

COMPLEXITY RESULTS FOR MODAL LOGIC WITH RECURSION VIA TRANSLATIONS AND TABLEAUX

LUCA ACETO ^{a,b}, ANTONIS ACHILLEOS ^a, ELLI ANASTASIADI ^{a,c},
ADRIAN FRANCALANZA ^d, AND ANNA INGÓLFSDÓTTIR ^a

^a Dept. of Computer Science, Reykjavik University, Reykjavik, Iceland
e-mail address: luca@ru.is, antonios@ru.is, annai@ru.is

^b Gran Sasso Science Institute L'Aquila, Italy

^c Dept. of Information Technology, Uppsala University, Uppsala, Sweden
e-mail address: elli.anastasiadi@it.uu.se

^d Dept. of Computer Science, ICT, University of Malta, Msida, Malta
e-mail address: adrian.francalanza@um.edu.mt

ABSTRACT. This paper studies the complexity of classical modal logics and of their extension with fixed-point operators, using translations to transfer results across logics. In particular, we show several complexity results for multi-agent logics via translations to and from the μ -calculus and modal logic, which allow us to transfer known upper and lower bounds. We also use these translations to introduce terminating and non-terminating tableau systems for the logics we study, based on Kozen's tableau for the μ -calculus and the one of Fitting and Massacci for modal logic. Finally, we describe these tableaux with μ -calculus formulas, thus reducing the satisfiability of each of these logics to the satisfiability of the μ -calculus, resulting in a general 2EXP upper bound for satisfiability testing.

1. INTRODUCTION

In this paper, we study a family of multi-modal logics with fixed-point operators that are interpreted over restricted classes of Kripke models. One can consider these logics as extensions of the usual multi-agent logics of knowledge and belief [FHMV95] by adding recursion to their syntax or of the μ -calculus [Koz83] by interpreting formulas over different classes of frames and thus giving an epistemic interpretation to the modalities. We define *translations* between these logics, and we demonstrate how one can rely on these translations to prove finite-model theorems, complexity bounds, and tableau termination for each logic in the family, in a systematic fashion.

This work has been funded by the projects “Open Problems in the Equational Logic of Processes (OPEL)” (grant no. 196050), “Epistemic Logic for Distributed Runtime Monitoring” (grant no. 184940), “Mode(1)s of Verification and Monitorability” (MoVeMent) (grant no 217987) of the Icelandic Research Fund, and the project “Runtime and Equational Verification of Concurrent Programs” (ReVoCoP) (grant 222021) of the Reykjavik University Research Fund.

Modal logic comes in several variations [BvBW06]. Some of these, such as multi-modal logics of knowledge and belief [FHMV95], are of particular interest to Epistemology and other application areas. Semantically, the classical modal logics used in epistemic (but also other) contexts result from imposing certain restrictions on their models. On the other hand, the modal μ -calculus [Koz83] can be seen as an extension of the smallest normal modal logic \mathbf{K} with greatest and least fixed-point operators, νX and μX respectively.

In this article, we explore the situation where one allows both recursion (*i.e.* fixed-point) operators in a multi-modal language and imposes restrictions on the semantic models. In particular, we are interested in the complexity of satisfiability for the resulting logics. Satisfiability for the μ -calculus is known to be EXP-complete [Koz83], while for the modal logics between \mathbf{K} and $\mathbf{S5}$ the problem is PSPACE-complete or NP-complete, depending on whether they have Negative Introspection [Lad77, HR07]. In the multi-modal case, satisfiability for those modal logics becomes PSPACE-complete, and is EXP-complete with the addition of a common knowledge operator [HM92].

Our primary method for proving complexity results is through translations to and from the multi-modal μ -calculus. We show that we can use surprisingly simple translations from modal logics without recursion to the base modal logic \mathbf{K}_n , that is, \mathbf{K} with n agent-indexed modalities, reproving the PSPACE upper bound for these logics (Theorem 3.6 and Corollary 3.11). These translations and our constructions to prove their correctness do not generally transfer to the corresponding logics with recursion. We present translations from specific logics to the μ -calculus and back, and we discuss the remaining open cases. We discover, through the existence of our translations, that several restrictions induced on the frames do not affect the complexity of the satisfiability problem. As a result, we prove that all logics with axioms among D , T , and 4, and the least-fixed-point fragments of logics that also have B , have their satisfiability in EXP, and a matching lower bound for the logics with axioms from D , T , and B (Corollaries 3.26 and 3.27).

To address the limitations of the translation-based approach to proving complexity results, we then present tableaux for the discussed logics, based on the ones by Kozen for the μ -calculus [Koz83], and by Fitting and Massacci for modal logic [Fit72, Mas94]. We give tableau-termination conditions for every logic with a finite-model property (Theorem 4.5). Based on these tableaux, we give a polynomial-space decision procedure for the single-agent logics with axioms among D , T , and 4, thus proving that satisfiability for these logics is PSPACE-complete (Theorem 5.6). Finally, we give a general satisfiability-preserving translation from each logic to the μ -calculus, using our tableaux. The translation describes a tableau branch with an exponentially larger μ -calculus formula, establishing a 2EXP-upper bound for all our logics. The decidability of the satisfiability problem for logics with B or 5 was an open problem (see, for example, Remark 3.22, based on [DL13]).

The addition of recursive operators to modal logic increases expressiveness. An important example is that of *common knowledge* or *common belief*, which can be expressed with a greatest fixed-point thus: $\nu X.(\varphi \wedge \bigwedge_{\alpha} [\alpha]X)$, where φ describes the property that is common knowledge, and α ranges over a (finite) set of agents. However, the combination of epistemic logics and fixed-points can potentially express more interesting epistemic concepts. For instance, in the context of a belief interpretation, the formula $\mu X. \bigvee_{\alpha} ([\alpha]\varphi \vee [\alpha]X)$, can be viewed as claiming that there is a rumour of φ . It would be interesting to see what other meaningful sentences of epistemic interest one can express using recursion. Furthermore, the family of logics we consider allows each agent to behave according to a different logic. This flexibility allows one to mix different interpretations of modalities, such as a temporal

interpretation for one agent and an epistemic interpretation for another. Such logics can even resemble hyper-logics [CS10] if a set of agents represents different streams of computation. Combinations of epistemic and temporal or hyper-logics have recently been used to express safety and privacy properties of systems [BMP15].

The paper is organized as follows. Section 2 gives the necessary background and an overview of current results. Section 3 defines a class of translations that provide us with several upper and lower bounds, and identifies conditions under which they can be composed. In Section 4 we give tableaux systems for our multi-modal logics with recursion. Section 5 proves a PSPACE upper bound for the single-agent logics with axioms among D , T , and 4; Section 6 presents the translation that establishes the 2EXP upper bounds for all the modal logics with recursion. We conclude in Section 7 with a set of open questions and directions.

This is an extended version of [AAA⁺22], with complete proofs for our results. The results from Sections 5 and 6 are also new in this version.

2. DEFINITIONS AND BACKGROUND

This section introduces the logics that we study and the necessary background on the complexity of modal logic and the μ -calculus.

2.1. The Multi-Modal Logics with Recursion. We start by defining the syntax of the logics.

Definition 2.1. We consider formulas constructed from the following grammar:

$$\begin{array}{l} \varphi, \psi \in L ::= p \quad | \neg p \quad | \mathbf{tt} \quad | \mathbf{ff} \quad | X \quad | \varphi \wedge \psi \quad | \varphi \vee \psi \\ \quad \quad \quad | \langle \alpha \rangle \varphi \quad | [\alpha] \varphi \quad | \mu X. \varphi \quad | \nu X. \varphi, \end{array}$$

where X comes from a countably infinite set of logical (or fixed-point) variables, LVAR, α from a finite set of agents, AG, and p from a finite set of propositional variables, PVAR. When $\text{AG} = \{\alpha\}$, $\Box\varphi$ stands for $[\alpha]\varphi$, and $\Diamond\varphi$ for $\langle \alpha \rangle \varphi$. We also write $[A]\varphi$, with $A \subseteq \text{AG}$, to mean $\bigwedge_{\alpha \in A} [\alpha]\varphi$ and $\langle A \rangle \varphi$ for $\bigvee_{\alpha \in A} \langle \alpha \rangle \varphi$. As usual, an empty conjunction stands for \mathbf{tt} and empty disjunction stands for \mathbf{ff} .

A formula is closed when every occurrence of a variable X is in the scope of some recursive operator νX or μX . Henceforth we consider only closed formulas, unless we specify otherwise.

Moreover, for recursion-free closed formulas we associate the notion of *modal depth*, which is the maximum nesting depth of the modal operators¹. The modal depth of φ is defined inductively as:

- $md(p) = md(\neg p) = md(\mathbf{tt}) = md(\mathbf{ff}) = 0$, where $p \in \text{PVAR}$,
- $md(\varphi \vee \psi) = md(\varphi \wedge \psi) = \max(md(\varphi), md(\psi))$, and
- $md([\alpha]\varphi) = md(\langle \alpha \rangle \varphi) = 1 + md(\varphi)$, where $\alpha \in \text{AG}$.

¹The modal depth of recursive formulas can be either zero, or infinite. However, this will not affect the results we present later on.

$$\begin{aligned}
\llbracket \mathbf{tt}, \rho \rrbracket &= W, & \llbracket \mathbf{ff}, \rho \rrbracket &= \emptyset, & \llbracket p, \rho \rrbracket &= \{s \mid p \in V(s)\}, & \llbracket \neg p, \rho \rrbracket &= W \setminus \llbracket p, \rho \rrbracket, \\
\llbracket [\alpha]\varphi, \rho \rrbracket &= \{s \mid \forall t. sR_\alpha t \text{ implies } t \in \llbracket \varphi, \rho \rrbracket\}, & \llbracket \varphi_1 \wedge \varphi_2, \rho \rrbracket &= \llbracket \varphi_1, \rho \rrbracket \cap \llbracket \varphi_2, \rho \rrbracket, \\
\llbracket \langle \alpha \rangle \varphi, \rho \rrbracket &= \{s \mid \exists t. sR_\alpha t \text{ and } t \in \llbracket \varphi, \rho \rrbracket\}, & \llbracket \varphi_1 \vee \varphi_2, \rho \rrbracket &= \llbracket \varphi_1, \rho \rrbracket \cup \llbracket \varphi_2, \rho \rrbracket, \\
\llbracket \mu X.\varphi, \rho \rrbracket &= \bigcap \{S \mid S \supseteq \llbracket \varphi, \rho[X \mapsto S] \rrbracket\}, & \llbracket X, \rho \rrbracket &= \rho(X), \\
\llbracket \nu X.\varphi, \rho \rrbracket &= \bigcup \{S \mid S \subseteq \llbracket \varphi, \rho[X \mapsto S] \rrbracket\}.
\end{aligned}$$

TABLE 1. Semantics of modal formulas on a model $\mathcal{M} = (W, R, V)$. We omit \mathcal{M} from the notation.

We assume that each recursion variable X appears in a unique fixed-point formula $\text{fx}(X)$, which is either of the form $\mu X.\varphi$ or $\nu X.\varphi$. If $\text{fx}(X)$ is a least-fixed-point (*resp.* greatest-fixed-point) formula, then X is called a least-fixed-point (*resp.* greatest-fixed-point) variable. We can define a partial order over fixed-point variables, such that $X \leq Y$ iff $\text{fx}(X)$ is a subformula of $\text{fx}(Y)$, and $X < Y$ when $X \leq Y$ and $X \neq Y$. If X is \leq -minimal among the free variables of φ , then we define the *closure* of φ to be $\text{cl}(\varphi) = \text{cl}(\varphi[\text{fx}(X)/X])$, where $\varphi[\psi/X]$ is the usual substitution operation. Note that if φ is closed, then $\text{cl}(\varphi) = \varphi$.

We define $\text{sub}(\varphi)$ as the set of subformulas of φ . The size of φ , denoted by $|\varphi|$, is the cardinality of $\text{sub}(\varphi)$, which is bounded by the length of φ as a string of symbols. Negation, $\neg\varphi$, and implication, $\varphi \rightarrow \psi$, can be defined in the usual way. Then, we define $\overline{\text{sub}}(\varphi) = \text{sub}(\varphi) \cup \{\neg\psi \mid \psi \in \text{sub}(\varphi)\}$.

Semantics. We interpret formulas on the states of a *Kripke model*. A Kripke model, or simply model, is a triple $\mathcal{M} = (W, R, V)$ where W is a nonempty set of states, $R \subseteq W \times \text{AG} \times W$ is a transition relation, and $V : W \rightarrow 2^{\text{PVAR}}$ determines the propositional variables that are true at each state. (W, R) is called a *frame*. We usually write $(u, v) \in R_\alpha$ or $uR_\alpha v$ instead of $(u, \alpha, v) \in R$, or uRv , when AG is a singleton $\{\alpha\}$.

Formulas are evaluated in the context of an *environment* $\rho : \text{LVAR} \rightarrow 2^W$, which gives values to the logical variables. For an environment ρ , variable X , and set $S \subseteq W$, we write $\rho[X \mapsto S]$ for the environment that maps X to S and all $Y \neq X$ to $\rho(Y)$. The semantics for our formulas is given through a function $\llbracket - \rrbracket_{\mathcal{M}}$, defined in Table 1. The semantics of $\neg\varphi$ is constructed as usual, where $\llbracket \neg X, \rho \rrbracket_{\mathcal{M}} = W \setminus \rho(X)$.

We call a pair (\mathcal{M}, s) of a model and a state a *pointed model*. We sometimes use $(\mathcal{M}, s) \models_\rho \varphi$ for $s \in \llbracket \varphi, \rho \rrbracket_{\mathcal{M}}$, and as the environment has no effect on the semantics of a closed formula φ , we often drop it from the notation and write $(\mathcal{M}, s) \models \varphi$ or $s \in \llbracket \varphi \rrbracket_{\mathcal{M}}$. If $(\mathcal{M}, s) \models \varphi$, we say that φ is true, or satisfied, in s . When the particular model does not matter, or is clear from the context, we may omit it.

Depending on how we further restrict our syntax and the model, we can describe several logics. Without further restrictions, the resulting logic is the μ -calculus [Koz83]. The max-fragment (*resp.* min-fragment) of the μ -calculus is the fragment that only allows the νX (*resp.* the μX) recursive operator. If $|\text{AG}| = k \in \mathbb{N}^+$ and we allow no recursive operators (or recursion variables), then we have the basic modal logic \mathbf{K}_k (or \mathbf{K} , if $k = 1$), and further restrictions on the frames can result in a wide variety of modal logics (see, for instance, [BdRV01]). We give names to the following frame conditions, or frame constraints,

for the case where $\text{AG} = \{\alpha\}$. These conditions correspond to the usual axioms for normal modal logics — see [BvBW06, BdRV01, FHMV95], which we will revisit in Section 3.

D : R is <i>serial</i> : $\forall s \exists t, sRt$; T : R is <i>reflexive</i> : $\forall s, sRs$; B : R is <i>symmetric</i> : $\forall s, t, sRt$ implies tRs ;	4 : R is <i>transitive</i> : $\forall s, t, r$, if sRt and tRr then sRr ; 5 : R is <i>euclidean</i> : $\forall s, t, r$, if sRt and sRr , then tRr .
--	---

We consider modal logics that are interpreted over models that satisfy a combination of these constraints for each agent. T , which we call *Factivity*, is a special case of D , which we call *Consistency*. Constraint 4 is *Positive Introspection* and 5 is called *Negative Introspection*.² Given a logic \mathbf{L} and constraint c , we write $\mathbf{L} + c$ for the logic that is interpreted over all models with frames that satisfy all the constraints of \mathbf{L} and c . The name of a single-agent logic is a combination of the constraints that apply to its frames, including K , if the constraints are contained in $\{4, 5\}$. Therefore, logic \mathbf{D} is $\mathbf{K} + D$, \mathbf{T} is $\mathbf{K} + T$, \mathbf{B} is $\mathbf{K} + B$, $\mathbf{K4}$ is $\mathbf{K} + 4$, $\mathbf{D4}$ is $\mathbf{K} + D + 4 = \mathbf{D} + 4$, and so on. We use the special names $\mathbf{S4}$ for $\mathbf{T4}$ and $\mathbf{S5}$ for $\mathbf{T45}$. We define a (multi-agent) logic \mathbf{L} on AG as a map from agents to single-agent logics. \mathbf{L} is interpreted on Kripke models of the form (W, R, V) , where for every $\alpha \in \text{AG}$, (W, R_α) is a frame for $\mathbf{L}(\alpha)$.

For a logic \mathbf{L} , we write \mathbf{L}^μ for the logic that results from \mathbf{L} after we allow recursive operators in the syntax — in case they were not allowed in \mathbf{L} . Furthermore, if for every $\alpha \in \text{AG}$, $\mathbf{L}(\alpha)$ is the same single-agent logic \mathbf{L} , we write \mathbf{L} as \mathbf{L}_k , where $|\text{AG}| = k$. Therefore, the μ -calculus is \mathbf{K}_k^μ .

From now on, unless we explicitly say otherwise, by a logic, we mean one of the logics we have defined above. We call a formula satisfiable for a logic \mathbf{L} , if it is satisfied in some state of a model for \mathbf{L} .

Example 2.2. For a formula φ , we define $\text{Inv}(\varphi) = \nu X.(\varphi \wedge [\text{AG}]X)$. $\text{Inv}(\varphi)$ asserts that φ is true in *all* reachable states, or, alternatively, it can be read as an assertion that φ is common knowledge. We dually define $\text{Eve}(\varphi) = \mu X.(\varphi \vee \langle \text{AG} \rangle X)$, which asserts that φ is true in *some* reachable state.

In what follows, we will often use the concept of *bisimilarity*, when arguing that two different Kripke models satisfy the same formulas.

Definition 2.3 (Bisimulation). A binary relation $S \subseteq W_1 \times W_2$ over the set of states of two Kripke models $\mathcal{M}_1 = (W_1, R_1, V_1)$ and $\mathcal{M}_2 = (W_2, R_2, V_2)$ is a bisimulation if and only if whenever $s_1 S s_2$, for each agent α :

- for each state s'_1 , if $(s_1, s'_1) \in R_{1,\alpha}$, then there exists a state s'_2 such that $(s_2, s'_2) \in R_{2,\alpha}$ and $s'_1 S s'_2$,
- for each state s'_2 , if $(s_2, s'_2) \in R_{2,\alpha}$, then there exists a state s'_1 such that $(s_1, s'_1) \in R_{1,\alpha}$ and $s'_1 S s'_2$, and
- $p \in V(s_1)$ iff $p \in V(s_2)$, for each propositional variable p .

Two pointed models (\mathcal{M}_1, s_1) and (\mathcal{M}_2, s_2) are bisimilar, notation $(\mathcal{M}_1, s_1) \sim (\mathcal{M}_2, s_2)$, if they are related by a bisimulation S .

²These are names for properties or axioms of a logic. When we refer to these conditions as conditions on a frame or model, we may refer to them with the name of the corresponding condition on relations, namely: seriality, reflexivity, symmetry, transitivity, and euclidicity.

It is well-known that bisimulation is an equivalence relation.

Theorem 2.4 (Hennessy-Milner Theorem [HM85, BW18, CHVB18]). *The μ -calculus is bisimulation-invariant. I.e. for every formula $\varphi \in L$ and pointed models $(\mathcal{M}_1, s_1), (\mathcal{M}_2, s_2)$, if $(\mathcal{M}_1, s_1) \sim (\mathcal{M}_2, s_2)$ and $(\mathcal{M}_1, s_1) \models \varphi$, then $(\mathcal{M}_2, s_2) \models \varphi$.*

2.2. Known Results. For logic \mathbf{L} , the satisfiability problem for \mathbf{L} , or \mathbf{L} -satisfiability, is the problem that asks, given a formula φ , if φ is satisfiable. Similarly, the model checking problem for \mathbf{L} asks if φ is true at a given state of a given finite model.

Ladner [Lad77] established the classical result of PSPACE-completeness for the satisfiability of \mathbf{K} , \mathbf{T} , \mathbf{D} , $\mathbf{K4}$, $\mathbf{D4}$, and $\mathbf{S4}$ as well as NP-completeness for the satisfiability of $\mathbf{S5}$. Halpern and Rêgo later characterized the NP–PSPACE gap for one-action logics by the presence or absence of Negative Introspection [HR07], resulting in Theorem 2.5 below. Later, Rybakov and Shkatov proved the PSPACE-completeness of \mathbf{B} and \mathbf{TB} in [RS18].

There is plenty of relevant work on the μ -calculus on restricted frames, mainly in its single-agent form. Alberucci and Facchini examine the alternation hierarchy of the μ -calculus over reflexive, symmetric, and transitive frames in [AF09]. D’Agostino and Lenzi have studied the μ -calculus over different classes of frames in great detail. In [DL10], they reduce the μ -calculus over finite transitive frames to first-order logic. In [DL13], they prove that $\mathbf{S5}^\mu$ -satisfiability is NP-complete, and that the two-agent version of $\mathbf{S5}^\mu$ does not have the finite-model property. In [DL15], they consider finite symmetric frames, and they prove that \mathbf{B}^μ -satisfiability is in 2EXP, and EXP-hard. They also examine planar frames in [DL18], where they show that the alternation hierarchy of the μ -calculus over planar frames is infinite.

Theorem 2.5 [Lad77, HR07, RS18, DL13]. *If $\mathbf{L} \in \{\mathbf{K}, \mathbf{T}, \mathbf{D}, \mathbf{B}, \mathbf{TB}, \mathbf{K4}, \mathbf{D4}, \mathbf{S4}\}$, then \mathbf{L} -satisfiability is PSPACE-complete; and $\mathbf{L} + 5$ -satisfiability and $(\mathbf{L} + 5)^\mu$ -satisfiability are NP-complete.*

Theorem 2.6 [HM92]. *If $k > 1$ and \mathbf{L} has a combination of constraints from $D, T, 4, 5$ and no recursive operators, then \mathbf{L}_k -satisfiability is PSPACE-complete.*

Remark 2.7. We note that Halpern and Moses in [HM92] only prove these bounds for the cases of $\mathbf{K}_k, \mathbf{T}_k, \mathbf{S4}_k, \mathbf{KD45}_k$, and $\mathbf{S5}_k$; moreover D’Agostino and Lenzi in [DL13] only prove the NP-completeness of satisfiability for $\mathbf{S5}^\mu$. However, it is not hard to see that their respective methods also work for the rest of the logics of Theorems 2.5 and 2.6. \square

Theorem 2.8 [Koz83]. *The satisfiability problem for the μ -calculus is EXP-complete.*

When a formula is given in conjunction with a model and one of its states, the question of whether the given formula holds in that fixed model at the given state is called the *model checking problem*.

Theorem 2.9 [EJS01]. *The model checking problem for the μ -calculus is in $\text{NP} \cap \text{coNP}$.³*

Finally we have the following results about the complexity of satisfiability, when we have recursive operators. Theorem 2.10 and Proposition 2.11 have already been observed in [AAF120].

³In fact, the problem is known to be in $\text{UP} \cap \text{coUP}$, and can be solved in pseudo-polynomial time — see [Jur98, CJK⁺22, JL17a, Leh18a].

Theorem 2.10. *The satisfiability problem for the min- and max-fragments of the μ -calculus is EXP-complete, even when $|\text{AG}| = 1$.*

Proof sketch. It is known that satisfiability for the min- and max-fragments of the μ -calculus (on one or more action) is EXP-complete. It is in EXP due to Theorem 2.8 and as shown in [Pra81], these fragments suffice to describe the PDL formula that is constructed by the reduction used in [FL79] to prove EXP-hardness for PDL. Therefore, that reduction can be adjusted to prove that satisfiability for the min- and max-fragments of the μ -calculus is EXP-complete. \square

As observed in Example 2.2, one can express that formula φ is common knowledge using the greatest fixed-point operator. Since validity for \mathbf{L}_k with common knowledge (and without recursive operators) and $k > 1$ is EXP-complete [HM92]⁴, \mathbf{L}_k^μ is EXP-hard.

Proposition 2.11. *Satisfiability for \mathbf{L}_k^μ , where $k > 1$, is EXP-hard.*

2.3. Background on Fixed Points. In this subsection, we mention useful facts about fixed-point formulas.

We fix a model \mathcal{M} , a formula $\mu X.\varphi$, and an environment ρ . Let $S_X^0 = \emptyset$; for every successor ordinal, let $S_X^{\zeta+1} = \llbracket \varphi, \rho[X \mapsto S_X^\zeta] \rrbracket_{\mathcal{M}}$, and for every limit ordinal ζ , let $S_X^\zeta = \bigcup_{\eta < \zeta} S_X^\eta$.

Lemma 2.12 (Iterative Characterization of the Least Fixed Point). *For every model \mathcal{M} , environment ρ , and $\mu X.\varphi \in \mathbf{L}$, there exists some ordinal ξ such that $\llbracket \mu X.\varphi, \rho \rrbracket_{\mathcal{M}} = \llbracket \varphi, \rho[X \mapsto S_X^\xi] \rrbracket_{\mathcal{M}} = S_X^\xi$.*

Proof. First we observe that the map $S \mapsto \llbracket \varphi, \rho[X \mapsto S] \rrbracket_{\mathcal{M}}$ is monotone, and therefore by the Knaster-Tarski Theorem [Tar55, Kna28], it has a least fixed-point, which is $\llbracket \mu X.\varphi, \rho \rrbracket_{\mathcal{M}} = \bigcap \{S \mid S \supseteq \llbracket \varphi, \rho[X \mapsto S] \rrbracket_{\mathcal{M}}\}$. Moreover, if $\xi \leq \zeta$, then S_X^ξ is included in S_X^ζ , which means that the approximation forms a non-decreasing chain w.r.t. set inclusion. By transfinite induction, for every ordinal ζ , $S_X^\zeta \subseteq \llbracket \mu X.\varphi, \rho \rrbracket_{\mathcal{M}}$, and by Hartogs' Theorem on well-ordered sets [Har15], there is an ordinal ξ such that $S_X^\xi = \llbracket \varphi, \rho[X \mapsto S_X^\xi] \rrbracket_{\mathcal{M}}$. As $\llbracket \mu X.\varphi, \rho \rrbracket_{\mathcal{M}}$ is a least fixed point, the lemma follows. \square

Let $\mathcal{M} = (W, R, V)$ be a \mathbf{L} -model and consider a subformula-closed set $\Phi = \text{sub}(\varphi)$ for some $\varphi \in L$, such that for every $X \in \Phi$, there exists a unique fixed-point formula $\text{fx}(X) \in \Phi$ that binds X . Let the environment ρ be such that $\rho(X) = \llbracket \text{fx}(X), \rho \rrbracket_{\mathcal{M}}$ for every variable X . We say that $\rightarrow \subseteq (W \times L)^2$ is a dependency relation on \mathcal{M} and φ when it satisfies the following conditions:

- if $u \models_\rho \psi_1 \wedge \psi_2$, then $(u, \psi_1 \wedge \psi_2) \rightarrow (u, \psi_1)$ and $(u, \psi_1 \wedge \psi_2) \rightarrow (u, \psi_2)$;
- if $u \models_\rho \psi_1 \vee \psi_2$, then $(u, \psi_1 \vee \psi_2) \rightarrow (u, \psi_1)$ and $u \models_\rho \psi_1$ or $(u, \psi_1 \vee \psi_2) \rightarrow (u, \psi_2)$ and $u \models_\rho \psi_2$;
- if $u \models_\rho [\alpha]\psi$, then $(u, [\alpha]\psi) \rightarrow (v, \psi)$ for all $v \in W$, such that $uR_\alpha v$;
- if $u \models_\rho \langle \alpha \rangle \psi$, then $(u, \langle \alpha \rangle \psi) \rightarrow (v, \psi)$ for some $v \in W$, such that $uR_\alpha v \models \psi$;
- if $u \models_\rho \mu X.\psi$ or $u \models_\rho \nu X.\psi$, then $(u, \text{fx}(X)) \rightarrow (u, \psi)$; and

⁴Similarly to Remark 2.7, [HM92] does not explicitly cover all these cases, but the techniques can be adjusted.

- if $u \models_\rho X$, then $(u, X) \rightarrow (u, \text{fx}(X))$.

For each variable X and dependency relation \rightarrow , we also define \xrightarrow{X} , such that $(w_1, \psi_1) \xrightarrow{X} (w_2, \psi_2)$ whenever $(w_1, \psi_1) \rightarrow (w_2, \psi_2)$ and $\psi_1 \neq Y$ for all variables Y where $\text{fx}(Y)$ is not a subformula of $\text{fx}(X)$. We call a dependency relation \rightarrow lfp-finite, if every least-fixed-point variable X appears finitely many times on every \xrightarrow{X} -sequence.

Theorem 2.13. *Let $\mathcal{M} = (W, R, V)$ be a \mathbf{L} -model, $w \in W$, and $\varphi \in L$, such that $\mathcal{M}, w \models \varphi$. Then, there exists an lfp-finite dependency relation on \mathcal{M} and φ .*

Proof. The claim amounts to the memoryless determinacy of parity games and our proof is inspired by [Zie98]. We prove the existence of an lfp-finite dependency relation on \mathcal{M} and φ by induction on φ . To do so, and to handle the case of free recursion variables, we can extend the definition of a dependency relation to formulas with free variables, such that the condition that if $u \models_\rho X$, then $(u, X) \rightarrow (u, \text{fx}(X))$, is omitted when X is free in φ . For the remainder of this proof, we maintain that if (u, ψ) appears in a pair in a dependency relation, then $u \models_\rho \psi$.

All cases for φ are straightforward, except for the fixed points. If $\varphi = \nu X.\psi$, then we can construct a lfp-finite dependency relation \rightarrow on \mathcal{M} and ψ , and then add the pairs (φ, ψ) and (X, φ) in \rightarrow . If Y is a least-fixed-point variable, then any new \rightarrow -path where Y appears infinitely often would go through X infinitely often, and therefore \rightarrow is still lfp-finite. If $\varphi = \mu X.\psi$, then by Lemma 2.12, there exists some ordinal ξ , with $\llbracket \varphi, \rho \rrbracket = \llbracket \psi, \rho[X \mapsto S_X^\xi] \rrbracket$. For each $\zeta \leq \xi$, we define \rightarrow_ζ and prove that it is an lfp-finite dependency relation on \mathcal{M} and ψ for $\rho[X \mapsto S_X^\zeta]$, where X does not appear infinitely often on any \xrightarrow{X}_ζ path:

Case $\zeta = 0$: for each subformula χ of ψ , let $V_\chi^0 = \llbracket \chi, \rho[X \mapsto \emptyset] \rrbracket$. Then, \rightarrow_0 is an lfp-finite dependency relation on \mathcal{M} and ψ , for $\rho[X \mapsto \emptyset]$ with the additional pairs $((u, \varphi), (u, \psi))$, for every $u \in V_\psi^0$.

