



NP Reasoning in the Monotone μ -Calculus

Daniel Hausmann^(✉) and Lutz Schröder^(✉)

Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany
{daniel.hausmann,lutz.schroeder}@fau.de

Abstract. Satisfiability checking for monotone modal logic is known to be (only) NP-complete. We show that this remains true when the logic is extended with alternation-free fixpoint operators as well as the universal modality; the resulting logic – the *alternation-free monotone μ -calculus with the universal modality* – contains both concurrent propositional dynamic logic (CPDL) and the alternation-free fragment of game logic as fragments. We obtain our result from a characterization of satisfiability by means of Büchi games with polynomially many Eloise nodes.

1 Introduction

Monotone modal logic differs from normal modal logics (such as K [4], equivalent to the standard description logic \mathcal{ALC} [1]) by giving up distribution of conjunction over the box modality, but retaining monotonicity of the modalities. Its semantics is based on (monotone) neighbourhood models instead of Kripke models. Monotone modalities have been variously used as epistemic operators that restrict the combination of knowledge by epistemic agents [27]; as next-step modalities in the evolution of concurrent systems, e.g. in *concurrent propositional dynamic logic* (CPDL) [24]; and as game modalities in systems where one transition step is determined by moves of two players, e.g. in Parikh’s *game logic* [7, 12, 20, 23]. The monotonicity condition suffices to enable formation of fixpoints; one thus obtains the *monotone μ -calculus* [8], which contains both CPDL and game logic as fragments (indeed, the recent proof of completeness of game logic [7] is based on embedding game logic into the monotone μ -calculus).

While many modal logics (including K/\mathcal{ALC}) have PSPACE-complete satisfiability problems in the absence of fixpoints, it is known that satisfiability in monotone modal logic is only NP-complete [27] (the lowest possible complexity given that the logic has the full set of Boolean connectives). In the present paper, we show that the low complexity is preserved under two extensions that usually cause the complexity to rise from PSPACE-complete to EXPTIME-complete: Adding the *universal modality* (equivalently global axioms or, in description logic parlance, a general TBox) and alternation-free *fixpoints*; that is, we show that satisfiability checking in the *alternation-free fragment of the monotone μ -calculus with the universal modality* [8] is only NP-complete. This logic subsumes both CPDL and the alternation-free fragment of game logic [23]. Thus, our results

imply that satisfiability checking in these logics is only NP-complete (the best previously known upper bound being EXPTIME in both cases [20, 23, 24]); for comparison, standard propositional dynamic logic (PDL) and in fact already the extension of K with the universal modality are EXPTIME-hard. (Our results thus seemingly contradict previous results on EXPTIME-completeness of CPDL. However, these results rely on embedding standard PDL into CPDL, which requires changing the semantics of CPDL to interpret atomic programs as sequential programs, i.e. as relations rather than neighbourhood systems [24].) Our results are based on a variation of the game-theoretic approach to μ -calculi [19]. Specifically, we reduce satisfiability checking to the computation of winning regions in a satisfiability game that has exponentially many Abelard-nodes but only polynomially many Eloise-nodes, so that history-free winning strategies for Eloise have polynomial size. From this approach we also derive a polynomial model property.

Organization. We recall basics on fixpoints and games in Sect. 2, and the syntax and semantics of the monotone μ -calculus in Sect. 3. We discuss a key technical tool, *formula tracking*, in Sect. 4. We adapt the standard tableaux system to the monotone μ -calculus in Sect. 5. In Sect. 6, we establish our main results using a game characterization of satisfiability.

2 Notation and Preliminaries

We fix basic concepts and notation on fixpoints and games.

Fixpoints. Let U be a set; we write $\mathcal{P}(U)$ for the powerset of U . Let $f : \mathcal{P}(U) \rightarrow \mathcal{P}(U)$ be a *monotone* function, i.e. $f(A) \subseteq f(B)$ whenever $A \subseteq B \subseteq U$. By the Knaster-Tarski fixpoint theorem, the *greatest* (GFP) and *least* (LFP) *fixpoints* of f are given by

$$\text{LFP } f = \bigcap \{V \subseteq U \mid f(V) \subseteq V\} \quad \text{GFP } f = \bigcup \{V \subseteq U \mid V \subseteq f(V)\},$$

and are thus also the least prefixpoint ($f(V) \subseteq V$) and greatest postfixpoint ($V \subseteq f(V)$) of f , respectively. For $V \subseteq U$ and $n \in \mathbb{N}$, we define $f^n(V)$ as expected by $f^0(V) = V$ and $f^{n+1}(V) = f(f^n(V))$. If U is finite, then $\text{LFP } f = f^{|U|}(\emptyset)$ and $\text{GFP } f = f^{|U|}(U)$ by Kleene's fixpoint theorem.

Infinite Words and Games. We denote the sets of finite and infinite sequences of elements of a set U by U^* and U^ω , respectively. We often view sequences $\tau = u_1, u_2, \dots$ in U^* (or U^ω) as partial (total) functions $\tau : \mathbb{N} \rightarrow U$, writing $\tau(i) = u_i$. We write

$$\text{Inf}(\tau) = \{u \in U \mid \forall i \in \mathbb{N}. \exists j > i. \tau(j) = u\}$$

for the set of elements of U that occur infinitely often in τ .

A *Büchi game* $G = (V, E, v_0, F)$ consists of a set V of *nodes*, partitioned into the sets V_\exists of *Eloise-nodes* and V_\forall of *Abelard-nodes*, a set $E \subseteq V \times V$ of *moves*, an *initial node* v_0 , and a set $F \subseteq V$ of *accepting* nodes. We write

$E(v) = \{v' \mid (v, v') \in E\}$. For simplicity, assume that $v_0 \in V_{\exists}$ and that the game is *alternating*, i.e. $E(v) \subseteq V_{\forall}$ for all $v \in V_{\exists}$, and $E(v) \subseteq V_{\exists}$ for all $v \in V_{\forall}$ (our games will have this shape). A *play* of G is a sequence $\tau = v_0, v_1, \dots$ of nodes such that $(v_i, v_{i+1}) \in E$ for all $i \geq 0$ and τ is either infinite or ends in a node without outgoing moves. Eloise *wins* a play τ if and only if τ is finite and ends in an Abelard-node or τ is infinite and $\text{Inf}(\tau) \cap F \neq \emptyset$, that is, τ infinitely often visits an accepting node. A *history-free Eloise-strategy* is a partial function $s : V_{\exists} \rightarrow V$ such that $s(v_0)$ is defined, and whenever $s(v)$ is defined, then $(v, s(v)) \in E$ and $s(v')$ is defined for all $v' \in E(s(v))$. A play v_0, v_1, \dots is an *s-play* if $v_{i+1} = s(v_i)$ whenever $v_i \in V_{\exists}$. We say that s is a *winning strategy* if Eloise wins every s -play, and that Eloise *wins* G if there is a winning strategy for Eloise. Büchi games are history-free determined, i.e. in every Büchi game, one of the players has a history-free winning strategy [17].

3 The Monotone μ -Calculus

We proceed to recall the syntax and semantics of the monotone μ -calculus.

Syntax. We fix countably infinite sets \mathbf{P} , \mathbf{A} and \mathbf{V} of *atoms*, *atomic programs* and (*fixpoint*) *variables*, respectively; we assume that \mathbf{P} is closed under duals (i.e. atomic negation), i.e. $p \in \mathbf{P}$ implies $\bar{p} \in \mathbf{P}$, where $\bar{\bar{p}} = p$. Formulae of the *monotone μ -calculus* (in negation normal form) are then defined by the grammar

$$\psi, \phi := \perp \mid \top \mid p \mid \psi \wedge \phi \mid \psi \vee \phi \mid \langle a \rangle \psi \mid [a] \psi \mid X \mid \eta X. \psi$$

where $p \in \mathbf{P}$, $a \in \mathbf{A}$, $X \in \mathbf{V}$; throughout, we use $\eta \in \{\mu, \nu\}$ to denote extremal fixpoints. As usual, μ and ν are understood as taking least and greatest fixpoints, respectively, and bind their variables, giving rise to the standard notion of *free variable* in a formula ψ . We write $\text{FV}(\psi)$ for the set of free variables in ψ , and say that ψ is *closed* if $\text{FV}(\psi) = \emptyset$. Negation \neg is not included but can be defined by taking negation normal forms as usual, with $\neg p = \bar{p}$. We refer to formulae of the shape $[a]\phi$ or $\langle a \rangle \phi$ as (*a*-) *modal literals*. As indicated in the introduction, the *modalities* $[a]$, $\langle a \rangle$ have been equipped with various readings, recalled in more detail in Example 7.

