Complexity through Translations for Modal Logic with Recursion[*]

Luca Aceto
Department of Computer Science
Reykjavik University
Reykjavik, Iceland
Gran Sasso Science Institute
L'Aquila, Italy
luca@ru.is

Antonis Achilleos
Department of Computer Science
Reykjavik University
Reykjavik, Iceland
antonios@ru.is

Elli Anastasiadi
Department of Computer Science
Reykjavik University
Reykjavik, Iceland
elli19@ru.is

Adrian Francalanza
Department of Computer Science, ICT
University of Malta
Msida, Malta
adrian.francalanza@um.edu.mt

Anna Ingolfsdottir
Department of Computer Science
Reykjavik University
Reykjavik, Iceland
annai@ru.is

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 a terminating tableau system for the logics we study, based on Kozen's tableau for the μ-calculus, and the one of Fitting and Massacci for modal logic.

1 Introduction

We introduce a family of multi-modal logics with fixed-point operators that are interpreted on restricted classes of Kripke models. One can consider these logics as extensions of the usual multi-agent logics of knowledge and belief [14] by adding recursion to their syntax or of the μ-calculus [21] by interpreting formulas on 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.

Modal logic comes in several variations [5]. Some of these, such as multi-modal logics of knowledge and belief [14], 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 [21] can be seen as an extension of the smallest normal modal logic **K** with greatest and least fixed-point operators, νX and μX respectively. 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.

We are interested in the complexity of satisfiability for the resulting logics. Satisfiability for the μ-calculus is known to be EXP-complete [21], while for the modal logics between **K** and **S5** the problem

[*]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(l)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.

D. Della Monica and P. Ganty (Eds.): 13th International Symposium on
Games, Automata, Logics and Formal Verification (GandALF 22)
EPTCS 370, 2022, pp. 34–48, doi:10.4204/EPTCS.370.3

is PSPACE-complete or NP-complete, depending on whether they have Negative Introspection [18, 22]. 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 [17].

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 [2]. D'Agostino and Lenzi have studied the μ-calculus on different classes of frames in great detail. In [11], they reduce the μ-calculus over finite transitive frames to first-order logic. In [9], 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 [12], they consider finite symmetric frames, and they prove that $\mathbf{B}^{\mu}$-satisfiability is in 2EXP, and EXP-hard. They also examine planar frames in [10], where they show that the alternation hierarchy of the μ-calculus over planar frames is infinite.

Our primary method of 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 $\mathbf{K}_n$, reproving the PSPACE upper bound for these logics (Theorem 8 and Corollary 9). 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 properties of our translations, that several behaviors induced on the transitions 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,B (Corollaries 15 and 16). Finally, we present tableaux for the discussed logics, based on the ones by Kozen for the μ-calculus [21], and by Fitting and Massacci for modal logic [16, 24]. We give tableau-termination conditions for every logic with a finite model property (Theorem 19).

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: $\nu X.(\varphi \wedge \bigwedge_{\alpha}[\alpha]X)$. But the combination of epistemic logics and fixed-points can potentially express more interesting epistemic concepts. For instance, the formula $\mu X.\bigvee_{\alpha}([\alpha]\varphi \vee [\alpha]X)$, in the context of a belief interpretation, can be thought to claim 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 [8] if a set of agents represents different streams, and combinations of epistemic and temporal or hyper-logics have recently been used to express safety and privacy properties of systems [7].

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 finally give tableaux for our multi-modal logics with recursion. We conclude in Section 5 with a set of open questions and directions. Omitted proofs can be found in the full version of the paper [23].

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 1. *We consider formulas constructed from the following grammar:*

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

where X comes from a countable 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 $[A]\varphi$ to mean $\bigwedge_{\alpha \in A} [\alpha]\varphi$ and $\langle A\rangle\varphi$ for $\bigvee_{\alpha \in A} \langle\alpha\rangle\varphi$.*

A formula is closed when every occurrence of a variable X is in the scope of 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 nesting depth of the modal operators[1]. The modal depth of φ is defined inductively as:

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

We assume that in formulas, each recursion variable X appears in a unique fixed-point formula $\mathsf{fx}(X)$, which is either of the form $\mu X.\varphi$ or $\nu X.\varphi$. If $\mathsf{fx}(X)$ 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 on fixed-point variables, such that $X \leq Y$ iff $\mathsf{fx}(X)$ is a subformula of $\mathsf{fx}(Y)$, 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 $cl(\varphi) = cl(\varphi[\mathsf{fx}(X)/X])$, where $\varphi[\psi/X]$ is the usual substitution operation, and if φ is closed, then $cl(\varphi) = \varphi$.

We define $\mathsf{sub}(\varphi)$ as the set of subformulas of φ, and $|\varphi| = |\mathsf{sub}(\varphi)|$ 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 $\overline{\mathsf{sub}}(\varphi) = \mathsf{sub}(\varphi) \cup \{\neg\psi \in L \mid \psi \in \mathsf{sub}(\varphi)\}$.

Semantics We interpret formulas on the states of a *Kripke model*. A Kripke model, or simply model, is a quadruple $M = (W, R, V)$ 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 on which states a propositional variable is true. (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 $[\![-]\!]_{\mathscr{M}}$, defined in Table 1. The semantics of $\neg\varphi$ are constructed as usual, where $[\![\neg X, \rho]\!]_{\mathscr{M}} = W \backslash \rho(X)$.

