The Complexity of Recursion in Modal Logic: First Steps*

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

[1] School of Computer Science, Reykjavik University, Reykjavik, Iceland
[2] Gran Sasso Science Institute, L'Aquila, Italy
[3] Dept. of Computer Science, ICT, University of Malta, Msida, Malta

Abstract

We consider systems of Modal Logic with fixpoint operators and examine their complexity. These can be thought of as extensions of the μ-calculus with modal axioms, or of the various classical modal logics with recursion operators.

1 Introduction

In this paper we explore how the complexity of the satisfiability problem for a (multi-)modal logic is affected when recursive operators are added to its syntax. Modal Logic comes in several variations [4]. Some of these, such as multi-modal logics of knowledge and belief [6], are of particular interest to Epistemology and other areas of application. Semantically, the classical modal logics that are used in epistemic (but also other) contexts result from imposing certain restrictions on their models. On the other hand, the modal μ-calculus [12] can be seen as an extension of the simplest normal modal logic **K** with greatest and least fixpoint operators. We explore the situation where one allows both recursion operators in the language and imposes restrictions on the semantics, in a multi-modal setting.

We are interested in the complexity of satisfiability for the resulting logics. Satisfiability for the plain μ-calculus is known to be EXP-complete [12], while for the modal logics between **K** and **S5** the problem is PSPACE-complete or NP-complete, depending on whether they have Negative Introspection [13, 9]. In the multi-modal case, satisfiability for those modal logics jumps to PSPACE-completeness [8].

We discover that in the unimodal case, the logics with Negative Introspection remain NP-complete, even when we add recursion. On the other hand, we see that when the logic has at least two kinds of boxes and the recursive operators its satisfiability problem becomes EXP-hard. We consider this paper a first glance into this topic. Therefore, we conclude with a set of open questions.

2 Background

This section introduces the logics that we focus on and the necessary background on the complexity of Modal Logic and the μ-calculus. We start by defining the formulae of our logics.

Definition 1. *We consider formulae constructed from the following grammar:*

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

*This research was supported by the project "TheoFoMon: Theoretical Foundations for Monitorability" (grant number: 163406-051) and the project "Epistemic Logic for Distributed Runtime Monitoring" (grant number: 184940-051) of the Icelandic Research Fund.

$$
\begin{aligned}
&[\![\texttt{tt},\rho]\!]_{\mathcal{M}} = \textsc{Prc}, \qquad [\![\texttt{ff},\rho]\!]_{\mathcal{M}} = \emptyset, \qquad [\![p,\rho]\!]_{\mathcal{M}} = \{s \mid p \in V(s)\}, \qquad [\![\neg p,\rho]\!]_{\mathcal{M}} = \textsc{Prc}\backslash[\![p,\rho]\!]_{\mathcal{M}},\\
&[\![[\alpha]\varphi,\rho]\!]_{\mathcal{M}} = \left\{s \mid \forall t.\ s \xrightarrow{\alpha} t \text{ implies } t \in [\![\varphi,\rho]\!]_{\mathcal{M}}\right\}, \quad [\![\varphi_1\wedge\varphi_2,\rho]\!]_{\mathcal{M}} = [\![\varphi_1,\rho]\!]_{\mathcal{M}} \cap [\![\varphi_2,\rho]\!]_{\mathcal{M}},\\
&[\![\langle\alpha\rangle\varphi,\rho]\!]_{\mathcal{M}} = \left\{s \mid \exists t.\ s \xrightarrow{\alpha} t \text{ and } t \in [\![\varphi,\rho]\!]_{\mathcal{M}}\right\}, \qquad [\![\varphi_1\vee\varphi_2,\rho]\!]_{\mathcal{M}} = [\![\varphi_1,\rho]\!]_{\mathcal{M}} \cup [\![\varphi_2,\rho]\!]_{\mathcal{M}},\\
&[\![\mu X.\varphi,\rho]\!]_{\mathcal{M}} = \bigcap\left\{S \mid S \supseteq [\![\varphi,\rho[X\mapsto S]]\!]_{\mathcal{M}}\right\}, \qquad [\![X,\rho]\!]_{\mathcal{M}} = \rho(X),\\
&[\![\nu X.\varphi,\rho]\!]_{\mathcal{M}} = \bigcup\left\{S \mid S \subseteq [\![\varphi,\rho[X\mapsto S]]\!]_{\mathcal{M}}\right\}.
\end{aligned}
$$

