by Pereira and Shieber [PS84], and it underpins the influential work of Kasper and Rounds [KR86, RK86, KR90]. Thus the move towards full Boolean expressivity marked an important turning point in the development of Attribute Value formalisms.

What do computational linguists do with AVMs? The answer is, they try to unify them. Intuitively, unifying two AVMs means forming another AVM that combines all the information about Attribute Value dependencies contained in the two constituent AVMs. For example, writing $\sqcup$ to indicate unification, we have:

$$\begin{bmatrix} \textsc{agr} & [\textsc{per}\ \textit{1st}] \\ \textsc{case} & \textit{nominative} \end{bmatrix} \sqcup [\textsc{agr}\ [\textsc{num}\ \textit{plural}]] = \begin{bmatrix} \textsc{agr} & \begin{bmatrix} \textsc{num} & \textit{plural} \\ \textsc{per} & \textit{1st} \end{bmatrix} \\ \textsc{case} & \textit{nominative} \end{bmatrix}$$

There is a clear sense in which the AVM on the right hand side embodies all the information in the two constituent structures; it is the result of unifying these structures.

But this is rather vague. *Precisely* when is unification possible? Answering this question will lead us first to AVSs, the semantics of AVMs, and then, quite naturally, to the link with modal languages.

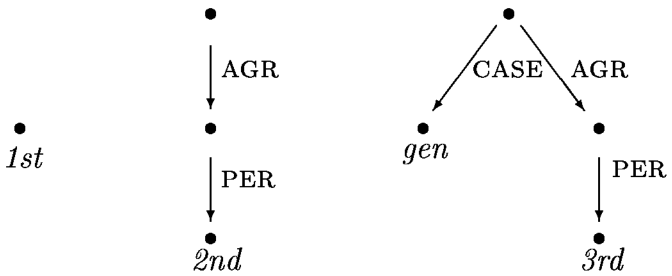
AVSs are certain kinds of *decorated labeled graphs*. Such graphs play the central role in unification based linguistics: they are the mathematical model of linguistic structure underlying these frameworks. A number of definitions of AVSs exist in the literature. We shall work with a particularly simple one:

Definition 5.2.1 (Attribute Value Structures) *Let $\mathcal{L}$ and $\mathcal{A}$ be non-empty finite or denumerably infinite sets, the set of* labels *and the set of* atomic information *respectively. An Attribute Value Structure (AVS) of signature $\langle \mathcal{L}, \mathcal{A} \rangle$ is a triple $\langle W, \{R_l\}_{l \in \mathcal{L}}, \{Q_\alpha\}_{\alpha \in \mathcal{A}} \rangle$, where W is a non-empty set, the set of nodes; for all $l \in \mathcal{L}$, R_l is a binary relation on W that is a partial function; and for all $\alpha \in \mathcal{A}$, Q_α is unary relation on W.* □

The most important thing to note about this definition is the requirement that all the binary relations be partial functions. As we shall see, this demand plays a crucial role in establishing some of our complexity results.

The definition covers all the well known definitions of Attribute Value Structures, and in particular those of Gazdar *et al.* [GPC+88] and Kasper and Rounds [KR86]. Moreover it's not too loose: there are only two reasonably common further restrictions on the binary relations that it doesn't insist on. The first is that AVSs must be *point generated*. In point generated AVSs there is always a starting node $w_0 \in W$ such that all other nodes $w \in W$ are reachable via transition sequences from w_0. The second is that AVSs must be *acyclic*, which means that it is never possible to return to a node w by following some sequence of R_l transitions from w. As neither of these restrictions plays a prominent role in the linguistics literature anymore, we ignore them here. This definition also ignores three constraints computational linguists used to routinely place on node decoration. The constraints in question are these. First, for all $w \in W$ and all $\alpha, \beta \in \mathcal{A}$, if $w \in Q_\alpha$ and $\alpha \neq \beta$ then $w \notin Q_\beta$. That is, the constraint forbids what linguists call "constant-constant clashes." Second, for all $w \in W$, w is in Q_α for some $\alpha \in \mathcal{A}$ iff w is a terminal node. This constraint rules out "constant-compound clashes." Third, for all $w, w' \in W$, if $w \in Q_\alpha$ and $w' \in Q_\alpha$ then $w = w'$. Once again, the main reason for ignoring these demands is that they no longer play the prominent role they once did. Indeed in more recent work in computational linguistics, particularly work in the Head Driven Phrase Structure Grammar (HPSG) framework, much use is made of *sorts* [Pol]; and sorts are essentially pieces of atomic information that don't obey these three restrictions.

Let's consider some concrete examples of AVSs. Suppose we are working with some linguistic theory which contains among its theoretical apparatus the attributes PERSON, CASE and AGREEMENT, and the atomic information *3rd*, *2nd*, *1st* and *genitive*. That is, our linguistic theorizing has specified a signature $\langle \mathcal{L}, \mathcal{A} \rangle$ such that {PERSON, CASE, AGREEMENT} $\subseteq \mathcal{L}$, and {*3rd*, *2nd*, *1st*, *genitive*} $\subseteq \mathcal{A}$. Then the following graphs are all examples of AVSs of this signature, as (modulo some obvious abbreviations) nodes are decorated only with items drawn from $\mathcal{A}$ and transitions are labeled only with items drawn from $\mathcal{L}$:



What do AVSs have to do with AVMs? As has already been remarked, AVMs are partial descriptions of linguistic structure, and in fact the structure they describe is the structure embodied in the definition of AVSs. That is, AVMs are a formal language for describing linguistic structure, AVSs provide the interpretation for AVMs, and thus the relationship is that which always exists between semantic and syntactic entities: we talk of AVSs satisfying (or failing to satisfy) the AVMs. To return to our examples, the first graph, consisting of a single node decorated with the atomic information *1st*, satisfies the atomic AVM *1st*. Why? Because this atomic AVM demands a node decorated with the atomic information *1st*, and the first graph is such a node. The second graph satisfies the AVM $[\textsc{agreement}\ [\textsc{person}\ \ \mathit{2nd}]]$ at its root node. Why? Because this AVM demands a node in some piece of linguistic structure that has the following property: a transition along an $R_{\textsc{agreement}}$ relation takes one to a node from which it is possible to make an $R_{\textsc{person}}$ transition to a node decorated with the information *2nd*. The root node of the second graph has this property. Finally, consider the third graph. This satisfies the AVM

$$\begin{bmatrix} \textsc{agreement} & [\textsc{person}\ \mathit{3rd}] \\ \textsc{case} & \mathit{genitive} \end{bmatrix}$$

at its root node.

Now, we could give a precise definition of what it means for an AVS to satisfy an AVM, but in fact this would be a waste of energy, for, as we'll now see, the satisfaction relation between AVSs and AVMs is just a disguised version of something very familiar: the satisfaction relation between Kripke models and modal wffs. There are two facets to this correspondence, the semantical and the syntactical. We'll treat each in turn, beginning with the semantical.

Consider once more the definition of AVSs as triples $\langle W, \{R_l\}_{l\in\mathcal{L}}, \{Q_\alpha\}_{\alpha\in\mathcal{A}}\rangle$. Such triples are just (multimodal) Kripke models: each R_l interprets a modal operator $\langle l\rangle$, and each unary relation Q_α interprets the propositional symbol p_α. To be sure, multimodal Kripke models are usually presented as triples $\langle W, \{R_l\}_{l\in\mathcal{L}}, \pi\rangle$, where π is a valuation function from a collection of propositional symbols $\mathcal{P}$ to $Pow(W)$. (In such presentations the pair $\langle W, \{R_l\}_{l\in\mathcal{L}}\rangle$ is usually given a special name, namely *multiframe*.) But obviously there is no mathematical substance to this difference. Given a traditionally presented Kripke model $\langle W, \{R_l\}_{l\in\mathcal{L}}, \pi\rangle$, we have that $\langle W, \{R_l\}_{l\in\mathcal{L}}, \{\pi(p) : p \in \mathcal{P}\}\rangle$ is an AVS of signature $\langle \mathcal{L}, \mathcal{P}\rangle$; and conversely, given any AVS $\langle W, \{R_l\}_{l\in\mathcal{L}}, \{Q_\alpha\}_{\alpha\in\mathcal{A}}\rangle$, we have that $\langle W, \{R_l\}_{l\in\mathcal{L}}, \pi\rangle$ is a Kripke model, where π is the function from the set of (α-indexed) propositional variables $\mathcal{P}$ to $Pow(W)$ defined by $\pi(p_\alpha) = Q_\alpha$. In short, every AVS is a Kripke model, and vice versa.

Now for the syntactical correspondence. Consider the following AVM.

$$\begin{bmatrix} \textsc{agreement} & [\textsc{person}\ \mathit{1st}] \\ \textsc{case} & \mathit{nominative} \end{bmatrix}$$

This corresponds to

$$\begin{array}{ll} & \langle\textsc{agreement}\rangle\langle\textsc{person}\rangle\,\mathit{1st} \\ \wedge & \langle\textsc{case}\rangle\mathit{nominative} \end{array}$$

The key point to grasp is that the function of the attributes AGREEMENT, PERSON and CASE in the AVM is precisely analogous to the function of the existential modalities $\langle\textsc{agreement}\rangle$, $\langle\textsc{person}\rangle$ and $\langle\textsc{case}\rangle$ in the modal wff. The function of the attributes is to demand the existence of certain transitions in AVSs, the function of the modalities is to demand the existence of certain transitions in Kripke models. But AVSs are just Kripke models, and thus the equivalence of the description languages is clear. The rest of the correspondence is straightforward: atomic values correspond to propositional symbols, and the modal wff is in effect just a linearization of the AVM. To put it more generally, AVMs are just modal wffs written in a particularly perspicuous manner.

This correspondence extends in the obvious manner to AVMs with full Boolean expressivity. For example corresponding to the following AVM:

$$\begin{bmatrix} \text{NUMBER} & \neg \textit{plural} \\ \text{CASE} & \{\textit{nominative, genitive, accusative}\} \end{bmatrix}$$

we have the wff

$$\langle\text{NUMBER}\rangle\neg \textit{plural} \wedge \langle\text{CASE}\rangle(\textit{nominative} \vee \textit{genitive} \vee \textit{accusative}).$$

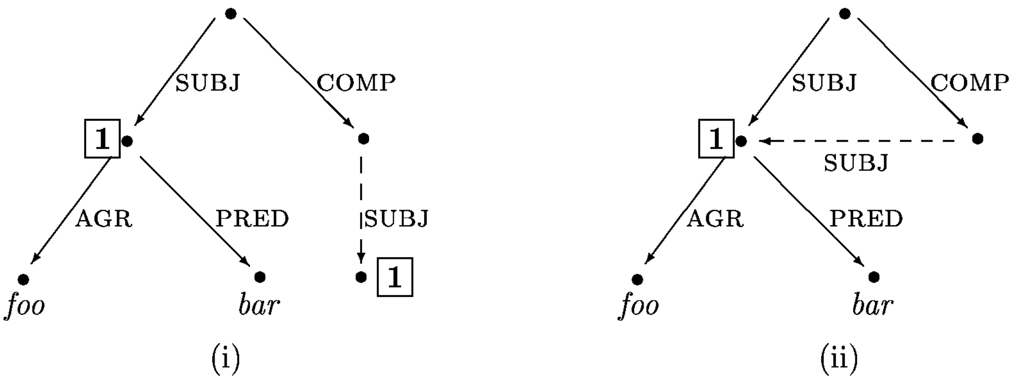
The most important aspect of the link between modal languages and AV formalisms is what it tells us about unification. Recall that unification is the attempt to coherently merge two AVMs. But what does "coherent" mean? It means that the demands that the two AVMs make can be simultaneously satisfied at some node in some AVS. Now, both AVMs correspond to a modal wff. Call these two wffs ϕ and ψ respectively. Then we have that unification succeeds iff $\phi \wedge \psi$ is satisfiable at some node in some Kripke model. That is, *testing whether unification is possible amounts to testing for modal satisfiability.* This observation (familiar from the work of Kasper and Rounds [KR86, RK86, KR90] and Kracht [Kra89]) lies at the heart of the chapter.

The correspondence we have noted extends to richer unification formalisms than the rather simple AVMs so far considered. In particular, it extends to formalisms that have the ability to encode *re-entrancy.* Re-entrancy is a very influential idea in unification based approaches to grammar, and we need to discuss it, and how it can be dealt with in modal languages.

One of the best known notations for forcing re-entrancy is to use AVMs with "boxlabels." Consider the following AVM:

$$\begin{bmatrix} \text{SUBJ} & \boxed{1}\begin{bmatrix} \text{AGR} & \textit{foo} \\ \text{PRED} & \textit{bar} \end{bmatrix} \\ \text{COMP} & [\text{SUBJ} \ \ \boxed{1}\,] \end{bmatrix}$$

The boxlabels are the $\boxed{1}$s. What is intended by this notation is explained by the following graphs:



The first graph does *not* satisfy the AVM at its root node. This is because $\boxed{1}$ is a name: it labels a *unique* node. The second graph *does* satisfy the AVM. The crucial difference is that in this graph the SUBJ *re-enters* the graph at the named node. Thus all the conditions demanded by the AVM are satisfied, including the demand that $\boxed{1}$ picks out a unique node.

How can we make modal languages referential in this way? The key idea needed can be traced back to early work by Arthur Prior [Pri67], and Robert Bull [Bul70]: it is to introduce a second sort of atomic symbol constrained to be true at exactly one node. These new symbols "name" the unique node they are true at. In this chapter these symbols are called *nominals*, and they are usually written as i, j, k and m.

AVM boxlabels correspond straightforwardly to nominals. Consider once more the following AVM:

$$\begin{bmatrix} \textsc{subj} & \boxed{1}\begin{bmatrix}\textsc{agr} & \mathit{foo}\\ \textsc{pred} & \mathit{bar}\end{bmatrix} \\ \textsc{comp} & [\textsc{subj}\ \ \boxed{1}\,] \end{bmatrix}$$

This corresponds to the following wff:

$$\begin{array}{ll} & \langle\textsc{subj}\rangle(i \wedge \langle\textsc{agr}\rangle \mathit{foo} \wedge \langle\textsc{pred}\rangle \mathit{bar})\\ \wedge & \langle\textsc{comp}\rangle\langle\textsc{subj}\rangle i\end{array}$$

Note that the nominal i is doing the same work in the modal wff that $\boxed{1}$ does in the AVM. More generally, the use of nominals permits a transparent linearization of those AVMs that utilize boxlabels.

Although AVM notation is widely used, it is certainly not the only notation computational linguists use to describe AVSs. Another influential notation arose from the command language of the PATR-II system [Shi86]. PATR-II is an "implemented grammar formalism," a program which provides a high level interface language geared towards the needs of the linguist, together with a parser. The linguist writes grammars in the interface language and tests them using the parser. The use of *path equations* for specifying re-entrancy arose from this source. A user of PATR-II might write:

$$\langle \textsc{vp head}\rangle = \langle \textsc{vp verb head}\rangle.$$

This path equation means that the sequence of transitions encoded by the list of attributes on the left takes one to the same node as the sequence of transitions encoded by the list of attributes on the right. That is, both transition sequences lead to the same node. Note that although this mechanism permits re-entrancy to be specified, it does so in a very different way from the "boxlabels" approach: no node labeling is involved.

To capture the effect of this in a modal language, we're going to extend the basic language in such a way as to permit "modal path equations" to be formed. In particular, we'll add a new primitive symbol $\approx$ to allow us to equate strings of modalities. This will permit wffs such as

$$\langle\textsc{vp}\rangle\langle\textsc{head}\rangle \approx \langle\textsc{vp}\rangle\langle\textsc{verb}\rangle\langle\textsc{head}\rangle,$$

to be formed, and we will define the semantics of these new wffs so that they capture the meaning of the PATR-II path equations. Actually, we'll also add a second new primitive symbol, 0. This will be a name for the null transition, and with its help we will be able to write such path equations as $\langle b\rangle\langle a\rangle \approx 0$. This wff, for example, will mean that making an R_b transition followed by an R_a transition is the same as making the null transition. That is, the path R_bR_a terminates at its starting point.

It should now be clear that various AV formalisms correspond straightforwardly to propositional modal languages. To conclude this section let's make our discussion of these modal languages more precise. Syntactically, the language L (of signature $\langle \mathcal{L}, \mathcal{A} \rangle$) is a language of propositional modal logic with an $\mathcal{L}$ indexed collection of distinct (existential) modalities and an $\mathcal{A}$ indexed collection of propositional symbols. As primitive Boolean symbols we choose $\neg$ and $\vee$. The wffs of the language are defined by saying that: (a) All propositional symbols p_α are wffs, for all $\alpha \in \mathcal{A}$; (b) If ϕ and ψ are wffs then so are $\neg\phi$, $\phi \vee \psi$, and $\langle l \rangle \phi$, for all $l \in \mathcal{L}$; (c) Nothing else is a wff. We define the other Boolean connectives $\rightarrow$, $\wedge$, $\leftrightarrow$, $\perp$, and $\top$ in the usual way. We also define $[l]\phi$ to be $\neg\langle l \rangle \neg\phi$, for all $l \in \mathcal{L}$ and all wffs ϕ. The following syntactic notions will be useful. The *degree* of a formula is the number of (primitive) connectives it contains. The *length* of a wff ϕ (denoted by $|\phi|$) is the number of (primitive) symbols it contains. (We will also use the "$|\ .\ |$" notation to indicate cardinality, but this double use should cause no confusion.)

To interpret L we use Kripke models M of signature $\langle \mathcal{L}, \mathcal{A} \rangle$. Such a Kripke model is a triple $\langle W, \{R_l\}_{l \in \mathcal{L}}, \pi \rangle$, where W is a non-empty set (the set of nodes); each R_l is a binary relation on W that is also a partial function, that is, for every node w there exists at most one w' such that wR_lw'; and π (the valuation) is a function which assigns each propositional symbol p_α a subset of W. We interpret wffs of L on models M in the familiar fashion:

$$\begin{array}{lll} M, w \models p_\alpha & \text{iff} & w \in \pi(p_\alpha) \\ M, w \models \neg\phi & \text{iff} & M, w \not\models \phi \\ M, w \models \phi \vee \psi & \text{iff} & M, w \models \phi \text{ or } M, w \models \psi \\ M, w \models \langle l \rangle \phi & \text{iff} & \exists w'(wR_lw' \ \&\ M, w' \models \phi) \end{array}$$

If $M, w \models \phi$ then we say that M *satisfies* ϕ at w, or ϕ is *true* in M at w. To sum up, the language L corresponds to the "core" AVM notation used by computational linguists. Its models are just AVSs, and the way L formulas are evaluated in a model is just the way AVMs are checked against AVSs.

L lacks any mechanism for enforcing re-entrancy. This lack is made good in its extensions, L^N and L^{KR}. The language L^N (of signature $\langle \mathcal{L}, \mathcal{A}, \mathcal{B} \rangle$) is the language L (of signature $\langle \mathcal{L}, \mathcal{A} \rangle$) augmented by a $\mathcal{B}$ indexed collection of distinct new propositional symbols called *nominals.* These symbols are typically written as i, j, k and m and can be freely combined with the other symbols in the usual fashion to make wffs. We assume that $\mathcal{B}$ is at most countably infinite. To interpret nominals we insist that any valuation must assign a singleton subset to each nominal. That is, an L^N model is just an L model whose valuation has been extended to assign singletons to nominals. Because each nominal is thus true at exactly one node in any model, it acts as a "name" identifying that node. L^N corresponds to AVMs augmented with "boxlabels" for indicating re-entrancy. There have been a number of logical investigations of intensional languages containing nominals. In addition to the early work by Prior and Bull already mentioned, see Passy and Tinchev [PT85], Gargov and Passy [GP88] and Passy and Tinchev [PT91] for an examination of nominals in the setting of Propositional Dynamic Logic (PDL); see Gargov, Passy and Tinchev [GPT86] and Gargov and Goranko [GG] for nominals in the setting of modal logic; and finally see Blackburn [Bla1] for nominals in tense logic.

The language L^{KR} is L augmented by two new symbols, 0 and $\approx$. The symbol 0 acts as a name for the null transition. In what follows we shall assume without loss of generality that $0 \notin \mathcal{L}$, and denote the identity relation on any set of nodes W by R_0.

(This convention simplifies the statement of the following truth definition.) We use $\approx$ to make *path equations*: given any nonempty sequences A and B made up of modalities and $\langle 0 \rangle$, then $A \approx B$ is a path equation. Path equations are wffs and can be combined with other wffs in the usual way to make more complex wffs. L^{KR} models are just L models, and we interpret the path equations as follows. For all $l_1, \ldots, l_k, l'_1, \ldots, l'_m \in \mathcal{L} \cup \{0\}$:

$$M, w \models \langle l_1 \rangle \cdots \langle l_k \rangle \approx \langle l'_1 \rangle \cdots \langle l'_m \rangle \text{ iff } \exists w'(wR_{l_1} \ldots R_{l_k} w' \ \& \ wR_{l'_1} \ldots R_{l'_m} w').$$

L^{KR} models the path equation mechanism of PATR-II. The negation free fragment of this language was first defined and studied by Kasper and Rounds [KR86][RK86]; a more detailed presentation of their work may be found in [KR90]. Further logical investigations of L^{KR} may be found in Moss [Mos] and Blackburn [Bla2].

It is instructive (and will later prove technically useful) to examine L, L^N and L^{KR} from the more general perspective provided by *modal correspondence theory*. This subject is the systematic study and exploitation of the relationships that exist between modal languages and various classical languages; an excellent overview is provided by [Ben84]. The correspondence between L, L^N and L^{KR} and first order logic arises as follows. Note that AVSs (that is, Kripke models) can equally well be regarded as models for a certain first order language, namely the first order language (with equality) that contains a binary relation symbol F_l for each R_l, and a unary relation symbol P_α for each Q_α; we will call this language L^1. There is an obvious translation from our modal languages to L^1, the *standard translation*. These are the clauses for L:

$$\begin{array}{lcl} ST(p_\alpha) & = & P_\alpha x \\ ST(\neg\phi) & = & \neg ST(\phi) \\ ST(\phi \vee \psi) & = & ST(\phi) \vee ST(\psi) \\ ST(\langle l \rangle \phi) & = & \exists y(xR_l y \wedge [y/x]ST(\phi)) \end{array}$$

Here x is the first order variable that represents the evaluation node, and the $[y/x]$ in the final clause means substitute y for all free occurrences of x, where y is some fresh first order variable. Note that the standard translation is essentially another way of looking at the satisfiability definition for L, thus it is clear that the standard translation is truth preserving: that is, $M, w \models \phi$ iff $M \models ST(\phi)[w]$. The standard translation shows that L can be regarded as a very simple fragment of L^1, namely a one-free-variable fragment in which only bounded quantification is used.

L^1 is also the first order correspondence language for both L^N and L^{KR}. To see this note that we can extend the standard translation to L^N by adding the following clause:

$$ST(i) \ = \ (x = x_i).$$

Again x is the first order variable that picks out the point of evaluation, and x_i is the first order variable that we have chosen to correspond to the nominal i. Similarly, we can extend the standard translation L^{KR} by adding the clause:

$$ST(\langle l_1 \rangle \cdots \langle l_k \rangle \approx \langle l'_1 \rangle \cdots \langle l'_m \rangle) \ = \ \exists y(xR_{l_1} \ldots R_{l_k} y \ \wedge \ xR_{l'_1} \ldots R_{l'_m} y).$$

Both extensions are truth preserving, thus the use of nominals can be seen as the use of certain extra equalities, while the use of $\approx$ is essentially the use of an additional form of bounded quantification. Thus all three of our base languages are rather small fragments of L^1.

These observations immediately link the modal approach of this chapter with other approaches to Attribute Value logic which may more familiar to the reader. Note in particular that the standard translation links our approach with that of Smolka [Smo89]. Smolka was perhaps the first person to make explicit the connection between AVSs and first order models, and he has proved a number of results concerning a certain first order language of AVSs, namely the language we have here called L^1. Thus, via correspondence theory, many of the results of the present chapter can be seen as an investigation of the complexity of certain fragments of Smolka's language; this includes the results concerning the yet to be introduced universal modality. However the word "many" is important. Modal operators *aren't* restricted to having first order correspondences, and when we later consider the master modality we will in effect be working with a small fragment of infinitary logic.

