Optimal Satisfiability Checking for Arithmetic μ-Calculi

Daniel Hausmann(✉) and Lutz Schröder

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

Abstract. The coalgebraic μ-calculus provides a generic semantic framework for fixpoint logics with branching types beyond the standard relational setup, e.g. probabilistic, weighted, or game-based. Previous work on the coalgebraic μ-calculus includes an exponential time upper bound on satisfiability checking, which however requires a well-behaved set of tableau rules for the next-step modalities. Such rules are not available in all cases of interest, in particular ones involving either integer weights as in the graded μ-calculus, or real-valued weights in combination with non-linear arithmetic. In the present work, we prove the same upper complexity bound under more general assumptions, specifically regarding the complexity of the (much simpler) satisfiability problem for the underlying *one-step logic*, roughly described as the nesting-free next-step fragment of the logic. The bound is realized by a generic global caching algorithm that supports on-the-fly satisfiability checking. Example applications include new exponential-time upper bounds for satisfiability checking in an extension of the graded μ-calculus with polynomial inequalities (including positive Presburger arithmetic), as well as an extension of the (two-valued) probabilistic μ-calculus with polynomial inequalities.

1 Introduction

Modal fixpoint logics are a well-established tool in the temporal specification, verification, and analysis of concurrent systems. One of the most expressive logics of this type is the modal μ-calculus [2,3,20], which features explicit least and greatest fixpoint operators; roughly speaking, these serve to specify liveness properties (least fixpoints) and safety properties (greatest fixpoints), respectively. Like most modal logics, the modal μ-calculus is traditionally interpreted over relational models such as Kripke frames or labelled transition systems. The growing interest in more expressive models where transitions are governed, e.g., by probabilities, weights, or games has sparked a commensurate growth of temporal logics and fixpoint logics interpreted over such systems; prominent examples include probabilistic μ-calculi [5,17,24], the alternating-time μ-calculus [1], and the monotone μ-calculus, which contains Parikh's game logic [28]. The graded μ-calculus [21] features next-step modalities that count successors; it is standardly interpreted over Kripke frames but, as pointed out by D'Agostino and

M. Bojańczyk and A. Simpson (Eds.): FOSSACS 2019, LNCS 11425, pp. 277–294, 2019.
https://doi.org/10.1007/978-3-030-17127-8_16

Visser [6], graded modalities are more naturally interpreted over so-called multigraphs, where edges carry integer weights, and in fact this modification leads to better bounds on minimum model size for satisfiable formulas.

Coalgebraic logic [29,34] has emerged as a unifying framework for modal logics interpreted over such more general models. It is based on casting the transition type of the systems at hand as a set functor, and the systems in question as coalgebras for this type functor, following the paradigm of universal coalgebra [31]; additionally, modalities are interpreted as so-called *predicate liftings*. The *coalgebraic μ-calculus* [4] caters for fixpoint logics within this framework, and essentially covers all mentioned (two-valued) examples as instances. It has been shown that satisfiability checking in a coalgebraic μ-calculus is in ExpTime, *provided* that one exhibits a set of tableau rules for the modalities, so-called *one-step rules*, that is *tractable* in a suitable sense (an assumption made also in our own previous work on the flat [14] and alternation-free [16] fragments of the coalgebraic μ-calculus). Such rules are known for many important cases, notably including alternating-time logics, the probabilistic μ-calculus even when extended with linear inequalities, and game logic [4,22,36]. There are, however, important cases where such rule sets are currently missing, and where there is in fact little perspective for finding suitable rules. One prominent case of this kind is graded modal logic; further cases arise when logics over systems with non-negative real weights, such as probabilistic systems, are taken beyond linear arithmetic to include polynomial inequalities.

The object of the current paper is to fill this gap by proving a generic ExpTime upper bound for coalgebraic μ-calculi in the absence of tractable sets of modal tableau rules. The method we use instead is to analyse the so-called *one-step satisfiability* problem of the logic on a semantic level – this problem is essentially the satisfiability problem of a very small fragment of the logic, the *one-step logic*, which excludes not only fixpoints, but also nested next-step modalities, with a correspondingly simplified semantics that no longer involves actual transitions. E.g. the one-step logic of the relational μ-calculus is interpreted over models essentially consisting of a set with a distinguished subset, abstracting the successors of a single state that is not itself part of the model. We have applied this principle to satisfiability checking in coalgebraic (next-step) modal logics [35], coalgebraic hybrid logics [26], and reasoning with global assumptions in coalgebraic modal logics [23]. It also appears implicitly in work on automata for the coalgebraic μ-calculus [8], which however establishes only a doubly exponential upper bound in the case without tractable modal tableau rules.

Our main example applications are on the one hand the graded modal μ-calculus and its extension with (monotone) polynomial inequalities, including Presburger modalities, i.e. (monotone) linear inequalities, and on the other hand the extension of the (two-valued) probabilistic μ-calculus [4,24] with (monotone) polynomial inequalities. While the graded μ-calculus as such is known to be in ExpTime [21], the other mentioned instances of our result are, to our best knowledge, new. At the same time, our proofs are fairly simple, even compared to specific ones, e.g. for the graded μ-calculus.

Technically, we base our results on an automata-theoretic treatment by means of standard parity automata with singly exponential branching degree (in particular on modal steps), thus precisely enabling the singly exponential upper bound, in contrast to previous work in [8] where the introduced Λ-automata lead to doubly exponential branching on modal steps in the resulting satisfiability games. Our algorithm witnessing the singly exponential time bound is, in fact, a global caching algorithm [11,12], and is able to decide the satisfiability of nodes on-the-fly, that is, possibly before the tableau is fully expanded, thus offering a perspective for practically feasible reasoning. A side result of our approach is a criterion for a polynomial bound on branching in models, which holds in all our examples.

Organization. In Sect. 2, we recall the basics of the coalgebraic μ-calculus. We outline our automata-theoretic approach in Sect. 3, and present the global caching algorithm and its runtime analysis in Sect. 4. Soundness and completeness of the algorithm are proved in Sect. 5.

2 The Coalgebraic μ-Calculus

We recall basic definitions in coalgebraic logic [29,34] and the coalgebraic μ-calculus [4].

Syntax. We fix a *modal similarity type* Λ, that is, a set of modal operators with assigned finite arities, possibly including propositional atoms as nullary modalities. For readability, we restrict the technical development to unary modalities, noting that all proofs generalize to higher arities by just writing more indices; in fact, we will liberally use higher arities in examples. We assume that Λ is closed under duals, i.e., that for each modal operator $\heartsuit \in \Lambda$, there is a *dual* $\overline{\heartsuit} \in \Lambda$ such that $\overline{\overline{\heartsuit}} = \heartsuit$ for all $\heartsuit \in \Lambda$. Let $\mathbf{V}$ be an infinite set of *fixpoint variables*. Formulas of the *coalgebraic μ-calculus* (over Λ) are given by the grammar

$$\psi, \phi ::= \bot \mid \top \mid \psi \wedge \phi \mid \psi \vee \phi \mid \heartsuit\phi \mid X \mid \mu X.\, \psi \mid \nu X.\, \psi \qquad \heartsuit \in \Lambda, X \in \mathbf{V}.$$

As usual, μ and ν take least and greatest fixpoints, respectively. Negation is not included but can be defined as usual. Throughout, we use $\eta \in \{\mu, \nu\}$ as a placeholder for fixpoint operators; we briefly refer to formulas of the form $\eta X.\, \phi$ as *fixpoints*. Fixpoint operators *bind* their fixpoint variables, so that we have standard notions of bound and free fixpoint variables; a formula is closed if it contains no free fixpoint variables. We assume w.l.o.g. that all formulas are *clean*, i.e. each fixpoint variable appears in at most one fixpoint operator, and *irredundant*, i.e. each bound variable is used at least once. Moreover, we restrict to *guarded* formulas, in which all occurrences of fixpoint variables are separated by at least one modal operator from their binding fixpoint operator (this is standard although possibly not w.l.o.g. [9]). For $\heartsuit \in \Lambda$, we denote by $\mathsf{size}(\heartsuit)$ the length of a suitable representation of $\heartsuit$; for natural or rational numbers indexing $\heartsuit$, we assume binary representation. The *length* $|\psi|$ of a formula is its length over the alphabet $\{\bot,\top,\wedge,\vee\}\cup\Lambda\cup\mathbf{V}\cup\{\eta X.\mid X\in\mathbf{V}\}$, while the *size* $\mathsf{size}(\psi)$ of ψ is defined by counting $\mathsf{size}(\heartsuit)$ for each $\heartsuit\in\Lambda$ (and 1 for all other operators). The *alternation depth* $\mathsf{ad}(\eta X.\psi)$ of a fixpoint $\eta X.\psi$ is the maximal depth of nesting of such alternating least and greatest fixpoints in ψ that depend on X, tweaked to be *even* for least fixpoint formulas and *odd* for greatest fixpoint formulas (that is, starting with $\mathsf{ad}(\mu X.\psi)=2$ and $\mathsf{ad}(\nu X.\psi)=1$ for closed ψ). For a more detailed definition of various flavours of alternation depth, see e.g. [27].