We sometimes use $\mathscr{M}, s \models_\rho \varphi$ for $s \in [\![\varphi, \rho]\!]_{\mathscr{M}}$, and as the environment has no effect on the semantics of a closed formula φ, we often drop it from the notation and write $\mathscr{M}, s \models \varphi$ or $s \in [\![\varphi]\!]_{\mathscr{M}}$. If $\mathscr{M}, s \models \varphi$, 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.

[1] The modal depth of recursive formulas can be either zero, or infinite. However, this is not relevant for the spectrum of this work.

$$
\begin{array}{llll}
[\![\mathtt{tt},\rho]\!] = W, & [\![\mathtt{ff},\rho]\!] = \emptyset, & [\![p,\rho]\!] = \{s \mid p \in V(s)\}, & [\![\neg p,\rho]\!] = W \backslash [\![p,\rho]\!],
\end{array}
$$

$$
\begin{array}{ll}
[\![[\alpha]\varphi,\rho]\!] = \left\{s \;\middle|\; \forall t.\ sR_\alpha t \text{ implies } t \in [\![\varphi,\rho]\!]\right\}, & [\![\varphi_1 \wedge \varphi_2,\rho]\!] = [\![\varphi_1,\rho]\!] \cap [\![\varphi_2,\rho]\!], \\
[\![\langle\alpha\rangle\varphi,\rho]\!] = \left\{s \;\middle|\; \exists t.\ sR_\alpha t \text{ and } t \in [\![\varphi,\rho]\!]\right\}, & [\![\varphi_1 \vee \varphi_2,\rho]\!] = [\![\varphi_1,\rho]\!] \cup [\![\varphi_2,\rho]\!], \\
[\![\mu X.\varphi,\rho]\!] = \bigcap\left\{S \;\middle|\; S \supseteq [\![\varphi,\rho[X \mapsto S]]\!]\right\}, & [\![X,\rho]\!] = \rho(X), \\
[\![\nu X.\varphi,\rho]\!] = \bigcup\left\{S \;\middle|\; S \subseteq [\![\varphi,\rho[X \mapsto S]]\!]\right\}. &
\end{array}
$$

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

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 [21]. The max-fragment (resp. min-fragment) of the μ-calculus is the fragment that only allows the νX (resp. the μX) recursive operator. If $|\textsc{Ag}| = k$ 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 [6]). We give names to the following frame conditions, or frame constraints, for the case where $\textsc{Ag} = \{\alpha\}$. These conditions correspond to the usual axioms for normal modal logics — see [5, 6, 14], which we will revisit in Section 3.

D: R is serial: $\forall s.\exists t.sRt$;

T: R is reflexive: $\forall s.sRs$;

B: R is symmetric: $\forall s,t.(sRt \Rightarrow tRs)$;

4: R is transitive: $\forall s,t,r.(sRtRr \Rightarrow sRr)$;

5: R is euclidean: $\forall s,t,r.$ 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. *D*, which we call Consistency, is a special case of *T*, called Factivity. Constraint 4 is Positive Introspection and 5 is called Negative Introspection.[2] Given a logic $\mathbf{L}$ and constraint c, $\mathbf{L} + c$ is 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 among 4 and 5. Therefore, logic $\mathbf{D}$ is $\mathbf{K} + D$, $\mathbf{T}$ is $\mathbf{K} + T$, $\mathbf{B}$ is $\mathbf{K} + B$, $\mathbf{K4} = \mathbf{K} + 4$, $\mathbf{D4} = \mathbf{K} + D + 4 = \mathbf{D} + 4$, 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 $\textsc{Ag}$ 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 $\alpha \in \textsc{Ag}$, (W, R_α) is a frame for $\mathbf{L}(\alpha)$.

For a logic $\mathbf{L}$, $\mathbf{L}^\mu$ is 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 $\alpha \in \textsc{Ag}$, $\mathbf{L}(\alpha)$ is the same single-agent logic $\mathbf{L}$, we write $\mathbf{L}$ as $\mathbf{L}_k$, where $|\textsc{Ag}| = k$. Therefore, the μ-calculus is $\mathbf{K}_k^\mu$.

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 1. For a formula φ, we define $\mathit{Inv}(\varphi) = \nu X.(\varphi \wedge [\textsc{Ag}]X)$. $\mathit{Inv}(\varphi)$ 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 $\mathit{Eve}(\varphi) = \mu X.(\varphi \vee \langle\textsc{Ag}\rangle X)$, which asserts that φ is true in *some* reachable state.

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

2.2 Known Results

For logic **L**, the satisfiability problem for **L**, or **L**-satisfiability is the problem that asks, given a formula φ, if φ is satisfiable. Similarly, the model checking problem for **L** asks if φ is true at a given state of a given finite model.

Ladner [22] established the classical result of PSPACE-completeness for the satisfiability of **K**, **T**, **D**, **K4**, **D4**, and **S4** and NP-completeness for the satisfiability of **S5**. Halpern and Rêgo later characterized the NP–PSPACE gap for one-action logics by the presence or absence of Negative Introspection [18], resulting in Theorem 1. Later, Rybakov and Shkatov [27] proved the PSPACE-completeness of **B** and **TB**. For formulas with fixed-point operators, D'Agostino and Lenzi in [9] show that satisfiability for single-agent logics with constraint 5 is also NP-complete.