Table 1: LTS semantics

where X comes from a countably infinite set of logical variables, $\textsc{LVr}$*, α from a finite set of actions,* $\textsc{Act}$*, and p from a finite set of propositional variables,* $\textsc{PVr}$*. When $\textsc{Act} = \{\alpha\}$, we may use $\Box\varphi$ instead of $[\alpha]\varphi$, and $\Diamond\varphi$ instead of $\langle\alpha\rangle\varphi$. We may write $[\textsc{Act}]\varphi$ to mean $\bigwedge_{\alpha\in\textsc{Act}}[\alpha]\varphi$.*

We interpret formulae on the states of a labelled transition system (LTS). An LTS, or model, is a quadruple $\langle\textsc{Prc},\textsc{Act},\rightarrow,V\rangle$ where $\textsc{Prc}$ is a set of states or processes, $\textsc{Act}$ is the set of actions, $\rightarrow\ \subseteq \textsc{Prc}\times\textsc{Act}\times\textsc{Prc}$ is a transition relation, and $V : \textsc{Prc} \rightarrow 2^{\textsc{PVr}}$ determines on which states a propositional variable is true. For simplicity, we assume that $\textsc{Prc}$ is always finite.

State $\texttt{nil}$ represents any state that cannot transition anywhere: $\forall\alpha\forall s.\ \texttt{nil} \not\xrightarrow{\alpha} s$. The set of states that are reachable from $s \in \textsc{Prc}$ by any sequence of zero or more transitions is called $\mathit{Reach}(s)$, the size of s is $|s| = |\mathit{Reach}(s)|$, and $|\varphi|$ is the length of φ as a string of symbols. All our complexity results are with respect to these measures.

Formulae are evaluated in the context of an LTS $\mathcal{M}$ and an environment, $\rho : \textsc{LVar} \rightarrow 2^{\textsc{Prc}}$, which gives values to the logical variables. For an environment ρ, variable X, and set $S \subseteq \textsc{Prc}$, 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 formulae is given through a function $[\![-]\!]_{\mathcal{M}}$, defined in Table 1. The negation $\neg\varphi$ of a formula and implication $\varphi \supset \psi$ (to be read as "φ implies ψ") are constructed as usual, where $[\![\neg X,\rho]\!]_{\mathcal{M}} = \textsc{Prc}\backslash\rho(X)$. 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 formulae. As the environment has no effect on the semantics of a closed formula φ, we write $\mathcal{M}, s \models_{\mathcal{M}} \varphi$ for $s \in [\![\varphi,\rho]\!]_{\mathcal{M}}$. If $s \models_{\mathcal{M}} \varphi$, we say that φ is true, or satisfied, in s. When the particular LTS does not matter, or is clear from the context, we often omit it from the notation.

Depending on how we further restrict our syntax, and the LTS, we can describe several logics. Without further restrictions, the resulting logic is the μ-calculus [12]. The max-fragment (resp. min-fragment) of the μ-calculus is the fragment that only allows the νX (resp. the μX) recursive operator. If $|\textsc{Act}| = k$ and we allow no recursive operators (or recursion variables), then we have the basic modal logic $\mathbf{K}_k$, and further restrictions on the LTS can result in a wide variety of modal logics (see [3]).

We give names to the following LTS constraints.[1] For every $\alpha \in \textsc{Act}$:

***D*:** there is no $\texttt{nil}$ state in the LTS — in other words, $\xrightarrow{\alpha}$ is serial;

***T*:** $\xrightarrow{\alpha}$ is reflexive — in other words, $\forall s.\ s \xrightarrow{\alpha} s$;

[1] These conditions correspond to the usual axioms for normal modal logics — see [4, 3, 6].

4: $\xrightarrow{\alpha}$ is transitive — in other words, $\forall s, t, r.\ (s \xrightarrow{\alpha} t \xrightarrow{\alpha} r \ \Rightarrow\ s \xrightarrow{\alpha} r)$;

5: $\xrightarrow{\alpha}$ is Euclidean — in other words, $\forall s, t, r.$ if $s \xrightarrow{\alpha} t$ and $s \xrightarrow{\alpha} r$, then $t \xrightarrow{\alpha} r$.