Semantics. As indicated above, the branching type of the underlying systems is a parameter of the framework, given by fixing a **Set**-endofunctor T. Elements of TU should be thought of as structured collections over U that serve as collections of successors of states – e.g. in the most basic example, classical relational systems, T is powerset $\mathcal{P}$. Formulas are then interpreted over *T-coalgebras* (C,ξ) consisting of a set C of *states* and a *transition function* $\xi : C\to TC$ that assigns a structured collection $\xi(x)\in TC$ of successors (and observations) to $x\in C$; e.g. $\mathcal{P}$-coalgebras are just Kripke frames, as they assign a set of successors to each state. We interpret each modal operator $\heartsuit\in\Lambda$ as a *T-predicate lifting* $[\![\heartsuit]\!]$, that is, a natural transformation $[\![\heartsuit]\!]:\mathcal{Q}\to\mathcal{Q}\circ T^{op}$ where $\mathcal{Q}:\mathbf{Set}^{op}\to\mathbf{Set}$ denotes the contravariant powerset functor. Predicate liftings thus are families of functions $[\![\heartsuit]\!]_U:\mathcal{Q}(U)\to\mathcal{Q}(TU)$ satisfying *naturality*, i.e. $[\![\heartsuit]\!]_U(f^{-1}[A])=(Tf)^{-1}[[\![\heartsuit]\!]_V(A)]$ for $f:U\to V$ and $A\subseteq V$, where f^{-1} denotes preimage. E.g. the standard relational box modality is interpreted by $[\![\Box]\!]_U(A)=\{B\in\mathcal{P}(U)\mid B\subseteq A\}$. For sets $U\subseteq V$, we write $\overline{U}=V\setminus U$ for the *complement* of U in V when V is understood from the context. We require that duality of modal operators is respected, i.e. $[\![\heartsuit]\!]_U(A)=\overline{[\![\overline{\heartsuit}]\!]_U\overline{A}}$ for $A\subseteq U$. To ensure existence of fixpoints, we require that all $[\![\heartsuit]\!]$ are *monotone*, i.e. $A\subseteq B\subseteq U$ implies $[\![\heartsuit]\!]_U(A)\subseteq[\![\heartsuit]\!]_U(B)$.

A *valuation* is a partial function $i:\mathbf{V}\rightharpoonup\mathcal{P}(C)$ that assigns sets $i(X)$ of states to fixpoint variables X. The *extension* $[\![\phi]\!]_i\subseteq C$ of a formula ϕ in a T-coalgebra (C,ξ) is defined by the expected clauses for propositional operators and

$$[\![\heartsuit\psi]\!]_i=\xi^{-1}[[\![\heartsuit]\!]_C([\![\psi]\!]_i)] \qquad [\![\mu X.\,\psi]\!]_i=\mathsf{LFP}([\![\psi]\!]_i^X)$$
$$[\![X]\!]_i=i(X) \qquad [\![\nu X.\,\psi]\!]_i=\mathsf{GFP}([\![\psi]\!]_i^X),$$

where LFP and GFP compute the least and greatest fixpoints of their argument functions, respectively, where $[\![\psi]\!]_i^X(A)=[\![\psi]\!]_{i[X\mapsto A]}$ for $A\subseteq C$, and where $(i[X\mapsto A])(X)=A$ and $(i[X\mapsto A])(Y)=i(Y)$ for $Y\neq X$. In particular, the extension is invariant under *unfolding* of fixpoints, i.e. $[\![\eta X.\,\psi]\!]_i=[\![\psi[X\mapsto\eta X.\,\psi]]\!]_i$. For closed formulas ψ, the valuation i is irrelevant, so we write $[\![\psi]\!]$ instead of $[\![\psi]\!]_i$. A state $x\in C$ *satisfies* a closed formula ψ (denoted $x\models\psi$) if $x\in[\![\psi]\!]$. Given a set Z, we define the set $\Lambda(Z)=\{\heartsuit z\mid\heartsuit\in\Lambda,z\in Z\}$ of *modal literals* (over Z). A closed formula χ is *satisfiable* if there is a coalgebra (C,ξ) and a state $x\in C$ such that $x\models\chi$.

Example 1. We now detail several instances of the coalgebraic μ-calculus; for further examples, e.g. the alternating-time μ-calculus, see [4].

1. To obtain the standard modal μ-calculus [19] (which contains CTL as a fragment), we take $\Lambda = \{\Diamond, \Box\} \cup P$ where P is a set of propositional atoms, seen as nullary modalities. The semantics is captured by $TU = \mathcal{P}(U) \times \mathcal{P}(P)$, so that T-coalgebras are Kripke models, as they assign to each state a set of successors and a set of atoms satisfied in the state. The relevant predicate liftings are

$$[\![\Diamond]\!]_U(A) = \{(B,Q) \in TU \mid A \cap B \neq \emptyset\} \quad [\![\Box]\!]_U(A) = \{(B,Q) \in TU \mid B \subseteq A\}$$

and $[\![p]\!]_U = \{(B,Q) \in TU \mid p \in Q\}$, a nullary predicate lifting. Standard example formulas include the CTL-formula $\mathsf{AF}\, p = \mu X.\,(p \vee \Box X)$, which states that on all paths, p eventually holds, and the fairness formula $\nu X.\,\mu Y.\,((p \wedge \Diamond X) \vee \Diamond Y)$, which asserts the existence of a path on which p holds infinitely often.
2. We interpret the *graded μ-calculus* [21] over multigraphs [6], i.e. T-coalgebras for the multiset functor $T = \mathcal{B}$, defined by

$$\mathcal{B}(U) = \{\theta : U \to \mathbb{N} \cup \{\infty\}\} \qquad \mathcal{B}(f)(\theta)(v) = \textstyle\sum_{u \in U \mid f(u)=v} \theta(u)$$

for sets U, V and functions $f : U \to V$, $\theta : U \to \mathbb{N} \cup \{\infty\}$. Thus $\mathcal{B}$-coalgebras (C, ξ) assign multisets $\xi(x)$ to states $x \in C$, with the intuition that x has $y \in C$ as successor with multiplicity m if $\xi(x)(y) = m$. We use the modal similarity type $\Lambda = \{\langle m \rangle, [m] \mid m \in \mathbb{N} \cup \{\infty\}\}$ and define the predicate liftings

$$[\![\langle m \rangle]\!]_U(A) = \{\theta \in \mathcal{B}(U) \mid \theta(A) > m\} \quad [\![[m]]\!]_U(A) = \{\theta \in \mathcal{B}(U) \mid \theta(\overline{A}) \leq m\}$$

for sets U and $A \subseteq U$, where $\theta(A) = \sum_{a \in A} \theta(a)$. E.g. a state satisfies $\nu X.\,(\psi \wedge \langle 1 \rangle X)$ if it is the root of an infinite binary tree in which ψ is satisfied globally.
3. Similarly, the two-valued *probabilistic μ-calculus* [4,24] is obtained by using the distribution functor $T = \mathcal{D}$ that maps sets U to probability distributions over U with countable support, defined by

$$\mathcal{D}(U) = \{d : U \to (\mathbb{Q} \cap [0,1]) \mid \textstyle\sum_{u \in U} d(u) = 1\}.$$

