Logical Methods in Computer Science
Volume 20, Issue 3, 2024, pp. 9:1–9:50
https://lmcs.episciences.org/

Submitted Dec. 22, 2022
Published Jul. 23, 2024

COALGEBRAIC SATISFIABILITY CHECKING FOR ARITHMETIC μ-CALCULI

DANIEL HAUSMANN [a] AND LUTZ SCHRÖDER [b]

[a] University of Liverpool, United Kingdom
e-mail address: d.hausmann@liverpool.co.uk

[b] Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
e-mail address: lutz.schroeder@fau.de

Abstract. The coalgebraic μ-calculus provides a generic semantic framework for fixpoint logics over systems whose branching type goes 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 relies on the availability of tableau rules for the next-step modalities that are sufficiently well-behaved in a formally defined sense; in particular, rule matches need to be representable by polynomial-sized codes, and the sequent duals of the rules need to absorb cut. While such rule sets have been identified for some important cases, they are not known to exist 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 algorithm that supports on-the-fly satisfiability checking. Notably, our approach directly accommodates unguarded formulae, and thus avoids use of the guardedness transformation. 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 [Koz83, BS07, BW18], which features explicit least and greatest fixpoint operators; roughly speaking, these serve to specify liveness properties (least fixpoints) and

Key words and phrases: μ-calculus, satisfiability checking, coalgebra, probabilistic transition systems, weighted transition systems, tableaux.

Work by the first author supported by the ERC Consolidator grant D-SynMA (No. 772459) and by the EPSRC through grant EP/Z003121/1. Work by the second author is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project no. 531706730.

DOI:10.46298/LMCS-20(3:9)2024

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 [CIN05, HK97, CKP11a, LSWZ15], the alternating-time μ-calculus [AHK02], and the monotone μ-calculus, which contains Parikh's game logic [Par85]. The graded μ-calculus [KSV02] features next-step modalities that count successors; it is standardly interpreted over Kripke frames but, as pointed out by D'Agostino and Visser [DV02], 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 formulae.

Coalgebraic logic [Pat03] 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 [Rut00]; additionally, modalities are interpreted as so-called *predicate liftings* [Pat03, Sch08]. The *coalgebraic μ-calculus* [CKP11a] 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 given instance of the 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 *one-step tableau complete* (a condition that, in the dual setting of sequent calculi, translates into the requirement that the rule absorb cut [SP09, PS10]) and moreover *tractable* in a suitable sense (an assumption made also in our own previous work on the flat [HS15] and alternation-free [HSE16] 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 the monotone μ-calculus [SP09, KP10, CKP11a]. There are, however, important cases where such rule sets are currently missing, and where there is in fact little perspective for finding suitable rules. Prominent cases of this kind are the graded μ-calculus and more expressive logics over integer weights featuring, e.g., Presburger arithmetic[1]. 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 even 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 standard relational μ-calculus is interpreted over models essentially consisting of a set with a distinguished subset, which abstracts the successors of a single state (which is not itself part of the model). We have applied this principle to satisfiability checking in coalgebraic (next-step) modal logics [SP08], coalgebraic hybrid logics [MPS09], and reasoning with global assumptions in coalgebraic modal logics [KPS15, KPS22]. It also appears implicitly in work on automata for the

[1]One-step tableau-complete sets of rules for the graded μ-calculus and more generally for the Presburger μ-calculus have been claimed in earlier work [SP09, KP10] but have since turned out to be in fact incomplete [KPS22, GHH+23]; see also Remark 6.7.

coalgebraic μ-calculus [FLV10], which however establishes only a doubly exponential upper bound in the case without tractable modal tableau rules.

Our main result states roughly that if the satisfiability problem of the one-step logic is in ExpTime under a particular stringent measure of input size, then satisfiability in the corresponding instance of the coalgebraic μ-calculus is in ExpTime. Since the criteria on rule sets featuring in previous work on the coalgebraic μ-calculus imply the ExpTime bound on satisfiability checking in the one-step logic, this result subsumes previous complexity estimates for the coalgebraic μ-calculus [CKP+11b]. Our leading example applications are on the one hand the graded 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 [CKP11a, LSWZ15] with (monotone) polynomial inequalities. While the graded μ-calculus as such is known to be in ExpTime [KSV02], 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- and game-theoretic treatment by means of standard parity automata and parity (satisfiability) games. Our satisfiability games are quite different from satisfiability games formulated previously for the very similar Λ-*automata* [FLV10]. Both types of game are of overall doubly exponential size; however, we show that our game can nevertheless be solved in singly exponential time even in the absence of a tractable set of modal tableau rules. Our algorithm witnessing the singly exponential time bound is able to decide the satisfiability of nodes on-the-fly, that is, possibly before the tableau is fully expanded, in the spirit of global caching algorithms for description logics [GW09, GN13]. It thus offers a perspective for practically feasible reasoning. Our construction implies a known singly-exponential bound on minimum model size for satisfiable formulae in coalgebraic μ-calculi [CKP11a, FLV10], calculated only in terms of the closure size of the formula and its alternation depth. Moreover, we identify a criterion for a polynomial bound on branching in models, which holds in all our examples.

This paper extends a previous conference publication [HS19]. Besides providing full proofs and additional background material, we also make do without the problematic guardedness transformation (cf. [BFL15, KMV20]) and directly establish our results for the unguarded coalgebraic μ-calculus. On a technical level, we rebase large parts of the development on newly introduced notions of model checking game and satisfiability game for the unguarded coalgebraic μ-calculus, obtaining streamlined proofs. A restriction of the algorithm to guarded formulae has been implemented within the *Coalgebraic Ontology Logic Solver (COOL 2)* [GHH+23], which in particular provides the first implemented reasoner for the graded μ-calculus.

Overview of the material. In Section 2, we recall the basics of coalgebra, coalgebraic logic, and the coalgebraic μ-calculus, including central syntactic notions such as closure and alternation depth. We discuss the automata-theoretic approach and *model checking games* in Section 3. Specifically, we introduce the *tracking automaton*, a nondeterministic parity automaton that essentially detects bad sequences of fixpoint unfoldings, and its co-determinization. The nondeterministic tracking automaton implicitly governs the model checking game, while the co-determinized tracking automaton forms the basis of our notion of *tableaux*, introduced in Section 4. Tableaux are certain partial subautomata of the co-determinized tracking automaton, characterized on the one hand by being *totally accepting* in the sense that every infinite run is accepting, and on the other hand by one-step satisfiability requirements on modal constraints. Tableaux allow for the construction of a model of the target formula: We show that on every tableau, one has a so-called coherent coalgebra structure, in what in coalgebraic logic is usually termed the existence lemma (Lemma 4.11). The associated truth lemma (Lemma 4.12) uses the model checking game to show that the target formula is actually satisfied in such a coherent coalgebra, precisely exploiting the fact that the co-determinized tracking automaton complements the tracking automaton.

The *satisfiability game*, introduced in Section 5, then essentially serves to determine whether there exists a tableau for the target formula. Its correctness, in the sense of actually capturing satisfiability of the target formula, is split into two implications: On the one hand, we show that a tableau can indeed be extracted from a winning strategy of the existential player (Lemma 5.3), and on the other hand we prove a soundness lemma (Lemma 5.4) stating that a winning strategy for the existential player in the satisfiability game can be extracted from a winning strategy in the model checking game on a given model of the target formula.

It remains to obtain an exponential-time satisfiability checking algorithm from the satisfiability game, which as indicated above has doubly exponential size. We show in Section 6 that winning regions in the satisfiability game can be characterized as a nested fixpoint that lives on the singly-exponential sized subset of those game positions that correspond to states of the co-determinized tracking automaton (Lemma 6.3). Our satisfiability checking algorithm, also presented in Section 6, essentially computes this fixpoint, or more precisely decides containment of game positions, in particular the root position, in the fixpoint on-the-fly. As indicated above, efficiency of the algorithm relies on sufficiently low complexity of the one-step satisfiability problem. We show that all our example logics satisfy this criterion, and are hence decidable in exponential time

Related Work. Our work builds on existing approaches to the algorithmic treatment of the relational μ-calculus, most centrally on the game-theoretic approach pioneered by [NW96]. Our correctness proof for the model checking game takes orientation from Venema [Ven06] in that it makes do without (ordinal) timeouts, i.e. without (transfinite) Kleene iteration, for least fixpoints. As indicated above, our approach is distinguished by working directly with unguarded formulae. Friedman and Lange [FL13] have previously provided a tableau method dealing with unguarded formulae in the relational μ-calculus; our approach differs technically from theirs, with details discussed in Remark 3.8.

We have already mentioned previous work on the algorithmics of the coalgebraic μ-calculus [CKP11a, FLV10] (other recent work on the coalgebraic μ-calculus concerns model theory, notably completeness [SV18, ESV19] and expressive completeness [ESV17]). As noted above, the main difference with work on tableau-based algorithms [CKP11a] is that we make do without a tractable complete set of tableau rules and cover unguarded formulae, a point that was previously explicitly left open [CKP11a, Section 1]. While the overall technical layout of our approach is similar, e.g. in the use of tracking automata as well as model checking and satisfiability games, the added generality in our work implies substantial differences in the actual implementation of these tools. These concern, for instance, the alphabet used by the tracking automata and the treatment of propositional operators. Fontaine et al. [FLV10] present an approach, interestingly not assuming guardedness, where a coalgebraic μ-calculus formula is first transformed into a so-called Λ-automaton, whose emptiness can then be checked by means of a satisfiability game, which in fact is a regular game but not a parity game. This game has doubly exponential size and in general yields only a doubly exponential complexity bound. We also use a satisfiability game in our approach, which works directly on the formula syntax; altogether, the design of our game appears to differ rather markedly from that used by Fontaine et al. First off, as mentioned above, our game has a parity winning condition. It does also have doubly exponential size but can nevertheless be solved in singly exponential time using a fixpoint calculation on a singly-exponential-sized subset of the game positions. It seems unlikely that similar ideas will work for the game used by Fontaine et al., as out of the two types of positions present in this game, one is of doubly exponential size and the other of only polynomial size; more details are found in Remark 6.8.

2. The Coalgebraic μ-Calculus

We recall basic definitions in coalgebra [Rut00], coalgebraic logic [Pat03, Sch08], and the coalgebraic μ-calculus [CKP11a]. While we do repeat definitions of some categorical terms, we assume some familiarity with basic category theory (e.g. [AHS90]).