Given a closed formula ψ , the *closure* $\text{cl}(\psi)$ of ψ is defined to be the least set of formulae that contains ψ and satisfies the following closure properties:

- if $\psi_1 \wedge \psi_2 \in \text{cl}(\psi)$ or $\psi_1 \vee \psi_2 \in \text{cl}(\psi)$, then $\{\psi_1, \psi_2\} \subseteq \text{cl}(\psi)$,
- if $\langle a \rangle \psi_1 \in \text{cl}(\psi)$ or $[a] \psi_1 \in \text{cl}(\psi)$, then $\psi_1 \in \text{cl}(\psi)$,
- if $\eta X. \psi_1 \in \text{cl}(\psi)$, then $\psi_1[\eta X. \psi_1 / X] \in \text{cl}(\psi)$,

where $\psi_1[\eta X. \psi_1 / X]$ denotes the formula that is obtained from ψ_1 by replacing every free occurrence of X in ψ_1 with $\eta X. \psi_1$. Note that all formulae in $\text{cl}(\psi)$ are closed. We define the *size* $|\psi|$ of ψ as $|\psi| = |\text{cl}(\psi)|$. A formula ψ is *guarded* if whenever $\eta X. \phi \in \text{cl}(\psi)$, then all free occurrences of X in ϕ are under the scope of at least one modal operator. *We generally restrict to guarded formulae;*

see however Remark 1. A closed formula ψ is *clean* if all fixpoint variables in ψ are bound by exactly one fixpoint operator. Then $\theta(X)$ denotes the subformula $\eta X.\phi$ that binds X in ψ , and X is a least (greatest) fixpoint variable if $\eta = \mu$ ($\eta = \nu$). We define a partial order \geq_μ on the least fixpoint variables in ρ_1 and ρ_0 by $X \geq_\mu Y$ iff $\theta(Y)$ is a subformula of $\theta(X)$ and $\theta(Y)$ is not in the scope of a greatest fixpoint operator within $\theta(X)$ (i.e. there is no greatest fixpoint operator between μX and μY). The *index* $\text{idx}(X)$ of such a fixpoint variable X is

$$\text{idx}(X) = |\{Y \in \mathbf{V} \mid Y \geq_\mu X\}|,$$

For a subformula ϕ of ψ , we write $\text{idx}(\phi) = \max\{\text{idx}(X) \mid X \in \text{FV}(\phi)\}$. We denote by $\theta^*(\phi_0)$ the closed formula that is obtained from a subformula ϕ_0 of ψ by repeatedly replacing free variables X with $\theta(X)$. Formally, we define $\theta^*(\phi_0)$ as $\phi_{|\text{FV}(\phi_0)|}$, where ϕ_{i+1} is defined inductively from ϕ_i . If ϕ_i is closed, then put $\phi_{i+1} = \phi_i$. Otherwise, pick the variable $X_i \in \text{FV}(\phi_i)$ with the greatest index and put $\phi_{i+1} = \phi_i[\theta(X_i)/X_i]$. Then $\text{idx}(\phi_{i+1}) < \text{idx}(\phi_i)$, so $\theta^*(\phi_0)$ really is closed; moreover, one can show that $\theta^*(\phi_0) \in \text{cl}(\psi)$. A clean formula is *alternation-free* if none of its subformulae contains both a free least and a free greatest fixpoint variable. Finally, ψ is *irredundant* if $X \in \text{FV}(\phi)$ whenever $\eta X.\phi \in \text{cl}(\psi)$.

Remark 1. We have defined the size of formulae as the cardinality of their closure, implying a very compact representation [3]. Our upper complexity bounds thus become *stronger*, i.e. they hold even for this small measure of input size. Moreover, the restriction to guarded formulae is then without loss of generality, since one has a *guardedness transformation* that transforms formulae into equivalent guarded ones, with only polynomial blowup of the closure [3].

Semantics. The monotone μ -calculus is interpreted over *neighbourhood models* (or *epistemic structures* [27]) $F = (W, N, I)$ where $N : A \times W \rightarrow 2^{(2^W)}$ assigns to each atomic program a and each state w a set $N(a, w) \subseteq 2^W$ of *a-neighbourhoods* of w , and $I : \mathbf{P} \rightarrow 2^W$ interprets propositional atoms such that $I(p) = W \setminus I(\bar{p})$ for $p \in \mathbf{P}$ (by 2, we denote the set $\{\perp, \top\}$ of Boolean truth values, and 2^W is the set of maps $W \rightarrow 2$, which is in bijection with the powerset $\mathcal{P}(W)$). Given such an F , each formula ψ is assigned an *extension* $\llbracket \psi \rrbracket_\sigma \subseteq W$ that additionally depends on a *valuation* $\sigma : V \rightarrow 2^W$, and is inductively defined by

$$\begin{aligned} \llbracket p \rrbracket_\sigma &= I(p) & \llbracket X \rrbracket_\sigma &= \sigma(X) \\ \llbracket \psi \wedge \phi \rrbracket_\sigma &= \llbracket \psi \rrbracket_\sigma \cap \llbracket \phi \rrbracket_\sigma & \llbracket \psi \vee \phi \rrbracket_\sigma &= \llbracket \psi \rrbracket_\sigma \cup \llbracket \phi \rrbracket_\sigma \\ \llbracket \langle a \rangle \psi \rrbracket_\sigma &= \{w \in W \mid \exists S \in N(a, w). S \subseteq \llbracket \psi \rrbracket_\sigma\} & \llbracket \mu X.\psi \rrbracket_\sigma &= \text{LFP}[\llbracket \psi \rrbracket_\sigma^X] \\ \llbracket [a] \psi \rrbracket_\sigma &= \{w \in W \mid \forall S \in N(a, w). S \cap \llbracket \psi \rrbracket_\sigma \neq \emptyset\} & \llbracket \nu X.\psi \rrbracket_\sigma &= \text{GFP}[\llbracket \psi \rrbracket_\sigma^X] \end{aligned}$$

where, for $U \subseteq W$ and fixpoint variables $X, Y \in \mathbf{V}$, we put $\llbracket \psi \rrbracket_\sigma^X(U) = \llbracket \psi \rrbracket_{\sigma[X \mapsto U]}$, $(\sigma[X \mapsto U])(X) = U$ and $(\sigma[X \mapsto U])(Y) = \sigma(Y)$ if $X \neq Y$. We omit the dependence on F in the notation $\llbracket \phi \rrbracket_\sigma$, and when necessary clarify the underlying neighbourhood model by phrases such as ‘in F ’. If ψ is closed, then its extension does not depend on the valuation, so we just write $\llbracket \psi \rrbracket$. A

closed formula ψ is *satisfiable* if there is a neighbourhood model F such that $\llbracket \psi \rrbracket \neq \emptyset$ in F ; in this case, we also say that ψ is *satisfiable over F* . Given a set Ψ of closed formulae, we write $\llbracket \Psi \rrbracket = \bigcap_{\psi \in \Psi} \llbracket \psi \rrbracket$. An (*infinite*) *path* through a neighbourhood model (W, N, I) is a sequence x_0, x_1, \dots of states $x_i \in W$ such that for all $i \geq 0$, there are $a \in A$ and $S \in N(a, x_i)$ such that $x_{i+1} \in S$.

The soundness direction of our game characterization will rely on the following immediate property of the semantics, which may be seen as soundness of a modal tableau rule [5].

Lemma 2 [27, Proposition 3.8]. *If $[a]\phi \wedge \langle a \rangle \psi$ is satisfiable over a neighbourhood model F , then $\phi \wedge \psi$ is also satisfiable over F .*

Remark 3. The dual box and diamond operators $[a]$ and $\langle a \rangle$ are completely symmetric, and indeed the notation is not uniform in the literature. Our use of $[a]$ and $\langle a \rangle$ is generally in agreement with work on game logic [20] and CPDL [24]; in work on monotone modal logics and the monotone μ -calculus, the roles of box and diamond are often interchanged [8, 26, 27].

Remark 4. The semantics may equivalently be presented in terms of *monotone* neighbourhood models, where the set of a -neighbourhoods of a state is required to be upwards closed under subset inclusion [8, 12, 20, 23]. In this semantics, the interpretation of $\langle a \rangle \phi$ simplifies to just requiring that the extension of ϕ is an a -neighbourhood of the current state. We opt for the variant where upwards closure is instead incorporated into the interpretation of the modalities, so as to avoid having to distinguish between monotone neighbourhood models and their representation as upwards closures of (plain) neighbourhood models, e.g. in small model theorems.

We further extend the expressiveness of the logic (see also Remark 9) by adding *global assumptions* or equivalently the *universal modality*:

Definition 5 (Global assumptions). Given a closed formula ϕ , a ϕ -*model* is a neighbourhood model $F = (W, N, I)$ in which $\llbracket \phi \rrbracket = W$. A formula ψ is ϕ -*satisfiable* if ψ is satisfiable over some ϕ -model; in this context, we refer to ϕ as the *global assumption*, and to the problem of deciding whether ψ is ϕ -satisfiable as *satisfiability checking under global assumptions*.