We consider modal logics that are interpreted over LTSs that satisfy a combination of these constraints. D, which we call Consistency, is a special case of T, called Factivity. Constraint 4 is Positive Introspection and 5 is called Negative Introspection. Given a logic $\mathbf{L}$ and constraint c, $\mathbf{L} + c$ is the logic that is interpreted over all LTSs that satisfy all the constraints of $\mathbf{L}$ and c. Logic $\mathbf{D}_k$ is $\mathbf{K}_k + D$, $\mathbf{T}_k$ is $\mathbf{K}_k + T$, $\mathbf{K4}_k = \mathbf{K}_k + 4$, $\mathbf{D4}_k = \mathbf{K}_k + D + 4 = \mathbf{D}_k + 4$, $\mathbf{S4}_k = \mathbf{K}_k + T + 4 = \mathbf{T}_k + 4 = \mathbf{K4}_k + T$, $\mathbf{KD45}_k = \mathbf{D4}_k + 5$, and $\mathbf{S5}_k = \mathbf{S4}_k + 5$. When $k = 1$, we usually omit it from the subscript of a logic's name. For $\mathbf{L}$ being one of the logics above, $\mathbf{L}^\mu$ is the logic that results from $\mathbf{L}$ after we allow recursive operators in the syntax. Therefore, the μ-calculus is $\mathbf{K}_k^\mu$.

From now on, unless we explicitly say otherwise, by a logic or modal 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 an LTS for $\mathbf{L}$.

Recursion and Belief The introduction of recursive operators to Modal Logic allows one to express more properties. An important example is that of Common Knowledge or Belief. For instance, in our language, we can write that φ is common knowledge with formula $\nu X.(\varphi \wedge [\text{Act}]X)$. Another example is $\bigvee_{\alpha \in \text{Act}}[\alpha](\mu X.\varphi \vee \bigvee_{\beta \in \text{Act}}[\beta]X)$, which, in the context of a belief interpretation of the boxes, 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.

3 Satisfiability and Recursion in Modal Logic

3.1 What is Known

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 LTS.

Ladner [13] established the classical result of PSPACE-completeness for the satisfiability of $\mathbf{K}$, $\mathbf{T}$, $\mathbf{D}$, $\mathbf{K4}$, $\mathbf{D4}$, and $\mathbf{S4}$ and 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 [9], resulting in Theorem 1.

Theorem 1 ([13, 9])**.** *If* $\mathbf{L} \in \{\mathbf{K}, \mathbf{T}, \mathbf{D}, \mathbf{K4}, \mathbf{D4}, \mathbf{S4}\}$*, then* $\mathbf{L}$*-satisfiability is* PSPACE*-complete and* $\mathbf{L} + 5$*-satisfiability is* NP*-complete.*

Theorem 2 ([8])**.** *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 1. We note that Halpern and Moses in [8] only prove these bounds for the cases of $\mathbf{K}_k, \mathbf{T}_k, \mathbf{S4}_k, \mathbf{KD45}_k$, and $\mathbf{S5}_k$. However, it is not hard to see that their methods also work for the rest of the logics of Theorem 2. ■

Theorem 3 ([12])**.** *The satisfiability problem for the* μ*-calculus is* EXP*-complete.*

Theorem 4 ([5])**.** *The model checking problem for the* μ*-calculus is in* NP $\cap$ coNP*.*[2]

[2] In fact, the problem is known to be in UP $\cap$ coUP [11].

3.2 First Complexity Observations

We start with a few observations about the complexity of satisfiability when we have recursive operators. Propositions 5 and 6 have already been observed in [1].

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

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 [15] to describe the PDL formula that is constructed by the reduction used in [7] 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. □

To the best of our knowledge, there are no complexity results for the validity of logics with both LTS constraints and recursion operators. However, it is not hard to express in such logics that formula φ is common knowledge, with formula $\nu X.\varphi \wedge [\textsc{Act}]X$. Since validity for $\mathbf{L}_k$ with common knowledge (and without recursive operators) and $k > 1$ is EXP-complete [8][3], $\mathbf{L}_k^\mu$ must be EXP-hard.

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

We can write that φ is invariantly true in every state that is reachable from the current one (or, in other words, that it is common knowledge) by using the following formula: $Inv(\varphi) := \nu X.(\varphi \wedge [\textsc{Act}]X)$. We can also restrict the scope of the invariance to one specific action: $Inv_\alpha(\varphi) := \nu X.(\varphi \wedge [\alpha]X)$. Then, we can use these formulae to either impose that the current LTS (or the part that we have access to) has serial accessibility relations — by writing $Inv(\bigvee_{\alpha\in\textsc{Act}}\langle\alpha\rangle\mathtt{tt})$ — or to simulate a modality $[i^*]$ that is based on a transitive accessibility relation — by using $Inv_i(\varphi)$ for $[i^*]\varphi$. Hence, the constraints Consistency and Positive Introspection can be simulated already in the usual μ-calculus. Therefore:

Proposition 7. *The satisfiability problem for* $\mathbf{D}_k^\mu$*,* $\mathbf{K4}_k^\mu$*, and* $\mathbf{D4}_k^\mu$ *is in* EXP*.*

3.3 Negative Introspection: The One-action Case

In this section, we explain how to adapt Halpern and Rêgo's techniques from [9] to prove that even with recursive operators, satisfiability for one-action logics with Negative Introspection remains in NP. Thus, we assume in this section that $|\textsc{Act}| = 1$.