Coalgebra. The semantics of our logics will be based on transition systems in a general sense, which we abstract as coalgebras for a type functor. The most basic example are relational transition systems, or Kripke frames, which are just pairs (C, R) consisting of a set C of *states* and a binary *transition relation* $R \subseteq C \times C$. We may equivalently view such a structure as a map of type $\xi\colon C \to \mathcal{P}C$ (where $\mathcal{P}$ denotes the powerset functor and we omit brackets around functor arguments following standard convention, while we continue to use brackets in uses of powerset that are not related to functoriality), via a bijective correspondence that maps $R \subseteq C \times C$ to the map $\xi_R\colon C \to \mathcal{P}C$ given by $\xi_R(c) = \{d \mid (c, d) \in R\}$ for $c \in C$; that is, ξ_R assigns to each state c a collection of some sort, in this case a set, of successor states. The essence of *universal coalgebra* [Rut00] is to base a general theory of state-based systems on encapsulating this notion of *collections of successors* in a functor, for our purposes on the category Set of sets and maps. We recall that such a functor $F\colon \mathsf{Set} \to \mathsf{Set}$ assigns to each set U a set FU, and to each map $f\colon U \to V$ a map $Ff\colon FU \to FV$, preserving identities and composition. As per the intuition given above, FU should be understood as consisting of some form of collections of elements of U. An *F-coalgebra* (C, ξ) then consists of a set C of *states* and a *transition map* $\xi\colon C \to FC$, understood as assigning to each state c a collection $\xi(c) \in FC$ of successors. As indicated above, a basic example is $F = \mathcal{P}$ (with $\mathcal{P}f(Y) = f[Y]$ for $f\colon U \to Y$), in which case F-coalgebras are relational transition systems. To see just one further example now, the *discrete distribution functor* $\mathcal{D}$ is given on sets U by $\mathcal{D}U$ being the set of all discrete probability distributions on U. We represent such distributions by their *probability mass functions*, i.e. functions $d\colon U \to (\mathbb{Q} \cap [0, 1])$ such that $\sum_{u \in U} d(u) = 1$, understood as assigning to each element of U its probability. (Note that the *support* $\{u \in U \mid d(u) > 0\}$ of d is then necessarily at most countable.) By abuse of notation, we also write d for the induced discrete probability measure on U, i.e. $d(Y) = \sum_{y \in Y} d(y)$ for $Y \in \mathcal{P}(U)$. The action of $\mathcal{D}$ on maps $f\colon U \to V$ is then given by $\mathcal{D}f(d)(Z) = d(f^{-1}[Z])$ for $d \in \mathcal{D}U$ and $Z \in \mathcal{P}(V)$, i.e. $\mathcal{D}f$ takes image measures. A $\mathcal{D}$-coalgebra $\xi\colon C \to \mathcal{D}C$ assigns to each state $x \in C$ a distribution $\xi(x)$ over successor states, so $\mathcal{D}$-coalgebras are Markov chains.

Coalgebraic logic. The fundamental abstraction that underlies coalgebraic logic is to encapsulate the semantics of next-step modalities in terms of *predicate liftings*. As a basic example, consider the standard diamond modality $\Diamond$, whose semantics over a transition system (C, R) is given by

$$c \models \Diamond\phi \quad \text{iff} \quad \exists d.\, (c, d) \in R \wedge d \models \phi$$

(where ϕ is a formula in some ambient syntax whose semantics is assumed to be already given by induction). We can rewrite this definition along the correspondence between transition systems and $\mathcal{P}$-coalgebras $\xi\colon C \to \mathcal{P}C$ described above, obtaining

$$\begin{aligned} c \models \Diamond\phi \quad &\text{iff} \quad \xi(c) \cap [\![\phi]\!] \neq \emptyset \\ &\text{iff} \quad \xi(c) \in \{Y \in \mathcal{P}C \mid Y \cap [\![\phi]\!] \neq \emptyset\} \end{aligned}$$

where $[\![\phi]\!] = \{d \in C \mid d \models \phi\}$ denotes the *extension* of ϕ in (C, ξ). We can see the set $\{Y \in \mathcal{P}C \mid Y \cap [\![\phi]\!] \neq \emptyset\}$ as arising from the application of a set operation $[\![\Diamond]\!]$ to the set $[\![\phi]\!]$, thought of as a predicate on C: For a predicate P on C, we put

$$[\![\Diamond]\!](P) = \{Y \in \mathcal{P}C \mid Y \cap P \neq \emptyset\}$$

and then have

$$c \models \Diamond\phi \quad \text{iff} \quad \xi(c) \in [\![\Diamond]\!]([\![\phi]\!]).$$

Notice that the operation $[\![\Diamond]\!]$ turns predicates on C into predicates on $\mathcal{P}C$, so we speak of a *predicate lifting*.

Generally, the notion of predicate lifting is formally defined as follows. The *contravariant powerset* functor $\mathcal{Q}\colon \mathsf{Set}^{op} \to \mathsf{Set}$ is given by $\mathcal{Q}U = \mathcal{P}U$ for sets U, and by $\mathcal{Q}f\colon \mathcal{Q}V \to \mathcal{Q}U$, $\mathcal{Q}f(Z) = f^{-1}[Z]$, for $f\colon U \to V$. We think of $\mathcal{Q}U$ as consisting of predicates on U. Then, an *n-ary predicate lifting* for F is a natural transformation

$$\lambda\colon \mathcal{Q}^n \to \mathcal{Q} \circ F^{op}$$

between functors $\mathsf{Set}^{op} \to \mathsf{Set}$. Here, we write $\mathcal{Q}^n$ for the functor given by $\mathcal{Q}^n U = (\mathcal{Q}U)^n$. Also, recall that F^{op} is the functor $\mathsf{Set}^{op} \to \mathsf{Set}^{op}$ that acts like F on both sets and maps. Unravelling this definition, we see that a predicate lifting λ is a family of maps

$$\lambda_U\colon (\mathcal{Q}U)^n \to \mathcal{Q}(FU),$$

indexed over all sets U, subject to the *naturality* condition requiring that the diagram

$$\begin{tikzcd} (\mathcal{Q}V)^n \arrow[r, "\lambda_V"] \arrow[d, "\mathcal{Q}(f)^n"] & \mathcal{Q}(FV) \arrow[d, "\mathcal{Q}(Ff)"] \\ (\mathcal{Q}U)^n \arrow[r, "\lambda_U"] & \mathcal{Q}(FU) \end{tikzcd}$$

commutes for all $f\colon U \to V$. Explicitly, this amounts to the equality

$$\lambda_U(f^{-1}[Z_1], \dots, f^{-1}[Z_n]) = (Ff)^{-1}\lambda_V(Z_1, \dots, Z_n)$$

for predicates $Z_1, \dots, Z_n \in \mathcal{Q}(V)$. It is easily checked that this condition does hold for the unary predicate lifting $\lambda = [\![\Diamond]\!]$ as defined above. As examples of predicate liftings for the above-mentioned discrete distribution functor $\mathcal{D}$, consider the unary predicate liftings λ^b, indexed over $b \in \mathbb{Q} \cap [0, 1]$, given by

$$\lambda^b_U(Y) = \{d \in \mathcal{D}U \mid d(Y) > b\},$$

which induce next-step modalities 'with probability more than b' (see also Example 2.1.3). Again, the naturality condition is readily checked. Further instances are discussed in Example 2.1.

Fixpoints. We recall basic results on fixpoints, and fix some notation. Recall that by the Knaster-Tarski fixpoint theorem, every monotone function $f\colon X \to X$ on a complete lattice X (such as a powerset lattice $X = \mathcal{P}(Y)$) has a least fixpoint μf and a greatest fixpoint νf, and indeed μf is the least prefixpoint and, dually, νf is the greatest postfixpoint of f. Here, $x \in X$ is a prefixpoint (postfixpoint) of f if $f(x) \leq x$ ($x \leq f(x)$). We use μ and ν also as binders in expressions $\mu X.\, E$ or $\nu X.\, E$ where E is an expression, in an informal sense, possibly depending on the parameter X, thus denoting μf and νf, respectively, for the function f that maps X to E. By the Kleene fixpoint theorem, if X is finite, then we can compute μf as the point $f^n(\bot)$ at which the ascending chain $\bot \leq f(\bot) \leq f^2(\bot) \ldots$ becomes stationary; dually, we compute νf as the point $f^n(\top)$ at which the descending chain $\top \geq f(\top) \geq f^2(\top) \ldots$ becomes stationary. More generally, this method can be applied to unrestricted X by extending the mentioned chains to ordinal indices, taking suprema or infima, respectively, in the limit steps [CC79].

The coalgebraic μ-calculus. Next-step modalities interpreted as predicate liftings can be embedded into ambient logical frameworks of various levels of expressiveness, such as plain modal next-step logics [SP08, SP09] or hybrid next-step logics [MPS09]. The *coalgebraic μ-calculus* [CKP11a] combines next-step modalities with a Boolean propositional base and fixpoint operators, and thus serves as a generic framework for temporal logics.

The **syntax** of the coalgebraic μ-calculus is parametric in a *modal similarity type* Λ, that is, a set of modal operators with assigned finite arities, possibly including propositional atoms as nullary modalities. *We fix a modal similarity type Λ for the rest of the paper.* We assume that Λ is closed under duals, i.e., that for each modal operator $\heartsuit \in \Lambda$, there is a *dual* $\overline{\heartsuit} \in \Lambda$ (of the same arity) such that $\overline{\overline{\heartsuit}} = \heartsuit$ for all $\heartsuit \in \Lambda$. Let $\mathbf{V}$ be a countably infinite set of *fixpoint variables* $X, Y, \ldots$. Formulae $\phi, \psi, \ldots$ of the *coalgebraic μ-calculus* (over Λ) are given by the grammar

$$\psi, \phi ::= \bot \mid \top \mid \psi \wedge \phi \mid \psi \vee \phi \mid \heartsuit(\phi_1, \ldots, \phi_n) \mid X \mid \neg X \mid \mu X.\, \phi \mid \nu X.\, \phi$$