Case $\zeta = \xi + 1$ for some ξ : then for each subformula χ of ψ , let

$$V_\chi^{\xi+1} = \llbracket \chi, \rho[X \mapsto S_X^{\xi+1}] \rrbracket \setminus \llbracket \chi, \rho[X \mapsto S_X^\xi] \rrbracket.$$

Let $\rightarrow_{\xi+1}^0$ be an lfp-finite dependency relation on \mathcal{M} and ψ , for $\rho[X \mapsto S_X^{\xi+1}]$. Then, we define

$$\rightarrow_{\xi+1} = \{((u, \chi), x) \mid (u, \chi) \rightarrow_\xi x; \text{ or } u \in V_\chi^{\xi+1} \text{ and } (u, \chi) \rightarrow_{\xi+1}^0 x\}$$

$$\cup \{((u, X), (u, \varphi)), ((v, \varphi), (v, \psi)) \mid u \in S_X^\xi, v \in V_\chi^{\xi+1}\}.$$

if ζ is a limit ordinal: we define $\rightarrow_\zeta = \bigcup_{\eta < \zeta} \rightarrow_\eta$.

It is straightforward to verify that \rightarrow_ζ is a dependency relation on \mathcal{M} and φ for ρ . Notice that if $(u, X) \rightarrow_\zeta (u, \varphi)$, then $u \in V_\psi^\eta$ for some $\eta < \zeta$. Therefore, no \xrightarrow{X}_ζ -path can have infinite occurrences of X , and \rightarrow_ζ is an lfp-finite dependency relation on \mathcal{M} and φ for ρ , thus completing the proof. \square

3. COMPLEXITY THROUGH TRANSLATIONS

We examine \mathbf{L} -satisfiability and use formula translations to reduce the satisfiability of one logic to the satisfiability of another. We investigate the properties of these translations and how they compose with each other, and we achieve complexity bounds for several logics.

In this section, a formula translation from logic \mathbf{L}_1 to logic \mathbf{L}_2 is a mapping f on formulas such that each formula φ is \mathbf{L}_1 -satisfiable if and only if $f(\varphi)$ is \mathbf{L}_2 -satisfiable. We only consider translations that can be computed in polynomial time, and therefore, our translations are polynomial-time reductions, transferring complexity bounds between logics.

According to Theorem 2.8, \mathbf{K}_k^μ -satisfiability, that is, satisfiability for the μ -calculus, is EXP-complete, and therefore for each logic \mathbf{L} , we aim to connect \mathbf{K}_k^μ and \mathbf{L} via a sequence of translations in either direction, to prove complexity bounds for \mathbf{L} -satisfiability.

3.1. Translating Towards \mathbf{K}_k . We begin by presenting translations from logics with more frame conditions to logics with fewer ones. To this end, we study how taking the closure of a frame under one condition affects other frame conditions.

3.1.1. Composing Frame Conditions. We now discuss how the conditions for frames affect each other. For example, to construct a transitive frame, one can take the transitive closure of a possibly non-transitive frame. The resulting frame will satisfy condition 4. As we will see, taking the closure of a frame under condition x may affect whether that frame maintains condition y , depending on x and y . In the following, we observe that one can apply the frame closures in certain orders that preserve the properties one acquires with each application. We begin by formally defining the closure of a frame with respect to some frame condition.

Definition 3.1. Let $F = (W, R)$ be a frame, and x a frame condition among $D, T, B, 4$, and 5 . The closure of R_α under x is denoted as R_α^x and defined by:

- $R_\alpha \cup \{(u, u) \mid u \in W \text{ and } \forall v \in W. (u, v) \notin R_\alpha\}$, if $x = D$.
- $R_\alpha \cup \{(u, u) \mid u \in W\}$, if $x = T$.
- $R_\alpha \cup \{(u, v) \mid (v, u) \in R_\alpha\}$, if $x = B$.
- R_α^+ (i.e., the transitive closure of R_α), if $x = 4$.
- R_α^5 is the least relation that includes R_α such that if $(s, u), (s, v) \in R_\alpha^5$, then $(u, v) \in R_\alpha^5$.

The definition of closure given above can be generalised to sets of agents thus:

Definition 3.2. Let $F = (W, R)$ be a frame, and x a frame restriction among $D, T, B, 4$, and 5 . For a set of agents $A \subseteq \text{AG}$, $R^{x,A}$ is defined by

- $R_\beta^{x,A} = R_\beta^x$, if $\beta \in A$, and
- $R_\beta^{x,A} = R_\beta$, otherwise.

We define $F^{x,A} = (W, R^{x,A})$.

We make the following observation.

Lemma 3.3. *Let x be a frame condition among $D, T, B, 4$, and 5 , and y a frame restriction among $T, B, 4$, and 5 , such that $(x, y) \notin \{(4, B), (5, T), (5, B)\}$. Then, for every frame F that satisfies x , for some agent α , $F^{y,\alpha}$ also satisfies x .*

According to Lemma 3.3, frame conditions are preserved as seen in Figure 1. In Figure 1, an arrow from x to y indicates that property x is preserved through the closure of a frame under y . Dotted red arrows indicate one-way arrows.

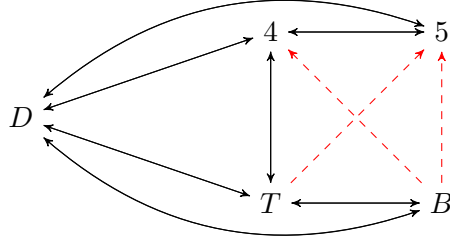


FIGURE 1. The frame-property-preservation graph

Remark 3.4. We note that, in general, not all frame conditions are preserved by the closure with respect to some other condition. For example, the accessibility relation $\{(s, t), (t, t)\}$ is euclidean, but its reflexive closure $\{(s, t), (t, t), (s, s)\}$ is not.

There is at least one linear ordering of the frame conditions $D, T, B, 4,$ and $5,$ such that all preceding conditions are preserved by closures under the conditions following them. We call such an order a closure-preserving order. We use the linear order $D, T, B, 4, 5$ in the rest of the paper.

For each logic $\mathbf{L}, \alpha \in \text{AG},$ and frame $F = (W, R),$ we write $R_\alpha^\mathbf{L}$ for the relation that results by applying the closures of the conditions of $\mathbf{L}(\alpha)$ following the above-mentioned linear order. For example, $R_\alpha^{\mathbf{B}4\alpha}$ is the relation we obtain by first applying the symmetric closure, and then the reflexive closure to the relation of agent $\alpha.$ We define $R^\mathbf{L} = (R_\alpha^\mathbf{L})_{\alpha \in \text{AG}},$ and $F^\mathbf{L} = (W, R^\mathbf{L}).$

3.1.2. Modal Logics. We start with translations that map logics without recursive operators to logics with fewer constraints. As mentioned in Subsection 2.2, all of the logics $\mathbf{L} \in \{\mathbf{K}, \mathbf{T}, \mathbf{D}, \mathbf{K4}, \mathbf{D4}, \mathbf{S4}\}$ and $\mathbf{L} + 5$ with one agent have known complexity results, and the complexity of modal logic is well-studied for multi-agent modal logics as well. The missing cases are very few and concern the combination of frame conditions (other than 5) as well as the multi-agent case. However, we take this opportunity to present an intuitive introduction to our general translation method. Each frame condition that we introduced in Section 2 is associated with an axiom for modal logic, such that whenever a model has the condition, every substitution instance of the axiom is satisfied in all states of the model (see [BvBW06, BdRV01, FHMV95]). For each frame condition x and agent $\alpha,$ we define the axiom ax_α^x as follows:

$$\begin{array}{lll} \text{ax}_\alpha^D : \langle \alpha \rangle \text{tt} & \text{ax}_\alpha^B : \langle \alpha \rangle [\alpha] p \rightarrow p & \text{ax}_\alpha^5 : \langle \alpha \rangle [\alpha] p \rightarrow [\alpha] p \\ \text{ax}_\alpha^T : [\alpha] p \rightarrow p & \text{ax}_\alpha^4 : [\alpha] p \rightarrow [\alpha] [\alpha] p & \end{array}$$

For each formula φ and $d \geq 0,$ let $\text{Inv}_d(\varphi) = \bigwedge_{i \leq d} [\text{AG}]^i \varphi,$ where $[\text{AG}]^0 \varphi = \varphi$ and $[\text{AG}]^{i+1} \varphi = [\text{AG}][\text{AG}]^i \varphi.$ Our first translations are defined using the above axioms. Intuitively, for each formula $\varphi,$ condition $x,$ and set of agents $A,$ the formula $\mathbf{F}_A^x(\varphi)$ is satisfiable over frames whose accessibility relations for the agents in A meet some set of conditions that precede x in the above-mentioned fixed linear order if, and only, if φ is satisfiable over frames where those accessibility relations also satisfy condition $x.$

Translation 3.5 (One-step Translation). Let $A \subseteq \text{AG}$ and let x be one of the frame conditions. For every formula φ , let $d = \text{md}(\varphi)$ if $x \neq 4, 5$, and $d = \text{md}(\varphi)|\varphi|$, if $x = 4$ or $x = 5$. We define:

$$\mathbf{F}_A^x(\varphi) = \varphi \wedge \text{Inv}_d \left(\bigwedge_{\substack{\psi \in \overline{\text{sub}}(\varphi) \\ \alpha \in A}} \text{ax}_\alpha^x[\psi/p] \right).$$

Theorem 3.6. *Let $A \subseteq \text{AG}$, x be one of the frame conditions, and let $\mathbf{L}_1, \mathbf{L}_2$ be logics without recursion operators, such that*

$$\mathbf{L}_1(\alpha) = \begin{cases} \mathbf{L}_2(\alpha) + x & \text{when } \alpha \in A, \text{ and} \\ \mathbf{L}_2(\alpha) & \text{otherwise.} \end{cases}$$

Assume that $\mathbf{L}_2(\alpha)$ only includes frame conditions that precede x in the fixed order of frame conditions. Then, φ is \mathbf{L}_1 -satisfiable if and only if $\mathbf{F}_A^x(\varphi)$ is \mathbf{L}_2 -satisfiable.

Before proving this theorem, we will define the *unfolding* of a model, which will be used both in its proof but also repeatedly through the paper. Our definition is a variation of the one in [GO07].

Definition 3.7 (Tree unfolding, [Sah75, GO07]). Let $\mathcal{M} = (W, R, V)$, and $w_0 \in W$. We define $\mathcal{M}' = (W', R', V')$, as follows:

- W' consists of all finite sequences $w_0\alpha_1w_1\alpha_2 \cdots \alpha_nw_n$ ($n \geq 0$), where $w_i \in W$, for every $0 \leq i \leq n$, and $w_iR_{\alpha_{i+1}}w_{i+1}$, for every $0 \leq i \leq n-1$.
- $R_\alpha = \{(u, u\alpha w) \mid (u, u\alpha w) \in W' \times W' \text{ and } w \in W\}$.
- For every propositional variable p , $w_0\alpha_1w_1\alpha_2 \cdots \alpha_nw_n \in V'(p)$ iff $w_n \in V(p)$.

The pair (\mathcal{M}', w_0) is called the *tree unfolding* of (\mathcal{M}, w_0) .

Lemma 3.8 [Sah75]. *The tree unfolding (\mathcal{M}', w) of (\mathcal{M}, w) is bisimilar to (\mathcal{M}, w) . Furthermore, if \mathcal{M} is an \mathbf{L} -model, then the closure of (\mathcal{M}', w) under the \mathbf{L} conditions is bisimilar to (\mathcal{M}, w) . Moreover, the identity mapping is a bisimulation between (\mathcal{M}', w) and its closure under the \mathbf{L} conditions.*

Proof. The first statement follows from [Sah75] – but see also [GO07]. We observe that one bisimulation in particular is the one that maps each state v of \mathcal{M} to all the paths in the unfolding that end with v . We call this bisimulation \mathcal{R} . For the second statement, simply observe that if R_α is serial in \mathcal{M} , then it is also serial in the unfolding. Moreover, any transition that is introduced to satisfy any other closure conditions preserves all bisimulations, and \mathcal{R} in particular, thus also yielding the third statement. \square

We also state the following general observation, for the case of transitive frames, which will be used in the proof of one of the cases of Theorem 3.6.

Lemma 3.9. *Let \mathcal{M} be a Kripke structure. For each $v \in W$ and $\alpha \in \text{AG}$, let $b_\alpha(v) = \{[\alpha]\psi \mid (\mathcal{M}, v) \models [\alpha]\psi\}$, and $d_\alpha(v) = \{\langle \alpha \rangle \psi \mid (\mathcal{M}, v) \models \langle \alpha \rangle \psi\}$. Assume R_α is transitive and $uR_\alpha v$. Then:*

- (1) $b_\alpha(u) \subseteq b_\alpha(v)$, and
- (2) $d_\alpha(v) \subseteq d_\alpha(u)$.

The proof of the above lemma is straightforward, using the proviso that R_α is transitive. We are now ready to proceed with the proof of Theorem 3.6. Intuitively, this theorem

claims that satisfiability over frames with condition x is reducible to satisfiability over frames without condition x —that is, if one is asking whether there exists some frame with condition x that satisfies a formula φ , it suffices to translate the formula as we have defined in the statement of that theorem, and verify the satisfiability of the resulting formula, over arbitrary frames.

The main step in the proof of Theorem 3.6 is to show that, from a model of the translation of a formula φ , we can construct a model for φ that has the property x . For each property x , we show the correctness of our model construction by proving that the constructed model satisfies all subformulas of φ and thus also φ .

Proof of Theorem 3.6. Assume first that $(\mathcal{M}, w) \models \varphi$, where \mathcal{M} is an \mathbf{L}_1 -model (and therefore also an \mathbf{L}_2 -model). For every subformula ψ of φ and $\alpha \in A$, $\text{ax}_\alpha^x[\psi/p]$ is an instantiation of the axiom ax_α^x , and therefore it holds at all states of \mathcal{M} that are reachable from w . Thus, $(\mathcal{M}, w) \models \text{Inv}_d \left(\bigwedge_{\substack{\psi \in \overline{\text{sub}}(\varphi) \\ \alpha \in A}} \text{ax}_\alpha^x[\psi/p] \right)$, which yields that $(\mathcal{M}, w) \models \text{F}_A^x(\varphi)$.

For the other direction, assume that $(\mathcal{M}, w) \models \text{F}_A^x(\varphi)$ for some \mathbf{L}_2 -model $\mathcal{M} = (W, R, V)$ and state w . We will use \mathcal{M} to construct an \mathbf{L}_1 -model for φ . For convenience and by Lemma 3.8 and Theorem 2.4, we assume we are working with the unfolding of (\mathcal{M}, w) which is defined in Definition 3.7. In particular, that implies that for every $\alpha, \beta \in \text{AG}$, if $\alpha \neq \beta$, then $R_\alpha \cap R_\beta = \emptyset$. For each $k \geq 0$, we use the notation $W_k = \{w_0\alpha_1w_1\alpha_2 \cdots \alpha_rw_r \in W \mid w_0 = w, \text{ and } 0 \leq r \leq k\}$, that is, W_k is the collection of paths from $w \in \mathcal{M}$ of length at most k . Thus, $W_i \subseteq W_j$, for each $i \leq j$. Let $d = \text{md}(\varphi)$ if $x \neq 4$ and $x \neq 5$, and $d = \text{md}(\varphi)|\varphi|$, if $x = 4$ or $x = 5$. We now distinguish the following cases:

- (1) $x \neq 4, 5$, therefore $\mathbf{L}_2(\alpha)$ does not have condition 4 for any α ,
- (2) $x = 4$, and
- (3) $x = 5$.

We tackle each of these cases separately.

First case: $x \neq 4, 5$ and $\mathbf{L}_2(\alpha)$ does not have condition 4 for any α . We define the models:

- $\mathcal{M}^c = (W, R^{\mathbf{L}_2}, V)$;
- $\mathcal{M}' = (W', R', V')$, where
 - $W' = W_d$,
 - for each $\alpha \in \text{AG}$, $R'_\alpha = (R^{\mathbf{L}_2} \cap W_d^2)^D$, if $\mathbf{L}_2(\alpha)$ has D , and $R'_\alpha = R^{\mathbf{L}_2} \cap W_d^2$, otherwise, and
 - V' is the restriction of V on W' ; and
- $\mathcal{M}^x = (W', R'^{x,A}, V')$ is the \mathbf{L}_1 -closure of \mathcal{M}' .

By Lemma 3.8, (\mathcal{M}, w) is bisimilar to (\mathcal{M}^c, w) . We also observe that \mathcal{M}' remains an \mathbf{L}_2 -model: removing states only affects condition D , and the D -closure does not affect the other conditions. Finally, \mathcal{M}^x is an \mathbf{L}_1 -model. We will now prove that \mathcal{M}^x satisfies φ . To this end, we first connect satisfaction in the structures \mathcal{M}^c and \mathcal{M}' , and then proceed to do the same for \mathcal{M}' and \mathcal{M}^x .

Claim: For every formula ψ with $\text{md}(\psi) \leq d$, and every $v \in W_{d-\text{md}(\psi)}$, $(\mathcal{M}^c, v) \models \psi$ if and only if $(\mathcal{M}', v) \models \psi$. We proceed to prove the claim by induction on ψ . All propositional cases are straightforward. For $\psi = \langle \alpha \rangle \psi'$, we have that $\text{md}(\psi) > 0$, and therefore $v \in W_{d-1}$.

By Lemma 3.8 and Theorem 2.4, $(\mathcal{M}^c, v) \models \psi$ if and only if $(\mathcal{M}, v) \models \psi$. Therefore, $(\mathcal{M}^c, v) \models \langle \alpha \rangle \psi'$ if and only if $(\mathcal{M}, v) \models \langle \alpha \rangle \psi'$ if and only if there is some $u \in W_d$, such that $vR_\alpha u$ and $(\mathcal{M}, u) \models \psi'$. We have that $(\mathcal{M}, u) \models \psi'$ if and only if $(\mathcal{M}^c, u) \models \psi'$ if and only if $(\mathcal{M}', u) \models \psi'$, by the inductive hypothesis. Therefore, if $(\mathcal{M}^c, v) \models \langle \alpha \rangle \psi'$, we have that there is some $u \in W_{d-md(\psi)+1} = W_{d-md(\psi')}$, such that $vR_\alpha u$ and $(\mathcal{M}', u) \models \psi'$, yielding that $(\mathcal{M}', v) \models \langle \alpha \rangle \psi'$. On the other hand, if $(\mathcal{M}', v) \models \langle \alpha \rangle \psi'$, then there exists some $u \in W_d$, such that $vR'_\alpha u$ and $(\mathcal{M}', u) \models \psi'$. Since $\mathbf{L}_2(\alpha)$ does not have condition 4, $u \in W_{d-md(\psi)+1} = W_{d-md(\psi')}$, and by the inductive hypothesis $(\mathcal{M}^c, u) \models \psi'$. By the construction of R' , we have that $vR_\alpha u$, and therefore $(\mathcal{M}^c, v) \models \langle \alpha \rangle \psi'$. The case of $\psi = [\alpha]\psi'$ is reduced to the case of $\langle \alpha \rangle \neg \psi'$, thus completing the proof of the claim. Specifically, we have that $(\mathcal{M}', w) \models \varphi$.

We will now conclude the proof of the first case by showing that for every subformula ψ of φ , and every $v \in W_{d-md(\psi)}$, $(\mathcal{M}', v) \models \psi$ implies that $(\mathcal{M}^x, v) \models \psi$. From this, it follows that \mathcal{M}^x is an \mathbf{L}_1 -model for φ , and we are done.

We proceed by considering the closure conditions in turn, in each case proving the above claim by induction on the structure of ψ . Observe that all propositional cases for ψ are straightforward. Furthermore, for every $\alpha \in \text{AG}$, $R'_\alpha \subseteq R_\alpha^{x,A}$, and therefore all diamond formulas are preserved from \mathcal{M}' to \mathcal{M}^x . Therefore, it suffices to consider the case for $\psi = [\alpha]\psi'$. If $\alpha \notin A$, then $R'_\alpha = R_\alpha^{x,A}$, and therefore $(\mathcal{M}', v) \models [\alpha]\psi'$ implies that $(\mathcal{M}^x, v) \models [\alpha]\psi'$ by the inductive hypothesis. To continue, we assume that $\psi = [\alpha]\psi'$, where $\alpha \in A$.

We observe that $md(\psi) > 0$, and therefore $v \in W_{d-1}$. Specifically, if $md(\psi) = e$, then $v \in W_{d-e}$. From $(\mathcal{M}, w) \models \mathbf{F}_A^x(\varphi)$, we have that $(\mathcal{M}, v) \models \text{ax}_\alpha^x[\psi'/p]$. By the inductive hypothesis, it suffices to prove that if there is some $vR_\alpha^{x,A}u$, but not $vR'_\alpha u$, then $\mathcal{M}', u \models \psi'$. We take cases for x :

$x = D$: We observe that from $(\mathcal{M}, v) \models \text{ax}_\alpha^x[\psi'/p]$ we get $(\mathcal{M}, v) \models \langle \alpha \rangle \mathbf{tt}$, which with $md(\langle \alpha \rangle \mathbf{tt}) = 1$, $v \in W_{d-1}$, and the claim above, yield that $(\mathcal{M}', v) \models \langle \alpha \rangle \mathbf{tt}$, implying that $vR'_\alpha u$ for some $u \in W'$. Therefore, from Definition 3.1, $vR_\alpha^{x,A}u$ implies $vR'_\alpha u$, contradicting our assumptions.

$x = T$: By Definition 3.1, if $vR_\alpha^{x,A}u$, but not $vR'_\alpha u$, then $u = v$. We observe that from $(\mathcal{M}, v) \models \text{ax}_\alpha^x[\psi'/p]$ we get $(\mathcal{M}, v) \models [\alpha]\psi' \rightarrow \psi'$. Furthermore, $md([\alpha]\psi' \rightarrow \psi') = md([\alpha]\psi') = e$. Therefore, $v \in W_{d-e}$ and the claim above yields that $(\mathcal{M}', v) \models [\alpha]\psi' \rightarrow \psi'$. Therefore, $(\mathcal{M}', v) \models \psi$ yields $(\mathcal{M}', v) \models \psi'$ and this case is complete.

$x = B$: By Definition 3.1, if $vR_\alpha^{x,A}u$, but not $vR'_\alpha u$, then $uR'_\alpha v$ and $u \in W_{d-e-1}$. We observe that from $(\mathcal{M}, u) \models \text{ax}_\alpha^x[\psi'/p]$ we get $(\mathcal{M}, u) \models \langle \alpha \rangle [\alpha]\psi' \rightarrow \psi'$. Furthermore, $md(\langle \alpha \rangle [\alpha]\psi' \rightarrow \psi') = md([\alpha]\psi') + 1 = e + 1$. The claim above yields that $(\mathcal{M}', u) \models \langle \alpha \rangle [\alpha]\psi' \rightarrow \psi'$. From $uR'_\alpha v$ we get that $(\mathcal{M}', u) \models \langle \alpha \rangle \psi$, which then yields together with $(\mathcal{M}', u) \models \langle \alpha \rangle [\alpha]\psi' \rightarrow \psi'$ that $(\mathcal{M}', u) \models \psi'$ and this case is complete.

We now consider the case for $x = 4$. For this case, we construct the models \mathcal{M}' and \mathcal{M}^x in a different way. Intuitively, we want to ensure that \mathcal{M}' does not prematurely cut off any long paths in the model that are relevant to the evaluation of the formula. Therefore, in its construction, we want to detect when certain states behave similarly and we (carefully) eliminate certain branches of the unfolding \mathcal{M}^c .

We first observe a statement that is analogous to Lemma 3.9. For each $v \in W_k$ and $\alpha \in A$, let

$$b_\alpha^\varphi(v) = \{[\alpha]\psi \in \overline{\text{sub}}(\varphi) \mid (\mathcal{M}^c, v) \models [\alpha]\psi\},$$

and

$$d_\alpha^\varphi(v) = \{\langle \alpha \rangle \psi \in \overline{\text{sub}}(\varphi) \mid (\mathcal{M}^c, v) \models \langle \alpha \rangle \psi\}.$$

Assume that $v \in W_k$, where $k \leq d$ and $uR_\alpha v$. Then, $(\mathcal{M}^c, v) \models \bigwedge_{\psi \in \overline{\text{sub}}(\varphi)} [\alpha]\psi \rightarrow [\alpha][\alpha]\psi$ and therefore:

- (1) $b_\alpha^\varphi(u) \subseteq b_\alpha^\varphi(v)$, and
- (2) $d_\alpha^\varphi(v) \subseteq d_\alpha^\varphi(u)$.

If, additionally, $\mathbf{L}_2(\alpha)$ has constraint B , then $b_\alpha^\varphi(u) = b_\alpha^\varphi(v)$ and $d_\alpha^\varphi(v) = d_\alpha^\varphi(u)$.

We call a state $sv \in W_{k+1}$ α -stable, for some $\alpha \in A$, when $sR_\alpha^c sv \in W_{k+1}$, and $b_\alpha^\varphi(s) = b_\alpha^\varphi(sv)$ and $d_\alpha^\varphi(s) = d_\alpha^\varphi(sv)$, and write $s \triangleright_\alpha sv$. Observe that if $uR_\alpha v$ and v is not α -stable, then either $b_\alpha^\varphi(v)$ is a strict superset of $b_\alpha^\varphi(u)$ or $d_\alpha^\varphi(v)$ is a strict subset of $d_\alpha^\varphi(u)$. Therefore, by the Pigeonhole Principle, for $\alpha \in A$, in any sequence $sR_\alpha sv_1 R_\alpha \cdots R_\alpha sv_1 \cdots v_l$, where each $sv_1 \cdots v_i \in W_i$ and $l \geq |\varphi|$, there must be an α -stable state.

Recall that we assume that $(\mathcal{M}, w) \models F_A^x(\varphi)$ for some \mathbf{L}_2 -model $\mathcal{M} = (W, R, V)$. Let $\mathcal{M}' = (W', R', V')$, where

- for each $\alpha \notin A$, $R'_\alpha = R_\alpha^c$, and for each $\alpha \in A$, $(a, b) \in R'_\alpha$ if one of the following holds:
 - (1) $aR_\alpha^c b$ and $\nexists c \triangleright_\alpha a$,
 - (2) $\exists c \triangleright_\alpha a$, such that $cR_\alpha^c b$,
 - (3) $b \triangleright_\alpha a$ and $\mathbf{L}_2(\alpha)$ has B , or
 - (4) $\exists c \triangleright_\alpha b$ such that $cR_\alpha^c a$ and $\mathbf{L}_2(\alpha)$ has B ;
- $W' = \{v \in W \mid v \text{ is reachable from } w \text{ by } R'\}$; and
- V' is the restriction of V on W' .

Observe that if $a \triangleright_\alpha bR_\alpha^c c$, then c is not reachable from w by R' . Therefore, for every $\alpha \in A$, there is no sequence $pR'_\alpha pv_1 R'_\alpha \cdots R'_\alpha pv_1 \cdots v_l$ in \mathcal{M}' , where each $pv_1 \cdots v_i \in W_{k_i}$, $k_1 < \cdots < k_l$, and $l > |\varphi|$. Moreover, if $\mathbf{L}_2(\alpha)$ has B , then $aR_\alpha^c b$ implies $a \triangleright_\alpha b$, and therefore item (4) is redundant.

\mathcal{M}' is now a \mathbf{L}_2 -model: it is not hard to see that conditions D and T are not affected from R^c to R' , and the last two items in the construction of R'_α above ensure that when $\mathbf{L}_s(\alpha)$ has condition B , then each new pair in R'_α also introduces its symmetric one.

We prove, by lexicographic induction on $(|\psi|, k)$, that for every formula $\psi \in \text{sub}(\varphi)$ and every $v \in W_k \cap W'$, where $k \leq d - md(\psi)|\varphi|$, $(\mathcal{M}^c, v) \models \psi$ if and only if $(\mathcal{M}', v) \models \psi$. The propositional cases are straightforward. The modal cases $\psi = \langle \alpha \rangle \psi'$ or $\psi = [\alpha]\psi'$ ensure that $md(\psi) > 0$, and therefore $v \in W_{d-1}$, and there is some $(v, u) \in R'_\alpha$. If $\alpha \notin A$, then this case follows from the inductive hypothesis and the observation that the accessible states from v in \mathcal{M}^c and in \mathcal{M}' are the same. Similarly, if $\alpha \in A$ and v is not α -stable, then $\mathbf{L}_2(\alpha)$ does not have constraint B and the accessible states from v in \mathcal{M}^c and in \mathcal{M}^s are the same.

If $\alpha \in A$ and $v' \triangleleft_\alpha v$ for some $v' \in W$, then v' is not α -stable, because otherwise $v \notin W'$, as it would not be reachable from w by R' in \mathcal{M}^c . From $v' \triangleleft_\alpha v$, we get that $(\mathcal{M}^c, v') \models \psi$ if and only if $(\mathcal{M}^c, v) \models \psi$, and $v' \in W_{k-1}$. The inductive hypothesis for $(|\psi|, k-1)$ then yields that $(\mathcal{M}', v') \models \psi$ if and only if $(\mathcal{M}^c, v') \models \psi$. Let $\psi = \langle \alpha \rangle \psi'$. From the above, $(\mathcal{M}^c, v) \models \langle \alpha \rangle \psi'$ if and only if $(\mathcal{M}^c, v') \models \langle \alpha \rangle \psi'$ if and only if $(\mathcal{M}', v') \models \langle \alpha \rangle \psi'$. But if $(\mathcal{M}', v') \models \langle \alpha \rangle \psi'$, then there exists some $v'R'_\alpha u$, such that $(\mathcal{M}', u) \models \psi'$. But by construction we also have that $vR'_\alpha u$, and therefore $(\mathcal{M}', v) \models \langle \alpha \rangle \psi'$. On the other hand, if

$(\mathcal{M}', v) \models \langle \alpha \rangle \psi'$, then there exists some $vR'_\alpha u$, such that $(\mathcal{M}', u) \models \psi'$. By construction we also have the following two cases:

- $\mathbf{L}_2(\alpha)$ has constraint B and $u = v'$, and therefore $(\mathcal{M}', v') \models \psi'$, yielding, by the induction hypothesis that $(\mathcal{M}^c, v') \models \psi'$. Therefore $(\mathcal{M}^c, v) \models \langle \alpha \rangle \psi'$.
- $v'R'_\alpha u$, and therefore $(\mathcal{M}', v') \models \langle \alpha \rangle \psi'$, and following the biimplications above, we also have $(\mathcal{M}^c, v) \models \langle \alpha \rangle \psi'$.

