Satisfiability-checking of modal logic with recursion via translations and tableaux*

Luca Aceto[1,2], Antonis Achilleos[1], Elli Anastasiadi[3], Adrian Francalanza[4], and Anna Ingólfsdóttir

[1] ICE-TCS, Department of Computer Science, Reykjavik University, Iceland
[2] Gran Sasso Science Institute L'Aquila, Italy
[3] Department of Information Technology, Uppsala University, Sweden
[4] Dept. of Computer Science, ICT, University of Malta, Msida, Malta
luca.aceto@gssi.it, antonios@ru.is, elli.anastasiadi@it.uu.se, adrian.francalanza@um.edu.mt, annai@ru.is

Abstract

In this talk proposal, we discuss a method based on tableau derivations that may not terminate, for achieving decidability and upper complexity bounds for a general family of modal logics with recursion. We show how to use this method to prove the decidability of certain modal logics that do not have a finite model property.

1 Introduction

In this talk we will be studying a family of multi-modal logics with fixed-point operators that are interpreted over restricted classes of Kripke models. The abstract is aimed to be a short summary of the methods and results that the authors recently gave in [2]. One can consider these logics as extensions of the usual multi-agent logics of knowledge and belief [10] by adding recursion to their syntax or of the μ-calculus [17] by interpreting formulas over different classes of frames and thus giving an epistemic interpretation to the modalities. We are concerned with the complexity of the satisfiability problem for these logics. Namely, given a formula φ, how much time it takes to determine whether there exists a model $\mathcal{M} \models \varphi$.

Modal logic comes in several variations [6]. Some of these, such as multi-modal logics of knowledge and belief [10], 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 [17] 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. We explore the situation where one allows recursion (*i.e.* fixed-point) operators in a multi-modal language and imposes restrictions on the models.

Satisfiability for the μ-calculus is known to be EXP-complete [17]. For the modal logics between $\mathbf{K}$ and $\mathbf{S5}$ the problem is PSPACE-complete or NP-complete, depending on the presence of Negative Introspection [14,18]. We discuss the two main methods that we developed in [2] for proving complexity bounds for the satisfiability problem for different logics. Our first method is a *translation*, and it follows the natural intuition where a formula in one logic is projected to a formula in another logic, preserving its satisfiability. We demonstrate how this straightforward method can prove the known upper complexity bounds for modal logics without recursion.

We then present tableaux for the discussed logics, based on the ones by Kozen for the μ-calculus [17], and by Fitting and Massacci for modal logic [12,21]. For some of our logics with

*The work reported in this paper is supported by the project 'Mode(l)s of Verification and Monitorability' (MoVeMent) (grant no 217987) of the Icelandic Research Fund.

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

Table 1: Semantics of formulas on model $\mathcal{M} = (W, R, V)$, which we omit from the notation.

axiom 5 ($\langle\alpha\rangle\varphi \to [\alpha]\langle\alpha\rangle\varphi$), or B ($\varphi \to [\alpha]\langle\alpha\rangle\varphi$), the tableaux may not terminate, as these logics have no finite-model property [8]. We give a general satisfiability-preserving translation from each logic to the μ-calculus, using our tableaux, which describes a tableau branch with an exponentially larger μ-calculus formula, establishing a 2EXP-upper bound for all our logics.

2 Modal logics with recursion

We consider formulas constructed from the following grammar:

$$\varphi, \psi ::= p \quad \mid \neg p \quad \mid \mathtt{tt} \quad \mid \mathtt{ff} \quad \mid X \quad \mid \varphi \wedge \psi \quad \mid \varphi \vee \psi \quad \mid \langle\alpha\rangle\varphi \quad \mid [\alpha]\varphi \quad \mid \mu X.\varphi \quad \mid \nu X.\varphi,$$

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.

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, $R \subseteq W \times \mathrm{Ag} \times W$ is a transition relation, and $V : W \to 2^{\mathrm{PVar}}$ 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 $(u, \alpha, v) \in R$.

Formulas are evaluated in the context of an *environment* $\rho : \mathrm{LVar} \to 2^W$, 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 $[\![-]\!]_{\mathcal{M}}$, defined in Table 1.