where $\heartsuit \in \Lambda$, $X \in \mathbf{V}$, and n is the arity of $\heartsuit$. We require that negated variables $\neg X$ do not occur in the scope of μX or νX. As usual, μ and ν take least and greatest fixpoints, respectively. We write $\phi[\psi/X]$ for the formula obtained by substituting ψ for X in ϕ. Full negation is not included but can be defined as usual (in particular, $\neg\heartsuit\phi = \overline{\heartsuit}\neg\phi$, and moreover $\neg\mu X.\, \phi = \nu X.\, \neg\phi[\neg X/X]$ and dually; one easily checks that the restriction on occurrences of negated variables is satisfied in the resulting formula). Throughout, we use $\eta \in \{\mu, \nu\}$ as a placeholder for fixpoint operators; we briefly refer to formulae of the form $\eta X.\, \phi$ as *fixpoints* or *fixpoint literals*. We follow the usual convention that the scope of a fixpoint extends as far to the right as possible. Fixpoint operators *bind* their fixpoint variables, so that we have standard notions of bound and free fixpoint variables; we write $\mathsf{FV}(\phi)$ and $\mathsf{BV}(\phi)$ for the sets of free and bound variables, respectively, that occur in a formula ϕ. A formula ϕ is closed if it contains no free fixpoint variables, i.e. $\mathsf{FV}(\phi) = \emptyset$. (Note in particular that closed formulae do not contain negated fixpoint variables, and hence no negation at all.) We assume w.l.o.g. that fixpoints are *irredundant*, i.e. use their fixpoint variable at least once. In *guarded* formulae, all occurrences of fixpoint variables are separated by at least one modal operator from their binding fixpoint operator. Guardedness is a wide-spread assumption although the actual blowup incurred by the transformation of unguarded into guarded formulae depends rather subtly on the notion of formula size [BFL15, KMV20]. We do *not* assume guardedness in this work, see also Remark 3.8. For $\heartsuit \in \Lambda$, we denote by $\mathsf{size}(\heartsuit)$ the length of a suitable representation of $\heartsuit$; for natural or rational numbers indexing $\heartsuit$ (cf. Example 2.1), we assume binary representation. The *length* $|\psi|$ of a formula is its length over the alphabet $\{\bot, \top, \wedge, \vee\} \cup \Lambda \cup \mathbf{V} \cup \{\eta X. \mid \eta \in \{\mu, \nu\}, X \in \mathbf{V}\}$, 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 **semantics** of the coalgebraic μ-calculus, on the other hand, is parametrized by the choice of a functor $F\colon \mathsf{Set} \to \mathsf{Set}$ determining the branching type of systems, as well as predicate liftings interpreting the modalities, as indicated in the above summary of coalgebraic logic. That is, we interpret each modal operator $\heartsuit \in \Lambda$ as an n-ary predicate lifting

$$[\![\heartsuit]\!]\colon \mathcal{Q}^n \to \mathcal{Q} \circ F^{op}$$

for F, where n is the arity of $\heartsuit$, extending notation already used in our lead-in example on the semantics of the diamond modality $\Diamond$. To ensure existence of fixpoints, we require that all $[\![\heartsuit]\!]$ are *monotone*, i.e. whenever $A_i \subseteq B_i \subseteq U$ for all $i = 1, \dots, n$, where n is the arity of $\heartsuit$, then $[\![\heartsuit]\!]_U(A_1, \dots, A_n) \subseteq [\![\heartsuit]\!]_U(B_1 \dots, B_n)$.

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 the assignment of predicate liftings to modalities respects duality, i.e.

$$[\![\overline{\heartsuit}]\!]_U(A_1, \dots, A_n) = \overline{[\![\heartsuit]\!]_U(\overline{A_1}, \dots, \overline{A_n})}$$

for n-ary $\heartsuit \in \Lambda$ and $A_1, \dots, A_n \subseteq U$.

We interpret formulae over F-coalgebras $\xi\colon C \to FC$. A *valuation* is a partial function $i\colon \mathbf{V} \rightharpoonup \mathcal{P}(C)$ that assigns sets $i(X)$ of states to fixpoint variables X (we generally write $\rightharpoonup$ to indicate partial functions). Given $A \subseteq C$ and $X \in \mathbf{V}$, we write $i[X \mapsto A]$ for the valuation given by $(i[X \mapsto A])(X) = A$ and $(i[X \mapsto A])(Y) = i(Y)$ for $Y \neq X$. We write ϵ for the empty valuation (i.e. ϵ is undefined on all variables). For a list $X_1, \dots, X_n$ of distinct variables, we write $i[X_1 \mapsto A_1, \dots, X_n \mapsto A_n]$ for $i[X_1 \mapsto A_1] \dots [X_n \mapsto A_n]$ (i.e. $i[X_1 \mapsto A_1, \dots, X_n \mapsto A_n]$ maps X_j to A_j for $j = 1, \dots, n$ and otherwise acts like i), and $[X_1 \mapsto A_1, \dots, X_n \mapsto A_n]$ for $\epsilon[X_1 \mapsto A_1, \dots, X_n \mapsto A_n]$. The *extension* $[\![\phi]\!]_i \subseteq C$ of a formula ϕ in (C, ξ), under a valuation i such that $i(X)$ is defined for all $X \in \mathsf{FV}(\phi)$, is given by the recursive clauses

$$\begin{aligned}
[\![\bot]\!]_i &= \emptyset \\
[\![\top]\!]_i &= C \\
[\![X]\!]_i &= i(X) \\
[\![\neg X]\!]_i &= C \setminus i(X) \\
[\![\phi \wedge \psi]\!]_i &= [\![\phi]\!]_i \cap [\![\psi]\!]_i \\
[\![\phi \vee \psi]\!]_i &= [\![\phi]\!]_i \cup [\![\phi]\!]_i \\
[\![\heartsuit(\phi_1, \dots, \phi_n)]\!]_i &= \xi^{-1}[[\![\heartsuit]\!]_C([\![\phi_1]\!]_i \dots, [\![\phi_n]\!]_i)] \\
[\![\mu X.\, \phi]\!]_i &= \mu A.\, [\![\phi]\!]_{i[X \mapsto A]} \\
[\![\nu X.\, \phi]\!]_i &= \nu A.\, [\![\phi]\!]_{i[X \mapsto A]}
\end{aligned}$$

(using notation introduced in the fixpoint paragraph above). Thus we have $x \in [\![\heartsuit(\psi_1, \dots, \psi_n)]\!]_i$ if and only if $\xi(x) \in [\![\heartsuit]\!]_C([\![\psi_1]\!]_i, \dots, [\![\psi_n]\!]_i)$. By monotonicity of predicate liftings and the restrictions on occurrences of negated variables, one shows inductively that the functions occurring in the clauses for $\mu X.\, \phi$ and $\nu X.\, \phi$ are monotone, so the corresponding extremal fixpoints indeed exist according to the Knaster-Tarski fixpoint theorem. By an evident substitution lemma, we obtain that the extension is invariant under *unfolding* of fixpoints, i.e.

$$[\![\eta X.\,\psi]\!]_i = [\![\psi[\eta X.\,\psi/X]]\!]_i.$$

For closed formulae 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 $x \in [\![\psi]\!]$. A closed formula χ is *satisfiable* if there is a coalgebra (C, ξ) and a state $x \in C$ such that $x \models \chi$.

For readability, we restrict the further 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. *For the remainder of the paper, we fix a functor F and predicate liftings $[\![\heartsuit]\!]$ interpreting the modalities $\heartsuit \in \Lambda$.*

Example 2.1 (Coalgebraic μ-calculi)**.** We proceed to discuss instances of the coalgebraic μ-calculus, focusing mainly on cases where no tractable set of modal tableau rules is known (details on this point are in Remark 6.7). Examples where such rule sets are available, including the alternating-time μ-calculus, have been provided by Cîrstea et al. [CKP11a]. We do discuss two examples where tractable sets of modal tableau rules are known, viz, the relational μ-calculus and the monotone μ-calculus. We include the former to show how our present coalgebraic notions match the most familiar example; and the latter to illustrate the importance of covering unguarded formulae by showing that these arise in the embedding of game logic [PP03].

(1) The *relational modal μ-calculus* [Koz83] (which contains CTL as a fragment), has $\Lambda = \{\Diamond, \Box\} \cup \mathsf{P} \cup \{\neg a \mid a \in \mathsf{P}\}$ where P is a set of propositional atoms, seen as nullary modalities, and $\Diamond, \Box$ are unary modalities. The modalities $\Diamond$ and $\Box$ are mutually dual ($\overline{\Diamond} = \Box$, $\overline{\Box} = \Diamond$), and the dual of $a \in \mathsf{P}$ is $\neg a$. The semantics is defined over *Kripke models*, which are coalgebras for the functor F given on sets U by $FU = (\mathcal{P}U) \times \mathcal{P}(\mathsf{P})$ – an F-coalgebra assigns to each state a set of successors and a set of atoms satisfied in the state. The relevant predicate liftings are

$$\begin{aligned}
[\![\Diamond]\!]_U(A) &= \{(B, Q) \in FU \mid A \cap B \neq \emptyset\}\\
[\![\Box]\!]_U(A) &= \{(B, Q) \in FU \mid B \subseteq A\}\\
[\![a]\!]_U &= \{(B, Q) \in FU \mid a \in Q\} && (a \in \mathsf{P})\\
[\![\neg a]\!]_U &= \{(B, Q) \in FU \mid a \notin Q\}
\end{aligned}$$

(of course, the predicate liftings for $a \in \mathsf{P}$ and $\neg a$ are nullary, i.e. take no argument predicates). Standard example formulae include the CTL-formula $\mathsf{AF}\,a = \mu X.\,(a \vee \Box X)$, which states that on all paths, a eventually holds, and the fairness formula $\nu X.\,\mu Y.\,((a \wedge \Diamond X) \vee \Diamond Y)$, which asserts the existence of a path on which a holds infinitely often.

(2) The *monotone μ-calculus* [Pau01] has a set P of propositional atoms, seen as nullary modalities in the same way as in our treatment of the relational μ-calculus, and modalities $\langle g \rangle$ with duals $[g]$, indexed over *atomic games* g from a fixed set $\mathcal{A}$. It is interpreted over *monotone neighbourhood models*, which are coalgebras for $\mathcal{M}^{\mathcal{A}} \times \mathcal{P}(\mathsf{P})$. Here, $(-)^{\mathcal{A}}$ denotes exponentiation with $\mathcal{A}$ (generally, for sets U and V, V^U is the set of maps $U \to V$), and $\mathcal{M}$ is the *monotone neighbourhood functor*, defined as follows. We first define the *neighbourhood functor* $\mathcal{N}$ as the composite $\mathcal{Q} \circ \mathcal{Q}^{op}$ of the contravariant powerset functor $\mathcal{Q}$ with itself; explicitly, $\mathcal{N}U = \mathcal{Q}(\mathcal{Q}U)$ is the double powerset of U, and for $f\colon U \to V$ and $N \in \mathcal{N}U$, $\mathcal{N}f(N) = \{Z \subseteq V \mid f^{-1}[Z] \in N\}$. Call $N \in \mathcal{N}U$ *upclosed* if $W \in N$ whenever $V \in N$ and $V \subseteq W$. Then, $\mathcal{M}$ is the subfunctor of $\mathcal{N}$ given by $\mathcal{M}U = \{N \in \mathcal{N}U \mid N \text{ upclosed}\}$. Thus, a monotone neighbourhood model (C, ξ) consists of a set C of states and a transition map ξ assigning to each state $x \in C$ a set of propositional atoms that hold in x, and for each $g \in \mathcal{A}$

an upwards closed set of *g-neighbourhoods*. The semantics of propositional atoms is given like in the relational μ-calculus. The interpretation of the modalities as predicate liftings is given by

$$\begin{aligned}
[\![\langle g\rangle]\!]_U(A) &= \{(N,Q) \in (\mathcal{M}U)^{\mathcal{A}} \times \mathcal{P}(\mathsf{P}) \mid A \in N(g)\}\\
[\![[g]]\!]_U(A) &= \{(N,Q) \in (\mathcal{M}U)^{\mathcal{A}} \times \mathcal{P}(\mathsf{P}) \mid \forall B \in N(g).\, B \cap A \neq \emptyset\}.
\end{aligned}$$