Let $\mathcal{M}^x = (W', R'^{x,A}, V')$. We now prove that for every subformula ψ of φ , and every $v \in W_{d-md(\psi)|\varphi|}$, $(\mathcal{M}', v) \models \psi$ if and only if $(\mathcal{M}^x, v) \models \psi$. As usual, the propositional cases are straightforward. If $\psi = \langle \alpha \rangle \psi'$ and $(\mathcal{M}', v) \models \psi$, then $(\mathcal{M}^x, v) \models \psi$, because diamond formulas are preserved by the introduction of pairs in the accessibility relation. If $\psi = [\alpha] \psi'$, $(\mathcal{M}', v) \models \psi$, and $\alpha \notin A$, then for every $(v, u) \in R'_\alpha$, $(\mathcal{M}', u) \models \psi'$ and $u \in W_{k+1}$. By the inductive hypothesis, since $vR'_\alpha u$ iff $vR'^{x,A}_\alpha u$, for every $(v, u) \in R'^{x,A}_\alpha$, $(\mathcal{M}^x, u) \models \psi'$, yielding $(\mathcal{M}^x, v) \models \psi$. If $\psi = [\alpha] \psi'$, $(\mathcal{M}', v) \models \psi$, and $\alpha \in A$, then by the induction hypothesis it suffices to prove that $(\mathcal{M}', u) \models \psi'$ for every R'_α -reachable state u from v . Let $vR'_\alpha v_1 R'_\alpha \dots R'_\alpha v_l = u$. We know that $v \in W_{d-md(\psi)|\varphi|}$ and that $v_i \in W_{d-md(\psi)|\varphi|+|\varphi|} = W_{d-(md(\psi)-1)|\varphi|}$. From $(\mathcal{M}, w) \models \mathbf{F}_A^x(\varphi)$, we have that $(\mathcal{M}, v_i) \models \mathbf{ax}_\alpha^x[\psi'/p]$, yielding $(\mathcal{M}^c, v_i) \models \mathbf{ax}_\alpha^x[\psi'/p]$. Therefore, $(\mathcal{M}^c, v_i) \models [\alpha] \psi' \rightarrow [\alpha][\alpha] \psi'$, which gives $(\mathcal{M}^c, v_i) \models \psi'$ by straightforward induction on i . Therefore, $(\mathcal{M}^c, u) \models \psi'$, yielding $(\mathcal{M}', u) \models \psi'$, as $u \in W_{d-(md(\psi)-1)|\varphi|} = W_{d-md(\psi')|\varphi|}$. This completes the case for $\psi = [\alpha] \psi'$. The other direction of the biimplication is dual, and the case for $x = 4$ is complete.

We now consider the case for $x = 5$. We construct \mathcal{M}^c and \mathcal{M}^x as in the above cases, but to construct model \mathcal{M}' we need to take into account the agents with constraint 4 or 5 in A , in a similar way as it was done for the case for $x = 4$. Similarly to the case for $x = 4$, we define $b_\alpha^\varphi(v)$ and $d_\alpha^\varphi(v)$. Assume that $uR_\alpha v$. Then, Lemma 3.9 tells us that if $\mathbf{L}_2(\varphi)$ has constraint 4, then $b_\alpha^\varphi(u) \subseteq b_\alpha^\varphi(v)$, and $d_\alpha^\varphi(v) \subseteq d_\alpha^\varphi(u)$. If, additionally, $\mathbf{L}_2(\alpha)$ has constraint B , then $b_\alpha(u) = b_\alpha(v)$ and $d_\alpha(v) = d_\alpha(u)$.

Observe that for every $\chi \in \overline{\mathbf{sub}}(\varphi)$, $\alpha \in A$ and $v_1, v_2 \in W_d$, where $v_1 R_\alpha^c v_2$, $(\mathcal{M}, w) \models \mathbf{F}_A^x(\varphi)$ yields that $(\mathcal{M}^c, v_1) \models \langle \alpha \rangle \chi$ implies $(\mathcal{M}^c, v_2) \models \langle \alpha \rangle \chi$, and $(\mathcal{M}^c, v_2) \models [\alpha] \chi$ implies $(\mathcal{M}^c, v_1) \models [\alpha] \chi$. Therefore, if $\alpha \in A$, $uR_\alpha v$, and $u, v \in W_d$, then $b_\alpha^\varphi(v) \subseteq b_\alpha^\varphi(u)$, and $d_\alpha^\varphi(u) \subseteq d_\alpha^\varphi(v)$. If, additionally, $\mathbf{L}_2(\alpha)$ has constraint B or 4, then $b_\alpha^\varphi(v) = b_\alpha^\varphi(u)$, and $d_\alpha^\varphi(u) = d_\alpha^\varphi(v)$.

We remind to the reader that, as in the case $x = 4$, every sufficiently long sequence of states has an α -stable state. Let $\mathcal{M}' = (W', R', V')$, where

- $W' = \{v \in W_d \mid v \text{ has no stable state as a strict prefix}\}$;
- for each $\alpha \in \text{AG}$, if $\mathbf{L}_2(\alpha)$ does not have constraint 4 and $\alpha \notin A$, then $R'_\alpha = R_\alpha^c$,
- if $\mathbf{L}_2(\alpha)$ has constraint 4 or $\alpha \in A$, then $(a, b) \in R'_\alpha$ if $a, b \in W'$ and one of the following holds:
 - (1) $aR_\alpha^c b$ and $\nexists c \triangleright_\alpha a$,
 - (2) $\exists c \triangleright_\alpha a$, such that $cR_\alpha^c b$,
 - (3) $b \triangleright_\alpha a$ and $\mathbf{L}_2(\alpha)$ has B , or
 - (4) $\exists c \triangleright_\alpha b$ such that $cR_\alpha^c a$ and $\mathbf{L}_2(\alpha)$ has B ; and
- V' is the restriction of V on W' .

Observe that if $a \triangleright_\alpha bR_\alpha^c c$, then c is not reachable from w by R' . Similarly to the case of $x = 4$, if $\mathbf{L}(\alpha)$ has constraint 4 or $\alpha \in A$, then there is no sequence $pR'_\alpha pv_1 R'_\alpha \dots R'_\alpha pv_1 \dots v_l$

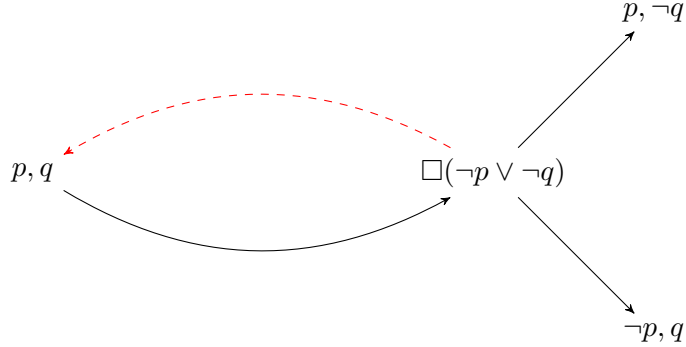


FIGURE 2. An example where the use of the alternative axiom for symmetricity would yield an incorrect translation.

in \mathcal{M}' , where each $pv_1 \cdots v_i \in W_{k_i}$, $k_1 < \cdots < k_l$, and $l > |\varphi|$. Moreover, if $\mathbf{L}_2(\alpha)$ has B , then $aR_\alpha^c b$ implies $a \triangleright_\alpha b$, and therefore item (4) is redundant.

\mathcal{M}' is now a \mathbf{L}_2 -model: it is not hard to see that conditions D , T , and 4 are not affected from R^c to R' , and the last two items in the construction of R'_α above ensure that when $\mathbf{L}_s(\alpha)$ has condition B , then each new pair in R'_α also introduces its symmetric one. We can prove that for every $v \in W_k$ and $\psi \in \text{sub}(\varphi)$, if $k \leq d - md(\psi)|\varphi|$, then $(\mathcal{M}^c, v) \models \psi$ if and only if $(\mathcal{M}', v) \models \psi$ with similar arguments as for the case of $x = 4$, using the observation that if $\mathbf{L}(\alpha)$ has constraint 4 or $\alpha \in A$, then there is no sequence $pR'_\alpha pv_1 R'_\alpha \cdots R'_\alpha pv_1 \cdots v_l$ in \mathcal{M}' , where each $pv_1 \cdots v_i \in W_{k_i}$, $k_1 < \cdots < k_l$, and $l > |\varphi|$.

What remains is to prove that for every subformula ψ of φ , and every $v \in W_{d-md(\psi)|\varphi|}$, $(\mathcal{M}', v) \models \psi$ if and only if $(\mathcal{M}^x, v) \models \psi$. As usual, we use induction on ψ , and the propositional cases are straightforward. If $\psi = \langle \alpha \rangle \psi'$ and $(\mathcal{M}', v) \models \psi$, then $(\mathcal{M}^x, v) \models \psi$, because diamond formulas are preserved by the introduction of pairs in the accessibility relation. If $\psi = [\alpha] \psi'$, $(\mathcal{M}', v) \models \psi$, and $\alpha \notin A$, then for every $(v, u) \in R'_\alpha$, $(\mathcal{M}', u) \models \psi'$ and $u \in W_{k+1}$. By the inductive hypothesis, since $vR'_\alpha u$ iff $vR_\alpha^{x,A} u$, for every $(v, u) \in R_\alpha^{x,A}$, $(\mathcal{M}^x, u) \models \psi'$, yielding $(\mathcal{M}^x, v) \models \psi$.

We now prove that $\psi = [\alpha] \psi'$, $\alpha \in A$, and $(\mathcal{M}', v) \models \psi$ implies $(\mathcal{M}^x, v) \models \psi$, for every $v \in W_{d-md(\psi)}$. To this end, assume that $(\mathcal{M}', v) \models \psi$ and u is a state such that $vR_\alpha^x u$, but $vR'_\alpha u$ does not hold. It suffices to prove that $(\mathcal{M}^c, u) \models \psi'$. From our assumption that $vR_\alpha^x u$, but not $vR'_\alpha u$, and by the euclidean closure condition, we can see that there are some $a, b, c \in W'$, such that $aR'_\alpha b, c$, and v is R'_α -reachable from b , and u is R'_α -reachable from c .

Assume that $(\mathcal{M}^c, u) \not\models \psi'$, to reach a contradiction. We can see that there is some u' that is R'_α -reachable from a and $u'R'_\alpha u$, such that $(\mathcal{M}^c, u') \models \langle \alpha \rangle \neg \psi'$; from the above observation, this yields that $(\mathcal{M}^c, a) \models \langle \alpha \rangle \neg \psi'$. But, using the same observation, $(\mathcal{M}', v) \models [\alpha] \psi'$ yields $(\mathcal{M}', a) \models [\alpha] \psi'$, which is a contradiction. Thus, we conclude that $(\mathcal{M}^c, u) \models \psi'$, which is what we wanted to prove.

The other direction of the biimplication is dual, and the proof is complete. \square

Remark 3.10. The attentive reader might have noted that we used the contra-positive version of the axioms for symmetric and euclidean conditions on a frame. Indeed, it turns out that our translation would not be correct if we adopted the usual axioms, namely $p \rightarrow \square \diamond p$ for

symmetry and $\diamond p \rightarrow \square \diamond p$ for euclidicity. To see this, let us assume that there is only one agent and consider the version of a translation defined based on the usual version of the axiom for symmetry, which would map a formula φ to

$$\varphi \wedge \text{Inv}_d \left(\bigwedge_{\psi \in \overline{\text{sub}}(\varphi)} \psi \rightarrow \square \diamond \psi \right).$$

It is possible in a non-symmetric frame with states sRt to have $t \models \square \psi$ but $s \not\models \psi$ for some subformula $\square \psi$ of φ . In this case our construction of a symmetric model would not work, as adding symmetric edge of sRt would not preserve the satisfiability of $\square \psi$ at t . For example, Figure 2 shows a model where adding edges for the symmetric closure would still satisfy the translation, but it would not preserve the truth of the formula $\square(\neg p \vee \neg q)$. It is then hard to see how such a model would describe a symmetric one, or how that version of the axiom could be used to “force” a model of the translation to be easily turned into a symmetric one. See Figure 2 for such an example.

Theorem 3.6 together with Ladner’s classic result for the complexity of the satisfiability of \mathbf{K} [Lad77] yield an alternative proof of the following statement.

Corollary 3.11. *The satisfiability problem for $\mathbf{L} + x$, with $x \in \mathbf{T}, \mathbf{D}, \mathbf{B}, \mathbf{K4}, \mathbf{K5}$, without fixed-point operators is in PSPACE.*

3.1.3. Modal Logics with Recursion. In the remainder of this section we will modify our translations and proof technique, in order to lift our results to logics with fixed-point operators. It is not clear whether the translations of Subsection 3.1.2 can be extended straightforwardly in the case of logics with recursion, by using unbounded invariance Inv , instead of the bounded Inv_d . To see this, consider the following example.

Example 3.12. Consider the formula $\varphi_f = \mu X. \square X$, which requires all paths in the model to be finite, and thus is satisfiable over arbitrary frames, but not over reflexive ones. In the setting of logics with fixed-point operators, formulas may have open subformulas. Therefore, in adapting our translations, it is natural to consider their closures, as defined on page 4. The set of the closures of the subformulas of φ_f is $\{\varphi_f, \square \varphi_f\}$. The reader may observe that for the case of reflexive frames, we do not need to conjunct over the negations of the subformulas of φ . Without those the translation would give

$$\varphi_t := \varphi_f \wedge \text{Inv}((\square \varphi_f \rightarrow \varphi_f) \wedge (\square \square \varphi_f \rightarrow \square \varphi_f)).$$

Indeed, on reflexive frames, the formulas $\square \varphi_f \rightarrow \varphi_f$ and $\square \square \varphi_f \rightarrow \square \varphi_f$ are valid, and therefore φ_t is equivalent to φ_f , which is \mathbf{K} -satisfiable. This was not an issue in Subsection 3.1.2, as the finiteness of all the paths in a model cannot be expressed without recursion.

One would then naturally wonder whether conjoining over $\overline{\text{sub}}(\varphi_f)$ in the translation would make a difference. The answer is affirmative, as the translation

$$\varphi_f \wedge \text{Inv} \left(\bigwedge_{\psi \in \overline{\text{sub}}(\varphi_f)} \square \psi \rightarrow \psi \right)$$

would then yield a formula that is not satisfiable over reflexive frames. However, our constructions would not work to prove that such a translation preserves satisfiability. For example, consider $\mu X. \square(p \rightarrow (r \wedge (q \rightarrow X)))$, whose translation is satisfied on a model that satisfies at the same time p and q . We invite the reader to verify the details.

The only case where the approach that we used for the logics without recursion can be applied is for the case of seriality (condition D), as $\text{Inv}(\langle \alpha \rangle \text{tt})$ directly ensures the seriality of a model.

Translation 3.13. Let $A \subseteq \text{AG}$. We define:

$$\mathbf{F}_A^{D^\mu}(\varphi) = \varphi \wedge \text{Inv}\left(\bigwedge_{\alpha \in A} \langle \alpha \rangle \text{tt}\right).$$

Theorem 3.14. Let $A \subseteq \text{AG}$ and $|\text{AG}| = k$, and let \mathbf{L} be a logic, such that $\mathbf{L}(\alpha) = \mathbf{D}$ when $\alpha \in A$, and \mathbf{K} otherwise. Then, φ is \mathbf{L} -satisfiable if and only if $\mathbf{F}_A^{D^\mu}(\varphi)$ is \mathbf{K}_k^μ -satisfiable.

For the cases of reflexivity and transitivity, our simple translations substitute the modal subformulas of a formula to implicitly enforce the corresponding condition.

Translation 3.15. Let $A \subseteq \text{AG}$. The function $\mathbf{F}_A^{T^\mu}(-)$ is defined homomorphically apart from the following two clauses when $\alpha \in A$.

- $\mathbf{F}_A^{T^\mu}([\alpha]\varphi) = [\alpha]\mathbf{F}_A^{T^\mu}(\varphi) \wedge \mathbf{F}_A^{T^\mu}(\varphi)$;
- $\mathbf{F}_A^{T^\mu}(\langle \alpha \rangle \varphi) = \langle \alpha \rangle \mathbf{F}_A^{T^\mu}(\varphi) \vee \mathbf{F}_A^{T^\mu}(\varphi)$.

Theorem 3.16. Let $\mathbf{L}_1, \mathbf{L}_2$ be logics, such that $\mathbf{L}_1(\alpha) = \mathbf{L}_2(\alpha) + T$ when $\alpha \in A$, and $\mathbf{L}_1(\alpha) = \mathbf{L}_2(\alpha)$ otherwise, and $\mathbf{L}_2(\alpha)$ at most includes frame conditions D or B . Then, φ is \mathbf{L}_1 -satisfiable if and only if $\mathbf{F}_A^{T^\mu}(\varphi)$ is \mathbf{L}_2 -satisfiable.

Proof. First we assume that $(\mathcal{M}, w) \models \varphi$, where $\mathcal{M} = (W, R, V)$ is an \mathbf{L}_1 -model and $w \in W$. We can easily verify, by induction on every subformula ψ of φ on every environment ρ and state s of W , that $(\mathcal{M}, s) \models_\rho \psi$ if and only if $(\mathcal{M}, s) \models_\rho \mathbf{F}_A^{T^\mu}(\psi)$. We then conclude that $(\mathcal{M}, w) \models \mathbf{F}_A^{T^\mu}(\varphi)$, and thus the translated formula remains satisfiable.

For the converse implication, we assume an \mathbf{L}_2 -model $\mathcal{M} = (W, R, V)$ and $w \in W$, such that $(\mathcal{M}, w) \models \mathbf{F}_A^{T^\mu}(\varphi)$. We construct an \mathbf{L}_1 -model that satisfies φ : let $\mathcal{M}_p = (W, R^p, V)$, where for each $\alpha \in A$, $R_\alpha^p = R_\alpha \cup \{(u, u) \mid u \in W\}$, the reflexive closure of R_α , and for each $\alpha \notin A$, $R_\alpha^p = R_\alpha$. By Lemma 3.3, we have that the symmetry and seriality conditions of \mathcal{M} are preserved in \mathcal{M}_p . It now suffices to prove that for every $v \in W$, environment ρ , and $\psi \in \text{sub}(\varphi)$, $(\mathcal{M}, v) \models_\rho \mathbf{F}_A^{T^\mu}(\psi)$ implies $(\mathcal{M}_p, v) \models_\rho \psi$. We proceed by induction on ψ . The cases of propositional and recursion variables and boolean connectives are straightforward. The cases of fixed-points are also not hard by using the inductive hypothesis. Finally, we treat the modal cases. If $\psi = \langle \alpha \rangle \psi'$, since we only introduce new pairs to the accessibility relation, it is straightforward to see that $(\mathcal{M}_p, v) \models_\rho \psi$. If $\psi = [\alpha] \psi'$ and $(\mathcal{M}, v) \models_\rho \mathbf{F}_A^{T^\mu}(\psi)$, then $(\mathcal{M}, v) \models_\rho [\alpha] \mathbf{F}_A^{T^\mu}(\psi') \wedge \mathbf{F}_A^{T^\mu}(\psi')$, and therefore $(\mathcal{M}, v) \models_\rho [\alpha] \mathbf{F}_A^{T^\mu}(\psi')$ and $(\mathcal{M}, v) \models_\rho \mathbf{F}_A^{T^\mu}(\psi')$. By the definition of R^p , if $v R_\alpha^p u$ for some $u \in W$, then $u = v$ or $v R_\alpha u$. From $(\mathcal{M}, v) \models_\rho [\alpha] \mathbf{F}_A^{T^\mu}(\psi')$ and $(\mathcal{M}, v) \models_\rho \mathbf{F}_A^{T^\mu}(\psi')$ we then get that $(\mathcal{M}, u) \models_\rho \mathbf{F}_A^{T^\mu}(\psi')$ for both cases, and the inductive hypothesis yields that $(\mathcal{M}_p, u) \models_\rho \mathbf{F}_A^{T^\mu}(\psi')$, completing the proof. \square

For the case of transitivity, we use a restricted form of the invariance operator: for each $\varphi \in L$ and $\alpha \in \text{AG}$, $\text{Inv}^\alpha(\varphi) = \nu X.(\varphi \wedge [\alpha]X)$ and $\text{Eve}^\alpha(\varphi) = \mu X.(\varphi \vee \langle \alpha \rangle X)$.

Translation 3.17. Let $A \subseteq \text{AG}$. The function $\mathbf{F}_A^{4^\mu}(-)$ is defined to be such that

- $\mathbf{F}_A^{4^\mu}([\alpha]\psi) = \text{Inv}^\alpha([\alpha]\mathbf{F}_A^{4^\mu}(\psi))$ for each $\alpha \in A$
- $\mathbf{F}_A^{4^\mu}(\langle \alpha \rangle \psi) = \text{Eve}^\alpha(\langle \alpha \rangle \mathbf{F}_A^{4^\mu}(\psi))$ for each $\alpha \in A$, and
- $\mathbf{F}_A^{4^\mu}(-)$ commutes with all other operations.

Theorem 3.18. *Let $A \subseteq \text{AG}$, $\mathbf{L}_1, \mathbf{L}_2$ be logics, such that $\mathbf{L}_1(\alpha) = \mathbf{L}_2(\alpha) + 4$ when $\alpha \in A$, and $\mathbf{L}_2(\alpha)$ otherwise, and $\mathbf{L}_2(\alpha)$ at most includes frame conditions D, T, B . Then, φ is \mathbf{L}_1 -satisfiable if and only if $\mathbf{F}_A^{4\mu}(\varphi)$ is \mathbf{L}_2 -satisfiable.*

Proof. If $\mathbf{F}_A^{4\mu}(\varphi)$ is satisfied in a model $\mathcal{M} = (W, R, V)$, let $\mathcal{M}' = (W, R^+, V)$, where R_α^+ is the transitive closure of R_α , if $\alpha \in A$, and $R_\alpha^+ = R_\alpha$, otherwise. It is now not hard to use induction on ψ to show that for every (possibly open) subformula ψ of φ and for every environment ρ , $\llbracket \mathbf{F}_A^{4\mu}(\psi), \rho \rrbracket_{\mathcal{M}} = \llbracket \psi, \rho \rrbracket_{\mathcal{M}'}$. The other direction is more straightforward, similarly demonstrating that if φ is satisfied in an \mathbf{L}_1 -model $\mathcal{M} = (W, R, V)$ (where R_α is already transitive for $\alpha \in A$), then for every (possibly open) subformula ψ of φ and for every environment ρ , $\llbracket \mathbf{F}_A^{4\mu}(\psi), \rho \rrbracket_{\mathcal{M}} = \llbracket \psi, \rho \rrbracket_{\mathcal{M}}$. \square

As Example 3.12 hints, in order to produce a similar translation for symmetric frames, we need to use a more involved construction. Moreover, we only prove the correctness of the following translation for formulas without least-fixed-point operators.

Translation 3.19. Let $A \subseteq \text{AG}$. Let $p \notin \text{PVAR}$ be a new propositional variable. The function $\mathbf{F}_A^{\mathbf{B}\mu}(-)$ is defined as

$$\begin{aligned} \mathbf{F}_A^{\mathbf{B}\mu}(\varphi) = & \varphi \wedge \bigwedge_{\alpha \in A} \text{Inv}([\alpha](\neg p \rightarrow \langle \alpha \rangle p) \\ & \wedge \bigwedge_{\psi \in \overline{\text{sub}}(\varphi)} (\text{cl}(\psi) \rightarrow [\alpha][\alpha](p \rightarrow \text{cl}(\psi))) \\ & \wedge \bigwedge_{\langle \alpha \rangle \psi \in \overline{\text{sub}}(\varphi)} (\langle \alpha \rangle \text{cl}(\psi) \rightarrow \langle \alpha \rangle (\neg p \wedge \text{cl}(\psi))). \end{aligned}$$

The intuition for this translation is in two parts. In the second conjunct, we force each existing pair (a, b) of an agent's accessibility relation to be given a continuation (b, c) leading to a marked state c , which represents a , the state from which b is accessible. However, we are not guaranteed that this continuation leads back to the original state – i.e. that $a = c$, – and it is not enough to help us construct a truly symmetric model as we would need to do later on. This is why, in the second conjunct, we now require that all subformulas of φ will have to be carried through these transitions to be satisfied in all p -marked states. Therefore, p -marked states act as accessible clones of a preceding state. This guarantees that later in the proof we will manage to identify all p -marked states with the original one the two transitions originated in (that is, a and c in the above example), thus resulting in a symmetric model. The third conjunct can be read in pair with the first one inside the invariant operator, and it ensures its correct behavior. Diamond formulas must be satisfied in state a through a transition to a state b that is not marked with p , and therefore, due to the first conjunct, can access a state c that is marked with p .

Theorem 3.20. *Let $\mathbf{L}_1, \mathbf{L}_2$ be logics, such that $\mathbf{L}_1(\alpha) = \mathbf{L}_2(\alpha) + \mathbf{B}$ when $\alpha \in A$, and $\mathbf{L}_2(\alpha)$ otherwise, and $\mathbf{L}_2(\alpha)$ at most includes frame condition D . Then, a formula φ that has no μX operators is \mathbf{L}_1 -satisfiable if and only if $\mathbf{F}_A^{\mathbf{B}\mu}(\varphi)$ is \mathbf{L}_2 -satisfiable.*

Proof. First we assume that the μ -free formula φ is satisfied in (\mathcal{M}, w) , where $\mathcal{M} = (W, R, V)$ is an \mathbf{L}_1 -model and $w \in W$. We assume that $p \notin V(u)$ for every $u \in W$. We construct an \mathbf{L}_2 -model that satisfies $\mathbf{F}_A^{\mathbf{B}\mu}(\varphi)$. Let $\mathcal{M}_u = (W_u, R^u, V^u)$ be the unfolding of (\mathcal{M}, w) , and let $W_p = \{u_p \mid u \in W_u\}$ be a set of distinct copies of states from W_u . We define $\mathcal{M}_p = (W_p \cup W_u, R^p, V^p)$, where:

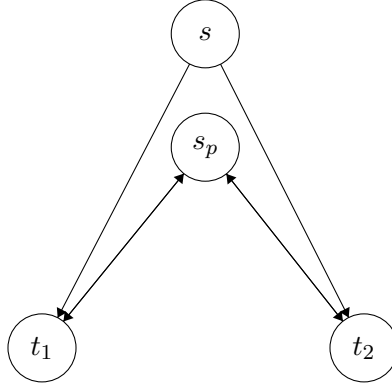


FIGURE 3. The new model, based on a state s in \mathcal{M} , with two neighbours t_1, t_2 .

- for each $u \in W_u$, $V^p(u) = V(u)$ and $V^p(u_p) = V(u) \cup \{p\}$; and
- $R_\alpha^p = R_\alpha^u \cup \{(s_p, t), (t, s_p) \mid (s, t) \in R_\alpha^u\}$, if $\alpha \in A$, and $R_\alpha^p = R_\alpha^u$ otherwise.

This construction is depicted in Figure 3.

We will now argue that $(\mathcal{M}_p, w) \models \mathbf{FB}_A^{\mathbf{B}^\mu}(\varphi)$. To this end, we first argue that $(\mathcal{M}_p, w) \models \varphi$. To see this, observe that $\mathcal{R} = \{(s, s), (s, s_p) \mid s \in W_u\}$ is a bisimulation between (\mathcal{M}_u, w) and (\mathcal{M}_p, w) , if we omit variable p from the language; Theorem 2.4 then yields that $(\mathcal{M}_p, w) \models \varphi$. More generally, let ρ, ρ' be environments on \mathcal{M}_u and \mathcal{M}_p respectively, such that $\rho(X) = \llbracket \text{fx}(X), \rho \rrbracket_{\mathcal{M}}$ and $\rho'(X) = \llbracket \text{fx}(X), \rho' \rrbracket_{\mathcal{M}_p}$, for every recursion variable X that appears in φ . Theorem 2.4 then yields that for every $s \in W_u$ and $\psi \in \text{sub}(\varphi)$, $(\mathcal{M}_u, s) \models_\rho \psi$ if and only if $(\mathcal{M}_p, s) \models_{\rho'} \psi$ if and only if $(\mathcal{M}_p, s_p) \models_{\rho'} \psi$.

We now prove that for each $\alpha \in A$,

$$(\mathcal{M}_p, w) \models \text{Inv} \left(\bigwedge_{\psi \in \overline{\text{sub}}(\varphi)} \text{cl}(\psi) \rightarrow [\alpha][\alpha](p \rightarrow \text{cl}(\psi)) \right).$$

It suffices to prove that for every $s \in W_u$ and $\psi \in \overline{\text{sub}}(\varphi)$, $(\mathcal{M}_p, s) \models_{\rho'} \psi$ implies that $(\mathcal{M}_p, s) \models_{\rho'} [\alpha][\alpha]\psi$ and $(\mathcal{M}_p, s_p) \models_{\rho'} \psi$ implies that $(\mathcal{M}_p, s) \models_{\rho'} [\alpha][\alpha]\psi$, but the second implication results from the first one. Let $(\mathcal{M}_p, s) \models_{\rho'} \psi$ and let $s_1, s_2 \in W_u \cup W_p$, such that $sR_\alpha^p s_1 R_\alpha^p s_2$ and $(\mathcal{M}_p, s_2) \models p$. It suffices to prove that $(\mathcal{M}_p, s_2) \models_{\rho'} \psi$. From $(\mathcal{M}_p, s_2) \models p$ we get that $s_2 \in W_p$, and therefore by the construction of \mathcal{M}_p , $s_1 \in W_u$. Since \mathcal{M}_u is an unfolding of a model, there is a unique $t \in W_u$, such that $tR_\alpha^p s_1$ and either $s = t$ or $s = t_p$. Therefore, $s_2 = t_p$ and $(\mathcal{M}_p, t) \models_{\rho'} \psi$, therefore $(\mathcal{M}_p, s_2) \models_{\rho'} \psi$, which is what we needed to prove.

We now show that, for each $\alpha \in A$,

$$(\mathcal{M}_p, w) \models \text{Inv}([\alpha](\neg p \rightarrow \langle \alpha \rangle p)).$$

To see this, it suffices to prove that each state $s \in W_u$ that is reachable from w in \mathcal{M}_p , $(\mathcal{M}_p, s) \models [\alpha](\neg p \rightarrow \langle \alpha \rangle p)$ and $(\mathcal{M}_p, s_p) \models [\alpha](\neg p \rightarrow \langle \alpha \rangle p)$, where $\alpha \in A$. Let $sR_\alpha^p t$ or $s_p R_\alpha^p t$. By the construction of R^p , if $(\mathcal{M}_p, t) \models \neg p$, then $t \in W_u$, and therefore $tR_\alpha^p s_p$ and $(\mathcal{M}_p, s_p) \models p$. Therefore, we can conclude that $(\mathcal{M}_p, s) \models [\alpha](\neg p \rightarrow \langle \alpha \rangle p)$ and $(\mathcal{M}_p, s_p) \models [\alpha](\neg p \rightarrow \langle \alpha \rangle p)$, concluding the proof of the first implication.