We also define an extension of the monotone μ -calculus, the *monotone μ -calculus with the universal modality*, by adding two alternatives

$$\dots \mid [\forall]\phi \mid [\exists]\phi$$

to the grammar, in both alternatives restricting ϕ to be closed. The definition of the semantics over a neighbourhood model (W, N, I) and valuation σ is correspondingly extended by $\llbracket [\forall]\phi \rrbracket_\sigma = W$ if $\llbracket \phi \rrbracket_\sigma = W$, and $\llbracket [\forall]\phi \rrbracket_\sigma = \emptyset$ otherwise; dually, $\llbracket [\exists]\phi \rrbracket_\sigma = W$ if $\llbracket \phi \rrbracket_\sigma \neq \emptyset$, and $\llbracket [\exists]\phi \rrbracket_\sigma = \emptyset$ otherwise. That is, $[\forall]\phi$ says that ϕ holds in all states of the model, and $[\exists]\phi$ that ϕ holds in some state.

Remark 6. In description logic, global assumptions are typically called (general) *TBoxes* or *terminologies* [1]. For many next-step modal logics (i.e. modal logics without fixpoint operators), satisfiability checking becomes harder under global assumptions. A typical case is the standard modal logic K (corresponding to the description logic \mathcal{ALC}), in which (plain) satisfiability checking is PSPACE-complete [15] while satisfiability checking under global assumptions is EXPTIME-complete [6,9]. Our results show that such an increase in complexity does not happen for monotone modalities.

For purposes of satisfiability checking, the universal modality and global assumptions are mutually reducible in a standard manner, where the non-trivial direction (from the universal modality to global assumptions) is by guessing beforehand which subformulae $\forall\phi$, $\exists\phi$ hold (see also [11]).

Example 7. 1. In *epistemic logic*, neighbourhood models have been termed *epistemic structures* [27]. In this context, the $a \in A$ are thought of as *agents*, the a -neighbourhoods of a state w are the facts known to agent a in w , and correspondingly the reading of $[a]\phi$ is ‘ a knows ϕ ’. The use of (monotone) neighbourhood models and the ensuing failure of normality imply that agents can still weaken facts that they know but are not in general able to combine them, i.e. knowing ϕ and knowing ψ does not entail knowing $\phi \wedge \psi$ [27].

2. In *concurrent propositional dynamic logic (CPDL)*, a -neighbourhoods of a state are understood as sets of states that can be reached concurrently, while the choice between several a -neighbourhoods of a state models sequential non-determinism. CPDL indexes modalities over composite *programs* α , formed using tests $?\phi$ and the standard operations of propositional dynamic logic (PDL) (union \cup , sequential composition ‘;’, and Kleene star $(-)^*$) and additionally intersection \cap . It forms a sublogic of Parikh’s game logic, recalled next, and thus in particular translates into the monotone μ -calculus.

As indicated in the introduction, CPDL satisfiability checking has been shown to be EXPTIME-complete [24], seemingly contradicting our results (Corollary 25). Note however that the interpretation of atomic programs in CPDL, originally defined in terms of neighbourhood systems [24, p. 453], is, for purposes of the EXPTIME-hardness proof, explicitly changed to relations [24, pp. 458–459]; EXPTIME-hardness then immediately follows since PDL becomes a sublogic of CPDL [24, p. 461]. Our NP bound applies to the original semantics.

3. *Game logic* [20,23] extends CPDL by a further operator on programs, *dualization* $(-)^d$, and reinterprets programs as games between two players *Angel* and *Demon*; in this view, dualization just corresponds to swapping the roles of the players. In comparison to CPDL, the main effect of dualization is that one obtains an additional *demonic iteration operator* $(-)^{\times}$, distinguished from standard iteration $(-)^*$ by letting Demon choose whether or not to continue the iteration. A game logic formula is *alternation-free* if it does not contain nested occurrences (unless separated by a test) of $(-)^{\times}$ within $(-)^*$ or vice versa [23].

Enqvist et al. [7] give a translation of game logic into the monotone μ -calculus that is quite similar to Pratt’s [25] translation of PDL into the standard μ -calculus. The translation $(-)^{\sharp}$ is defined by commutation with all Boolean connectives and by

$$(\langle \gamma \rangle \phi)^{\sharp} = \tau_{\gamma}(\phi^{\sharp}),$$

in mutual recursion with a function τ_{γ} that translates the effect of applying $\langle \gamma \rangle$ into the monotone μ -calculus. (Boxes $[\gamma]$ can be replaced with $\langle \gamma^d \rangle$). We refrain from repeating the full definition of τ_{γ} by recursion over γ ; some key clauses are

$$\tau_{\gamma \cap \delta}(\psi) = \tau_{\gamma}(\psi) \wedge \tau_{\delta}(\psi) \quad \tau_{\gamma^*}(\psi) = \mu X. (\psi \vee \tau_{\gamma}(X)) \quad \tau_{\gamma^{\times}}(\psi) = \nu Y. (\psi \wedge \tau_{\gamma}(Y))$$

where in the last two clauses, X and Y are chosen as fresh variables (for readability, we gloss over a more precise treatment of this point given in [7]). The first clause (and a similar one for \cup) appear at first sight to cause exponential blowup but recall that we measure the size of formulae by the cardinality of their closure; in this measure, there is in fact no blowup. The translated formula ψ^{\sharp} need not be guarded as the clauses for $*$ and \times can introduce unguarded fixpoint variables; as mentioned in Remark 1, we can however apply the guardedness transformation, with only quadratic blowup of the closure [3].

Under this translation, the alternation-free fragment of game logic ends up in the (guarded) alternation-free fragment of the monotone μ -calculus.

For later use, we note

Lemma 8. *The monotone μ -calculus with the universal modality has the finite model property.*

Proof (sketch). We reduce to global assumptions as per Remark 6, and proceed by straightforward adaptation of the translations of monotone modal logic [14] and game logic [23] into the relational μ -calculus, thus inheriting the finite model property [2]. This translation is based on turning neighbourhoods into additional states, connected to their elements via a fresh relation e . Then, e.g., the monotone modality $[a]$ (in our notation, cf. Remark 3) is translated into $[a]\langle e \rangle$ (relational modalities). Moreover, we translate a global assumption ϕ into a formula saying that all reachable states satisfy ϕ , expressed in the μ -calculus in a standard fashion. □

Remark 9. In the relational μ -calculus, we can encode a modality \boxtimes ‘in all reachable states’, generalizing the AG operator from CTL. As already indicated in the proof of Lemma 8, this modality allows for a straightforward reduction of satisfiability under global assumptions to plain satisfiability in the relational μ -calculus: A formula ψ is satisfiable under the global assumption ϕ iff $\psi \wedge \boxtimes \phi$ is satisfiable, where ‘if’ is shown by restricting the model to reachable states. To

motivate separate consideration of the universal modality, we briefly argue why an analogous reduction does not work in the monotone μ -calculus.

It is not immediately clear what reachability would mean on neighbourhood models. We can however equivalently rephrase the definition of \boxtimes in the relational case to let $\boxtimes\phi$ mean ‘the present state is contained in a submodel in which every state satisfies ϕ ’, where as usual a submodel of a relational model C with state set W is a model C' with state set $W' \subseteq W$ such that the graph of the inclusion $W' \hookrightarrow W$ is a bisimulation from C' to C . We thus refer to \boxtimes as the *submodel modality*. This notion transfers to neighbourhood models using a standard notion of bisimulation: A *monotone bisimulation* [21, 22] between neighbourhood models (W_1, N_1, I_1) , (W_2, N_2, I_2) is a relation $S \subseteq W_1 \times W_2$ such that whenever $(x, y) \in S$, then

- for all $a \in A$ and $A \in N_1(a, x)$, there is $B \in N_2(a, y)$ such that for all $v \in B$, there is $u \in A$ such that $(u, v) \in S$,
- for all $a \in A$ and $B \in N_2(a, y)$, there is $A \in N_1(a, x)$ such that for all $u \in A$, there is a $v \in B$ such that $(u, v) \in S$,
- for all $p \in P$, $x \in I_1(p)$ if and only if $y \in I_2(p)$.

We then define a *submodel* of a neighbourhood model $F = (W, N, I)$ to be a neighbourhood model $F' = (W', N', I')$ such that $W' \subseteq W$ and the graph of the inclusion $W' \hookrightarrow W$ is a monotone bisimulation between F' and F . If $(x, y) \in S$ for some monotone bisimulation S , then x and y satisfy the same formulae in the monotone μ -calculus [8]. It follows that the submodel modality \boxtimes on neighbourhood models, defined verbatim as in the relational case, allows for the same reduction of satisfiability under global assumptions as the relational submodel modality.