Theorem 1 ([18,22,27]). *If* $\mathbf{L} \in \{\mathbf{K},\mathbf{T},\mathbf{D},\mathbf{B},\mathbf{TB},\mathbf{K4},\mathbf{D4},\mathbf{S4}\}$*, then* **L***-satisfiability is* PSPACE*-complete; and* $\mathbf{L}+5$*-satisfiability and* $(\mathbf{L}+5)^{\mu}$*-satisfiability is* NP*-complete.*

Theorem 2 ([17]). *If $k > 1$ and* **L** *has a combination of constraints from $D, T, 4, 5$ and no recursive operators, then* $\mathbf{L}_k$*-satisfiability is* PSPACE*-complete.*

Remark 1. Note that Halpern and Moses in [17] prove these bounds for the cases of $\mathbf{K}_k, \mathbf{T}_k, \mathbf{S4}_k, \mathbf{KD45}_k$, and $\mathbf{S5}_k$ only. Similarly, D'Agostino and Lenzi in [9] 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 1 and 2. ■

Theorem 3 ([21]). *The satisfiability problem for the μ-calculus is* EXP*-complete.*

Theorem 4 ([13]). *The model checking problem for the μ-calculus is in* $\mathsf{NP} \cap \mathsf{coNP}$.[3]

Finally we have the following initial known results about the complexity of satisfiability, when we have recursive operators. Theorems 5 and 6 have already been observed in [1].

Theorem 5. *The satisfiability problem for the min- and max-fragments of the μ-calculus is* EXP*-complete, even when* $|\mathrm{AG}| = 1$.

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 3, and these fragments suffice [26] to describe the PDL formula that is constructed by the reduction used in [15] 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. □

It is not hard to express in logics with both frame constraints and recursion operators that formula φ is common knowledge, with formula $\nu X.\varphi \wedge [\mathrm{AG}]X$. Since validity for $\mathbf{L}_k$ with common knowledge (and without recursive operators) and $k > 1$ is EXP-complete [17][4], $\mathbf{L}_k^{\mu}$ is EXP-hard.

Proposition 6. *Satisfiability for* $\mathbf{L}_k^{\mu}$*, where $k > 1$, is* EXP*-hard.*

3 Complexity through Translations

In this section, we examine **L**-satisfiability. We 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.

[3] In fact, the problem is known to be in $\mathsf{UP} \cap \mathsf{coUP}$ [20].

[4] Similarly to Remark 1, [17] does not explicitly cover all these cases, but the techniques can be adjusted.

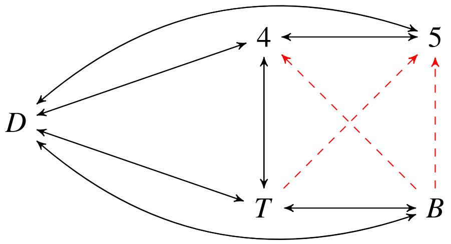

Figure 1: The frame property hierarchy

In the context of this paper, 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, transfering complexity bounds between logics.

According to Theorem 3, $\mathbf{K}_k^\mu$-satisfiability is EXP-complete, and therefore for each logic $\mathbf{L}$, we aim to connect $\mathbf{K}_k^\mu$ 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 to logics with fewer frame conditions. To this end, we study how taking the closure of a frame under one condition affects any 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 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 aquires with each application.

Let $F = (W,R)$ be a frame, $\alpha \in A \subseteq \textsc{Ag}$, and x a frame restriction among $T,B,4,5$. Then, $\overline{R_\alpha}^x$ is the closure of R_α under x, $\overline{R}^{x,A}$ is defined by $\overline{R}^{x,A}_\beta = \overline{R_\beta}^x$, if $\beta \in A$, and $\overline{R}^{x,A}_\beta = R_\beta$, otherwise. Then, $\overline{F}^{x,A} = (W,\overline{R}^{x,A})$. We make the following observation.

Lemma 7. *Let x be a frame restriction among $D,T,B,4,5$, and y a frame restriction among $T,B,4,5$, such that $(x,y) \neq (4,B),(5,T),(5,B)$. Then, for every frame F that satisfies x, $\overline{F}^y$ also satisfies x.*

According to Lemma 7, frame conditions are preserved as seen in Figure 1. In Figure 1, an arrow from x to y indicates that property x is preserved though the closure of a frame under y. Dotted red arrows indicate one-way arrows. For convenience, we define $\overline{F}^D = (W,\overline{R}^D)$, where $\overline{R}^D = R \cup \{(a,a) \in W^2 \mid \nexists (a,b) \in R\}$.

Remark 2. We note that, in general, not all frame conditions are preserved through all closures under another condition. For example, the accessibility relation $\{(a,b),(b,b)\}$ is euclidean, but its reflexive closure $\{(a,b),(b,b),(a,a)\}$ is not.

There is at least one linear ordering of the frame conditions $D, T, B, 4, 5$, such that all preceding conditions are preserved by closures under the following conditions. 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.