Finally, we prove that for each $\alpha \in A$,

$$(\mathcal{M}_p, w) \models \text{Inv} \left(\bigwedge_{\langle \alpha \rangle \psi \in \overline{\text{sub}}(\varphi)} (\langle \alpha \rangle \text{cl}(\psi) \rightarrow \langle \alpha \rangle (\neg p \wedge \text{cl}(\psi))) \right).$$

Let $(\mathcal{M}_p, s) \models \langle \alpha \rangle \text{cl}(\psi)$. Then, there exists some t , such that $sR_\alpha^p t$ and $(\mathcal{M}_p, t) \models \text{cl}(\psi)$. If $s \in W_p$, then by construction, $t \in W_u$ and therefore $(\mathcal{M}_p, t) \models \neg p$, yielding that $(\mathcal{M}_p, s) \models \langle \alpha \rangle (\neg p \wedge \text{cl}(\psi))$. On the other hand, if $s \in W_u$, then we have that $(\mathcal{M}, s) \models \langle \alpha \rangle \text{cl}(\psi)$, and therefore there exists some $t' \in W_u$, such that $sR_\alpha t'$ and $(\mathcal{M}, t') \models \text{cl}(\psi)$. As we have observed above, (\mathcal{M}_u, t') and (\mathcal{M}_p, t') are bisimilar, and by the construction of \mathcal{M}_p we have that $sR_\alpha^p t'$ and $(\mathcal{M}_p, t') \models \text{cl}(\psi)$, yielding that $(\mathcal{M}_p, s) \models \langle \alpha \rangle (\neg p \wedge \text{cl}(\psi))$.

For the converse, we assume an $\mathbf{L}_2(\alpha)$ model $\mathcal{M} = (W, R, V)$ and $w \in W$, such that $(\mathcal{M}, w) \models \mathbf{F}_A^{B^\mu}(\varphi)$, and we construct an \mathbf{L}_1 -model \mathcal{M}' that satisfies φ . Let $\mathcal{M}' = (W', R', V')$, where

- $W' = W \setminus \{s \in W \mid (\mathcal{M}, s) \models p \text{ and } \exists (t, s) \in R_\alpha\}$;
- R'_α is the symmetric closure of $R_\alpha \cap (W')^2$, if $\alpha \in A$, and $R'_\alpha = R_\alpha \cap (W')^2$ otherwise;
- V' is the restriction of V on W' .

Let ρ be an environment such that for every Y , $\rho(Y) = \llbracket \text{cl}(Y) \rrbracket$ – that is, each X is mapped to the interpretation of its recursive closure – and let $\rho'(Y) = \rho(Y) \cap W'$. We prove that for every state $v \in W'$ that is reachable from w and $\psi \in \text{sub}(\varphi)$, $(\mathcal{M}, v) \models_\rho \psi$ implies that $(\mathcal{M}', v) \models_{\rho'} \psi$. We now proceed by induction on ψ .

- Propositional cases, the case of logical variables, and the cases of $\psi = [\alpha]\psi'$ and $\psi = \langle \alpha \rangle \psi'$, where $\alpha \notin A$, are straightforward.
- For the case of $\psi = \langle \alpha \rangle \psi'$, let $(\mathcal{M}, v) \models_\rho \psi$. From $(\mathcal{M}, w) \models \mathbf{F}_A^{B^\mu}(\varphi)$ we get that $(\mathcal{M}, v) \models_\rho \langle \alpha \rangle (\neg p \wedge \psi')$, and therefore there exists some $u \in W'$, such that $vR_\alpha u$ and $(\mathcal{M}, u) \models_\rho \psi'$. By the construction of R' and the inductive hypothesis, we get that $vR'_\alpha u$ and $(\mathcal{M}', u) \models_{\rho'} \psi'$, yielding that $(\mathcal{M}', v) \models_{\rho'} \psi$.
- For the case of $\psi = [\alpha]\psi'$, where $\alpha \in A$, let $(v, v') \in R'_\alpha$. It suffices to prove that $(\mathcal{M}', v') \models_{\rho'} \psi'$. If $(v, v') \in R_\alpha$, then we are done by the inductive hypothesis. If not, then $v'R_\alpha v$, and therefore there is some $(v, v'') \in R_\alpha$, such that $(\mathcal{M}, v'') \models p$. Therefore, $(\mathcal{M}, v'') \models_\rho \psi'$. From $(\mathcal{M}, w) \models \mathbf{F}_A^{B^\mu}(\varphi)$ we get that $(\mathcal{M}, v) \models_\rho \neg \psi' \rightarrow [\alpha][\alpha](p \rightarrow \neg \psi')$. Therefore, if we assume $(\mathcal{M}, v') \not\models_{\rho'} \psi'$, we see that $(\mathcal{M}, v'') \models_\rho \neg \psi'$, which contradicts our observations above. We conclude that $(\mathcal{M}, v') \models_\rho \psi'$, which by the inductive hypothesis yields $(\mathcal{M}', v') \models_{\rho'} \psi'$.
- For the case of $\psi = \nu Y.\psi'$, notice that $\llbracket \psi', \rho \rrbracket_{\mathcal{M}} \subseteq \llbracket \psi', \rho' \rrbracket_{\mathcal{M}'}$ by the inductive hypothesis, and

$$\llbracket \psi, \rho \rrbracket_{\mathcal{M}} = \llbracket \psi', \rho[Y \mapsto \llbracket \psi, \rho \rrbracket_{\mathcal{M}}] \rrbracket_{\mathcal{M}} = \llbracket \psi', \rho \rrbracket_{\mathcal{M}}$$

and

$$\llbracket \psi', \rho' \rrbracket_{\mathcal{M}'} = \llbracket \psi', \rho'[Y \mapsto \llbracket \psi, \rho' \rrbracket_{\mathcal{M}'}] \rrbracket_{\mathcal{M}'} = \llbracket \psi, \rho' \rrbracket_{\mathcal{M}'},$$

by the properties of the fixed points and the definitions of ρ, ρ' . By combining these observations, we get that $\llbracket \psi, \rho \rrbracket_{\mathcal{M}} \subseteq \llbracket \psi, \rho' \rrbracket_{\mathcal{M}'}$, which is what we wanted to prove. \square

Lemma 3.3 assures us that closure under some conditions does not affect other conditions. This observation allows our translations above to remove a frame condition without affecting others, which in turn lets us safely compose our translations. For example, we can combine Theorems 3.16 and 3.20 to translate from logics with agents that have both the symmetry and reflexivity condition (B and T). One can first apply the translation of Theorem 3.16 to

reduce the satisfiability problem to one where the reflexivity condition is not in the logic, and then apply the translation of Theorem 3.20 to obtain the following result.

Corollary 3.21. *Let $\mathbf{L}_1, \mathbf{L}_2$ be logics, such that $\mathbf{L}_1(\alpha) = \mathbf{L}_2(\alpha) + B + T$ when $\alpha \in A$, and $\mathbf{L}_2(\alpha)$ otherwise, and $\mathbf{L}_2(\alpha)$ at most includes frame condition D . Then, a formula φ that has no μX operators is \mathbf{L}_1 -satisfiable if and only if $\mathbf{F}_A^{\mathbf{B}^\mu}(\mathbf{F}_A^{T^\mu}(\varphi))$ is \mathbf{L}_2 -satisfiable.*

Remark 3.22. A translation for euclidean frames and for the full logic on symmetric frames would need different approaches. D’Agostino and Lenzi show in [DL13] that $\mathbf{S5}_2^\mu$ does not have a finite-model property, and their result can be easily extended to any logic \mathbf{L} with fixed-point operators, where there are at least two distinct agents α, β , such that $\mathbf{L}(\alpha)$ and $\mathbf{L}(\beta)$ have constraint B or 5 . Therefore, as our constructions for the translations to \mathbf{K}_k^μ guarantee the finite-model property for the corresponding logics, they do not apply to multimodal logics with B or 5 .

3.2. Embedding \mathbf{K}_n^μ . In this subsection, we present translations from logics with fewer frame conditions to ones with more conditions. These translations will allow us to prove EXP-completeness results in the following subsection. Let p, q be distinguished propositional variables that do not appear in our formulas. We let \vec{p} range over $p \wedge q$, $p \wedge \neg q$, and $\neg p \wedge q$.

Definition 3.23 (Function *next*). The function *next* is defined thus: $next(p \wedge q) = p \wedge \neg q$, $next(p \wedge \neg q) = \neg p \wedge q$, and $next(\neg p \wedge q) = p \wedge q$.

We use a uniform translation from \mathbf{K}_k^μ to any logic with a combination of conditions D, T, B . The intuition behind this translation is to describe an unfolded model where all of the unfolded states are iteratively marked with the next value of \vec{p} . Then we use the value of p and q in a state to distinguish whether the current state is a duplicate or a “true” successor of the original state (a duplicate might be caused due to reflexive or symmetric edges while unfolding). In other words, we want to be able to use modalities to quantify over the transitions in a reflexive or symmetric frame, while excluding any loops introduced by the frame’s conditions.

Translation 3.24. Let $A \subseteq \text{AG}$. The function $\mathbf{F}_A^{\mathbf{K}^\mu}(-)$ on formulas is defined such that:

- $\mathbf{F}_A^{\mathbf{K}^\mu}(\langle \alpha \rangle \psi) = \bigwedge_{\vec{p}} (\vec{p} \rightarrow \langle \alpha \rangle (next(\vec{p}) \wedge \mathbf{F}_A^{\mathbf{K}^\mu}(\psi)))$, if $\alpha \in A$;
- $\mathbf{F}_A^{\mathbf{K}^\mu}([\alpha] \psi) = \bigwedge_{\vec{p}} (\vec{p} \rightarrow [\alpha] (next(\vec{p}) \rightarrow \mathbf{F}_A^{\mathbf{K}^\mu}(\psi)))$, if $\alpha \in A$;
- $\mathbf{F}_A^{\mathbf{K}^\mu}(-)$ commutes with all other operations.

We note that there are simpler translations for the cases of logics with only D or T as a constraint, but the $\mathbf{F}_A^{\mathbf{K}^\mu}(-)$ is uniform for all the logics that we consider in this subsection.

Theorem 3.25. *Let $\emptyset \neq A \subseteq \text{AG}$, $|\text{AG}| = k$, and let \mathbf{L} be such that $\mathbf{L}(\alpha)$ includes only frame conditions from D, T, B when $\alpha \in A$, and $\mathbf{L}(\alpha) = \mathbf{K}$ otherwise. Then, φ is \mathbf{K}_k^μ -satisfiable if and only if $\mathbf{F}_A^{\mathbf{K}^\mu}(\varphi)$ is \mathbf{L} -satisfiable.*

Proof. For the “only if” direction, let $\mathcal{M} = (W, R, V)$ be an unfolded model and $w \in W$ be its root, such that $(\mathcal{M}, w) \models \varphi$. Variables p and q do not appear in φ , so we can assume that $(\mathcal{M}, w) \models p \wedge q$, and that, for each $vR_\alpha v'$ with $\alpha \in A$ and each \vec{p} , if $(\mathcal{M}, v) \models \vec{p}$ then $(\mathcal{M}, v') \models next(\vec{p})$. Let $\mathcal{M}' = (W, R', V)$, where R' is the closure of R under the conditions of \mathbf{L} . We observe that for every $\alpha \in A$, $R_\alpha = R'_\alpha \cap \bigcup_{\vec{p}} (\llbracket \vec{p} \rrbracket \times \llbracket next(\vec{p}) \rrbracket)$. We prove that for every environment ρ , every $\psi \in \text{sub}(\varphi)$, and every $v \in W$, $(\mathcal{M}, v) \models_\rho \psi$ iff $(\mathcal{M}', v) \models_\rho \mathbf{F}_A^{\mathbf{K}^\mu}(\psi)$. The proof proceeds by induction on ψ . We fix a $v \in W$ and a \vec{p} , such that $(\mathcal{M}, v) \models \vec{p}$.

- The propositional cases and the case of $\psi = X$ are immediate, and so are the cases of $\psi = \langle \alpha \rangle \psi'$ and $\psi = [\alpha] \psi'$, where $\alpha \notin A$.
- For the case of $\psi = \nu X.\psi'$ or $\psi = \mu X.\psi'$, note that due to the inductive hypothesis, for any $S \subseteq W$, $\llbracket \psi', \rho[X \mapsto S] \rrbracket_{\mathcal{M}} = \llbracket \mathbf{F}_A^{\mathbf{K}^\mu}(\psi'), \rho[X \mapsto S] \rrbracket_{\mathcal{M}'}$, and therefore $\llbracket \psi, \rho \rrbracket_{\mathcal{M}} = \llbracket \mathbf{F}_A^{\mathbf{K}^\mu}(\psi), \rho \rrbracket_{\mathcal{M}'}$.
- For the case of $\psi = \langle \alpha \rangle \psi'$, where $\alpha \in A$, we reason as follows: $(\mathcal{M}, v) \models_\rho \psi$ iff $(\mathcal{M}, v) \models_\rho \langle \alpha \rangle \psi'$ iff there is some $(v, v') \in R_\alpha$, such that $(\mathcal{M}, v') \models_\rho \psi'$ iff there is some $(v, v') \in R'_\alpha$, where $(\mathcal{M}, v') \models \text{next}(\vec{p})$, $(\mathcal{M}, v') \models_\rho \psi'$ iff there is some $(v, v') \in R'_\alpha$, such that $(\mathcal{M}, v') \models \text{next}(\vec{p}) \wedge \psi'$ iff

$$(\mathcal{M}, v') \models \mathbf{F}_A^{\mathbf{K}^\mu}(\langle \alpha \rangle \psi') = \bigwedge_{\vec{p}} (\vec{p} \rightarrow \langle \alpha \rangle (\text{next}(\vec{p}) \wedge \mathbf{F}_A^{\mathbf{K}^\mu}(\psi'))).$$

- For the case of $\psi = [\alpha] \psi'$, where $\alpha \in A$, $(\mathcal{M}, v) \models_\rho \psi$ iff $(\mathcal{M}, v) \models_\rho [\alpha] \psi'$ iff for every $(v, v') \in R_\alpha$, $(\mathcal{M}, v') \models_\rho \psi'$ iff for every $(v, v') \in R'_\alpha$, where $(\mathcal{M}, v') \models \text{next}(\vec{p})$, $(\mathcal{M}, v') \models_\rho \psi'$ iff for every $(v, v') \in R'_\alpha$, $(\mathcal{M}, v') \models \text{next}(\vec{p}) \rightarrow \psi'$ iff

$$(\mathcal{M}, v') \models \mathbf{F}_A^{\mathbf{K}^\mu}([\alpha] \psi') = \bigwedge_{\vec{p}} (\vec{p} \rightarrow [\alpha] (\text{next}(\vec{p}) \rightarrow \mathbf{F}_A^{\mathbf{K}^\mu}(\psi'))),$$

which completes the proof by induction.

For the “if” direction, let $\mathcal{M} = (W, R, V)$ be an \mathbf{L} -model and $w \in W$, be such that $(\mathcal{M}, w) \models \mathbf{F}_A^{\mathbf{K}^\mu}(\varphi)$. Let $\mathcal{M}' = (W, R', V)$, where for every $\alpha \in A$, $R'_\alpha = R_\alpha \cap \bigcup_{\vec{p}} (\llbracket \vec{p} \rrbracket \times \llbracket \text{next}(\vec{p}) \rrbracket)$, and for every $\alpha \notin A$, $R'_\alpha = R_\alpha$. We can prove that for any environment ρ , any $\psi \in \text{sub}(\varphi)$, and any $v \in W$, $(\mathcal{M}, v) \models_\rho \mathbf{F}_A^{\mathbf{K}^\mu}(\psi)$ iff $(\mathcal{M}', v) \models_\rho \psi$. The proof proceeds by induction on ψ and is very similar to the one for the “only if” direction. \square

3.3. Complexity Results. We observe that, with the exception of Translations 3.5 and 3.19, our translations yield formulas of size at most linear with respect to the input formula. Translations 3.5 and 3.19 have a quadratic cost.

Corollary 3.26. *If \mathbf{L} only has frame conditions D, T , then its satisfiability problem is EXP-complete; if \mathbf{L} only has frame conditions $D, T, 4$, then its satisfiability problem is in EXP.*

Proof. Immediately from Theorems 3.14, 3.16, 3.18, and 3.25. \square

Corollary 3.27. *If \mathbf{L} only has frame conditions D, T, B , then*

- (1) *\mathbf{L} -satisfiability is EXP-hard; and*
- (2) *the restriction of \mathbf{L} -satisfiability on formulas without μX operators is EXP-complete.*

Proof. Immediately from Theorems 3.14, 3.16, 3.20, and 3.25. \square

A summary of our results can be seen in Table 2. The only mentioned result that was already known is that the satisfiability problem for the μ -calculus over one agent is EXP-complete.

We can clearly see from the summary in Table 2 that the bound we have obtained via the translation-based methods presented in this section is not tight for the satisfiability problem for the single-agent μ -calculus over transitive frames. Specifically, the translations established already only guarantee that the problem is in EXP, but the only available lower bound is PSPACE-hardness from the corresponding modal logic (without recursion). Thus,

# agents	Restrictions on syntax/frames	Upper Bound	Lower Bound
≥ 2	frames with B or 5	?	EXP-hard 2.11
	not B , 5	EXP 3.26	EXP-hard 2.11
	not 5, not $\mu.X$	EXP 3.27	EXP-hard 2.10
1	with 5 (or B4)	NP 2.5	NP-hard 2.5
	with 4	EXP 3.26	PSPACE-hard 2.5
	Any other restrictions	EXP 2.8	EXP-hard 2.8

TABLE 2. Summary of the complexity of satisfiability checking for various modal logics with recursion up to section 3. A logic has the complexity of the first line from the top of the table that contains its description. The respective theorems from either this work or already known are linked.

to have a tight bound, we would need to prove that the satisfiability checking for $\mathbf{K4}^\mu$ is either in PSPACE, or that it is EXP-hard. We do not have an answer to this question that relies on translations, but we will present some musings on this topic in the discussion section (Sec. 7).

In what follows, instead of applying the translation method to obtain a tight complexity bound for satisfiability for the single-agent μ -calculus over transitive frames, we will utilize the tableaux we will present in Section 4 to produce a PSPACE algorithm, showing that the problem is, in fact, PSPACE-complete.

4. TABLEAUX FOR \mathbf{L}_k^μ

We give a sound and complete tableau system for logic \mathbf{L} . Furthermore, if \mathbf{L} has a finite-model property, then we give terminating conditions for its tableau. The system that we give in this section is based on Kozen’s tableaux for the μ -calculus [Koz83] and the tableaux of Fitting [Fit72] and Massacci [Mas94] for modal logic. We can use Kozen’s finite model theorem [Koz83] to help us ensure the termination of the tableau for some of these logics.

Theorem 4.1 [Koz83]. *There is a computable $\kappa : \mathbb{N} \rightarrow \mathbb{N}$, such that every \mathbf{K}_k^μ -satisfiable formula φ is satisfied in a model with at most $\kappa(|\varphi|)$ states.*⁵

Corollary 4.2. *If \mathbf{L} only has frame conditions $D, T, 4$, then there is a computable $\kappa : \mathbb{N} \rightarrow \mathbb{N}$, such that every \mathbf{L} -satisfiable formula φ is satisfied in a model with at most $\kappa(|\varphi|)$ states.*

Proof. Immediately, from Theorems 4.1, 3.14, 3.16, and 3.18, and Lemma 3.3. \square

Remark 4.3. We note that not all modal logics with recursion have a finite-model property — see Remark 3.22.

Intuitively, a tableau attempts to build a model that satisfies the given formula. When it needs to consider two possible cases, it branches, and thus it may generate several branches. Each branch that satisfies certain consistency conditions, which we define below, represents a corresponding model.

⁵The tableau in [Koz83] yields an upper bound of $2^{2^{O(n^3)}}$ for $\kappa_0(n)$, but that bound is not useful to obtain a “good” decision procedure. The purpose of this section is not to establish any good upper bound for satisfiability testing, which is done in Sections 5 and 6.

$$\begin{array}{cccc}
\frac{\sigma \pi X.\varphi}{\sigma \varphi} \text{ (fix)} & \frac{\sigma X}{\sigma \text{fx}(X)} \text{ (X)} & \frac{\sigma \varphi \vee \psi}{\sigma \varphi \mid \sigma \psi} \text{ (or)} & \frac{\sigma \varphi \wedge \psi}{\sigma \varphi} \text{ (and)} \\
\frac{\sigma [\alpha]\varphi}{\sigma.\alpha\langle\psi\rangle \varphi} \text{ (B)} & \frac{\sigma \langle\alpha\rangle\varphi}{\sigma.\alpha\langle\varphi\rangle \varphi} \text{ (D)} & \frac{\sigma [\alpha]\varphi}{\sigma.\alpha\langle\varphi\rangle \varphi} \text{ (d)} & \frac{\sigma [\alpha]\varphi}{\sigma.\alpha\langle\psi\rangle [\alpha]\varphi} \text{ (4)}
\end{array}$$

where, for rules (B) and (4), $\sigma.\alpha\langle\psi\rangle$ has already appeared in the branch; and for (D), σ is not α -flat.

$$\begin{array}{cccc}
\frac{\sigma.\alpha\langle\psi\rangle [\alpha]\varphi}{\sigma [\alpha]\varphi} \text{ (B5)} & \frac{\sigma.\alpha\langle\psi\rangle \langle\alpha\rangle\varphi}{\sigma.\alpha\langle\psi\rangle.\alpha\langle\varphi\rangle \varphi} \text{ (D5)} & \frac{\sigma.\alpha\langle\psi\rangle [\alpha]\varphi}{\sigma \varphi} \text{ (b)} & \frac{\sigma [\alpha]\varphi}{\sigma \varphi} \text{ (t)} \\
\frac{\sigma.\alpha\langle\psi\rangle [\alpha]\varphi}{\sigma.\alpha\langle\psi'\rangle [\alpha]\varphi} \text{ (B55)} & \frac{\sigma.\alpha\langle\psi\rangle.\alpha\langle\psi'\rangle \langle\alpha\rangle\varphi}{\sigma.\alpha\langle\psi\rangle.\alpha\langle\varphi\rangle \varphi} \text{ (D55)} & \frac{\sigma.\alpha\langle\psi\rangle [\alpha]\varphi}{\sigma [\alpha]\varphi} \text{ (b4)} &
\end{array}$$

where, for rule (B55), $\sigma.\alpha\langle\psi'\rangle$ has already appeared in the branch; for rule (D5), σ is not α -flat, and $\sigma \langle\alpha\rangle\varphi$ does not appear in the branch; for rule (D55), $\sigma \langle\alpha\rangle\varphi$ does not appear in the branch.

TABLE 3. The tableau rules for $\mathbf{L} = \mathbf{L}_n^\mu$.

Our tableaux use *prefixed formulas*, that is, formulas of the form $\sigma \varphi$, where $\sigma \in (\text{AG} \times L)^*$ and $\varphi \in L$; σ is the prefix of φ in that case, and we say that φ is prefixed by σ . We note that we separate the elements of σ with a dot. Furthermore, in the tableau prefixes, we write $\alpha\langle\psi\rangle$ to mean the pair $(\alpha, \psi) \in \text{AG} \times L$. Therefore, for example, $(\alpha, \phi)(\beta, \chi)(\alpha, \psi)$ is written $\alpha\langle\phi\rangle.\beta\langle\chi\rangle.\alpha\langle\psi\rangle$ as a tableau prefix. We say that the prefix σ is α -flat when agent α has axiom 5 and $\sigma = \sigma'.\alpha\langle\psi\rangle$ for some ψ . Each prefix possibly represents a state in a corresponding model, and a prefixed formula $\sigma \varphi$ declares that φ is satisfied in the state represented by σ . As we will see below, the prefixes from $(\text{AG} \times L)^*$ allow us to keep track of the diamond formula that generates a prefix through the tableau rules. For agents with condition 5, this allows us to restrict the generation of new prefixes and avoid certain redundancies, due to the similarity of euclidean binary relations to equivalence relations [NT85, HR07]. For single-agent logics with condition 4, this notation will further let us avoid redundancies and achieve a more efficient decision procedure — see Section 5.

The tableau rules that we use appear in Table 3. These include fixed-point and propositional rules, as well as rules that deal with modalities. Depending on the logic that each agent α is based on, a different set of rules applies for α : for rule (d), $\mathbf{L}(\alpha)$ must have condition *D*; for rule (t), $\mathbf{L}(\alpha)$ must have condition *T*; for rule (4), $\mathbf{L}(\alpha)$ must have condition 4; for rules (B5), (D5), and (D55), $\mathbf{L}(\alpha)$ must have condition 5; for rule (b) $\mathbf{L}(\alpha)$ must have condition *B*; and for rule (b4) $\mathbf{L}(\alpha)$ must have both *B* and 4. Rule (or) is the only rule that splits the current tableau branch into two. When an agent α does not have condition 5, diamond formulas such as $\sigma \langle\alpha\rangle\psi$ are analysed by extending the current prefix σ to $\sigma.\alpha\langle\psi\rangle$,

using rule (D). But when α has condition 5, we need to take into account that due to the Euclidean condition, formulas prefixed with σ may affect prefixes that neither extend nor are prefixes of σ , in a similar way as for states in a Kripke model. As Lemma 4.6 demonstrates later on in this section, condition 5 results in an accessibility relation that is very close to an equivalence relation, and our tableau rules for agents with condition 5 (*i.e.* rules (B5), (B55), (D5), and (D55)) reflect this.

A tableau branch is propositionally closed when $\sigma \text{ ff}$ or both σp and $\sigma \neg p$ appear in the branch for some prefix σ . For each prefix σ that appears in a fixed tableau branch, let $\Phi(\sigma)$ be the set of formulas prefixed by σ in that branch. We use the notation $\sigma \prec \sigma'$ to mean that $\sigma' = \sigma.\sigma''$ for some σ'' , in which case σ is an ancestor of σ' . We denote with $\sigma(b)$ the set of prefixes that appear in a branch b .

We define the dependence relation \xrightarrow{X} on prefixed formulas in a tableau branch as $\chi_1 \xrightarrow{X} \chi_2$, if χ_2 was introduced to the branch by a tableau rule with χ_1 as its premise, and χ_1 is not of the form σY , where $X < Y$; then, $\xrightarrow{X+}$ is the transitive closure of \xrightarrow{X} and $\xrightarrow{X*}$ is its reflexive and transitive closure. We can also extend this relation to prefixes, so that $\sigma \xrightarrow{X} \sigma'$, if and only if $\sigma \psi \xrightarrow{X} \sigma' \psi'$, for some $\psi \in \Phi(\sigma)$ and $\psi' \in \Phi(\sigma')$. If in a branch there is a \xrightarrow{X} -sequence where X is a least fixed-point and appears infinitely often, then the branch is called fixed-point-closed. A branch is closed when it is either fixed-point-closed or propositionally closed; if it is not closed, then it is called open.

Now, assume that there is a $\kappa : \mathbb{N} \rightarrow \mathbb{N}$, such that every \mathbf{L} -satisfiable formula φ is satisfied in a model with at most $\kappa(|\varphi|)$ states. An open tableau branch is called (*resp. locally*) *maximal* when all tableau rules (*resp. the tableau rules that do not produce new prefixes*) have been applied. A branch is called *sufficient* for φ when it is locally maximal and for every $\sigma \psi$ in the branch, for which a rule can be applied and has not been applied to $\sigma \psi$, $|\sigma| > |\text{AG}| \cdot \kappa(|\varphi|)^{|\varphi|^2} \cdot 2^{|\varphi|+1}$. A tableau is called maximal when all of its open branches are maximal, and closed when all of its branches are closed. It is called sufficiently closed for φ if it is propositionally closed, or for some least fixed-point variable X , it has a \xrightarrow{X} -path, where X appears at least $\kappa(|\varphi|) + 1$ times. A sufficient branch for φ that is not sufficiently closed is called sufficiently open for φ .

A tableau for φ starts from $\varepsilon \varphi$ and is built using the tableau rules of Table 3. A tableau proof for φ is a closed tableau for the negation of φ .

The following example demonstrates the tableau system in action.

Example 4.4. Let $\text{AG} = \{a, b\}$ and \mathbf{L} be a logic, such that $\mathbf{L}(a) = \mathbf{K}^\mu$ and $\mathbf{L}(b) = \mathbf{K5}^\mu$. Let

$$\varphi_1 = (p \wedge \langle a \rangle p) \wedge \mu X.(\neg p \vee [a]X) \quad \text{and} \quad \varphi_2 = \langle b \rangle p \wedge \mu X.([b]\neg p \vee [b]X).$$

As we see in Figure 4, the tableau for φ_1 produces an open branch, while the one for φ_2 has all of its branches closed, the leftmost one due to an infinite \xrightarrow{X} -sequence.

Theorem 4.5 (Soundness, Completeness, and Termination of \mathbf{L}_k^μ -Tableaux). *The first two of the following three statements are equivalent for any formula $\varphi \in L$ and any logic \mathbf{L} . Furthermore, if there is a $\kappa : \mathbb{N} \rightarrow \mathbb{N}$, such that every \mathbf{L} -satisfiable formula φ is satisfied in a model with at most $\kappa(|\varphi|)$ states, then all the following statements are equivalent.*

- (1) φ has a maximal \mathbf{L} -tableau with an open branch;
- (2) φ is \mathbf{L} -satisfiable; and
- (3) φ has an \mathbf{L} -tableau with a sufficiently open branch for φ .

$$\begin{array}{c}
\frac{\varepsilon (p \wedge \langle a \rangle p) \wedge \mu X.(\neg p \vee [a]X)}{\varepsilon \mu X.(\neg p \vee [a]X)} \\
\frac{\varepsilon p \wedge \langle a \rangle p}{\varepsilon p} \\
\frac{\varepsilon \langle a \rangle p}{\varepsilon \neg p \vee [a]X} \text{ (fix)} \\
\frac{\varepsilon [a]X}{\frac{a \langle p \rangle p}{a \langle p \rangle X} \text{ (B)}} \text{ (D)} \quad \frac{\varepsilon \neg p}{\mathbf{x}} \\
\frac{a \langle p \rangle \mu X.(\neg p \vee [a]X)}{a \langle p \rangle \neg p \vee [a]X} \text{ (X)} \text{ (fix)} \\
\frac{a \langle p \rangle [a]X \quad a \langle p \rangle \neg p}{\mathbf{x}}
\end{array}
\qquad
\begin{array}{c}
\frac{\varepsilon \langle b \rangle p \wedge \mu X.([b]\neg p \vee [b]X)}{\varepsilon \mu X.([b]\neg p \vee [b]X)} \\
\frac{\varepsilon \langle b \rangle p}{b \langle p \rangle p} \text{ (D)} \\
\frac{b \langle p \rangle p}{\varepsilon [b]\neg p \vee [b]X} \text{ (fix)} \\
\frac{\varepsilon [b]X}{b \langle p \rangle X} \text{ (B)} \quad \frac{\varepsilon [b]\neg p}{b \langle p \rangle \neg p} \text{ (B)} \\
\frac{b \langle p \rangle \mu X.([b]\neg p \vee [b]X)}{b \langle p \rangle [b]\neg p \vee [b]X} \text{ (fix)} \text{ (X)} \\
\frac{b \langle p \rangle [b]X}{\varepsilon [b]X} \text{ (B5)} \quad \frac{b \langle p \rangle [b]\neg p}{\varepsilon [b]\neg p} \text{ (B5)} \\
\frac{\varepsilon [b]X}{b \langle p \rangle X} \text{ (B)} \quad \frac{\varepsilon [b]\neg p}{b \langle p \rangle \neg p} \text{ (B)} \\
\frac{b \langle p \rangle [b]X}{\varepsilon [b]X} \text{ (X)} \quad \frac{b \langle p \rangle [b]\neg p}{\varepsilon [b]\neg p} \text{ (B)} \\
\vdots
\end{array}$$

FIGURE 4. Tableaux for φ_1 and φ_2 . The dots represent that the tableau keeps repeating as from the identical node above. The \mathbf{x} mark represents a propositionally closed branch.