This completes our discussion of the theoretical background of the chapter. Let's now turn to the issue of most immediate relevance to computational linguistics: the complexity of various satisfiability problems. As most AV grammar formalisms assume a finite collection of both attributes and atomic symbols, the key problem is the satisfiability problem for languages of signature $\langle \mathcal{L}, \mathcal{A} \rangle$ where both $\mathcal{L}$ and $\mathcal{A}$ are *finite*. Actually, with one interesting exception, our results are insensitive to the cardinality of $\mathcal{L}$ for $|\mathcal{L}| \geq 2$, however when we treat the richer languages involving the universal or master modalities extra work is required to show that our results go through for the case of $\mathcal{A}$ finite. In order to minimize the work involved we shall proceed as follows. We will first prove results which hold for languages $|\mathcal{L}| \geq 2$ and $\mathcal{A}$ countably infinite; this allows natural proofs to be given. Later on a very general result is proved *(the Single Variable Reduction Theorem)* which allows all these results to be sharpened to cover languages containing only one propositional variable p. (In fact, in order to give a complete classification of the problem we're even going to show that our results hold for $|\mathcal{L}| \geq 2$ when no propositional variables at all are used; all one needs is a primitive truth symbol $\top$. We will call languages with a primitive $\top$ symbol and no propositional variables languages of signature $\langle \mathcal{L}, \emptyset \rangle$.) Finally, we know of no linguistic theory which puts a fixed finite upper bound on the number of boxlabels that may be used, thus for languages with nominals the complexity of the satisfiability problem when $\mathcal{B}$ is countably infinite is the most important.

5.3 Complexity Results for L, L^N and L^{KR}

In this section we show that the satisfiability problems for L, L^N and L^{KR} are all NP-complete. The fundamental result is that for L, for it turns out that the method used for this language generalizes straightforwardly to its two extensions. The key to the NP completeness result for L is to show that given a formula ϕ which is satisfiable at a node w in some model M, we can always find a suitably small model $M|\mathit{nodes}(\phi, w)$ which also satisfies ϕ. Once we have defined $M|\mathit{nodes}(\phi, w)$ and determined its size the NP completeness result is immediate.

The definition of $M|\mathit{nodes}(\phi, w)$ follows from the following general property of modal languages: when evaluating a wff in a model, only a certain selection of the model's nodes are actually relevant to the truth or falsity of the wff; all other nodes can be discarded. The nodes that are relevant when evaluating a wff ϕ at a node w in a model M are the nodes picked out by the function $\mathit{nodes} : \mathrm{WFF} \times W \longrightarrow \mathit{Pow}(W)$ that satisfies the following conditions:

$$\begin{array}{lcl} nodes(p,w) & = & \{w\} \\ nodes(\neg\phi,w) & = & nodes(\phi,w) \\ nodes(\phi\vee\psi,w) & = & nodes(\phi,w)\cup nodes(\phi,w) \\ nodes(\langle l\rangle\phi,w) & = & \{w\}\cup\bigcup_{w':wR_lw'} nodes(\psi,w') \end{array}$$

Given a model M, a wff ϕ and a node w we form $M|nodes(\phi,w)$ in the obvious way: the nodes of this model are $nodes(\phi,w)$, and the relations and valuation are the restriction of those of M to this set. The following lemma shows that *nodes* selects the correct nodes:

Lemma 5.3.1 (Selection Lemma) *For all models M, all nodes w of M and all wffs ϕ,*

$$M,w\models\phi \;\; \textit{iff} \;\; M|nodes(\phi,w),w\models\phi.$$

Proof. By induction on the degree of ϕ. Note that it follows from the definition of *nodes* that $w\in nodes(\phi,w)$, which is all that is needed to drive the induction through. □

The selection lemma is a completely general fact about modal languages. It *doesn't* depend on any assumptions we have made in this chapter; in particular we haven't yet made use of the fact that we're only concerned with models in which each of the R_l is a partial function. However when we take this into account we notice that $M|nodes(\phi,w)$ has a pleasant property: it is very small. There can only be one more node in $M|nodes(\phi,w)$ than there are occurrences of modalities in ϕ.

Lemma 5.3.2 (Size Lemma) *Let $mod(\phi)$ be the number of occurrences of modalities in ϕ. Then for all models M and all nodes w in M we have that $|nodes(\phi,w)\backslash\{w\}|\leq mod(\phi)$.*

Proof. By induction on the degree of ϕ. For the base case note that for all atomic formulas p we have that $|nodes(p,w)\backslash\{w\}|=\emptyset$, thus the result holds. So assume the result for all wffs of degree less than k. Now if ϕ is a wff of degree k of the form $\psi\vee\theta$ then we have:

$$\begin{array}{lcll} |nodes(\psi\vee\theta,w)\backslash\{w\}| & \leq & |nodes(\psi,w)\backslash\{w\}|+|nodes(\theta,w)\backslash\{w\}| & \\ & \leq & mod(\psi)+mod(\theta) & \text{(by Inductive Hypothesis)} \\ & = & mod(\psi\vee\theta). & \end{array}$$

Thus the required result holds for disjunctions. The case for negations is similar.

There only remains the case for modalities, so suppose that ϕ is a wff of degree k of the form $\langle l\rangle\psi$. We wish to show that $|nodes(\langle l\rangle\psi,w)\backslash\{w\}|\leq mod(\langle l\rangle\psi)$. There are two cases to consider. The first is that there there are no nodes w' such that wR_lw'. But then $|nodes(\langle l\rangle\psi,w)\backslash\{w\}|=\emptyset$ and the result is immediate. So next consider the case when there is a node w' such that wR_lw'. *Note that as we are working with partial functional relations this w' must be unique.* Thus we have the following:

$$\begin{array}{lcll} |nodes(\langle l\rangle\psi,w)\backslash\{w\}| & \leq & |nodes(\psi,w')| & \\ & \leq & |(nodes(\psi,w')\backslash\{w'\})\cup\{w'\}| & \\ & \leq & |nodes(\psi,w')\backslash\{w'\}|+1 & \\ & \leq & mod(\psi)+1 & \text{(by Inductive Hypothesis)} \\ & = & mod(\langle l\rangle\psi) & \end{array}$$

Thus the required result also holds for modalities, and hence the truth of the lemma follows by induction. □

Together the selection lemma and the size lemma lead directly to the main result:

Theorem 5.3.3 *Let L be a language of signature $\langle \mathcal{L}, \mathcal{A} \rangle$ where $|\mathcal{L}| \geq 2$ and $\mathcal{A}$ is countably infinite. Then the satisfiability problem for L is NP-complete.*

Proof. That this satisfiability problem is NP hard is clear, for as we have a countably infinite collection of propositional variables at our disposal the problem contains the satisfiability problem for propositional calculus as a special case. That the problem is in NP follows directly from the fact that any satisfiable L wff ϕ can be satisfied in a model containing at most $mod(\phi) + 1$ nodes; this we know from the selection and size lemmas. Thus, given ϕ we can non-deterministically choose a suitable model of at most this size, and evaluate ϕ in this model in polynomial time. □

Let's turn to the complexity of the satisfiability problem for the language L^N. Recall that this language is L augmented by a distinct new set of atomic symbols called nominals which are constrained to be true at exactly one node in any model. It is easy to use the machinery developed above to prove that the satisfiability problem for L^N is also NP-complete, in fact there is almost nothing new to be done. Given a L^N model M, a node w in M, and an L^N wff ϕ we define $M|nodes(\phi, w)$ exactly as described above. Both the selection and size lemmas hold, thus we are almost through. There is only one snag: $M|nodes(\phi, w)$ is not guaranteed to be an L^N model as some nominals may be not denote any node at all. But this problem is more apparent than real. By adjoining a brand new node (say $*$) to $M|nodes(\phi, w)$ and insisting that all "unassigned nominals" denote $*$ we convert $M|nodes(\phi, w)$ into an L^N model $[M|nodes(\phi, w)]^*$. Of course to maintain the truth of the selection lemma we have to be careful where we place $*$, but there are two obvious "safe" choices. The simplest choice is to insist that $*$ is unrelated (by any of the relations) to any of the points in $M|nodes(\phi, w)$. The second, which is slightly more elegant, is to insist that $*$ is related to w by some relation, but that none of the points in S is related to $*$; choosing this second option means that $*$ point generates $[M|nodes(\phi, w)]^*$. Either way it it clear that the addition of $*$ is harmless: we still have that that $[M|nodes(\phi, w)]^*, w \models \phi$. And $[M|nodes(\phi, w)]^*$ is still small, having at most $mod(\phi) + 2$ nodes. Thus by precisely the same argument as for L we have:

Theorem 5.3.4 *Let L^N be a language with nominals of signature $\langle \mathcal{L}, \mathcal{A}, \mathcal{B} \rangle$, where $|\mathcal{L}| \geq 2$ and both $\mathcal{A}$ and $\mathcal{B}$ are countably infinite. Then the satisfiability problem for L^N is NP-complete.* □

Finally we turn to L^{KR}. The satisfiability problem for this language is also NP-complete, but how are we to show this? Our definition of *nodes* says nothing about occurrences of path equations. Actually the easiest way to proceed is not to extend the definition of *nodes*, but rather to first transform L^{KR} wffs into a certain special form. The following example shows what is involved.

Suppose we have a model M which verifies $\langle a \rangle \approx \langle b \rangle$ at a node w. This means there is a node w' such that $wR_a w'$ and $wR_b w'$. But as $nodes(\langle a \rangle \approx \langle b \rangle, w)$ is undefined, in general we will not have that w' is a part of the small model we build. However if we first rewrite $\langle a\rangle \approx \langle b\rangle$ into a logically equivalent form that makes explicit the existential demands of the path equations, everything proceeds smoothly. Rewrite $\langle a\rangle \approx \langle b\rangle$ as $\langle a\rangle \approx \langle b\rangle \wedge \langle a\rangle\top \wedge \langle b\rangle\top$. Clearly this formula is logically equivalent to the original, however the new syntactic form is very useful: the two new conjuncts make the the modalities $\langle a\rangle$ and $\langle b\rangle$ available to *nodes*. Consider what happens when we apply *nodes* to this new formula at w. As *nodes* commutes over $\wedge$, we must calculate $nodes(\langle a\rangle \approx \langle b\rangle, w)$, $nodes(\langle a\rangle\top, w)$ and $nodes(\langle b\rangle\top, w)$. As before, we can't do anything further with $nodes(\langle a\rangle \approx \langle b\rangle, w)$, but we can evaluate both $nodes(\langle a\rangle\top, w)$ and $nodes(\langle b\rangle\top, w)$, as *nodes* is defined for such expressions. Evaluating these formulas will produce the point w' that we need to build an equivalent small model.

Let's make this precise. Any path equation $\langle A\rangle \approx \langle B\rangle$ is logically equivalent to $\langle A\rangle \approx \langle B\rangle \wedge \langle A\rangle\top \wedge \langle B\rangle\top$. For any path equation $\langle A\rangle \approx \langle B\rangle$ we'll call $\langle A\rangle \approx \langle B\rangle \wedge \langle A\rangle\top \wedge \langle B\rangle\top$ its *explicit form*. Given an L^{KR} wff ϕ which we seek to satisfy, we'll first form a new L^{KR} wff ϕ^* by simultaneously substituting, for each occurrence of a path equation in ϕ, its explicit form. Note that ϕ^* is logically equivalent to ϕ, and that the length of ϕ^* is linear in the length of ϕ. The effect of this rewriting of ϕ means that our existing definition of *nodes* suffices to produce all the points needed for the small model: precisely as illustrated in the above example, when we apply *nodes* the occurrences of the new subformulas of the form $\langle A\rangle\top$ and $\langle B\rangle\top$ ensure that all the needed evaluation points are selected. Thus we can make $M|nodes(\phi, w)$ as before and both the selection and size lemmas hold. So, by exactly the same argument we have that:

Theorem 5.3.5 *Let L^{KR} be a Kasper Rounds language of signature $\langle \mathcal{L}, \mathcal{A}\rangle$ where $|\mathcal{L}| \geq 2$ and $\mathcal{A}$ is countably infinite. Then the satisfiability problem for L^{KR} is NP-complete.* □

In the above proofs was assumed that we had a countably infinite supply of atomic symbols at our disposal. However most Attribute Value formalism use a finite number of atomic symbols. Given that the number of atomic symbols is some fixed finite number, might this not permit us to evade the NP hardness result? (As is well known, for both propositional logic and for S5, such a restriction lowers the complexity of the satisfiability problem to P.) However this is *not* the case here: the satisfiability problem for L (and thus for L^N and L^{KR}) remains NP-hard, even if we use only one propositional variable, and one modal operator. This can be seen as follows. Consider the following set of L formulas: $\{p, \langle a\rangle p, \langle a\rangle\langle a\rangle p, \ldots, \langle a\rangle^k p\}$. The values of these formulas are all independent, that is, for any sequence of truth values $b_0, \ldots, b_k$, there exists a model such that $M \models \langle a\rangle^i p$ iff b_i is true. Now define function f from propositional formulas to L-formulas as follows:

$$f(\phi(p_0, \ldots, p_k)) = \phi(p, \langle a\rangle p, \langle a\rangle\langle a\rangle p, \ldots, \langle a\rangle^k p).$$

Obviously, f is polynomial time computable, and ϕ is satisfiable iff $f(\phi)$ is L satisfiable. Thus, we can summarize the complexity results of this section as follows:

Theorem 5.3.6 *If $|\mathcal{L}| \geq 1$ and $|\mathcal{A}| \geq 1$, the satisfiability problems for L, L^N, and L^{KR} are NP-complete.* □

Actually, if we look at the previous encoding carefully, we can see that if our language contains at least two modalities, we don't need any propositional variables to encode propositional satisfiability in an L formula; all we need is a primitive constant truth symbol $\top$. Define:

$$f(\phi(p_0, \ldots, p_k)) = \phi(\langle b\rangle\top, \langle a\rangle\langle b\rangle\top, \langle a\rangle\langle a\rangle\langle b\rangle\top, \ldots, \langle a\rangle^k\langle b\rangle\top).$$

Obviously, f is polynomial time computable, and ϕ is satisfiable iff $f(\phi)$ is L satisfiable, which leads to the following theorem:

Theorem 5.3.7 *If $|\mathcal{L}| \geq 2$ and $|\mathcal{A}| \geq 0$, the satisfiability problems for L, L^N, and L^{KR} are NP-complete.* □

Let's summarize our results so far. The satisfiability problem for the core AV language L is NP-complete. Adding either of two re-entrancy forcing mechanisms — nominals or the Kasper Rounds path equality — does not increase the complexity: satisfiability remains NP-complete. These results hold even if we have only one modal operator and one atomic symbol at our disposal. There is a result from the literature worth noting here: Kasper and Rounds [KR90] show, using a disjunctive normal form argument, that when attention is confined to those models in which a) each atom is true at at most one node, b) no two atoms are true at the same node, and c) atoms are true only at terminal nodes, then the satisfiability problem for the negation free fragment of L^{KR} is NP hard (and in fact NP-complete). The interesting part of their result is the NP hardness part, for as their language lacks negation this is not obvious. The non-trivial part of our result, on the other hand, is our model theoretic proof that an NP time algorithm exists even if full Boolean expressivity is allowed.

What can be said at a more general level about these results? From the point of view of modal logic they're somewhat unexpected: with the exception of S5 most familiar modal logics are PSPACE-complete. To put it loosely, usually adding modalities to a language of propositional logic makes matters worse, but here it hasn't. The reason, of course, is due to the fundamental constraint on our models, namely that all the relations be partial functional. It's this requirement which enabled us to build small models and thus kept the complexity to that of propositional logic. It's worth adding that this constraint seems to be peculiar to the representational formalisms used in computational linguistics. Various representation formalisms used in AI, such as KL-ONE, can be viewed from a modal perspective, and as Schild [Sch90] has recently observed, terminological logics are also modal logics. But from the point of view of complexity there is a difference: the modal logics inspired by AI typically don't usually obey the partial functionality constraint. Usually they are multimodal versions of K, the modal logic which puts no constraints on accessibility relations. As is well known, the satisfiability problem for this logic is PSPACE-complete [Lad77].

5.4 The Universal Modality

In this section we are going to examine the complexity of the satisfiability problems for three stronger modal languages, $L^\Box$, $L^{N\Box}$ and $L^{KR\Box}$. These languages are, respectively, L, L^N and L^{KR} augmented by the *universal modality*. The universal modality is a modal operator written as $\Box$ which has the following semantics: for all models M, all nodes w, and all wffs ϕ

$$M, w \models \Box\phi \text{ iff } M, w' \models \phi \text{ for all nodes } w' \text{ in } M.$$

That is, $\Box\phi$ holds iff ϕ is true at all nodes. Note that all three enriched languages are fragments of L^1, the first order language of AVSs, as adding the following (truth preserving) clause to the standard translation correctly deals with occurrences of the universal modality:

$$ST(\Box\phi) \;=\; \forall y([y/x]ST(\phi)).$$

For a detailed discussion of the logical consequences of enriching modal languages with the universal modality, see [GP92]. The authors know of only one explicit application of the universal modality to linguistic theorizing, namely Evan's [Eva87] analysis of the feature specification defaults of GPSG, which we shall consider shortly. However, as we shall see, the universal modality seems to have been implicitly used on other occasions.

But why should linguists be interested in $L^{\Box}$, $L^{N\Box}$ and $L^{KR\Box}$? One answer is as follows. Underlying much work in Attribute Value grammar is an idea that can loosely be described as "grammar equals feature logic." Somewhat more precisely, the use of the apparatus of unification formalisms is attractive to many linguists because it enables them to view grammars of natural languages as theories in some sort of calculus of attributes and values. According to such a view, linguistic structure can be adequately modeled by Attribute Value Structures (possibly augmented by the notion of phrase structure), and the linguists' business is to state general constraints about which AVSs are admissible. Such views are discernible in some of the earliest work in attribute value grammar, namely Lexical Functional Grammar (LFG) [KB82]. Generalized Phrase Structure Grammar (GPSG) [GKPS85], explicitly espouses such views, and its work on feature co-occurrence restrictions remains one of the best examples of the approach in action. More recently, Head Driven Phrase Structure Grammar (HPSG)[PS], has taken this approach even further. Whereas in both GPSG and LFG the idea of unification was only one component (albeit an important one) of the systems, in HPSG the unificational apparatus completely dominates.

It is these ideas that motivate the work of the present section. As we have seen the most common unificational formalisms are nothing but modal languages. However as they stand these languages aren't strong enough to express generalizations, and indeed as the "grammar equals feature logic" equation has become more widely accepted, work in Attribute Value grammar has tended to abandon the simple languages we have considered so far in favor of increasingly powerful formalisms. The work of this section is an exploration of the computational consequences of adding just enough power to the base languages to enable generalizations to be expressed.

Let's consider matters more concretely. Suppose we strengthen our languages by adding the universal modality: what linguistic principles can we now express? Consider a typical GPSG feature co-occurrence restriction, for example

$$[\text{VFORM FIN}] \Rightarrow [\text{-N, +V}].$$

This states that if a node has the value FIN for the attribute VFORM, then that node has the properties of being -N and +V. In other words, only a verb can have tense.

The important thing about this constraint is its generality. It's not something which happens to hold of this or that piece of linguistic structure, it's a pervasive fact of English. Any AVS which doesn't satisfy this generalization can't represent an English sentence. We can express this generalization in $L^{\Box}$ as follows:

$$\Box(\langle\textsc{vform}\rangle fin \rightarrow -n \wedge +v).$$

(Here *fin*, $-n$ and $+v$ are propositional symbols and $\langle \textsc{vform} \rangle$ is a modality.) In short we can view the $\Rightarrow$ notation of GPSG as what modal logicians have traditionally called *strict implication.* Viewing $\phi \Rightarrow \psi$ in this way accounts for the generality of feature co-occurrence restrictions.

Evans [Eva87] also makes use of the universal modality in connection with GPSG, but to express defaults, not generalizations. Evans uses $L^{\Box}$ and mostly works with the dual of the universal modality ($\Diamond\phi = \neg\Box\neg\phi$), which he gives an autoepistemic reading: $\Diamond\phi$ means that ϕ is consistent with all known information. For example he uses the wff $\Diamond\langle \textsc{case} \rangle dat \rightarrow \langle \textsc{case} \rangle dat$ to express the feature specification default: "If it is consistent with all known information that case is dative, then case is dative." The idea of using a modal operator to express linguistic defaults is interesting, though we would argue that such an operator would need to be added in addition to the generalization expressing universal modality. But this is to argue over details. There are many ideas worth pursuing in Evans work, and the underlying philosophy is in harmony with that of the present chapter: indeed in a footnote Evans raises the possibility of formalizing all of GPSG in a modal language.

Let's consider the use of $L^{KR\Box}$. This language is powerful enough to capture the content of the Head Feature Convention of GPSG (or indeed HPSG). The essence of the GPSG version is that for any phrasal constituent, the value of its head attribute is shared with the value of the head attribute of its head child. For a discussion of what this terminology means, and why it's linguistically useful the reader is referred to [GKPS85]; here we'll be content to indicate how the constraint can be expressed:

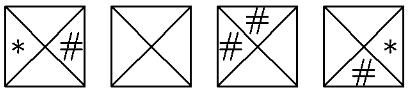
$$\Box(phrasal \rightarrow \langle \textsc{head} \rangle \approx \langle \textsc{head-dtr} \rangle \langle \textsc{head} \rangle).$$

Once again note that this is a strict implication; we could rewrite it as:

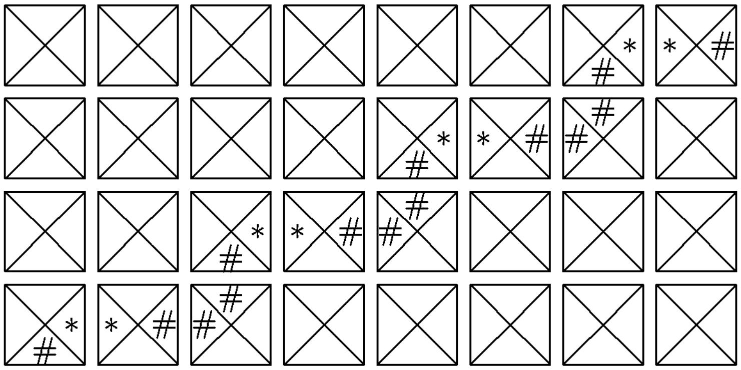
$$phrasal \Rightarrow \langle \textsc{head} \rangle \approx \langle \textsc{head-dtr} \rangle \langle \textsc{head} \rangle.$$

Further experimentation will convince the reader that $L^{KR\Box}$ is a language capable of expressing interesting linguistic constraints. However it has also crossed an important complexity boundary; as we shall now show its satisfiability problem is undecidable. To prove the undecidability result it suffices to give a reduction from a Π^0_1-hard problem to $L^{KR\Box}$ satisfiability. As is shown in [Har83], *tiling problems* provide a particularly elegant method of proving lower bounds for modal logics, so we'll use such an approach here.

A tile T is a 1×1 square fixed in orientation with colored edges *right*(T), *left*(T), *up*(T), and *down*(T) taken from some denumerable set. A tiling problem takes the following form: given a finite set of $\mathcal{T}$ of tile types, can we cover a certain part of $\mathbf{Z} \times \mathbf{Z}$, using only tiles of this type, in such a way that adjacent tiles have the same color on the common edge, and such that the tiling obeys certain constraints? One of the attractive features of tiling problems is that they are very easy to visualize. As an example, consider the following puzzle. Suppose $\mathcal{T}$ consists of the following four types of tile:



Can an 8 by 4 rectangle be tiled with the fourth type placed in the left hand corner? Indeed it can:



There exist complete tiling problems for many complexity classes. In the proof that follows we make use of a certain Π^0_1-complete tiling problem.

Theorem 5.4.1 *If $|\mathcal{L}| \geq 2$, and $\mathcal{A}$ is countably infinite then the satisfiability problem for $L^{KR\Box}$ is Π^0_1-hard.*

Proof. As shown in [Ber66] [Rob71], the following problem is Π^0_1-complete:

N × N tiling: Given a finite set $\mathcal{T}$ of tiles, can $\mathcal{T}$ tile $\mathbf{N} \times \mathbf{N}$?

That is, does there exist a function t from $\mathbf{N} \times \mathbf{N}$ to $\mathcal{T}$ such that:

$$\begin{array}{rcl} \mathit{right}(t(n,m)) & = & \mathit{left}(t(n+1,m)), \text{ and} \\ \mathit{up}(t(n,m)) & = & \mathit{down}(t(n,m+1))? \end{array}$$

Let $\mathcal{T} = \{T_1, \ldots, T_k\}$ be a set of tiles. We construct a formula ϕ such that:

$$\mathcal{T} \text{ tiles } \mathbf{N} \times \mathbf{N} \text{ iff } \phi \text{ is satisfiable.}$$

First of all we will ensure that, if ϕ is satisfiable in M, then M contains a gridlike structure, the nodes of M (henceforth W), play the role of points in a grid, R_r is the right successor relation, and R_u is the upward successor relation. Define:

$$\phi_{grid} = \Box(\langle r\rangle\langle u\rangle \approx \langle u\rangle\langle r\rangle).$$

Suppose $M, w_0 \models \phi_{grid}$. Then there exists a function f from $\mathbf{N} \times \mathbf{N}$ to W such that: $f(0,0) = w_0$, $f(n,m)R_r f(n+1,m)$, and $f(n,m)R_u f(n,m+1)$.