Then T-coalgebras are just Markov chains. We use the modal similarity type $\Lambda = \{\langle p \rangle, [p] \mid p \in \mathbb{Q} \cap [0,1]\}$ and define the predicate liftings

$$[\![\langle p \rangle]\!]_U(A) = \{d \in \mathcal{D}(U) \mid d(A) > p\} \quad [\![[p]]\!]_U(A) = \{d \in \mathcal{D}(U) \mid d(\overline{A}) \leq p\},$$

for sets U and $A \subseteq U$, where again $d(A) = \sum_{a \in A} d(a)$.
4. We interpret the *graded μ-calculus with polynomial inequalities* over the semantic domain from item 2 (i.e. multigraphs). We put $\Lambda = \{L_{p,b}, M_{p,b} \mid p \in \mathbb{N}_{>0}[X_1, \dots, X_n], b, n \in \mathbb{N}\}$ (that is, p ranges over multivariate polynomials with positive integer coefficients) and define the predicate liftings

$$[\![L_{p,b}]\!]_U(A_1, \dots, A_n) = \{\theta \in \mathcal{B}(U) \mid p(\theta(A_1), \dots, \theta(A_n)) > b)\}$$
$$[\![M_{p,b}]\!]_U(A_1, \dots, A_n) = \{\theta \in \mathcal{B}(U) \mid p(\theta(\overline{A_1}), \dots, \theta(\overline{A_n})) \leq b)\},$$

for sets U and $A_1, \dots, A_n \subseteq U$, where $\theta(A) = \sum_{a \in A} \theta(a)$. This logic subsumes the *Presburger μ-calculus*, that is, the extension of the graded μ-calculus with (monotone) linear inequalities, which may be seen as the fixpoint variant of *Presburger modal logic* [7]. E.g. the formula $\mu Y. (r \vee L_{2X_1+X_2^2,2}(p \wedge Y, q \wedge Y))$ says that the current state is the root of a finite tree all whose leaves satisfy r, and each of whose inner nodes has n_1 children satisfying p and n_2 children satisfying q where $2n_1 + n_2^2 > 2$. One sees an apparent coding of the logic into the graded μ-calculus, which however incurs exponential blowup.

5. Similarly, we use the semantic domain from item 3, Markov chains, to obtain the *probabilistic μ-calculus with polynomial inequalities* [23]: We put $\Lambda = \{L_{p,b}, M_{p,b} \mid p \in \mathbb{Q}_{>0}[X_1, \dots, X_n], b \in \mathbb{Q}_{\geq 0}, n \in \mathbb{N}\}$ (i.e. p ranges over polynomials) and

$$[\![L_{p,b}]\!]_U(A_1, \dots, A_n) = \{d \in \mathcal{D}(U) \mid p(d(A_1), \dots, d(A_n)) > b\}$$
$$[\![M_{p,b}]\!]_U(A_1, \dots, A_n) = \{d \in \mathcal{D}(U) \mid p(d(\overline{A_1}), \dots, d(\overline{A_n})) \leq b\}$$

for sets U and $A_1, \dots, A_n \subseteq U$. This logic presumably does not encode into the probabilistic μ-calculus as in 3 above, and can express constraints on independent products of events (see also [25]). E.g. the formula $\nu Y. L_{X_1 X_2, 0.9}(p \wedge Y, q \wedge Y)$ says roughly that two independently sampled successors of the current state will satisfy p and q, respectively, and then satisfy the same property again, with probability at least 0.9.

(The modalities in the last two items are inevitably less general than in the corresponding next-step logics [7,23], due to the need to ensure monotonicity.)

3 Tracking Automata

We use *parity automata* (e.g. [13]) that track single formulas along paths through potential models to decide whether it is possible to construct a model in which all least fixpoint formulas are eventually satisfied. Formally, (nondeterministic) parity automata are tuples $\mathsf{A} = (V, \Sigma, \Delta, q_0, \alpha)$ where V is a set of *nodes*; Σ is a finite set, the *alphabet*; $\Delta \subseteq V \times \Sigma \times V$ is the *transition relation* assigning a set $\Delta(v, a) = \{u \mid (v, a, u) \in \Delta\}$ of nodes to all $v \in V$ and $a \in \Sigma$; $q_0 \in V$ is the *initial node*; and $\alpha : \Delta \to \mathbb{N}$ is the *priority function*, assigning priorities $\alpha(v, a, u) \in \mathbb{N}$ to *transitions* $(v, a, u) \in \Delta$ (this is the standard in recent work since it yields slightly more succinct automata). If Δ is a (partial) functional relation, then A is said to be *deterministic*, and we denote the corresponding partial function by $\delta : V \times \Sigma \rightharpoonup V$. The automaton A *accepts* an infinite word $w = w_0, w_1, \dots \in \Sigma^\omega$ if there is a w-path through A on which the highest priority that is passed infinitely often is even; formally, the language that is accepted by A is defined by $L(\mathsf{A}) = \{w \in \Sigma^\omega \mid \exists \rho \in \mathsf{run}(\mathsf{A}, w).\ \max(\mathsf{Inf}(\alpha \circ \rho))$ is even$\}$, where $\mathsf{run}(\mathsf{A}, w)$ denotes the set of infinite sequences $(\rho_0, w_0, \rho_1), (\rho_1, w_1, \rho_2), \dots \in \Delta^\omega$ such that $\rho_0 = q_0$ and where, given an infinite sequence S, $\mathsf{Inf}(S)$ denotes the elements that occur infinitely often in S. Here, we see infinite sequences $\rho \in U^\omega$ over some set U as functions $\mathbb{N} \to U$ and write ρ_i to denote the i-th element of ρ.

We now *fix a target formula* χ and put $n_0 = |\chi|$, $n_1 = \mathsf{size}(\chi)$. We let $\mathbf{F}$ denote the *Fischer-Ladner closure* [20] of χ; i.e. $\mathbf{F}$ contains all formulas that can arise as subformulas when unfolding each fixpoint in χ exactly once. We put $k = \max\{\mathsf{ad}(\psi) \mid \psi \in \mathbf{F}\}$ and $\mathsf{selections} = \mathcal{P}(\mathbf{F} \cap \Lambda(\mathbf{F}))$ ($\mathbf{F} \cap \Lambda(\mathbf{F})$ is the set of modal literals in $\mathbf{F}$). We have $|\mathbf{F}| \leq n$ and hence $|\mathsf{selections}| \leq 2^n$.

Definition 2 (Tracking automaton). The *tracking automaton* for χ is a nondeterministic parity automaton $\mathsf{A}_\chi = (\mathbf{F}, \Sigma, \Delta, q_0, \alpha)$, where $q_0 = \chi$,

$$\begin{aligned}\Sigma = &\{(\psi_0 \vee \psi_1, b) \in \mathbf{F} \times \{0,1\}\} \cup \{(\psi_0 \wedge \psi_1, 0) \in \mathbf{F} \times \{0\}\} \cup \\ &\{(\eta X.\, \psi_1, 0) \in \mathbf{F} \times \{0\}\} \cup \ \mathsf{selections}\ ,\end{aligned}$$

and for $\psi, \psi_0, \psi_1 \in \mathbf{F}$, $\kappa \in \mathsf{selections}$ and $b \in \{0,1\}$,

$$\begin{aligned}\Delta(\psi, \kappa) &= \{\psi_0 \in \mathbf{F} \mid \psi \in \kappa \cap \Lambda(\{\psi_0\})\} \\ \Delta(\psi, (\psi_0 \vee \psi_1, b)) &= \{\psi_b \mid \psi = \psi_0 \vee \psi_1\} \cup \{\psi \mid \psi \neq \psi_0 \vee \psi_1\} \\ \Delta(\psi, (\psi_0 \wedge \psi_1, 0)) &= \{\psi_0, \psi_1 \mid \psi = \psi_0 \wedge \psi_1\} \cup \{\psi \mid \psi \neq \psi_0 \wedge \psi_1\} \\ \Delta(\psi, (\eta X.\, \psi_1, 0)) &= \{\psi_1[X \mapsto \psi] \mid \psi = \eta X.\, \psi_1\} \cup \{\psi \mid \psi \neq \eta X.\, \psi_1\}\end{aligned}$$

E.g. the last clause means that when tracking the unfolding of a fixpoint $\eta X.\, \psi_1$ at ψ, we track ψ to the unfolding $\psi_1[X \mapsto \psi]$ if ψ equals the unfolded fixpoint, and to ψ otherwise; similarly for the other clauses, and in particular a modal literal $\psi = \heartsuit\psi_0$ is only tracked to ψ_0 through a selection κ if $\heartsuit\psi_0 \in \kappa$, i.e. if κ selects $\heartsuit\psi_0$ to be tracked. The priority function α is derived from the alternation depths of formulas, counting only unfoldings of fixpoints (i.e. all other transitions have priority 1). Formally, $\alpha(\psi, \sigma, \psi') = 1$ if $\psi = \psi'$ or ψ is not a fixpoint literal; if ψ is a fixpoint literal and $\psi \neq \psi'$, then we put $\alpha(\psi, \sigma, \psi') = \mathsf{ad}(\psi)$.