Without further restrictions, the resulting logic is the μ-calculus [17]. If $|\mathrm{Ag}| = k \in \mathbb{N}^+$ and we allow no recursive operators and variables, we have the basic modal logic $\mathbf{K}_k$, and further restrictions on the frames can result in a variety of modal logics (see, for instance, [5]). We give names to the following frame conditions, or frame constraints, for an agent $\alpha \in \mathrm{Ag}$:

D: R_α is *serial* ($\forall s \exists t, sR_\alpha t$); *B*: R_α is *symmetric*; 5: R_α is *euclidean* (if $sR_\alpha t$ and $sR_\alpha r$, then $tR_\alpha r$).

T: R_α is *reflexive* ($\forall s, sR_\alpha s$); 4: R_α is *transitive*;

Each frame condition x for agent α is associated with an axiom ax^x_α, such that whenever a model has condition x, every substitution instance of ax^x_α is satisfied in all its states (see [5,6,10]):

$$\mathsf{ax}^D_\alpha = \langle\alpha\rangle\mathtt{tt}; \qquad \mathsf{ax}^T_\alpha = [\alpha]p \to p; \qquad \mathsf{ax}^B_\alpha = \langle\alpha\rangle[\alpha]p \to p;$$
$$\mathsf{ax}^4_\alpha = [\alpha]p \to [\alpha][\alpha]p; \text{ and } \qquad \mathsf{ax}^5_\alpha = \langle\alpha\rangle[\alpha]p \to [\alpha]p.$$

We consider logics interpreted over models that satisfy a combination of these constraints for each agent. For each logic $\mathbf{L}$ and agent α, $\mathbf{L}(\alpha)$ is the single-agent logic that includes exactly the frame conditions that $\mathbf{L}$ has for α.

3 Translations for Recursion-free formulas

Translations are functions that transform each formula to another. We require that translations preserve satisfiability, allowing us to transfer decidability results among logics. We also require that they only increment the size of the formula by a polynomial factor. This is because we will use the translations to prove complexity bounds as well. Finally, we want translations to be compositional. Compositionality ensures that we can apply a sequence of translation steps, resulting in a composite translation that has the above good properties.

We fix an order to the frame conditions: $D, T, B, 4, 5$. We present a straightforward and uniform translation for recursion-free logics. Let $\overline{\mathsf{sub}}(\varphi) = \{\psi, \neg\psi \mid \psi \text{ is a subformula of } \varphi\}$.

Translation 3.1 (One-step Translation)**.** Let $A \subseteq \textsc{Ag}$ and let x be one of the frame conditions. For every formula φ, let $d = md(\varphi)$ if $x \neq 4, 5$, and $d = md(\varphi)|\varphi|$, if $x = 4$ or $x = 5$. We define:

$$\mathsf{F}^x_A(\varphi) = \varphi \wedge \bigwedge_{k \leq d} \bigwedge_{\alpha_1 \in \textsc{Ag}} [\alpha_1] \cdots \bigwedge_{\alpha_k \in \textsc{Ag}} [\alpha_k] \left(\bigwedge_{\substack{\psi \in \overline{\mathsf{sub}}(\varphi) \\ \alpha \in A}} \mathsf{ax}^x_\alpha[\psi/p] \right).$$

Theorem 1. *Let $A \subseteq \textsc{Ag}$, 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. 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 $\mathsf{F}^x_A(\varphi)$ is $\mathbf{L}_2$-satisfiable.*

In the above we see that indeed, a translation preserves satisfiability and is only changing the size of the formula by a small factor. The compositionality gives us that for a logic $\mathbf{L}$ with multiple frame restrictions, one would have to apply these translations in series to acquire a formula that is satisfiable over general frames if and only if the original was $\mathbf{L}$-satisfiable.

Corollary 1. *If $\mathbf{L}$ has no recursion operators, then $\mathbf{L}$-satisfiability is in* PSPACE*.*

4 Tableaux and General Upper Bounds

Our tableaux are based on the ones given by Kozen in [17], and are extended similarly to the tableaux of Fitting [12] and Massacci [21] for taking into account the different conditions of modal logic. 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 successful branch represents a corresponding model.