Next we must tile the model. To do this we use propositional variables $t_1, \ldots, t_k$, such that t_i is true at some node w, iff tile T_i is placed at w. To force a proper tiling, we need to satisfy the following three requirements:

1. There is exactly one tile placed at each node.

$$\phi_1 = \Box(\bigvee_{i=1}^{k} t_i \wedge \bigwedge_{1\leq i<j\leq k} \neg(t_i \wedge t_j))$$

2. If T is the tile at w, and T' the tile at the right successor of w, then *right*(T) = *left*(T').

$$\phi_2 = \Box(\bigvee_{\mathit{right}(T_i)=\mathit{left}(T_j)} (t_i \wedge \langle r\rangle t_j))$$

3. If T is the tile at w, and T' the tile at the up successor of w, then $up(T) = down(T')$.

$$\phi_3 = \Box(\bigvee_{up(T_i)=down(T_j)} (t_i \wedge \langle u \rangle t_j))$$

Putting this all together, we define ϕ to be $\phi_{grid} \wedge \phi_1 \wedge \phi_2 \wedge \phi_3$. We will prove that $\mathcal{T}$ tiles $\mathbf{N} \times \mathbf{N}$ iff ϕ is satisfiable.

First suppose $t : \mathbf{N} \times \mathbf{N} \rightarrow \mathcal{T}$ is a tiling of $\mathbf{N} \times \mathbf{N}$. We construct the satisfying model for ϕ as follows: $M = \langle W, R_r, R_u, \pi \rangle$ such that:

$$\begin{array}{lcl} W & = & \{w_{n,m} : n, m \in \mathbf{N}\} \\ R_r & = & \{\langle w_{n,m}, w_{n,m+1} \rangle : n, m \in \mathbf{N}\} \\ R_u & = & \{\langle w_{n,m}, w_{n+1,m} \rangle : n, m \in \mathbf{N}\} \\ \pi(t_i) & = & \{w_{n,m} : n, m \in \mathbf{N} \text{ and } t(n,m) = T_i\} \end{array}$$

Clearly, ϕ holds at any node w of M. To see that the converse also holds, suppose that $M, w_0 \models \phi$. Let f from $\mathbf{N} \times \mathbf{N}$ to W be such that $f(0,0) = w_0$, $f(n,m)R_rf(n+1,m)$ and $f(n,m)R_uf(n,m+1)$. Define the tiling $t : \mathbf{N} \times \mathbf{N} \rightarrow \mathcal{T}$ by $t(n,m) = T_i$ iff $M \models t_i[f(n,m)]$. Note that t is well-defined and total by ϕ_1. Furthermore, if $t(n,m) = T_i$ and $t(n+1,m) = T_j$, then $f(n,m)R_rf(n,m+1)$, $M \models t_i[f(n,m)]$, and $M \models t_j[f(n,m+1)]$. Since M satisfies ϕ_2, we can conclude that $right(T_i) = left(T_j)$. Similarly, if $t(n,m) = T_i$ and $t(n,m+1) = T_j$, then ϕ_3 ensures that $up(T_i) = down(T_j)$. Thus, $\mathcal{T}$ tiles $\mathbf{N} \times \mathbf{N}$. □

Thus the satisfiability problem for $L^{KR\Box}$ is undecidable. Note, however, that the above proof depends on having access to an unlimited supply of propositional variables. (The above argument shows how any tiling problem can be reduced to $L^{KR\Box}$ satisfiability by representing tiles as propositional symbols. But there is no pre-determined size limit on the set of tiles $\mathcal{T}$ that we may be given.) This problem will be dealt with later.

The satisfiability problem for $L^{KR\Box}$ is in fact Π^0_1-complete. Given the previous result, all we need to to show is that the $L^{KR\Box}$ validities can be recursively enumerated. One way of doing this is to give a recursive axiomatization of $L^{KR\Box}$. This can be done by building on the completeness proof for L^{KR} given in [Bla2], but it has the drawback of requiring the introduction of the (otherwise irrelevant) machinery of modal completeness theory. Fortunately correspondence theory comes to the rescue with a general argument showing (at least for the case of finite $\mathcal{L}$) that $L^{KR\Box}$ validity is a r.e. notion. The argument is due to van Benthem [Ben84, page 175] who observes that when working with *elementary* classes of frames (that is, frames defined by a single L^1 formula) it is not necessary to give an explicit axiomatization to show that modal validity is r.e.: if φ is the L^1 wff that defines the elementary class, and if ϕ is a modal formula such that $ST(\phi) \in L^1$ then ϕ is a validity iff $\varphi \models \forall x ST(\phi)$. But here "$\models$" denotes the first order consequence relation, and as this is an r.e. relation we would be through if we could show that the multiframes underlying our Kripke models form an elementary class. This is trivial: we are working with the class of multiframes that are partial functional. Given that $\mathcal{L}$ is finite we need merely define:

$$\varphi = \bigwedge_{l \in \mathcal{L}} \forall xyz(xR_ly \wedge xR_lz \rightarrow y = z).$$

Thus we are working with an elementary class, namely the class that satisfies φ. Thus we conclude:

Theorem 5.4.2 *If $|\mathcal{L}| \geq 2$ and $\mathcal{A}$ is countably infinite then the satisfiability problem for $L^{KR\Box}$ is Π^0_1-complete.* □

What are we to make of this undecidability result? The key technical point is that it is genuinely due to the interaction between the ability to state generalizations and the ability to enforce re-entrancy. The subsequent results elaborate on this theme and reveal an interesting difference between $L^{N\Box}$ and $L^{KR\Box}$. We begin by showing, using a filtration argument (see Fisher and Ladner [FL79] for filtrations in Propositional Dynamic Logic), that the satisfiability problems for $L^{\Box}$ and $L^{N\Box}$ are decidable.

Theorem 5.4.3 *If ϕ is a satisfiable $L^{\Box}$ or $L^{N\Box}$ formula, then ϕ is satisfiable in a model with at most $2^{2|\phi|}$ nodes.*

Proof. Suppose that ϕ is an $L^{\Box}$ wff, $M = \langle W, \{R_l\}_{l\in\mathcal{L}}, \pi\rangle$, and $M, w_0 \models \phi$. Let $Cl(\phi)$ be the smallest set that contains ϕ, and is closed under subformulas and single negations. Define an equivalence relation $\sim$ on W as follows:

$$w \sim w' \text{ iff } \forall\psi \in Cl(\phi)(M, w \models \psi \Leftrightarrow M, w' \models \psi).$$

Let $W^F \subseteq W$ be such that W^F contains exactly one element from each equivalence class. Let π^F be the restriction of π to W^F, and define R^F_l as follows:

$$wR^F_l w' \text{ iff } \exists w''(wR_l w'' \wedge w' \sim w'').$$

Let $M^F = \langle W^F, \{R^F_l\}_{l\in\mathcal{L}}, \pi^F\rangle$. M^F is a filtration of M through $Cl(\phi)$ in the sense of Hughes and Cresswell [HC84], thus it follows immediately that M^F satisfies ϕ. Since the size of $Cl(\phi)$ is at most $2|\phi|$, the size of W^F is bounded by $2^{2|\phi|}$. Furthermore, M^F is an $L^{\Box}$ model, since the definition of R^F_l ensures that R^F_l is a partial function for any modality l.

Essentially the same argument works for wffs ϕ of $L^{N\Box}$. We need only observe that for all nominals i in $Cl(\phi)$, if $\pi(i) = \{w\}$ then $w \sim w'$ iff $w' = w$. In short, all nominals in $Cl(\phi)$ denote singletons in the filtrations, and all other nominals can be assigned arbitrary singletons of W^F, thus we again have a small model for ϕ. □

From theorem 5.4.3, it follows immediately that the satisfiability problems for $L^{\Box}$ and $L^{N\Box}$ are both decidable in nondeterministic exponential time. But we can improve these results. Using methods similar to [Pra79] and [HM85] we sketch a construction of a deterministic exponential time algorithm for both $L^{\Box}$ and $L^{N\Box}$ satisfiability.

Theorem 5.4.4 *The satisfiability problems for $L^{\Box}$ and $L^{N\Box}$ are decidable in EXPTIME.*

Proof. Let $Cl(\phi)$ be defined as in the proof of the previous theorem. Let S be the set of all subsets Γ of $Cl(\phi)$ that are maximally propositionally consistent, and are closed under reflexivity of $\Box$; that is, if $\Box\psi \in \Gamma$ then ψ is also in Γ. Suppose ϕ is satisfiable in model M. Let S_M be the set of subsets of $Cl(\phi)$ that actually occur in M, that is, $S_M = \{\Gamma \in S : M, w \models \Gamma$, for some $w \in M\}$. Obviously, $S_M \subseteq S$, but we can say more about S_M. First of all, note that every element of S_M contains the same $\Box$ formulas. Furthermore, if ϕ contains a nominal m, there is exactly one set in S_M that contains m. Let $\Sigma \subseteq Pow(S)$, consisting of all maximal $S' \subseteq S$ such that:

1. $\forall \Gamma, \Gamma' \in S', \forall \Box\psi \in Cl(\phi) : \Box\psi \in \Gamma \Leftrightarrow \Box\psi \in \Gamma'$, and

2. For every nominal m occurring in ϕ, there is exactly one set $\Gamma \in S'$ such that $m \in \Gamma$.

If ϕ is satisfiable in M, then there exists a set $S' \in \Sigma$ such that $S_M \subseteq S'$. What can we say about the size of Σ? Since $Cl(\phi)$ contains at most $2|\phi|$ elements, there exist at most $2^{2|\phi|}$ maximal sets $\widehat{S} \subseteq S$ fulfilling the first condition. If ϕ contains k nominals, at most $|\widehat{S}|^k$ subsets of $\widehat{S}$ occur in Σ. Since k is bounded by $|\phi|$, the size of Σ is exponential in the length of ϕ.

For every $S_1 \in \Sigma$, we will construct a sequence of sets $S_1 \supset S_2 \supset S_3 \supset \cdots$ such that: if ϕ is satisfiable in a model M, and $S_M \subseteq S_1$, then $S_M \subseteq S_i$.

Suppose we have defined S_i. Call a set $\Gamma \in S_i$ inconsistent iff one of the following situations occurs:

1. $\neg\Box\psi \in \Gamma$, but for all $\Gamma' \in S_i$: $\psi \in \Gamma'$, or

2. For some modality l, $\langle l \rangle \psi \in \Gamma$ for some ψ, but there is *no* $\Gamma' \in S_i$ such that $\forall \langle l \rangle \xi \in Cl(\phi)(\langle l \rangle \xi \in \Gamma \Leftrightarrow \xi \in \Gamma')$.

If there are inconsistent sets in S_i, then we let S_{i+1} consist of all sets of S_i that are *not* inconsistent. Otherwise, ϕ is satisfiable iff $\phi \in \Gamma$ for some set $\Gamma \in S_i$, and for every nominal m occurring in ϕ, there is exactly one set $\Gamma \in S_i$ that contains m.

Since S_1 is of exponential size, and S_{i+1} is strictly included in S_i, the algorithm terminates after at most exponentially many cycles. Determining which sets in S_i are inconsistent takes polynomial time in the length of the representation of S_i. Thus, for every member of Σ, the algorithm takes at most deterministic exponential time. Since Σ is of exponential size, we can determine if if ϕ is satisfiable in EXPTIME. □

However as the next result shows, there is a clear sense in which this result cannot be improved.

Theorem 5.4.5 *The satisfiability problems for $L^{\Box}$ and $L^{N\Box}$ are EXPTIME-complete for $|\mathcal{L}| \geq 2$, and $\mathcal{A}$ countably infinite.*

Proof. The upper bounds follows from theorem 5.4.4. To prove the corresponding lower bounds, it suffices to give a polynomial time computable reduction from an EXPTIME-hard set to $L^{\Box}$ satisfiability. We will use a suitable subset of Propositional Dynamic Logic. Let PDL$(a,*)$ be the bimodal propositional language with modalities $\langle a \rangle$ and $\langle a^* \rangle$. We interpret wffs of PDL$(a,*)$ on Kripke models $M = \langle W, R_a, \pi \rangle$, where R_a is an arbitrary binary relation on W, in the usual way, the key clause being:

$$M, w \models \langle a^* \rangle \phi \ \text{ iff } \ \exists w'(w R^*_a w' \ \&\ M, w' \models \phi).$$

where R^*_a denotes the reflexive, transitive closure of R_a. In [FL79], it is proven that the satisfiability problem for PDL$(a,*)$ is EXPTIME-hard. In fact, from careful inspection of this proof, we can conclude that even the following set is EXPTIME-hard: Let C consist of all PDL$(a,*)$ formulas ϕ such that: $\phi = \phi_1 \wedge [a^*]\phi_2$, and

1. ϕ_1, ϕ_2 are $*$-less and have modal depth ≤ 1,

2. ϕ is satisfiable in a model where every node has at most two successors.

Define the reduction f from C to $L^{\Box}$ satisfiability as follows:

1. If ϕ is *not* of the form $\phi_1 \wedge [a^*]\phi_2$, where ϕ_1 and ϕ_2 are $*$-less and of modal depth ≤ 1, then $f(\phi) = \bot$
2. For ϕ_1, ϕ_2 $*$-less and of modal depth ≤ 1, $f(\phi_1 \wedge [a^*]\phi_2) = s(\phi_1) \wedge \Box s(\phi_2)$, where s is defined on $*$-less formulas as follows:

$$\begin{array}{lcl} s(p) & = & p \\ s(\neg\psi) & = & \neg s(\psi) \\ s(\psi \vee \xi) & = & s(\psi) \vee s(\xi) \\ s(\langle a\rangle\psi) & = & \langle a_1\rangle s(\psi) \vee \langle a_2\rangle s(\psi) \end{array}$$

Since s is polynomial time computable on $*$-less formulas of modal depth ≤ 1, f is polynomial time computable. Now, it is straightforward to prove the following fact by induction. If $M = \langle W, R_a, \pi\rangle$ is a PDL-model, and $M' = \langle W, R_{a_1}, R_{a_2}, \pi\rangle$ is an $L^{\Box}$-model, such that $R_a = R_{a_1} \cup R_{a_2}$, then for all $*$-less PDL(a)-formulas ϕ, and for all nodes $w \in W$, $M, w \models \phi$ iff $M', w \models s(\phi)$. By making use of this it is easy to prove that f is indeed a reduction from C to $L^{\Box}$ satisfiability. □

Note that once again this reduction depends on having an unlimited supply of propositional variables. The following theorem will dispose of this issue once and for all:

Theorem 5.4.6 (Single variable reduction theorem) *If $|\mathcal{L}| \geq 1$, then there exist polynomial time reductions from the satisfiability problems for $L^{\Box}$ and $L^{KR\Box}$ over signature $\langle \mathcal{L}, \mathcal{A}\rangle$ to the corresponding satisfiability problems over signature $\langle \mathcal{L}, \{p\}\rangle$.*

Proof. Recall that we used the following reduction from propositional satisfiability to L satisfiability over signature $\langle \{a\}, \{p\}\rangle$ in theorem 5.3.6:

$$f(\phi(p_0, \ldots, p_k)) = \phi(p, \langle a\rangle p, \langle a\rangle\langle a\rangle p, \ldots, \langle a\rangle^k p).$$

If ϕ is satisfied in w, we build the corresponding model for $f(\phi)$ by replacing w by a sequence of nodes $w_0 R'_a w_1 R'_a \ldots R'_a w_k$ such that p is true in w_i iff p_i is true in w. We will use a similar encoding to to prove the theorem. Fix a signature $\langle \mathcal{L}, \mathcal{A}\rangle$, $\mathcal{L} \neq \emptyset$. We'll use a fixed element $a \in \mathcal{L}$ to encode worlds. Suppose $M = \langle W, \{R_l\}_{l \in \mathcal{L}}, \pi\rangle$ is a model, and we use propositional variables $p_0, \ldots, p_k$. As a first attempt to obtain an equivalent model with one propositional variable, look at the encoding given above: replace each world w by a sequence of worlds $w_0 R'_a w_1 R'_a \ldots R'_a w_k$ such that p is true in w_i iff p_i is true in w. This doesn't quite work: consider for instance the formula $\Box p_1$. The obvious translation would be $\Box\langle a\rangle p$. But this would mean that $\langle a\rangle p$ has to be satisfied in *every* world w_i. This is too strong a requirement: we just want $\langle a\rangle p$ to be satisfied in every world of the form w_0. We therefore need to be able to determine if we are at a world of the form w_0. We can't use a propositional variable for this: we have already used our sole propositional variable p. The solution is to use a slightly different encoding: we will replace each world w by a list of $2k+3$ worlds $w_0 R'_a w_1 R'_a \ldots R'_a w_{2k+2}$ such that: p is true in w_i iff either $i \leq k$ and p_i is true in w, or $i = 2k+2$. Define:

$$\sigma_{0,k} = \bigwedge_{i=k+1}^{2k+1} \langle a\rangle^i \neg p \wedge \langle a\rangle^{2k+2} p.$$

Then $\sigma_{0,k}$ is true in every world w_0, and we will ensure that for every $i > 0$, $\sigma_{0,k}$ is false in w_i. Now we will show how to define the relations R'_l. If $l \neq a$, this is easy: we let R'_l consist of all pairs $\langle w_0, w'_0\rangle$ such that $\langle w, w'\rangle \in R_l$. We can't do this for R'_a, since every world w_0 already has w_1 as its R'_a successor. If $\langle w, w'\rangle \in R_a$, we will add $\langle w_{2k+2}, w'_0\rangle$ to R'_a, that is, we add an R'_a edge from the last node of the encoding of w to the first node of the encoding of w'.

Now we are ready to define the reduction:

$$f(\phi(p_0, \ldots, p_k)) = \sigma_{0,k} \wedge g_k(\phi).$$

Where g_k is inductively defined as follows:

$$\begin{array}{lcl}
g_k(p_i) & = & \langle a\rangle^i p \\
g_k(\neg\psi) & = & \neg g_k(\psi) \\
g_k(\psi_1 \vee \psi_2) & = & g_k(\psi_1) \vee g_k(\psi_2) \\
g_k(\langle l\rangle\psi) & = & \langle l\rangle(\sigma_{0,k} \wedge g_k(\psi)) \text{ for } l \neq a \\
g_k(\langle a\rangle\psi) & = & \langle a\rangle^{2k+3}(\sigma_{0,k} \wedge g_k(\psi)) \\
g_k(\Box\psi) & = & \Box(\sigma_{0,k} \rightarrow g_k(\psi)) \\
g_k(\langle A\rangle \approx \langle B\rangle) & = & \langle A\rangle \approx \langle B\rangle[\langle a\rangle := \langle a\rangle^{2k+3}] \wedge g_k(\langle A\rangle T) \wedge g_k(\langle B\rangle T)
\end{array}$$

(The notation $[\langle a\rangle := \langle a\rangle^{2k+3}]$ denotes the result of substituting $\langle a\rangle^{2k+3}$ for $\langle a\rangle$.) Obviously, f is polynomial time computable. Furthermore, if ϕ does not contain path formulas, then neither does $f(\phi)$. It remains to prove that ϕ is satisfiable iff $f(\phi)$ is satisfiable.

Let $M = \langle W, \{R_l\}_{l\in\mathcal{L}}, \pi\rangle$. Define the corresponding model $M_k = \langle \widehat{W}, \{\widehat{R}_l\}_{l\in\mathcal{L}}, \widehat{\pi}\rangle$ as follows:

$$\begin{array}{lcl}
\widehat{W} & = & \{w \in W : M \models \sigma_{0,k}\} \\
\widehat{R}_l & = & R_l|\widehat{W} \text{ for } l \neq a \\
\widehat{R}_a & = & (R_a)^{2k+3}|\widehat{W} \\
\widehat{\pi}(p_i) & = & \{w : M \models \langle a\rangle^i p\}
\end{array}$$

With induction the structure of ψ, it is easy to prove that for all formulas ψ with propositional variables in $\{p_0, \ldots, p_k\}$, and for all $w \in \widehat{W}$:

$$M, w \models g_k(\psi) \text{ iff } M_k, w \models \psi.$$

Now suppose $M, w \models f(\phi)$. Then $w \in W'$, since $M, w \models \sigma_{0,k}$. Therefore, $M_k, w \models \phi$, and hence ϕ is satisfiable.

For the converse, suppose that ϕ is satisfiable. Let $M = \langle W, \{R_l\}_{l\in\mathcal{L}}, \pi\rangle$ be a model such that $M \models \phi[v]$. Let $M' = \langle W', \{R'_l\}_{l\in\mathcal{L}}, \pi'\rangle$ be the corresponding model with one propositional variable, as sketched before the definition of the reduction:

$$\begin{array}{lcl}
W' & = & \{w_0, \ldots, w_{2k+2} : w \in W\} \\
R'_l & = & \{\langle w_0, w'_0\rangle : wR_lw'\} \text{ (for } l \neq a) \\
R'_a & = & \{\langle w_i, w_{i+1}\rangle : i \leq 2k+1\} \cup \{\langle w_{2k+2}, w'_0\rangle : wR_aw'\} \\
\pi'(p) & = & \{w_i : i = 2k+2 \text{ or } (w \in \pi(p_i) \text{ and } i \leq k)\}
\end{array}$$

It is easy to see that M'_k is isomorphic to M, and therefore $M' \models \sigma_{0,k} \wedge g(\phi)[v_0]$. □

As in theorem 5.3.7, we can prove that if $\mathcal{L}$ contains at least two modalities, we can dispense with propositional variables all together. Recall that we used the following reduction in theorem 5.3.7:

$$f(\phi(p_0, \ldots, p_k)) = \phi(\langle b\rangle\top, \langle a\rangle\langle b\rangle\top, \langle a\rangle\langle a\rangle\langle b\rangle\top, \ldots, \langle a\rangle^k\langle b\rangle\top).$$

We can strengthen this. It is easy to see that the techniques of the previous theorem can be applied to prove the analog of theorem 5.3.7. We leave the details to the reader.

Theorem 5.4.7 *If $|\mathcal{L}| \geq 2$, then there exist polynomial time reductions from the satisfiability problems for $L^{\Box}$ and $L^{KR\Box}$ over signature $\langle \mathcal{L}, \mathcal{A} \rangle$ to the corresponding satisfiability problems over signature $\langle \mathcal{L}, \emptyset \rangle$.* □

Combining the previous theorem with the earlier obtained lower bounds, we can summarize the complexity results of this section as follows:

Corollary 5.4.8 *If $|\mathcal{L}| \geq 2$, and $|\mathcal{A}| \geq 0$ the satisfiability problems for $L^{\Box}$ and $L^{N\Box}$ are EXPTIME-complete, and the satisfiability problem for $L^{KR\Box}$ is Π^0_1-complete.* □

An interesting aspect of the results of this section is the wedge they drive between $L^{N\Box}$ and $L^{KR\Box}$. At first sight the difference seems puzzling: after all, both are languages in which generalizations can be stated and re-entrancy forced. A closer look shows that the two languages work very differently. We might say that whereas in $L^{KR\Box}$ we can state genuine generalizations involving re-entrancy, in $L^{N\Box}$ there is a clear sense in which re-entrancy is only expressed within a given model. $L^{N\Box}$ isn't powerful enough to force labelings. An example will make this clear. Consider the GPSG head feature convention again. We've already seen that its force is captured in $L^{KR\Box}$ by the following wff:

$$\Box(\mathit{phrasal} \rightarrow \langle \textsc{head} \rangle \approx \langle \textsc{head-dtr} \rangle \langle \textsc{head} \rangle).$$

But when we attempt to capture its force using nominals we run into a problem: how can we label the desired re-entrancy point? It seems we must step beyond the boundaries of $L^{N\Box}$ and write an expression such as the following:

$$\Box(\mathit{phrasal} \rightarrow \exists i(\langle \textsc{head} \rangle i \wedge \langle \textsc{head-dtr} \rangle \langle \textsc{head} \rangle i)).$$

Now, this expression clearly captures what is required, but unfortunately it's not an $L^{N\Box}$ wff but a wff of a more powerful language in which explicit quantification over nominals is possible. Such languages have been investigated before; in fact Bull's paper on the subject seems to have been the first technical investigation of nominals [Bul70]. Moreover Reape [Rea91] has used such language to investigate problems in unification based grammar. However when used together with the universal modality, explicit quantification over nominals is (from the point of view of complexity theory at any rate) rather uninteresting: it is straightforward to show that strengthening $L^{N\Box}$ to allow explicit quantification over nominals results in a notational variant of L^1, the first order language of AVSs. Such a language thus has a Π^0_1 satisfiability problem, just as $L^{KR\Box}$ does.

In short, it is asking a lot to be able to express generalizations involving re-entrancy. The nearest we can get to it in a decidable framework seems to be $L^{N\Box}$. However, while generalizations are expressible in this language, these generalizations don't involve re-entrancy in any strong sense. It's precisely for this reason that we're not able to force a tiling in this language, but (alas) it's also precisely for this reason that it is not able express some linguistically useful principles such as the head feature convention.