Intuitively, words from Σ^ω encode infinite paths through coalgebras (C, ξ) in which states $x \in C$ are labelled with sets $l(x)$ of formulas, where letters $\kappa \in \mathsf{selections}$ encode modal steps from states $x \in C$ with label $l(x)$ to states $y \in C$ with label $\{\psi \mid \heartsuit\psi \in \kappa \cap l(x)\}$. The automaton is built to accept $L(\mathsf{A}_\chi) = \mathsf{BadBranch}_\chi$ where $\mathsf{BadBranch}_\chi$ is the set of words that encode a path on which a least fixpoint formula ψ is unfolded infinitely often without being dominated by any outer fixpoint formula (i.e. one with alternation depth greater than $\mathsf{ad}(\psi)$). Letters $(\psi_0 \vee \psi_1, b)$ choose disjuncts according to b, while for letters $(\psi_0 \wedge \psi_1, 0)$, the tracking automaton is nondeterministic, reflecting the fact that bad fixpoints can reside in either ψ_0 or ψ_1. The automaton A_χ has size n_0 and priorities 1 to k. Using a standard construction (e.g. [18]), we transform A_χ into an equivalent Büchi automaton of size $n_0 k$. Then we determinize the Büchi automaton using, e.g., the Safra/Piterman-construction [30,32] and obtain an equivalent deterministic parity automaton with priorities 0 to $2n_0k - 1$ and size $\mathcal{O}(((n_0k)!)^2)$. Finally we complement this parity automaton by increasing every priority by 1, obtaining a deterministic parity automaton $\mathsf{B}_\chi = (D_\chi, \Sigma, \delta, v_0, \beta)$ of size $\mathcal{O}(((n_0k)!)^2)$, with priorities 1 to $2n_0k$ and with

$$L(\mathsf{B}_\chi) = \overline{L(\mathsf{A}_\chi)} = \overline{\mathsf{BadBranch}_\chi} =: \mathsf{GoodBranch}_\chi,$$

i.e. B_χ is a deterministic parity automaton that accepts the words that encode paths along which satisfaction of least fixpoints is never deferred indefinitely. We define a labelling function $l : D_\chi \to \mathcal{P}(\mathbf{F})$ mapping each state of B_χ (e.g. a Safra tree) to the set of formulas occurring in it.

Remark 3. It has been noted that the standard tracking automata for *alternation-free* formulas are, in fact, Co-Büchi automata [10,16] and that the tracking automata for *aconjunctive* formulas are *limit-deterministic* parity automata [15]. These considerably simpler automata can be determinized to deterministic Büchi automata of size 3^{n_0} and to deterministic parity automata of size $\mathcal{O}((n_0k)!)$ and with $2n_0k$ priorities, respectively. This observation also holds true for the tracking automata in this work so that for formulas of suitable syntactic shape, Lemma 11 below yields accordingly lower bounds on the runtime of our satisfiability checking algorithm.

4 Global Caching for the Coalgebraic μ-Calculus

We now introduce a generic global caching algorithm for satisfiability in the coalgebraic μ-calculus. Given an input formula χ, the algorithm expands the determinized and complemented tracking automaton B_χ step by step and propagates (un)satisfiability through this graph; the algorithm terminates as soon as the initial node v_0 is marked as (un)satisfiable. The algorithm bears similarity to standard game-based algorithms for μ-calculi [8,9,15]; however, it crucially deviates from these algorithms in the treatment of modal steps: Intuitively, our algorithm decides whether it is possible to remove some of the modal transitions as well as one of the transitions from each reachable pair $((\psi_0 \vee \psi_1), 0), ((\psi_0 \vee \psi_1), 1)$ of disjunction transitions within the automaton B_χ in such a way that the resulting sub-automaton of B_χ is totally accepting, that is, accepts any word for which there is an infinite run. In doing so, it is crucial that the labels of state nodes v in the reduced automaton are *one-step satisfiable*, in a sense introduced next, in the set of states that are reachable from v by the remaining modal transitions. Propagating (un)satisfiability over modal transitions thus involves *one-step satisfiability checking*, a functor-specific problem that in many instances can be solved in time singly exponential in $\mathsf{size}(\chi)$. In previous work [8], a variant of one-step satisfiability has been used in satisfiability games for coalgebraic μ-calculi, which however leads to a doubly exponential number of modal moves for one of the players and hence does not yield a singly exponential upper bound on satisfiability checking (unless a suitable set of tableau rules is provided).

Definition 4 (One-step satisfiability problem [26,33,35]). Let V be a finite set, let $v \subseteq \Lambda(V)$ such that $a \neq b$ whenever $\heartsuit_1 a, \heartsuit_2 b \in v$, and let $U \subseteq \mathcal{P}(V)$. The *one-step satisfiability problem* for inputs v and U is to decide whether $TU \cap [\![v]\!]_1 \neq \emptyset$, where

$$[\![v]\!]_1 = \bigcap\nolimits_{\heartsuit a \in v} [\![\heartsuit]\!] \{u \in U \mid a \in u\}.$$

We put $\mathsf{size}(v) = \sum_{\heartsuit a \in v} \mathsf{size}(\heartsuit)$, and denote the time it takes to solve the problem on v, U with $\mathsf{size}(v) = a$ and $|V| = b$ (hence $|U| \leq 2^b$) by $t(a, b)$.

Remark 5. We keep the definition of the actual one-step logic as mentioned in the introduction somewhat implicit in the above definition of the one-step satisfiability problem. One can see that it contains two layers: a purely propositional layer embodied in U, which postulates which propositional formulas over V are satisfiable; and a modal layer with nesting depth of modalities uniformly equal to 1, embodied in the set v, which specifies constraints on an element of TU.

Example 6. For the standard modal μ-calculus (Example 1.1), the one-step satisfiability problem is to decide for given $v \subseteq \Lambda(V)$ and $U \subseteq \mathcal{P}(V)$ whether there is $A \in \mathcal{P}(U) \cap [\![v]\!]_1$, that is, a subset $A \subseteq U$ such that for each $\Diamond a \in v$, there is $u \in A$ such that $a \in u$, and for each $\Box a \in v$ and each $u \in A$, $a \in u$. Here we have $t(a, b) \leq a \cdot 2^b$ where $a = \mathsf{size}(v)$, $b = |V|$. For the graded μ-calculus (Example 1.2), the one-step satisfiability problem is to decide for v, U as above whether there is a multiset $\theta \in \mathcal{B}(U)$ such that $\sum_{u \in U \mid a \in u} \theta(u) > m$ for each $\langle m \rangle a \in v$ and $\sum_{u \in U \mid a \notin u} \theta(u) \leq m$ for each $[m]a \in v$.

Definition 7 (States and Prestates). A node v of B_χ is a *state* if its label contains only modal literals ($l(v) \subseteq \Lambda(\mathbf{F})$), and otherwise a *prestate*, in which case we fix $\psi_v \in l(v) \setminus \Lambda(\mathbf{F})$. We write $\mathsf{states}, \mathsf{prestates} \subseteq D_\chi$ for the sets of states and prestates, respectively.

We next define $2n_0k$-ary set functions f and g that compute one-step (un)satisfiability w.r.t. their argument sets.