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 completeness 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. In fact, the translations that we use for logics without recursion are surprisingly straightforward. 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 worlds of the model (see [5, 6, 14]). We give for each frame condition x and agent α, the axiom ax_α^x:

ax_α^D: $\langle\alpha\rangle\mathtt{tt}$	ax_α^B: $\langle\alpha\rangle[\alpha]p \rightarrow p$	ax_α^5: $\langle\alpha\rangle[\alpha]p \rightarrow [\alpha]p$
ax_α^T: $[\alpha]p \rightarrow p$	ax_α^4: $[\alpha]p \rightarrow [\alpha][\alpha]p$	

For each formula φ and $d \geq 0$, let $Inv_d(\varphi) = \bigwedge_{i \leq d} [\mathrm{AG}]^i \varphi$. Our first translations are straightforwardly defined from the above axioms.

Translation 1 (One-step Translation)**.** Let $A \subseteq \mathrm{AG}$ and let x be one of the frame conditions. For every formula φ, let $d = md(\varphi)$ if $x \neq 4$, and $d = md(\varphi)|\varphi|$, if $x = 4$. We define:

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

Theorem 8. *Let $A \subseteq \mathrm{AG}$, x be one of the frame conditions, and let $\mathbf{L}_1, \mathbf{L}_2$ be logics without recursion operators, such that $\mathbf{L}_1(\alpha) = \mathbf{L}_2(\alpha) + x$ when $\alpha \in A$, and $\mathbf{L}_2(\alpha)$ otherwise, and $\mathbf{L}_2(\alpha)$ 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 $\mathsf{F}_A^x(\varphi)$ is $\mathbf{L}_2$-satisfiable.*

We present here a short proof sketch of this theorem.

Proof sketch. The proof of the "only if" direction is straightforward, as for any agent α with frame condition x, ax_α^x is valid in $\mathbf{L}_1$. Thus the translation holds on any $\mathbf{L}_1$-model that satisfies φ.

The other, and more involved direction requires the construction of an $\mathbf{L}_1$-model for φ from an $\mathbf{L}_2$-model for $\mathsf{F}_A^x(\varphi)$. We make use of the observation that no modal logic formula can describe a model at depth more than the constant d. Therefore, we use the unfolding of the $\mathbf{L}_2$-model, to keep track of the path that one takes to reach a certain state, and that path's length. We then carefully reapply the necessary closures on the accessibility relations and we use induction on its subformulas, to prove that φ is true in the constructed model. We do this as a separate case for each axiom. It is worth noting that we pay special care for the case of $x = 4$ to account for the fact that the translation does not allow the modal depth of the relevant subformulas to decrease with each transition during the induction; and that we needed to include the negations of subformulas of φ in the conjunction of Translation 1, only for the case of $x = 5$. □

Corollary 9. *The satisfiability problem for every logic 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.

Example 2. Let $\varphi_f = \mu X.\Box X$, which requires all paths in the model to be finite, and thus it is not satisfiable in reflexive frames. In Subsection 3.1.2, to translate formulas from reflexive models, we did not need to add the negations of subformulas as conjuncts. In this case, such a translation would give

$$\varphi_t := \varphi_f \wedge Inv((\Box\varphi_f \to \varphi_f) \wedge (\Box\Box\varphi_f \to \Box\varphi_f)).$$

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

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

$$\varphi_f \wedge Inv\left(\bigwedge_{\psi \in \overline{\mathsf{sub}}(\varphi_f)} \Box\psi \to \psi \right)$$

would then yield a formula that is not satisfiable. However, our constructions would not work to prove that such a translation preserves satisfiability. For example, consider $\mu X.\Box(p \to (r \wedge (q \to X)))$, whose translation is satisfied on a pointed 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 $Inv(\langle\alpha\rangle\mathtt{tt})$ directly ensures the seriality of a model.

Translation 2.

$$\mathsf{F}_A^{D^\mu}(\varphi) = \varphi \wedge Inv\big(\bigwedge_{\alpha \in A} \langle\alpha\rangle\mathtt{tt}\big).$$

Theorem 10. *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* $\mathsf{F}_A^{D^\mu}(\varphi)$ *is* $\mathbf{K}_k^\mu$*-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. The operation $\mathsf{F}_A^{T^\mu}(-)$ is defined to be such that

- $\mathsf{F}_A^{T^\mu}([\alpha]\varphi) = [\alpha]\mathsf{F}_A^{T^\mu}(\varphi) \wedge \mathsf{F}_A^{T^\mu}(\varphi)$;
- $\mathsf{F}_A^{T^\mu}(\langle\alpha\rangle\varphi) = \langle\alpha\rangle\mathsf{F}_A^{T^\mu}(\varphi) \vee \mathsf{F}_A^{T^\mu}(\varphi)$;
- and it commutes with all other operations.

Theorem 11. *Let* $\emptyset \neq A \subseteq \text{AG}$*, and 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* $\mathbf{L}_2(\alpha)$ *otherwise, and* $\mathbf{L}_2(\alpha)$ *at most includes frame condition D. Then,* φ *is* $\mathbf{L}_1$*-satisfiable if and only if* $\mathsf{F}_A^{T^\mu}(\varphi)$ *is* $\mathbf{L}_2$*-satisfiable.*

Proof sketch. The "only if" direction is proven by taking the appropriate reflexive closure and showing by induction that the subformulas of φ are preserved. □

Translation 4. The operation $\mathsf{F}_A^{4^\mu}(-)$ is defined to be such that

- $\mathsf{F}_A^{4^\mu}([\alpha]\psi) = \mathit{Inv}([\alpha](\mathsf{F}_A^{4^\mu}(\psi))$,
- $\mathsf{F}_A^{4^\mu}(\langle\alpha\rangle\psi) = \mathit{Eve}(\langle\alpha\rangle(\mathsf{F}_A^{4^\mu}(\psi))$,
- $\mathsf{F}_A^{4^\mu}(-)$ commutes with all other operations.