5.5 The Master Modality

In this section we consider the complexity of the satisfiability problems for $L^{[*]}$, $L^{N[*]}$ and $L^{KR[*]}$, our base languages extended with the master modality $[*]$. Gazdar *et al.* [GPC+88] define the master modality as follows:

$$\begin{array}{lll} M, w \models [*]\phi & \text{iff} & M, w \models \phi \text{ and} \\ & & M, w' \models [*]\phi, \text{ for all } w' : wR_l w', \text{ for some } l \in \mathcal{L}. \end{array}$$

As they only work with finite AVSs this definition is not circular, indeed it has the advantage of making the intended use of $[*]$ particularly clear: $[*]$ expresses recursive constraints over AVSs. (See Carpenter [Car92] for a discussion of recursive constraints.) However it will make the following technicalities more straightforward if we extend the definition to cover arbitrary AVSs. We do this as follows.

$$M, w \models [*]\phi \text{ iff } M, w' \models \phi, \text{ for all } w' \in W \text{ such that } w(\bigcup_{l \in \mathcal{L}} R_l)^* w'$$

That is, ϕ must be satisfied at all nodes w' that are reachable by any finite sequence of transitions (including the null transition) from w. Clearly this definition reduces to the previous one for finite AVSs. It's also worth mentioning that we have introduced a notational change; Gazdar *et al.* use $\Box$ for the master modality. We prefer to reserve this for the universal modality.

The most important thing to note about both semantic definitions given above is their infinitary force: L^1 is *not* the correspondence language for $[*]$. As with PDL, the natural correspondences are with classical languages in which infinite disjunctions are allowed; in effect we are working with a fragment of infinitary logic.

A number of logical results for $L^{[*]}$, including the construction of a complete tableaux system, have been proved by Kracht [Kra89]. However his methods only yield a nondeterministic exponential time upper bound for the satisfiability problem; we improve on this below. Neither $L^{N[*]}$ nor $L^{KR[*]}$ seem to have been treated in the literature, though Gazdar *et al.* note that some re-entrancy coding mechanism would be desirable, and Kasper and Rounds mention the possibility of combining the two approaches. $L^{KR[*]}$ is this combination.

We begin our investigation with a lemma which enables us to utilize results from the previous section.

Lemma 5.5.1 *Let ϕ be a formula that contains no occurrences of $\Box$ or $[*]$. Then $\Box\phi$ is satisfiable iff $[*]\phi$ is satisfiable.*

Proof. First suppose $M = \langle W, \{R_l\}_{l \in \mathcal{L}}, \pi \rangle$, and $M, w_0 \models \Box\phi$. Then for all $w \in W$, $M, w \models \phi$, and therefore certainly $M, w_0 \models [*]\phi$.

Conversely suppose $M = \langle W, \{R_l\}_{l \in \mathcal{L}}, \pi \rangle$, and $M, w_0 \models [*]\phi$. Let W' equal $\{w \in W : w_0(\bigcup_{l \in \mathcal{L}} R_l)^* w\}$, and let M' be the restriction of M to W'. It follows by the usual generated submodel argument that for all formulas ψ without $\Box$ or $[*]$, and for all $w \in W'$: $M, w \models \psi$ iff $M', w \models \psi$. It follows that $M, w \models \phi$, for all $w \in W'$. But then $M', w_0 \models \Box\phi$. $\Box$

From this lemma, and the form of the reductions in the proofs of theorems 5.4.1 and 5.4.5, it follows immediately that the lower bounds for languages with $\Box$ go through for the corresponding languages with $[*]$:

Corollary 5.5.2 *The satisfiability problems for $L^{[*]}$ and $L^{N[*]}$ are EXPTIME-hard. The satisfiability problem for $L^{KR[*]}$ is Π^0_1-hard.* □

But do we have the the same upper bounds? The answer is almost always "yes," but there is one notable exception. If $\mathcal{L}$ is *finite*, and contains at least two elements, the complexity of the satisfiability problem for $L^{KR[*]}$ is much higher than that of the corresponding satisfiability problem for $L^{KR\Box}$: we will show that in this case $L^{KR[*]}$ satisfiability is Σ^1_1-complete instead of "just" Π^0_1-complete.

Lemma 5.5.3 *If ϕ is satisfiable in M, then ϕ is satisfiable in a* countable *submodel of M.*

Proof. Suppose $M, w \models \phi$. Let W' be $\{w' \in W : w(\bigcup_{l \in \mathcal{L}} R_l)^* w'\}$. It follows by induction on the degree of ϕ that $M|W', w \models \phi$. But as all our relations are partial functions, and as we only have countably many of them, W' must be countable. □

Theorem 5.5.4 *If $\mathcal{L}$ is finite, and $|\mathcal{L}| \geq 2$, the satisfiability problem for $L^{KR[*]}$ is Σ^1_1-complete.*

Proof. The upper bound follows directly from lemma 5.5.3. To prove the corresponding lower bound, we will construct a reduction from the following Σ^1_1-complete tiling problem [Har86]:

> **N × N recurrent tiling**: Given a finite set $\mathcal{T}$ of tiles, and a tile $T_1 \in \mathcal{T}$, can $\mathcal{T}$ tile $\mathbf{N} \times \mathbf{N}$ such that T_1 occurs in the tiling infinitely often on the first row.

That is, does there exist a function t from $\mathbf{N} \times \mathbf{N}$ to $\mathcal{T}$ such that: $\mathit{right}(t(n,m)) = \mathit{left}(t(n+1,m))$, $\mathit{up}(t(n,m)) = \mathit{down}(t(n,m+1))$, and the set $\{i : t(i,0) = T_1\}$ is infinite?

Let $\mathcal{T} = \{T_1, \ldots, T_k\}$ be a set of tiles. We construct a formula ϕ_{rt} such that:

$$\langle \mathcal{T}, T_1 \rangle \in \mathbf{N} \times \mathbf{N} \text{ recurrent tiling} \quad \text{iff} \quad \phi_{rt} \text{ is satisfiable.}$$

To ensure that ϕ_{rt} forces a tiling of $\mathbf{N} \times \mathbf{N}$, we use the formula ϕ constructed in the proof of theorem 5.4.1. Let ϕ' be the result of replacing every occurrence of $\Box$ by $[*]$ in ϕ. Then, as in theorem 5.4.1, the following holds:

1. If ϕ' is not satisfiable, then $\mathcal{T}$ does not tile $\mathbf{N} \times \mathbf{N}$.

2. If $M, w_0 \models \phi'$, then there exists a tiling t of $\mathbf{N} \times \mathbf{N}$, and a function f from $\mathbf{N} \times \mathbf{N}$ to W be such that $f(0,0) = w_0$, $f(n,m) R_r f(n+1,m)$ and $f(n,m) R_u f(n,m+1)$, and $M, f(n,m) \models t_i$ iff $t(n,m) = T_i$.

Now we force the recurrence: we will use a new propositional variable row_0, which can only be true at worlds of the form $f(n,0)$, and we will ensure that there exist an infinite number of worlds where row_0 holds and tile T_1 is placed. Define:

$$\phi_{rec} = ([*] \bigwedge_{l \in \mathcal{L}, l \neq r} [l][*]\neg row_0) \wedge row_0 \wedge [*](row_0 \rightarrow \langle * \rangle (row_0 \wedge t_1)).$$

Let ϕ_{rt} be the conjunction of ϕ' and ϕ_{rec}. In the same way as in theorem 5.4.1, we can now prove that $\langle \mathcal{T}, T_1 \rangle \in \mathbf{N} \times \mathbf{N}$ recurrent tiling iff ϕ_{rt} is satisfiable. □

In the previous proof it is essential that we can force a propositional variable to be true at w only if w is reachable from w_0 in a finite number of R_r steps. We can't force this in $L^{KR\Box}$, nor in $L^{KR[*]}$ if $\mathcal{L}$ is infinite. (Indeed the previous proof doesn't go through for $\mathcal{L}$ infinite as then ϕ_{rec} is not a formula.) As we shall now see, in the case where $\mathcal{L}$ is infinite, the satisfiability problem for a language with $[*]$ is never more complex than the satisfiability problem for the corresponding language with $\Box$.

Theorem 5.5.5 *If $\mathcal{L}$ is infinite, then*

1. *The satisfiability problems for $L^{[*]}$ and $L^{N[*]}$ are EXPTIME-complete.*

2. *The satisfiability problem for $L^{KR[*]}$ is Π^0_1 complete.*

Proof. The lower bounds follow from corollary 5.5.2. For the upper bounds, we will reduce the satisfiability problems for $L^{[*]}, L^{N[*]}$, and $L^{KR[*]}$ to the satisfiability problems for the corresponding languages with $\Box$. The claim then follows from theorems 5.4.5 and 5.4.1. To get rid of occurrences of $[*]$, we define function g from $\Box$-less formulas to formulas without $\Box$ or $[*]$ as follows:

$$\begin{array}{lll} g(p) = p & g(\neg\psi) = \neg g(\psi) & g(\psi \vee \xi) = g(\psi) \vee g(\xi) \\ g(\langle l \rangle \psi) = \langle l \rangle g(\psi) & g([*]\psi) = p_{[*]\psi} & g(\langle A \rangle \approx \langle B \rangle) = \langle A \rangle \approx \langle B \rangle \end{array}$$

We have to ensure that $p_{[*]\psi}$ mimics the behavior of $[*]\psi$. In particular, if $\neg p_{[*]\psi}$ holds at some world, this world should have a (multi-step) successor where $g(\neg\psi)$ holds. We introduce new modalities $\langle\neg\psi\rangle$ for all formulas $[*]\psi \in Cl(\phi)$, and we will force that for every world w satisfying $\neg p_{[*]\psi}$, there exists a world w' such that $wR_{\neg\psi}w'$ and $g(\neg\psi)$ holds at w'. Let $\mathcal{L}'$ consist of the modalities occurring in ϕ, and the new modalities $\langle\neg\psi\rangle$ for $[*]\psi \in Cl(\phi)$. Since $\mathcal{L}$ is infinite, we may assume that $\mathcal{L}' \subset \mathcal{L}$. Our reduction f is defined as follows:

$$f(\phi) = g(\phi) \wedge \Box(p_{[*]\psi} \rightarrow g(\psi) \wedge \bigwedge_{l \in \mathcal{L}'} [l] p_{[*]\psi}) \wedge \Box(\neg p_{[*]\psi} \rightarrow \langle\neg\psi\rangle g(\neg\psi)).$$

Obviously, f is polynomial time computable. Furthermore, if ϕ doesn't contain nominals and/or path equations, then neither does $f(\phi)$. It remains to prove that ϕ is satisfiable iff $f(\phi)$ is satisfiable.

First suppose ϕ is satisfiable. By lemma 5.5.3, there exist a countable model $M = \langle W, \{R_l\}_{l \in \mathcal{L}}, \pi\rangle$, and a world $w_0 \in W$ such that $M, w_0 \models \phi$. Define a model $\widehat{M}$ as follows: $\widehat{M} = \langle W, \{\widehat{R}_l\}_{l \in \mathcal{L}}, \widehat{\pi}\rangle$, such that:

1. $\widehat{R}_l = R_l$ for l occurring in ϕ; $\widehat{R}_l = \emptyset$ for $l \notin \mathcal{L}'$

2. For $[*]\psi \in Cl(\phi)$, $\widehat{R}_{\neg\psi}$ is such that:

   a. $w\widehat{R}_{\neg\psi}w' \Rightarrow M \models \neg\psi$, and $w(\bigcup_{l \in \mathcal{L}} R_l)^* w'$; and

   b. $\exists w' : w\widehat{R}_{\neg\psi}w'$ iff $M \models \neg[*]\psi$.

3. $\widehat{\pi}(p) = \pi(p)$ for p occurring in ϕ; $w \in \widehat{\pi}(p_{[*]\psi})$ iff $M, w \models [*]\psi$.

Obviously, if $M, w_0 \models \phi$, then $\widehat{M}$ is well defined, and $\widehat{M}, w_0 \models f(\phi)$.

For the converse, suppose $f(\phi)$ is satisfiable. let $M = \langle W, \{R_l\}_{l \in \mathcal{L}}, \pi \rangle$, and $w_0 \in W$ be such that $M, w_0 \models f(\phi)$. We may assume that $R_l = \emptyset$ for $l \notin \mathcal{L}$. It is easy to prove that for all formulas $\psi \in Cl(\phi)$ and for all $w \in W$, $M, w \models \phi$ iff $M, w \models g(\phi)$, and thus ϕ is indeed satisfiable. □

It remains to prove EXPTIME upper bounds for $L^{[*]}$ and $L^{N[*]}$ for finite $\mathcal{L}$.

Theorem 5.5.6 *If $\mathcal{L}$ is finite, and $|\mathcal{L}| \geq 2$, then the satisfiability problem for $L^{[*]}$ is EXPTIME complete.*

The lower bound follows from corollary 5.5.2. For the corresponding upper bound, we will give a reduction from this satisfiability problem to the satisfiability problem for a suitable subset of Deterministic Propositional Dynamic Logic (DPDL). This proves the theorem, since the satisfiability problem for DPDL is in EXPTIME (see Ben-Ari, Halpern and Pnueli [BHP82]). Our DPDL subset is the multi-modal propositional language with modalities $\langle l \rangle$ for all $l \in \mathcal{L}$, and $\langle (\bigcup_{l \in \mathcal{L}} l)^* \rangle$, which we will abbreviate as $\langle * \rangle$. We interpret wffs of this language on Kripke models $M = \langle W, \{R_l\}_{l \in \mathcal{L}}, \pi \rangle$, where R_l is a partial functional binary relation on W, in the usual way, the key clause being:

$$M, w \models \langle * \rangle \phi \text{ iff } \exists w'(w(\bigcup_{l \in \mathcal{L}} R_l)^* w' \;\&\; M, w' \models \phi).$$

Let ϕ be an $L^{[*]}$ formula. It is obvious that ϕ is a satisfiable $L^{[*]}$ formula iff ϕ is a satisfiable DPDL formula. □

Theorem 5.5.7 *If $\mathcal{L}$ is finite, and $|\mathcal{L}| \geq 2$, then the satisfiability problem for $L^{N[*]}$ is EXPTIME-complete.*

The lower bound follows from corollary 5.5.2. For the corresponding upper bound, we will give a reduction from the satisfiability problem for $L^{N[*]}$ to the corresponding satisfiability problem for $L^{[*]}$. The theorem then follows from theorem 5.5.6. Suppose ϕ is an $L^{N[*]}$ formula, and $m_1, \ldots, m_k$ are all the nominals occurring in ϕ. We can view nominals as ordinary propositional variables, with the extra requirement that each nominal is satisfied exactly once. We can't quite force that, but it turns out that forcing the following requirements for every nominal m that occurs in ϕ are enough to obtain the required reduction.

1. All nodes where m holds are equivalent with respect to $Cl(\phi)$
2. If m is true, and $\neg[*]\psi, \psi$ hold at w for some $[*]\psi \in Cl(\phi)$, then there exists a node w' reachable from w by a non-m path such that $\neg\psi$ holds at w'

To force the second requirement, we introduce new propositional variables $m_{\langle * \rangle \neg\psi}$, for each $[*]\psi \in Cl(\phi)$, and each occurring nominal m. $m_{\langle * \rangle \neg\psi}$ will be true if $\neg\psi$ has to be fulfilled by a world reachable by a non-m path. Now define the reduction f:

$$\begin{aligned} f(\phi) = \phi \quad \wedge \quad & \textstyle\bigwedge_{\psi \in Cl(\phi)}([*](m \to \psi) \vee [*](m \to \neg\psi)) \\ \wedge \quad & \textstyle\bigwedge_{[*]\psi \in Cl(\phi)}([*](m \wedge \neg[*]\psi \wedge \psi \to \bigvee_{l \in \mathcal{L}} \langle l \rangle m_{\langle * \rangle \neg\psi}) \wedge \\ & \qquad [*](m_{\langle * \rangle \neg\psi} \wedge \psi \to \textstyle\bigvee_{l \in \mathcal{L}} \langle l \rangle m_{\langle * \rangle \neg\psi}) \wedge \\ & \qquad [*](m \to \neg m_{\langle * \rangle \neg\psi}) \wedge \\ & \qquad [*](m_{\langle * \rangle \neg\psi} \to \langle * \rangle \neg\psi)) \end{aligned}$$

It is obvious that if ϕ is satisfiable in a model where every nominal m occurs exactly once, then $f(\phi)$ is satisfiable.

For the converse, suppose $f(\phi)$ is a satisfiable $L^{[*]}$ formula. Let $M = \langle W, \{R_l\}_{l \in \mathcal{L}}, \pi \rangle$ be a model such that $M, w \models f(\phi)$. Define relation $\sim$ such that: $w \sim w' \Leftrightarrow (w = w')$ or $M, w \models m$ and $M, w' \models m$ for some nominal m occurring in ϕ. It is easy to see that $\sim$ is an equivalence relation, and filtrating over $\sim$ yields a satisfying model for ϕ. □

As in the case of languages with □, we can reduce the number of propositional variables. Define $g_k([*]\psi) = [*](\sigma_{0,k} \rightarrow g(\psi))$ in the construction of theorem 5.4.6, and define

$$f(\phi(p_0, \ldots, p_k)) = \sigma_{0,k} \wedge g_k(\phi) \wedge [*](\sigma_{0,k} \rightarrow ([a]^{2k+3}\sigma_{0,k} \wedge \bigwedge_{l \neq a} [l]\sigma_{0,k} \wedge \bigwedge_{i=1}^{2k+2} \langle a \rangle^i (\neg\sigma_{0,k} \wedge \bigwedge_{l \neq a} [l]\bot)))$$

to get the analogue of theorem 5.4.6 for languages with $[*]$. The extra conjunct in f forces more similarity between the original model and the encoded model: $[*]$ can force more structure than □. In a similar way, we can get the analogue of theorem 5.4.7. Details are left to the reader.

We can summarize the complexity results of this section as follows:

Corollary 5.5.8 *If $|\mathcal{L}| \geq 2$, and $|\mathcal{A}| \geq 0$ the satisfiability problems for $L^{[*]}$ and $L^{N[*]}$ are EXPTIME-complete, and the satisfiability problem for $L^{KR[*]}$ is Π^0_1-complete for $\mathcal{L}$ infinite, and Σ^1_1-complete for $\mathcal{L}$ finite.* □

Clearly the results of this section are very bad; does this mean such infinitary extensions should be abandoned? We believe not: an interesting case for their linguistic interest is made by Keller [Kel91], who works with a language even *stronger* than $L^{KR[*]}$, namely PDL augmented with the Kasper Rounds path equality. Among other things Keller shows how this language can give a neat account of the LFG idea of functional uncertainty. Thus the idea seems linguistically interesting: the pressing task becomes the search for well behaved fragments.

Finally it should be remarked that Gazdar *et al.* emphasize a different application for $L^{[*]}$. Instead of viewing it as a grammar specification formalism, they use it to define syntactic categories; indeed the greater part of their paper is devoted to showing how a wide variety of treatments of syntactic category can be modeled and compared using $L^{[*]}$. An interesting corollary of this is that they are not particularly interested in the satisfaction problem: the problem of most concern to them is how expensive it is to check a category structure against some fixed category description ϕ. Clearly this is a very much simpler problem; in fact they show that it is solvable in linear time if ϕ is a wff of $L^{[*]}$. Their result extends to wffs of $L^{N[*]}$ and $L^{KR[*]}$.

5.6 Concluding Remarks

In this chapter we have investigated the satisfiability problem for a variety of modal languages of AVSs. The following table summarizes the results for the case of most interest in computational linguistics: both $\mathcal{L}$ and $\mathcal{A}$ finite ($|\mathcal{L}| \geq 2, |\mathcal{A}| \geq 0$).

	L	L^N	L^{KR}
	NP-complete	NP-complete	NP-complete
□	EXPTIME-complete	EXPTIME-complete	Π^0_1-complete
[*]	EXPTIME-complete	EXPTIME-complete	Σ^1_1-complete

As a final remark, let's see what happens if $|\mathcal{L}| = 1$. Intuitively, this should make things easier, and indeed it does. Consider for instance the languages with only $[*]$ and $\langle a \rangle$ as modalities. It is easy to see that a formula in these languages is satisfiable iff it is satisfiable on a (possibly infinite) model of the form $w_0 R_a w_1 R_a w_2 R_a \cdots$ or on a model of the form $w_0 R_a w_1 R_a \cdots R_a w_k R_a w_{k+1} R_a \cdots R_a w_m R_a w_k$. In this situation path equations or nominals don't make the situation more complex that $L^{[*]}$.

In fact $L^{[*]}$ is very like propositional linear temporal logic with operators X (next time) and G (always in the future). Formulas of this language are interpreted on $\mathbf{N}$, the natural numbers in their usual order, as follows: $X\phi$ holds at i if ϕ holds at $i+1$, and $G\phi$ holds at i iff for all $i' \geq i$, ϕ holds at i'. Using the fact that satisfiability for this language is PSPACE-complete [SC85], it is easy to prove that the satisfiability problems for the languages with only $\langle a \rangle$ and $[*]$ as modalities are PSPACE-complete as well. Using similar methods, we get the same results for the corresponding languages with □. We leave the details to the reader. Combining these remarks with theorem 5.4.6, and theorem 5.3.6, we can summarize the results for $|\mathcal{L}| = 1$ as follows:

Theorem 5.6.1 *If $|\mathcal{L}| = 1$, and $|\mathcal{A}| \geq 1$, the satisfiability problems for L, L^N, and L^{KR} are NP-complete, and the satisfiability problems for $L^{\Box}$, $L^{N\Box}$, $L^{KR\Box}$, $L^{[*]}$, $L^{N[*]}$, and $L^{KR[*]}$ are PSPACE-complete.* □

There remains much to do. In this chapter we have confined ourselves to languages with full Boolean expressivity, hence the results of this chapter are essentially limitative. An important problem to turn to next is the exploration of weaker fragments. Obvious choices would be fragments with only conjunction as a Boolean operator, fragments with only conjunction and disjunction, or fragments with only conjunction and negation on atoms. Results for such fragments exist in the literature, but a more detailed examination seems both possible and desirable. Further, it would be interesting to look for tractable fragments involving □ or [*]. A good way of approaching this topic would be to add strict implication $\Rightarrow$ as a primitive symbol to various fragments of L, L^N or L^{KR} (as we saw earlier, it the implicit combination of □ and $\rightarrow$ provided by $\Rightarrow$ that is the most important use of the universal modality) and then to look for restricted forms of strict implication that are useful but tractable.

It is our belief, however, that modal logic has more to offer computational linguistics than an analysis of unification formalisms. We've already seen a hint of this in Evan's use of □ to look at feature specification defaults, and in the the use of $L^{[*]}$ to specify grammatical categories. Moreover modalities figure in recent work in categorial grammar; see [Roo] for example. However there seem to be further possibilities. A particularly interesting one concerns the organization of computational lexicons. An important task in this application is the developed of formalisms for representing and manipulating lexical entries. DATR [EG89] is such a formalism, and an examination of its syntax and semantics suggests that it is open to modal analysis. What sort of benefits might result from such an analysis? Complexity results and inference systems are obvious answers, but there is another possibility that might be more important: modal logic might provide "logical maps" of possible extensions.

This point seems to be of wider relevance. In recent years modal logicians have explored a wide variety of enriched systems, some of which clearly bear on issues of knowledge representation. As has already been mentioned, Schild [Sch90] has made use of correspondences between core terminological logic and modal logic to obtain a number of complexity results for terminological reasoning. However more correspondences are involved. For example, terminological reasoning may also involve the "counting quantifiers"; that is, we may want to perform numerical comparisons. The modal logic of such counting quantifiers (and a great deal more besides) has been investigated by van der Hoek and de Rijke [HR]. Their work covers such topics as completeness, normal forms and computational complexity and is of obvious relevance to the knowledge representation community.

Finally, there may be deeper mathematical reasons for taking the modal connection seriously. Modal logic comes equipped not only with a Kripke semantics, but with an algebraic semantics, and *duality theory*, the study of the connections between the algebraic semantics and the Kripke semantics, is a highly developed branch of model logic; see [Gol89] for a detailed recent account. While some use of the algebraic semantics has been made in connection with Attribute Value structures (Reape [Rea91] for example, uses it to make connections with Smolka's work, and Schild [Sch90] utilizes an algebraic approach to simplify his presentation) in general it seems that a tool of potential value has been neglected.

Chapter 6

Nexttime is Not Necessary

6.1 Introduction

Recent work has shown that modal logics are extremely useful in formalizing the design and analysis of distributed protocols. (see [Hal87] for a survey). In [HV89], Halpern and Vardi categorize these logics in terms of two parameters: the language used and the class of distributed systems considered. The languages they consider depend on the number of processors, the absence or presence of an operator for common knowledge and the use of linear versus branching time. As in [HV89], we denote these languages by $CKL_{(m)}$, $KL_{(m)}$, $CKB_{(m)}$ and $KB_{(m)}$, where m is the number of processors, C denotes the presence of an operator for common knowledge, and L and B stand for linear and branching time. All of these languages include temporal operators for nexttime, until and sometimes.