Our tableaux use *prefixed formulas*, that is, formulas of the form $\sigma\ \varphi$, where $\sigma \in (\textsc{Ag} \times L)^*$ 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 $(\alpha, \psi) \in \textsc{Ag} \times L$. 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 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 model, and a prefixed formula $\sigma\ \varphi$ declares that φ is true in that state. The tableau rules appear in Table 2.

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. 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 $\sigma\ Y$, where $X < Y$. If in a branch there is a $\xrightarrow{X}$-sequence

$$\frac{\sigma\ \pi X.\varphi}{\sigma\ \varphi}\,(\mathsf{fix})\quad \frac{\sigma\ X}{\sigma\ \mathsf{fx}(X)}\,(\mathsf{X})\quad \frac{\sigma\ [\alpha]\varphi}{\sigma.\alpha\langle\psi\rangle\ \varphi}\,(\mathsf{B})\quad \frac{\sigma\ \langle\alpha\rangle\varphi}{\sigma.\alpha\langle\varphi\rangle\ \varphi}\,(\mathsf{D})\quad \frac{\sigma\ [\alpha]\varphi}{\sigma.\alpha\langle\varphi\rangle\ \varphi}\,(\mathsf{d})$$

$$\frac{\sigma\ [\alpha]\varphi}{\sigma.\alpha\langle\psi\rangle\ [\alpha]\varphi}\,(\mathsf{4})\quad \frac{\sigma.\alpha\langle\psi\rangle\ [\alpha]\varphi}{\sigma\ \varphi}\,(\mathsf{b})\quad \frac{\sigma\ [\alpha]\varphi}{\sigma\ \varphi}\,(\mathsf{t})\quad \frac{\sigma.\alpha\langle\psi\rangle\ [\alpha]\varphi}{\sigma\ [\alpha]\varphi}\,(\mathsf{b4})$$

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

$$\frac{\sigma.\alpha\langle\psi\rangle\ [\alpha]\varphi}{\sigma\ [\alpha]\varphi}\,(\mathsf{B5})\quad \frac{\sigma.\alpha\langle\psi\rangle\ \langle\alpha\rangle\varphi}{\sigma.\alpha\langle\psi\rangle.\alpha\langle\varphi\rangle\ \varphi}\,(\mathsf{D5})\quad \frac{\sigma.\alpha\langle\psi\rangle\ [\alpha]\varphi}{\sigma.\alpha\langle\psi'\rangle\ [\alpha]\varphi}\,(\mathsf{B55})\quad \frac{\sigma.\alpha\langle\psi\rangle.\alpha\langle\psi'\rangle\ \langle\alpha\rangle\varphi}{\sigma.\alpha\langle\psi\rangle.\alpha\langle\varphi\rangle\ \varphi}\,(\mathsf{D55})$$

where, for rule (B55), $\sigma.\alpha\langle\psi'\rangle$ already appears 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 $\mathbf{L} = \mathbf{L}_n^\mu$. The propositional cases are omitted.

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.

Theorem 2 (Soundness and Completeness of $\mathbf{L}_k^\mu$-Tableaux, [1])**.** *For every formula φ and logic $\mathbf{L}$, φ has a maximal $\mathbf{L}$-tableau with an open branch if and only if φ is $\mathbf{L}$-satisfiable.*

Although for many of our logics, we can give sound, complete, and terminating tableaux, in general it is possible for a tableau to be non-terminating. Moreover, some of our logics do not have a finite model property. To give a general upper bound for the satisfiability problem for our family of modal logics with recursion, we devised the following strategy. From the above, we have, possibly non-terminating, sound and complete tableaux for all our logics. Moreover, we know that μ-calculus satisfiability can be decided in exponential time. Therefore, our idea is to encode the tableaux themselves as μ-calculus formulas, interpreted over arbitrary frames. These formulas assert that a satisfying model encodes an open branch. The total overhead of this construction was that the initial tableaux constructed from a formula is exponential larger from it, and the final satisfiability checking for the formula that represented this tableaux costs exponential time. Therefore the resulting complexity upper bound is a double exponential.