As indicated by the nomenclature, one way to understand the monotone μ-calculus is as a logic of two-player games, where $\langle g\rangle\phi$ says that the first player ('Angel') can enforce ϕ after playing game g. *Game logic* [Par83] features modalities $\langle\gamma\rangle$, $[\gamma]$ for (non-atomic) games γ; here, *games* γ,δ are defined by the grammar

$$\gamma,\delta ::= g \mid \gamma\cup\delta \mid \gamma;\delta \mid \gamma^d \mid \gamma^* \qquad (g \in \mathcal{A})$$

(where for simplification we omit the test construct $\phi?$, for game logic formulae ϕ). Here, $\gamma\cup\delta$ is a game in which Angel decides whether to play γ or δ; $\gamma;\delta$ is the game where γ and δ are played in sequence; γ^* is a game in which γ is played repeatedly, and Angel decides before each round (including the first) whether γ is played again; and γ^d is like γ but with the roles of the players reversed. Thus, $\gamma\cap\delta := (\gamma^d\cup\delta^d)^d$ is a game where the second player ('Demon') decides whether γ or δ is played, and $\gamma^\times := ((\gamma^d)^*)^d$ is a game where γ is played repeatedly, with Demon deciding before each round whether γ is played another time. This logic can be embedded into the monotone μ-calculus, with polynomial blowup when formulae are measured in terms of subformula or closure size [Pau01], as we will do here. We slightly simplify the original translation, which aims at using only two fixpoint variables; the translation t is then defined by mutual recursion with operators τ_γ that translate the effect of applying $\langle\gamma\rangle$ to an argument formula. The translation t is given by commutation with all propositional operators and by

$$t(\langle\gamma\rangle\phi) = \tau_\gamma(t(\phi)).$$

We refrain from listing the clauses for τ_ϕ in full; the most salient clauses are

$$\tau_g(\phi) = \langle g\rangle\phi \qquad \tau_{\gamma^d}(\phi) = \neg\tau_\gamma(\neg\phi) \qquad \tau_{\gamma^*}(\phi) = \mu X.\,(\phi \vee \tau_\gamma(X))$$

where X is a fresh fixpoint variable. For instance, we have

$$t(\langle(g^*)^\times\rangle a) = \nu X.\,(a \wedge \mu Y.\,(X \vee \langle g\rangle Y));$$

that is, the translation of game logic formulae may produce unguarded formulae in the monotone μ-calculus.

(3) The two-valued *probabilistic μ-calculus* [CKP11a, LSWZ15, CK16] (not to be confused with the real-valued probabilistic μ-calculus [HK97, MM07]) is modelled using the distribution functor $\mathcal{D}$ as discussed above; recall in particular that $\mathcal{D}$-coalgebras are Markov chains. To avoid triviality (see Remark 2.3), we include propositional atoms, i.e. we work with coalgebras for the functor $F = \mathcal{D}\times\mathcal{P}(\mathsf{P})$ for a set P of propositional atoms like in the previous items. We use the modal similarity type $\Lambda = \{\langle b\rangle, [b] \mid b \in \mathbb{Q}\cap[0,1]\} \cup \mathsf{P} \cup \{\neg a \mid a \in \mathsf{P}\}$, with $\langle b\rangle$ and $[b]$ mutually dual, and again with $a \in \mathsf{P}$ and $\neg a$ mutually dual; we interpret these modalities by the predicate liftings

$$\begin{aligned}
[\![\langle b\rangle]\!]_U(A) &= \{(d,Q) \in FU \mid d(A) > b\}\\
[\![[b]]\!]_U(A) &= \{(d,Q) \in FU \mid d(U\setminus A) \le b\}\\
[\![a]\!]_U &= \{(d,Q) \in FU \mid a \in Q\}\\
[\![\neg a]\!]_U &= \{(d,Q) \in FU \mid a \notin Q\}
\end{aligned}$$

for sets U and $A \subseteq U$ (so $[\![\langle b\rangle]\!]$ applies the above-mentioned predicate liftings λ^b to the $\mathcal{D}$-component of F); that is, a state satisfies $\langle b\rangle\phi$ if the probability of reaching a state satisfying ϕ in the next step is more than b, and a state satisfies $[b]\phi$ if the probability of reaching a state not satisfying ϕ in the next step is at most b. For example, the formula

$$\phi = \nu X.\, \mathsf{safe} \wedge \langle 0.95\rangle X$$

expresses that current state is safe and will reach a state in which ϕ holds again with probability more than 0.95 (a property that may be more realistically expected to hold in practice than formulae demanding that safety holds forever with a given probability).

(4) Similarly, we interpret the *graded μ-calculus* [KSV02] over *multigraphs* [DV02], in which states are connected by directed edges that are annotated with non-negative integer *multiplicities*. Multigraphs correspond to coalgebras for the *multiset functor* $\mathcal{B}$, defined on sets U by

$$\mathcal{B}(U) = \{\beta : U \to \mathbb{N} \cup \{\infty\}\}.$$

We view $d \in \mathcal{B}(U)$ as an integer-valued measure on U, and for $V \subseteq U$ employ the same notation $d(V) = \sum_{x\in V} d(x)$ as in the case of probability distributions. In this notation, we can, again, define $\mathcal{B}f$, for $f\colon U \to V$, as taking image measures, i.e. $\mathcal{B}f(d)(W) = d(f^{-1}[W])$ for $W \subseteq V$. A $\mathcal{B}$-coalgebra $\xi\colon C \to \mathcal{B}C$ assigns a multiset $\xi(x)$ to each state $x \in C$, which we may read as indicating that given states $x, y \in C$, there is an edge from x to y with multiplicity $\xi(x)(y)$. We use the modal similarity type $\Lambda = \{\langle m\rangle, [m] \mid m \in \mathbb{N} \cup \{\infty\}\}$, with $\langle m\rangle$ and $[m]$ mutually dual, and define the predicate liftings

$$[\![\langle m\rangle]\!]_U(A) = \{\beta \in \mathcal{B}U \mid \beta(A) > m\}$$
$$[\![[m]]\!]_U(A) = \{\beta \in \mathcal{B}U \mid \beta(U \setminus A) \le m\}$$

for sets U and $A \subseteq U$. For instance, somewhat informally speaking, a state satisfies $\nu X.\, (\psi \wedge \langle 1\rangle X)$ if it is the root of an embedded infinite binary tree in which all states satisfy ψ (more formally, this holds in the tree unfolding of the coalgebra).

(5) The *probabilistic μ-calculus with polynomial inequalities* [KPS15] extends the probabilistic μ-calculus (item 3) by introducing, in addition to propositional atoms as in item 3, mutually dual *polynomial modalities* $\langle p\rangle, [p]$ for $n \in \mathbb{N}$, $p \in \mathbb{Q}[X_1, \dots, X_n]$ (the set of polynomials in variables $X_1, \dots, X_n$ with rational coefficients) such that p is monotone on $[0, 1]$ in all variables (in particular this holds when all coefficients of non-constant monomials in p are non-negative, but also, e.g., $X_1 - \frac{1}{2}X_1^2 - \frac{1}{4}$ is monotone on $[0, 1]$). These modalities are interpreted by predicate liftings defined by

$$[\![\langle p\rangle]\!]_U(A_1, \dots, A_n) = \{(d, Q) \in FU \mid p(d(A_1), \dots, d(A_n)) > 0\}$$
$$[\![[p]]\!]_U(A_1, \dots, A_n) = \{(d, Q) \in FU \mid p(d(\overline{A_1}), \dots, d(\overline{A_n})) \le 0\}$$

for sets U and $A_1, \dots, A_n \subseteq U$. Of course, the polynomial modalities subsume the modalities mentioned in item 3, explicitly via $\langle b\rangle = \langle X_1 - b\rangle$ and $[b] = [X_1 - b]$. The monotonicity restriction on polynomials ensures that polynomial modalities are monotone, which in turn is needed to guarantee existence of least and greatest fixpoints. Polynomial inequalities over probabilities have received some previous interest in probabilistic logics (e.g. [FHM90, GJLS17]), in particular as they can express constraints on independent events (and hence play a role analogous to independent products as used in the real-valued probabilistic μ-calculus [Mio11]). E.g. the formula

$$\nu Y.\, \langle X_1X_2 - 0.9\rangle(\mathsf{ready} \wedge Y, \mathsf{idle} \wedge Y)$$

says roughly that two independently sampled successors of the current state will satisfy ready and idle, respectively, and then satisfy the same property again, with probability more than 0.9.

We note that polynomial inequalities clearly increase the expressiveness of the logic strictly, not only in comparison to the logic with operators $\langle b \rangle$, $[b]$ as in item 3 but also w.r.t. the logic with only linear inequalities: The restrictions on probability distributions imposed by linear inequalities on probabilities are rational polytopes, so already $\langle X_1^2 + X_2^2 - 1 \rangle(a, c)$ is not expressible using only linear inequalities.

(6) Similarly, the *graded μ-calculus with polynomial inequalities* extends the graded μ-calculus with more expressive modalities $\langle p \rangle, [p]$, again mutually dual, where in this case $n \in \mathbb{N}$, $p \in \mathbb{Z}[X_1, \dots, X_n]$ (that is, p ranges over multivariate polynomials with integer coefficients). Again, we restrict polynomials to be monotone in all variables; we do in this case in fact require that all coefficients of non-constant monomials are non-negative. To avoid triviality, we correspondingly require the coefficient b_0 of the constant monomial to be non-positive; we refer to the number $-b_0$ as the *index* of the modality. These modalities are interpreted by the predicate liftings

$$\llbracket \langle p \rangle \rrbracket_U(A_1, \dots, A_n) = \{\beta \in \mathcal{B}U \mid p(\beta(A_1), \dots, \beta(A_n)) > 0)\}$$
$$\llbracket [p] \rrbracket_U(A_1, \dots, A_n) = \{\beta \in \mathcal{B}U \mid p(\beta(\overline{A_1}), \dots, \beta(\overline{A_n})) \leq 0)\}.$$

This logic subsumes the *Presburger μ-calculus*, that is, the extension of the graded μ-calculus with linear inequalities (with non-negative coefficients), which may be seen as the fixpoint variant of *Presburger modal logic* [DL06]. E.g. the formula

$$\mu Y. \, (a \vee \langle 3X_1 + X_2^2 - 10 \rangle (c \wedge Y, a \wedge Y))$$

says that (in the tree unfolding of the coalgebra) the current state is the root of a finite tree all whose leaves satisfy a, and each of whose inner nodes has n_1 children satisfying c and n_2 children satisfying a where $3n_1 + n_2^2 - 10 > 0$. The index of the modality $\langle 3X_1 + X_2^2 - 10 \rangle$ is 10.