Theorem 12. *Let $\emptyset \neq A \subseteq$ AG, and let $\mathbf{L}_1, \mathbf{L}_2$ 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 $\mathsf{F}_A^{4^\mu}(\varphi)$ is $\mathbf{L}_2$-satisfiable.*

Proof. If $\mathsf{F}_A^{4^\mu}(\varphi)$ is satisfied in a model $M = (W, R, V)$, let $M' = (W, R^+, V)$, 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, for every environment ρ, $[\![\mathsf{F}_A^{4^\mu}(\psi), \rho]\!]_{\mathscr{M}} = [\![\psi, \rho]\!]_{\mathscr{M}'}$. The other direction is more straightforward. □

In order to produce a similar translation for symmetric frames, we needed to use a more intricate type of construction. Moreover, we only prove the correctness of the following translation for formulas without least-fixed-point operators.

Translation 5. The operation $\mathsf{F}_A^{\mathbf{B}^\mu}(-)$ is defined as

$$\mathsf{F}_A^{\mathbf{B}^\mu}(\varphi) = \varphi \wedge \mathit{Inv}\big([\alpha]\langle\alpha\rangle p \wedge \bigwedge_{\psi \in \overline{\mathsf{sub}}(\varphi)} (\psi \to [\alpha][\alpha](p \to \psi))\big),$$

where p is a new propositional variable, not occurring in φ.

Theorem 13. *Let $\emptyset \neq A \subseteq$ AG, and 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 conditions D, T. Then, a formula φ that has no μX operators is $\mathbf{L}_1$-satisfiable if and only if $\mathsf{F}_A^{\mathbf{B}^\mu}(\varphi)$ is $\mathbf{L}_2$-satisfiable.*

Remark 3. A translation for euclidean frames and for the full syntax on symmetric frames would need different approaches. D'Agostino and Lenzi show in [9] 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 $\mathbf{K}_k^\mu$ guarantee the finite model property to the corresponding logics, they do not apply to multimodal logics with B or 5.

3.2 Embedding $\mathbf{K}_n^\mu$

In this subsection, we present translations from logics with fewer frame conditions to ones with more conditions. This will allow us to prove EXP-completeness 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, $\neg p$, $p \wedge q$, $p \wedge \neg q$, and $\neg p \wedge q$.

Definition 2 (function *next*). *$\mathit{next}(p \wedge q) = p \wedge \neg q$, $\mathit{next}(p \wedge \neg q) = \neg p \wedge q$, and $\mathit{next}(\neg p \wedge q) = p \wedge q$; and $\mathit{next}(p) = \neg p$ and $\mathit{next}(\neg p) = p$.*

We use a uniform translation from $\mathbf{K}_k^\mu$ to any logic with a combination of conditions D, T, B.

Translation 6. The operation $\mathsf{F}_A^{\mathbf{K}^\mu}(-)$ on formulas is defined such that:

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

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

Theorem 14. *Let $\emptyset \neq A \subseteq \mathrm{AG}$, $|\mathrm{AG}| = k$, and let $\mathbf{L}$ be such that $\mathbf{L}(\alpha)$ includes only frame conditions from $\mathbf{D}, \mathbf{T}, \mathbf{B}$ when $\alpha \in A$, and $\mathbf{L}(\alpha) = \mathbf{K}$ otherwise. Then, φ is $\mathbf{K}_k^\mu$-satisfiable if and only if $\mathsf{F}_A^{\mathbf{K}^\mu}(\varphi)$ is $\mathbf{L}$-satisfiable.*

The proof of Theorem 14 can be found in [23]. It is worth noting that the "if" direction uses the symmetric closure to construct a new model, while the "only if" direction requires the introduction of new states that behave as each original state in the model.

3.3 Complexity results

We observe that our translations all result in formulas of size at most linear with respect to the original. The exceptions are Translations 1 and 5, which have a quadratic cost.

Corollary 15. *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 10, 11, 12, and 14. □

Corollary 16. *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 10, 11, 13, and 14. □

4 Tableaux for $\mathbf{L}_k^\mu$

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 [21] and the tableaux of Fitting [16] and Massacci [24] for modal logic. We can use Kozen's finite model theorem [21] to help us ensure the termination of the tableau for some of these logics.