Definition 8 (One-step propagation). For sets $G \subseteq D_\chi$ and $\mathbf{X} = X_1, \ldots, X_{2n_0k} \in \mathcal{P}(G)^{2n_0k}$, we put

$$\begin{aligned} f(\mathbf{X}) = &\{v \in \mathsf{prestates} \mid \exists b \in \{0,1\}.\, \delta(v, (\psi_v, b)) \in X_{\beta(v,(\psi_v,b))}\} \cup \\ &\{v \in \mathsf{states} \mid T(\textstyle\bigcup_{1 \leq i \leq 2n_0k} X_i(v)) \cap [\![l(v)]\!]_1 \neq \emptyset\} \\ g(\mathbf{X}) = &\{v \in \mathsf{prestates} \mid \forall b \in \{0,1\}.\, \delta(v, (\psi_v, b)) \in X_{\beta(v,(\psi_v,b))}\} \cup \\ &\{v \in \mathsf{states} \mid T(\textstyle\bigcup_{1 \leq i \leq 2n_0k} \overline{X_i}(v)) \cap [\![l(v)]\!]_1 = \emptyset\}, \end{aligned}$$

where $\beta(v, (\psi_v, b))$ abbreviates $\beta(v, (\psi_v, b), \delta(v, (\psi_v, b)))$ and where

$$X_i(v) = \{l(u) \mid u \in X_i, \exists \kappa \in \mathsf{selections}.\, \delta(v, \kappa) = u, \beta(v, \kappa, u) = i\}.$$

Since for states v, $l(v) \subseteq \Lambda(\mathbf{F})$ and $X_i(v) \subseteq \mathcal{P}(\mathbf{F})$ for all i, one-step propagation steps for states are instances of the one-step satisfiability problem with $|V| = |\mathbf{F}|$, solvable in time $t(n_1, n_0)$ because $\mathsf{size}(l(v)) \leq n_1$ and $|\mathbf{F}| \leq n_0$.

Definition 9 (Propagation). Given a set G, we put

$$\begin{aligned} \mathbf{E}_G &= \eta_{2n_0k} X_{2n_0k}. \ldots \eta_2 X_2.\eta_1 X_1.f(\mathbf{X}) \\ \mathbf{A}_G &= \overline{\eta_{2n_0k}} X_{2n_0k} \ldots \overline{\eta_2} X_2.\overline{\eta_1} X_1.g(\mathbf{X}), \end{aligned}$$

where $\mathbf{X} = X_1, \ldots, X_{2n_0k}$ for $X_i \subseteq G$, where $\eta_i = \mu$ for odd i, $\eta_i = \nu$ for even i and where $\overline{\nu} = \mu$ and $\overline{\mu} = \nu$.

The set $\mathbf{E}_G$ contains nodes $v \in G$ for which there are choices for all disjunctions and modal transitions that are reachable from v within G (as indicated at the beginning of the section) such that the labels of all reachable states in the chosen sub-automaton of B_χ are one-step satisfiable and such that on all paths through the chosen sub-automaton, the highest priority that is passed infinitely often is even, the intuition being that no least fixpoint is unfolded infinitely often without being dominated. Dually, the set $\mathbf{A}_G$ contains nodes for which there exist no such suitable choices.

We recall that $v_0 \in D_\chi$ is the initial state of the determinized and complemented tracking automaton B_χ. The algorithm expands B_χ step-by-step starting from v_0; for prestates u, the expansion step adds nodes according to the fixed non-modal formula ψ_u that is to be expanded next (Definition 7), and for states, the expansion follows all (matching) selections. The order of expansion can be chosen freely, e.g. by heuristic methods. Optional intermediate propagation steps can be used judiciously to realize on-the-fly solving.

Algorithm 10 (Global caching). To decide the satisfiability of the input formula χ, initialize the sets of *unexpanded* and *expanded* nodes, $U = \{v_0\}$ and $G = \emptyset$, respectively.

1. Expansion: Choose some unexpanded node $u \in U$, remove u from U, and add u to G. If u is a prestate, then add the set $\{\delta(u,\sigma) \mid \sigma \in \Sigma \cap (\psi_u \times \{0,1\})\}$ to U. If u is a state, then add the set $\{\delta(u,\kappa) \mid \kappa \in \mathsf{selections}\}$ to U.
2. Optional propagation: Compute $\mathbf{E}_G$ and/or $\mathbf{A}_G$. If $v_0 \in \mathbf{E}_G$, then return 'satisfiable', if $v_0 \in \mathbf{A}_G$, then return 'unsatisfiable'.
3. If $U \neq \emptyset$, then continue with step 1.
4. Final propagation: Compute $\mathbf{E}_G$. If $v_0 \in \mathbf{E}_G$, then return 'satisfiable', otherwise return 'unsatisfiable'.

Lemma 11. *Algorithm 10 runs in time* $\mathcal{O}(((n_0k)!)^{4n_0k} \cdot t(n_1, n_0))$.

Proof. The loop of the algorithm expands the determinized and complemented tracking automaton node by node and hence is executed at most $|D_\chi| \in \mathcal{O}(((n_0k)!)^2) \subseteq 2^{\mathcal{O}(n_0k \log n_0)}$ times. A single expansion step can be implemented in time $\mathcal{O}(2^{n_0})$ since propositional expansion is unproblematic and for the modal expansion of a state u, all (matching) selections, of which there are (at most) 2^{n_0}, have to be considered. A single propagation step consists in computing two fixpoints of nesting depth $2n_0k$ of the functions f and g over $\mathcal{P}(D_\chi)^{2n_0k}$ and can hence be implemented in time $2(|D_\chi|^{2n_0k} \cdot t(n_1, n_0)) \in \mathcal{O}(((n_0k!)^2)^{2n_0k} \cdot t(n_1, n_0)) \subseteq 2^{\mathcal{O}(n_0^2k^2 \log n_0 + \log(t(n_1,n_0)))}$, noting that a single computation of $f(\mathbf{X})$ and $g(\mathbf{X})$ for a tuple $\mathbf{X} \in \mathcal{P}(D_\chi)^{2n_0k}$ can be implemented in time $\mathcal{O}(t(n_1, n_0))$ – this has been noted above for states, and prestates are unproblematic. Thus the complexity of the whole algorithm is dominated by the complexity of the propagation step. ☐

Corollary 12. *If the one-step satisfiability problem of a coalgebraic logic can be solved in time $t(a, b)$ exponential in $a + b$ on inputs $v \subseteq \Lambda(V)$, $U \subseteq \mathcal{P}(V)$*

with $\mathsf{size}(v) = a$, $|V| = b$, *then the satisfiability problem of the corresponding coalgebraic* μ*-calculus is in* ExpTime.

Since the existence of a tractable set of tableau rules implies the required time bound on one-step satisfiability, the above result subsumes earlier bounds obtained by tableau-based approaches in [4,15,16]; however, it covers additional example logics for which no suitable tableau rules are known. In particular we have

Proposition 13. *The satisfiability problems of the following logics are in* ExpTime*:*

1. *the standard* μ*-calculus,*
2. *the graded* μ*-calculus,*
3. *the (two-valued) probabilistic* μ*-calculus,*
4. *the graded* μ*-calculus with polynomial inequalities,*
5. *the (two-valued) probabilistic* μ*-calculus with polynomial inequalities.*

(Tractable sets of tableau rules have previously been claimed for the graded [36] and Presburger [22] μ-calculus but have since been discovered to be flawed [23].)

Proof. It suffices to show that the respective one-step satisfiability problems can be solved on inputs $v \subseteq \Lambda(V)$, $U \subseteq \mathcal{P}(V)$ with $\mathsf{size}(v) = a$ and $|V| = b$ in singly exponential time in $a + b$, i.e. in time $t(a, b) \in 2^{p(a+b)}$ for p at most polynomial. E.g. for standard relational modalities, we have $t(a, b) = a \cdot 2^b = 2^{b+\log a}$, see Example 6. While the bounds can be established by relatively easy arguments (e.g. using known bounds on sizes of solutions of systems of real or integer linear inequalities) for all of our remaining example logics, we import them from previous work for brevity. For the one-step satisfiability problem of graded modal logic, by [21, Lemma 1], we have $t(a, b) \le (2 \cdot 2^a + 2)^b \le 2^{ab+2b}$; the Lemma uses counters to check joint one-step satisfiability of constraints and directly extends to the one-step satisfiability problem of graded modal logic with monotone polynomial inequalities, in which case we require n counters for each n-ary polynomial. The bound for (two-valued) probabilistic modal logic (with polynomial inequalities) is shown in [23, Example 7]. ◻

Remark 14. We also obtain a polynomial bound on branching width in models for all our example logics simply by importing Lemma 6 and the observations in Example 7 from [23]. With the exception of the standard μ-calculus, this bound appears to be new in all our example logics. Of course, for graded and Presburger μ-calculi, polynomial branching holds only in their coalgebraic semantics, i.e. over multigraph models but not over Kripke models.