Here we state and prove certain definitions and lemmata that will be used in the proof of Theorem 4.5. For an agent $\alpha \in \text{AG}$ and a state s in a model (W, R, V) , let $\text{Reach}_\alpha(s)$ be the states that are reachable in W by R_α . We also use $R|_S$ for the restriction of a relation R on a set S .

For a logic \mathbf{L} , we say that a state s in a model $\mathcal{M} = (W, R, V)$ is \mathbf{L} flat when, for every $\alpha \in \text{AG}$ for which $\mathbf{L}(\alpha)$ has constraint 5, there is a set of states W_0 , such that:

- $\text{Reach}_\alpha(s) = \{s\} \cup W_0$;
- $R_\alpha|_{\text{Reach}_\alpha(s)} = E_0 \cup E_1$, where
 $E_0 \subseteq \{s\} \times W_0$ and
 $E_1 = W_0^2$; and
- if $\mathbf{L}(\alpha)$ has constraint T , or $E_0 \neq \emptyset$ and $\mathbf{L}(\alpha)$ has constraint B , then $s \in W_0$.

Lemma 4.6 [NT85, HR07]. *Every pointed \mathbf{L} -model is bisimilar to \mathbf{L} -model whose states are all \mathbf{L} flat.*

We are now ready to proceed with our proof of Theorem 4.5.

Proof of Theorem 4.5. This proof is in three parts.

We first prove that statement 1 implies statement 2 To this end, let b be a maximal open branch in the tableau for φ . We construct a \mathbf{K}_k^μ -model $\mathcal{M} = (W, R, V)$ for φ in the following way. Let W be the set of prefixes that appear in the branch, and let, for each $\alpha \in \text{AG}$,

$$R_\alpha^0 = \{(\sigma, \sigma.\alpha\langle\psi\rangle) \in W^2\} \cup \left\{ (\sigma, \sigma) \in W^2 \mid \mathbf{L}(\alpha) \text{ has } \begin{array}{l} \text{reflexive frames, or} \\ \text{serial frames and} \\ \forall\psi. \sigma.\alpha\langle\psi\rangle \notin W^2 \end{array} \right\};$$

R_α^1 is the symmetric closure of R_α^0 , if $\mathbf{L}(\alpha)$ has symmetric frames, and it is R_α^0 otherwise;
 R_α^2 is the euclidean closure of R_α^1 , if $\mathbf{L}(\alpha)$ has euclidean frames, and it is R_α^1 otherwise;
and finally, R_α is the transitive closure of R_α^2 , if $\mathbf{L}(\alpha)$ has transitive frames, and it is R_α^2

otherwise. By Lemma 3.3, R_a satisfies all the necessary closure conditions. We also set $V(p) = \{\sigma \in W \mid \sigma p \text{ appears in the branch}\}$.

It is now possible to prove, by straightforward induction, that for every subformula ψ of φ , if $\sigma \psi$ appears in the branch, then for any environment ρ , such that $\{\sigma' \in W \mid \sigma \psi \xrightarrow{X^*} \sigma' X\} \subseteq \rho(X)$, $\sigma \in \llbracket \psi, \rho \rrbracket$. The only interesting cases are fixed-point formulas, so let $\psi = \nu X.\psi'$. Let S_X be the set of prefixes of X in the branch. We can immediately see that if σX appears in the branch, then so does $\sigma \psi'$, and therefore, by the inductive hypothesis, $S_X \subseteq \llbracket \psi', \rho[X \mapsto S_X] \rrbracket$. From the semantics in Table 1, $\sigma \in \llbracket \psi, \rho \rrbracket$.

On the other hand, if $\psi = \mu X.\psi'$, then we prove that if $\sigma \notin S \subseteq W$, then $S \not\subseteq \llbracket \psi', \rho[X \mapsto S] \rrbracket$. Let $\Psi = \{\sigma' \chi \mid \sigma \psi' \xrightarrow{X^*} \sigma' \chi \text{ and } \sigma' \notin S\}$. We know that $\sigma \psi' \in \Psi$, so $\Psi \neq \emptyset$. There are no infinite \xrightarrow{X} -paths in the branch, so there is some $\sigma' \psi' \in \Psi$, such that $\sigma' \psi' \not\xrightarrow{X^+} \sigma'' \psi'$ for any σ'' . Then, we see that $\{\sigma'' \in W \mid \sigma' \psi \xrightarrow{X^*} \sigma'' X\} \subseteq S$, because if $\sigma' \psi \xrightarrow{X^*} \sigma'' X$, then $\sigma' \psi' \xrightarrow{X^*} \sigma'' \psi'$. But then, by the inductive hypothesis, $\sigma' \in \llbracket \psi', \rho[X \mapsto S] \rrbracket$, and therefore $S \not\subseteq \llbracket \psi', \rho[X \mapsto S] \rrbracket$, which was what we wanted to prove.

We now prove that 2 implies 3, when there is a $\kappa : \mathbb{N} \rightarrow \mathbb{N}$, such that every \mathbf{L} -satisfiable formula φ is satisfied in a model with at most $\kappa(|\varphi|)$ states. Assume that $\mathcal{M} = (W, R, V)$ be a \mathbf{L} -model, $w \in W$, $(\mathcal{M}, w) \models \varphi$, and W has at most $\kappa(|\varphi|)$ states.

In the rest of the proof, we fix a lfp-finite dependency relation on \mathcal{M} and φ , according to Theorem 2.13, and use it to construct an \mathbf{L} -tableau with a sufficiently open branch for φ . The tableau starts with $\varepsilon \varphi$ and we can keep expanding this branch to a sufficient one using the tableau rules, such that every prefix is mapped to a state in W , whenever σ is mapped to u , $(\mathcal{M}, u) \models_\rho \psi$ for every $\psi \in \Phi(\sigma)$, and if $\sigma.\alpha\langle\psi\rangle$ to v , then $uR_\alpha v$; furthermore, this can be done by following the lfp-finite dependency relation. This is by straightforward induction on the application of the tableau rules. A special case are the agents with euclidean accessibility relations, for which we can use Lemma 4.6. It is not hard to see that in this way we generate a set of branches that are not propositionally closed. Furthermore, since the tableau rule applications follow a lfp-finite dependency relation and W has at most $\kappa(|\varphi|)$ states, it is not hard to see that for every least-fixed-point variable X , on every \xrightarrow{X} -path, X appears at most $\kappa(|\varphi|)$ times.

We now prove that 2 implies 1, without assuming the finite model property for \mathbf{L} . To do this, we follow exactly the reasoning of the implication from 2 to 3, without assuming that the model has $\kappa(|\varphi|)$ states.

Finally, we prove that 3 implies 1, when there is a $\kappa : \mathbb{N} \rightarrow \mathbb{N}$, such that every \mathbf{L} -satisfiable formula φ is satisfied in a model with at most $\kappa(|\varphi|)$ states. To do so, we assume that there is a sufficient open branch b in a tableau for φ and we demonstrate that φ has a maximal tableau with an open branch — specifically, we construct such an open branch from b .

We call a prefix σ a leaf when there is no $\sigma' \neq \sigma$ in the branch, such that $\sigma \prec \sigma'$; we call σ productive when a tableau rule on a formula $\sigma \psi$ in the branch can produce a new prefix. We call σ ready if it is of the form $\sigma'.\alpha\langle\psi_1\rangle.\alpha\langle\psi_2\rangle$, or if σ is not α -flat for any $\alpha \in \text{AG}$.

For each $\sigma \psi$ in b and each least-fixed-point variable X , let

$$c(\sigma \psi, X) = \max \left\{ n \mid \text{there is a } \xrightarrow{X} \text{-path that ends in } \sigma \psi, \text{ where } X \text{ appears } n \text{ times} \right\}.$$

Since b is not sufficiently closed, always $c(\sigma \psi, X) \leq \kappa(|\varphi|)$. We use the notation $\sigma \sim \sigma'$ to mean that $\sigma = \sigma_1.\alpha\langle\psi_1\rangle$, $\sigma' = \sigma_2.\alpha\langle\psi_2\rangle$ for some $\sigma_1, \sigma_2, \alpha$, and $\Phi(\sigma) = \Phi(\sigma')$; and the notation $\sigma \equiv \sigma'$ to mean that

- $\sigma \sim \sigma'$,
- for every least-fixed-point variable X and $\psi \in \Phi(\sigma)$, $c(\sigma \psi, X) = c(\sigma' \psi, X)$, and
- either both σ and σ' are ready, or both σ and σ' are not ready and $\sigma_1 \sim \sigma_2$, where $\sigma = \sigma_1.\alpha\langle\psi\rangle$ and $\sigma' = \sigma_2.\alpha\langle\psi'\rangle$.

We then say that σ and σ' are equivalent.

The \equiv -equivalence class of a ready prefix σ is determined by $c(\sigma \psi, X)$ for every ψ and X ; by the last action $\alpha \in \text{AG}$ that appears in σ ; and by $\Phi(\sigma) \in 2^{\text{sub}(\varphi)}$. Therefore, there are at most $|\text{AG}| \cdot \kappa(|\varphi|)^{|\varphi|^2} \cdot 2^{|\varphi|}$ equivalence classes of ready prefixes.

Since b is sufficiently open, every productive leaf σ in the branch has more than $|\text{AG}| \cdot \kappa(|\varphi|)^{|\varphi|^2} \cdot 2^{|\varphi|+1}$ ancestors, and at least half of these must be ready. Therefore, by the Pigeonhole Principle, every productive leaf σ has an ancestor $e(\sigma)$ that has a distinct, ready, and \equiv -equivalent ancestor; let $s(\sigma)$ be such an ancestor of $e(\sigma)$. We further assume that $e(\sigma)$ is the \prec -minimal ancestor of σ with these properties. We note that for any two productive leaves σ_1 and σ_2 , if $e(\sigma_1) \prec \sigma_2$, then $e(\sigma_1) = e(\sigma_2)$.

Let b_0 be the branch that results by removing from b all prefixed formulas of the form $\sigma.\alpha\langle\psi_1\rangle \psi_2$, where $e(\sigma') \prec \sigma$ for some productive leaf σ' . Observe that b_0 is such that each of its productive leaves has an equivalent ancestor that is not a leaf. To complete the proof, it suffices to show how to extend any branch b_i , where each of its productive leaves σ has an equivalent proper ancestor $s(\sigma)$, to a branch b_{i+1} that preserves this property, and has an increased minimum length of its productive leaves. To form b_{i+1} , simply add to b_i all formulas of the form $\sigma.\sigma' \psi$, where $s(\sigma).\sigma' \psi$ appears in b_i . \square

Corollary 4.7. *L-tableaux are sound and complete for L.*

5. THE CASE OF $\mathbf{K4}^\mu$

In this section, we prove that the satisfiability problem for the (single-agent) logics $\mathbf{K4}^\mu$, $\mathbf{D4}^\mu$, and $\mathbf{S4}^\mu$ is in PSPACE, using an optimized decision procedure based on the tableaux from Section 4. To achieve this, we will examine how transitivity allows us to restrict the parts of a branch that we need to examine, in order to verify that a formula is satisfiable.

The tableaux rules we use are shown in Table 4, and are the restriction of the rules defined in Table 3 to $\mathbf{K4}^\mu$, $\mathbf{D4}^\mu$, and $\mathbf{S4}^\mu$ for the single-agent setting. We note that as the prefixes of the tableau represent states in a satisfying model, the combination of rules (B) and (4) ensures that from $\Box\varphi$ we introduce φ to all subsequent prefixes, as one would want in a transitive frame.

Given a branch b and prefix $\sigma \in \sigma(b)$, we define $b[\sigma :] = \{\sigma' \psi \mid \sigma.\sigma' \psi \in b\}$. A branch b for φ is called *repeating* if $b[\sigma\langle\psi\rangle :] = b[\sigma\langle\psi\rangle\sigma'\langle\psi\rangle :]$ for every $\sigma\langle\psi\rangle\sigma'\langle\psi\rangle \in \sigma(b)$.

Lemma 5.1. *Let $\mathbf{L} \in \{\mathbf{K4}^\mu, \mathbf{D4}^\mu, \mathbf{S4}^\mu\}$. A formula φ is L-satisfiable if and only if it has a maximal open, repeating branch b .*

Proof. Let φ be satisfiable. By Corollary 4.2, φ has a finite model $\mathcal{M} = (W, R, V)$. We revisit the proof of Theorem 4.5, and specifically the construction of the tableau from the finite model. We can ensure that when a new prefix $\sigma\langle\psi\rangle$ is mapped to a state $u_{\sigma\langle\psi\rangle}$, that state is in a strongly connected component (SCC) that is maximally deep in the dag of the

$$\begin{array}{c}
\frac{\sigma \pi X.\varphi}{\sigma \varphi} \text{ (fix)} \quad \frac{\sigma X}{\sigma \text{fx}(X)} \text{ (X)} \quad \frac{\sigma \varphi \vee \psi}{\sigma \varphi \mid \sigma \psi} \text{ (or)} \quad \frac{\sigma \varphi \wedge \psi}{\sigma \varphi} \text{ (and)} \\
\frac{\sigma \Box \varphi}{\sigma.\langle \psi \rangle \varphi} \text{ (B)} \quad \frac{\sigma \Diamond \varphi}{\sigma.\langle \varphi \rangle \varphi} \text{ (D)} \quad \frac{\sigma \Box \varphi}{\sigma.\langle \varphi \rangle \varphi} \text{ (d)} \quad \frac{\sigma \Box \varphi}{\sigma.\langle \psi \rangle \Box \varphi} \text{ (4)} \quad \frac{\sigma \Box \varphi}{\sigma \varphi} \text{ (t)}
\end{array}$$

where, for rules (B) and (4), $\sigma.\langle \psi \rangle$ has already appeared in the branch.

TABLE 4. The tableau rules for $\mathbf{L} \in \{\mathbf{K4}^\mu, \mathbf{D4}^\mu, \mathbf{S4}^\mu\}$.

SCCs in (W, R) , such that $(\mathcal{M}, u_{\sigma\langle \psi \rangle}) \models \psi$. So, if an extension $\sigma\langle \psi \rangle\sigma'\langle \psi \rangle$ is mapped to some state v , then either the SCC of $u_{\sigma\langle \psi \rangle}$ is not maximal or v and $u_{\sigma\langle \psi \rangle}$ are in the same SCC, and therefore we can ensure that $v = u_{\sigma\langle \psi \rangle}$. The converse direction is immediate from Theorem 4.5. \square

Repeating branches ensure that an algorithm that checks for the satisfiability of φ only needs to consider prefixes of length linear with respect to $|\varphi|$. We can go further and consider branches with less redundancy: due to transitivity, the satisfaction of certain diamond formulas in some prefix may be further postponed.

Definition 5.2. We will use the following terms:

- (1) We call a branch b *terse*, if
 - no formula appears twice in a prefix, and
 - for all $\psi_1 \neq \psi_2$, if $\sigma.\langle \psi_1 \rangle, \sigma.\langle \psi_2 \rangle \in \sigma(b)$, then $\Diamond \psi_2$ is not prefixed by any extension of $\sigma.\langle \psi_1 \rangle$ in b .
- (2) The branch b is *maximally terse*, if
 - it is terse, and
 - closed under the tableau rules, except (D) and (d), but if $\sigma \Diamond \psi \in b$ (*resp.* or $\sigma \Box \psi \in b$, if (d) is a tableau rule), then $\sigma'\langle \psi \rangle \psi \in b$ for some σ' , where σ' is an extension of σ or σ is an extension of σ' ; and if $\sigma \Box \psi \in b$ and $\sigma \Diamond \psi' \in b$, then $\sigma'\langle \psi' \rangle \psi \in b$ for any σ' that is an extension of $\sigma'\langle \psi' \rangle$.
- (3) A maximally terse branch b is called *4-open* when b remains open, after extending the definition of \xrightarrow{X} , so that additionally:
 - $\sigma \Diamond \psi \xrightarrow{X} \sigma'\langle \psi \rangle \psi$ for every σ' , where σ' is an extension of σ or σ is an extension of σ' ;
 - $\sigma \Box \psi \xrightarrow{X} \sigma'\langle \psi \rangle \psi$ for every σ' , where σ' is an extension of σ or σ is an extension of σ' , if (d) is included in the tableau rules; and
 - $\sigma \Box \psi \xrightarrow{X} \sigma'\langle \psi' \rangle \psi$ for every σ' , where σ is an extension of $\sigma'\langle \psi \rangle$ and $\sigma \Diamond \psi' \in b$.

Lemma 5.3. *Let $\mathbf{L} \in \{\mathbf{K4}^\mu, \mathbf{D4}^\mu, \mathbf{S4}^\mu\}$. A formula φ is \mathbf{L} -satisfiable if and only if it has a maximally terse, 4-open branch.*

Proof. If φ has a maximally terse, 4-open repeating branch, we can construct a model for φ , similarly as in the proof of Theorem 4.5.

Conversely, Let φ be satisfiable and let b be a maximal, open, repeating branch for φ by Lemma 5.1. Let b^- be a maximal terse sub-branch of b . It is immediate that b^- is closed

under the tableau rules, except (D) and (d). Due to the maximality of b^- , if $\sigma \diamond\psi \in b^-$ (resp. or $\sigma \Box\psi \in b$, if (d) is a tableau rule), then $\sigma'\langle\psi\rangle \psi \in b$ for some σ' , where σ' is an extension of σ or σ is an extension of σ' – otherwise, b^- can be extended to a larger sub-branch of b . Since b is repeating, if $\sigma \Box\psi \in b^-$ and $\sigma \diamond\psi' \in b^-$, then $\sigma'\langle\psi'\rangle \psi \in b^-$ for any σ that is an extension of $\sigma'\langle\psi'\rangle^-$. Therefore, b^- is maximally terse.

We use the extended definition of \xrightarrow{X} as in the definition of 4-open branches in Definition 5.2. By the transitivity of R and following the proof of Theorem 4.5, since b is repeating, and due to rule (4), every \xrightarrow{X} -cycle in b^- corresponds to an infinite \xrightarrow{X} -path in b , and therefore b^- is 4-open. \square

Corollary 5.4. *Let $\mathbf{L} \in \{\mathbf{K4}^\mu, \mathbf{D4}^\mu, \mathbf{S4}^\mu\}$. A formula φ is \mathbf{L} -satisfiable if and only if it is satisfied in a model of at most $2^{|\varphi|} - 1$ states.*

Lemma 5.5. *Let b be a terse branch. If a \xrightarrow{X} -path in b visits two prefixes, then one is an extension of the other.*

Theorem 5.6. *The satisfiability problem for $\mathbf{K4}^\mu$, $\mathbf{D4}^\mu$, and $\mathbf{S4}^\mu$, is PSPACE-complete.*

Proof. PSPACE-hardness results from Theorem 2.5. We prove that $\mathbf{K4}^\mu$ -, $\mathbf{D4}^\mu$ -, and $\mathbf{S4}^\mu$ -satisfiability is in PSPACE. We describe a non-deterministic polynomial-space algorithm that, given a formula φ , constructs and explores a maximally terse, 4-open $\mathbf{K4}^\mu$ -branch for φ , if one exists, and it verifies that the branch is 4-open. By Lemma 5.3, this suffices for checking the satisfiability of φ .

Our algorithm can generate and explore a maximally terse branch b with a depth-first-search on its prefixes. Note that the maximality and 4-openness conditions for b require only checking pairs of prefixes where one is an extension of the other. The algorithm keeps track of the diamond formulas (or also the box formulas, in case of rule (D)) that have generated a prefix. When the algorithm has visited a maximally long prefix σ , it can verify that it contains no \xrightarrow{X} -cycles where X appears. It also verifies that no $\sigma' \diamond\psi$ is marked as having generated $\sigma'\langle\psi\rangle$, where σ extends σ' but not $\sigma'\langle\psi\rangle$. If either of those verifications fail, the algorithm rejects. The algorithm then backtracks until all these formulas have generated a prefix. If there is some maximally terse, 4-open branch for φ , then the algorithm will succeed by following that branch. If, on the other hand, the algorithm succeeds, then the prefixed formulas it generates form a branch that is terse, as it marks the diamond formulas that generate a prefix, and verifies that the terseness condition is satisfied. It is also not hard to see that the generated branch is maximally terse and 4-open. The algorithm uses at most polynomial space to store a maximal prefix, the prefixed formulas, and the markings of these formulas, while it performs the depth-first-search exploration. The verifications it performs at each maximal prefix only require polynomial space. \square

6. TRANSLATIONS THROUGH TABLEAUX

In this section, we give a uniform translation from every modal logic with fixed-points that we have defined in this paper to the μ -calculus. This translation thus establishes the decidability and an upper bound for the complexity of satisfiability for all the cases that were not provided an upper bound via the methods used in Section 3. The key observation in this section is that we already have prefixed tableau proof systems for our logics, with the following properties:

- the subformula property, in that all formulas that appear in a tableau are prefixed subformulas of the starting formula;
- the tableau rules are local, in that they relate formulas prefixed either with the same prefix or with prefixes that differ very little from one another;
- the tableau proof systems produce a complete open branch when the starting formula is satisfiable.

An open branch has a bounded-branching tree structure with respect to the prefixes, but that tree can be infinite. The fact that a model or a branch can be infinite appears to cause difficulties in deciding the satisfiability of a logic, especially when symmetry or negative introspection is involved [DL13]. However, we observe that the fixed-points in the μ -calculus, the logic that is the target of our translations, are explicitly designed to describe infinite models. Our approach is to view an open branch as a model and to describe the property that the branch is open via a μ -calculus formula. The caveat is that our translation results in a formula that can be exponentially larger than the original, and thus yields a double-exponential upper bound for the satisfiability problem. For the sake of simplicity, in what follows we first assume that axiom 5 is not present. Afterward, we explain how to adjust our construction to handle agents with negative introspection.

6.1. The case of logics without Negative Introspection. For a given formula φ , we construct a new formula that describes an open tableau branch for φ . To that end, we introduce exponentially many new propositional variables, each a directed graph of subformulas of φ and their negations, with edges that describe the tableau productions. We then describe in the μ -calculus a propositionally open branch that is closed under the tableau rules, and then that the branch is open.

Let φ be a formula where all fixed-point variables are distinct, and let mV be the set of least-fixed-point variables in φ . Let \mathbf{G} be the set of labelled directed graphs of the form $g = (\Phi, (\overset{X}{\rightarrow})_{X \in mV}, \ell)$, where

- $\Phi \subseteq \text{sub}(\varphi)$ is its set of vertices,
- each edge is labelled with a variable from mV , and
- each node ψ is additionally labelled with a set $\ell(\psi) \subseteq \{in, out\} \times \{top, bot\}$.

We use the notation Φ_g , $\overset{X}{\rightarrow}_g$, and ℓ_g for Φ , $\overset{X}{\rightarrow}$, and ℓ , respectively, when $g = (\Phi, (\overset{X}{\rightarrow})_{X \in mV}, \ell)$. We also abuse notation and write that $(\psi, u, v) \in g$ to mean that $\psi \in \Phi_g$ and $(u, v) \in \ell_g(\psi)$.

We use graphs from \mathbf{G} to represent a part of a tableau branch that is prefixed with a single tableau prefix. Intuitively, we want a $g \in \mathbf{G}$ to represent a tableau prefix σ , such that Φ_g is the set of formulas prefixed with σ ; $\overset{X}{\rightarrow}_g$ represents the relation $\overset{X}{\rightarrow}$ as defined in Section 4, restricted in the σ -prefixed part of the branch; and $(in, bot) \in \ell_g(\psi)$ (*resp.* $(out, bot) \in \ell_g(\psi)$) represents that ψ may be affected by (*resp.* may affect) a subsequent prefix in the branch through some rule. This intuition is reflected in the following definition, where we describe the graphs that are suitable to represent prefixes of a complete branch.

Definition 6.1. We say that $g = (\Phi, (\overset{X}{\rightarrow})_{X \in mV}, \ell) \in \mathbf{G}$ is *locally compatible with the tableau rules* and write $com(g)$ when for every $\psi \in \Phi$ and $X \in mV$, where $\psi \neq Y$ for any $X < Y$, the following conditions are satisfied. For every pair σ, σ' of possible tableau prefixes and $\psi' \in \text{sub}(\varphi)$:

- (1) if $\frac{\sigma \ \psi}{\sigma \ \psi'}$ is a tableau rule, then $\psi' \in \Phi$ and $\psi \overset{X}{\rightarrow} \psi'$;

- (2) if $\frac{\sigma \psi}{\sigma \psi_1 \mid \sigma \psi_2}$ is a tableau rule, then $\psi_1 \in \Phi$, $\psi \xrightarrow{X} \psi_1$, $\psi_2 \in \Phi$, and $\psi \xrightarrow{X} \psi_2$;
- (3) if $\frac{\sigma \psi}{\sigma \psi_1 \mid \sigma \psi_2}$ is a tableau rule, then $\psi_1 \in \Phi$ and $\psi \xrightarrow{X} \psi_1$, or $\psi_2 \in \Phi$ and $\psi \xrightarrow{X} \psi_2$;
- (4) if $\frac{\sigma \psi}{\sigma' \psi'}$ is a tableau rule and σ' is a strict extension of σ , then $(out, bot) \in \ell(\psi)$;
- (5) if $\frac{\sigma \psi}{\sigma' \psi'}$ is a tableau rule and σ is a strict extension of σ' , then $(out, top) \in \ell(\psi)$; and
- (6) $\neg\psi \notin \Phi$ and $\mathbf{ff} \notin \Phi$.

Let $\mathbf{G}^t = \{g \in \mathbf{G} \mid com(g)\}$.

Definition 6.2. Let $\alpha \in \mathbf{AG}$, $\chi \in \mathbf{sub}(\varphi)$, and $g = (\Phi_g, (\xrightarrow{X}_g)_{X \in mV}, \ell_g)$ and $h = (\Phi_h, (\xrightarrow{X}_h)_{X \in mV}, \ell_h) \in \mathbf{G}^t$. We say that g is a prospective $\alpha\langle\chi\rangle$ -child of h , and write $c_{\alpha\langle\chi\rangle}(h, g)$ when:

- (1) $\langle\alpha\rangle\chi \in \Phi_h$, or $[\alpha]\chi \in \Phi_h$ if (d) is a rule, and $\chi \in \Phi_g$ and $(in, top) \in \ell_g(\chi)$;
- (2) if $[\alpha]\psi \in \Phi_h$ and $\frac{\sigma [\alpha]\psi}{\sigma.\alpha\langle\psi_1\rangle \psi_2}$ is a tableau rule, then $\psi_2 \in \Phi_g$ and $(in, top) \in \ell_g(\psi_2)$; and
- (3) if $[\alpha]\psi \in \Phi_g$ and $\frac{\sigma.\alpha\langle\psi_1\rangle [\alpha]\psi}{\sigma \psi_2}$ is a tableau rule, then $\psi_2 \in \Phi_h$ and $(in, bot) \in \ell_h(\psi_2)$.

We can similarly define that g is a prospective α -child of h , and write $c_\alpha(h, g)$, by omitting the first condition.

Remark 6.3. We note that if one applies the conditions of Definition 6.1 for $com(g)$ to a set of tableau formulas with a common prefix, they do not suffice to declare that the tableau rules have been applied correctly. As such, $com(g)$ ensures that if g represents a prefix σ , then Φ_g is closed under the first line of Table 3, and possibly rule (t). The labeling ℓ_g marks the formulas that interact with other prefixes in the tableau – the immediate extensions or prefixes of σ . Similarly, in the context of $com(g)$ and $com(h)$, $c_{\alpha\langle\chi\rangle}(h, g)$ ensures that if h represents a tableau prefix σ , then the branch can be extended in a way such that $\sigma\alpha\langle\chi\rangle$ is represented by g . If one views a tableau branch as the tree of its prefixes, labelled with sets of formulas, then the collection of pairs h, g , such that $c_{\alpha\langle\chi\rangle}(h, g)$, gives us the allowed $\alpha\langle\chi\rangle$ -children from a node labelled with Φ_h .

Naturally, these predicates do not account for the conditions for applying the box tableau rules, such as (B) and (4). For that, we use the formula **rules**, introduced below (see also Lemma 6.6).

Observe that the size of \mathbf{G} is at most exponential with respect to φ and that it takes at most polynomial time to verify $com(g)$, $c_{\alpha\langle\chi\rangle}(h, g)$, and $c_\alpha(h, g)$.