However, the submodel modality fails to be expressible in the monotone μ -calculus, as seen by the following example. Let $F_1 = (W_1, N_1, I_1)$, $F_2 = (W_2, N_2, I_2)$ be the neighbourhood models given by $W_1 = \{x_1, u_1, v_{11}, v_{12}\}$, $W_2 = \{x_2, u_2, v_2\}$, $N_1(a, x_1) = \{\{u_1, v_{11}\}, \{v_{11}, v_{12}\}\}$, $N_2(a, x_2) = \{\{v_2\}\}$, $N_1(b, y) = N_2(b, z) = \emptyset$ for $(b, y) \neq (a, x_1)$, $(b, z) \neq (a, x_2)$, $I_1(p) = \{x_1, v_{11}, v_{12}\}$, $I_2(p) = \{x_2, v_2\}$, and $I_1(q) = I_2(q) = \emptyset$ for $q \neq p$. Then $S = \{(x_1, x_2), (u_1, u_2), (v_{11}, v_2), (v_{12}, v_2)\}$ is a monotone bisimulation, so x_1 and x_2 satisfy the same formulae in the monotone μ -calculus. However, x_2 satisfies $\boxtimes p$ because x_2 is contained in a submodel with set $\{x_2, v_2\}$ of states, while x_1 does not satisfy $\boxtimes p$.

4 Formula Tracking

A basic problem in the construction of models in fixpoint logics is to avoid infinite unfolding of least fixpoints, also known as infinite deferral. To this end, we use a *tracking function* to follow formulae along paths in prospective models. For unrestricted μ -calculi, infinite unfolding of least fixpoints is typically detected by means of a *parity condition* on tracked formulae. However, since we restrict to alternation-free formulae, we can instead use a *Büchi condition* to detect (and

then reject) such infinite unfoldings by sequential *focussing* on sets of formulae in the spirit of focus games [16].

We fix *closed, clean, irredundant formulae* ρ_1 and ρ_0 , aiming to check ρ_0 -satisfiability of ρ_1 ; we also require both ρ_1 and ρ_0 to be alternation-free. We put $\text{cl} = \text{cl}(\rho_1) \cup \text{cl}(\rho_0)$ and $n = |\text{cl}|$.

Next we formalize our notion of deferred formulae that originate from least fixpoints; these are the formulae for which infinite unfolding has to be avoided.

Definition 10 (Deferrals). A formula $\phi \in \text{cl}(\psi)$ is a *deferral* if there is a subformula χ of ψ such that $\text{FV}(\chi)$ contains a least fixpoint variable and $\phi = \theta^*(\chi)$. We put $\text{dfr} = \{\phi \in \text{cl} \mid \phi \text{ is a deferral}\}$.

Since we assume formulas to be alternation-free, no formula $\nu X. \phi$ is a deferral.

Example 11. For $\psi = \mu X. [b]\top \vee \langle a \rangle X$, the formula $[b]\top \vee \langle a \rangle \psi = ([b]\top \vee \langle a \rangle X)[\psi/X] \in \text{cl}(\psi)$ is a deferral since X is a least fixpoint variable and occurs free in $[b]\top \vee \langle a \rangle X$; the formula $[b]\top$ on the other hand is not a deferral since it cannot be obtained from a subformula of ψ by replacing least fixpoint variables with their binding fixpoint formulae.

We proceed to define a tracking function that nondeterministically tracks deferrals along paths in neighbourhood models.

Definition 12 (Tracking function). We define an alphabet $\Sigma = \Sigma_p \cup \Sigma_m$ for traversing the closure, separating propositional and modal traversal, by

$$\begin{aligned} \Sigma_p &= \{(\phi_0 \wedge \phi_1, 0), (\phi_0 \vee \phi_1, b), (\eta X. \phi_0, 0) \mid \\ &\quad \phi_0, \phi_1 \in \text{cl}, \eta \in \{\mu, \nu\}, X \in \mathbf{V}, b \in \{0, 1\}\} \\ \Sigma_m &= \{(\langle a \rangle \phi_0, [a]\phi_1) \mid \phi_0, \phi_1 \in \text{cl}, a \in \mathbf{A}\}. \end{aligned}$$

The *tracking function* $\delta : \text{dfr} \times \Sigma \rightarrow \mathcal{P}(\text{dfr})$ is defined by $\delta(\text{foc}, (\chi, b)) = \{\text{foc}\}$ for $\text{foc} \in \text{dfr}$ and $(\chi, b) \in \Sigma_p$ with $\chi \neq \text{foc}$, and by

$$\begin{aligned} \delta(\text{foc}, (\text{foc}, b)) &= \begin{cases} \{\phi_0, \phi_1\} \cap \text{dfr} & \text{foc} = \phi_0 \wedge \phi_1 \\ \{\phi_b\} \cap \text{dfr} & \text{foc} = \phi_0 \vee \phi_1 \\ \{\phi_0[\theta(X)/X]\} & \text{foc} = \mu X. \phi_0 \end{cases} \\ \delta(\text{foc}, (\langle a \rangle \phi_0, [a]\phi_1)) &= \begin{cases} \{\phi_0\} & \text{foc} = \langle a \rangle \phi_0 \\ \{\phi_1\} & \text{foc} = [a]\phi_1, \end{cases} \end{aligned}$$

noting that $\mu X. \phi_0 \in \text{dfr}$ implies $\phi_0[\theta(X)/X] \in \text{dfr}$. We extend δ to sets $\text{Foc} \subseteq \text{dfr}$ by putting $\delta(\text{Foc}, l) = \bigcup_{\text{foc} \in \text{Foc}} \delta(\text{foc}, l)$ for $l \in \Sigma$. We also extend δ to words in the obvious way; e.g. we have $\delta(\text{foc}, w) = [a]\text{foc}$ for

$$\text{foc} = \mu X. (p \vee (\langle b \rangle p \wedge [a]X)) \quad w = (\text{foc}, 0), (p \vee (\langle b \rangle p \wedge [a]\text{foc}), 1), (\langle b \rangle p \wedge [a]\text{foc}, 0).$$

Remark 13. Formula tracking can be modularized in an elegant way by using tracking automata to accept exactly the paths that contain infinite deferral of some least fixpoint formula. While tracking automata for the full μ -calculus are in general nondeterministic parity automata [10], the automata for alternation-free formulae are nondeterministic co-Büchi automata. Indeed, our tracking function δ can be seen as the transition function of a nondeterministic co-Büchi automaton with set dfr of states and alphabet Σ in which all states are accepting. Since models have to be constructed from paths that do *not* contain infinite deferral of some least fixpoint formula, the tracking automata have to be complemented in an intermediate step (which is complicated by nondeterminism) when checking satisfiability. In our setting, the complemented automata are deterministic Büchi automata obtained using a variant of the Miyano-Hayashi construction [18] similarly as in our previous work on EXPTIME satisfiability checking in alternation-free coalgebraic μ -calculi [13]; for brevity, this determinization procedure remains implicit in our satisfiability games (see Definition 20 below).

To establish our game characterization of satisfiability, we combine the tracking function δ with a function for propositional transformation of formula sets guided by letters from Σ_p , embodied in the function γ defined next.

Definition 14 (Propositional transformation). We define $\gamma : \mathcal{P}(\text{cl}) \times \Sigma_p \rightarrow \mathcal{P}(\text{cl})$ on $\Gamma \subseteq \text{cl}$, $(\chi, b) \in \Sigma_p$ by $\gamma(\Gamma, (\chi, b)) = \Gamma$ if $\chi \notin \Gamma$, and

$$\gamma(\Gamma, (\chi, b)) = (\Gamma \setminus \chi) \cup \begin{cases} \{\phi_b\} & \chi = \phi_0 \vee \phi_1 \\ \{\phi_0, \phi_1\} & \chi = \phi_0 \wedge \phi_1 \\ \{\phi_0[\eta X. \phi_0/X]\} & \chi = \eta X. \phi_0 \end{cases}$$

if $\chi \in \Gamma$. We extend γ to words over Σ_p in the obvious way.

Example 15. Take $\phi = \chi \vee \nu X. (\psi_1 \wedge \psi_2)$ and $w = (\chi \vee \nu X. (\psi_1 \wedge \psi_2), 1), (\nu X. (\psi_1 \wedge \psi_2), 0), ((\psi_1 \wedge \psi_2)[\theta(X)/X], 1)$. The letter $(\chi \vee \nu X. (\psi_1 \wedge \psi_2), 1)$ picks the right disjunct and $(\nu X. (\psi_1 \wedge \psi_2), 0)$ passes through the fixpoint operator νX to reach $\psi_1 \wedge \psi_2$; the letter $(\psi_1 \wedge \psi_2, 1)$ picks both conjuncts. Thus,

$$\gamma(\{\phi\}, w) = \{\psi_1[\theta(X)/X], \psi_2[\theta(X)/X]\}.$$

5 Tableaux

As a stepping stone between neighbourhood models and satisfiability games, we now introduce *tableaux* which are built using a variant of the standard *tableau rules* [10] (each consisting of one *premise* and a possibly empty set of *conclusions*), where the *modal rule* ($\langle a \rangle$) reflects Lemma 2, taking into account the global assumption ρ_0 :

$$\begin{array}{lll} (\perp) & \frac{\Gamma, \perp}{\Gamma} & (\not\perp) & \frac{\Gamma, p, \bar{p}}{\Gamma} & (\wedge) & \frac{\Gamma, \phi_1 \wedge \phi_2}{\Gamma, \phi_1, \phi_2} \\ (\vee) & \frac{\Gamma, \phi_1 \vee \phi_2}{\Gamma, \phi_1} & (\langle a \rangle) & \frac{\Gamma, \langle a \rangle \phi_1, [a] \phi_2}{\phi_1, \phi_2, \rho_0} & (\eta) & \frac{\Gamma, \eta X. \phi}{\Gamma, \phi[\eta X. \phi/X]} \end{array}$$

(for $a \in \mathbf{A}$, $p \in \mathbf{P}$); we usually write rule instances with premise Γ and conclusion $\Theta = \Gamma_1, \dots, \Gamma_n$ ($n \leq 2$) inline as (Γ/Θ) . Looking back at Sect. 4, we see that letters in Σ designate rule applications, e.g. the letter $(\langle a \rangle\phi_1, [a]\phi_2) \in \Sigma_m$ indicates application of $(\langle a \rangle)$ to formulae $\langle a \rangle\phi_1, [a]\phi_2$.