5 Soundness and Completeness

We now prove the central result, that is, the soundness and completeness of Algorithm 10. As the sets $\mathbf{E}_G$ and $\mathbf{A}_G$ grow monotonically with G, it suffices to prove equivalence of satisfiability and containment of the initial node v_0 in $\mathbf{E} := \mathbf{E}_{D_\chi}$. Our program is as follows: We show that $v_0 \in \mathbf{E}$ if and only if there is a *pre-semi-tableau* (Definition 15) for χ with *unfolding timeouts* (Definition 17), which in turn is the case if and only if χ is satisfiable. We establish the latter equivalence by constructing a model for χ from a given pre-semi-tableau with unfolding timeouts and, for the converse direction, extracting a pre-semi-tableau with unfolding timeouts from the model.

Definition 15 (Pre-semi-tableau). Given a ternary relation $R \subseteq A \times B \times A$ and $a \in A$, $b \in B$, we generally write $R(a) = \{a' \in A \mid \exists b \in B.\, (a, b, a') \in R\}$ and $R(a, b) = \{a' \in A \mid (a, b, a') \in R\}$. Let $W \subseteq D_\chi$ and put $U = W \cap \mathsf{prestates}$ and $V = W \cap \mathsf{states}$. Given a ternary relation $L \subseteq W \times \Sigma \times W$, the pair (W, L) is a *pre-semi-tableau* for χ if the following conditions hold: $L \subseteq \delta$; $T(L(v)) \cap [\![l(v)]\!]_1 \neq \emptyset$ for all $v \in V$; for each $u \in U$, there is exactly one $b \in \{0, 1\}$ such that $L(u, (\psi_u, b)) = \{\delta(u, (\psi_u, b))\}$, and for all other $\sigma \in \Sigma$, $L(u, \sigma) = \emptyset$; and there is no L-cycle that contains only elements from U. A *path* through a pre-semi-tableau is an infinite sequence $(v_0, \sigma_0), (v_1, \sigma_1), \ldots \in (W \times \Sigma)^\omega$ such that for all i, $v_{i+1} \in L(v_i, \sigma_i)$. We denote *the* first state that is reachable by zero or more L-steps from a node $v \in W$ by $\lceil v \rceil$ (since there is no L-cycle within U, such a state always exists).

Given a state v, the relation L of a pre-semi-tableau thus picks a set $L(v)$ of nodes in which $l(v)$ is one-step satisfiable; given a prestate u, L picks a single (pre)state that is obtained from u by transforming the formula ψ_u.

Definition 16 (Tracking timeouts). Given a path $\rho = (v_0, \sigma_0), (v_1, \sigma_1), \ldots$ through a pre-semi-tableau, we say that priority i *occurs* (at position j) in ρ if $\beta(v_j, \sigma_j, v_{j+1}) = i$, recalling that β is the priority function of the determinised and complemented tracking automaton B_χ. Then the path ρ has *tracking timeouts* $\overline{m} = (m_{2n_0k}, \ldots, m_1)$ if for each odd $1 \leq i < 2n_0k$, priority i occurs at most m_i times in ρ before some priority greater than i occurs in ρ. Nothing is said about the m_i for even i, which are in fact irrelevant and serve only to ease notation. A node $w \in W$ in a pre-semi-tableau (W, L) has *tracking timeouts* $\overline{m}$ if every path through (W, L) starting at w has tracking timeouts $\overline{m}$. A pre-semi-tableau (W, L) *has tracking timeouts* if each $w \in W$ has tracking timeouts $\overline{m}$ for some $\overline{m}$.

Intuitively, a pre-semi-tableau (W, L) has tracking timeouts if every word that encodes an infinite L-path through W is accepted by B_χ. The next definition is geared towards characterizing non-acceptance by A_χ:

Definition 17 (Traces and unfolding timeouts). Let (W, L) be a graph with $L \subseteq W \times \Sigma \times W$ and labeling function $l : W \to \mathcal{P}(\mathbf{F})$. Given an L-path $\rho = (v_0, \sigma_0), (v_1, \sigma_1), \ldots$ (with $(v_i, \sigma_i, v_{i+1}) \in L$ for $i \geq 0$) and a sequence of formulas $\Psi = \psi_0, \psi_1, \ldots$, we say that Ψ is a *trace* of ψ_0 along ρ (we also say that ρ *contains* the trace Ψ) if $\psi_i \in l(v_i)$ and $\psi_{i+1} \in \Delta(\psi_i, \sigma_i)$ for all i. For i with $\psi_i = \eta X.\psi$ for some fixpoint variable X and some formula ψ, we say that Ψ *unfolds at level* $\mathsf{ad}(\psi_i)$ at position i. Then the trace Ψ has *unfolding* *timeout* $m \in \mathbb{N}$ for ψ_0 at level j if Ψ unfolds at most m times at level j before Ψ unfolds at some level greater than j. The path ρ has *unfolding timeouts* for ψ_0 at level j if there is, for all its traces Ψ of ψ_0, some m such that Ψ has unfolding timeout m for ψ_0 at level j. A node $w \in W$ has *unfolding timeouts* at level j for some formula ψ if every path through (W, L) that starts at w and that contains infinitely many steps (v_i, σ_i) such that $\sigma_i \in \mathsf{selections}$ has unfolding timeouts for ψ at level i. (Since fixpoint variables are by assumption guarded by modal operators, it suffices to require timeouts just for such paths that contain infinitely many modal steps.) A node $w \in W$ has *unfolding timeouts* $\overline{m} = (m_k, \ldots, m_1)$ for some formula ψ if every path through (W, L) that starts at w and that contains infinitely many steps (v_i, σ_i) such that $\sigma_i \in \mathsf{selections}$ has, for each odd $1 \leq i \leq k$, unfolding timeouts $\overline{m}$ for ψ at level i; again the unfolding timeouts for even i, that is, for greatest fixpoints, are irrelevant. The graph (W, L) has *unfolding timeouts* if for each element $w \in W$ and each formula $\psi \in l(v)$, there is some vector $\overline{m}$ such that w has unfolding timeouts $\overline{m}$ for ψ. We denote the set of nodes that have unfolding timeouts $\overline{m}$ for ψ by $\mathsf{uto}(\psi, \overline{m}) \subseteq W$.

A graph (W, L) has unfolding timeouts if for all words that encode an infinite L-path through (W, L), all runs of the nondeterministic tracking automaton A_χ on the word are *non*-accepting. We recall that a run of A_χ is accepting if it unfolds some least fixpoint infinitely often without having it dominated.

Lemma 18. *Let (W, L) be a pre-semi-tableau. Then (W, L) has tracking timeouts if and only if it has unfolding timeouts.*

Proof. We recall that B_χ is obtained from A_χ by determinization and subsequent complementation so that we have $L(\mathsf{B}_\chi) = \overline{L(\mathsf{A}_\chi)}$. The result thus follows directly from the fact that having tracking timeouts means that B_χ accepts all words that encode a path in (W, L) while having unfolding timeouts means that A_χ does not accept any word that encodes a path in (W, L). □

Lemma 19. *We have $v_0 \in \mathbf{E}$ if and only if there is a pre-semi-tableau for χ that has tracking timeouts.*

Combining Lemmas 19 and 18, we obtain

Corollary 20. *We have $v_0 \in \mathbf{E}$ if and only if there is a pre-semi-tableau for χ that has unfolding timeouts.*

We now show that satisfiability of χ and the existence of a semi-pre-tableau for χ with unfolding timeouts coincide.

Definition 21. Given a pre-semi-tableau (W, L) with set of states V, we put

$$\widehat{[\![\psi]\!]} = \{v \in V \mid l(v) \vdash_{\mathsf{PL}} \psi\} \qquad \widehat{[\![\psi]\!]}_{\overline{m}} = \widehat{[\![\psi]\!]} \cap \{\lceil u \rceil \in V \mid u \in \mathsf{uto}(\psi, \overline{m})\}$$

where $\psi \in \mathbf{F}$, where $\vdash_{\mathsf{PL}}$ denotes propositional entailment and where $\overline{m}$ is a vector of k natural numbers.