For a logic $\mathbf{L} + 5$, we call a state s in an LTS for $\mathbf{L} + 5$ *flat* when there is a set of states W, such that:

- $Reach(s) = \{s\} \cup W$;
- the restriction of $\rightarrow$ on $Reach(s)$ is $R \cup E$, where $R \subseteq \{s\} \times W$ and E is an equivalence relation on W; and
- if $\mathbf{L} \in \{\mathbf{T}, \mathbf{S4}\}$, then $s \in W$.

Lemma 8 ([14, 9])**.** *In an LTS with restriction* 5*, every state is a flat state.*

The construction from [13] and [9] filters the states of $Reach(s)$, where s is a flat state, resulting in a small state for a formula φ. We adapt the proof of this result for the case where we have recursive operators.

[3] Similarly to Remark 1, [8] does not explicitly cover all these cases, but the techniques can be adjusted.

Lemma 9. *Formula φ is $\mathbf{L}^\mu + 5$-satisfiable if and only if it is satisfied in a flat state of size at most $O(|\varphi|)$ in an LTS for $\mathbf{L} + 5$.*

Proof. The "if" direction is immediate. To prove the other direction, let (by Lemma 8) s be a flat state in LTS $\mathcal{M}$ that satisfies φ, where $Reach(s) = \{s\} \cup W$, satisfying the conditions given above for flat states. If $W = \emptyset$, then we are done. Otherwise, we assume that each X that appears in φ in in the scope of a unique recursive operator. Therefore, we can define for each open subformula ψ of φ its closure $cl(\psi)$, which substitutes each open variable with its corresponding recursive formula that binds it. For closed ψ, $cl(\psi) = \psi$.

Let $S_\Diamond$ be the set which contains every $\Diamond\psi$ subformula of φ, such that $cl(\Diamond\psi)$ is true in s and every $\Box\psi$ subformula of φ, such that $cl(\Box\psi)$ is not true in s; let $T_\Diamond$ be the set which contains every $\Diamond\psi$ subformula of φ, such that $cl(\Diamond\psi)$ is true in some state of $Reach(s)$ and every $\Box\psi$ subformula of φ, such that $cl(\Box\psi)$ is not true in some state of $Reach(s)$. For every $\Diamond\psi \in S_\Diamond$ we fix a state $s_\psi \in W$, such that $s \to s_\psi$ and where $cl(\psi)$ is true; for every $\Box\psi \in S_\Diamond$ we fix a state $s_\psi \in W$, such that $s \to s_\psi$ and where $cl(\psi)$ is not true; for every $\Diamond\psi \in T_\Diamond \setminus S_\Diamond$ we fix a state $s_\psi \in W$, where $cl(\psi)$ is true; finally, for every $\Box\psi \in T_\Diamond \setminus S_\Diamond$ we fix a state $s_\psi \in W$, where $cl(\psi)$ is not true. Notice that each $s_\psi \in W$ and thus it is accessible from every state in W. We construct the LTS $\mathcal{M}_\varphi = (W_\varphi, \textsc{Act}, \to_\varphi, V_\varphi)$ for $\mathbf{L} + 5$, where

$$W_\varphi = \{s\} \cup \{s_\psi \in W \mid \Diamond\psi \in T_\Diamond \text{ or } \Box\psi \in T_\Diamond\},$$

$\to_\varphi$ the restriction of R on W_φ, and $V_\varphi(a) = V(a)$ for all $a \in W_\varphi$. In the following, we abuse notation and we identify the environments for $\mathcal{M}$ with the ones for $\mathcal{M}_\varphi$.