We will now briefly describe the classes of systems considered in [HV89]. We view a distributed system as a set of possible runs of the system. We assume that runs proceed in discrete steps, and if r is a run then (r, i) (for $i \in \mathbf{N}$) describes the state of the system at the i-th step of run r. We say that a processor knows a fact ϕ at a given point, if ϕ is true at all points (r', i') that the processor considers possible at that point.

A processor does not forget if the set of runs it considers possible stays the same or decreases over time. The dual notion is no learning: we say that a processor does not learn if the set of runs it considers possible stays the same or increases over time. A system is synchronous if all processors have access to a global clock. Finally, a system has a unique initial state if no processor can distinguish $(r, 0)$ from $(r', 0)$ for all runs r and r'.

We use $\mathcal{C}$ to represent the class of all models and use subscripts nf, nl, $sync$ and uis to indicate classes of models where, respectively, all processors do not forget, all processors do not learn, where time is synchronous, and where there exists a unique initial state.

In [HV89], Halpern and Vardi completely characterize the computational complexity for all combinations of their languages and classes of models, including some results from Ladner and Reif [LR86]. In the cases of most interest to distributed systems, the satisfiability problems for these logics are undecidable. The following theorem states the complexity for all undecidable combinations. undecidable.

Theorem 6.1.1 (HV89)

1. *The satisfiability problems for* $CKL_{(\geq 2)}$ *and* $CKB_{(\geq 2)}$ *with respect to* $\mathcal{C}_{(nf)}$, $\mathcal{C}_{(nf,uis)}$, $\mathcal{C}_{(nf,sync)}$, $\mathcal{C}_{(nf,nl)}$, $\mathcal{C}_{(nf,sync,uis)}$, $\mathcal{C}_{(nf,nl,sync)}$, $\mathcal{C}_{(nl,sync)}$ *and* $\mathcal{C}_{(nl)}$ *are* Σ_1^1*-complete.*

2. *The satisfiability problems for $CKL_{(\geq 2)}$, $KL_{(\geq 2)}$, $CKB_{(\geq 2)}$ and $KB_{(\geq 2)}$ with respect to $\mathcal{C}_{(nf,nl,uis)}$ are Σ_1^1-complete.*

3. *The satisfiability problems for $CKL_{(\geq 2)}$, $KL_{(\geq 2)}$, $CKB_{(\geq 2)}$ and $KB_{(\geq 2)}$ with respect to $\mathcal{C}_{(nl,uis)}$ are RE-complete.*

Since the satisfiability problem for linear temporal logic with the three operators mentioned earlier is PSPACE-complete, while the satisfiability problem for linear temporal logic with just $\Diamond$ (sometimes) as temporal operator is only NP-complete [SC85], it is interesting to examine the impact on the complexity if we restrict the temporal operators in our languages to $\Diamond$ for linear time (resp. $\forall\Diamond$ and $\exists\Diamond$ for branching time). Let $C\overline{KL}_{(m)}$, $\overline{KL}_{(m)}$, $C\overline{KB}_{(m)}$ and $\overline{KB}_{(m)}$ denote the languages where $\Diamond$ (resp. $\forall\Diamond$ and $\exists\Diamond$) are the only temporal operators. Although the proofs in [LR86] and [HV89] rely heavily on the use of either the nexttime or the until operator, it turns out that, by using new techniques, we can prove the same complexity results if we restrict the temporal operators to $\Diamond$ (resp. $\forall\Diamond$ and $\exists\Diamond$). Using approximately the same techniques, we can prove that the well-known Σ_1^1-hardness result for the satisfiability problem for two-dimensional temporal logic [Har83] goes through if we restrict the temporal operators to the sometimes operators in both directions $\Diamond_u$ and $\Diamond_r$.

The rest of the chapter is organized as follows. In the next section we describe the formal model, following the notation from [HV89]; in section 6.3 we describe the specific problems encountered if we have only $\Diamond$ as a temporal operator; in section 6.4 we prove the analog of part 1 of theorem 6.1.1 for the linear time language $C\overline{KL}_{(\geq 2)}$, by forcing models to be gridlike; in section 6.5 we prove the analogs of 2 and 3 for the linear time cases and a Σ_1^1 lower bound for two-dimensional linear logic, by appropriately modifying the proof of Ladner and Reif [LR86]. Finally, in section 6.6, we prove that for all classes of models considered, the satisfiability problems for the branching time languages $C\overline{KB}_{(m)}$ and $\overline{KB}_{(m)}$ are at least as hard as the corresponding satisfiability problems for $C\overline{KL}_{(m)}$ and $\overline{KL}_{(m)}$.

6.2 Syntax and Semantics

We start by giving the syntax of languages $CKL_{(m)}$ and $CKB_{(m)}$. We assume we have a set of propositional variables $\mathcal{P}$ and define the set of $CKL_{(m)}$ and $CKB_{(m)}$ formulas as follows:

- if $p \in \mathcal{P}$ then p is a $CKL_{(m)}$ ($CKB_{(m)}$) formula.

- if ϕ, ψ are $CKL_{(m)}$ ($CKB_{(m)}$) formulas, then so are $\neg\phi$ and $\phi \wedge \psi$.

- if ϕ is a $CKL_{(m)}$ ($CKB_{(m)}$) formulas, then so are $K_k\phi$ (k knows ϕ), $E\phi$ (everyone knows ϕ) and $C\phi$ (ϕ is common knowledge).

- if ϕ, ψ are $CKL_{(m)}$ formulas, then so are $\odot\phi$ (nexttime ϕ), $\Diamond\phi$ (sometimes ϕ) and $\phi U \psi$ (ϕ until ψ).

- if ϕ, ψ are $CKB_{(m)}$ formulas, then so are $\forall\odot\phi$, $\exists\odot\phi$, $\forall\Diamond\phi$, $\exists\Diamond\phi$, $\forall\phi U\psi$, $\exists\phi U\psi$.

We define $\top$, $\vee$ and $\rightarrow$ in the usual way from $\neg$ and $\wedge$. In addition, we define for linear time $\Box\phi$ (always ϕ) as an abbreviation of $\neg\Diamond\neg\phi$, and for branching time we view $\forall\Box\phi$ (resp. $\exists\Box\phi$) as an abbreviation of $\neg\exists\Diamond\neg\phi$ (resp. $\neg\forall\Diamond\neg\phi$).

We define the sublanguages $KL_{(m)}$ (resp. $KB_{(m)}$) as the set of $CKL_{(m)}$ (resp. $CKB_{(m)}$) formulas without the C operator. By restricting the temporal operators in each language to $\Diamond$ (resp. $\forall\Diamond$ and $\exists\Diamond$), we get the corresponding languages $C\overline{KL}_{(m)}$, $\overline{KL}_{(m)}$, $C\overline{KB}_{(m)}$ and $\overline{KB}_{(m)}$.

We will now give the semantics for $CKL_{(m)}$. A linear time model M for m processors is a tuple $(R, \sim_1, \ldots \sim_m, \pi)$, where R is a set of runs, $\pi : R \times \mathbf{N} \rightarrow Pow(\mathcal{P})$ assigns to each point the set of propositional variables that are true at that point, and $\sim_k$ is an equivalence relation on $R \times \mathbf{N}$. We define $M, (r, i) \models \phi$ (ϕ is satisfied by point (r, i) of M) with induction on ϕ:

- $M, (r, i) \models p \Longleftrightarrow p \in \pi(r, i)$
- $M, (r, i) \models \neg\phi \Longleftrightarrow M, (r, i) \not\models \phi$
- $M, (r, i) \models \phi \wedge \psi \Longleftrightarrow M, (r, i) \models \phi$ and $M, (r, i) \models \psi$
- $M, (r, i) \models K_k\phi \Longleftrightarrow \forall(r', i') \sim_k (r, i) : M, (r', i') \models \phi$
- $M, (r, i) \models E\phi \Longleftrightarrow \forall k \leq m : M, (r, i) \models K_k\phi$
- $M, (r, i) \models C\phi \Longleftrightarrow \forall n : M, (r, i) \models E^n\phi$
- $M, (r, i) \models \odot\phi \Longleftrightarrow M, (r, i + 1) \models \phi$
- $M, (r, i) \models \Diamond\phi \Longleftrightarrow \exists j \geq i : M, (r, j) \models \phi$
- $M, (r, i) \models \phi U \psi \Longleftrightarrow \exists i' \geq i : M, (r, i') \models \psi$ and $\forall i''(i \leq i'' < i' \Rightarrow M, (r, i'') \models \phi)$

Given a model M for m processors, we define $\sim_c$ as the reflexive transitive closure of $\bigcup_{k=1}^{m} \sim_k$. Then $M, (r, i) \models C\phi$ if and only if $\forall(r', i') \sim_c (r, i) : M, (r', i') \models \phi$.

We will now give the semantics for $CKB_{(m)}$. A branching time model M for m processors is a tuple $(F, \sim_1, \ldots \sim_m, \pi)$ where F is a forest, π assigns to each point of F the set of propositional variables that are true at that point, and $\sim_k$ is an equivalence relation on points of F. We assume that each node in F has some successor. We will view F as a tuple $< R_F, =_F >$ where R_F is the set of the infinite branches of F that start at the root of some tree in F. (r, i) denotes the i-th node of r and $(r, i) =_F (r', i)$ if and only if (r, i) and (r', i) denote the same node of F. We will define $M, (r, i) \models \phi$ with induction on ϕ. We only give the clauses for $\exists\Diamond$ and $\forall\Diamond$, the other temporal operators are defined analogously.

- $M, (r, i) \models \forall\Diamond\phi \Longleftrightarrow \forall(r', i) =_F (r, i)\ \exists j \geq i : M, (r', j) \models \phi$
- $M, (r, i) \models \exists\Diamond\phi \Longleftrightarrow \exists(r', i) =_F (r, i)\ \exists j \geq i : M, (r', j) \models \phi$

As usual, we say that a formula ϕ is valid with respect to a class of models $\mathcal{D}$, if and only if for all models $M \in \mathcal{D}$ and for all points (r, i) of M : $M, (r, i) \models \phi$. A formula ϕ is satisfiable with respect to $\mathcal{D}$ if and only if there is some model $M \in \mathcal{D}$ and some point (r, i) of M such that $M, (r, i) \models \phi$.

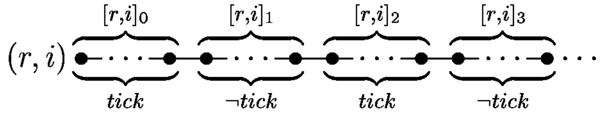
We will now define the classes of models of [HV89]:

- Processor k *does not forget* in M if for all $r, r' \in R$ and $i, i' \in \mathbf{N}$: if $(r,i) \sim_k (r',i')$ then $\forall j \leq i \, \exists j' \leq i'$ such that $(r,j) \sim_k (r',j')$.
- Processor k *does not learn* in M if for all $r, r' \in R$ and $i, i' \in \mathbf{N}$: if $(r,i) \sim_k (r',i')$ then $\forall j \geq i \, \exists j' \geq i'$ such that $(r,j) \sim_k (r',j')$.
- Time is *synchronous* in M if for all $r, r' \in R$ and $i, i' \in \mathbf{N}$ and all processors k: $(r,i) \sim_k (r',i')$ implies that $i = i'$.
- M has a *unique initial state* if for all $r, r' \in R$ and all processors k: $(r,0) \sim_k (r',0)$.

We use $\mathcal{C}$ to represent the class of all models and use subscripts $nf, nl, sync, uis$ to indicate classes of models where, respectively, all processors do not forget, all processors do not learn, where time is synchronous, and where there is a unique initial state.

6.3 From Points to Intervals

In all our proofs, it is essential that the constructed formulas force runs to encode certain strings. The obvious way to encode some string on run r starting at i is by letting $(r, i+j)$ encode the j-th symbol of the string. However, if we restrict the temporal operators to just $\diamond$, we can't distinguish adjacent points satisfying the same set of formulas. To solve this problem, we introduce a propositional variable *tick*, alternating on runs. *tick* partitions each run into an infinite number of intervals. For all n, let $[r,i]_n$ be the set of points in the n-th interval of r starting at i. (Note that we start counting the intervals from 0.)



We will encode strings at consecutive intervals of a run. We say that (r,i) encodes some string if and only if each point in the j-th interval of (r,i) ($[r,i]_j$) encodes the j-th element of the string. It is possible to mark a fixed number of consecutive intervals on a run by propositional constants. Let *1-int*(p, up_p) be the conjunction of the following five formulas:

$$\diamond p \wedge \Box(p \wedge tick \rightarrow \Box(\neg tick \rightarrow \Box\neg p)) \wedge \Box(p \wedge \neg tick \rightarrow \Box(tick \rightarrow \Box\neg p))$$

(p holds somewhere at some interval, and p holds nowhere outside that interval.)

$$\Box(tick \wedge \diamond(p \wedge tick) \rightarrow \diamond(\neg tick \wedge \diamond p) \vee p) \wedge \Box(\neg tick \wedge \diamond(p \wedge \neg tick) \rightarrow \diamond(tick \wedge \diamond p) \vee p)$$

(p holds exactly at some prefix of an interval.)

$$\Box(p \wedge tick \rightarrow \diamond(up_p \wedge \neg tick)) \wedge \Box(p \wedge \neg tick \rightarrow \diamond(up_p \wedge tick))$$

(up_p holds somewhere after the p interval.)

$$\Box(up_p \wedge \neg tick \rightarrow \Box(tick \rightarrow \Box\neg up_p)) \wedge \Box(up_p \wedge tick \rightarrow \Box(\neg tick \rightarrow \Box\neg up_p))$$

(if up_p holds somewhere at some interval, then up_p holds nowhere outside that interval.)

$$\Box(p \rightarrow \Box((\neg up_p \wedge \diamond up_p) \rightarrow p))$$

Thus, $(r, i) \models \textit{1-int}(p, up_p)$ if and only if there is exactly one interval at which p holds, p holds nowhere else and up_p holds exactly at some prefix of the next interval.
Therefore if $(r, i) \models \textit{1-int}(p_0, p_1) \wedge \textit{1-int}(p_1, p_2) \wedge \ldots \wedge \textit{1-int}(p_{n-1}, up_p)$ then there are n consecutive intervals on r starting at i such that p_j holds exactly at the j-th interval.

Now we can define $\models$ and $\sim_k$ on intervals:

- $[r, i]_n \models \phi$ iff $\forall (r, j) \in [r, i]_n : (r, j) \models \phi$
(Note that $[r, i]_n \not\models \phi$ does *not* imply that $[r, i]_n \models \neg\phi$)
- $[r, i]_n \sim_k [r', i']_{n'}$ iff
  - $\forall (r, j) \in [r, i]_n \, \exists (r', j') \in [r', i']_{n'} : (r, j) \sim_k (r', j')$, and
  - $\forall (r', j') \in [r', i']_{n'} \, \exists (r, j) \in [r, i]_n : (r, j) \sim_k (r', j')$.

Though the specific classes considered do imply specific behavior for the epistemic relations with respect to points, not much of this behavior carries over to intervals. For example, it is perfectly possible that a model in $\mathcal{C}_{(sync)}$ is not synchronous with respect to intervals. However, the following formula ψ_k will force that some of the structural properties of points hold for intervals as well.

$$K_k \Box((tick \rightarrow K_k tick) \wedge (\neg tick \rightarrow K_k \neg tick))$$

Lemma 6.3.1 *If $M \in \mathcal{C}_{(nf)}$, $M, (r, i) \models \psi_k$, and $(r, j) \sim_k (r', j')$ for some $(r, j) \in [r, i]_n$ with $n > 0$, then*

- *there exists some i' such that $(r, i) \sim_k (r', i')$, and*
- *for all i' such that $(r, i) \sim_k (r', i')$ it holds that $(r', j') \in [r', i']_n$ and $\forall n' < n$: $[r, i]_{n'} \sim_k [r', i']_{n'}$.*

Lemma 6.3.2 *If $M \in \mathcal{C}_{(nl,sync)}$, $M, (r, i) \models \psi_k$ and $(r, i) \sim_k (r', i)$, then $[r, i]_n \sim_k [r', i]_n$ for all n.*

The proofs of these lemmas are similar to the proofs that force not necessarily synchronous models to be essentially synchronous from [HV89].

6.4 Forcing Models to be Gridlike

Theorem 6.4.1 *The satisfiability problems for $CK\overline{L}_{(\geq 2)}$ with respect to $\mathcal{C}_{(nf)}$, $\mathcal{C}_{(nf,uis)}$, $\mathcal{C}_{(nf,sync)}$, $\mathcal{C}_{(nf,nl)}$, $\mathcal{C}_{(nf,sync,uis)}$ and $\mathcal{C}_{(nf,nl,sync)}$ are Σ_1^1-complete.*

Since the Σ_1^1 upper bounds for these classes follow directly from [HV89], it is enough to prove the lower bounds for two processors. As in [HV89], all Σ_1^1 lower bounds for linear time classes will be proved by a reduction from the recurrence problem for nondeterministic Turing machines. We say that a Turing machine is *recurrent* if and only if it has an infinite computation that starts on the empty tape and reenters its start state infinitely often.

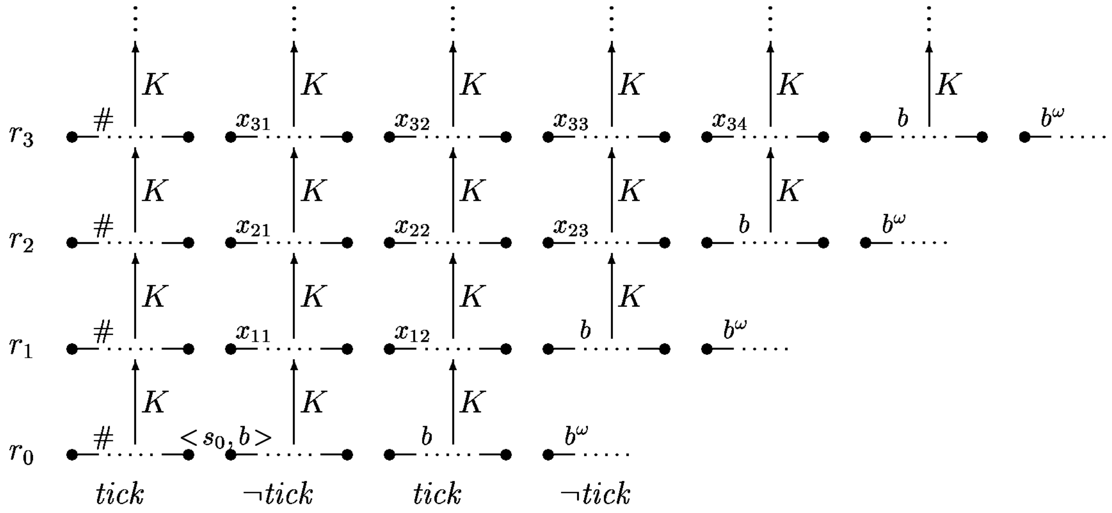
Theorem 6.4.2 ([HPS83]) *If $A_1, A_2, A_3, \ldots$ is a recursive enumeration of the 1-tape, right-infinite nondeterministic Turing machines, then $\{n \mid A_n \text{ is recurrent}\}$ is Σ_1^1-complete.*

Given an arbitrary 1-tape, right infinite NTM A, we will construct a formula ϕ_A such that:

- If ϕ_A is satisfiable with respect to $\mathcal{C}_{(nf)}$ then A is recurrent, and
- if A is recurrent then ϕ_A is satisfiable with respect to $\mathcal{C}_{(nf,nl,sync)}$ and $\mathcal{C}_{(nf,sync,uis)}$.

This implies the Σ_1^1 lower bound for the satisfiability problems for all six classes.

For the remainder of the proof, we fix a 1-tape right-infinite NTM A. Suppose A has state space S, start state $s_0 \in S$; tape alphabet Γ; $b \in \Gamma$: the blank; and transition function δ. We use a special symbol # to mark the left side of the tape. Let CD (the set of cell descriptors) be the set $\Gamma \cup \{\#\} \cup (S \times \Gamma)$. We view the IDs of A as infinite strings over CD, where $\langle s, a\rangle$ denotes a cell with contents a, which is currently read by the head while A is in state s. A starts on the empty tape in state s_0, so the start ID of A (id_0) is equal to $\#\langle s_0, b\rangle b^\omega$. Now suppose $id_0 \vdash id_1 \vdash id_2 \vdash \cdots$ is an infinite computation of A. Then for all n: $id_n = \#x_{n,1}x_{n,2}\ldots x_{n,n}x_{n,n+1}b^\omega$ ($x_{n,i} \in \mathrm{CD}$). The idea is to encode this computation in a model, by letting the runs represent the IDs (using the interval techniques of the previous section) and using the epistemic relations to simulate the transition function.



Since the encoding of IDs will be done at the intervals of runs, we start by partitioning each run into an infinite number of intervals, using the propositional variable *tick*. The following formula ϕ_1 will take care of this:

$$C\Box((tick \to \Diamond\neg tick) \wedge (\neg tick \to \Diamond tick))$$

The epistemic relation $\xrightarrow{K}$ is used to determine the contents of a cell at the next step of a computation. Therefore, $\xrightarrow{K}$ should not be reflexive, transitive or symmetric. As in [HV89], we use both epistemic relations $\sim_1$ and $\sim_2$ and introduce a propositional variable p_Δ to avoid reflexivity. We define the relation $\xrightarrow{K}$ as follows:

$$\begin{aligned}(r,i) \xrightarrow{K} (r',i') \text{ iff } \quad \exists r'', i'' : \quad & (r,i) \sim_1 (r'',i'') \sim_2 (r',i') \text{ and} \\ & (r'',i'') \models \neg p_\Delta \text{ and } (r',i') \models p_\Delta\end{aligned}$$

The associated modal operator K is defined by: $K\psi := K_1(\neg p_\Delta \to K_2(p_\Delta \to \psi))$.

Let ϕ_2 be the following formula:

$$C((p_\Delta \to \Box p_\Delta) \wedge (\neg p_\Delta \to \Box\neg p_\Delta)) \wedge C\Box C(\neg K_1 p_\Delta \wedge \neg K_2 \neg p_\Delta)$$

If $(r_0, i_0) \models \phi_2$ and $(r, i) \sim_c (r_0, i_0)$ then the value of p_Δ on r from i upwards is constant, and by the second conjunct we can take an infinite number of $\xrightarrow{K}$ steps from each point on r after i. This will ensure that we encode an infinite computation.

Since we are interested in the behavior of $\xrightarrow{K}$ with respect to intervals, we define:

$$\begin{array}{lll} [r,i]_n \xrightarrow{K} [r',i']_{n'} \text{ iff} & \forall (r,j) \in [r,i]_n \, \exists (r',j') \in [r',i']_{n'}: & (r,j) \xrightarrow{K} (r',j') \wedge \\ & \forall (r',j') \in [r',i']_{n'} \, \exists (r,j) \in [r,i]_n: & (r,j) \xrightarrow{K} (r',j') \end{array}$$

For the IDs to match up right, we need synchrony and no forgetting of $\xrightarrow{K}$ with respect to intervals. Let ϕ_3 be the following formula:

$$C\Box((\mathit{tick} \to C\mathit{tick}) \wedge (\neg \mathit{tick} \to C\neg \mathit{tick}))$$

By lemma 6.3.1 and the fact that p_Δ is constant on runs (as forced by ϕ_2), we immediately obtain the following:

Lemma 6.4.3 *If* $(r_0, i_0) \models \phi_3$, $(r,i) \sim_c (r_0, i_0)$, $(r,j) \xrightarrow{K} (r',j')$, *and* $(r,j) \in [r,i]_n (n > 0)$, *then*

- *there exists some* i' *such that* $(r,i) \xrightarrow{K} (r',i')$
- *for all* i', *if* $(r,i) \xrightarrow{K} (r',i')$ *then* $(r',j') \in [r',i']_n$ *and* $\forall n' < n : [r,i]_{n'} \xrightarrow{K} [r',i']_{n'}$.

Now we turn to the encoding of IDs. We will encode IDs on runs where p_Δ holds. To encode the cell descriptors, we introduce for each $x \in \mathrm{CD}$ a propositional variable p_x. Let ϕ_4 be the formula

$$C\Box(p_\Delta \to \bigvee_{x \in \mathrm{CD}} (p_x \wedge \neg \bigvee_{y \in \mathrm{CD},\ y \neq x} p_y))$$

If $(r_0, i_0) \models \phi_1 \wedge \cdots \wedge \phi_4$ and $(r,i) \sim_c (r_0, i_0)$ and p_Δ holds at (r,i), then each point on r after i encodes exactly one cell descriptor.