Thus we have $v \in [\![\psi]\!]_{\overline{m}}$ if there is a node $u \in W$ such that $\lceil u \rceil = v$ and u has timeouts $\overline{m}$ for ψ. This serves to ease the proofs of the upcoming existence and truth lemmas as it anchors the timeout vector $\overline{m}$ at the node u instead of anchoring it at the state v which may not have timeouts $\overline{m}$ for ψ (namely, if a greatest fixpoint is unfolded on the L-path from u to v).

Definition 22 (Strong coherence). Let (W, L) be a pre-semi-tableau with set V of states. A coalgebra $\mathcal{C} = (V, \xi)$ is *strongly coherent* if for all states $v \in V$, for all formulas $\heartsuit\psi \in \mathbf{F}$ and for all timeout-vectors $\overline{m}$,

$$v \in \widehat{[\![\heartsuit\psi]\!]_{\overline{m}}} \text{ implies } \xi(v) \in [\![\heartsuit]\!](\widehat{[\![\psi]\!]_{\overline{m}}}).$$

Strongly coherent coalgebras exist over pre-semi-tableaux:

Lemma 23 (Existence). *Let (W, L) be a pre-semi-tableau with set of states V. Then there is a strongly coherent coalgebra over V.*

Since all least fixpoint literals are satisfied after finitely many unfolding steps in strongly coherent coalgebras with unfolding timeouts, they are models, i.e. satisfy all the formulas in their labels:

Lemma 24 (Truth). *In strongly coherent coalgebras that have unfolding timeouts, we have that for all $\psi \in \mathbf{F}$,*

$$\widehat{[\![\psi]\!]} \subseteq [\![\psi]\!].$$

Definition 25 (Timed-out satisfaction). Given sets $U \subseteq W$, a function $f : \mathcal{P}(W) \to \mathcal{P}(W)$ and an ordinal number λ, we define $f^\lambda(U) = U$ if $\lambda = 0$, $f^\lambda(U) = f(f^{\lambda'}(U))$ if $\lambda = \lambda' + 1$ and $f^\lambda(U) = \bigcup_{k<\lambda} f^k(U)$ if λ is a limit-ordinal. The target formula χ is clean so that it contains, for each fixpoint variable $X \in \mathbf{V}$, at most a single fixpoint literal $\eta X.\psi_0$ as a subformula; we denote this formula by $\theta(X)$. Given a coalgebra (C, ξ), a formula ψ and a vector $\overline{\lambda} = (\lambda_k, \dots, \lambda_j)$ of ordinal numbers, we define $[\![\psi]\!]^{\overline{\lambda}} = [\![\psi]\!]_i$ where $i : \mathbf{V} \nrightarrow \mathcal{P}(C)$ is defined, for fixpoint variables X_j that occur freely in ψ and for which we have $\theta(X_j) = \eta X_j \psi_j$, by $i(X_j) = ([\![\psi_j]\!]_{i'}^{X_j})^{\lambda_j}(\emptyset)$ if $\eta = \mu$ and by $i(X_j) = [\![\nu X_j.\psi_j]\!]_{i'}$ if $\eta = \nu$, where $i'(X_{j'})$ is undefined for $j' \geq j$ and where $i'(X_{j'}) = i(X_{j'})$ for $j' < j$. Again the timeouts for greatest fixpoint variables are irrelevant and serve only to ease notation.

Definition 26 (Strongly supporting Kripke frame). Let (C, ξ) be a coalgebra. For states $x \in C$ and formulas ψ such that $x \in [\![\psi]\!]$, let $\overline{\lambda}_\psi$ denote the least vector of ordinal numbers such that $x \in [\![\psi]\!]^{\overline{\lambda}_\psi}$. Also let, for $\psi \in \mathbf{F}$, $\overline{\psi}$ be *the* subformula of χ such that ψ is obtained from $\overline{\psi}$ by repeatedly replacing free variables X by $\theta(X)$. A graph (C, L) with $L \subseteq C \times \Sigma \times C$ and with labeling function $l : C \to \mathcal{P}(\mathbf{F})$ such that $l(x) = \{\psi \in \mathbf{F} \mid x \in [\![\psi]\!]\}$ is a *strongly supporting Kripke frame* (for C, ξ) if

– for all $\psi \in \mathbf{F}$ and $x \in C$, if $x \notin [\![\psi]\!]$, then $L(x,(\psi,b)) = \emptyset$ for $b \in \{0,1\}$; if $x \in [\![\psi]\!]$, then we distinguish upon the shape of ψ: if $\psi = \psi_0 \vee \psi_1$, then we require $L(x,(\psi,b)) = \{x\}$ for exactly one $b \in \{0,1\}$ with $x \in [\![\overline{\psi_b}]\!]^{\overline{\lambda}_{\overline{\psi}}}$ and $L(x,(\psi,\bar{b})) = \emptyset$, where $\bar{1} = 0$, $\bar{0} = 1$; if $\psi = \psi_0 \wedge \psi_1$ or $\psi = \eta X.\psi_0$, then we require $L(x,(\psi,0)) = \{x\}$.
– for all $x \in C$ and $\kappa \in \mathsf{selections}$, we have $L(x,\kappa) = \{y\}$ for *some* $y \in A = \bigcap_{\heartsuit\psi\in\kappa}[\![\overline{\psi}^{\overline{\lambda}_{\heartsuit\psi}}]\!]$ if $A \neq \emptyset$, and $L(x,\kappa) = \emptyset$ otherwise.

Lemma 27. *Every coalgebra has a strongly supporting Kripke frame.*

Definition 28. Given a coalgebra (C,ξ) with strongly supporting Kripke frame (C,L), a formula ψ and a valuation $i : \mathbf{V} \nrightarrow \mathcal{P}(C)$, we define $[\![\psi]\!]_i^L$ by the same clauses as $[\![\psi]\!]_i$ in all cases except the following:

$$
\begin{aligned}
[\![\psi_0 \vee \psi_1]\!]_i^L =& \{x \in C \mid\ x \in [\![\psi_b]\!]_i^L, b \in \{0,1\}, L(x,(\phi_0 \vee \phi_1, b)) = \{x\}\} \\
[\![\heartsuit\psi_0]\!]_i^L =& \{x \in C \mid\ (Tg_x)(\xi(x)) \in [\![\heartsuit]\!](g_x[[\![\psi_0]\!]_i^L])\} \\
[\![\mu X.\psi_0]\!]_i^L =& \{x \in C \mid\ x \text{ has unfolding timeouts at level } \mathsf{ad}(\mu X.\phi_0) \\
& \qquad\qquad \text{for } \mu X.\phi_0 \text{ in } (C,L)\},
\end{aligned}
$$

where $\mu X.\psi_0 = \overline{\mu X.\phi_0}$ and $\psi_0 \vee \psi_1 = \overline{\phi_0 \vee \phi_1}$, and where $g_x : C \to \{y_\kappa \mid L(x,\kappa) = \{y_\kappa\}\}$ is defined by $g_x(c) = y_\kappa$ if and only if $\kappa = \{\heartsuit\psi \in \mathbf{F} \mid c \in [\![\psi]\!]\}$.