Unlike in the probabilistic case (item 5), polynomial inequalities do not increase the expressiveness of the graded μ-calculus (item 4): Given a polynomial $p \in \mathbb{Z}[X_1, \dots, X_n]$, one can just replace $\langle p \rangle(\phi_1, \dots, \phi_n)$ with the disjunction of all formulae $\bigwedge_{i=1}^n \langle m_i - 1 \rangle \phi_i$ over all solutions $(m_1, \dots, m_n)$ of $p(X_1, \dots, X_n) > 0$ that are minimal w.r.t. the componentwise ordering of $\mathbb{N}^n$. As these minimal solutions form an antichain in $\mathbb{N}^n$, there are only finitely many of them [Lav76], so the disjunction is indeed finite. However, the blowup of this translation is exponential in the binary size of the coefficient of the constant monomial and in n: For instance, even the linear inequality $X_1 + \dots + X_n - nb > 0$ has, by a somewhat generous estimate, at least $(b+1)^{n-1}$ minimal solutions (one for each assignment of numbers between 0 and b to the variables $X_1, \dots, X_{n-1}$; more precisely, the number of minimal solutions is the number $\binom{n+nb}{n-1} = \binom{n+nb}{nb+1}$ of weak compositions of $nb + 1$ into n parts), which is exponential both in n and in the binary size of b (in which the polynomial itself has linear size when n is fixed). Therefore, this translation does not allow inheriting an exponential-time upper complexity bound on satisfiability checking from the graded μ-calculus.

(7) Coalgebraic logics in general combine along functor composition, and essentially all their properties including their algorithmic treatment propagate; the arising composite logics are essentially typed sublogics of the fusion, to which we refer as *multi-sorted coalgebraic logics* [SP11]. For instance, Markov decision processes or simple Segala systems may (in the simplest version) be seen as coalgebras for the composite functor $\mathcal{P} \circ \mathcal{D}$. A logic for

such systems has nondeterministic modalities $\Diamond, \Box$ as well as probabilistic modalities such as $\langle p \rangle$, $[p]$ (items 1 and 5). The logic distinguishes two sorts of *nondeterministic* and *probabilistic* formulae, respectively, with $\Diamond, \Box$ taking probabilistic formulae as arguments and producing nondeterministic formulae, and with $\langle p \rangle$, $[p]$ working the other way around (so that, e.g., $\langle X^2 - 0.5 \rangle \Diamond \top$ is a probabilistic formula while $\Diamond \top \wedge \langle X^2 - 0.5 \rangle \Diamond \top$ is not a formula). Two further particularly pervasive and basic functors are $(-)^A$, which represents indexing of transitions (and modalities) over a fixed set A of actions or labels (like in item 2 above for $A = \mathcal{A}$), and $(-) \times \mathcal{P}(\mathsf{P})$, which represents state-dependent valuations for a set P of propositional atoms as featured already in several items above. Our results will therefore cover modular combinations of these features with the logics discussed above, e.g. logics for labelled Markov chains or Markov decision processes. We refrain from repeating the somewhat verbose full description of the framework of multi-sorted coalgebraic logic. Instead, we will consider only the fusion of logics in the following – since, as indicated above, multi-sorted logics embed into the fusion [SP11], this suffices for purposes of algorithmic satisfiability checking. We discuss the fusion of coalgebraic logics in Remarks 2.5 and 6.9.

We remark that the modalities in items 5 and 6 are less general than in the corresponding next-step logics [FHM90, GJLS17, DL06, KPS15] in that the polynomials involved are restricted to be monotone; e.g. they do not support statements of the type 'ϕ is more probable than ψ'. This is owed to their use in fixpoints, which requires modalities to be monotone as indicated above.

Coalgebraic logics relate closely to a generic notion of behavioural equivalence, which generalizes bisimilarity of transition systems. A *morphism* $h\colon (C, \xi) \to (D, \zeta)$ of F-coalgebras is a map $h\colon C \to D$ such that the square

$$\begin{array}{ccc} C & \xrightarrow{h} & D \\ {\scriptstyle \xi}\downarrow & & \downarrow{\scriptstyle \zeta} \\ FC & \xrightarrow[Fh]{} & FD \end{array}$$

commutes. For instance, a morphism $h\colon (C, \xi) \to (D, \zeta)$ of $\mathcal{P}$-coalgebras, i.e. of transition systems or Kripke frames, is precisely what is known as a bounded morphism or a p-morphism (e.g. [BdRV01]), that is, (i) whenever $c' \in \xi(c)$ for $c, c' \in C$, then $h(c') \in \zeta(h(c))$, and (ii) whenever $d' \in \zeta(d)$ for $d, d' \in D$, then there exists $c' \in \xi(c)$ such that $h(c') = d'$. States $x \in C$, $y \in D$ in coalgebras (C, ξ), (D, ζ) are *behaviourally equivalent* if there exist a coalgebra (E, θ) and morphisms $g\colon (C, \xi) \to (E, \theta)$, $h\colon (D, \zeta) \to (E, \theta)$ such that $g(x) = h(y)$. This notion instantiates to standard equivalences in our running examples; e.g. as indicated above, states in (labelled) transition systems are behaviourally equivalent iff they are bisimilar in the usual sense [AM89], and states in labelled Markov chains are behaviourally equivalent iff they are probabilistically bisimilar [RdV99], [BSdV04].

In view of the definition of behavioural equivalence via morphisms of coalgebras, we can phrase invariance of the coalgebraic μ-calculus under behavioural equivalence, generalizing the well-known bisimulation invariance of the relational μ-calculus, as invariance under coalgebra morphisms. Formally, this takes the following shape:

Lemma 2.2 (Invariance under behavioural equivalence). *Let $h\colon (C, \xi) \to (D, \zeta)$ be a morphism of coalgebras, let ϕ by a coalgebraic μ-calculus formula, and let $i\colon \mathbf{V} \rightharpoonup \mathcal{P}(D)$ be a valuation. Then*

$$[\![\phi]\!]_{h^{-1}i} = h^{-1}[\![\phi]\!]_i$$

where $h^{-1}i$ denotes the valuation given by $h^{-1}i(X) = h^{-1}[i(X)]$.

Since as mentioned above, fixpoints can be approximated by ordinal-indexed iteration, this follows from the fact that (fixpoint-free) coalgebraic modal logic with infinite conjunctions and disjunctions is invariant under behavioural equivalence [Pat04, Sch08]. Similar statements have been shown for a version of the coalgebraic μ-calculus featuring the coalgebraic cover modality instead of predicate-lifting-based modalities [Ven04] and for the single-variable fragment of the coalgebraic μ-calculus in the present sense [SV18], using essentially the same argument.

Remark 2.3. As indicated in Example 2.1.3, Markov chains are coalgebras for the discrete distribution functor $\mathcal{D}$. The reason why we immediately extended Markov chains with propositional atoms is that in plain Markov chains, all states are behaviourally equivalent and therefore satisfy the same formulae of any coalgebraic μ-calculus interpreted over Markov chains: Since for any singleton set 1, $\mathcal{D}1$ is again a singleton, we have a unique $\mathcal{D}$-coalgebra structure on 1, and every $\mathcal{D}$-coalgebra has a (unique) morphism into this coalgebra. This collapse under behavioural equivalence is avoided by adding propositional atoms.

Remark 2.4 (Multigraph semantics of the graded μ-calculus)**.** One important consequence of the invariance of the coalgebraic μ-calculus under coalgebra morphisms according to Lemma 2.2 is that the multigraph semantics of the graded μ-calculus as per Example 2.1.4 (and its extension with polynomial inequalities introduced in Example 2.1.6) is equivalent to the more standard Kripke semantics, i.e. the semantics over $\mathcal{P}$-coalgebras [KSV02]. We can obtain the latter from the definitions in Example 2.1.4 by converting Kripke frames into multigraphs in the expected way, i.e. by regarding transitions in the given Kripke frame as transitions with multiplicity 1 in a multigraph. The conversion shows trivially that every formula that is satisfiable over (finite) Kripke frames is also satisfiable over (finite) multigraphs. The converse is shown by converting a given multigraph into a Kripke model that has copies of states and transitions according to the multiplicity of transitions in the multigraph. The arising Kripke frame, converted back into a multigraph as described previously, has a coalgebra morphism into the original multigraph, implying by Lemma 2.2 that the copies satisfy the same formulae as the original multigraph states. Details are given in [SV18, Lemma 2.4].

Remark 2.5 (Fusion)**.** The standard term *fusion* refers to a form of combination of logics, in particular modal logics, where the ingredients of both logics are combined disjointly and essentially without semantic interaction, but may then be intermingled in composite formulae. In the framework of coalgebraic logic, this means that the fusion of coalgebraic logics with modal similarity types Λ_i interpreted over functors F_i, $i = 1, 2$, is formed by taking the disjoint union Λ of Λ_1 and Λ_2 (assuming w.l.o.g. that Λ_1 and Λ_2 are disjoint to begin with) as the modal similarity type, and by interpreting modalities over the product functor $F = F_1 \times F_2$ (with the product of functors computed componentwise, e.g. $FU = F_1U \times F_2U$ for sets U). The predicate lifting $[\![\heartsuit]\!]^F$ interpreting $\heartsuit \in \Lambda_i$ over F, for $i = 1, 2$, is given by

$$[\![\heartsuit]\!]^F_U(A) = \{(t_1, t_2) \in F_1U \times F_2U \mid t_i \in [\![\heartsuit]\!]^{F_i}_U(A)\}$$

where $[\![\heartsuit]\!]^{F_i}$ is the predicate lifting for F_i interpreting $\heartsuit$ in the respective component logic. For example, the fusion of the standard relational modal μ-calculus (Example 2.1.1) and the probabilistic μ-calculus with polynomial inequalities (Example 2.1.5) allows for unrestricted use of non-deterministic modalities $\Diamond, \Box$ and probabilistic modalities $\langle p\rangle$, $[p]$ in formulae such as $\Diamond\top \wedge \langle X^2 - 0.5\rangle\Diamond\top$, instead of only the alternating discipline imposed by the two-sorted logic for Markov decision processes described in Example 2.1.7; such formulae are interpreted over coalgebras for the functor $\mathcal{P} \times \mathcal{D}$, in which every state has both nondeterministic and probabilistic successors.