We say that the n-th interval of $(r,i)([r,i]_n)$ encodes $x \in \mathrm{CD}$, if each point in $[r,i]_n$ encodes x. To encode the start ID (id_0) at the first run, we introduce the following formula ϕ_{start} (up_s is a dummy variable):

$$p_\Delta \wedge \textit{1-int}(p_\#, p_{<s_0,b>}) \wedge \textit{1-int}(p_{<s_0,b>}, up_s) \wedge \Box(p_\# \vee p_{<s_0,b>} \vee p_b)$$

To simulate the transition function we just have to make sure that $\xrightarrow{K}$ points to the corresponding cell of a next ID. Suppose $id \vdash id'$. The only cells that can be affected by the transition are the cell holding the state and its neighbors. On each run we mark the three consecutive intervals corresponding to these cells with propositional variables *left*, *state* and *right*. Let $\phi_{5,1}$ be the conjunction of the following formulas (up_{state} is a dummy variable):

$$C(p_\Delta \to \textit{1-int}(\mathit{left}, \mathit{state}) \wedge \textit{1-int}(\mathit{state}, \mathit{right}) \wedge \textit{1-int}(\mathit{right}, up_{state}))$$

$$C\Box(p_\Delta \wedge \bigvee_{\langle s,a\rangle \in S\times\Gamma} p_{\langle s,a\rangle} \rightarrow \mathit{state})$$

For the transition on all non-marked cells, let $\phi_{5,2}$ be the conjunction of the following formulas:

$$\bigwedge_{x\in\mathrm{CD}} C\Box(p_\Delta \wedge (\neg \mathit{left} \wedge \neg \mathit{state} \wedge \neg \mathit{right}) \wedge p_x \rightarrow K p_x)$$

$$C\Box(p_\Delta \wedge (\neg \mathit{left} \wedge \neg \mathit{state} \wedge \neg \mathit{right}) \wedge \Box p_b \rightarrow K\Box p_b)$$

Now the transition on the three marked intervals. Let $N(x,y,z)$ be the set of successor triples of $\langle x,y,z\rangle$ as given by the transition function δ. Let $\phi_{5,3}$ be the following formula:

$$C(p_\Delta \wedge \Diamond(\mathit{left} \wedge p_x) \wedge \Diamond(\mathit{state} \wedge p_y) \wedge \Diamond(\mathit{right} \wedge p_z) \longrightarrow \bigvee_{\langle x',y',z'\rangle \in N(x,y,z)} (\Box(\mathit{left} \rightarrow K p_{x'}) \wedge \Box(\mathit{state} \rightarrow K p_{y'}) \wedge \Box(\mathit{right} \rightarrow K p_{z'})))$$

Let ϕ_5 be the conjunction of $\phi_{5,1}, \phi_{5,2}$ and $\phi_{5,3}$. By lemma 6.4.3, if $(r,i) \models \phi_1 \wedge \cdots \wedge \phi_5$ and (r,i) encodes some ID $\mathit{id} = x_0 \ldots x_{n-1} b^\omega$ of A, and for some $j \in [r,i]_m (m > n+1)$, $(r,j) \xrightarrow{K} (r',j')$ then there exists some i' such that $(r,i) \xrightarrow{K} (r',i')$ and (r',i') encodes a successor ID $y_0 \ldots y_n b^\omega$ of id.

Since by ϕ_2, we can take an infinite number of $\xrightarrow{K}$ steps from any point in the model, we know that we encode an infinite computation of A.

The only thing left to do now is to force the encoded computation to be recurrent. That is, at each time in the computation, there must be some later time when the computation is in the start state. To be able to express this requirement in a formula, we must be able to discriminate at each time those IDs which occur at some later step in the computation. Therefore, we time stamp each run that encodes an ID with the time of the computation. Say (r,i) is at time t if and only if exactly the $(t+2)$-nd and $(t+3)$-rd interval of (r,i) are marked with *time₁* and *time₂*. The first run is at time 0; we will mark the second and third intervals on this run with *time₁* and *time₂*, and we will ensure that a run at time t has a successor at time $t+1$. Let ϕ_6 be the conjunction of the following two formulas (with $\mathit{up}_{\mathit{time}}$ a new dummy variable):

$$C(p_\Delta \rightarrow \mathit{1\text{-}int}(\mathit{time}_1, \mathit{time}_2) \wedge \mathit{1\text{-}int}(\mathit{time}_2, \mathit{up}_{\mathit{time}})) \;\wedge\; \mathit{1\text{-}int}(p_{\langle s_0,b\rangle}, \mathit{time}_1)$$

$$C\Box(p_\Delta \wedge \mathit{time}_2 \rightarrow K \mathit{time}_1)$$

By the first conjunct, each p_Δ run is time stamped, and (r_0, i_0) is at time 0. Next suppose that $(r,i) \sim_c (r_0,i_0)$ and (r,i) is at time t, and $(r,j) \xrightarrow{K} (r',j')$ for some $j \in [r,i]_m$ with $m > t+2$. By lemma 6.4.3 and the second conjunct, it follows that for each i' such that $(r,i) \xrightarrow{K} (r',i')$ it holds that (r',i') is at time $t+1$.

To check whether an infinite computation is recurrent or not, we need to discriminate between runs that encode IDs in the start state and those that are in a different state. To this end we introduce the following formula ϕ_7:

$$C(\Diamond \bigvee_{x\in\Gamma} p_{<s_0,x>} \rightarrow \Box \mathit{startstate}) \wedge C(\Box \bigwedge_{x\in\Gamma} \neg p_{<s_0,x>} \rightarrow \Box\neg \mathit{startstate})$$

If $(r_0, i_0) \models \phi_1 \wedge \ldots \wedge \phi_7$ and $(r, i) \sim_c (r_0, i_0)$ and (r, i) encodes an ID, then *startstate* is constant on the run, *startstate* is true if the state of the encoded ID is the start state, false otherwise.

Finally, we state the formula to force recurrence ϕ_{rec}

$$C\Box(\Diamond C(p_\Delta \wedge \mathit{time}_2 \rightarrow \mathit{startstate}))$$

Let ϕ_A be the conjunction of ϕ_1 through ϕ_7, ϕ_{start} and ϕ_{rec}. Suppose $M, (r_0, i_0) \models \phi_A$ for some $M \in \mathcal{C}_{(nf)}$. Then (r_0, i_0) encodes id_0. Suppose $(r, i) \sim_c (r_0, i_0)$ and (r, i) encodes some ID id at time t. By ϕ_{rec}, for each i' there must exist some $i'' \geq i'$ such that $(r, i'') \models C(p_\Delta \wedge \mathit{time}_2 \rightarrow \mathit{startstate})$. In particular, there must exist some $m > 0$ and some j such that $(r, j) \in [r, i]_{t+3+m}$ and $(r, j) \models C(p_\Delta \wedge \mathit{time}_2 \rightarrow \mathit{startstate})$. By ϕ_2, we can take m $\xrightarrow{K}$ steps from (r, j), say $(r, j)(\xrightarrow{K})^m(r', j')$. By ϕ_5, there must exist some i' such that $(r, i)(\xrightarrow{K})^m(r', i')$, (r', i') encodes some ID id' such that $id(\vdash)^m id'$ and $(r', j') \in [r', i']_{t+3+m}$. By ϕ_6, (r', i') is at time $t + m$, but then $(r', j') \models \mathit{time}_2 \wedge p_\Delta$ and therefore *startstate* is true at run r'. Thus, id' is in the start state. Since (r, i) was chosen arbitrarily, we have shown that M encodes a recurrent computation of A.

To conclude the proof of theorem 6.4.1, we still have to show that ϕ_A is satisfiable with respect to $\mathcal{C}_{(nf,nl,sync)}$ and $\mathcal{C}_{(nf,sync,uis)}$ if A is recurrent. Let $id_0 \vdash id_1 \vdash id_2 \vdash \cdots$ be an infinite computation of A that starts on the empty tape and reenters its start state infinitely often. We will construct models in $\mathcal{C}_{(nf,nl,sync)}$ and $\mathcal{C}_{(nf,sync,uis)}$ that satisfy ϕ_A.

- ϕ_A is satisfiable with respect to $\mathcal{C}_{(nf,nl,sync)}$.

  We construct model $M = (\{r_\ell \mid \ell \in \mathbf{N}\}, \sim_1, \sim_2, \pi) \in \mathcal{C}_{(nf,nl,sync)}$ such that $M, (r_0, 0) \models \phi_A$:

  - $\sim_1$ is the reflexive, symmetric and transitive closure of the set

    $$\{\langle (r_{2\ell}, i), (r_{2\ell+1}, i)\rangle \mid \ell, i \in \mathbf{N}\}$$

  - $\sim_2$ is the reflexive, symmetric and transitive closure of the set

    $$\{\langle (r_{2\ell+1}, i), (r_{2\ell+2}, i)\rangle \mid \ell, i \in \mathbf{N}\}.$$

  - $p_\Delta \in \pi(r_\ell, i)$ iff ℓ even; *tick* $\in \pi(r_\ell, i)$ iff i even.

  It is immediate that $M \in \mathcal{C}_{(nf,nl,sync)}$, $[r_\ell, i]_n = \{(r_\ell, i + n)\}$ and $(r_{2\ell}, i) \models K\psi$ iff $(r_{2(\ell+1)}, i) \models \psi$. It is now easy to define π such that $M, (r_{2\ell}, 0)$ encodes id_ℓ at time ℓ and $M, (r_0, 0)$ satisfies all conjuncts of ϕ_A with the possible exception of ϕ_{rec}. But if $(r_0, 0)$ satisfies all other conjuncts of ϕ_A, then ϕ_{rec} is satisfied as well. For let (r_ℓ, i) be a point of M. We need to show that $(r_\ell, i) \models \Diamond C(p_\Delta \wedge \mathit{time}_2 \rightarrow \mathit{startstate})$. Since the computation is recurrent, there must exist some $j > i$ such that id_j is in the start state. We claim that $(r_\ell, j + 3) \models C(p_\Delta \wedge \mathit{time}_2 \rightarrow \mathit{startstate})$. Suppose $(r_m, j + 3) \models p_\Delta \wedge \mathit{time}_2$. Then m is even and $(r_m, 0)$ encodes some ID at time j. This can only the case if $m = 2j$. And since id_j is in the start state, $(r_m, j + 3) \models \mathit{startstate}$.

- ϕ_A is satisfiable with respect to $\mathcal{C}_{(nf,sync,uis)}$.

  Let M be the $\mathcal{C}_{(nf,nl,sync)}$ model defined above. Transform this model into an $\mathcal{C}_{(nf,sync,uis)}$ model $M' = (R, \sim_1', \sim_2', \pi')$ by adding a unique initial state:

- $\sim'_k = \{\langle(r,i+1),(r',i+1)\rangle \mid (r,i) \sim_k (r',i)\} \cup \{\langle(r,0)(r',0)\rangle \mid r,r' \in R\}$,
- $\pi'(r_\ell, i+1) = \pi(r_\ell, i)$

It is immediate that $M' \in \mathcal{C}_{(nf,sync,uis)}$ and $M',(r_0,1) \models \phi_A$. □

Theorem 6.4.4 *The satisfiability problem for $CK\overline{L}_{(\geq 2)}$ with respect to $\mathcal{C}_{(nl,sync)}$ is Σ^1_1-complete.*

We will show that formula ϕ_A works for $\mathcal{C}_{(nl,sync)}$ as well, i.e. A is recurrent iff ϕ_A is satisfiable with respect to $\mathcal{C}_{(nl,sync)}$. By lemma 6.3.2, we immediately obtain the following analog of lemma 6.4.3.

Lemma 6.4.5 *If $M \in \mathcal{C}_{(nl,sync)}$, $(r,i) \models \phi_3$ and $(r,i) \xrightarrow{K} (r',i)$, then $\forall n : [r,i]_n \xrightarrow{K} [r',i]_n$.*

Using this lemma, we can use the same argument as in the proof of theorem 6.4.1 to show that if $M,(r_0,i) \models \phi_1 \wedge \ldots \wedge \phi_6$, $(r_0,i) \sim_c (r,i) \xrightarrow{K} (r',i)$ and (r,i) encodes some ID *id* at time t then (r',i) encodes a successor ID of *id* at time $t+1$.

Suppose $M,(r_0,i) \models \phi_A$ for some $M \in \mathcal{C}_{(nl,sync)}$. Then (r_0,i) encodes id_0. Suppose $(r,i) \sim_c (r_0,i)$ and (r,i) encodes some ID *id* at time t. As in the previous proof, there must exist some $m > 0$ and some j such that $(r,j) \in [r,i]_{t+3+m}$ and $(r,j) \models C(p_\Delta \wedge time_2 \rightarrow startstate)$. By ϕ_2, we can take $m \xrightarrow{K}$ steps from (r,i), say $(r,i)(\xrightarrow{K})^m(r',i)$. By ϕ_5 and ϕ_6, we know that (r',i) encodes some ID id' such that $id(\vdash)^m id'$ and that (r',i) is at time $t+m$. By $(nl, sync)$, $(r,j) \sim_c (r',j)$ and by lemma 6.4.5, $(r',j) \in [r',i]_{t+m+3}$. Therefore, *startstate* is true at run r'. Thus, id' is in the start state. Since (r,i) was chosen arbitrarily, we have shown that M encodes a recurrent computation of A. The converse follows from theorem 6.4.1, since if A is recurrent, then ϕ_A is satisfiable with respect to $\mathcal{C}_{(nf,nl,sync)}$ and thus certainly satisfiable with respect to $\mathcal{C}_{(nl,sync)}$. □

Theorem 6.4.6 *The satisfiability problem for $CK\overline{L}_{(\geq 2)}$ with respect to $\mathcal{C}_{(nl)}$ is Σ^1_1-complete.*

In our proof of the Σ^1_1 lower bound of Theorem 6.4.4, it was essential that formula ϕ_3 forced synchrony with respect to intervals. This is not the case for models in $\mathcal{C}_{(nl)}$, since lemmas 6.3.2 and 6.4.5 do not hold for non-synchronous models. However, in [HV89] it is shown that we can force synchrony on finite prefixes of runs. This will enable us to force synchrony with respect to intervals on finite prefixes of runs. It turns out that this suffices to prove a Σ^1_1 lower bound for satisfiability with respect to $\mathcal{C}_{(nl)}$ with minor changes to ϕ_A.

Following [HV89], we call a point (r,i) k-repeating if there exist infinitely many $j > i$ such that $(r,i) \sim_k (r,j)$.

Lemma 6.4.7 ([HV89]) *Let M be a model in $\mathcal{C}_{(nl)}$ and let $r,r' \in R$:*

- *If $(r,i) \sim_k (r,i')$, $i \leq i'$ and (r,i) is not k-repeating, then $(r,i) \sim_k (r,i'')$ for all i'' between i and i'.*
- *If (r,i) is not k-repeating and $j < i$ then (r,j) is not k-repeating.*
- *If $(r,i) \sim_k (r',i')$ and (r,i) is not k-repeating, then (r',i') is not k-repeating.*

Using this lemma, we can force synchrony with respect to intervals on finite prefixes of runs. Let ψ_k be the formula:

$$K_k\Box((tick \rightarrow \Diamond\neg tick) \wedge (\neg tick \rightarrow \Diamond tick) \wedge (tick \rightarrow K_k tick) \wedge (\neg tick \rightarrow K_k \neg tick))$$

Lemma 6.4.8 *Let M be a model in $\mathcal{C}_{(nl)}$ such that $M,(r,i) \models \psi_k$. Suppose $(r,i) \sim_k (r',i')$, $j \geq i$ and (r,j) is not k repeating.*

- *If $(r,j) \sim_k (r',j')$ and $j' \geq i'$ then $((r,j) \in [r,i]_n$ iff $(r',j') \in [r',i']_n)$.*
- *If $(r,j) \in [r,i]_n$ then $\forall m < n : [r,i]_m \sim_k [r',i']_m$.*

For part 1, suppose $(r,j) \in [r,i]_n$ and $(r',j') \in [r',i']_{n'}$ with $n' < n$. Take $i_0 < i_1 < \cdots < i_n$ such that $i_0 = i$, $i_n = j$ and for each $\ell \leq n$ $(r,i_\ell) \in [r,i]_\ell$. Because we assume no learning, there must exist $i'_0 < i'_1 < \cdots < i'_n$ such that $i'_0 = i'$ and for each $\ell \leq n$ $(r,i_\ell) \sim_k (r',i'_\ell)$. For each $\ell < n$, (r,i_ℓ) and $(r,i_{\ell+1})$ belong to consecutive intervals, so *tick* has a different truth value for (r,i_ℓ) and $(r,i_{\ell+1})$. By ψ_k, it follows that (r',i'_ℓ) and $(r',i'_{\ell+1})$ must belong to different intervals as well. Therefore, $(r',i'_n) \in [r',i'_0]_m = [r',i']_m$ for some $m \geq n$. Since $n' < n$ and $(r',j') \in [r',i']_{n'}$ it follows that $i'_n > j'$ and (r',i'_n) and (r',j') belong to different intervals. But since also $(r,j) \sim_k (r',j')$ and $(r,j) \sim_k (r',i'_n)$ and (r,j) is not k-repeating, it follows by lemma 6.4.7 that every point on r' between j' and i'_n must be k-equivalent to (r',j'), and therefore, by ψ_k the truth value of *tick* must be constant between (r',j') and (r',i'_n). But then (r',j') and (r',i'_n) belong to the same interval.

For part 2, suppose $(r,\ell) \in [r,i]_m$ for some $m < n$. By no learning, there exists some $\ell' \geq i'$ such that $(r,\ell) \sim_k (r',\ell')$. Since by lemma 6.4.7 (r,ℓ) is not k-repeating, it follows by part 1 that $(r',\ell') \in [r',i']_m$. □

We can force certain points to be not k-repeating. As in [HV89], define:

$$nonrep := q \wedge \Diamond\Box\neg q$$

If $M \in \mathcal{C}_{(nl)}$ and $M,(r,i) \models K_k nonrep$ then (r,i) is not k-repeating.

Where do we need synchrony? Reviewing the proof for $\mathcal{C}_{(nl,sync)}$, we need the following: if (r,i) encodes an ID *id* at time t, there exist some $m > 0$ and $(r,j) \in [r,i]_{t+m+3}$ such that $(r,j) \models C(p_\Delta \wedge time_2 \rightarrow startstate)$ and if $(r,i)(\xrightarrow{K})^m(r',i')$ then $(r,j)(\xrightarrow{K})^m(r',j')$ for some $(r',j') \in [r',i']_{t+3+m}$. Let ϕ^{nl}_{rec} be the formula:

$$C\Diamond(\Box\neg time_2 \wedge C(p_\Delta \wedge time_2 \rightarrow startstate) \wedge K_1 nonrep)$$

And let ϕ_{nl} be the formula:

$$C\Box(p_\Delta \wedge K_1 nonrep \rightarrow (K_1(\neg p_\Delta \rightarrow K_2 nonrep) \wedge KK_1 nonrep))$$

Let ϕ^{nl}_A be the conjunction of ϕ_1 through ϕ_7, ϕ_{start}, ϕ_{nl} and ϕ^{nl}_{rec}. We show that ϕ^{nl}_A is satisfiable with respect to $\mathcal{C}_{(nl)}$ iff A is recurrent.

We first prove that if ϕ^{nl}_A is satisfiable with respect to $\mathcal{C}_{(nl)}$, then A is recurrent. This follows from the following lemma:

Lemma 6.4.9 *Let M be a model in $\mathcal{C}_{(nl)}$ and suppose $M,(r_0,i_0) \models \phi^{nl}_A$ and $(r_0,i_0) \sim_c (r,i)$.*

- *If* $(r,i) \xrightarrow{K} (r',i')$, $(r,j) \in [r,i]_n$, *and* $(r,j) \models K_1 nonrep$, *then* $\forall m < n : [r,i]_m \xrightarrow{K} [r',i']_m$ *and* $(r,j) \xrightarrow{K} (r',j')$ *for some* $(r',j') \in [r',i']_n$ *and* $(r',j') \models K_1 nonrep$.

- *If* (r,i) *encodes ID id at time t, then there exist a point* (r',i') *and* $m > 0$ *such that:*

  - $(r,i) \sim_c (r',i')$,
  - (r',i') *encodes an ID id′ in the start state at time* $t+m$ *and* $id(\vdash)^m id_{t+m}$.

Suppose that the lemma holds and let $M \in \mathcal{C}_{(nl)}$ be such that $M,(r_0,i_0) \models \phi_A^{nl}$. Then $M,(r_0,i_0)$ encodes id_0 at time 0, and by part 2 of this lemma there exist $t_0 < t_1 < \cdots$ such that $t_0 = 0$, for all ℓ: $id_{m_\ell} \vdash^+ id_{m_{\ell+1}}$ and id_{m_ℓ} is in the start state. This implies that A is recurrent. □

It remains to prove lemma 6.4.9. For the first part, suppose that $(r,i) \xrightarrow{K} (r',i')$. By definition of $\xrightarrow{K}$ $(r',i') \models p_\Delta$ and there must exists some (r'',i'') such that $(r,i) \sim_1 (r'',i'') \sim_2 (r',i')$ and $(r'',i'') \models \neg p_\Delta$. By no learning, there exist some $j' > i'$ and $j'' > i''$ such that $(r,j) \sim_1 (r'',j'') \sim_2 (r',j')$ Since (r,j) is not 1-repeating, it follows by lemma 6.4.8 that $\forall m < n : [r,i]_m \sim_1 [r'',i'']_m$ and $(r'',j'') \in [r'',i'']_n$. By ϕ_2, $(r,j) \models p_\Delta$ and $(r'',j'') \models \neg p_\Delta$. Therefore, by ϕ_{nl}, $(r'',j'') \models K_2 nonrep$, and again by lemma 6.4.8 $\forall m < n : [r'',i'']_m \sim_2 [r',i']_m$ and $(r',j') \in [r',i']_n$. Since p_Δ is constant on runs, $\forall m < n : [r,i]_m \xrightarrow{K} [r',i']_m$ and $(r,j) \xrightarrow{K} (r',j') \in [r'i']_n$ and by ϕ_{nl}, $(r',j') \models K_1 nonrep$.

For the second part of the lemma, note that by ϕ_{rec}^{nl}, $M,(r,i)$ satisfies the formula:

$$C\Diamond(\Box\neg time_2 \wedge C(p_\Delta \wedge time_2 \rightarrow startstate) \wedge K_1 nonrep).$$

Therefore, there must exist some $j \geq i$ such that $(r,j) \models \Box\neg time_2 \wedge C(p_\Delta \wedge time_2 \rightarrow startstate) \wedge K_1 nonrep$. Since (r,i) encodes an ID at time t, $[r,i]_{t+3} \models time_2$. Therefore (r,j) must be in some interval after the $(t+3)$-rd interval of (r,i). Suppose $(r,j) \in [r,i]_{t+3+m}$ for $m > 0$. ϕ_2 ensures that we can take m $\xrightarrow{K}$ steps from (r,i): $(r,i) \xrightarrow{K} (r_1,i_1) \xrightarrow{K} (r_2,i_2) \xrightarrow{K} \cdots \xrightarrow{K} (r_m,i_m)$. By repeatedly applying the first part of the lemma, it follows that for all $n \leq t+3+m$: $[r,i]_n \xrightarrow{K} [r_1,i_1]_n \xrightarrow{K} [r_2,i_2]_n \xrightarrow{K} \cdots \xrightarrow{K} [r_m,i_m]_n$ and $(r,j) \sim_c (r_m,j_m)$ for some $(r_m,j_m) \in [r_m,i_m]_{t+3+m}$. It follows that (r_m,i_m) encodes an ID id' at time $m+t$ such that $id(\vdash)^m id'$. Thus, $[r_m,i_m]_{t+m+3} \models time_2$. Since $(r,j) \sim_c (r_m,j_m) \in [r_m,i_m]_{t+3+m}$, it follows that $(r_m,i_m) \models startstate$, and therefore, id' is in the start state. □

Finally, suppose that A is recurrent. We show that ϕ_A^{nl} is satisfiable with respect to $\mathcal{C}_{(nl)}$. Suppose $id_0 \vdash id_1 \vdash id_2 \vdash \cdots$ is an infinite computation of A that starts on the empty tape and reenters its start state infinitely often. Let M be the $\mathcal{C}_{(nl,nf,sync)}$ model such that $M,(r_0,0) \models \phi_A$ as defined on page 105. Recall that $M = (\{r_\ell \mid \ell \in \mathbf{N}\}, \sim_1, \sim_2, \pi)$ and

- $\sim_1$ is the rst closure of $\{\langle (r_{2\ell},i),(r_{2\ell+1},i)\rangle \mid \ell, i \in \mathbf{N}\}$
- $\sim_2$ is rst closure of $\{\langle (r_{2\ell+1},i),(r_{2\ell+2},i)\rangle \mid \ell, i \in \mathbf{N}\}$,
- $p_\Delta \in \pi(r_\ell,i)$ iff ℓ even; $tick \in \pi(r_\ell,i)$ iff i even.

$M, (r_{2\ell}, 0)$ encodes id_ℓ at time ℓ and $M, (r_0, 0)$ satisfies ϕ_1 through ϕ_7 and ϕ_{start}.

It remains to show that we can define an assignment on propositional variable q in such a way that $(r_0, 0)$ satisfies ϕ_{nl} and ϕ_A^{nl} as well.