We now start to describe a tableau branch in $\mathbf{K}_{|\mathbf{sub}(\varphi)| \cdot |\mathbf{AG}|}^\mu$. We construct formulas that use each $g \in \mathbf{G}^t$ as a propositional variable and each $\alpha\langle\psi\rangle$, where $\alpha \in \mathbf{AG}$ and $\psi \in \mathbf{sub}(\varphi)$, as an agent. We use the notation $\mathbf{AG}_\varphi = \{\alpha\langle\psi\rangle \mid \alpha \in \mathbf{AG} \text{ and } \psi \in \mathbf{sub}(\varphi)\}$, $[\alpha]\psi$ as a

shorthand for $\bigwedge_{\chi \in \text{sub}(\varphi)} [\alpha \langle \chi \rangle] \psi$, and dually, $\langle \alpha \rangle \psi$ as a shorthand for $\bigvee_{\chi \in \text{sub}(\varphi)} \langle \alpha \langle \chi \rangle \rangle \psi$. Let

$$\mathbf{rules} := \text{Inv} \left(\begin{array}{c} \bigwedge_{\substack{g \in G^t \\ \langle \alpha \rangle \psi \in \Phi_g}} \left(g \rightarrow \langle \alpha \langle \psi \rangle \rangle \bigvee_{c_{\alpha \langle \psi \rangle}(g,h)} h \right) \wedge \\ \bigvee_{g \in G^t} g \wedge \bigwedge_{\substack{g,h \in G^t \\ g \neq h}} \neg(g \wedge h) \wedge \bigwedge_{\substack{g \in G^t \\ [\alpha] \psi \in \Phi_g \\ \text{(d) is a rule}}} \left(g \rightarrow \langle \alpha \langle \psi \rangle \rangle \bigvee_{c_{\alpha \langle \psi \rangle}(g,h)} h \right) \wedge \\ \bigwedge_{\substack{g \in G^t \\ \alpha \in \text{AG}}} \left(g \rightarrow [\alpha] \bigvee_{c_{\alpha}(g,h)} h \right) \end{array} \right)$$

Formula **rules** ensures that a model behaves in a way that its states correspond to prefixes in a maximal branch that is not propositionally closed. We now make explicit the correspondence between the graphs in G^t and the prefixes in a branch.

Let b be a tableau branch with dependence relations \xrightarrow{X} for each $X \in mV$, and σ a prefix in b . We define $g_b(\sigma) = (\Phi, (\xrightarrow{X}_g)_{X \in mV}, \ell)$, where

- $\Phi = \{\psi \mid \sigma \psi \in b\}$;
- for each $X \in mV$, $\xrightarrow{X}_g = \{(\psi_1, \psi_2) \in \Phi^2 \mid (\sigma \psi_1, \sigma \psi_2) \in \xrightarrow{X}\}$; and
- for each $\psi \in \Phi$, $\ell(\psi)$ is such that:

$(in, top) \in \ell(\psi)$ (*resp.* $(in, bot) \in \ell(\psi)$) iff $\sigma' \psi' \xrightarrow{X} \sigma \psi$ for some $\sigma' \psi' \in b$, where σ' is a strict prefix of σ (*resp.* σ is a strict prefix of σ'); and

$(out, top) \in \ell(\psi)$ (*resp.* $(out, bot) \in \ell(\psi)$) iff $\sigma \psi \xrightarrow{X} \sigma' \psi'$ for some $\sigma' \psi' \in b$, where σ' is a strict prefix of σ (*resp.* σ is a strict prefix of σ').

We may omit b from $g_b(\sigma)$ and write $g(\sigma)$ instead, if b is clear from the context.

Definition 6.4. Let b be a tableau branch, $\mathcal{M} = (W, R, V)$ be a model, using \mathbf{G}^t as the set of propositional variables and AG_φ as the set of agents, and $s \in W$. We say that \mathcal{M}, s represent b when there is a mapping $f : \sigma(b) \rightarrow W$, such that:

- (1) $f(\epsilon) = s$;
- (2) for every $\sigma \in \sigma(b)$, $g_b(\sigma) \in \mathbf{G}^t$ and $(\mathcal{M}, f(\sigma)) \models g_b(\sigma)$;
- (3) for every $\sigma \in \sigma(b)$ and $g \in \mathbf{G}^t$, if $(\mathcal{M}, f(\sigma)) \models g$, then $g = g_b(\sigma)$;
- (4) for every $\sigma.\alpha \langle \psi \rangle \in \sigma(b)$, $f(\sigma) R_{\alpha \langle \psi \rangle} f(\sigma.\alpha \langle \psi \rangle)$; and
- (5) for every $\sigma \in \sigma(b)$ and $t \in W$, $f(\sigma) R_{\alpha \langle \psi \rangle} t$ implies that $\sigma.\alpha \langle \psi \rangle \in \sigma(b)$ and $f(\sigma.\alpha \langle \psi \rangle) = t$.

If the above conditions are satisfied, then we use the notation $f : b \xrightarrow{r} \mathcal{M}, s$.

Lemma 6.5. For every model $\mathcal{M} = (W, R, V)$ and state $s \in W$:

- (1) $(\mathcal{M}, s) \models \mathbf{rules}$ if and only if \mathcal{M}, s represent some branch that is maximal and not propositionally closed; and
- (2) if $(\mathcal{M}, s) \models \mathbf{rules}$ and \mathcal{M}, s represent b , then b is maximal and not propositionally closed.

Proof. We first prove the second statement of the lemma. Let b be represented by $(\mathcal{M}, s) \models \mathbf{rules}$. As every prefix is mapped to a state that is reachable from s (item 4 of Definition

6.4), because of item 2, b is locally maximal. We further observe that item 4 of Definition 6.4 and the conjunct

$$\bigwedge_{g \in \mathbf{G}^t} g \rightarrow [\alpha] \bigvee_{c_{\alpha}(g,h)} h$$

in **rules** ensures that b is closed under all box rules except possibly for (d); and item 5 of Definition 6.4 and the conjuncts

$$\bigwedge_{\substack{g \in \mathbf{G}^t \\ \langle \alpha \rangle \psi \in \Phi_g}} g \rightarrow \langle \alpha \rangle \bigvee_{c_{\alpha\langle \psi \rangle}(g,h)} h \wedge \bigwedge_{\substack{g \in \mathbf{G}^t \\ [\alpha] \psi \in \Phi_g \\ \text{(d) is a rule}}} g \rightarrow \langle \alpha \rangle \langle \psi \rangle \bigvee_{c_{\alpha\langle \psi \rangle}(g,h)} h$$

in **rules** ensures that b is closed under all diamond rules and (d), if it is a rule of the logic for the tableau. Therefore, b is maximal and not propositionally closed.

We now prove the first statement of the lemma. If $(\mathcal{M}, s) \models \mathbf{rules}$, then we demonstrate how to construct a branch b that is represented by \mathcal{M}, s , together with the mapping f of the prefixes to the states that are reachable from s , as in Definition 6.4. As we have already proven the second part of the lemma, b is maximal and not propositionally closed. Note that at every state t that is reachable from s , exactly one variable from \mathbf{G}^t is true, and we call it $g(t)$. The construction of b is by straightforward recursion:

- The branch b includes the prefix ε , $f(\varepsilon) = s$, and $g(\varepsilon) = g(s)$.
- For every $\sigma \in \sigma(b)$ and $\langle \alpha \rangle \psi \in \Phi_{g(f(\sigma))}$, $\sigma.\alpha\langle \psi \rangle$ is a prefix in b ; from the conjunct

$$\bigwedge_{\substack{g \in \mathbf{G}^t \\ \langle \alpha \rangle \psi \in \Phi_g}} g \rightarrow \langle \alpha \rangle \bigvee_{c_{\alpha\langle \psi \rangle}(g,h)} h$$

in **rules**, there is at least one state t , such that $f(\sigma)R_{\alpha}t$ and $c_{\alpha\langle \psi \rangle}(g(f(\sigma)), g(t))$. Let $f(\sigma.\alpha\langle \psi \rangle) = t$ for one of these states t , and $g(\sigma.\alpha\langle \psi \rangle) = g(t)$.

- For every $\sigma \in \sigma(b)$ and $[\alpha]\psi \in \Phi_{g(f(\sigma))}$, if (d) is a tableau rule, then $\sigma.\alpha\langle \psi \rangle$ is a prefix in b ; from the conjunct

$$\bigwedge_{\substack{g \in \mathbf{G}^t \\ [\alpha]\psi \in \Phi_g \\ \text{(d) is a rule}}} g \rightarrow \langle \alpha \rangle \langle \psi \rangle \bigvee_{c_{\alpha\langle \psi \rangle}(g,h)} h$$

in **rules**, there is at least one state t , such that $f(\sigma)R_{\alpha}t$ and $c_{\alpha\langle \psi \rangle}(g(f(\sigma)), g(t))$. Let $f(\sigma.\alpha\langle \psi \rangle) = t$ for one of these states t , and $g(\sigma.\alpha\langle \psi \rangle) = g(t)$.

It is now straightforward to verify that \mathcal{M}, s represent b .

We are now left to prove that if \mathcal{M}, s represent some branch b that is maximal and not propositionally closed, then $(\mathcal{M}, s) \models \mathbf{rules}$. First observe that for every prefix $\sigma \in \sigma(b)$, $g(\sigma) \in \mathbf{G}^t$. Together with items 2 and 5 of Definition 6.4, this means that $(\mathcal{M}, s) \models \text{Inv} \left(\bigvee_{g \in \mathbf{G}^t} g \right)$. Item 3 of Definition 6.4 yields that

$$(\mathcal{M}, s) \models \text{Inv} \left(\bigwedge_{\substack{g, h \in \mathbf{G}^t \\ g \neq h}} \neg(g \wedge h) \right);$$

while items 4 and 5 yield that

$$(\mathcal{M}, s) \models \text{Inv} \left(\begin{array}{l} \bigwedge_{\substack{g \in \mathbf{G}^t \\ \langle \alpha \rangle \psi \in \Phi_g}} \left(g \rightarrow \langle \alpha \rangle \langle \psi \rangle \quad \bigvee_{c_{\alpha \langle \psi \rangle}(g,h)} h \right) \wedge \\ \bigwedge_{\substack{g \in \mathbf{G}^t \\ [\alpha] \psi \in \Phi_g \\ \text{(d) is a rule}}} \left(g \rightarrow \langle \alpha \rangle \langle \psi \rangle \quad \bigvee_{c_{\alpha \langle \psi \rangle}(g,h)} h \right) \wedge \\ \bigwedge_{\substack{g \in \mathbf{G}^t \\ \alpha \in \text{AG}}} \left(g \rightarrow [\alpha] \quad \bigvee_{c_{\alpha}(g,h)} h \right) \end{array} \right).$$

Since $\text{Inc}(\varphi_1) \wedge \text{Inv}(\varphi_2)$ is logically equivalent to $\text{Inv}(\varphi_1 \wedge \varphi_2)$, it follows that $(\mathcal{M}, s) \models \mathbf{rules}$, which was what we wanted to demonstrate. \square

Lemma 6.6. *For every propositionally open, maximal branch b , there is a pointed model (\mathcal{M}, s) , such that \mathcal{M}, s represent b .*

Proof. We define model $\mathcal{M} = (W, R, V)$, such that $W = \sigma(b)$, $R_{\alpha \langle \psi \rangle} = \{(\sigma, \sigma.\alpha \langle \psi \rangle) \mid (\sigma, \sigma.\alpha \langle \psi \rangle) \in W^2\}$ for each $\alpha \langle \psi \rangle \in \text{AG}_\varphi$, and $V(\sigma) = \{g(\sigma)\}$ for each $\sigma \in W$; and $s = \varepsilon$. It is now straightforward to verify that \mathcal{M}, s represent b with $f : b \xrightarrow{r} \mathcal{M}, s$, such that $f(\sigma) = \sigma$ for every σ . \square

Let $X \in mV$. For a given prefix σ of the tableau, we want to describe that there is no \xrightarrow{X} -path where X appears infinitely often. Infinite \xrightarrow{X} -paths may include infinitely many prefixed formulas, or a cycle with finitely many formulas. From a local view on a tableau prefix, it can be hard to tell whether two paths are part of the same infinite path of concern. Thus, we need to rule out all possible combinations of paths that, combined with paths inside other prefixes, can give us infinitely many occurrences of X .

We need to describe that there is a finite \xrightarrow{X} -path from one formula to another, and we do so by defining recursive predicates on pairs of formulas. For this, it is useful to define additional notation:

- we write $(\psi_1, \psi_2) \in g$ to mean that $(\psi_1, \text{in}, \text{top}) \in g$ and $(\psi_2, \text{out}, \text{top}) \in g$;
- for $(\langle \alpha \rangle \psi_1, \text{out}, \text{bot}), (\psi_2, \text{in}, \text{bot}) \in g$, we define $nx(\langle \alpha \rangle \psi_1, \psi_2) = (\psi_1, [\alpha] \psi_2)$, $nx(\langle \alpha \rangle \psi_1) = \psi_1$, and $\alpha(\langle \alpha \rangle \psi_1, \psi_2) = \alpha(\langle \alpha \rangle \psi_1) = \alpha \langle \psi_1 \rangle$;
- for $([\alpha] \psi_1, \text{out}, \text{bot}), (\psi_2, \text{in}, \text{bot}) \in g$, we define $nx([\alpha] \psi_1, \psi_2) = (\psi_1, [\alpha] \psi_2)$, $nx([\alpha] \psi_1) = \psi_1$, and $\alpha([\alpha] \psi_1, \psi_2) = \alpha([\alpha] \psi_1) = \alpha$.

Intuitively, some of the formulas that we construct search through a model to find all the pieces of a finite path from a formula ψ_1 to ψ_2 that starts and ends in g . Definition 6.7 below defines variations of $(\psi_1, \psi_2) \xrightarrow{g} S$ and $\psi_1 \xrightarrow{g} S, \psi_2$, which mean that the search should continue with all the pairs in S , as illustrated in Figure 5. In the case of $(\psi_1, \psi_2) \xrightarrow{g} S$, the path is part of a path that comes from and returns to a parent prefix; in the case of $\psi_1 \xrightarrow{g} S, \psi_2$, the finite path is part of an infinite path that comes from the parent prefix and continues with an infinite path from ψ_2 to a child prefix.

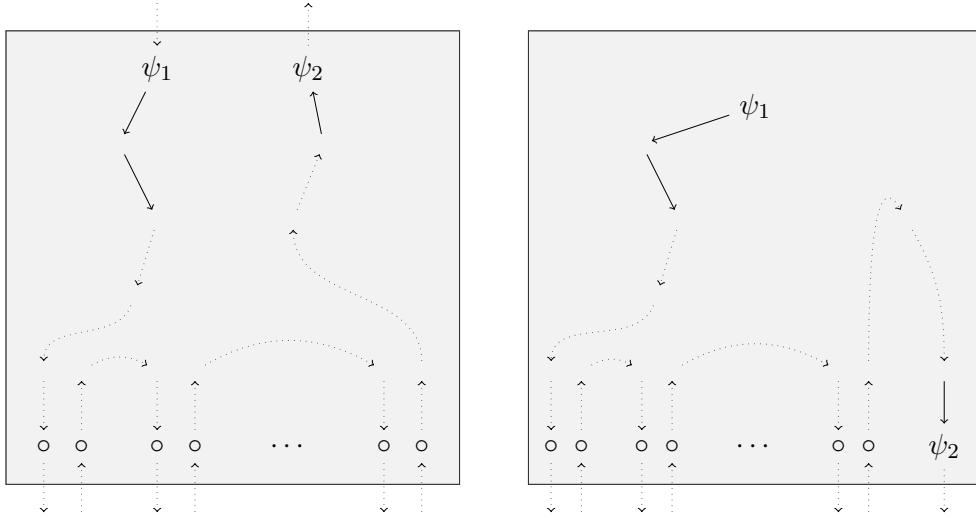


FIGURE 5. A finite \xrightarrow{X} -path from $\sigma \psi_1$ to $\sigma \psi_2$ may visit other tableau prefixes, and an infinite \xrightarrow{X} -path from $\sigma \psi_1$ may include finite segments that visit other prefixes, before continuing with an infinite path from a formula ψ_2 . Each square area represents a tableau prefix, or a graph $g \in G^t$. For S the set of pairs of formulas that appear near the bottom, the left figure illustrates the relation $(\psi_1, \psi_2) \xrightarrow{g} S$ and the right one the relation $\psi_1 \xrightarrow{g} S, \psi_2$ (Definition 6.7).

Definition 6.7. Given a graph $g = (\Phi, (\xrightarrow{X})_{X \in mV}, \ell) \in \mathbf{G}$, $T = (\psi_1, \psi_2) \in (\text{sub}(\varphi))^2$, $S \subseteq (\text{sub}(\varphi))^2$, and $X \in mV$, we write $T \xrightarrow{g, X} S$ to mean that

- (1) $\psi_1, \psi_2 \in \Phi$, $(in, top) \in \ell(\psi_1)$, $(out, top) \in \ell(\psi_2)$,
- (2) for each $(\psi'_1, \psi'_2) \in S$, $\psi'_1, \psi'_2 \in \Phi$, $(out, bot) \in \ell(\psi'_1)$, $(in, bot) \in \ell(\psi'_2)$, and
- (3) $\{\psi'_1, \psi'_2, X \mid (\psi'_1, \psi'_2) \in S\}$ are on a path from ψ_1 to ψ_2 in $(\Phi, \xrightarrow{X} \cup \{(\psi_2, \psi_1)\} \cup S)$.

We write $\emptyset \xrightarrow{g, X} S$ to mean that Condition 2 above is satisfied, and $\{\psi'_1, \psi'_2, X \mid (\psi'_1, \psi'_2) \in S\}$ is part of a cycle in $(\Phi, \xrightarrow{X} \cup S)$. We can similarly define $T \xrightarrow{g} S$ and $\emptyset \xrightarrow{g} S$ by not requiring that X is part of the path.

We write $\psi_1 \xrightarrow{g, X} S, \psi_2$ to mean that

- (1) $\psi_1, \psi_2 \in \Phi$, $(out, bot) \in \ell(\psi_2)$,
- (2) for each $(\psi'_1, \psi'_2) \in S$, $\psi'_1, \psi'_2 \in \Phi$, $(out, bot) \in \ell(\psi'_1)$, $(in, bot) \in \ell(\psi'_2)$, and
- (3) $\{\psi'_1, \psi'_2, X \mid (\psi'_1, \psi'_2) \in S\}$ are on a path from ψ_1 to ψ_2 in $(\Phi, \xrightarrow{X} \cup \{(\psi_2, \psi_1)\} \cup S)$.

We can similarly define $\psi_1 \xrightarrow{g} S, \psi_2$ by not requiring that X is part of path.

To describe that there are no \xrightarrow{X} -paths where X appears infinitely often, we first describe that *there exists* some \xrightarrow{X} -path where X appears infinitely often. We do so, by defining the following formulas, recursively on two parameters ϖ and ϖ_X , that are *strings from* $\{\emptyset\} \cup \text{sub}(\varphi)^2$ or from $\text{sub}(\varphi)$, i.e. $(\{\emptyset\} \cup \text{sub}(\varphi)^2)^* \cup (\text{sub}(\varphi))^*$. These strings keep track of the path from the root of the syntax tree of the constructed formula, to the current

subformula, allowing the construction to detect when to use a recursive call. For $T \in \text{sub}(\varphi)^2$ or $T = \emptyset$:

$$\begin{aligned} \mathbf{fp}^X(T, \varpi) &:= \begin{cases} \bigvee_{\emptyset \xrightarrow{g,X} \emptyset} g \vee \bigvee_{\emptyset \xrightarrow{g,X} \emptyset} (g \wedge \mathbf{fp}^X(\emptyset, \varepsilon, g)) & \text{if } T = \emptyset \\ Z_T^X & \text{if } T \text{ appears in } \varpi \\ \mu Z_T^X. \bigvee_{\substack{T \in g \\ T \xrightarrow{g,X} \emptyset}} g \vee \bigvee_{\substack{T \in g \\ T \xrightarrow{g,X} \emptyset}} (g \wedge \mathbf{fp}^X(T, \varpi, g)), & \text{otherwise} \end{cases} \\ \mathbf{fp}(T, \varpi) &:= \begin{cases} Z_T & \text{if } T \text{ appears in } \varpi \\ \mu Z_T. \bigvee_{\substack{T \in g \\ T \xrightarrow{g} \emptyset}} g \vee \bigvee_{\substack{T \in g \\ T \xrightarrow{g} \emptyset}} (g \wedge \mathbf{fp}(T, \varpi, g)), & \text{otherwise} \end{cases} \\ \mathbf{fp}^X(T, \varpi, g) &:= \bigvee_{\substack{S \subseteq \Phi_g^2 \\ T \xrightarrow{g} S}} \bigvee_{s \in S} \langle \alpha(s) \rangle \mathbf{fp}^X(nx(s), \varpi T) \wedge \bigwedge_{s \in S} \langle \alpha(s) \rangle \mathbf{fp}(nx(s), \varepsilon) \\ &\quad \vee \bigvee_{\substack{S \subseteq \Phi_g^2 \\ T \xrightarrow{g,X} S}} \bigwedge_{s \in S} \langle \alpha(s) \rangle \mathbf{fp}(nx(s), \varepsilon) \\ \mathbf{fp}(T, \varpi, g) &:= \bigvee_{\substack{S \subseteq \Phi_g^2 \\ T \xrightarrow{g} S}} \bigwedge_{s \in S} \langle \alpha(s) \rangle \mathbf{fp}(nx(s), \varpi T) \end{aligned}$$

The formula $\mathbf{fp}((\psi_1, \psi_2), \varpi)$ (*resp.* $\mathbf{fp}^X((\psi_1, \psi_2), \varpi)$) describes that there is a finite \xrightarrow{X} -path from ψ_1 to ψ_2 that does not visit prefixes smaller than the current one (*resp.* where X appears). Formula $\mathbf{fp}(\emptyset, \varpi)$ describes that there is a finite cycle that does not visit prefixes smaller than the current one. We note an asymmetry between $\alpha(\langle \alpha \rangle \psi_1, \psi_2)$ and $\alpha([\alpha] \psi_1, \psi_2)$, used in the above formulas. That is the case, because in a tableau branch, $\sigma [\alpha] \psi$ has a dependency edge to all $\sigma. \alpha \langle \psi' \rangle \psi$, but $\sigma \langle \alpha \rangle \psi$ only has a dependency edge to $\sigma. \alpha \langle \psi \rangle \psi$.

It is also worth noting that the way these formulas are constructed, we do not need to consider any strings ϖ or ϖ_X where the same element appears twice. Therefore, in effect, ϖ and ϖ_X are permutations of subsets of $\{\emptyset\} \cup \text{sub}(\varphi)^2$ and $\text{sub}(\varphi)$.

Lemma 6.8. *For every $T, T' \in \text{sub}(\varphi)^2$,*

$$\llbracket \mathbf{fp}^X(T, \varepsilon) \rrbracket = \llbracket \mathbf{fp}^X(T, T') [\mathbf{fp}^X(T', \varepsilon) / Z_{T'}^X] \rrbracket = \llbracket \mathbf{fp}^X(T, T'), \rho [Z_{T'}^X \mapsto \llbracket \mathbf{fp}^X(T', \varepsilon) \rrbracket] \rrbracket.$$

Proof. Specifically, we prove the stronger statement, that for any string ϖ where T' does not appear,

$$\llbracket \mathbf{fp}^X(T, \varpi), \rho \rrbracket = \llbracket \mathbf{fp}^X(T, T' \varpi) [\mathbf{fp}^X(T', \varpi) / Z_{T'}^X], \rho \rrbracket$$

and

$$\llbracket \mathbf{fp}^X(T, T' \varpi) [\mathbf{fp}^X(T', \varpi) / Z_{T'}^X], \rho \rrbracket = \llbracket \mathbf{fp}^X(T, T' \varpi), \rho [Z_{T'}^X \mapsto \llbracket \mathbf{fp}^X(T', \varpi) \rrbracket], \rho \rrbracket.$$

The second equality is a straightforward property of fixed-point operators. We prove the first one by induction on $\mathbf{fp}^X(T, T' \varpi)$.

If $\text{fp}^X(T, T'\varpi) = Z_T^X$: then T appears in $T'\varpi$. If $T = T'$, then

$$\text{fp}^X(T, \varpi) = \text{fp}^X(T', \varpi) = Z_T^X[\text{fp}^X(T', \varpi)/Z_{T'}^X] = \text{fp}^X(T, T'\varpi)[\text{fp}^X(T', \varpi)/Z_{T'}^X],$$

yielding the required semantic equality. On the other hand, if $T \neq T'$, then T appears in ϖ , and therefore $\text{fp}^X(T, \varpi) = \text{fp}^X(T, T'\varpi)[\text{fp}^X(T', \varpi)/Z_{T'}^X] = Z_T^X$.

Otherwise: the inductive step is straightforward from the definition of the formulas. \square

The following Definition 6.9 defines the predicates $\text{cyc}((\psi_1, \psi_2), \sigma)$ and $\text{cyc}_X((\psi_1, \psi_2), \sigma)$, which are the counterparts of $\text{fp}(T, \varpi)$ and $\text{fp}^X(T, \varpi)$ for tableau branches.

Definition 6.9. Let $X \in mV$, $\psi_1, \psi_2 \in \text{sub}(\varphi)$, $f : b \xrightarrow{r} \mathcal{M}, s$, and $\sigma \in \sigma(b)$. We write $\text{cyc}((\psi_1, \psi_2), \sigma)$ to denote that the following hold:

- (1) $\sigma \psi_1, \sigma \psi_2 \in b$,
- (2) there is a finite \xrightarrow{X} -path in b , from $\sigma \psi_1$ to $\sigma \psi_2$, which only goes through prefixes that extend σ .

We write $\text{cyc}_X((\psi_1, \psi_2), \sigma)$ to denote that the same conditions as for $\text{cyc}((\psi_1, \psi_2), \sigma)$ are satisfied, with the addition that X appears on the finite \xrightarrow{X} -path from $\sigma \psi_1$ to $\sigma \psi_2$.

Lemma 6.10. Let $X \in mV$, $\psi_1, \psi_2 \in \text{sub}(\varphi)$, $f : b \xrightarrow{r} \mathcal{M}, s$, and $\sigma \in \sigma(b)$.

- (1) $(\mathcal{M}, f(\sigma)) \models \text{fp}((\psi_1, \psi_2), \varepsilon)$ if and only if $\text{cyc}((\psi_1, \psi_2), \sigma)$ and $(\psi_1, \psi_2) \in f(\sigma)$,
- (2) $(\mathcal{M}, f(\sigma)) \models \text{fp}^X((\psi_1, \psi_2), \varepsilon)$ if and only if $\text{cyc}_X((\psi_1, \psi_2), \sigma)$ and $(\psi_1, \psi_2) \in f(\sigma)$,
- (3) $(\mathcal{M}, f(\sigma)) \models \text{fp}(\emptyset, \varepsilon)$ if and only if $\text{cyc}((\psi, \psi), \sigma)$ for some $\psi \in \Phi_{f(\sigma)}$, and
- (4) $(\mathcal{M}, f(\sigma)) \models \text{fp}^X(\emptyset, \varepsilon)$ if and only if $\text{cyc}_X((\psi, \psi), \sigma)$ for some $\psi \in \Phi_{f(\sigma)}$.

Proof. We use the first biimplication of the lemma to prove the second one. All other cases can be proven with similar reasoning. We first observe that $(\mathcal{M}, f(\sigma)) \models \text{fp}^X((\psi_1, \psi_2), \varepsilon)$ implies $(\psi_1, \psi_2) \in f(\sigma)$, and prove that $(\mathcal{M}, f(\sigma)) \models \text{fp}^X((\psi_1, \psi_2), \varepsilon)$ implies $\text{cyc}_X((\psi_1, \psi_2), \sigma)$.

For each T , let $C_{Z_T^X} = \{f(\sigma') \mid \sigma' \in \sigma(b) \text{ and } \text{cyc}_X(T, \sigma')\}$. Let $T = (\psi_1, \psi_2)$. To prove the implication, we proceed to prove the more general statement that $\neg \text{cyc}_X(T, \sigma)$ implies that $(\mathcal{M}, f(\sigma)) \not\models_\rho \text{fp}^X(T, \varpi)$ for every $\varpi \in (\text{sub}(\varphi)^2)^*$ and environment ρ , such that $\rho(Z_{T'}^X) = C_{Z_{T'}^X}$ for each $Z_{T'}^X$. We assume that $\neg \text{cyc}_X(T, \sigma)$ and proceed by induction on the structure of $\text{fp}^X(T, \varpi)$. **If T appears in ϖ** or, equivalently, $\text{fp}^X(T, \varpi) = Z_T^X$, by the definition of ρ , $f(\sigma) \notin \rho(Z_T^X)$, so $(\mathcal{M}, f(\sigma)) \not\models_\rho \text{fp}^X(T, \varpi)$. **We now assume that T does not appear in ϖ** If Condition 1 for $\text{cyc}_X(T, \sigma)$ does not hold, then it is evident from the definitions of $T \xrightarrow{g} S$, $T \xrightarrow{g, X} S$, and $T \xrightarrow{g, X} \emptyset$ that $(\mathcal{M}, f(\sigma)) \not\models \text{fp}^X(T, \varpi)$, and we are done. We now assume that Condition 1 holds, but there is no finite \xrightarrow{X} -path in b , from $\sigma \psi_1$ to $\sigma \psi_2$, that only visits extensions of σ , where X appears. From the fixed-point semantics of Table 1, it suffices to prove two statements:

- (1) for every $\sigma' \in \sigma(b)$, if

$$(\mathcal{M}, f(\sigma')) \models \bigvee_{\substack{T \in g \\ T \xrightarrow{g, X} \emptyset}} g \vee \bigvee_{\substack{T \in g \\ T \xrightarrow{g, X} \emptyset}} (g \wedge \text{fp}^X(T, \varpi, g)),$$

then $f(\sigma') \in C_{Z_T^X}$; and

- (2) $f(\sigma) \notin C_{Z_T^X}$.

The second statement is immediate from our assumption that $\neg cyc_X(T, \sigma)$. We now prove the first statement. Clearly, if $(\mathcal{M}, f(\sigma')) \models \bigvee_{T \in g} g$, we have $cyc_X(T, \sigma')$. Otherwise, we have

$$T \xrightarrow{g, X} \emptyset$$

that there is some $g \in G^t$, such that $T \in g$, $(\mathcal{M}, f(\sigma')) \models g$, and $(\mathcal{M}, f(\sigma')) \models \text{fp}^X(T, \varpi, g)$. Therefore, there is some $S \subseteq \text{sub}(\varphi)^2$, such that either

$T \xrightarrow{g, X} S$: and for each $s \in S$, there is some $\varsigma = \alpha(s)$ or $\varsigma = \alpha(s)\langle\psi\rangle$, such that $\sigma'.\varsigma \in \sigma(b)$, $f(\sigma')R_\varsigma f(\sigma'.\varsigma)$ and $(\mathcal{M}, f(\sigma'.\varsigma)) \models \text{fp}(nx(s), \varepsilon)$, or

$T \xrightarrow{g} S$: and for each $s \in S$, there is some $\varsigma = \alpha(s)$ or $\varsigma = \alpha(s)\langle\psi\rangle$, such that $\sigma'.\varsigma \in \sigma(b)$, $f(\sigma')R_\varsigma f(\sigma'.\varsigma)$ and $(\mathcal{M}, f(\sigma'.\varsigma)) \models \text{fp}(nx(s), \varepsilon)$ and for some $s \in S$, there is some $\varsigma = \alpha(s)$ or $\varsigma = \alpha(s)\langle\psi\rangle$, such that $\sigma'.\varsigma \in \sigma(b)$, $f(\sigma')R_\varsigma f(\sigma'.\varsigma)$ and $(\mathcal{M}, f(\sigma'.\varsigma)) \models \text{fp}^X(nx(s), \varpi T)$.