Theorem 17 ([21])**.** *There is a computable $\kappa : \mathbb{N} \to \mathbb{N}$, such that every $\mathbf{K}_k^\mu$-satisfiable formula φ is satisfied in a model with at most $\kappa(|\varphi|)$ states.*[5]

Corollary 18. *If $\mathbf{L}$ only has frame conditions $D, T, 4$, then there is a computable $\kappa : \mathbb{N} \to \mathbb{N}$, such that every $\mathbf{L}$-satisfiable formula φ is satisfied in a model with at most $\kappa(|\varphi|)$ states.*

[5] The tableau in [21] 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 Section 3.

$$\frac{\sigma\ \pi X.\varphi}{\sigma\ \varphi}\,(\mathsf{fix}) \qquad \frac{\sigma\ X}{\sigma\ \mathsf{fx}(X)}\,(\mathsf{X}) \qquad \frac{\sigma\ \varphi\vee\psi}{\sigma\ \varphi \mid \sigma\ \psi}\,(\mathsf{or}) \qquad \frac{\sigma\ \varphi\wedge\psi}{\begin{array}{c}\sigma\ \varphi\\ \sigma\ \psi\end{array}}\,(\mathsf{and})$$

$$\frac{\sigma\ [\alpha]\varphi}{\sigma.\alpha\langle\psi\rangle\ \varphi}\,(\mathsf{B}) \qquad \frac{\sigma\ \langle\alpha\rangle\varphi}{\sigma.\alpha\langle\varphi\rangle\ \varphi}\,(\mathsf{D}) \qquad \frac{\sigma\ [\alpha]\varphi}{\sigma.\alpha\langle\varphi\rangle\ \varphi}\,(\mathsf{d}) \qquad \frac{\sigma\ [\alpha]\varphi}{\sigma.\alpha\langle\psi\rangle\ [\alpha]\varphi}\,(\mathsf{4})$$

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.

$$\frac{\sigma.\alpha\langle\psi\rangle\ [\alpha]\varphi}{\sigma\ [\alpha]\varphi}\,(\mathsf{B5}) \qquad \frac{\sigma.\alpha\langle\psi\rangle\ \langle\alpha\rangle\varphi}{\sigma.\alpha\langle\psi\rangle.\alpha\langle\varphi\rangle\ \varphi}\,(\mathsf{D5}) \qquad \frac{\sigma.\alpha\langle\psi\rangle\ [\alpha]\varphi}{\sigma\ \varphi}\,(\mathsf{b}) \qquad \frac{\sigma\ [\alpha]\varphi}{\sigma\ \varphi}\,(\mathsf{t})$$

$$\frac{\sigma.\alpha\langle\psi\rangle\ [\alpha]\varphi}{\sigma.\alpha\langle\psi'\rangle\ [\alpha]\varphi}\,(\mathsf{B55}) \qquad \frac{\sigma.\alpha\langle\psi\rangle.\alpha\langle\psi'\rangle\ \langle\alpha\rangle\varphi}{\sigma.\alpha\langle\psi\rangle.\alpha\langle\varphi\rangle\ \varphi}\,(\mathsf{D55}) \qquad \frac{\sigma.\alpha\langle\psi\rangle\ [\alpha]\varphi}{\sigma\ [\alpha]\varphi}\,(\mathsf{b4})$$

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 2: The tableau rules for $\mathbf{L} = \mathbf{L}_n^{\mu}$

Proof. Immediately, from Theorems 17, 10, 11, and 12, and Lemma 7. □

Remark 4. We note that not all modal logics with recursion have a finite model property – see Remark 3.

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.

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. We say that the prefix σ is α-flat when 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 [18,25].

The tableau rules that we use appear in Table 2. 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 rule (B5), (D5), and (D55), $\mathbf{L}(\alpha)$ must have condition 5; for (b) $\mathbf{L}(\alpha)$ must have condition B; and for (b4) $\mathbf{L}(\alpha)$ must have both B and 4. Rule (or) is the only rule that splits the current tableau branch into two. A tableau branch is propositionally closed when $\sigma\ \mathtt{ff}$ or both $\sigma\ p$ 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 define the relation $\xrightarrow{X}$ on prefixed formulas in a tableau branch as $\chi_1 \xrightarrow{X} \chi_2$, if $\frac{\chi_1}{\chi_2}$ is a tableau rule and χ_1 is not of the form $\sigma\ Y$, 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 $\kappa : \mathbb{N} \to \mathbb{N}$, 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 $\sigma\ \psi$, $|\sigma| > |\textsc{Ag}| \cdot \kappa(|\varphi|)^{|\varphi|^2} \cdot 2^{2|\varphi|+1}$. 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 2. A tableau proof for φ is a closed tableau for the negation of φ.

Theorem 19 (Soundness, Completeness, and Termination of $\mathbf{L}^{\mu}_{k}$-Tableaux)**.** *From the following, the first two are equivalent for any formula $\varphi \in L$ and any logic $\mathbf{L}$. Furthermore, if there is a $\kappa : \mathbb{N} \to \mathbb{N}$, such that every $\mathbf{L}$-satisfiable formula φ is satisfied in a model with at most $\kappa(|\varphi|)$ states, then all the following are equivallent.*

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.*

Proof sketch. The direction from 1 to 2 uses the usual model construction, but where one needs to take into account the fixed-point formulas; the direction from 2 to 3 uses techniques and results from [21,28], including Corollary 18; and the direction from 3 to 1 shows how to detect appropriate parts of the branch to repeat until we safely get a maximal branch. □

Corollary 20. *$\mathbf{L}$-tableaux are sound and complete for $\mathbf{L}$.*

Example 3. Let $\textsc{Ag} = \{a, b\}$ and $\mathbf{L}$ be a logic, such that $\mathbf{L}(a) = \mathbf{K}^{\mu}$ and $\mathbf{L}(b) = \mathbf{K5}^{\mu}$. Let

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

As we see in Figure 2, 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.