Since our computation is recurrent, there exist $t_0 \leq t_1 \leq t_2 \leq \ldots$ such that $\forall m : t_m > m$ and id_{t_m} is in the start state. Now define

$$q \in \pi(r_\ell, i) \Leftrightarrow i \leq t_{\ell/2} + 3.$$

Since *nonrep* was defined as $q \wedge \Diamond\Box\neg q$, it follows that if $i \leq t_{\ell/2}+3$ then $\forall \ell' \geq \ell : (r_{\ell'}, i) \models$ *nonrep*. By definition of $\sim_1$ and $\sim_2$, it follows that if ℓ is even and $(r_\ell, i) \models K_1 \mathit{nonrep}$ then $(r_\ell, i) \models K_1(\neg p_\Delta \rightarrow K_2 \mathit{nonrep})$ and $(r_\ell, i) \models KK_1 \mathit{nonrep}$. Therefore $M, (r_0, 0) \models \phi_{nl}$ as required.

For ϕ_{rec}^{nl}, suppose $(r_\ell, 0) \models p_\Delta$. Then ℓ is even, say $\ell = 2m$ and $(r_{2m}, 0)$ encodes id_m at time m. Since $t_m > m$, $(r_{2m}, t_m+3) \models K_1 \mathit{nonrep} \wedge \Box\neg \mathit{time}_2$. If $(r_{\ell'}, t_m+3) \models p_\Delta \wedge \mathit{time}_2$, then $(r_{\ell'}, 0)$ encodes id_{t_m}. Since id_{t_m} is in the start state, $(r_{\ell'}, t_m + 3) \models \mathit{startstate}$. Therefore $(r_\ell, t_m + 3) \models C(p_\Delta \wedge \mathit{time}_2 \rightarrow \mathit{startstate})$. Now we have proved that

$$(r_\ell, 0) \models \Diamond(\Box\neg \mathit{time}_2 \wedge C(p_\Delta \wedge \mathit{time}_2 \rightarrow \mathit{startstate}) \wedge K_1 \mathit{nonrep})).$$

And since r_ℓ was an arbitrary p_Δ run, it follows that $(r_0, 0)$ satisfies ϕ_{rec}^{nl}. □

6.5 Variations on a Theme by Ladner and Reif

Theorem 6.5.1 *The satisfiability problems for $K\overline{L}_{(\geq 2)}$ and $CK\overline{L}_{(\geq 2)}$ with respect to $\mathcal{C}_{(nf,nl,uis)}$ are Σ_1^1-complete.*

Since the Σ_1^1 upper bounds for these classes follow directly from [HV89], it suffices to prove the Σ_1^1 lower bound for $K\overline{L}_{(2)}$. In [LR86], Ladner and Reif prove that the satisfiability problem for $KB_{(2)}$ is undecidable with respect to $\mathcal{C}_{(nf,nl,uis)}$. In particular, they construct for each deterministic Turing machine A a formula that forces a run to encode an infinite computation of A. As pointed out in [HV89], their proof can be trivially modified to obtain a Σ_1^1 lower bound for $KL_{(2)}$. We will use the main idea of Ladner and Reif's proof to obtain for each nondeterministic Turing machine A a $K\overline{L}_{(2)}$ formula that encodes the recurrence problem for A.

Let A be a 1-tape right-infinite NTM. Suppose A has state space S, start state $s_0 \in S$; tape alphabet Γ; $b \in \Gamma$: the blank; and transition function δ. Let Δ be the set $\Gamma \cup \{\#, \$\} \cup (S \times \Gamma)$. We start by giving the definitions from [LR86], extended to nondeterministic Turing Machines.

We view the IDs of A as *finite* strings of the form: $\$a_0\$a_1\$\ldots\$a_n\$$ with $a_0 \ldots a_n \in \Gamma^*(S \times \Gamma)\Gamma^*$. A starts on the empty tape in state s_0 and we define the start ID of A id_0 as the string $\$\langle s_0, b\rangle\$$. Define an infinite computation as an infinite string over Δ of the form: $\#^{m_0} id_0 \#^{m_1} id_1 \#^{m_2} \ldots$ with for each i: $m_i > 0$, $id_i \vdash id_{i+1}$, $|id_i| = 2i + 3$.

Define a function *collapse* : $\Delta^\omega \cup \Delta^* \rightarrow \Delta^\omega \cup \Delta^*$, that replaces multiple contiguous occurrences of the same symbol by one occurrence, that is:

$$\begin{aligned} \mathit{collapse}(a_0^{m_0} a_1^{m_1} a_2^{m_2} \ldots) &= a_0 a_1 a_2 \ldots \in \Delta^\omega \text{ (if for all } i\text{: } m_i > 0, a_i \neq a_{i+1}) \\ \mathit{collapse}(a_0^{m_0} a_1^{m_1} a_2^{m_2} \ldots a_r^{m_r}) &= \mathit{collapse}(a_0^{m_0} a_1^{m_1} a_2^{m_2} \ldots a_r^{\omega}) = \\ & \quad a_0 a_1 a_2 \ldots a_r \text{ (if for all } i\text{: } m_i > 0, a_i \neq a_{i+1}) \end{aligned}$$

Suppose σ and τ are infinite computations of the form:

$$\begin{array}{l} \sigma = \#\#\#\#\#\#id_0\#\#\#id_1\#\#\#id_2\#\#\#\cdots \\ \tau = \# \quad id_0 \;\#\; id_1 \quad \# \quad id_2 \quad \# \quad id_3 \quad \#\cdots \end{array}$$

Analogously to [LR86], we can define a function $N : \Delta^6 \to Pow(\Delta^6)$ that verifies the matching of these strings. If σ and τ are infinite computations as given, then $\forall i(\tau_i, \ldots, \tau_{i+5}) \in N(\sigma_i, \ldots, \sigma_{i+5})$.

The following lemma shows how we can use N to determine if A has an infinite computation.

Lemma 6.5.2 (LR) *Let σ, τ be infinite strings over Δ such that:*

1. $\sigma \in \#^6\$((\neg\{\#,\$\}\$)^*\#^3\$)^\omega$
2. $\tau \in (\neg\$\$)^\omega$
3. $\forall i : (\tau_i, \ldots, \tau_{i+5}) \in N(\sigma_i, \ldots, \sigma_{i+5})$
4. *$collapse(\sigma) = collapse(\tau)$*

Then σ and τ are infinite computations.

We will construct a formula ψ_A such that ψ_A is satisfiable with respect to $\mathcal{C}_{(nf,nl,sync)}$ if and only if A is recurrent. As in [LR86], we will encode two infinite computations on each run. Again we partition runs into an infinite number of intervals by the propositional variable *tick*. Let ψ_1 be the formula:

$$E\Box((tick \to \Diamond\neg tick) \wedge (\neg tick \to \Diamond tick))$$

If $(r_0, i_0) \models \psi_1$ then by (uis, nl), *tick* alternates on all runs.

Since we will encode two strings on each run, we need to encode 2 elements of Δ per point. Therefore we introduce for each $c \in \Delta$ two propositional variables s_c and t_c. Let ψ_2 be the conjunction of the following formulas:

$$E\Box(\bigvee_{c\in\Delta}(s_c \wedge \neg \bigvee_{d\in\Delta, d\neq c} s_d)) \wedge E\Box(\bigvee_{c\in\Delta}(t_c \wedge \neg \bigvee_{d\in\Delta, d\neq c} t_d))$$

If $(r_0, i_0) \models \psi_2$ and $(r, i) \sim_k (r_0, i_0)$ $(k \in \{1, 2\})$ then each point on r after i encodes exactly 2 elements of Δ, say a point encodes s = a and t = b if exactly s_a and t_b hold. An interval $[r, i]_n$ encodes $s = a$ [resp. $t = b$] if each point in that interval encodes $s = a$ [$t = b$]. Now we can define the encoding of strings on a run: (r, i) encodes $s^\omega = \sigma$ [$t^\omega = \tau$] if for all n : $[r, i]_n$ encodes $s = \sigma_n$ [$t = \tau_n$].

The formula ψ_A that we will construct will force the existence of strings σ and τ fulfilling the conditions of lemma 6.5.2. Following [LR86], we will encode *collapse*(σ) and *collapse*(τ) on the current run and σ and τ on other runs. We use propositional variable *coll*, constant on runs, to discriminate between the current run, where we want *coll* to hold, and the runs that encode the noncollapsed computations. The following formula ψ_3 will take care of this.

$$coll \wedge \neg K_1 coll \wedge E((coll \to \Box coll) \wedge (\neg coll \to \Box\neg coll))$$

As in [LR86], we will enforce the following situation: if $(r_0, i_0) \models \psi_A$ then there exist strings σ and τ fulfilling conditions 1,2,3 and 4 of lemma 6.5.2 such that:

- if $(r_0, i_0) \sim_1 (r, i)$ and $(r, i) \models \neg \mathit{coll}$ then (r, i) encodes $s^\omega = \sigma$ and $t^\omega = \tau$
- (r_0, i_0) encodes $s^\omega = \mathit{collapse}(\sigma)$ and $t^\omega = \mathit{collapse}(\tau)$

If we have constructed ψ_A and $(r_0, i_0) \models \psi_A$, then by lemma 6.5.2, (r_0, i_0) encodes an infinite computation of A. We can then easily force this computation to be recurrent by adding the following conjunct ψ_{rec} to ψ_A:

$$\Box\Diamond \bigvee_{a\in\Gamma} s_{<s_0,a>}$$

We now turn to the construction of the formula ψ_A. First of all we have to make sure that if $(r_0, i_0) \sim_1 (r, i) \sim_1 (r', i')$ and $(r, i), (r', i') \models \neg \mathit{coll}$, then (r, i) and (r', i') encode the same strings. As a first step, we force synchrony for $\sim_1$, by the following formula ψ_4:

$$K_1\Box((\mathit{tick} \rightarrow K_1 \mathit{tick}) \wedge (\neg \mathit{tick} \rightarrow K_1 \neg \mathit{tick}))$$

If $(r_0, i_0) \models \psi_1, \ldots, \psi_4$ and $(r, i) \sim_1 (r_0, i_0)$ then by (nl) for each $j_0 \geq i_0$ there exists some $j \geq i$ such that $(r, j) \sim_1 (r_0, j_0)$. By lemma 6.3.1, it follows that for all n : $[r, i]_n \sim_1 [r_0, i_0]_n$. We can now force all $\neg \mathit{coll}$ runs to encode the same strings, by formula ψ_5:

$$K_1\Box(\neg \mathit{coll} \wedge s_c \rightarrow K_1(\neg \mathit{coll} \rightarrow s_c)) \wedge K_1\Box(\neg \mathit{coll} \wedge t_c \rightarrow K_1(\neg \mathit{coll} \rightarrow t_c))$$

If $(r_0, i_0) \models \psi_1, \ldots, \psi_5$, $(r_0, i_0) \sim_1 (r, i) \sim_1 (r', i')$ and $(r, i), (r', i') \models \neg \mathit{coll}$ then, by ψ_4, $\forall n : [r, i]_n \sim_1 [r', i']_n$. By ψ_2 each point on r_0 and r encodes exactly two elements in Δ and therefore by ψ_5 $\forall n : [r, i]_n$ encodes $s = a$ $[t = b]$ if and only if $[r', i']_n$ encodes $s = a$ $[t = b]$.

We have to ensure that $\neg \mathit{coll}$ runs encode strings σ and τ fulfilling the conditions of lemma 6.5.2, i.e. $\sigma \in \#^6\$((\neg\{\#, \$\}\$)^*\#^3\$)^\omega$ and $\tau \in (\neg\$\$)^\omega$ such that $\forall i : (\tau_i, \ldots, \tau_{i+5}) \in N(\sigma_i, \ldots, \sigma_{i+5})$.

Following [LR86], it can easily be seen that these conditions can be checked locally: we can construct a local condition such that if for all n this condition holds for $\sigma_n \ldots \sigma_{n+5}, \tau_n \ldots \tau_{n+5}$ (taking some extra care for the first seven symbols), then σ and τ are of the appropriate form. This is the reason why Ladner and Reif can force this situation using just one run. Obviously, one run won't suffice in our situation, since we don't have the nexttime operator. However, we can force the local condition for one interval at each run.

If $(r, i) \sim_1 (r_0, i_0)$, and $(r, i) \models \neg \mathit{coll}$ we use (r, i) to check the local condition for some interval $[r, i]_n$. In order to do this, we have to be able to distinguish the first 7 intervals of (r, i). We mark interval 0 to interval 6 of (r, i) by propositional variables $\mathit{start_0}$ to $\mathit{start_6}$, by the following formula ψ_6.

$$K_1(\neg \mathit{coll} \rightarrow \mathit{start_0} \wedge \bigwedge_{k=0}^{5} \mathit{1\text{-}int}(\mathit{start_k}, \mathit{start_{k+1}}) \wedge \mathit{1\text{-}int}(\mathit{start_6}, \mathit{up_{start}}))$$

We can check the local condition for some interval by just looking at that interval and its 5 successors. We mark 6 consecutive intervals on each $\neg \mathit{coll}$ run by $\mathit{arg_0}$ to $\mathit{arg_5}$, using formula ψ_7:

$$K_1(\neg \mathit{coll} \rightarrow \bigwedge_{k=0}^{4} \mathit{1\text{-}int}(\mathit{arg_k}, \mathit{arg_{k+1}}) \wedge \mathit{1\text{-}int}(\mathit{arg_5}, \mathit{up_{arg}}))$$

It is now easy to construct a formula ψ_8 such that: if $(r_0, i_0) \models \psi_1, \ldots, \psi_8$ and $(r_0, i_0) \sim_1 (r, i)$, $(r, i) \models \neg \mathit{coll}$ and $[r, i]_n \models \mathit{arg}_0$ then $[r, i]_n$ fulfills the local condition.

By ψ_5 we know that for each $(r', i') \sim_1 (r_0, i_0)$ such that $(r', i') \models \neg\mathit{coll}$ the following holds: $\forall n : [r, i]_n$ encodes $s = a\ [t = b]$ if and only if $[r', i']_n$ encodes $s = a\ [t = b]$. Therefore $[r', i']_n$ fulfills the local conditions as well. We have to make sure that each interval is checked, i.e. for each n there must be some $(r, i) \sim_1 (r_0, i_0)$ such that $(r, i) \models \neg\mathit{coll}$ and $[r, i]_n \models \mathit{arg}_0$. The following formula ψ_9 provides for this:

$$\Box\neg K_1 \neg(\neg\mathit{coll} \wedge \mathit{arg}_0)$$

Suppose $(r_0, i_0) \models \psi_1, \ldots, \psi_9$. Choose some $(r_0, j_0) \in [r_0, i_0]_n$. By ψ_9 there exists some $(r, j) \sim_1 (r_0, j_0)$ such that $(r, j) \models \neg\mathit{coll} \wedge \mathit{arg}_0$. By (nf), there is some $i \leq j$ such $(r, i) \sim_1 (r_0, i_0)$. Then by lemma 6.3.1, $(r, j) \in [r, i]_n$ and $[r, i]_n \models \mathit{arg}_0$ as required.

We have proved that if $(r_0, i_0) \models \psi_1, \ldots, \psi_9$, there exist σ and τ fulfilling conditions 1,2 and 3 of lemma 6.5.2 such that for all $(r, i) \sim_1 (r_0, i_0)$ with $(r, i) \models \neg\mathit{coll}$: (r, i) encodes $s^\omega = \sigma$ and $t^\omega = \tau$.

In order to apply lemma 6.5.2, we have to ensure that condition 4 holds as well, i.e. $\mathit{collapse}(\sigma) = \mathit{collapse}(\tau)$. We will let (r_0, i_0) encode $s^\omega = \mathit{collapse}(\sigma)$ and $t^\omega = \mathit{collapse}(\tau)$ and force the two strings encoded by (r_0, i_0) to be equal. First we will force the condition for τ. Let ψ_{10} be the formula:

$$\Box(t_c \rightarrow K_1 t_c)$$

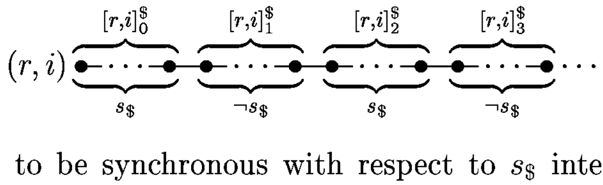
If $(r_0, i_0) \models \psi_1, \ldots, \psi_{10}$ then by ψ_3 there exists some $(r, i) \sim_1 (r_0, i_0)$ such that $(r, i) \models \neg\mathit{coll}$. By ψ_4, $\forall n : [r_0, i_0]_n \sim_1 [r, i]_n$. Since (r, i) encodes $t^\omega = \tau$ and each point on r_0 encodes exactly one value for t, (r_0, i_0) encodes $t^\omega = \tau = \mathit{collapse}(\tau)$.

We ensure that (r_0, i_0) encodes two equal strings by the following formula ψ_{11}:

$$\Box(t_c \leftrightarrow s_c)$$

If $(r_0, i_0) \models \psi_1, \ldots, \psi_{11}$ then (r_0, i_0) encodes $s^\omega = t^\omega = \mathit{collapse}(\tau)$.

Finally, we force (r_0, i_0) to encode $s^\omega = \mathit{collapse}(\sigma)$. As in [LR86] we will use $\sim_2$ to simulate the collapse function for σ. Since $\sigma \neq \mathit{collapse}(\sigma)$, $\sim_2$ must behave differently from $\sim_1$. Therefore we partition the runs into different intervals, this time using our propositional variable $s_\$$. Let $[r, i]^\$_n$ be the n-th $s_\$$-interval of (r, i).



We can now force $\sim_2$ to be synchronous with respect to $s_\$$ intervals. Let ψ_{12} be the formula:

$$K_2\Box((s_\$ \rightarrow K_2 s_\$) \wedge (\neg s_\$ \rightarrow K_2 \neg s_\$))$$

If $(r_0, i_0) \models \psi_1, \ldots, \psi_{12}$ and $(r, i) \sim_2 (r_0, i_0)$ then by (nl) $\forall j_0 \geq i_0$ there exists some $j \geq i$ such that $(r, j) \sim_2 (r_0, j_0)$. By lemma 6.3.1, it follows that $\forall n : [r, i]^\$_n \sim_2 [r_0, i_0]^\$_n$. We want the n-th $s_\$$ intervals of (r_0, i_0) and (r, i) to encode the same value for s. The following formula ψ_{13} will take care of this:

$$K_2\Box(s_c \rightarrow K_2 s_c).$$

Suppose $(r_0, i_0) \models \psi_1, \ldots, \psi_{13}$, $(r_0, i_0) \sim_2 (r, i)$ and (r, i) encodes $s^\omega = \alpha$ and $\mathit{collapse}(\alpha) \in (\neg\$\$)^\omega$. Then $[r, i]^\$_n$ must encode $s = (\mathit{collapse}(\alpha))_n$, since the $s_\$$ intervals take adjacent identical s-symbols together. By ψ_{13}, $[r_0, i_0]^\$_n$ encodes $s = (\mathit{collapse}(\alpha))_n$ as well. Since we already know that (r_0, i_0) encodes $s^\omega = \mathit{collapse}(\tau) \in (\neg\$\$)^\omega$, the $s_\$$ and tick intervals of (r_0, i_0) coincide. Thus, $[r_0, i_0]_n$ encodes $s = (\mathit{collapse}(\alpha))_n$ and therefore (r_0, i_0) encodes $s^\omega = \mathit{collapse}(\alpha)$. Since we want (r_0, i_0) to encode $\mathit{collapse}(\sigma)$, and $\mathit{collapse}(\sigma) \in (\neg\$\$)^\omega$, we just need to force the existence of some $(r, i) \sim_2 (r_0, i_0)$ such that (r, i) encodes $s^\omega = \sigma$. Let ψ_{14} be the formula:

$$\neg K_2 \neg(\neg \mathit{coll} \wedge \mathit{arg}_0 \wedge \mathit{1\text{-}int}(\mathit{start}_0, \mathit{start}_1) \wedge \Box(\mathit{arg}_0 \rightarrow s_\#))$$

If $(r_0, i_0) \models \psi_1, \ldots, \psi_{14}$ then there exists some $(r, i) \sim_2 (r_0, i_0)$ such that $(r, i) \models \neg \mathit{coll}$ and start_0 holds exactly at $[r, i]_0$ and $[r, i]_0$ encodes $s = \#$. By $(\mathit{uis}, \mathit{nl})$, there must exist some j such that $(r, j) \sim_1 (r_0, i_0)$. By ψ_3, $(r, j) \models \neg \mathit{coll}$, and therefore (r, i) encodes $s^\omega = \sigma$ and start_0 holds exactly at $[r, j]_0$. But then (r, i) also encodes $s^\omega = \sigma$, and by ψ_{13} it follows that (r_0, i_0) encodes $s^\omega = \mathit{collapse}(\sigma)$.

Finally, let ψ_A be the conjunction of ψ_1 through ψ_{14}. If $(r_0, i_0) \models \psi_A$ then by lemma 6.5.2 (r_0, i_0) encodes $s^\omega = \mathit{collapse}(\sigma)$ and $\mathit{collapse}(\sigma)$ is an infinite computation of A. If (r_0, i_0) satisfies ψ_{rec} as well, then $\mathit{collapse}(\sigma)$ is an infinite recurrent computation of A. Therefore, if $\psi_A \wedge \psi_{rec}$ is satisfiable with respect to $\mathcal{C}_{(nf,nl,uis)}$, then A is recurrent.

To conclude the proof of theorem 6.5.1 it remains to check that $\psi_A \wedge \psi_{rec}$ is satisfiable with respect to $\mathcal{C}_{(nf,nl,uis)}$ if A is recurrent. Suppose $\mathit{id}_0 \vdash \mathit{id}_1 \vdash \mathit{id}_2 \vdash \cdots$ is an infinite computation of A that starts on the empty tape and reenters its start state infinitely often. We will construct a model $M = (\{r_\ell | \ell \in \mathrm{N}\}, \sim_1, \sim_2, \pi) \in \mathcal{C}_{(nf,nl,uis)}$ such that $M, (r_0, 0) \models \psi_A \wedge \psi_{rec}$:

- $\sim_1$ is the reflexive symmetric and transitive closure of $\{\langle (r, i), (r', i) \rangle | r, r' \in R\}$,
- $\mathit{tick} \in \pi(r, i)$ iff i even,
- $\mathit{coll} \in \pi(r_\ell, i)$ iff $\ell = 0$.

Then processor 1 does not learn or forget and has a unique initial state. And for all runs r : $[r, 0]_n = \{(r, n)\}$.

Suppose σ and τ are infinite computations of the form:

$$\sigma = \#\#\#\#\#\#\mathit{id}_0\#\#\#\mathit{id}_1\#\#\#\mathit{id}_2\#\#\#\cdots$$
$$\tau = \# \quad \mathit{id}_0 \ \# \quad \mathit{id}_1 \quad \# \quad \mathit{id}_2 \quad \# \quad \mathit{id}_3 \quad \#\cdots$$

Then $\forall i$: $(\tau_i, \ldots, \tau_{i+5}) \in N(\sigma_i, \ldots, \sigma_{i+5})$ and collapse (σ) = collapse (τ).

Define π on variables s_c and t_c $(c \in \Delta)$ such that $(r_0, 0)$ encodes $(s)^\omega = \mathit{collapse}(\sigma)$ and $(t)^\omega = \mathit{collapse}(\tau)$ and for each $k > 0$ $(r_k, 0)$ encodes $(s)^\omega = \sigma$ and $(t)^\omega = \tau$. On each run r_{k+1} we mark the first 8 points with $\mathit{start}_0, \ldots, \mathit{start}_6, \mathit{up}_{\mathit{start}}$. and points $k, \ldots, k+6$ with $\mathit{arg}_0, \ldots, \mathit{arg}_5, \mathit{up}_{\mathit{arg}}$. Then $(r_0, 0)$ satisfies all conjuncts of ψ_A that use only K_1 as epistemic operator. Since the computation encoded by $(r_0, 0)$ is recurrent, $(r_0, 0)$ satisfies ψ_{rec} as well.

We have to define $\sim_2$ in such a way that it fulfils ψ_{12} and ψ_{13}:

$$K_2\Box((s_\$ \rightarrow K_2 s_\$) \wedge (\neg s_\$ \rightarrow K_2 \neg s_\$))$$

$$K_2\Box(s_c \rightarrow K_2 s_c)$$

Take $n_0 < n_1 < n_2 < \ldots$ such that:

$n_0 = 0$, $\sigma_{n_i} \neq \sigma_{n_{i+1}}$ and $n_i \leq j < n_{i+1} \Rightarrow \sigma_j = (collapse(\sigma))_i$

And define:

$\sim_2$ as the rst closure of $\{\langle (r_0, i), (r_\ell, j)\rangle | \ell > 0,\ n_i \leq j < n_{i+1}\}$.

Then processor 2 does not learn or forget and has a unique initial state. It follows that $M \in \mathcal{C}_{(nf,nl,uis)}$.