In the first case, for each $s = (\chi_1, \chi_2) \in S$, where $nx(s) = (\chi'_1, \chi'_2)$, by the first item of the lemma, there is a finite \xrightarrow{X} -path in b , from $\sigma'.\varsigma \chi'_1$ to $\sigma'.\varsigma \chi'_2$, that only visits extensions of $\sigma'.\varsigma$. We know that $\chi'_2 = [\alpha]\chi''_2$, and either $\chi_1 = [\alpha]\chi'_1$ or $\chi_1 = \langle\alpha\rangle\chi'_1$. By the conditions of \xrightarrow{X} for box formulas, $\sigma'.\varsigma \chi'_2 \xrightarrow{X} \sigma' \chi_2$, and if $\chi_1 = [\alpha]\chi'_1$, then $\sigma' \chi_1 \xrightarrow{X} \sigma'.\varsigma \chi'_1$. If $\chi_1 = \langle\alpha\rangle\chi'_1$, then we know that $(\varsigma =) \alpha(s) = \alpha\langle\chi'_1\rangle$, and therefore $\frac{\sigma' \chi_1}{\sigma'.\alpha\langle\chi'_1\rangle \chi'_1}$ is a tableau rule, and by the conditions of \xrightarrow{X} , $\sigma' \chi_1 \xrightarrow{X} \sigma'.\alpha(s) \chi'_1$. Therefore, there is a finite \xrightarrow{X} -path in b , from $\sigma'.\chi_1$ to $\sigma'.\chi_2$ that only visits extensions of σ' , where σX appears.

In the second case, we can similarly conclude that there is a finite \xrightarrow{X} -path in b , from $\sigma'.\chi_1$ to $\sigma'.\chi_2$ that only visits extensions of σ' . From $f(\sigma')R_\varsigma f(\sigma'.\varsigma)$ and $(\mathcal{M}, f(\sigma'.\varsigma)) \models \text{fp}^X(nx(s), \varpi T)$ and the inductive hypothesis, we see that some $\sigma'' X$ occurs on that path. Therefore, both conditions for $cyc_X(T, \sigma')$ are satisfied and $f(\sigma') \in C_{Z_T^X}$, completing the proof of the first statement and the induction.

We now prove that $cyc_X((\psi_1, \psi_2), \sigma)$ implies $(\mathcal{M}, f(\sigma)) \models \text{fp}^X((\psi_1, \psi_2), \varepsilon)$. By Condition 2 from $cyc_X((\psi_1, \psi_2), \sigma)$, there is a finite \xrightarrow{X} -path from ψ_1 to ψ_2 that only visits prefixes that extend σ , including σ . Let k be the length of the longest σ' , such that the path goes through $\sigma.\sigma'$. We proceed by induction on k . Let $T = (\psi_1, \psi_2)$. **If** $k = 0$ then the finite path remains in σ , and therefore $(\psi_1, \psi_2) \xrightarrow{g(\sigma), X} \emptyset$. As $(\mathcal{M}, f(\sigma)) \models g(\sigma)$, we conclude that $(\mathcal{M}, f(\sigma)) \models_\rho \text{fp}^X((\psi_1, \psi_2), \varepsilon)$. **If** $k > 0$ we demonstrate that

$$(\mathcal{M}, f(\sigma)) \models \text{fp}^X((\psi_1, \psi_2), \varepsilon).$$

Let p be the finite \xrightarrow{X} -path from ψ_1 to ψ_2 . Since $k > 0$, p will leave σ and return, perhaps multiple times. As p leaves and returns to σ r times, in σ it will visit exactly the formulas $\sigma \chi_1, \sigma \chi_2, \dots, \sigma \chi_{2r} \in b$, in that order. We note that for every $1 \leq i \leq r$, $(out, bot) \in \ell_{g(\sigma)}(\chi_{2i-1})$ and $(in, bot) \in \ell_{g(\sigma)}(\chi_{2i})$. Let $S = \{s_i \mid 1 \leq i \leq r\}$, where $s_i = (\chi_{2i-1}, \chi_{2i})$ for every i ; we see that $(\psi_1, \psi_2) \xrightarrow{g(\sigma)} S$. Furthermore, if p goes through σX , then $(\psi_1, \psi_2) \xrightarrow{g(\sigma), X} S$, and otherwise p goes through some $\sigma' X$ between some $\sigma \chi_{2j-1}$ and $\sigma \chi_{2j}$. For every $i \leq r$, let $\varsigma_i = \alpha(s_i)$ or $\alpha(s_i)\langle\psi'_i\rangle$, depending on whether χ_{2i-1} is a box or a diamond formula; there are some $\sigma.\varsigma_i \chi'_{2i-1} \in b$ and $\sigma.\varsigma_i \chi'_{2i} \in b$, such that $\sigma \chi_{2i-1} \xrightarrow{X} \sigma.\varsigma_i \chi'_{2i}$ and $\sigma.\varsigma_i \chi'_{2i} \xrightarrow{X} \sigma \chi_{2i}$. Then, the part of p from $\sigma.\varsigma_i \chi'_{2i-1}$ to $\sigma.\varsigma_i \chi'_{2i}$ is a finite \xrightarrow{X} -path that only goes through extensions of $\sigma.\varsigma_i$ that extend the prefix by a length at most $k - 1$. Therefore, $cyc(nx(s_i), \sigma.\varsigma_i)$ and by the inductive hypothesis, $(\mathcal{M}, f(\sigma.\varsigma_i)) \models \text{fp}(nx(s_i), \varepsilon)$,

and if it is not the case that $(\psi_1, \psi_2) \xrightarrow{g(\sigma), X} S$, then $(\mathcal{M}, f(\sigma.\varsigma_j)) \models \mathbf{fp}^X(nx(s_j), \varepsilon)$. Using Lemma 6.8, we see that

$$(\mathcal{M}, f(\sigma.\varsigma_j)) \models \mathbf{fp}^X(nx(s_j), (\psi_1, \psi_2))[\mathbf{fp}^X((\psi_1, \psi_2), \varepsilon)/Z_T^X].$$

From the fact that $f(\sigma)R_{\alpha(s_i)}f(\sigma.\varsigma_i)$, we conclude that

$$(\mathcal{M}, f(\sigma)) \models \mathbf{fp}^X((\psi_1, \psi_2), \varepsilon, g(\sigma))[\mathbf{fp}^X((\psi_1, \psi_2), \varepsilon)/Z_T^X].$$

As $\mathbf{fp}^X((\psi_1, \psi_2), \varepsilon)$ is a fixed-point formula, this yields that

$$(\mathcal{M}, f(\sigma)) \models \mathbf{fp}^X((\psi_1, \psi_2), \varepsilon). \quad \square$$

We now express that there is an infinite path in b , where X appears infinitely often, starting from $\sigma \psi$.

$$\mathbf{ip}^X(\psi, \varpi, \varpi_X) := \begin{cases} Z_\psi^X & \text{if } \psi \text{ appears in } \varpi_X \\ Z_\psi & \text{if } \psi \in \varpi \text{ and } \notin \varpi_X \\ \nu Z_\psi \cdot \mu Z_\psi^X \cdot \bigvee_{\psi \in \Phi_g} g \wedge \mathbf{ip}^X(\psi, \varpi, \varpi_X, g) & \text{otherwise} \end{cases}$$

$$\mathbf{ip}^X(\psi, \varpi, \varpi_X, g) := \bigvee_{\psi \xrightarrow{g, X} S, \psi'} \bigwedge_{s \in S} \langle \alpha(s) \rangle \mathbf{fp}(nx(s), \varepsilon) \wedge \langle \alpha(\psi') \rangle \mathbf{ip}^X(nx(\psi'), \varpi\psi, \varepsilon)$$

$$\bigvee_{\psi \xrightarrow{g} S, \psi'} \left(\begin{array}{c} \left(\bigvee_{s \in S} \langle \alpha(s) \rangle \mathbf{fp}^X(nx(s), \varepsilon) \right) \\ \wedge \\ \left(\bigwedge_{s \in S} \langle \alpha(s) \rangle \mathbf{fp}(nx(s), \varepsilon) \wedge \langle \alpha(\psi') \rangle \mathbf{ip}^X(nx(\psi'), \varpi\psi, \varepsilon) \right) \\ \vee \\ \left(\bigwedge_{s \in S} \langle \alpha(s) \rangle \mathbf{fp}(nx(s), \varepsilon) \wedge \langle \alpha(s) \rangle \mathbf{ip}^X(nx(\psi'), \varpi\psi, \varpi_X\psi) \right) \end{array} \right)$$

$$\mathbf{inf_path}_X := \mathbf{fp}^X(\emptyset, \varepsilon) \vee \bigvee_{\psi \in g} (g \wedge \mathbf{ip}^X(\psi, \varepsilon, \varepsilon))$$

$$\mathbf{inf_path} := \bigvee_{X \in mV} \mathbf{inf_path}_X$$

The formula $\mathbf{ip}^X(\psi, \varpi, \varpi_X, g)$ describes that there is an *infinite* path in the branch that does not visit prefixes smaller than the current one, where X appears infinitely often, according to the intuition in Figure 5. We use the formulas \mathbf{fp} to express that two formulas in the current prefix are connected by a finite path (that does not revisit the current prefix), and we use π_X to keep track of whether we have encountered X along the path, after the latest recursive call of $\mathbf{ip}^X(\psi, \varpi, \varpi_X, g)$. In the definition of $\mathbf{ip}^X(\psi, \varpi, \varpi_X, g)$, if ψ appears in ϖ_X , then we activate the least-fixed-point operator, because ψ appearing in ϖ_X marks that X was not encountered since the last activation of $\mathbf{ip}^X(\psi, -, -, g)$, and otherwise the greatest-fixed-point operator. We do that to ensure that as the formula evaluates along a \xrightarrow{X} -path, X is encountered infinitely often, because if it is not, then only the least-fixed-point operator would activate infinitely often.

Lemma 6.11. *For every $\psi, \psi' \in \text{sub}(\varphi)$ and $\varpi \in (\text{sub}(\varphi))^*$, such that ψ' does not appear in ϖ ,*

$$\begin{aligned} \llbracket \text{ip}^X(\psi, \varpi, \varepsilon) \rrbracket &= \llbracket \text{ip}^X(\psi, \varpi\psi', \psi')[\text{ip}^X(\psi', \varpi, \varepsilon)/Z_{\psi'}^X] \rrbracket \\ &= \llbracket \text{ip}^X(\psi, \varpi\psi', \psi'), \rho[Z_{\psi'}^X \mapsto \llbracket \text{ip}^X(\psi', \varpi, \varepsilon) \rrbracket] \rrbracket. \end{aligned}$$

Proof. The reasoning to prove this lemma is as in the proof of Lemma 6.8. \square

Lemma 6.12. *Let $X \in mV$, $f : b \xrightarrow{r} \mathcal{M}, s$, and $\sigma \in \sigma(b)$. Then $(\mathcal{M}, f(\sigma)) \models \text{ip}^X(\psi, \varepsilon, \varepsilon)$ if and only if*

- (1) $\sigma \psi \in b$, and
- (2) *there is an infinite \xrightarrow{X} -path in b , from $\sigma \psi$, where X appears infinitely often, that only visits prefixes that extend σ .*

Proof. We first assume that $\sigma \psi \in b$, and that there is an infinite \xrightarrow{X} -path in b , from $\sigma \psi$, where X appears infinitely often, that only visits prefixes that extend σ , and prove that $(\mathcal{M}, f(\sigma)) \models \text{ip}^X(\psi, \varepsilon, \varepsilon)$. For every $\psi' \in \text{sub}(\varphi)$, let $IP_{\psi'}$ be the set of prefixes $\sigma' \in \sigma(b)$, such that $\sigma' \psi' \in b$, and that there is an infinite \xrightarrow{X} -path in b , from $\sigma' \psi'$, where X appears infinitely often, that only visits prefixes that extend σ' . To prove the implication, we prove the stronger statement that for every ϖ , $(\mathcal{M}, f(\sigma)) \models_{\rho} \text{ip}^X(\psi, \varpi, \varepsilon)$, where for every $\psi' \in \text{sub}(\varphi)$, $\rho(Z_{\psi'}) = \{f(\sigma') \mid \sigma' \in IP_{\psi'}\}$. We proceed by induction on the structure of the formula.

If ψ appears in ϖ , then $\llbracket \text{ip}^X(\psi, \varpi, \varepsilon), \rho \rrbracket = \llbracket Z_{\psi} \rrbracket = \{f(\sigma') \mid \sigma' \in IP_{\psi}\} \ni f(\sigma)$, and we are done. Otherwise, using the semantics of Table 1, since

$$f(\sigma) \in \{f(\sigma') \mid \sigma' \in IP_{\psi'}\} = \rho(Z_{\psi'}),$$

it suffices to prove that

$$\{f(\sigma') \mid \sigma' \in IP_{\psi'}\} \subseteq \left[\left[\mu Z_{\psi'}^X. \bigvee_{\psi \in \Phi_g} \text{ip}^X(\psi, \varpi, \varepsilon, g), \rho \right] \right].$$

Let $\sigma' \in IP_{\psi'}$. There must be an infinite \xrightarrow{X} -path in b from $\sigma' \psi$ that visits X infinitely often, and avoids strict prefixes of σ . That path may visit certain prefixes more than once. Let $\sigma'.\sigma''$ be minimal, such that the path goes through some $\sigma'.\sigma'' \psi' \in b$ after visiting X at least once after $\sigma' \psi$. In particular, this means that X is visited only in extensions of $\sigma'.\sigma''$. Furthermore, since the path is infinite, we can assume that the path visits a strict extension of $\sigma'.\sigma''$ right after $\sigma'.\sigma'' \psi'$. Let k be the length of σ'' .

We prove that $(\mathcal{M}, f(\sigma')) \models_{\rho} \mu Z_{\psi'}^X. \bigvee_{\psi \in \Phi_g} \text{ip}^X(\psi, \varpi, \varepsilon, g)$ and we do so by induction on k .

Case of $k = 0$: in this case, the path starts from $\sigma' \psi$ and visits $\sigma' \psi'$ after visiting X at least once. Therefore, since we have assumed that the path visits a strict extension of σ' right after $\sigma' \psi'$, we have that for some $S \subseteq \text{sub}^2(\varphi)$, where $\text{cyc}(s, \sigma')$ through a path that only visits extensions of some $\sigma'.a_s$, for each $s \in S$, $\psi \xrightarrow{g, X} S\psi'$, or $\psi \xrightarrow{g} S\psi'$ and $\text{cyc}_X(s_X, \sigma')$ through a path that only visits extensions of $\sigma'.a_{s_X}$, for some $s_X \in S$. For each $s \in S$, notice that $f(\sigma')R_{\alpha(s)}f(\sigma'.a_s)$, and that $\text{cyc}(nx(s), \sigma'.a_s)$ and

$cyc_X(nx(s_X), \sigma'.a_{s_X})$. Therefore, for $g = g(\sigma')$, using Lemma 6.10 and the inductive hypothesis for $ip^X(nx(\psi'), \varpi\psi, \varepsilon)$, either $\psi \xrightarrow{g, X} S\psi'$ and

$$(\mathcal{M}, f(\sigma')) \models_{\rho} \bigwedge_{s \in S} \langle \alpha(s) \rangle \mathbf{fp}(nx(s), \varepsilon) \wedge \langle \alpha(\psi') \rangle \mathbf{ip}^X(nx(\psi'), \varpi\psi, \varepsilon);$$

or $\psi \xrightarrow{g} S\psi'$,

$$\begin{aligned} &(\mathcal{M}, f(\sigma')) \models_{\rho} \langle \alpha(s_X) \rangle \mathbf{fp}^X(nx(s_X), \varepsilon); \text{ and} \\ &(\mathcal{M}, f(\sigma')) \models_{\rho} \bigwedge_{s \in S} \langle \alpha(s) \rangle \mathbf{fp}(nx(s), \varepsilon) \wedge \langle \alpha(\psi') \rangle \mathbf{ip}^X(nx(\psi'), \varpi\psi, \varepsilon). \end{aligned}$$

For both cases, we observe that Z_{ψ}^X is not free in $ip^X(nx(\psi'), \varpi\psi, \varepsilon)$, and therefore either $\psi \xrightarrow{g, X} S\psi'$ and

$$(\mathcal{M}, f(\sigma')) \models_{\rho'} \bigwedge_{s \in S} \langle \alpha(s) \rangle \mathbf{fp}(nx(s), \varepsilon) \wedge \langle \alpha(\psi') \rangle \mathbf{ip}^X(nx(\psi'), \varpi\psi, \varepsilon);$$

or $\psi \xrightarrow{g} S\psi'$,

$$\begin{aligned} &(\mathcal{M}, f(\sigma')) \models_{\rho'} \langle \alpha(s_X) \rangle \mathbf{fp}^X(nx(s_X), \varepsilon); \text{ and} \\ &(\mathcal{M}, f(\sigma')) \models_{\rho'} \bigwedge_{s \in S} \langle \alpha(s) \rangle \mathbf{fp}(nx(s), \varepsilon) \wedge \langle \alpha(\psi') \rangle \mathbf{ip}^X(nx(\psi'), \varpi\psi, \varepsilon), \end{aligned}$$

where $\rho' = \rho \left[Z_{\psi}^X \mapsto \left[\mu Z_{\psi}^X. \bigvee_{\psi \in \Phi_g} \mathbf{ip}^X(\psi, \varpi, \varepsilon, g) \right] \right]$. We conclude therefore that $(\mathcal{M}, f(\sigma')) \models_{\rho'} \bigvee_{\psi \in \Phi_g} \mathbf{ip}^X(\psi, \varpi, \varepsilon, g)$, and using the fact that $\mu Z_{\psi}^X. \bigvee_{\psi \in \Phi_g} \mathbf{ip}^X(\psi, \varpi, \varepsilon, g)$ is a fixed point formula, we conclude that

$$(\mathcal{M}, f(\sigma')) \models_{\rho} \mu Z_{\psi}^X. \bigvee_{\psi \in \Phi_g} \mathbf{ip}^X(\psi, \varpi, \varepsilon, g).$$

Case of $k > 0$: let $\sigma' \psi'$ be the last formula that the path visits in σ' . Therefore, the path visits a strict extension of $\sigma' \psi'$ right after $\sigma' \psi'$, and we have that for some $S \subseteq \mathbf{sub}^2(\varphi)$, where $cyc(s, \sigma')$ through a path that only visits extensions of some $\sigma'.a_s$, for each $s \in S$, $\psi \xrightarrow{g} S\psi'$. For each $s \in S$, notice that $f(\sigma')R_{\alpha(s)}f(\sigma'.a_s)$. Therefore, for $g = g(\sigma')$, using Lemma 6.10 and the inductive hypothesis for k , $\psi \xrightarrow{g} S\psi'$ and

$$(\mathcal{M}, f(\sigma')) \models_{\rho} \bigwedge_{s \in S} \langle \alpha(s) \rangle \mathbf{fp}(nx(s), \varepsilon) \wedge \langle \alpha(\psi') \rangle \mathbf{ip}^X(nx(\psi'), \varpi, \varepsilon).$$

From this, Lemma 6.11 yields that

$$(\mathcal{M}, f(\sigma')) \models_{\rho} \bigwedge_{s \in S} \langle \alpha(s) \rangle \mathbf{fp}(nx(s), \varepsilon) \wedge \langle \alpha(\psi') \rangle \mathbf{ip}^X(nx(\psi'), \varpi\psi, \psi) [\mathbf{ip}^X(nx(\psi), \varpi, \varepsilon) / Z_{\psi}^X].$$

Therefore, $(\mathcal{M}, f(\sigma')) \models_{\rho} \mathbf{ip}^X(\psi, \varpi, \varpi_X, g(\sigma')) [\mathbf{ip}^X(nx(\psi), \varpi, \varepsilon) / Z_{\psi}^X]$, and therefore

$$(\mathcal{M}, f(\sigma')) \models_{\rho} \bigvee_{\psi \in \Phi_g} \mathbf{ip}^X(\psi, \varpi, \varpi_X, g) [\mathbf{ip}^X(nx(\psi), \varpi, \varepsilon) / Z_{\psi}^X].$$

But $\bigvee_{\psi \in \Phi_g} \text{ip}^X(\psi, \varpi, \varpi_X, g)[\text{ip}^X(nx(\psi), \varpi, \varepsilon)/Z_{\psi}^X]$ is an unfolding of $\text{ip}^X(nx(\psi), \varpi, \varepsilon)$, and we conclude that

$$(\mathcal{M}, f(\sigma')) \models_{\rho} \text{ip}^X(nx(\psi), \varpi, \varepsilon).$$

We now assume that $(\mathcal{M}, f(\sigma)) \models \text{ip}^X(\psi, \varepsilon, \varepsilon)$ and prove that

- (1) $\sigma \psi \in b$, and
- (2) there is an infinite \xrightarrow{X} -path in b , from $\sigma \psi$, where X appears infinitely often, that only visits prefixes that extend σ .

It is straightforward to see that for every prefix $\sigma \in \sigma(b)$, $(\mathcal{M}, f(\sigma)) \models \text{ip}^X(\psi, \varpi, \varpi_X)$ implies that $\sigma \psi \in b$.

Let \rightarrow be an lfp-finite dependency relation, as ensured by Theorem 2.13. We demonstrate how to recursively construct an infinite \xrightarrow{X} -path in b , starting from $\sigma \psi$, where X appears infinitely often. Let $p_0 = \sigma \psi = \sigma_0 \psi_0$. For a finite \xrightarrow{X} -path p_i in b that ends in $\sigma_i \psi_i$, such that $(\mathcal{M}, f(\sigma_i)) \models \text{ip}^X(\psi_i, \varpi^i, \varpi_X^i)$, we show how to extend the path to p_{i+1} that ends in $\sigma_{i+1} \psi_{i+1}$, such that $(\mathcal{M}, f(\sigma_{i+1})) \models \text{ip}^X(\psi_{i+1}, \varpi^{i+1}, \varpi_X^{i+1})$, and

$$(\sigma_i, \text{ip}^X(\psi_i, \varpi^i, \varpi_X^i)) \rightarrow^+ (\sigma_{i+1}, \text{ip}^X(\psi_{i+1}, \varpi^{i+1}, \varpi_X^{i+1})).$$

By observing the definition of $\text{ip}^X(\psi_i, \varpi^i, \varpi_X^i)$, and using Lemma 6.10, it is not hard to observe that

$$(\sigma_i, \text{ip}^X(\psi_i, \varpi^i, \varpi_X^i)) \rightarrow^+ (\sigma_{i+1}, \text{ip}^X(\psi_{i+1}, \varpi^{i+1}, \varpi_X^{i+1}))$$

for some $(\sigma_{i+1}, \text{ip}^X(\psi_{i+1}, \varpi^{i+1}, \varpi_X^{i+1}))$, where

- (1) $\sigma_i R_a \sigma_{i+1}$ for some $a \in \text{AG}_{\varphi}$;
- (2) there is a finite \xrightarrow{X} -path \varkappa_i in b from $\sigma_i \psi_i$ to $\sigma_{i+1} \psi_{i+1}$; and
- (3) \varkappa_i goes through some $\sigma' X$ if and only if $\varpi_X^{i+1} = \varepsilon$.

We define $p_{i+1} = p_i \varkappa_i$. The infinite path p is the path that has every p_i as a prefix. If X does not appear infinitely often in p , then there is some $i > 0$, such that every \varkappa_j , where $j > i$, does not go through any $\sigma' X$. The dependency relation \rightarrow goes through finitely many formulas, and therefore it goes through some infinitely many times. Let $\text{ip}^X(\psi', \varpi, \varpi_X)$ be the largest subformula of $\text{ip}^X(\psi, \varepsilon, \varepsilon)$ that appears infinitely often as a $\text{ip}^X(\psi_j, \varpi^j, \varpi_X^j)$.

Then, we observe that there is an infinite $\xrightarrow{Z_{\psi'}^X}$ -path from some $(f(\sigma_j), \text{ip}^X(\psi', \varpi, \varpi_X))$ that visits $Z_{\psi'}^X$ infinitely often. Therefore, the dependency relation is not lfp-finite, contradicting our assumption that $(\mathcal{M}, f(\sigma)) \models \text{ip}^X(\psi, \varepsilon, \varepsilon)$. We conclude that X appears infinitely often in p , which completes our proof. \square

Theorem 6.13. *The formula $\text{rules} \wedge \text{Inv}(\text{-inf_path}) \wedge \bigvee_{\varphi \in g \in G^t} g$ is satisfiable if and only if φ is \mathbf{L} -satisfiable.*

Proof. If $\text{rules} \wedge \text{Inv}(\text{-inf_path}) \wedge \bigvee_{\varphi \in g \in G^t} g$ is satisfiable, then there is a model $(\mathcal{M}, s) \models \text{rules} \wedge \text{Inv}(\text{-inf_path}) \wedge \bigvee_{\varphi \in g \in G^t} g$, and therefore $(\mathcal{M}, s) \models \text{rules}$. By Lemma 6.5, \mathcal{M}, s represent a maximal branch b that is not locally closed. From $(\mathcal{M}, s) \models \bigvee_{\varphi \in g \in G^t} g$, we have that b is a branch of a tableau for φ , and from $(\mathcal{M}, s) \models \text{Inv}(\text{-inf_path})$ and Lemma 6.12, that b is open. Theorem 4.5 now yields that φ is \mathbf{L} satisfiable and we are done.

On the other hand, if φ is \mathbf{L} -satisfiable, then its tableau has an open branch, and similarly, by Lemmata 6.6, 6.5, and 6.12, $\text{rules} \wedge \text{Inv}(\text{-inf_path}) \wedge \bigvee_{\varphi \in g \in G^t} g$ is satisfiable. \square

Corollary 6.14. *\mathbf{L} -satisfiability is in 2EXP, for every logic \mathbf{L} , where no agent has constraint 5.*

Proof. Given a formula φ , the height of the syntax tree of each $\mathbf{fp}(T, \varepsilon)$, $\mathbf{fp}^X(T, \varepsilon)$, and $\mathbf{ip}^X(\psi, \varepsilon)$, due to the dependency on the strings ϖ , is $O(|\varphi|^2)$. The syntax tree furthermore branches with at most $2^{|\varphi|}$ disjunctions or conjunctions at each node. Therefore, the formula $\mathbf{rules} \wedge \mathbf{Inv}(\neg\mathbf{inf_path}) \wedge \bigvee_{\varphi \in g \in G^t} g$ has size at most $2^{O(|\varphi|)}$. Each step of the construction only requires up to double-exponential time with respect to $|\varphi|$. Theorem 2.8 then yields the resulting double-exponential upper bound. \square

6.2. The case of logics with Negative Introspection. We now explain how to adjust the construction of \mathbf{rules} and $\mathbf{inf_path}$, to also handle the case of agents with negative introspection. What differs in this case, is that the tableau rules for these agents relate prefixes that are not necessarily consecutive. These prefixes are, however, close, in that their closest common ancestor is at worst their parent prefix. Therefore, the tableau rules are still, in that sense, local. We adjust the contents of the graphs that we use as variables, to take into account these extended formula interactions through the tableau rules. In effect, in the following construction, graphs are equipped with limited lookahead capabilities. Still, a graph represents one tableau prefix, but it also contains certain prefixed formulas, so it can describe certain extensions of the current prefix. This allows our formulas to describe \xrightarrow{X} -paths that traverse non-consecutive tableau prefixes, resulting from the tableau rules in Table 3 for agents with constraint 5.

We extend the set of formulas that appear in a graph, to include certain prefixed formulas. We also keep track of any agent with condition 5 that appears at the end of the prefix. Let $\mathbf{AG}_5 = \{\alpha \in \mathbf{AG} \mid \mathbf{L}(\alpha) \text{ has condition 5}\}$. For every $\alpha \in \mathbf{AG}_5$, let:

$$\begin{aligned} \mathbf{sub}_5^{\alpha\alpha}(\varphi) &= \{\varepsilon \psi, \beta\langle\psi_1\rangle \psi, \beta\langle\psi_1\rangle.\beta\langle\psi_2\rangle.\psi \mid \psi, \psi_1, \psi_2 \in \mathbf{sub}(\varphi), \alpha \neq \beta \in \mathbf{AG}_5\} \\ \mathbf{sub}_5^\alpha(\varphi) &= \mathbf{sub}_5^{\alpha\alpha}(\varphi) \cup \{\alpha\langle\psi_1\rangle \psi \mid \psi, \psi_1 \in \mathbf{sub}(\varphi)\} \\ \mathbf{sub}_5^\varepsilon(\varphi) &= \mathbf{sub}_5^\alpha(\varphi) \cup \{\alpha\langle\psi_1\rangle.\alpha\langle\psi_2\rangle.\psi \mid \psi, \psi_1, \psi_2 \in \mathbf{sub}(\varphi)\}. \end{aligned}$$

We modify the definition of \mathbf{G} , so that the graphs in \mathbf{G} are of the form

$$g = (\Phi, (\xrightarrow{X})_{X \in mV}, \ell, b),$$

where $b \in \{\varepsilon, \alpha, \alpha\alpha \mid \alpha \in \mathbf{AG}_5\}$, and $\Phi \subseteq \mathbf{sub}_5^b(\varphi)$. We also define $\mathbf{G}^b = \{(\Phi, (\xrightarrow{X})_{X \in mV}, \ell, b) \in \mathbf{G}\}$, for each $b \in \{\varepsilon, \alpha, \alpha\alpha \mid \alpha \in \mathbf{AG}_5\}$. For a tableau prefix σ and $\delta \in \mathbf{sub}_5(\varphi)$, we say that σ is δ -flat when $\sigma = \sigma' \cdot \alpha\langle\psi\rangle$ and $\delta = \alpha.\langle\psi'\rangle.\delta'$ or $\delta = \langle\alpha\rangle\psi'$ for some α, ψ, ψ' , such that $\mathbf{L}(\alpha)$ has condition 5. We say that $\alpha \in \mathbf{AG}$ is b -inactive when $b = \alpha\langle\psi\rangle b'$ for some ψ, b' . A tableau rule $\frac{\sigma \psi}{\sigma' \psi'}$ is b -active when ψ is not of the form $[\alpha]\psi''$ or $\langle\alpha\rangle\psi''$, where α is b -inactive.

The purpose of the additional prefixes in the graph formulas is to control the application of the tableau rules that relate to condition 5. A graph g that represents a prefix σ may also describe the branch at later prefixes. This ensures that our constructed formula has models that adhere to the more involved tableau rules and the conditions for applying these rules. However, this alteration does not affect the construction of the formulas for the infinite paths. In this case, however, we need a more detailed treatment of the tableau rules, due to their more complex behaviour. Figure 6 (top) illustrates the structure of these new graphs.