We avoid presenting the full extensive construction of the formula that describes the tableau branch. We present the idea in Figure 1. The reader can read [2] for more details. A summary of our results from [2] can be seen in Table 3. The highlighted result is the ones that occurred due to the final idea of translating tableaux into formulas.

Theorem 3. *$\mathbf{L}$-satisfiability is in* 2EXP*.*

5 Conclusion and future work

We presented a simple translation method to prove the known PSPACE upper bound for the complexity of satisfiability for all logics without recursion. For the ones with recursion, we presented sound and complete tableaux, which in turn we encoded as μ-calculus formulas, thus proving that satisfiability is in 2EXP for all our logics.

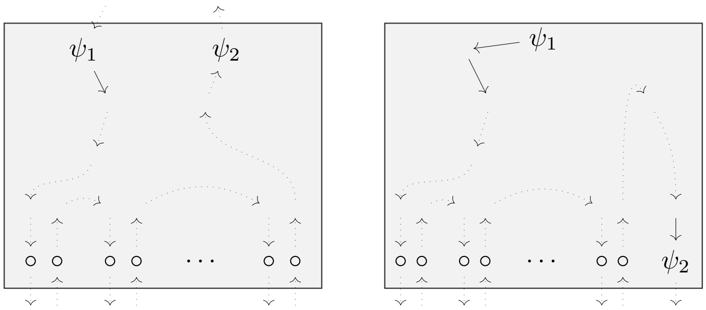

Figure 1: A finite $\xrightarrow{X}$-path from $\sigma\ \psi_1$ to $\sigma\ \psi_2$ may visit other tableau prefixes, and an infinite $\xrightarrow{X}$-path from $\sigma\ \psi_1$ may include finite segments that visit other prefixes, before continuing with an infinite path from a formula ψ_2. Each square area represents a tableau prefix and the part of an infinite $\xrightarrow{X}$-path that visits this prefix. We can describe this local behavior with exponentially many propositional variables, and the infinite path with a μ-calculus formula.

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

Table 3: The updated summary of the complexity of satisfiability checking for various modal logics with recursion. Our additional contributions from [2] are highlighted.

As Table 3 indicates, we currently do not have a tight complexity bound for the case of the multi-agent μ-calculus over symmetric or euclidean frames. 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 [7, 11, 16, 19, 20]. 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 [3, 8, 9], they also lack any axiomatizations and completeness theorems. We do expect the classical methods from [13, 17, 18] 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.

Finally, 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, 15], we consider the logics we presented to be natural extensions of modal logic. We are interested in exploring what other natural concepts we can define with this enlarged language.

References

[1] Luca Aceto, Antonis Achilleos, Elli Anastasiadi, Adrian Francalanza, and Anna Ingólfsdóttir. 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.

[2] Luca Aceto, Antonis Achilleos, Elli Anastasiadi, Adrian Francalanza, and Anna Ingólfsdóttir. Complexity results for modal logic with recursion via translations and tableaux. *CoRR*, abs/2306.16881, 2023.

[3] 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.

[4] 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.

[5] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic.* Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2001.

[6] Patrick Blackburn, Johan van Benthem, and Frank Wolter. *Handbook of Modal Logic*, volume 3 of *Studies in Logic and Practical Reasoning.* Elsevier Science, 2006.

[7] 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.

[8] Giovanna D'Agostino and Giacomo Lenzi. On modal μ-calculus in s5 and applications. *Fundamenta Informaticae*, 124(4):465–482, 2013.

[9] Giovanna D'Agostino and Giacomo Lenzi. On the μ-calculus over transitive and finite transitive frames. *Theoretical Computer Science*, 411(50):4273–4290, 2010.

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

[11] 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.

[12] Melvin Fitting. Tableau methods of proof for modal logics. *Notre Dame Journal of Formal Logic*, 13(2):237–247, 1972.

[13] 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.

[14] 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.

[15] Gilbert Harman. Review of linguistic behavior by Jonathan Bennett. *Language*, 53:417–424, 1977.

[16] 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.

[17] Dexter Kozen. Results on the propositional μ-calculus. *TCS*, 27:333–354, 1983.

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

[19] 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.

[20] 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.

[21] Fabio Massacci. Strongly analytic tableaux for normal modal logics. In *CADE*, pages 723–737, 1994.