Closure and Alternation Depth. A key measure of the complexity of a formula is its *alternation depth*, which roughly speaking describes the maximal number of alternations between μ and ν in chains of dependently nested fixpoints. The treatment of this issue is simplified if one excludes reuse of fixpoint variables:

Definition 2.6 (Clean formulae)**.** A closed formula ϕ is *clean* if each fixpoint variable appears in at most one fixpoint operator in ϕ.

We fix a clean closed target formula χ *for the remainder of the paper*, and define the following syntactic notions relative to this target formula. Given $X \in \mathsf{BV}(\chi)$, we write $\theta(X)$ for the unique subformula of χ of the shape $\eta X.\, \psi$. We say that a fixpoint variable X is an *η-variable*, for $\eta \in \{\mu, \nu\}$, if $\theta(X)$ has the form $\eta X.\, \psi$. As indicated, the formal definition of alternation depth simplifies within clean formulae (e.g., [Wil01, KMV20, KMV22]):

Definition 2.7 (Alternation depth)**.** For $X, Y \in \mathsf{BV}(\chi)$, we write $Y \prec_{\mathsf{dep}} X$ if $X \in \mathsf{FV}(\theta(Y))$ (which implies that $\theta(Y)$ is a subformula of $\theta(X)$). A *dependency chain* in χ is a chain of the form

$$X_n \prec_{\mathsf{dep}} X_{n-1} \prec_{\mathsf{dep}} \dots \prec_{\mathsf{dep}} X_0 \qquad (n \geq 0);$$

the *alternation number* of the chain is $k+1$ where k is the number of indices $i \in \{0, \dots, n-1\}$ such that X_i is a μ-variable iff X_{i+1} is a ν-variable (i.e. the chain toggles between μ- and ν-variables at i). The *alternation depth* $\mathsf{ad}(X)$ of a variable $X \in \mathsf{BV}(\chi)$ is the maximal alternation number of dependency chains as above ending in $X_0 = X$, and the *alternation depth* of χ is $\mathsf{ad}(\chi) := \max\{\mathsf{ad}(X) \mid X \in \mathsf{BV}(\chi)\}$.

(In particular, the alternation depth of χ is 0 if χ does not contain any fixpoints, and 1 if χ is *alternation-free*, i.e. no ν-variable occurs freely in $\theta(X)$ for a μ-variable X, and vice versa.)

Example 2.8 (Alternation depth)**.** The formula $\nu X.\, (\mu Y.\, a \vee \Diamond Y) \wedge \Box X$ (in the relational modal μ-calculus) has alternation depth 1, i.e. is alternation-free. The formula $\nu X.\, \Box(\mu Y.\, X \vee \Diamond Y)$ has alternation depth 2, as witnessed by the alternating chain $Y \prec_{\mathsf{dep}} X$.

In the automata-theoretic approach to checking for infinite deferrals, automata states will be taken from the Fischer-Ladner closure of χ in the usual sense [Koz83], in which subformulae of χ are expanded into closed formulae by means of fixpoint unfolding:

Definition 2.9 (Fischer-Ladner closure)**.** The *Fischer-Ladner closure* $\mathbf{F}$ of χ is the least set of formulae containing χ such that

$$\begin{aligned} \phi \wedge \psi \in \mathbf{F} &\implies \phi, \psi \in \mathbf{F} \\ \phi \vee \psi \in \mathbf{F} &\implies \phi, \psi \in \mathbf{F} \\ \heartsuit\phi \in \mathbf{F} &\implies \phi \in \mathbf{F} \\ \eta X.\, \phi \in \mathbf{F} &\implies \phi[\eta X.\, \phi / X] \in \mathbf{F}. \end{aligned}$$

(One should note that although χ is clean, the elements of $\mathbf{F}$ will in general fail to be clean, as fixpoint unfolding of $\eta X.\, \phi$ as per the last clause may create multiple copies of $\eta X.\, \phi$.)

Furthermore, we let $\mathsf{sub}(\chi)$ denote the set of subformulae of χ; unlike formulae in $\mathbf{F}$, formulae in $\mathsf{sub}(\chi)$ may contain free fixpoint variables. The *innermost* free fixpoint variable in a subformula of χ is the one whose binder lies furthest to the right in χ. Each $\phi \in \mathsf{sub}(\chi)$ induces a formula $\theta^*(\phi) \in \mathbf{F}$, which is obtained by repeatedly transforming ϕ, in each step substituting the innermost fixpoint variable X occurring in the present formula with $\theta(X)$ [Koz83]. This map witnesses finiteness of the closure (Lemma 2.10) and moreover will serve as a connection between two variants of the model checking game respectively based on subformulae and on the closure (Lemma 3.14). For instance, for $\chi = \mu X.\, \nu Y.\, (Y \vee \heartsuit X)$, we have $Y \vee \heartsuit X \in \mathsf{sub}(\chi)$, $\theta(X) = \chi$ and $\theta(Y) = \nu Y.\, (Y \vee \heartsuit X)$, and thus

$$\begin{aligned}\theta^*(Y \vee \heartsuit X) &= \theta^*(\theta(Y) \vee \heartsuit X)\\ &= \theta^*((\nu Y.\, (Y \vee \heartsuit X)) \vee \heartsuit X)\\ &= (\nu Y.\, (Y \vee \heartsuit\theta(X))) \vee \heartsuit\theta(X)\\ &= (\nu Y.\, (Y \vee \heartsuit\chi)) \vee \heartsuit\chi.\end{aligned}$$

Lemma 2.10 [Koz83]**.** *The function $\theta^* \colon \mathsf{sub}(\chi) \to \mathbf{F}$ is surjective; in particular, the cardinality of $\mathbf{F}$ is at most the number of subformulae of χ.*

Remark 2.11 (Closure vs. subformulae)**.** The main reason we base our constructions on the closure $\mathbf{F}$ is that this provides the most succinct size measure for μ-calculus formulae [BFL15], thus strengthening our complexity results. Subformulae are needed occasionally for technical purposes; in particular, they support proofs by structural induction.

3. Tracking Automata and Model Checking Games

Generally, the main problem both in satisfiability checking for temporal logics and the associated model constructions on the one hand, and in model checking on the other hand, is to ensure that the unfolding of least fixpoints does not lead to infinite deferrals, i.e. that least fixpoints are indeed eventually satisfied. We encode this condition using *parity automata* (e.g. [GTW02]) that track the evolution of formulae in such procedures.

Recall that a *(nondeterministic) parity automaton* is a tuple

$$\mathsf{A} = (V, \Sigma, \Delta, q_{\mathsf{init}}, \alpha)$$

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*; $q_{\mathsf{init}} \in V$ is the *initial node*; and $\alpha : V \to \mathbb{N}$ is the *priority function*, which assigns priorities $\alpha(v) \in \mathbb{N}$ to states $v \in V$. Given a ternary relation $R \subseteq A \times B \times A$ and $a \in A$, $b \in B$, $B' \subseteq B$, we generally write $R(a, b) = \{a' \in A \mid (a, b, a') \in R\}$, $R(a, B') = \bigcup_{b \in B'} R(a, b)$ and $R(a) = R(a, B)$. If Δ is a (partial) functional relation, then A is said to be *deterministic*, and we denote the corresponding partial function by $\delta : V \times \Sigma \rightharpoonup V$. We often treat infinite sequences $s = x_1, x_2, \ldots$ over a base set X as maps $s \colon \mathbb{N} \to X$. We generally write

$$\mathsf{Inf}(s) = \{x \in X \mid |s^{-1}[\{x\}]| = \infty\}$$

for the set of elements that occur infinitely often in s. The automaton A *accepts* an *infinite word*, i.e. an infinite sequence $w = \sigma_0, \sigma_1, \ldots \in \Sigma^\omega$ over Σ, 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

$$L(\mathsf{A}) = \{w \in \Sigma^\omega \mid \exists \rho \in \mathsf{run}(\mathsf{A}, w).\ \max(\mathsf{Inf}(\alpha \circ \rho)) \text{ is even}\},$$

where $\mathsf{run}(\mathsf{A}, w)$ denotes the set of *runs of* A *on* w, i.e. infinite sequences $q_0, q_1, q_2, \ldots \in V^\omega$ (starting in the initial state $q_0 = q_{\mathsf{init}}$) such that $q_{i+1} \in \Delta(q_i, w_i)$ for all $i \geq 0$.

Recall moreover that we are working with a fixed clean closed target formula χ. We put

$$n_0 = |\chi|, \quad n_1 = \mathsf{size}(\chi), \quad k = \mathsf{ad}(\phi).$$

The states of the tracking automaton for χ will be the elements of the Fischer-Ladner closure $\mathbf{F}$ of χ (cf. Section 2); in particular, $|\mathbf{F}| \leq n_0$.

Definition 3.1 (Modal literals)**.** Given a set Z, we write $\Lambda(Z) = \{\heartsuit z \mid \heartsuit \in \Lambda, z \in Z\}$, and refer to elements of $\Lambda(Z)$ as *modal literals* (over Z).

In particular, $\mathbf{F} \cap \Lambda(\mathbf{F})$ is the set of modal literals in $\mathbf{F}$. We put

$$\mathsf{selections} = \mathcal{P}(\mathbf{F} \cap \Lambda(\mathbf{F})),$$

and indeed refer to elements of this set as *selections*, with a view to using selections as letters for modal steps in our automata construction. Furthermore, we let $\mathbf{F}_\vee = \{\psi \vee \chi \mid \psi \vee \chi \in \mathbf{F}\}$ denote the set of disjunctive formulae contained in $\mathbf{F}$, and put

$$\mathsf{choices} = \{\tau : \mathbf{F}_\vee \to \mathbf{F} \mid \forall (\psi \vee \chi) \in \mathbf{F}_\vee.\, \tau(\psi \vee \chi) \in \{\psi, \chi\}\};$$

that is, $\mathsf{choices}$ consists of *choice functions* that pick disjuncts from disjunctions. These choice functions will be used as letters for propositional steps in the tracking automaton. We note $|\mathsf{selections}|, |\mathsf{choices}| \leq 2^{n_0}$.