Definition 16 (Tableaux). Let **states** denote the set of (*formal*) *states*, i.e. sets $\Gamma \subseteq \text{cl}$ such that $\perp \notin \Gamma$, $\{p, \bar{p}\} \not\subseteq \Gamma$ for all $p \in \mathbf{P}$, and such Γ does not contain formulae that contain top-level occurrences of the operators \wedge, \vee , or ηX .

A *pre-tableau* is a directed graph (W, L) , consisting of a finite set W of nodes labelled with subsets of cl by a labeling function $l : W \rightarrow \mathcal{P}(\text{cl})$, and of a relation $L \subseteq W \times W$ such that for all nodes $v \in W$ with label $l(v) = \Gamma \in \text{states}$ and all applications $(\Gamma/\Gamma_1, \dots, \Gamma_n)$ of a tableau rule to Γ , there is an edge $(v, w) \in L$ such that $l(w) = \Gamma_i$ for some $1 \leq i \leq n$. For nodes with label $l(v) = \Gamma \notin \text{states}$, we require that there is exactly one node $w \in W$ such that $(v, w) \in L$; then we demand that there is an application $(\Gamma/\Gamma_1, \dots, \Gamma_n)$ of a non-modal rule to Γ such that $l(w) = \Gamma_i$ for some $1 \leq i \leq n$. Finite or infinite words over Σ then encode sequences of rule applications (and choices of conclusions for disjunctions). That is, given a starting node v , they encode *branches* with root v , i.e. (finite or infinite) paths through (W, L) that start at v .

Deferrals are tracked along rule applications by means of the function δ . E.g. for a deferral $\phi_0 \vee \phi_1$, the letter $l = (\phi_0 \vee \phi_1, 0)$ identifies application of (\vee) to $\{\phi_0 \vee \phi_1\}$, and the choice of the left disjunct; then $\phi_0 \vee \phi_1$ is tracked from a node with label $\Gamma \cup \{\phi_0 \vee \phi_1\}$ to a successor node with label $\Gamma \cup \{\phi_0\}$ if $\phi_0 \in \delta(\phi_0 \vee \phi_1, l)$, that is, if ϕ_0 is a deferral. A *trace* (of ϕ_0) along a branch with root v (whose label contains ϕ_0) encoded by a word w is a (finite or infinite) sequence $t = \phi_0, \phi_1 \dots$ of formulae such that $\phi_{i+1} \in \delta(\phi_i, w(i))$. A *tableau* is a finite pre-tableau in which all traces are finite, and a *tableau for* ρ_1 if some node label contains ρ_1 .

Given a tableau (W, L) and a node $v \in W$, let $\text{tab}(v)$ (for *tableau timeout*) denote the least number m such that for all formulae ϕ in $l(v)$ and all branches of (W, L) that are rooted at v , all traces of ϕ along the branch have length at most m ; such an m always exists by the definition of tableaux.

To link models and tableaux, we next define an inductive measure on unfolding of least fixpoint formulae in models.

Definition 17 (Extension under timeouts). Let k be the greatest index of any least fixpoint variable in ψ , and let (W, N, I) be a finite neighbourhood model. Then a *timeout* is a vector $\bar{m} = (m_1, \dots, m_k)$ of natural numbers $m_i \leq |W|$. For $1 \leq i \leq k$ such that $m_i > 0$, we put $\bar{m} \circledast i = (m_1, \dots, m_{i-1}, m_i - 1, |W|, \dots, |W|)$. Then $\bar{m} >_l \bar{m} \circledast i$, where $>_l$ denotes lexicographic ordering. For $\phi \in \text{cl}$, we inductively define the *extension* $\llbracket \phi \rrbracket_{\bar{m}}$ under timeout \bar{m} by $\llbracket \phi \rrbracket_{\bar{m}} = \llbracket \phi \rrbracket$

for $\phi \notin \text{dfr}$ and, for $\phi \in \text{dfr}$, by

$$\begin{aligned} \llbracket \psi_0 \wedge \psi_1 \rrbracket_{\bar{m}} &= \llbracket \psi_0 \rrbracket_{\bar{m}} \cap \llbracket \psi_1 \rrbracket_{\bar{m}} \\ \llbracket \psi_0 \vee \psi_1 \rrbracket_{\bar{m}} &= \llbracket \psi_0 \rrbracket_{\bar{m}} \cup \llbracket \psi_1 \rrbracket_{\bar{m}} \\ \llbracket \langle a \rangle \psi \rrbracket_{\bar{m}} &= \{w \in W \mid \exists S \in N(a, w). S \subseteq \llbracket \psi \rrbracket_{\bar{m}}\} \\ \llbracket [a] \psi \rrbracket_{\bar{m}} &= \{w \in W \mid \forall S \in N(a, w). S \cap \llbracket \psi \rrbracket_{\bar{m}} \neq \emptyset\} \\ \llbracket \mu X. \psi \rrbracket_{\bar{m}} &= \begin{cases} \emptyset & m_{\text{idx}(X)} = 0 \\ \llbracket \psi[\mu X. \psi / X] \rrbracket_{\bar{m} @ \text{idx}(X)} & m_{\text{idx}(X)} > 0 \end{cases} \end{aligned}$$

using the lexicographic ordering on (\bar{m}, ϕ) as the termination measure; crucially, unfolding least fixpoints reduces the timeout. We extend this definition to sets of formulae by $\llbracket \Psi \rrbracket_{\bar{m}} = \bigcap_{\phi \in \Psi} \llbracket \phi \rrbracket_{\bar{m}}$ for $\Psi \subseteq \text{cl}$.

Lemma 18. *In finite neighbourhood models, we have that for all $\psi \in \text{cl}$ there is some timeout \bar{m} such that*

$$\llbracket \psi \rrbracket \subseteq \llbracket \psi \rrbracket_{\bar{m}}.$$

Proof. Let W be the set of states. Since $\llbracket \psi \rrbracket_{\bar{m}}$ is defined like $\llbracket \psi \rrbracket$ in all cases but one, we only need to consider the inductive case where $\psi = \mu X. \psi_0 \in \text{cl}$. We show that for all *subformulae* ϕ of ψ_0 , for all timeout vectors $\bar{m} = (m_1, \dots, m_k)$ such that $m_i = |W|$ for all i such that ϕ does not contain a free variable with index at least i , we have

$$\llbracket \phi \rrbracket_{\sigma(\bar{m})} \subseteq \llbracket \theta^*(\phi) \rrbracket_{\bar{m}}, \quad (1)$$

where $\sigma(\bar{m})$ maps each $Y \in \text{FV}(\phi)$ to $(\llbracket \phi_1 \rrbracket_{\sigma(\bar{m})}^Y)^{m_{\text{idx}(Y)}}(\emptyset)$ where $\theta(Y) = \mu Y. \phi_1$ (this is a recursive definition of $\sigma(\bar{m})$ since the value of $\sigma(\bar{m})$ for Y is overwritten in $(\llbracket \phi_1 \rrbracket_{\sigma(\bar{m})}^Y)^{m_{\text{idx}(Y)}}(\emptyset)$ so that this set depends only on values m_i such that $i < \text{idx}(Y)$). This shows that the claimed property $\llbracket \mu X. \psi_0 \rrbracket \subseteq \llbracket \mu X. \psi_0 \rrbracket_{\bar{m}}$ holds for $\bar{m} = (|W|, \dots, |W|)$: By Kleene's fixpoint theorem,

$$\begin{aligned} \llbracket \mu X. \psi_0 \rrbracket &= (\llbracket \psi_0 \rrbracket^X)^{|W|}(\emptyset) = \llbracket \psi_0 \rrbracket_{[X \mapsto (\llbracket \psi_0 \rrbracket^X)^{|W|-1}(\emptyset)]} = \llbracket \psi_0 \rrbracket_{\sigma(\bar{m} @ \text{idx}(X))} \\ &\subseteq \llbracket \theta^*(\psi_0) \rrbracket_{\bar{m} @ \text{idx}(X)} = \llbracket \mu X. \psi_0 \rrbracket_{\bar{m}}, \end{aligned}$$

using (1) in the second-to-last step.