5 Conclusions

We studied multi-modal logics with recursion. These logics mix the frame conditions from epistemic modal logic, and the recursion of the μ-calculus. We gave simple translations among these logics that connect their satisfiability problems. This allowed us to offer complexity bounds for satisfiability and to prove certain finite model results. We also presented a sound and complete tableau that has termination guarantees, conditional on a logic's finite model property.

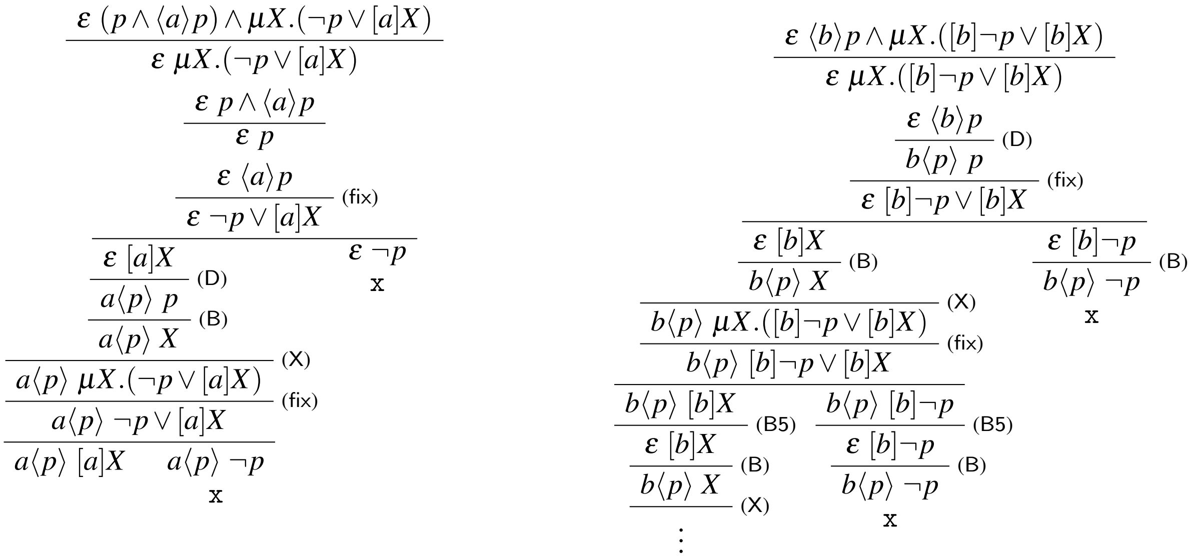

Figure 2: Tableaux for φ_1 and φ_2. The dots represent that the tableau keeps repeating as from the identical node above. The x mark represents a propositionally closed branch.

Conjectures and Future Work We currently do not posses full translations for the cases of symmetric and euclidean frames. What is interesting is that we also do not have a counterexample to prove that the translations that we already have, as well as other attempts, are *not* correct. In the case of symmetric frames, we have managed to prove that our construction works for formulas without least-fixed-point operators. A translation for euclidean frames and for the full syntax on symmetric frames is left as future work. We know that we cannot use the same model constructions that preserve the finiteness of the model as in Subection 3.1.3 (see Remark 3).

We do not prove the finite model property on all logics. We note that although it is known that logics with recursion with at least two agents with either B or 5 do not have this property (see 3, [9]), the situation is unclear if there is only one such agent.

We further conjecture that it is not possible to prove EXP-completenes for all the single-agent cases. Specifically, we expect $\mathbf{K4}^{\mu}$-satisfiability to be in PSPACE, similarly to how $\mathbf{K5}^{\mu}$-satisfiability is in NP [9]. As such, we do not expect Translation 6 to be correct for these cases.

The model checking problem for the μ-calculus is an important open problem. 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. We are not aware of a way to use our translations to solve model checking more efficiently.

As, to the best of our knowledge, most of the logics described in this paper have not been explicitly defined before, with notable exceptions such as [3, 9, 11], they also lack any axiomatizations and completeness theorems. We do expect the classical methods from [17, 21, 22] and others to work out in these cases as well. However it would be interesting to 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 [4, 19], 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.

References

[1] Luca Aceto, Antonis Achilleos, Adrian Francalanza & Anna Ingólfsdóttir (2020): *The complexity of identifying characteristic formulae*. J. Log. Algebraic Methods Program. 112, p. 100529. Available at `https://doi.org/10.1016/j.jlamp.2020.100529`.

[2] Luca Alberucci & Alessandro Facchini (2009): *The modal μ-calculus hierarchy over restricted classes of transition systems*. The Journal of Symbolic Logic 74(4), p. 1367–1400, doi:10.2178/jsl/1254748696.

[3] Luca Alberucci & Alessandro Facchini (2009): *The modal μ-calculus hierarchy over restricted classes of transition systems*. The Journal of Symbolic Logic 74(4), pp. 1367–1400, doi:10.2178/jsl/1254748696.

[4] Jon Barwise (1988): *Three views of common knowledge*. In: Proceedings of the 2nd Conference on Theoretical Aspects of Reasoning About Knowledge, TARK '88, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 365–379. Available at `http://dl.acm.org/citation.cfm?id=1029718.1029753`.

[5] Patrick Blackburn, Johan van Benthem & Frank Wolter (2006): *Handbook of Modal Logic*. Studies in Logic and Practical Reasoning 3, Elsevier Science.