Definition 3.2 (Tracking automaton)**.** The *tracking automaton* for χ is the nondeterministic parity automaton $\mathsf{A}_\chi = (\mathbf{F}, \Sigma, \Delta, q_{\mathsf{init}}, \alpha)$ where $q_{\mathsf{init}} = \chi$, $\Sigma = \mathsf{choices} \cup \mathsf{selections}$, and for $\psi \in \mathbf{F}$, $\tau \in \mathsf{choices}$ and $\kappa \in \mathsf{selections}$,

$$\Delta(\psi, \tau) = \begin{cases} \{\tau(\psi)\} & \text{if } \psi \in \mathbf{F}_\vee \\ \{\psi_0, \psi_1\} & \text{if } \psi = \psi_0 \wedge \psi_1 \\ \{\psi_1[\psi/X]\} & \text{if } \psi = \eta X.\, \psi_1 \\ \{\psi\} & \text{if } \psi = \heartsuit\psi_0 \text{ for some } \heartsuit \in \Lambda \\ \emptyset & \text{if } \psi \in \{\top, \bot\} \end{cases}$$

$$\Delta(\psi, \kappa) = \begin{cases} \{\psi_0\} & \text{if } \psi = \heartsuit\psi_0 \in \kappa \text{ for some } \heartsuit \in \Lambda \\ \emptyset & \text{otherwise} \end{cases}$$

The priority function α is derived from the alternation depths of variables, counting only unfoldings of fixpoints (i.e. all other formulae have priority 1) and ensuring that least fixpoints receive even priority and greatest fixpoints receive odd priority. That is, we put

$$\begin{aligned} \alpha(\mu X.\phi) &= 2\lfloor(\mathsf{ad}(X) - 1)/2\rfloor + 2 \\ \alpha(\nu X.\phi) &= 2\lfloor\mathsf{ad}(X)/2\rfloor + 1 \\ \alpha(\psi) &= 1 && \text{if } \psi \text{ is not a fixpoint literal.} \end{aligned}$$

(For instance, a formula of the form $\chi_1 = \nu X.\, \mu Y.\nu Z.\, \phi(X, Y, Z)$ has $\mathsf{ad}(X) = 3$ and $\alpha(\chi_1) = 3$, while its unfolding $\chi_1' = \mu Y.\nu Z.\, \phi(\chi_1, Y, Z)$ has $\mathsf{ad}(Y) = 2$ and $\alpha(\chi_1') = 2$. Contrastingly, a formula of the form $\chi_2 = \mu X.\, \nu Y.\mu Z.\, \phi(X, Y, Z)$ has $\mathsf{ad}(X) = 3$ and $\alpha(\chi_2) = 4$, while its unfolding $\chi_2' = \nu Y.\mu Z.\, \phi(\chi_2, Y, Z)$ has $\mathsf{ad}(Y) = 2$ and $\alpha(\chi_2') = 3$.)

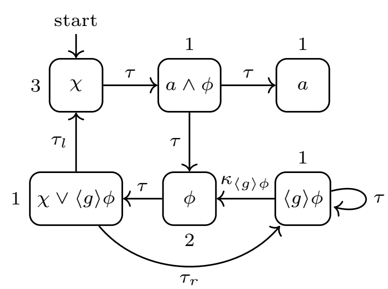
Propositional tracking along a choice function $\tau \in \mathsf{choices}$ thus follows the choice of τ for disjunctions. Conjunctions are tracked nondeterministically to one of their conjuncts; fixpoint literals $\psi = \eta X.\, \psi_1$ are tracked to their unfolding $\psi_1[\psi/X]$; modal literals are left unchanged by propositional tracking; and truth constants $\top, \bot$ are not further tracked at all. In modal tracking along a selection κ, a modal literal $\psi = \heartsuit\psi_0$ is tracked (to ψ_0) only if $\heartsuit\psi_0 \in \kappa$, i.e. if κ selects $\heartsuit\psi_0$ to be tracked.

The priority function α of A_χ is designed to ensure that a run ρ – that is, a sequence of formulae – is accepting iff a least fixpoint formula ψ is unfolded infinitely often on ρ without being dominated by any outer fixpoint formula ϕ, i.e. one with $\mathsf{ad}(\phi) > \mathsf{ad}(\psi)$. Here, we use the term *dominated* to indicate both the greater alternation depth of ϕ and the fact that ϕ is also unfolded infinitely often. As indicated above, the model checking game will relate closely to non-deterministic tracking, and the proof of its correctness (Theorem 3.15) will clarify that alternation depth indeed provides an adequate mechanism to detect which of two fixpoints is the inner one. For purposes of the nondeterministic tracking automaton, the alphabet Σ is in fact overlarge, and could be reduced to just individual choices picking a disjunct from a single disjunction; the importance of Σ arises in the (co-)determinization of A_χ, where it ensures that enough branching is retained.

Example 3.3 (Tracking automaton)**.** For an example of the tracking automaton construction, recall from Example 2.1.2 the monotone μ-calculus formula $\chi = \nu X. (a \wedge \mu Y. (X \vee \langle g\rangle Y))$ obtained from the game logic formula $\langle (g^*)^\times\rangle a$. As Fischer-Ladner closure of χ, we have

$$\mathbf{F} = \{\chi,\ a \wedge \phi,\ a,\ \phi,\ \chi \vee \langle g\rangle\phi,\ \langle g\rangle\phi\}$$

with ϕ abbreviating the formula $\mu Y. (\chi \vee \langle g\rangle Y)$. Furthermore, we have $\mathsf{ad}(X) = 2$ and $\mathsf{ad}(Y) = 1$. For this small example we have $\mathbf{F}_\vee = \{\chi \vee \langle g\rangle\phi\}$, so that $\mathsf{choices}$ consists of just the two functions τ_l and τ_r, defined by $\tau_l(\chi \vee \langle g\rangle\phi) = \chi$ and $\tau_r(\chi \vee \langle g\rangle\phi) = \langle g\rangle\phi$, respectively. Omitting the treatment of propositional atoms as modal operators, we have $\mathbf{F} \cap \Lambda(\mathbf{F}) = \{\langle g\rangle\phi\}$ so that $\mathsf{selections} = \{\emptyset, \{\langle g\rangle\phi\}\}$. Then we obtain the tracking automaton A_χ depicted below, where τ stands for any letter from $\mathsf{choices}$ and $\kappa_{\langle g\rangle\phi}$ denotes the letter $\{\langle g\rangle\phi\} \in \mathsf{selections}$.



Thus A_χ accepts infinite words over $\mathsf{choices} \cup \mathsf{selections}$ that branch to the left on the disjunction $\chi \vee \langle g\rangle\phi$ only finitely often, and on which the automaton can infinitely often read the letter $\{\langle g\rangle\phi\}$ in the node $\langle g\rangle\phi$ (and avoid reading letters from $\mathsf{selections}$ at any other node). Such words have a run in which the maximal priority that is visited infinitely often is 2; as intended, such words encode situations where the least fixpoint ϕ is unfolded infinitely often while the outer greatest fixpoint χ is unfolded only finitely often. We point out that the fixpoint variable X is unguarded in χ; this induces the left cycle in A_χ, on which no letter from $\mathsf{selections}$ is ever read.

Remark 3.4. The above definition of tracking automata deviates from the one we used in earlier work [HS19] in two respects: We now attach priorities to the states of the automaton rather than to its transitions; and we have changed the propositional part of the alphabet in such a way that every top-level propositional or fixpoint operator is processed when a propositional letter is read (while we previously had a separate letter for every conjunction, disjunction, and fixpoint literal in $\mathbf{F}$). Both choices serve mainly to ease and clarify the presentation.

The non-deterministic parity automaton A_χ introduced above has size at most n_0 and priorities at most 1 to $k+1$. In order to use $L(\mathsf{A}_\chi)$ as an objective in our upcoming satisfiability games, we require a deterministic automaton accepting this language. To this end, we use a standard construction (e.g. [KKV01]) to transform A_χ into an equivalent Büchi automaton of size n_0k' (which has additional states (ϕ, i) where $\phi \in \mathbf{F}$ and $1 \leq i \leq k+1$ is even) where

$$k' = \lfloor (k+1)/2 \rfloor + 1 \in \mathcal{O}(k). \tag{3.1}$$

Then we determinize the Büchi automaton using, e.g., the Safra/Piterman-construction [Saf88, Pit07] and obtain an equivalent deterministic parity automaton with $2n_0k'$ priorities and size $\mathcal{O}(((n_0k')!)^2)$. Alternatively, direct determinization from parity automata to parity automata [SV14] can be used. Finally we complement the obtained automaton by decreasing every priority by 1, obtaining a deterministic parity automaton

$$\mathsf{B}_\chi = (D_\chi, \Sigma, \delta, q_{\mathsf{init}}, \Omega) \tag{3.2}$$

with priorities 0 to $2n_0k' - 1$ and of size $\mathcal{O}(((n_0k')!)^2)$ such that $L(\mathsf{B}_\chi) = \overline{L(\mathsf{A}_\chi)}$, i.e. B_χ is a deterministic parity automaton that accepts the words that encode sequences of fixpoint unfoldings without infinite deferral of least fixpoints.

We refer to B_χ as the *co-determinized tracking automaton*. The states of B_χ are like macrostates in the standard powerset construction for finite-word automata, but instead of being mere sets of states, they organize the states of the original automaton into a tree structure. Due to the preceding conversion of A_χ into a Büchi automaton, the tree nodes are labelled with sets of pairs consisting of a formula in $\mathbf{F}$ and a priority. We define a labelling function

$$l^A \colon D_\chi \to \mathcal{P}(\mathbf{F})$$

that maps each state q of B_χ (e.g. a Safra tree) to the set of formulae that occur in q. That is, the labelling function forgets the structuring of the set of formulae in macrostates (in fact, for compact Safra trees [Pit07], $l^A(q)$ can be obtained from the label of the root of q by forgetting the priorities). We do not need to know anything about how determinization works, except the following fact: in all (history-)determinization procedures that we refer to in this work, labels in the above sense evolve under transitions like macrostates in the standard powerset construction, i.e. we have

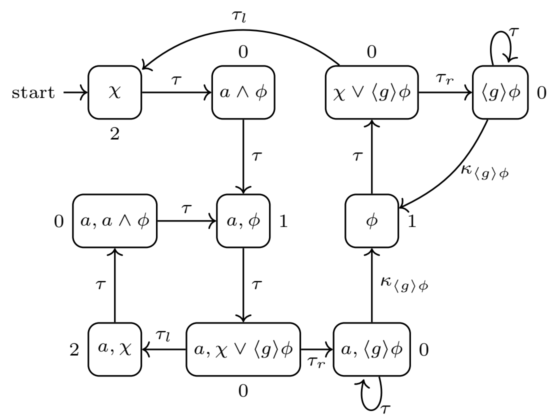
$$l^A(\delta(q,\sigma)) = \bigcup\{\Delta(\psi,\sigma) \mid \psi \in l^A(q)\} \qquad \text{for } q \in D_\chi,\ \sigma \in \Sigma,\ \psi \in \mathbf{F}. \tag{3.3}$$

In particular,

$$l^A(\delta(q,\kappa)) = \{\psi \mid \heartsuit\psi \in \kappa \cap l^A(q)\} \qquad \text{for } q \in D_\chi,\ \kappa \in \mathsf{selections}. \tag{3.4}$$

Note also that when B_χ reads a choice function in node q, then all top-level propositional operators and fixpoints in $l^A(q)$ are processed in parallel, i.e. one disjunct is picked from each disjunction, every conjunction is decomposed into its conjuncts, and every fixpoint is unfolded.