The proof of (1) is by induction on ϕ ; we do only the non-trivial cases. If $\theta^*(\phi) \notin \text{dfr}$, then ϕ is closed (since ψ_0 does not contain a free greatest fixpoint variable and since ϕ is not in the scope of a greatest fixpoint operator within ψ_0 , that is, no free greatest fixpoint variable is introduced during the induction). Hence we have $\llbracket \phi \rrbracket_{\sigma(\bar{m})} = \llbracket \phi \rrbracket$ and $\llbracket \theta^*(\phi) \rrbracket_{\bar{m}} = \llbracket \theta^*(\phi) \rrbracket = \llbracket \phi \rrbracket$ so that we are done. If $\phi = Y$, then $\llbracket Y \rrbracket_{\sigma(\bar{m})} = (\llbracket \phi_1 \rrbracket_{\sigma(\bar{m})}^Y)^{m_{\text{idx}(Y)}}(\emptyset)$ where $\theta(Y) = \mu Y. \phi_1$. If $m_{\text{idx}(Y)} = 0$, then $(\llbracket \phi_1 \rrbracket_{\sigma(\bar{m})}^Y)^{m_{\text{idx}(Y)}}(\emptyset) = \emptyset$ so there is nothing to show. If

$m_{\text{id}\times(Y)} > 0$, then

$$\begin{aligned}
(\llbracket \phi_1 \rrbracket_{\sigma(\bar{m})}^Y)^{m_{\text{id}\times(Y)}}(\emptyset) &= \llbracket \phi_1 \rrbracket_{\sigma(\bar{m})}^Y((\llbracket \phi_1 \rrbracket_{\sigma(\bar{m})}^Y)^{m_{\text{id}\times(Y)}-1}(\emptyset)) \\
&= \llbracket \phi_1 \rrbracket_{(\sigma(\bar{m}))_{[Y \mapsto (\llbracket \phi_1 \rrbracket_{\sigma(\bar{m})}^Y)^{m_{\text{id}\times(Y)}-1}(\emptyset)]}} \\
&\subseteq \llbracket \theta^*(\phi_1) \rrbracket_{\bar{m} @ \text{id}\times(Y)} \\
&= \llbracket \theta^*(\phi_1[\mu Y. \phi_1/Y]) \rrbracket_{\bar{m} @ \text{id}\times(Y)} \\
&= \llbracket \theta^*(\mu Y. \phi_1) \rrbracket_{\bar{m}} = \llbracket \theta^*(Y) \rrbracket_{\bar{m}},
\end{aligned}$$

where the inclusion is by the inductive hypothesis. If $\phi = \mu Y. \phi_1$, then

$$\begin{aligned}
\llbracket \mu Y. \phi_1 \rrbracket_{\sigma(\bar{m})} &= (\llbracket \phi_1 \rrbracket_{\sigma(\bar{m})}^Y)^{|W|}(\emptyset) = \llbracket \phi_1 \rrbracket_{\sigma(\bar{m})}^Y((\llbracket \phi_1 \rrbracket_{\sigma(\bar{m})}^Y)^{|W|-1}(\emptyset)) \\
&= \llbracket \phi_1 \rrbracket_{(\sigma(\bar{m}))_{[Y \mapsto (\llbracket \phi_1 \rrbracket_{\sigma(\bar{m})}^Y)^{|W|-1}(\emptyset)]}} \\
&\subseteq \llbracket \theta^*(\phi_1) \rrbracket_{\bar{m} @ \text{id}\times(Y)} \\
&= \llbracket \theta^*(\phi_1[\theta(Y)/Y]) \rrbracket_{\bar{m} @ \text{id}\times(Y)} \\
&= \llbracket \theta^*(\mu Y. \phi_1) \rrbracket_{\bar{m}},
\end{aligned}$$

where the first equality is by Kleene's fixpoint theorem and the inclusion is by the inductive hypothesis since $m_{\text{id}\times(Y)} = |W|$ (hence $m @ \text{id}\times(Y)_{\text{id}\times(Y)} = |W| - 1$) by assumption as $\mu Y. \phi_1$ does not contain a free variable with index at least $\text{id}\times(Y)$. \square

To check satisfiability it suffices to decide whether a tableau exists:

Theorem 19. *There is a tableau for ρ_1 if and only if ρ_1 is ρ_0 -satisfiable.*

Proof. Let ρ_1 be ρ_0 -satisfiable. Then there is a finite neighbourhood model (W, N, I) such that $W \subseteq \llbracket \rho_0 \rrbracket$ and $W \cap \llbracket \rho_1 \rrbracket \neq \emptyset$ by Lemma 8. We define a tableau over the set $V = \{(x, \Psi, w) \in W \times \mathcal{P}(\text{cl}) \times \Sigma_p^* \mid x \in \llbracket \Psi \rrbracket, \text{u}(\Psi) \leq 3n - |w|\}$, where $\text{u}(\Psi)$ denotes the sum of the numbers of unguarded operators in formulae from Ψ . Let $(x, \Psi, w) \in V$. For $\Psi \notin \text{states}$, $\text{u}(\Psi) \leq 3n - |w|$, we pick some $\phi \in \Psi$, distinguishing cases. If $\phi = \phi_0 \wedge \phi_1$ or $\phi = \eta X. \phi_0$, then we put $b = 0$. If $\phi = \phi_0 \vee \phi_1$, then let \bar{m} be the least timeout such that $x \in \llbracket \phi \rrbracket_{\bar{m}}$ (such \bar{m} exists by Lemma 18). Then there is some $b' \in \{0, 1\}$ such that $x \in \llbracket \phi_{b'} \rrbracket_{\bar{m}}$ and we put $b = b'$. In any case, we put $l = (\phi, b)$ and add an edge from (x, Ψ, w) to $(x, \gamma(\Psi, l), (w, l))$ to L , having $\text{u}(\gamma(\Psi, l)) = \text{u}(\Psi) - 1$ and hence $\text{u}(\gamma(\Psi, l)) \leq 3n - |w, l|$. Since we are interested in the nodes that can be reached from nodes (x, Ψ, ϵ) with $|\Psi| \leq 3$ and since each formula in Ψ contains at most n unguarded operators, we indeed only have to construct nodes (x, Ψ', w) with $\text{u}(\Psi) \leq 3n - |w|$ before reaching state labeled nodes. For $\Psi \in \text{states}$ and $\{\langle a \rangle \phi_0, [a] \phi_1\} \subseteq \Psi$, let \bar{m} be the least timeout such that $x \in \llbracket \{\langle a \rangle \phi_0, [a] \phi_1\} \rrbracket_{\bar{m}}$ (again, such \bar{m} exists by Lemma 18). Then there is $S \in N(a, x)$ such that $S \subseteq \llbracket \phi_0 \rrbracket_{\bar{m}} \cap \llbracket \rho_0 \rrbracket$ and $S \cap \llbracket \phi_1 \rrbracket_{\bar{m}} \neq \emptyset$. Pick $y \in \llbracket \phi_0 \rrbracket_{\bar{m}} \cap \llbracket \phi_1 \rrbracket_{\bar{m}} \cap \llbracket \rho_0 \rrbracket$ and add an edge from (x, Ψ, w) to $(y, \{\phi_0, \phi_1, \rho_0\}, \epsilon)$ to L , having $\text{u}(\{\phi_0, \phi_1, \rho_0\}) \leq 3n - |\epsilon|$. Define the label $l(x, \Psi, w)$ of nodes (x, Ψ, w) to be just Ψ . Then the structure (V, L) indeed is a tableau for ρ_1 : Since there is $z \in W \cap \llbracket \rho_1 \rrbracket$, we have $(z, \{\rho_1\}, \epsilon) \in V$, that is, ρ_1 is contained in the label

of some node in (V, L) . The requirements of tableaux for matching rule applications are satisfied by construction of L . It remains to show that each trace in (V, L) is finite. So let $\tau = \phi_0, \phi_1, \dots$ be a trace of some formula ϕ_0 along some branch (encoded by a word w) that is rooted at some node $v = (x, \Psi, w') \in V$ such that $\phi_0 \in \Psi$. Let \bar{m} be the least timeout such that $x \in \llbracket \phi_0 \rrbracket_{\bar{m}}$ (again, such \bar{m} exists by Lemma 18). For each i , we have $\phi_{i+1} \in \delta(\phi_i, w(i))$ and there are $(x_i, \Psi_i, w_i) \in V$ and \bar{m}_i such that $\phi_i \in \Psi_i$ and $x_i \in \llbracket \phi_i \rrbracket_{\bar{m}_i}$. We have chosen disjuncts and modal successors in a minimal fashion, that is, in such a way that we always have $\bar{m}_{i+1} \leq_l \bar{m}_i$. Since traces can be infinite only if they contain infinitely many unfolding steps for some least fixpoint, τ is finite or some least fixpoint is unfolded in it infinitely often. In the former case, we are done. In the latter case, each unfolding of a least fixpoint by Definition 17 reduces some digit of the timeout so that we have an infinite decreasing chain $\bar{m}_{i_1} >_l \bar{m}_{i_2} >_l \dots$ with $i_{j+1} > i_j$ for all j , which is a contradiction to $<_l$ being a well-order.