It is not hard to confirm that $|W_\varphi| \leq |\varphi|$, since $|T_\Diamond| \leq |\varphi| - 1$ (at least one of the subformulae of φ is a propositional variable or $\mathtt{ff}$). Furthermore, $\mathcal{M}_\varphi, s \models \varphi$. Specifically, for all environments ρ, $t \in W_\varphi$, and $\psi \in sub(\varphi)$, we prove by induction on ψ that $\psi \in [\![t,\rho]\!]_{\mathcal{M}_\varphi}$ if and only if $\psi \in [\![t,\rho]\!]_{\mathcal{M}}$. Propositional cases are easy. If $\psi = \Diamond\chi$ and $\psi \in [\![t,\rho]\!]_{\mathcal{M}}$, then there is some s_χ, such that $\chi \in [\![s_\chi,\rho]\!]_{\mathcal{M}}$ and by the definition of s_χ, $t \to s_\chi$, therefore by the inductive hypothesis, $\chi \in [\![s_\chi,\rho]\!]_{\mathcal{M}_\varphi}$ and thus $\psi \in [\![t,\rho]\!]_{\mathcal{M}_\varphi}$. If $\psi = \Diamond\chi$ and $\psi \in [\![t,\rho]\!]_{\mathcal{M}_\varphi}$, then there is some $t \to_\varphi c \in W_\varphi$, such that $\chi \in [\![c,\rho]\!]_{\mathcal{M}_\varphi}$; by the inductive hypothesis, $\chi \in [\![c,\rho]\!]_{\mathcal{M}}$ and since $\to_\varphi$ is the restriction of $\to$ on W_φ, $t \to c$, so $\psi \in [\![t,\rho]\!]_{\mathcal{M}}$. The cases where $\psi = \Box\chi$ are similar. If $\psi = \nu X.\chi$ and $\psi \in [\![t,\rho]\!]_{\mathcal{M}_\varphi}$, then let $S = [\![\psi,\rho]\!]_{\mathcal{M}_\varphi}$. From the semantics of Table 1, it suffices now to see that from the inductive hypothesis, $S = [\![\chi,\rho[X \mapsto S]]\!]_{\mathcal{M}_\varphi} \subseteq [\![\chi,\rho[X \mapsto S]]\!]_{\mathcal{M}}$. The case for $\psi = \nu X.\chi$ and $\psi \in [\![t,\rho]\!]_{\mathcal{M}}$ is similar, as well as the cases for $\psi = \mu X.\chi$.

What remains is to demonstrate that $\mathcal{M}_\varphi$ remains an LTS for $\mathbf{L} + 5$. It is not hard to confirm that through this filtering, transitivity, Euclidicity, and reflexivity are preserved for $\to_\varphi$ (since they are preserved by restrictions on subsets of binary relations). As for seriality, it is enough to run this construction on $\varphi \wedge \Diamond\mathtt{tt}$ if necessary, thus increasing the upper bound on the number of states from $|\varphi|$ to $|\varphi| + 1$. $\Box$

Theorem 10. *The complexity of the satisfiability problem for logic $\mathbf{L}^\mu_k + 5$ is* NP*-complete.*

Proof. NP-hardness is easy to see, as propositional logic is a special case of $\mathbf{L}^\mu_k + 5$. To see that satisfiability is in NP, given a formula φ, an algorithm can nondeterministically guess a small LTS with a satisfying state for φ (by Lemma 9), and then use the nondeterministic polynomial-time algorithm for model-checking φ on the LTS (by Theorem 4). $\Box$

4 Open Directions

We still lack tight bounds for many of the logics, especially the ones with one action, and the ones with constraints T and 5 (with multiple actions). As, to the best of our knowledge, most of the logics described in this paper have not been explicitly defined before, they also lack any axiomatizations and completeness theorems. We do expect the classical methods from [12, 13, 8] 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. To us, it was somewhat surprising that the NP–PSPACE gap, with respect to Negative Introspection, has been preserved for one-action logics, even with recursive operators.

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) fixpoint already from [10, 2], 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, and Anna Ingólfsdóttir. The complexity of identifying characteristic formulae. Manuscript under submission.

[2] 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.

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

[4] Patrick Blackburn, J. F. A. K. van Benthem, and Frank Wolter. *Handbook of Modal Logic*, volume 3 of *Studies in Logic and Practical Reasoning*. Elsevier Science, 2006.

[5] 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.

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

[7] Michael J Fischer and Richard E Ladner. Propositional dynamic logic of regular programs. *Journal of computer and system sciences*, 18(2):194–211, 1979.

[8] 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.

[9] 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.

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

[11] Marcin Jurdziński. Deciding the winner in parity games is in UP $\cap$ co-UP. *Information Processing Letters*, 68(3):119–124, 1998.

[12] Dexter Kozen. Results on the propositional μ-calculus. *Theoretical Computer Science*, 27(3):333–354, 1983.

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

[14] Michael C. Nagle and S. K. Thomason. The extensions of the modal logic K5. *Journal of Symbolic Logic*, 50(1):102—-109, 1985.

[15] Vaughan R. Pratt. A decidable mu-calculus: Preliminary report. In *22nd Annual Symposium on Foundations of Computer Science (SFCS 1981)*. IEEE, oct 1981.