Strongly supporting Kripke frames have unfolding timeouts:

Lemma 29. *For all coalgebras (C,ξ) with strongly supporting Kripke frame (C,L), all formulas ψ and all valuations $i : \mathbf{V} \nrightarrow \mathcal{P}(C)$, we have $[\![\psi]\!]_i \subseteq [\![\psi]\!]_i^L$.*

Lemma 30 (Soundness). *Let χ be satisfiable. Then a pre-semi-tableau for χ with unfolding timeouts can be constructed over a subset of D_χ.*

Proof (Sketch). By Lemmas 27 and 29, any model of χ has a strongly supporting Kripke frame (C,L) with unfolding timeouts. We derive a pre-semi-tableau for χ from (C,L), inheriting unfolding timeouts. □

Corollary 31 (Soundness and completeness). *We have*

$$v_0 \in \mathbf{E} \text{ if and only if } \chi \text{ is satisfiable.}$$

Our model construction moreover yields the same bound on minimum model size as in earlier work on the coalgebraic μ-calculus [4]:

Corollary 32 (Small model property). *Let χ be a satisfiable coalgebraic μ-calculus formula. Then χ has a model of size $\mathcal{O}(((nk)!)^2) \in 2^{\mathcal{O}(nk\log n)}$.*

6 Conclusion

We have shown that the satisfiability problem of the coalgebraic μ-calculus is in ExpTime, subject to establishing a suitable time bound on the much simpler one-step satisfiability problem. Prominent examples include the graded μ-calculus, the (two-valued) probabilistic μ-calculus, and extensions of the probabilistic and the graded μ-calculus, respectively, with (monotone) polynomial inequalities; the ExpTime bound appears to be new for the last two logics. We have also presented a generic satisfiability algorithm that realizes the time bound and supports global caching and on-the-fly solving. Moreover, we have obtained a polynomial bound on minimum branching width in models for all example logics mentioned above.

References

1. Alur, R., Henzinger, T., Kupferman, O.: Alternating-time temporal logic. J. ACM **49**, 672–713 (2002)
2. Bradfield, J., Stirling, C.: Modal μ-calculi. In: Handbook of Modal Logic, pp. 721–756. Elsevier (2006)
3. Bradfield, J., Walukiewicz, I.: The mu-calculus and model checking. In: Clarke, E., Henzinger, T., Veith, H., Bloem, R. (eds.) Handbook of Model Checking, pp. 871–919. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-10575-8_26
4. Cîrstea, C., Kupke, C., Pattinson, D.: EXPTIME tableaux for the coalgebraic μ-calculus. Log. Methods Comput. Sci. **7**, 1–33 (2011)
5. Cleaveland, R., Iyer, S., Narasimha, M.: Probabilistic temporal logics via the modal μ-calculus. Theor. Comput. Sci. **342**, 316–350 (2005)
6. D'Agostino, G., Visser, A.: Finality regained: a coalgebraic study of Scott-sets and multisets. Arch. Math. Logic **41**, 267–298 (2002)
7. Demri, S., Lugiez, D.: Presburger modal logic is PSPACE-complete. In: Furbach, U., Shankar, N. (eds.) IJCAR 2006. LNCS (LNAI), vol. 4130, pp. 541–556. Springer, Heidelberg (2006). https://doi.org/10.1007/11814771_44
8. Fontaine, G., Leal, R., Venema, Y.: Automata for coalgebras: an approach using predicate liftings. In: Abramsky, S., Gavoille, C., Kirchner, C., Meyer auf der Heide, F., Spirakis, P.G. (eds.) ICALP 2010, Part II. LNCS, vol. 6199, pp. 381–392. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14162-1_32
9. Friedmann, O., Lange, M.: Deciding the unguarded modal μ-calculus. J. Appl. Non-Classical Log. **23**, 353–371 (2013)
10. Friedmann, O., Latte, M., Lange, M.: Satisfiability games for branching-time logics. Log. Methods Comput. Sci. **9**, 1–36 (2013)
11. Goré, R., Nguyen, L.A.: Exptime tableaux for $\mathcal{ALC}$ using sound global caching. J. Autom. Reason. **50**, 355–381 (2013)
12. Goré, R., Widmann, F.: Sound global state caching for $\mathcal{ALC}$ with inverse roles. In: Giese, M., Waaler, A. (eds.) TABLEAUX 2009. LNCS (LNAI), vol. 5607, pp. 205–219. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02716-1_16
13. Grädel, E., Thomas, W., Wilke, T. (eds.): Automata Logics, and Infinite Games: A Guide to Current Research. LNCS, vol. 2500. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-36387-4

14. Hausmann, D., Schröder, L.: Global caching for the flat coalgebraic μ-calculus. In: Grandi, F., Lange, M., Lomuscio, A. (eds.) Temporal Representation and Reasoning, TIME 2015, pp. 121–143. IEEE (2015)
15. Hausmann, D., Schröder, L., Deifel, H.-P.: Permutation games for the weakly aconjunctive μ-calculus. In: Beyer, D., Huisman, M. (eds.) TACAS 2018, Part II. LNCS, vol. 10806, pp. 361–378. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-89963-3_21
16. Hausmann, D., Schröder, L., Egger, C.: Global caching for the alternation-free coalgebraic μ-calculus. In: Desharnais, J., Jagadeesan, R. (eds.) Concurrency Theory, CONCUR 2016. LIPIcs, vol. 59, pp. 34:1–34:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2016)
17. Huth, M., Kwiatkowska, M.: Quantitative analysis and model checking. In: Logic in Computer Science, LICS 1997, pp. 111–122. IEEE (1997)
18. King, V., Kupferman, O., Vardi, M.Y.: On the complexity of parity word automata. In: Honsell, F., Miculan, M. (eds.) FOSSACS 2001. LNCS, vol. 2030, pp. 276–286. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45315-6_18
19. Kozen, D.: Results on the propositional μ-calculus. Theor. Comput. Sci. **27**, 333–354 (1983)
20. Kozen, D.: A finite model theorem for the propositional μ-calculus. Stud. Log. **47**, 233–241 (1988)
21. Kupferman, O., Sattler, U., Vardi, M.Y.: The complexity of the graded μ-calculus. In: Voronkov, A. (ed.) CADE 2002. LNCS (LNAI), vol. 2392, pp. 423–437. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45620-1_34
22. Kupke, C., Pattinson, D.: On modal logics of linear inequalities. In: Beklemishev, L., Goranko, V., Shehtman, V. (eds.) Advances in Modal Logic, AiML 2010, pp. 235–255. College Publications (2010)
23. Kupke, C., Pattinson, D., Schröder, L.: Reasoning with global assumptions in arithmetic modal logics. In: Kosowski, A., Walukiewicz, I. (eds.) FCT 2015. LNCS, vol. 9210, pp. 367–380. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-22177-9_28
24. Liu, W., Song, L., Wang, J., Zhang, L.: A simple probabilistic extension of modal mu-calculus. In: Yang, Q., Wooldridge, M. (eds.) International Joint Conference on Artificial Intelligence, IJCAI 2015, pp. 882–888. AAAI Press (2015)
25. Mio, M.: Probabilistic modal μ-calculus with independent product. In: Hofmann, M. (ed.) FOSSACS 2011. LNCS, vol. 6604, pp. 290–304. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19805-2_20
26. Myers, R., Pattinson, D., Schröder, L.: Coalgebraic hybrid logic. In: de Alfaro, L. (ed.) FOSSACS 2009. LNCS, vol. 5504, pp. 137–151. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00596-1_11
27. Niwinski, D., Walukiewicz, I.: Games for the μ-calculus. Theor. Comput. Sci. **163**, 99–116 (1996)
28. Parikh, R.: The logic of games and its applications. Ann. Discret. Math. **24**, 111–140 (1985)
29. Pattinson, D.: Coalgebraic modal logic: soundness, completeness and decidability of local consequence. Theor. Comput. Sci. **309**, 177–193 (2003)
30. Piterman, N.: From nondeterministic Büchi and Streett automata to deterministic parity automata. Log. Methods Comput. Sci. **3**(3:5), 1–21 (2007)
31. Rutten, J.: Universal coalgebra: a theory of systems. Theor. Comput. Sci. **249**, 3–80 (2000)
32. Safra, S.: On the complexity of omega-automata. In: Foundations of Computer Science, FOCS 1988, pp. 319–327. IEEE Computer Society (1988)

33. Schröder, L.: A finite model construction for coalgebraic modal logic. J. Log. Algebr. Prog. **73**, 97–110 (2007)
34. Schröder, L.: Expressivity of coalgebraic modal logic: the limits and beyond. Theor. Comput. Sci. **390**(2–3), 230–247 (2008)
35. Schröder, L., Pattinson, D.: Shallow models for non-iterative modal logics. In: Dengel, A.R., Berns, K., Breuel, T.M., Bomarius, F., Roth-Berghofer, T.R. (eds.) KI 2008. LNCS (LNAI), vol. 5243, pp. 324–331. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85845-4_40
36. Schröder, L., Pattinson, D.: PSPACE bounds for rank-1 modal logics. ACM Trans. Comput. Log. **10**(2), 13:1–13:33 (2009)