For the converse direction, let (V, L) be a tableau for ρ_1 labeled by some function $l : V \rightarrow \mathcal{P}(\text{cl})$. We construct a model (W, N, I) over the set

$$W = \{x \in V \mid l(x) \in \text{states}\}.$$

For $p \in \mathsf{P}$, put $I(p) = \{x \in W \mid p \in l(x)\}$; since (V, L) is a tableau, the rules (\perp) and $(\not\leq)$ do not match the label of any node, so we have $I(p) = W \setminus I(\bar{p})$, as required. Let $x \in W$ and $a \in \mathsf{A}$. If $l(x)$ contains no a -box literal, then we put $N(a, x) = \{\emptyset\}$. If $l(x)$ contains some a -box literal but no a -diamond literal, then we put $N(a, x) = \emptyset$. Otherwise, let the a -modalities in $l(x)$ be exactly $\langle a \rangle \chi_1, \dots, \langle a \rangle \chi_o, [a] \psi_1, \dots, [a] \psi_m$. We put $N(a, x) = \{\{y_{1,1}, \dots, y_{1,m}\}, \dots, \{y_{o,1}, \dots, y_{o,m}\}\}$, where, for $1 \leq i \leq o$, $1 \leq j \leq m$, the state $y_{i,j}$ is picked minimally with respect to **tab** among all nodes z such that $(x, z) \in L$ and $\{\chi_i, \psi_j, \rho_0\} = l(z)$; such $y_{i,j}$ with minimal tableau timeout always exists since (V, L) is a tableau. It remains to show that (W, N, I) is a ρ_0 -model for ρ_1 . We put

$$\llbracket \widehat{\phi} \rrbracket = \{x \in W \mid l(x) \vdash_{\text{PL}} \phi\}$$

for $\phi \in \text{cl}$, where \vdash_{PL} denotes *propositional entailment* (modal literals $[a]\phi$, $\langle a \rangle \phi$ are regarded as propositional atoms and ηX . ϕ and $\phi[\eta X$. $\phi/X]$ entail each other). Since we have $W \subseteq \llbracket \widehat{\rho_0} \rrbracket$ and $W \cap \llbracket \widehat{\rho_1} \rrbracket \neq \emptyset$ by definition of (W, N, I) and tableaux, it suffices to show that we have $\llbracket \widehat{\phi} \rrbracket \subseteq \llbracket \phi \rrbracket$ for all $\phi \in \text{cl}$. The proof of this is by induction on ϕ , using coinduction in the case for greatest fixpoint formulae and a further induction on tableau timeouts in the case for least fixpoint formulae. □

6 Satisfiability Games

We now define a game characterizing ρ_0 -satisfiability of ρ_1 . Player Eloise tries to establish the existence of a tableau for ρ_1 using only polynomially many supporting points for her reasoning. To this end, sequences of propositional reasoning steps are contracted into single Eloise-moves. Crucially, the limited branching of

monotone modalities (and the ensuing limited need for tracking of deferrals) has a restricting effect on the nondeterminism that the game needs to take care of.

Definition 20 (Satisfiability games). We put $U = \{\Psi \subseteq \text{cl} \mid 1 \leq |\Psi| \leq 2\}$ (note $|U| \leq n^2$) and $Q = \{\text{Foc} \subseteq \text{dfr} \mid |\text{Foc}| \leq 2\}$. The ρ_0 -satisfiability game for ρ_1 is the Büchi game $G = (V, E, v_0, F)$ with set $V = V_{\exists} \cup V_{\forall}$ of nodes (with the union made disjoint by markers omitted in the notation) where $V_{\exists} = \{(\Psi, \text{Foc}) \in U \times Q \mid \text{Foc} \subseteq \Psi\}$ and $V_{\forall} = \text{states} \times Q$, with initial node $v_0 = (\{\rho_1\}, \emptyset) \in V_{\exists}$, and with set $F = \{(\Psi, \text{Foc}) \in V_{\exists} \mid \text{Foc} = \emptyset\}$ of accepting nodes. The set E of moves is defined by

$$\begin{aligned} E(\Psi, \text{Foc}) &= \{(\gamma(\Psi \cup \{\rho_0\}), w), \delta(\text{Foc}, w)\} \in \text{states} \times Q \mid w \in (\Sigma_p)^*, |w| \leq 3n\} \\ E(I, \text{Foc}) &= \{(\{\phi_0, \phi_1\}, \text{Foc}') \in U \times Q \mid \{\langle a \rangle \phi_0, [a] \phi_1\} \subseteq I, \\ &\quad \text{if } \text{Foc} \neq \emptyset, \text{ then } \text{Foc}' = \delta(\text{Foc}, (\langle a \rangle \phi_0, [a] \phi_1)), \\ &\quad \text{if } \text{Foc} = \emptyset, \text{ then } \text{Foc}' = \{\phi_0, \phi_1\} \cap \text{dfr}\} \end{aligned}$$

for $(\Psi, \text{Foc}) \in V_{\exists}$, $(I, \text{Foc}) \in V_{\forall}$.

Thus, Eloise steers the propositional evolution of formula sets into formal states, keeping track of the focussed formulae, while Abelard picks an application of Lemma 2, and resets the focus set after it is *finished*, i.e. becomes \emptyset ; Eloise wins plays in which the focus set is finished infinitely often.

Remark 21. It is crucial that while the game has exponentially many Abelard-nodes, there are only polynomially many Eloise-nodes. In fact, all Eloise-nodes (Ψ, Foc) have $\text{Foc} \subseteq \Psi$, so the game has at most $4|U| \leq 4n^2$ Eloise-nodes.

Next we prove the correctness of our satisfiability games.

Theorem 22. *There is a tableau for ρ_1 if and only if Eloise wins G .*

Proof. Let s be a winning strategy for Eloise in G with which she wins every node in her winning region win_{\exists} . For $v = (\Psi, \text{Foc}) \in \text{win}_{\exists}$, we let $w_{s(v)}$ denote a fixed propositional word such that $s(\Psi, \text{Foc}) = (\gamma(\Psi, w_{s(v)}), \delta(\text{Foc}, w_{s(v)}))$, that is, a witness word for the move of Eloise that s prescribes at v . We construct a tableau (W, L) over the set

$$\begin{aligned} W &= \{(\gamma(\Psi \cup \{\rho_0\}), w'), \delta(\text{Foc}, w')\} \in \mathcal{P}(\text{cl}) \times \mathcal{P}(\text{dfr}) \mid \\ &\quad (\Psi, \text{Foc}) \in \text{win}_{\exists}, w' \text{ is a prefix of } w_{s(\Psi, \text{Foc})}\}. \end{aligned}$$

We define the label $l(\Phi, \text{Foc}')$ of nodes from $(\Phi, \text{Foc}') \in W$ to be just Φ . Let $(\Phi, \text{Foc}') = (\gamma(\Psi \cup \{\rho_0\}), w'), \delta(\text{Foc}, w') \in W$. If w' is a proper prefix of $w_{s(\Psi, \text{Foc})}$, then we have $\Phi \notin \text{states}$; let $l \in \Sigma_p$ be the letter such that (w', l) is a prefix of $w_{s(\Psi, \text{Foc})}$ and add the pair $((\Phi, \text{Foc}'), (\gamma(\Phi, l), \delta(\text{Foc}', l)))$ to L . If $w' = w_{s(\Psi, \text{Foc})}$, then we have $\Phi \in \text{states}$. For $\{\langle a \rangle \phi, [a] \chi\} \subseteq \Phi$, we distinguish cases. If $\text{Foc}' = \emptyset$, then put $\text{Foc}'' = \{\phi, \chi\} \cap \text{dfr}$; otherwise, put $\text{Foc}'' = \delta(\text{Foc}', (\langle a \rangle \phi, [a] \chi))$. Then add the pair $((\Phi, \text{Foc}'), (\{\phi, \chi, \rho_0\}, \text{Foc}''))$ to L , having $(\{\phi, \chi\}, \text{Foc}'') \in \text{win}_{\exists}$.

It remains to show that all traces in (W, L) are finite. So let $\tau = \phi_0, \phi_1, \dots$ be a trace along some branch (encoded by a word w) that is rooted at some node $(\Phi, \text{Foc}') \in W$ such that $\phi_0 \in \Phi$. By construction, this branch gives rise to an s -play $(\Psi_0, \text{Foc}_0), (\Gamma_0, \text{Foc}'_0), (\Psi_1, \text{Foc}_1), (\Gamma_1, \text{Foc}'_1), \dots$ that starts at $(\Psi_0, \text{Foc}_0) = (\Psi, \text{Foc})$. Let i be the least position such that $\text{Foc}'_i = \emptyset$ (i exists because s is a winning strategy). Since the ϕ_j are tracked along rule applications, we have $\phi_i \in \Gamma_i$; hence $\phi_{i+1} \in \text{Foc}_{i+1}$. Let i' be the least position greater than i such that $\text{Foc}'_{i'} = \emptyset$ (again, i' exists because s is a winning strategy). Between $(\Psi_{i+1}, \text{Foc}_{i+1})$ and $(\Gamma_{i'}, \text{Foc}'_{i'})$, all formulae from Foc_{i+1} (including ϕ_{i+1}) are transformed to a non-deferral by the formula manipulations encoded in w . In particular, the trace τ ends between node (Ψ_i, Foc_i) and node $(\Psi_{i'}, \text{Foc}_{i'})$, and hence is finite.