It remains to verify that $M, (r_0, 0) \models \psi_{12} - \psi_{14}$. We know that (r_0, i) encodes $s = (collapse(\sigma))_i$ and $\forall r' \neq r_0\ \forall j (n_i \leq j < n_{i+1} \Leftarrow (r', j)$ encodes $s = (collapse(\sigma))_i)$. Therefore, $(r_0, 0) \models \psi_{12} \wedge \psi_{13}$. The last conjunct that has to be satisfied by $(r_0, 0)$ is ψ_{14}:

$$\neg K_2 \neg(\neg coll \wedge arg_0 \wedge \mathit{1\text{-}int}(start_0, start_1) \wedge \Box(arg_0 \to s_\#))$$

Since $(r_0, 0) \sim_2 (r_1, 0)$, $(r_1, 0) \models \neg coll \wedge s_\#$ and the only point on r_1 where $start_0$ is true is $(r_1, 0)$, $(r_0, 0) \models \psi_{14}$ and the claim follows. □

Theorem 6.5.3 *The satisfiability problems for $K\overline{L}_{(\geq 2)}$ and $CK\overline{L}_{(\geq 2)}$ with respect to $\mathcal{C}_{(nl,uis)}$ are RE-complete.*

In the proof of theorem 6.4.6, we have shown that no learning enables us to force intervals to be synchronous with respect to finite prefixes of runs. Now we can add an extra conjunct to formula ψ_A of the previous proof to encode the halting problem. This gives us a RE lower bound for satisfiability. The corresponding upper bound follows from [HV89]. □

Two Dimensional Temporal Logic

We can apply the techniques of the proof of theorem 6.5.1 to obtain a Σ_1^1 lower bound for the satisfiability problem of two-dimensional temporal logic with only the two “sometimes” operators as temporal connectives. The models for two-dimensional temporal logic are two-dimensional grids, infinite to the right and upwards, i.e. each point is a pair (i, j) of natural numbers[1]. Let $\overline{L}_2$ be the propositional language with operators $\Diamond_r$ (sometimes to the right) and $\Diamond_u$ (sometimes upwards) such that: $M, (i, j) \models \Diamond_r \phi \Leftrightarrow \exists i' \geq i : M, (i', j) \models \phi$, and $M, (i, j) \models \Diamond_u \phi \Leftrightarrow \exists j' \geq j : M, (i, j') \models \phi$.

Theorem 6.5.4 *The satisfiability problem for $\overline{L}_{(2)}$ is Σ_1^1-hard.*

We will briefly sketch how to construct a formula ϕ_A such that ϕ_A is satisfiable if and only if A is recurrent. First of all, note that if ϕ_A is satisfiable, then there exists a model M such that $M, (0, 0) \models \phi_A$. Therefore, we will assume that the constructed formula is satisfiable in $(0, 0)$. Introduce two propositional variables $tick_r$ and $tick_u$ such that $tick_r$ alternates on horizontal runs and is constant on vertical runs, and $tick_u$ alternates on vertical runs and is constant on horizontal runs. We will use two-dimensional intervals $[(n, m)]$ $(n, m \in \mathbf{N})$ to take over the role of points:

$$\begin{aligned}[(n, m)] := \{(i, j) :\ & (i, 0) \text{ in the } n\text{-th } tick_r \text{ interval of } (0, 0) \text{ and} \\ & (0, j) \text{ in the } m\text{-th } tick_u \text{ interval of } (0, 0)\}\end{aligned}$$

[1] Note that the present complexity results no longer go through if we admit more general model classes (cf [Ven92]).

Now we do have a gridlike structure: for all n and m, there exist i_1, i_2, j_1, j_2 such that $[(n,m)] := \{(i,j) | i_1 \leq i \leq i_2 \wedge j_1 \leq j \leq j_2\}$

Using the same trick as in the proof of theorem 6.5.1, we can force the existence of strings σ and τ fulfilling conditions 1,2 and 3 of lemma 6.5.2, such that each horizontal run encodes $s^\omega = \sigma$ and $t^\omega = \tau$ on its consecutive horizontal $tick_r$ intervals.

We want $(0,0)$ to encode the same strings $collapse(\sigma)$ and $collapse(\tau)$ vertically, i.e. on its consecutive $tick_u$ intervals. We use two new sets of propositional variables $\{\hat{s}_c : c \in \Delta\}$ and $\{\hat{t}_c : c \in \Delta\}$ and force the values of $\hat{s}$ and $\hat{t}$ to be constant on horizontal runs. To ensure that $(0,0)$ encodes $\hat{t}^\omega = collapse(\tau)$, we mark the diagonal with propositional variable D, i.e. $(i,j) \models D$ if and only $(i,j) \in [(n,n)]$ for some n. Now we can force points on the diagonal to encode the same values for t and $\hat{t}$. This ensures that $(0,0)$ encodes $\hat{t} = collapse(\tau)$.

It is easy to ensure that the strings vertically encoded on $(0,0)$ are equal. To ensure that $(0,0)$ encodes $\hat{s} = collapse(\sigma)$, we partition horizontal runs into intervals with $s_\$$ and vertical runs into intervals with $\hat{s}_\$$. Since the value of $s_\$$ is constant on vertical runs, and the value of $\hat{s}_\$$ is constant on horizontal runs, this gives us again a two-dimensional gridlike structure. We can mark the diagonal in this structure with D_s, and force the points where D_s holds to encode the same values for s and $\hat{s}$. This ensures that $(0,0)$ encodes $\hat{s} = collapse(\sigma)$. By lemma 6.5.2, $(0,0)$ encodes an infinite computation of A, and it is trivial to add a conjunct that forces this computation to be recurrent. □

6.6 A Generic Reduction from Linear to Branching Time

Intuitively, the satisfiability problems for branching time languages are harder then the corresponding satisfiability problems for linear time. We will show that we can uniformly reduce the satisfiability problems for $CK\overline{L}_{(m)}$ and $K\overline{L}_{(m)}$ to the corresponding satisfiability problem for $CK\overline{B}_{(m)}$ and $K\overline{B}_{(m)}$, thus corroborating our intuition.

There is an obvious way to associate a branching time model with each linear time model and vice versa: suppose $M = (R, \sim_1, \ldots \sim_m, \pi)$ is a linear time model, then M is a branching time model as well; if $M = (F, \sim_1, \ldots \sim_m, \pi)$ is a branching time model then we define the corresponding linear time model M_L as $(R_F, \sim_1, \ldots \sim_m\ \pi)$ (recall that R_F is the set of branches in F). Note that if $M \in \mathcal{D}$ where $\mathcal{D}$ is one of our sixteen classes of models, then $M_L \in \mathcal{D}$.

Theorem 6.6.1 *There exists a polynomial time computable function f from $CK\overline{L}_{(m)}$ to $CK\overline{B}_{(m)}$ formulas such that:*

1. *For every linear time model M and all (r,i): $M,(r,i) \models \phi \Rightarrow M,(r,i) \models f(\phi)$, and*

2. *For every branching time model M and all (r,i): $M,(r,i) \models f(\phi) \Rightarrow M_L,(r,i) \models \phi$.*

Moreover, if $\phi \in K\overline{L}_{(m)}$ then $f(\phi) \in K\overline{B}_{(m)}$.

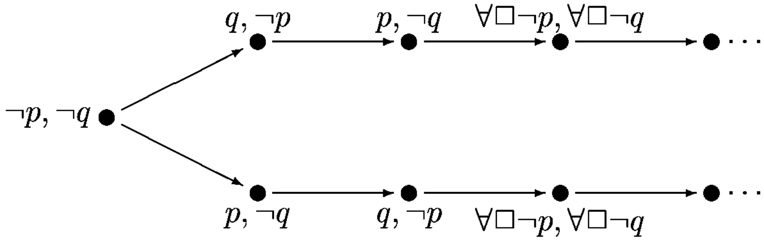
As a first attempt, we take g to be the function that replaces all $\Diamond$ occurrences in a $CK\overline{L}_{(m)}$ formula by $\forall\Diamond$. Function g does not satisfy the conditions. Take for example the following $\overline{L}$ formula ϕ :

$$\Diamond p \wedge \Diamond q \rightarrow \Diamond(p \wedge \Diamond q) \vee \Diamond(q \wedge \Diamond p)$$

Then ϕ is valid in all linear time models, but $g(\phi)$:

$$\forall\Diamond p \land \forall\Diamond q \rightarrow \forall\Diamond(p \land \forall\Diamond q) \lor \forall\Diamond(q \land \forall\Diamond p)$$

is not valid in the following branching time model:


$q, \neg p$
$p, \neg q$
$\forall\Box\neg p, \forall\Box\neg q$
$\neg p, \neg q$
$p, \neg q$
$q, \neg p$
$\forall\Box\neg p, \forall\Box\neg q$

The problem is that in branching time models $\exists\Diamond g(\psi)$ can hold, while $\forall\Diamond g(\psi)$ does not hold. Given a $C\overline{KL}_{(m)}$ formula ϕ, we will exclude this situation for all subformulas $\Diamond\psi$ of ϕ in all relevant points. Define a function *lin* from $C\overline{KL}_{(m)}$ to $C\overline{KB}_{(m)}$ formulas:

$$lin(p) = T;\ lin(\neg\phi) = lin(\phi);\ lin(\phi \land \psi) = lin(\phi) \land lin(\psi)$$
$$lin(K_k\phi) = K_k lin(\phi);\ lin(E\phi) = E\ lin(\phi);\ lin(C\phi) = C\ lin(\phi)$$
$$lin(\Diamond\phi) = (\forall\Diamond g(\phi) \leftrightarrow \exists\Diamond g(\phi)) \land \forall\Box lin(\phi)$$

By an easy induction on the structure of formula ϕ, we can prove the following lemma:

Lemma 6.6.2 *If $M = (F, \sim_1, \ldots \sim_m, \pi)$ is a branching time model such that $M, (r,i) \models lin(\phi)$ then $M, (r,i) \models g(\phi) \Leftrightarrow M_L, (r,i) \models \phi$.*

The only interesting case is $\Diamond\phi$. Suppose $M, (r,i) \models lin(\Diamond\phi)$. Then $M, (r,i) \models \forall\Diamond\phi \leftrightarrow \exists\Diamond\phi$ and $\forall i' \geq i : M, (r,i') \models lin(\phi)$.

$$\begin{aligned} M,(r,i) \models g(\Diamond\phi) &\Rightarrow M,(r,i) \models \forall\Diamond g(\phi) \Rightarrow \exists i' \geq i : M,(r,i') \models g(\phi) \\ &\Rightarrow \exists i' \geq i : M_L,(r,i') \models \phi \Rightarrow M_L,(r,i) \models \Diamond\phi \\ M_L,(r,i) \models \Diamond\phi &\Rightarrow \exists i' \geq i : M_L,(r,i') \models \phi \Rightarrow \exists i' \geq i : M,(r,i') \models g(\phi) \Rightarrow \\ &\Rightarrow M,(r,i) \models \exists\Diamond g(\phi) \Rightarrow M,(r,i) \models \forall\Diamond g(\phi) \Rightarrow \\ &\Rightarrow M,(r,i) \models g(\Diamond\phi) \end{aligned}$$

□

Let $f(\phi) := g(\phi) \land lin(\phi)$; we will prove that f fulfills the conditions of theorem 6.6.1. Suppose M is a linear time model and $M, (r,i) \models \phi$. If we view M as a branching time model, then for all points (r',i') in M and all branching time formulas $\psi : M, (r',i') \models \forall\Diamond\psi \leftrightarrow \exists\Diamond\psi$. Therefore, $M, (r,i) \models lin(\phi)$ and by lemma 6.6.2, $M, (r,i) \models g(\phi)$. But then $M, (r,i) \models f(\phi)$ as required. If M is a branching time model and $M, (r,i) \models f(\phi)$, then by lemma 6.6.2 $M_L, (r,i) \models \phi$. □

Corollary 6.6.3 *If $\mathcal{D}$ is one of our sixteen classes of models then the satisfiability problem for $C\overline{KL}_{(m)}$ (resp. $\overline{KL}_{(m)}$) with respect to $\mathcal{D}$ is polynomial time reducible to the satisfiability problem for $C\overline{KB}_{(m)}$ (resp. $\overline{KB}_{(m)}$) with respect to $\mathcal{D}$.*

Combining the lower bounds from the previous sections with the upper bounds from [HV89] leads to a complete classification for branching time.

Corollary 6.6.4

- *The satisfiability problems for* $CK\overline{B}_{(\geq 2)}$ *with respect to* $\mathcal{C}_{(nf)}$, $\mathcal{C}_{(nf,uis)}$, $\mathcal{C}_{(nf,sync)}$, $\mathcal{C}_{(nf,nl)}$, $\mathcal{C}_{(nf,sync,uis)}$, $\mathcal{C}_{(nf,nl,sync)}$, $\mathcal{C}_{(nl,sync)}$ *and* $\mathcal{C}_{(nl)}$ *are* Σ^1_1*-complete.*

- *The satisfiability problems for* $CK\overline{B}_{(\geq 2)}$ *and* $K\overline{B}_{(\geq 2)}$ *with respect to* $\mathcal{C}_{(nf,nl,uis)}$ *are* Σ^1_1*-complete.*

- *The satisfiability problems for* $CK\overline{B}_{(\geq 2)}$ *and* $K\overline{B}_{(\geq 2)}$ *with respect to* $\mathcal{C}_{(nl,uis)}$ *are RE-complete.*

Bibliography

[BDG88] J. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I.* EATCS Monographs on Theoretical Computer Science 11, Springer Verlag, 1988.

[BHP82] M. Ben-Ari, J. Halpern, and A. Pnueli. Deterministic propositional dynamic logic: finite models, complexity and completeness. *Journal of Computer and System Sciences* 25, 402–417, 1982.

[Ben84] J. van Benthem. Correspondence Theory. In *Handbook of Philosophical Logic* 2, D. Reidel, Dordrecht, 1984.

[Ber66] R. Berger. The undecidability of the dominoe problem. *Mem. Amer. Math. Soc.* 66, 1966.

[Bla1] P. Blackburn. Nominal tense logic. *Notre Dame Journal of Formal Logic.* To appear.

[Bla2] P. Blackburn. Modal logic and attribute value structures. *Diamonds and Defaults*, Studies in Logic, Language and Information, Kluwer. To appear.

[BS] P. Blackburn and E. Spaan. A modal perspective on the computational complexity of attribute value grammar. *Journal of Logic, Language and Information.* To appear.

[Bul66] R. Bull, That all normal extensions of $S4.3$ have the finite model property. *Zeitschrift für Mathematische Logik and Grundlagen der Mathematik* 12, 341–344, 1966.

[Bul70] R. Bull, An approach to tense logic. *Theoria* 12, 171–182, 1970.

[Car92] B. Carpenter. *The Logic of Typed Feature Structures.* Cambridge University Press, 1992.

[CKP81] A. Chandra, D. Kozen, and L. Stockmeyer, Alternation. *Journal of the ACM* 28, 114–133, 1981.

[Emd83] P. van Emde Boas. Dominoes are forever. In *1st GTI Workshop*, 75–95, 1983.

[EH85] E. Emerson J. and Halpern. Decision procedures and expressiveness in the temporal logic of branching time. *Journal of Computer and System Sciences* 30, 194–211, 1985.

[Eva87] R. Evans. Towards a formal specification for defaults in GPSG. In *Categories, Polymorphism and Unification*, Centre for Cognitive Science, Edinburgh University, Scotland; and ITLI, University of Amsterdam, the Netherlands, 1987.

[EG89] R. Evans and G. Gazdar. The semantics of DATR. In A. Cohn (ed.) *AISB-89, Proceedings of the Seventh Conference of the Society for the Study of Artificial Intelligence and the Simulation of Behaviour*, Pitman, London, 1989.

[Fin72] K. Fine. Logics containing $S4.3$, *Zeitschrift für Mathematische Logik and Grudlagen der Mathematik* 17, 371–376, 1972.

[Fin74] K. Fine. An ascending chain of $S4$ logics, *Theoria* 40, 110–116, 1974.

[Fin85] K. Fine. Logics containing K4, part II. *Journal of Symbolic Logic* 50, 619–651, 1985.

[FS] K. Fine and R. Schurz. Transfer theorems for multimodal logics. In *Proceedings Arthur Prior Memorial Conference, Christchurch, New Zealand.* To appear.

[FL79] M. Fischer and R. Ladner. Propositional dynamic logic of regular programs, *Journal of Computer and System Sciences* 18, 194–211, 1979.

[GPT86] G. Gargov, S. Passy. and T. Tinchev. Modal environment for boolean speculations (preliminary report). In *Mathematical Logic and its Applications. Proceedings of the 1986 Godel Summer School and Conference Bulgaria*, Plenum Press, New York, 1986.

[GP88] G. Gargov and S. Passy. Determinism and looping in combinatory PDL, *Theoretical Computer Science* 61, 259–277, 1988.

[GG] G. Gargov and V. Goranko. Modal logic with names, *Journal of Philosophical Logic.* To appear.

[GKPS85] G. Gazdar, E. Klein, G. Pullum, and I. Sag. *Generalized Phrase Structure Grammar*, Basil Blackwell, 1985.

[GPC+88] G. Gazdar, G. Pullum, R. Carpenter, E. Klein, T. Hukari, and R. Levine, Category structures. *Computational Linguistics* 14, 1–19, 1988.

[Gol89] R. Goldblatt. Varieties of Complex Algebras, *Annals of Pure and Applied Logic* 44, 173–242, 1989.

[GP92] V. Goranko and S. Passy, Using the universal modality: gains and questions, *Journal of Logic and Computation* 2, 5–30, 1992.

[Hal87] J. Halpern. Using reasoning about knowledge to analyze distributed systems. *Annual Review of Computer Science*, Annual Reviews Inc., 1987.

[HM85] J. Halpern and M. Moses. A guide to the modal logics of knowledge and belief. In *Proceedings 9th International Joint Conference on Artificial Intelligence*, 480–490, 1985.

[HV89] J. Halpern and M. Vardi, The complexity of reasoning about knowledge and time, I: Lower Bounds. *Journal of Computer and System Sciences* 38, 195–237, 1989.

[Har83] D. Harel. Recurring dominoes: making the highly undecidable highly understandable. In *Proc. of the Conference on Foundations of Computing Theory*, Springer Lecture Notes in Computer Science 158, 177–194, 1983.

[Har86] D. Harel. Effective transformations on infinite trees, with applications to high undecidability, dominoes, and fairness. *Journal of the ACM* 33(1), 224–248, 1986.

[Har84] D. Harel. Dynamic Logic. In *Handbook of Philosophical Logic* 2, D. Reidel, Dordrecht, 1984.

[HPS83] D. Harel, A. Pnueli, and J. Stavi. Propositional dynamic logic of nonregular programs. *Journal of Computer and System Sciences* 26, 222-243, 1983.

[HR] W. van der Hoek and M. de Rijke. Generalized quantifiers and modal logic, LP-91-01, ITLI Prepublication Series, University of Amsterdam, The Netherlands. *Journal of Logic, Language and Information.* To appear.

[HC84] G. Hughes and M. Cresswell. *A Companion to Modal Logic*, Methuen & Co. Ltd, 1984.

[Imm87] N. Immerman. Languages that capture complexity classes. *SIAM Journal on Computing* 17(5), 760–778, 1987.

[KB82] K. Kaplan and J. Bresnan. Lexical functional grammar: a formal system for grammatical representation. In *The Mental Representation of Grammatical Relations*, MIT Press, 173–281, 1982.

[Kar84] L. Karttunen, Features and values. In *Proceedings of the 10th International Conference on Computational Linguistics and the 22nd Annual Meeting of the Association for Computational Linguistics*, Stanford, California, 28–33, 1984.

[KR86] R. Kasper and W. Rounds A logical semantics for feature structures. In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*, Columbia University, New York, 257–266, 1986.

[KR90] R. Kasper and W. Rounds. The logic of unification in grammar. *Linguistics and Philosophy* 13, 33–58, 1990.

[Kel91] B. Keller. *Feature Logics, Infinitary Descriptions and The Logical Treatment of Grammar.* PhD thesis, School of Cognitive and Computing Sciences, University of Sussex, United Kingdom, 1991.

[Kra89] M. Kracht. On the Logic of Category Definition. *Computational Linguistics* 15, 111–113, 1989.

[KW91] M. Kracht and F. Wolter. Properties of independently axiomatizable bimodal logics. *Journal of Symbolic Logic*, 56(4), 1469–1485, 1991.

[Lad77] R. Ladner. The computational complexity of provability in systems of modal logic, *SIAM Journal on Computing* 6(3), 467–480, 1977.

[Lew78] H. Lewis. Complexity of solvable cases of the decision problem for the predicate calculus. In *19th FOCS*, 35–47, 1978.

[LR86] R. Ladner and J. Reif. The logics of distributed protocols, In *Theoretical Aspects of Reasoning About Knowledge: Proceedings of the 1986 Conference*, 207–221. 1986.

[Mos] L. Moss. Completeness theorems for logics of feature structures, Indiana University Logic Group Preprint No. IULG-91-2. In *Proceedings of the MSRI Workshop on Logic From Computer Science*, Springer Verlag. To appear.

[ON80] H. Ono and A. Nakamura. On the size of refutation Kripke models for some linear modal and tense logics. *Studia Logica* 39, 325-333, 1980.

[Par80] R. Parikh. Propositional logics of programs: systems, models, and complexity. In *Seventh Annual ACM Symposium on Principles of Programming Languages*, 186–192, 1980.

[PT85] S. Passy and T. Tinchev. PDL with Data Constants, *Information Processing Letters* 20, pages 35-41. 1985.

[PT91] S. Passy and T. Tinchev. An essay in combinatory dynamic logic, *Information and Computation* 93, 263 – 332, 1991.

[PS84] F. Pereira and S. Shieber. The semantics of grammar formalisms seen as computer languages. In *Proceedings of the 10th International Conference on Computational Linguistics and the 22nd Annual Meeting of the Association for Computational Linguistics*, Stanford, California, 123–129, 1984.

[PS] C. Pollard and I. Sag. *Information-Based Syntax and Semantics: Volume 1 - Fundamentals.* CSLI Lecture Notes 13, Stanford.

[Pol] J. Pollard. Sorts in unification-based grammar and what they mean. *Unification in Linguistic Analysis.* To appear.

[Pra79] V. Pratt. Models of program logics. In *Proceedings of the 20th IEEE Symposium on Foundations of Computer Science*, 115–122, 1979.

[Pri67] A. Prior. *Past, Present and Future.* Oxford University Press, 1967.

[Rea91] M. Reape. *An Introduction to the Semantics of Unification-Based Grammar Formalisms.* DYANA deliverable R3.2.A, Centre for Cognitive Science, University of Edinburgh, Scotland, 1991.

[Rob71] R. Robinson. Undecidability and nonperiodicity for tilings of the plane. *Inventiones Math.* 12, 177–209, 1971.

[Rog67] H. Rogers. *Theory of Recursive Functions and Effective Comptability.* McGraw-Hill, 1967.

[Roo] D. Roorda. Dyadic Modalities and Lambek Calculus. In *Diamonds and Defaults*, Studies in Logic, Language and Information, Kluwer. To appear.

[RK86] W. Rounds R. and Kasper. A Complete Logical Calculus for Record Structures Representing Linguistic Information. In *Proceedings of the 15th Annual Symposium on Logic in Computer Science*, Cambridge, Massachussets, 1986.

[Sah75] H. Sahlqvist. Completeness and correspondence in the first and second order semantics for modal logics. In *Proc. of the third Scandinavian Logic Symposium*, North-Holland, Amsterdam, 1975.

[Sav70] W. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences* 4, 177–192, 1970.

[Sch90] K. Schild. A Correspondence Theory for Terminological Logics: Preliminary Report. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, Sydney, Australia, 1990.

[Sel88] A. Selman. Natural self-reducible sets. *SIAM Journal on Computing* 17, 989–996, 1988.

[Shi86] S. Shieber. *An Introduction to Unification-Based Approaches to Grammar.* Chicago University Press, 1986.

[SC85] A. Sistla and E. Clarke The complexity of propositional linear temporal logics. *Journal of the ACM* 32, 733–749, 1985.

[Smo89] G. Smolka. Feature Constraint Logics for Unification Grammars. IWBS Report 93, IWBS, IBM Deutschland, Stuttgart, Germany, 1989.

[Spa90] E. Spaan. Nexttime is not necessary. In *Proceedings of the Third Conference on Theoretical Aspects of Reasoning About Knowledge*, 241–256. Morgan Kaufman Publishers, 1990.

[Spa] E. Spaan. The complexity of propositional tense logics, In *Diamonds and Defaults*, Studies in Logic, Language and Information, Kluwer. To appear.

[Urq81] A. Urquhart. Decidability and the finite model property, *Journal of Philosophical Logic* 10, 367–370, 1981.

[Ven92] Y. Venema. *Many-Dimensional Modal Logic*, PhD thesis, University of Amsterdam, the Netherlands, 1992.

[Zak92] M. Zakharyaschev. Canonical formulas for K4, Part II. Manuscript, 1992.