[6] Patrick Blackburn, Maarten de Rijke & Yde Venema (2001): *Modal Logic*. Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, doi:10.1017/cbo9781107050884.

[7] Laura Bozzelli, Bastien Maubert & Sophie Pinchinat (2014): *Unifying Hyper and Epistemic Temporal Logic*. abs/1409.2711, doi:10.1007/978-3-662-46678-0_11. arXiv:1409.2711.

[8] Michael R. Clarkson & Fred B. Schneider (2010): *Hyperproperties*. Journal of Computer Security 18(6), p. 1157–1210, doi:10.3233/JCS-2009-0393.

[9] Giovanna D'Agostino & Giacomo Lenzi (2013): *On modal μ-calculus in S5 and applications*. Fundamenta Informaticae 124(4), pp. 465–482, doi:10.3233/FI-2013-844.

[10] Giovanna D'Agostino & Giacomo Lenzi (2018): *The μ-Calculus Alternation Depth Hierarchy is infinite over finite planar graphs*. Theoretical Computer Science 737, pp. 40–61, doi:10.1016/j.tcs.2018.04.009. Available at `https://www.sciencedirect.com/science/article/pii/S0304397518302317`.

[11] Giovanna D'Agostino & Giacomo Lenzi (2010): *On the μ-calculus over transitive and finite transitive frames*. Theoretical Computer Science 411(50), pp. 4273–4290, doi:10.1016/j.tcs.2010.09.002. Available at `https://www.sciencedirect.com/science/article/pii/S030439751000469X`.

[12] Giovanna D'Agostino & Giacomo Lenzi (2015): *On the modal μ-Calculus over finite symmetric graphs*. Mathematica Slovaca 65(4), pp. 731–746, doi:10.1515/ms-2015-0052.

[13] E Allen Emerson, Charanjit S Jutla & A Prasad Sistla (2001): *On model checking for the μ-calculus and its fragments*. Theoretical Computer Science 258(1-2), pp. 491–522, doi:10.1016/S0304-3975(00)00034-7.

[14] Ronald Fagin, Joseph Y. Halpern, Yoram Moses & Moshe Y. Vardi (1995): *Reasoning About Knowledge*. The MIT Press, doi:10.7551/mitpress/5803.001.0001.

[15] Michael J. Fischer & Richard E. Ladner (1979): *Propositional dynamic logic of regular programs*. Journal of computer and system sciences 18(2), pp. 194–211, doi:10.1016/0022-0000(79)90046-1.

[16] Melvin Fitting (1972): *Tableau methods of proof for modal logics*. Notre Dame Journal of Formal Logic 13(2), pp. 237–247, doi:10.1305/ndjfl/1093894722.

[17] Joseph Y. Halpern & Yoram Moses (1992): *A guide to completeness and complexity for modal logics of knowledge and belief*. Artificial Intelligence 54(3), pp. 319–379, doi:10.1016/0004-3702(92)90049-4.

[18] Joseph Y. Halpern & Leandro Chaves Rêgo (2007): *Characterizing the NP-PSPACE gap in the satisfiability problem for modal logic*. Journal of Logic and Computation 17(4), pp. 795–806, doi:10.1093/logcom/exm029.

[19] Gilbert Harman (1977): *Review of linguistic behavior by Jonathan Bennett*. Language 53, pp. 417–424, doi:10.1353/lan.1977.0036.

[20] Marcin Jurdziński (1998): *Deciding the winner in parity games is in UP $\cap$ co-UP*. Information Processing Letters 68(3), pp. 119–124, doi:10.1016/S0020-0190(98)00150-1.

[21] Dexter Kozen (1983): *Results on the propositional μ-calculus*. Theoretical Computer Science 27(3), pp. 333–354, doi:10.1016/0304-3975(82)90125-6.

[22] Richard E. Ladner (1977): *The computational complexity of provability in systems of modal propositional logic*. SIAM Journal on Computing 6(3), pp. 467–480, doi:10.1137/0206033.

[23] Elli Anastasiadi Adrian Francalanza Anna Ingólfsdóttir Luca Aceto, Antonis Achilleos: *Complexity through Translations for Modal Logic with Recursion*. Available at http://icetcs.ru.is/movemnt/papers/ComplexityTranslationsMLRecFullGand.pdf.

[24] Fabio Massacci (1994): *Strongly analytic tableaux for normal modal logics*. In: CADE, pp. 723–737, doi:10.1007/3-540-58156-1_52.

[25] Michael C. Nagle & S. K. Thomason (1985): *The extensions of the modal logic K5*. Journal of Symbolic Logic 50(1), pp. 102—-109, doi:10.2307/2273793.

[26] Vaughan R. Pratt (1981): *A decidable mu-calculus: Preliminary report*. In: 22nd Annual Symposium on Foundations of Computer Science (SFCS 1981), IEEE, doi:10.1109/sfcs.1981.4.

[27] Mikhail Rybakov & Dmitry Shkatov (2018): *Complexity of finite-variable fragments of propositional modal logics of symmetric frames*. Logic Journal of the IGPL 27(1), pp. 60–68, doi:10.1093/jigpal/jzy018.

[28] Wieslaw Zielonka (1998): *Infinite games on finitely coloured graphs with applications to automata on infinite trees*. Theoretical Computer Science 200(1-2), pp. 135–183, doi:10.1016/S0304-3975(98)00009-7.