For the converse direction, let (W, L) be tableau for ρ_1 , labeled with $l : W \rightarrow \mathcal{P}(\text{cl})$. We extract a strategy s for Eloise in G . A game node (Ψ, Foc) is *realized* if there is $v \in W$ such that $\Psi \subseteq l(v)$; then we say that v *realizes* the game node. For all realized game nodes (Ψ, Foc) , we pick a realizing tableau node $v(\Psi, \text{Foc})$ that is minimal with respect to tab among the tableau nodes that realize (Ψ, Foc) . Then we construct a propositional word $w = l_0, l_1, \dots$ as follows, starting with $\Psi_0 = \Psi \cup \{\rho_0\}$ and $v_0 = v(\Psi, \text{Foc})$. For $i \geq 0$, pick some non-modal letter $l_i = (\phi, b)$ such that $\phi \in \Psi_i$, $b \in \{0, 1\}$ and such that $l(v_{i+1}) = \gamma(\Psi_i, l_i)$ where $v_{i+1} \in W$ is the node such that $(v_i, v_{i+1}) \in L$. Such a letter l_i exists since (W, L) is a tableau. By guardedness of fixpoint variables, this process will eventually terminate with a word $w = l_0, l_1, \dots, l_m$ such that $m \leq 3n$, since Ψ_0 contains at most three formulae and each formula contains at most n unguarded operators. Put $s(\Psi, \text{Foc}) = (\gamma(\Psi \cup \{\rho_0\}, w), \delta(\text{Foc}, w))$, having $\gamma(\Psi \cup \{\rho_0\}, w) = l(v_m)$. It remains to show that s is a winning strategy. So let $\tau = (\Psi_0, \text{Foc}_0), (\Gamma_0, \text{Foc}'_0), (\Psi_1, \text{Foc}_1), (\Gamma_1, \text{Foc}'_1), \dots$ be an s -play, where $\Psi_0 = \{\rho_1\}$ and $\text{Foc}_0 = \emptyset$. It suffices to show that for all i such that $\text{Foc}_i \neq \emptyset$, there is $j \geq i$ such that $\text{Foc}_j = \emptyset$. So let $\text{Foc}_i \neq \emptyset$ and let w_i be the word that is constructed in the play from (Ψ_i, Foc_i) on. Since (W, L) is a tableau and since s has been constructed using realizing nodes with minimal tableau timeouts, all traces of formulae from Foc_i along the branch that is encoded by w_i are finite. Let j be the least number such that all such traces have ended after $2j$ further moves from (Ψ_i, Foc_i) . Then we have $\text{Foc}'_{i+j} = \emptyset$, as required. \square

Corollary 23. *Every satisfiable formula of size n in the alternation-free monotone μ -calculus with the universal modality has a model of size at most $4n^2$.*

Corollary 24. *The satisfiability checking problem for the alternation-free monotone μ -calculus with the universal modality is in NP (hence NP-complete).*

Proof. Guess a winning strategy s for Eloise in G and verify that s is a winning strategy. Verification can be done in polynomial time since the structure obtained from G by imposing s is of polynomial size and since the admissibility of single moves can be checked in polynomial time. \square

By the translations recalled in Example 7, we obtain moreover

Corollary 25. *Satisfiability-checking in concurrent propositional dynamic logic CPDL and in the alternation-free fragment of game logic is in NP (hence NP-complete).*

7 Conclusion

We have shown that satisfiability checking in the alternation-free fragment of the monotone μ -calculus with the universal modality is only NP-complete, even when formula size is measured as the cardinality of the closure. Via straightforward translations (which have only quadratic blow-up under the mentioned measure of formula size), it follows that both concurrent propositional dynamic logic (CPDL) and the alternation-free fragment of game logic are also only NP-complete under their original semantics, i.e. with atomic programs interpreted as neighbourhood structures (they become EXPTIME-complete when atomic programs are interpreted as relations). We leave as an open problem whether the upper bound NP extends to the full monotone μ -calculus, for which the best known upper bound thus remains EXPTIME, by results on the coalgebraic μ -calculus [5], or alternatively by the translation into the relational μ -calculus that we give in the proof of the finite model property (Lemma 8).

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook. Cambridge University Press, Cambridge (2003)
2. Bradfield, J., Stirling, C.: Modal μ -calculi. In: Handbook of Modal Logic, pp. 721–756. Elsevier (2006)
3. Bruse, F., Friedmann, O., Lange, M.: On guarded transformation in the modal μ -calculus. Log. J. IGPL **23**(2), 194–216 (2015)
4. Chellas, B.F.: Modal Logic. Cambridge University Press, Cambridge (1980)
5. Cirstea, C., Kupke, C., Pattinson, D.: EXPTIME tableaux for the coalgebraic μ -calculus. Log. Methods Comput. Sci. **7**(3:03), 1–33 (2011)
6. Donini, F.: Complexity of reasoning. In: Baader, F., et al. [1], pp. 96–136
7. Enqvist, S., Hansen, H.H., Kupke, C., Marti, J., Venema, Y.: Completeness for game logic. In: Logic in Computer Science (LICS 2019), pp. 1–13. IEEE (2019)
8. Enqvist, S., Seifan, F., Venema, Y.: Monadic second-order logic and bisimulation invariance for coalgebras. In: Logic in Computer Science (LICS 2015), pp. 353–365. IEEE (2015)
9. Fischer, M., Ladner, R.: Propositional dynamic logic of regular programs. J. Comput. Syst. Sci. **18**, 194–211 (1979)
10. Friedmann, O., Lange, M.: Deciding the unguarded modal μ -calculus. J. Appl. Non-Classical Log. **23**, 353–371 (2013)
11. Goranko, V., Passy, S.: Using the universal modality: gains and questions. J. Log. Comput. **2**, 5–30 (1992)
12. Hansen, H., Kupke, C.: A coalgebraic perspective on monotone modal logic. In: Adámek, J., Milius, S. (eds.) Coalgebraic Methods in Computer Science (CMCS 2004) of ENTCS, vol. 106, pp. 121–143. Elsevier (2004)

13. Hausmann, D., Schröder, L., Egger, C.: Global caching for the alternation-free coalgebraic μ -calculus. In: Concurrency Theory (CONCUR 2016) of LIPIcs, vol. 59, pp. 34:1–34:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2016)
14. Kracht, M., Wolter, F.: Normal monomodal logics can simulate all others. *J. Symb. Log.* **64**(1), 99–138 (1999)
15. Ladner, R.: The computational complexity of provability in systems of modal propositional logic. *SIAM J. Comput.* **6**, 467–480 (1977)
16. Lange, M., Stirling, C.: Focus games for satisfiability and completeness of temporal logic. In: Logic in Computer Science (LICS 2001), pp. 357–365. IEEE Computer Society (2001)
17. Mazala, R.: Infinite games. In: Grädel, E., Thomas, W., Wilke, T. (eds.) Automata Logics, and Infinite Games. LNCS, vol. 2500, pp. 23–38. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-36387-4_2
18. Miyano, S., Hayashi, T.: Alternating finite automata on ω -words. *Theory Comput. Sci.* **32**, 321–330 (1984)
19. Niwinski, D., Walukiewicz, I.: Games for the μ -calculus. *Theory Comput. Sci.* **163**, 99–116 (1996)
20. Parikh, R.: The logic of games and its applications. *Ann. Discrete Math.* **24**, 111–140 (1985)
21. Pauly, M.: Bisimulation for general non-normal modal logic. Unpublished manuscript (1999)
22. Pauly, M.: Logic for social software. Ph.D. thesis, Universiteit van Amsterdam (2001)
23. Pauly, M., Parikh, R.: Game logic - an overview. *Stud. Log.* **75**(2), 165–182 (2003)
24. Peleg, D.: Concurrent dynamic logic. *J. ACM* **34**, 450–479 (1987)
25. Pratt, V.: A decidable mu-calculus: preliminary report. In: Foundations of Computer Science (FOCS 1981), pp. 421–427. IEEE Computer Society (1981)
26. van Benthem, J., Bezhanishvili, N., Enqvist, S.: A propositional dynamic logic for instantial neighborhood semantics. *Stud. Log.* **107**(4), 719–751 (2019)
27. Vardi, M.: On the complexity of epistemic reasoning. In: Logic in Computer Science (LICS 1989), pp. 243–252. IEEE Computer Society (1989)