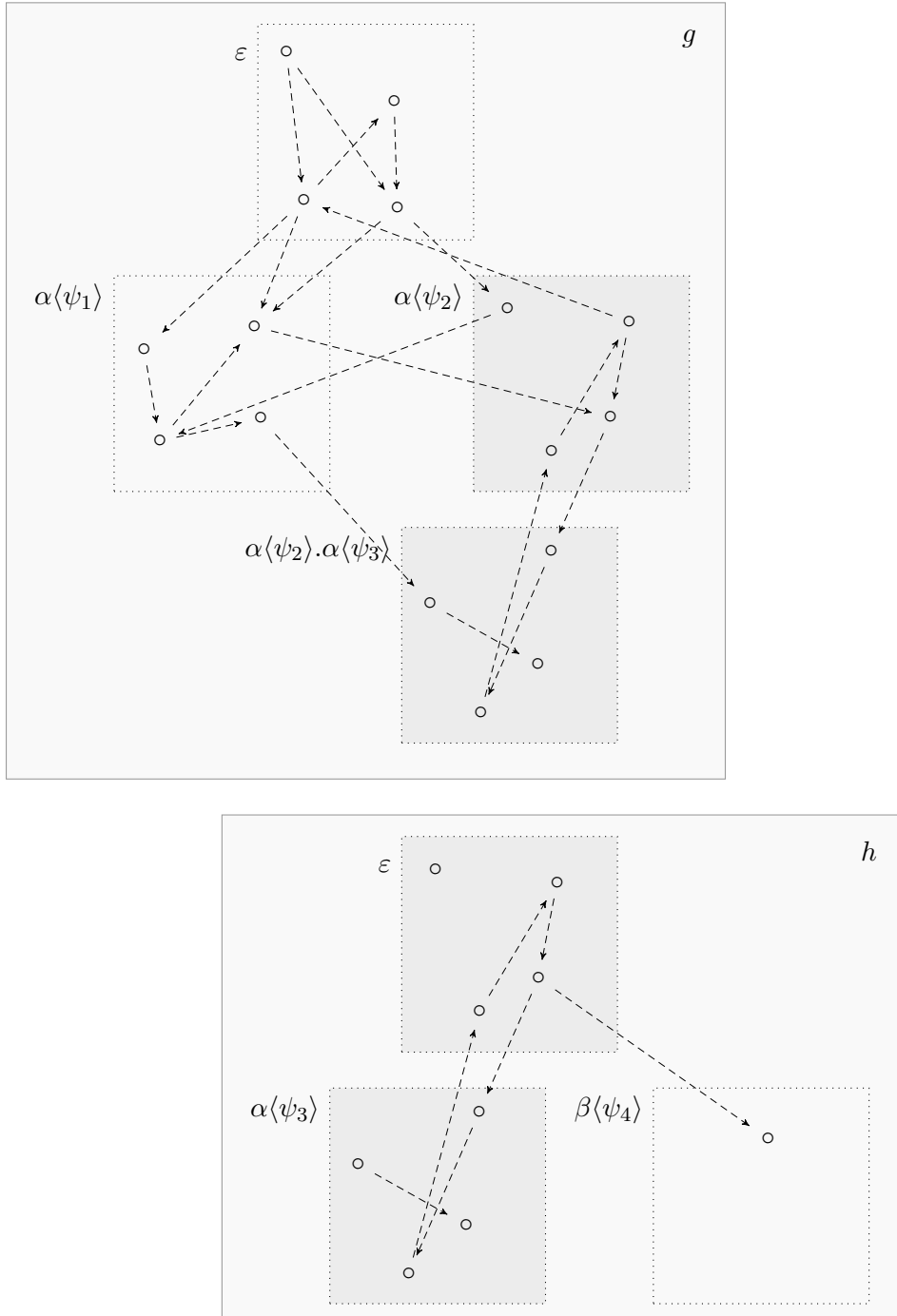


FIGURE 6. Graphs for logics with Condition 5. Each rectangle represents a graph and each dotted square represents the set of formulas that are prefixed by its label. Graph h is a prospective $\alpha\langle\psi_2\rangle$ -child of g and we see that the darker squares of the two graphs match, as in Definition 6.16.

Definition 6.15. We say that $g = (\Phi, (\overset{X}{\rightarrow})_{X \in mV}, \ell, b) \in \mathbf{G}$ is *locally compatible with the tableau rules* and write $com(g)$ when for every $\delta = \varsigma \psi \in \Phi$, $\beta \neq \alpha$, and $X \in mV$, where $\psi \neq Y$ for any $X < Y$, the following conditions are satisfied. For every pair σ, σ' of possible tableau prefixes and $\delta', \delta_1, \delta_2 \in \text{sub}_5(\varphi)$, where σ is not δ -flat:

- (1) if $\frac{\sigma.\varsigma \psi}{\sigma.\varsigma \psi'}$ is a b -active tableau rule, then $\varsigma \psi' \in \Phi$ and $\delta \overset{X}{\rightarrow} \varsigma \psi'$;
- (2) if $\frac{\sigma.\delta}{\sigma.\delta'}$ is a b -active instance of tableau rule (D), (d), (4), (B5), (D5), (b), or (b4), then $\delta' \in \Phi$ and $\delta \overset{X}{\rightarrow} \delta'$;
- (3) if $\frac{\sigma.\delta}{\sigma.\delta'}$ is a b -active instance of tableau rule (B), $\delta = \varsigma [\alpha]\psi_1$, and $\varsigma.\alpha\langle\psi_2\rangle \psi_3 \in \Phi$ for some $\psi_1, \psi_2, \psi_3 \in \text{sub}(\varphi)$, then $\delta' \in \Phi$ and $\delta \overset{X}{\rightarrow} \delta'$;
- (4) if $\frac{\sigma.\delta}{\sigma.\delta'}$ is a b -active instance of tableau rule (B55), $\delta = \varsigma [\alpha]\psi_1$, and $\sigma.\alpha\langle\psi_2\rangle \psi_3 \in \Phi$ for some $\psi_1, \psi_2, \psi_3 \in \text{sub}(\varphi)$, then $\delta' \in \Phi$ and $\delta \overset{X}{\rightarrow} \delta'$;
- (5) if $\frac{\sigma.\delta}{\sigma.\delta'}$ is a b -active instance of tableau rule (D55), $\delta = \varsigma \langle\alpha\rangle\psi'$, and $\sigma \langle\alpha\rangle\psi' \notin \Phi$ for some $\psi' \in \text{sub}(\varphi)$, then $\delta' \in \Phi$ and $\delta \overset{X}{\rightarrow} \delta'$;
- (6) if $\frac{\sigma.\delta}{\sigma.\delta_1 \mid \sigma.\delta_2}$ is an instance of a b -active tableau rule, then $\delta_1, \delta_2 \in \Phi$, $\delta \overset{X}{\rightarrow} \delta_1$, and $\delta \overset{X}{\rightarrow} \delta_2$;
- (7) if $\frac{\sigma.\delta}{\sigma.\delta_1 \mid \sigma.\delta_2}$ is a b -active tableau rule, then $\delta_1 \in \Phi$ and $\delta \overset{X}{\rightarrow} \delta_1$, or $\delta_2 \in \Phi$ and $\delta \overset{X}{\rightarrow} \delta_2$;
- (8) if $\frac{\sigma.\varsigma \psi}{\sigma.\varsigma' \psi'}$, where $\varsigma' \psi' \notin \text{sub}_5(\varphi)$ and ς' is a strict extension of ς , is a b -active tableau rule, then $(out, bot) \in \ell(\delta)$;
- (9) if $\frac{\sigma.\varsigma \psi}{\sigma' \psi'}$ is a b -active tableau rule, where $\sigma.\varsigma$ is a strict extension of σ' , then $(out, top) \in \ell(\delta)$ and
- (10) $\varsigma \neg\psi \notin \Phi$ and $\varsigma \mathbf{ff} \notin \Phi$.

Let $\mathbf{G}^t = \{g \in G \mid com(g)\}$.

Let $\sigma = \varepsilon$ or $\sigma = \alpha\langle\psi\rangle$, where $\alpha \in \text{AG}_5$ and $\psi \in \text{sub}(\varphi)$, and $g = (\Phi, (\overset{X}{\rightarrow})_{X \in mV}, \ell, b) \in \mathbf{G}$. We define $g^\sigma = (\Phi^\sigma, (\overset{X}{\rightarrow})_{X \in mV}, \ell^\sigma)$, where $\Phi^\sigma = \{\delta \mid \sigma.\delta \in \Phi\}$; $\overset{X}{\rightarrow}^\sigma = \{(\delta_1, \delta_2) \in (\Phi^\sigma)^2 \mid \sigma.\delta_1 \overset{X}{\rightarrow} \sigma.\delta_2\}$; and $\ell^\sigma(\delta) = \ell(\sigma.\delta) \cap \{(out, bot)\}$ for every $\delta \in \Phi^\sigma$.

Definition 6.16. Let $\alpha \in \text{AG}$, $\chi \in \text{sub}(\varphi)$, and

$$g = (\Phi_g, (\overset{X}{\rightarrow}_g)_{X \in mV}, \ell_g, b_g), \quad h = (\Phi_h, (\overset{X}{\rightarrow}_h)_{X \in mV}, \ell_h, b_h) \in G^t.$$

If $\alpha \in \text{AG}_5$, then we say that g is a prospective $\alpha\langle\chi\rangle$ -child of h , and write $c_{\alpha\langle\chi\rangle}(h, g)$ when

- (1) $\langle\alpha\rangle\chi \in \Phi_h$, $\langle\alpha\rangle\chi \overset{X}{\rightarrow}_h \alpha\langle\chi\rangle \chi$ for some X , $\chi \in \Phi_g$, and $(in, top) \in \ell_g(\chi)$;
- (2) $b_h \neq \alpha\alpha$;
- (3) $g^\varepsilon = h^{\alpha\langle\chi\rangle}$; and
- (4) if $b_h = \alpha$, then $b_g = \alpha\alpha$, otherwise $b_g = \alpha$.

If $\alpha \notin \text{AG}_5$, then we say that g is a prospective $\alpha\langle\chi\rangle$ -child of h , and write $c_{\alpha\langle\chi\rangle}(h, g)$ when:

- (1) $\langle\alpha\rangle\chi \in \Phi_h$ and $\chi \in \Phi_g$ and $(in, top) \in \ell_g(\chi)$;
- (2) if $[\alpha]\psi \in \Phi_h$ and $\frac{\sigma.[\alpha]\psi}{\sigma.\alpha\langle\psi_1\rangle \psi_2}$ is a tableau rule, then $\psi_2 \in \Phi_g$ and $(in, top) \in \ell_g(\psi_2)$;
- (3) if $[\alpha]\psi \in \Phi_g$ and $\frac{\sigma.\alpha\langle\psi_1\rangle [\alpha]\psi}{\sigma \psi_2}$ is a tableau rule, then $\psi_2 \in \Phi_h$ and $(in, bot) \in \ell_h(\psi_2)$; and
- (4) $b_g = \varepsilon$.

We can similarly define that g is a prospective α -child of h , and write $c_\alpha(h, g)$, by omitting the first condition in each case.

Figure 6 depicts two graphs, g and h , where $c_{\alpha(\psi_2)}(g, h)$ for some formula ψ_2 .

The size of \mathbf{G} is still at most exponential with respect to φ and it takes at most polynomial time to verify $com(g)$ and $c(g, h)$; furthermore, $|AG_5^\varphi| \leq 2|AG| \cdot |\text{sub}(\varphi)|^2$.

Let b be a tableau branch. For the remaining construction, to express when there exists a path that visits some $a X$ infinitely often, it does not matter what objects the vertices of the graphs are. We can then continue with the construction, similarly as for the case of logics with no agents with negative introspection.

Corollary 6.17. *L-satisfiability is in 2EXP.*

7. CONCLUSIONS

In this work we studied the complexity of the satisfiability problem for multi-modal logics with recursion. These logics mix the frame conditions from epistemic modal logic and the recursion operators of the μ -calculus.

In the first part we gave translations among several of these logics that connect their satisfiability problems. This approach allowed us to offer complexity bounds for satisfiability and to prove finite-model results. We also presented a sound and complete tableau that has termination guarantees, conditional on a logic's finite-model property. Using our translations, we obtained several hardness and completeness results for individual logics and their combinations, which are summarized in Table 2, with a few isolated cases missing.

In the second part of this work we aimed to fill these gaps in complexity results, specifically for the case of the multi-agent μ -calculus over frames containing the **B** and **K5** restrictions and the case of single-agent μ -calculus over transitive frames. To tackle the multi-agent case, we gave a translation from tableaux to μ -calculus formulas, which allowed us to use the known algorithm for testing satisfiability over the μ -calculus in that setting. However, the translation we provided was not polynomial in size. Thus, it yielded an upper bound on the complexity of satisfiability that did not match the PSPACE hardness result stemming from classic work on modal logics. To address this complexity gap, we customized the tableaux exploration algorithm we developed to utilize the transitivity condition on frames. Doing so allowed us to establish a tight PSPACE bound for the satisfiability problem in that case.

Table 5 summarises the results.

7.1. Discussion on the existence of a translation from $\mathbf{K4}^\mu$ to $\mathbf{K4}$. The reader might wonder whether some of the complexity results we obtained with the use of tableaux might have also been obtained via translations. For example, Section 5 is dedicated to proving the tight bound for the complexity of satisfiability for formulas in $\mathbf{K4}^\mu$. We will discuss here why we think this would be very hard to do via translations. To apply our translation technique over $\mathbf{K4}^\mu$, we would need to translate formulas in $\mathbf{K4}^\mu$ to formulas of some logic whose satisfiability problem is in PSPACE. A natural choice would be modal logics without recursion. Amongst those, a prime candidate would be **K4** as the transitivity of the models would be easier to tackle. Therefore we would need to focus our efforts on developing “efficient” translations that remove fixed-point operators from formulas.

# agents	Restrictions on syntax/frames	Upper Bound	Lower Bound
≥ 2	frames with B or 5	2EXP 6.17	EXP-hard 2.11
	not B , 5	EXP 3.26	EXP-hard 2.11
	not 5, not μ . X	EXP 3.27	EXP-hard 2.10
1	with 5 (or B4)	NP 2.5	NP-hard 2.5
	with 4	PSPACE 5.6	PSPACE-hard 2.5
	Any other restrictions	EXP 2.8	EXP-hard 2.8

TABLE 5. The updated summary of the complexity of satisfiability checking for various modal logics with recursion. Our additional contributions are highlighted.

The above task seems relatively easy for \Box -free formulas since we can remove consecutive diamonds from such formulas due to the transitivity restriction on the frames. This is straightforward over transitive frames as any subformula starting with two or more consecutive diamonds is exactly satisfied when the similar formula, where all the consecutive diamonds are merged into one, is satisfiable.

At the same time, we know that we can express the invariance operator $\text{Inv}(-)$ in $\mathbf{K4}$ by using a box, and therefore we can simulate some form of recursion. We would hope that an intuition like this would help us design a translation. Unfortunately, it turns out that tackling arbitrary formulas with boxes is less easy. We now discuss the use of our translation formulas for the case of $\mathbf{K4}^\mu$.

Till now, all translations we have presented have a simple characteristic. Namely, given a formula φ over a logic \mathbf{L} and its translation $f(\varphi)$ for a logic \mathbf{L}' , we have that all models that model the formula $f(\varphi)$, will also model the formula φ , or, in some sense, describe a model of φ . This can be seen by observing the way we have defined translations so far, and the proofs of their correctness. Being able to extract a model for φ from a model of $f(\varphi)$ is an important aspect of our translations, because it is at the core of our arguments regarding compositionality (see Section 3.1.1). We demonstrate here our argument on why such a (well-behaved) translation may not be possible.

To do so, assume that such a translation exists and consider again the formula $\varphi_f = \mu X. \Box X$, which requires all paths in the model to be finite, from Example 3.12. The formula is both recursive and its satisfiability is not affected by the transitivity restriction. Therefore, due to the translation, there should exist a satisfiable property ϕ' in $\mathbf{K4}$ that is the translation of φ_f . Let k be the number of subformulas of ϕ' . We now write a formula $\psi \in \mathbf{K4}^\mu$ such that ψ is satisfied on a (transitive) model having $2k$ -many states connected in a path. Said formula can even be expressed without recursion and a model \mathcal{M} for it is shown in Figure 7 (before the addition of the red edge connecting n_{i+1} to itself).

We now study \mathcal{M} and its relation to ϕ' . We see that we should have $\mathcal{M} \models \phi'$ as all paths of \mathcal{M} are indeed finite. However, the length of the longest path of \mathcal{M} is $2 * k$ which is larger than the modal depth of ϕ' . Moreover, since \mathcal{M} is transitive we know that as we “move away” from the root node, more box-formulas are satisfied, due to the transitivity axiom $\Box p \implies \Box \Box p$. Thus, the number of subformulas of ϕ' satisfiable in the states of \mathcal{M} increases the further we move away from the root. Finally, we know the number of \Box prefixed sub-formulas of ϕ' is finite and bounded by k .

Thus, there must exist nodes n_i, n_{i+1} , as shown in Figure 7, that satisfy exactly the same number of box-prefixed subformulas of ϕ' . We now add the red loop-edge on n_{i+1} as seen

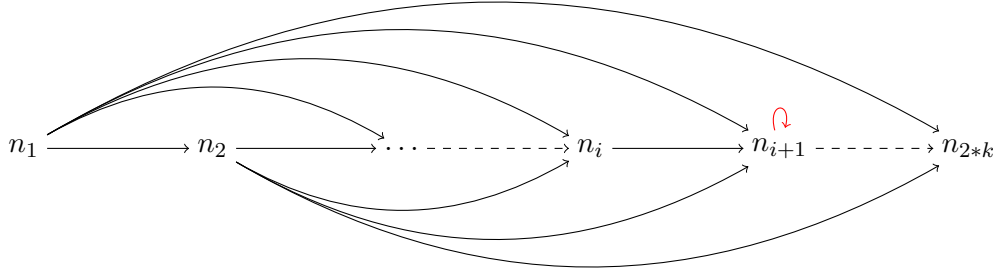


FIGURE 7. The model \mathcal{M} of property ψ , before and after adding the loop edge.

in the figure. The addition of said edge cannot affect the validity of any diamond-prefixed subformula of ϕ' , since it only extends the reachable states of n_{i+i} . Regarding the boxes, we inspect whether the addition of the loop edge could affect the boxes of n_{i+1} . However, the boxes of n_i are the same as those of n_{i+i} (before the addition of the loop), and due to transitivity and the form of \mathcal{M} the states reachable by n_i are exactly n_{i+1} and the states reachable by n_{i+1} . Therefore, all box-prefixed formulas of n_i are also applied in n_{i+1} , and thus the addition of the loop preserves the satisfaction of the box-formulas of n_{i+1} .

We now have constructed the model \mathcal{M}' (after the addition of the red loop-edge), which satisfies identical subformulas of ϕ' in the identical states as \mathcal{M} . Note here that the assumption was that a translation that preserves satisfiability exists, and we used such a translation on the formula φ . Thus, it must be the case that $\mathcal{M}' \models \phi'$. However, φ_f states that all paths are finite, which means that $\mathcal{M} \models \varphi_f$, and $\mathcal{M}' \not\models \varphi_f$, while $\mathcal{M}, \mathcal{M}' \models \varphi'$. Therefore, it is hard to see how any translated formula would “agree” with the initial one on certain models, or describe models for φ . So far the translations we have defined always work, with the proof producing a model that satisfies both the old formula and the new. It is possible that more complicated translations exist that do not have this property, but we were not able to think of one.

Conjectures and Future Work. We currently do not have a tight complexity bound for the case of the multi-agent μ -calculus over symmetric or euclidean frames. What is interesting is that we also do not have a counterexample to prove that the translations of Section 3.1.3 that we already have, as well as other ones we attempted, do not preserve satisfiability.

The complexity of the model checking problem for the μ -calculus is an important open problem, known to have a quasi-polynomial time solution, but not known whether it is in P [LPSW22, Leh18b, JL17b, CJK⁺17, FJS⁺17]. The problem does not depend on the frame restrictions of the particular logic, though one may wonder whether additional frame restrictions would help solve the problem more efficiently. Currently, we are not aware of a way to use our translations to obtain such an improvement.

As, to the best of our knowledge, most of the logics described in this chapter have not been explicitly defined before, with notable exceptions such as [DL10, DL13, AF09], they also lack any axiomatizations and completeness theorems. We do expect the classical methods from [Koz83, Lad77, HM92] and others to work in these cases as well. However, it would

be interesting and desirable to flesh out the details and see if there are any unexpected situations that arise.

Given the importance of common knowledge for epistemic logic and the fact that it has been known that common knowledge can be thought of as a (greatest) fixed point already from [Har77, Bar88], we consider the logics that we presented to be natural extensions of modal logic. Besides the examples given in Section 2, we are interested in exploring what other natural concepts can be defined with this enlarged language. Finally, it would be interesting to prove a result showing that, under some natural assumptions, there is no “efficient” satisfiability-preserving translation from $\mathbf{K4}^\mu$ to $\mathbf{K4}$ or other modal logics.

ACKNOWLEDGEMENTS

We thank the anonymous reviewers for their insightful comments and criticisms on the submitted manuscript. We are most grateful to Laura Bozzelli, whose detailed feedback and pointers led to substantial improvements in both the presentation and the technical contents in this article.

REFERENCES

- [AAA⁺22] Luca Aceto, Antonis Achilleos, Elli Anastasiadi, Adrian Francalanza, and Anna Ingólfssdóttir. Complexity through translations for modal logic with recursion. In Pierre Ganty and Dario Della Monica, editors, *Proceedings of the 13th International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2022, Madrid, Spain, September 21-23, 2022*, volume 370 of *EPTCS*, pages 34–48, 2022. doi:10.4204/EPTCS.370.3.
- [AAFI20] Luca Aceto, Antonis Achilleos, Adrian Francalanza, and Anna Ingólfssdóttir. The complexity of identifying characteristic formulae. *J. Log. Algebraic Methods Program.*, 112:100529, 2020. URL: <https://doi.org/10.1016/j.jlamp.2020.100529>.
- [AF09] Luca Alberucci and Alessandro Facchini. The modal μ -calculus hierarchy over restricted classes of transition systems. *The Journal of Symbolic Logic*, 74(4):1367–1400, 2009. doi:10.2178/jsl/1254748696.
- [Bar88] Jon Barwise. Three views of common knowledge. In *Proceedings of the 2nd Conference on Theoretical Aspects of Reasoning About Knowledge, TARK '88*, pages 365–379, San Francisco, CA, USA, 1988. Morgan Kaufmann Publishers Inc. URL: <http://dl.acm.org/citation.cfm?id=1029718.1029753>.
- [BdRV01] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2001. doi:10.1017/cbo9781107050884.
- [BMP15] Laura Bozzelli, Bastien Maubert, and Sophie Pinchinat. Unifying hyper and epistemic temporal logics. In Andrew Pitts, editor, *Foundations of Software Science and Computation Structures*, pages 167–182, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.
- [BvBW06] Patrick Blackburn, Johan van Benthem, and Frank Wolter. *Handbook of Modal Logic*, volume 3 of *Studies in Logic and Practical Reasoning*. Elsevier Science, 2006.
- [BW18] Julian C. Bradfield and Igor Walukiewicz. The mu-calculus and model checking. In Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem, editors, *Handbook of Model Checking*, pages 871–919. Springer, 2018. doi:10.1007/978-3-319-10575-8_26.
- [CHVB18] Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem, editors. *Handbook of Model Checking*. Springer, 2018. doi:10.1007/978-3-319-10575-8.
- [CJK⁺17] Cristian S Calude, Sanjay Jain, Bakhadyr Khoussainov, Wei Li, and Frank Stephan. Deciding parity games in quasipolynomial time. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 252–263, 2017.
- [CJK⁺22] Cristian S. Calude, Sanjay Jain, Bakhadyr Khoussainov, Wei Li, and Frank Stephan. Deciding parity games in quasi-polynomial time. *SIAM Journal on Computing*, 51(2):17–152, 2022. URL: <https://doi.org/10.1137/17m1145288>.

- [CS10] Michael R. Clarkson and Fred B. Schneider. Hyperproperties. *Journal of Computer Security*, 18(6):1157–1210, September 2010. doi:10.3233/JCS-2009-0393.
- [DL10] Giovanna D’Agostino and Giacomo Lenzi. On the μ -calculus over transitive and finite transitive frames. *Theoretical Computer Science*, 411(50):4273–4290, 2010. URL: <https://www.sciencedirect.com/science/article/pii/S030439751000469X>, doi:10.1016/j.tcs.2010.09.002.
- [DL13] Giovanna D’Agostino and Giacomo Lenzi. On modal μ -calculus in S5 and applications. *Fundamenta Informaticae*, 124(4):465–482, 2013.
- [DL15] Giovanna D’Agostino and Giacomo Lenzi. On the modal μ -calculus over finite symmetric graphs. *Mathematica Slovaca*, 65(4):731–746, 2015.
- [DL18] Giovanna D’Agostino and Giacomo Lenzi. The μ -calculus alternation depth hierarchy is infinite over finite planar graphs. *Theoretical Computer Science*, 737:40–61, 2018. URL: <https://www.sciencedirect.com/science/article/pii/S0304397518302317>, doi:10.1016/j.tcs.2018.04.009.
- [EJS01] E Allen Emerson, Charanjit S Jutla, and A Prasad Sistla. On model checking for the μ -calculus and its fragments. *Theoretical Computer Science*, 258(1-2):491–522, 2001. doi:10.1016/S0304-3975(00)00034-7.
- [FHMV95] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning About Knowledge*. The MIT Press, 1995. doi:10.7551/mitpress/5803.001.0001.
- [Fit72] Melvin Fitting. Tableau methods of proof for modal logics. *Notre Dame Journal of Formal Logic*, 13(2):237–247, 1972.
- [FJS⁺17] John Fearnley, Sanjay Jain, Sven Schewe, Frank Stephan, and Dominik Wojtczak. An ordered approach to solving parity games in quasi polynomial time and quasi linear space. In *Proceedings of the 24th ACM SIGSOFT International SPIN Symposium on Model Checking of Software*, pages 112–121, 2017.
- [FL79] Michael J. Fischer and Richard E. Ladner. Propositional dynamic logic of regular programs. *Journal of computer and system sciences*, 18(2):194–211, 1979. doi:10.1016/0022-0000(79)90046-1.
- [GO07] Valentin Goranko and Martin Otto. Model theory of modal logic. In Patrick Blackburn, Johan Van Benthem, and Frank Wolter, editors, *Handbook of Modal Logic*, volume 3 of *Studies in Logic and Practical Reasoning*, pages 249–329. Elsevier, 2007. doi:10.1016/S1570-2464(07)80008-5.
- [Har15] Friedrich Hartogs. Über das problem der wohlordnung. *Mathematische Annalen*, 76(4):438–443, 1915.
- [Har77] Gilbert Harman. Review of linguistic behavior by Jonathan Bennett. *Language*, 53:417–424, 1977.
- [HM85] Matthew Hennessy and Robin Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM (JACM)*, 32(1):137–161, 1985. doi:10.1145/2455.2460.
- [HM92] Joseph Y. Halpern and Yoram Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54(3):319–379, 1992. doi:10.1016/0004-3702(92)90049-4.
- [HR07] Joseph Y. Halpern and Leandro Chaves Rêgo. Characterizing the NP-PSPACE gap in the satisfiability problem for modal logic. *Journal of Logic and Computation*, 17(4):795–806, 2007. doi:10.1093/logcom/exm029.
- [JL17a] Marcin Jurdzinski and Ranko Lazic. Succinct progress measures for solving parity games. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017*, pages 1–9. IEEE Computer Society, 2017. URL: <https://doi.org/10.1109/LICS.2017.8005092>.
- [JL17b] Marcin Jurdziński and Ranko Lazić. Succinct progress measures for solving parity games. In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–9. IEEE, 2017.
- [Jur98] Marcin Jurdziński. Deciding the winner in parity games is in $UP \cap co-UP$. *Information Processing Letters*, 68(3):119–124, 1998.
- [Kna28] Bronisław Knaster. Un theoreme sur les fonctions d’ensembles. *Ann. Soc. Polon. Math.*, 6:133–134, 1928.
- [Koz83] Dexter Kozen. Results on the propositional μ -calculus. *TCS*, 27:333–354, 1983. doi:10.1016/0304-3975(82)90125-6.

- [Lad77] Richard E. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal on Computing*, 6(3):467–480, 1977. doi:10.1137/0206033.
- [Leh18a] Karoliina Lehtinen. A modal μ perspective on solving parity games in quasi-polynomial time. In Anuj Dawar and Erich Grädel, editors, *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018*, pages 639–648. ACM, 2018. URL: <https://doi.org/10.1145/3209108.3209115>.
- [Leh18b] Karoliina Lehtinen. A modal μ perspective on solving parity games in quasi-polynomial time. In Anuj Dawar and Erich Grädel, editors, *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*, pages 639–648. ACM, 2018. doi:10.1145/3209108.3209115.
- [LPSW22] Karoliina Lehtinen, Pawel Parys, Sven Schewe, and Dominik Wojtczak. A recursive approach to solving parity games in quasipolynomial time. *Log. Methods Comput. Sci.*, 18(1), 2022. doi:10.46298/lmcs-18(1:8)2022.
- [Mas94] Fabio Massacci. Strongly analytic tableaux for normal modal logics. In *CADE*, pages 723–737, 1994. doi:10.1007/3-540-58156-1_52.
- [NT85] Michael C. Nagle and S. K. Thomason. The extensions of the modal logic K5. *Journal of Symbolic Logic*, 50(1):102–109, 1985. doi:10.2307/2273793.
- [Pra81] Vaughan R. Pratt. A decidable mu-calculus: Preliminary report. In *22nd Annual Symposium on Foundations of Computer Science (SFCS 1981)*. IEEE, oct 1981. doi:10.1109/sfcs.1981.4.
- [RS18] Mikhail Rybakov and Dmitry Shkatov. Complexity of finite-variable fragments of propositional modal logics of symmetric frames. *Logic Journal of the IGPL*, 27(1):60–68, 07 2018. doi:10.1093/jigpal/jzy018.
- [Sah75] Henrik Sahlqvist. Completeness and correspondence in the first and second order semantics for modal logic. In Stig Kanger, editor, *Proceedings of the Third Scandinavian Logic Symposium*, volume 82 of *Studies in Logic and the Foundations of Mathematics*, pages 110–143. Elsevier, 1975. URL: <https://www.sciencedirect.com/science/article/pii/S0049237X08707286>, doi:10.1016/S0049-237X(08)70728-6.
- [Tar55] Alfred Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5(2):285–309, 1955.
- [Zie98] Wiesław Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theoretical Computer Science*, 200(1-2):135–183, 1998.