Example 3.5 (Co-determinized tracking automaton)**.** To give a flavour of the co-determinized tracking automaton, we go back to the non-deterministic automaton A_χ for the monotone μ-calculus formula $\chi = \nu X.\,(a \wedge \mu Y.\,(X \vee \langle g\rangle Y))$ as detailed in Example 3.3. Applying the above-mentioned method of co-determinization via an intermediate Büchi automaton and Safra/Piterman trees yields an automaton that is too large for presentation here. Instead we use as B_χ the equivalent but minimized automaton given below. For readability, the diagram omits an accepting sink state to which all but two modal transitions lead; the only modal transitions that do not lead to this sink state are the two $\kappa_{\langle g\rangle\phi}$-transitions that are explicitly shown.



Nodes in this automaton are labelled with sets of formulae as shown, according to the labelling function $l^A : D_\chi \to \mathcal{P}(\mathbf{F})$ (however, we emphasize that a node need not be uniquely determined by its label, even though this happens to be the case in the example). For instance, let q denote the node in the lower left corner of the automaton; then $l^A(q) = \{a, \chi\}$. Acceptance is dual to the tracking automaton A_χ. That is, an infinite word w is accepted by B_χ if either a) w picks the left disjunct from $\chi \vee \langle g\rangle\phi$ infinitely often, ensuring that the left part of the automaton, and thereby also a node with priority 2, is visited infinitely often; or b) w contains only finitely many letters from $\mathsf{selections}$ (so the automaton eventually loops forever at the bottom right node); or c) w contains a letter from $\mathsf{selections}$ at a position such that B_χ is, after reading the word up to this position, in a node that does not contain $\langle g\rangle\phi$ in its label (so the run ends up in the accepting sink state).

Remark 3.6. It has been noted that for the relational μ-calculus, tracking automata for *aconjunctive* formulae are *limit-deterministic* parity automata [HSD18]. These considerably simpler automata can be determinized to deterministic parity automata of size $\mathcal{O}((n_0k')!)$ and with $2n_0k'$ priorities [EKRS17, HSD18] (with k' as per (3.1)), an observation that can also be used for the tracking automata in this work. For aconjunctive formulae, one thus has a correspondingly improved bound on the runtime of our satisfiability checking algorithm than stated for the general case in Lemma 6.4 below.

It has also been shown that tracking automata for *guarded* and *alternation-free* formulae can be seen as co-Büchi automata [FLL13] so that the simpler determinization procedure for co-Büchi automata [MH84] can be used for such formulae, and the resulting satisfiability games have a Büchi objective rather than the more involved parity objective required in the general case. However, in our setting the tracking automata have to correctly deal with unguarded fixpoint variables and hence assign priorities greater than 1 exclusively to fixpoint formulae (for instance priority 2 is assigned to formulae of the shape $\mu X.(\psi \vee \Diamond X)$ but priority 1 is assigned to formulae of the shape $\psi \vee \Diamond(\mu X.(\psi \vee \Diamond X))$ even though the latter formula arises by unfolding the fixpoint in the former formula) so that our tracking automata are not immediately co-Büchi automata when the target formula is alternation-free. Having said that, it indeed is possible to use co-Büchi automata for *unguarded* alternation-free formulae, namely by separating local and global tracking and using two different automata, one being a reachability automaton used for detecting infinite local unfolding of unguarded least fixpoint formulae and the other being a co-Büchi automaton used for detecting infinite deferral of guarded least fixpoints; this method has been described in [EJ99]. We refrain from introducing this more involved setup here, so for the moment, our results do not immediately allow the use of co-Büchi methods for unguarded alternation-free formulae.

Remark 3.7. *History-deterministic* automata [HP06] allow a limited amount of nondeterminism but still can easily be complemented and combined with a game arena to obtain a game (and hence have been introduced under the name *good-for-games automata*). They allow nondeterministic transitions under the condition that all nondeterministic choices in an accepting run can be resolved by looking only at the history of the run so far. Intuitively, the non-determinism in history-deterministic automata cannot make guesses about the future of runs. It follows from the results of Henzinger and Piterman [HP06] that instead of full determinization of A_χ, it suffices to turn A_χ into an equivalent history-deterministic automaton, which then can be complemented and used instead of B_χ in the subsequent development. For general formulae, A_χ is a parity automaton and can be history-determinized by first transforming to a Büchi automaton and then using the method for history-determinization of Büchi automata described in [HP06]. This method is conceptually simpler than full determinization of Büchi automata by the Safra-Piterman construction and avoids constructing Safra-trees, even though it does not reduce the number of states in the obtained automaton in comparison to full determinization.

Remark 3.8 (Tracking automata for unguarded formulae)**.** A tableau-based method for deciding satisfiability of unguarded formulae of the relational μ-calculus has been introduced by Friedmann and Lange [FL13]. The method augments states in the tracking automaton with an additional bit indicating *activity* of formulae (meaning that the respective formula has been manipulated by the last transition of the tracking automaton), doubling the size of the tracking automaton. The acceptance condition of the tracking automaton then is modified in order to accept only such branches that contain some trace that is both infinitely often active and on which some least fixpoint is unfolded infinitely often without being dominated. As the propositional tableau rules in [FL13] manipulate one propositional formula at a time, one also needs to introduce an additional tableau rule in order to ensure fairness of unfolding of unguarded fixpoint formulae.

In the current work we define propositional transitions of the tracking automaton A_χ in such a way that all propositional formulae are processed whenever a single propositional letter $\tau \in \mathsf{choices}$ is read. Hence fairness of fixpoint unfolding is inherent to our method. Moreover, the only inactive transitions (i.e. transitions that do not manipulate the tracked formula) in A_χ are transitions of the shape $(\heartsuit\psi, \tau, \heartsuit\psi)$ for some modal literal $\heartsuit\psi$ and $\tau \in \mathsf{selections}$. Since $\alpha(\heartsuit\psi) = 1$, all accepting runs of A_χ are active by construction. Thus our method readily handles unguarded formulae without requiring an activity bit.

Model Checking Games. It will be technically convenient to use a game characterization of μ-calculus semantics. Recall that parity games are infinite-duration two-player games, played by the *existential* and the *universal* player (also referred to as players $\exists$ and $\forall$). A parity game is given by a tuple

$$\mathsf{G} = (V_\forall, V_\exists, E, v_0, \Omega)$$

where $V = V_\forall \cup V_\exists$ is a set of *positions*, partitioned disjointly into the set $V_\exists$ of positions owned by the existential player and the set $V_\forall = V \setminus V_\exists$ of positions owned by the universal player; $E \subseteq V \times V$ is the set of *moves*; v_0 is an *initial position* and $\Omega : V \to \mathbb{N}$ is the *priority function*, which assigns priorities $\Omega(v) \in \mathbb{N}$ to states $v \in V$. We write $E(v) = \{v' \in V \mid (v, v') \in E\}$ for $v \in V$. A *path* through (V, E) is a finite or infinite sequence $v_0, v_1, \ldots$ such that $v_i \in E(v_{i-1})$ for all $i > 0$. A *play* is a *maximal* path $\pi = v_0, v_1, \ldots$ through (V, E), i.e. π is either infinite or ends in a position v such that $E(v) = \emptyset$. A play π is *winning* for the existential player if it is finite and ends with the universal player being stuck, or if it is infinite and the highest priority that is visited infinitely often on π is even; otherwise, π is winning for the universal player. Formally, π is *winning* for the existential player if either $\pi = v_0, v_1, \ldots, v_n$ is finite and $v_n \in V_\forall$, or π is infinite and $\max(\mathsf{Inf}(\Omega \circ \pi))$ is even. A (history-dependent) *strategy* for the existential player is a partial function $s : V^* V_\exists \rightharpoonup V$ such that $(v_n, s(v_0, v_1, \ldots, v_n)) \in E$ for all partial plays $v_0, v_1, \ldots, v_n \in V^* V_\exists$ such that $E(v_n) \neq \emptyset$; for positions $v_n \in V_\exists$ such that $E(v_n) = \emptyset$, s is undefined on all inputs $v_0, v_1, \ldots, v_n \in V^* V_\exists$. A play $\pi = v_0, v_1, \ldots$ is *compatible* with (or *follows*) a strategy s for the existential player if $v_{n+1} = s(v_0, \ldots, v_n)$ for all i such that $v_n \in V_\exists$ and v_n is not the last position in π. A strategy s is *history-free* if $s(v_0, \ldots, v_{n-1}, v_n)$ depends only on v_n; then s is a partial function from $V_\exists$ to V. The existential player wins a position $v \in V$ if there is a strategy s such that every play that starts at v and is compatible with s is won by the existential player; similar notions of (winning) strategies for the universal player are defined dually. The game G is won by the player that wins v_0.

Lemma 3.9 (History-free determinacy [Mar75, EJ91])**.** *Parity games are history-free determined, that is, every position is won by (exactly) one of the two players, and then there is a history-free strategy that wins the position for the respective player.*

Given a parity game G with set V of positions and a set $W \subseteq V$, we let $\mathsf{win}^\exists_W$ and $\mathsf{win}^\forall_W$ denote the set of positions for which the respective player has a winning strategy in G such that every play that is compatible with the strategy remains within W. Positions $v \in W$ such that $v \notin \mathsf{win}^\exists_W \cup \mathsf{win}^\forall_W$ are undetermined (w.r.t. W); for such v, neither player has a strategy that wins v while staying in W. As parity games are determined, we always have $V = \mathsf{win}^\exists_V \cup \mathsf{win}^\forall_V$. We write $\mathsf{win}^\exists$ for $\mathsf{win}^\exists_V$ and $\mathsf{win}^\forall$ for $\mathsf{win}^\forall_V$.

Winning regions in parity games are clearly invariant under bisimulation (e.g. [DG08, CKW18]). We need only invariance under functional bisimulations. Explicitly, given parity games $\mathsf{G} = (V_\forall, V_\exists, E, \Omega)$, $\mathsf{G}' = (V'_\forall, V'_\exists, E', \Omega)$ (we elide initial positions), a *bounded morphism* $f \colon \mathsf{G} \to \mathsf{G}'$ is a map $f \colon V \to V'$ whose graph is a bisimulation (see also the definition for Kripke frames in Section 2); that is: f preserves position ownership and priorities, and (i) whenever $(v, v') \in E$, then $(f(v), f(v')) \in E'$; and (ii) whenever $(f(v), u') \in E'$, then there exists $(v, v') \in E$ such that $f(v') = u'$. The mentioned invariance then takes the following shape:

Lemma 3.10 (Invariance of parity games under bounded morphisms)**.** *Let* $\mathsf{G} = (V_\forall, V_\exists, E, \Omega)$, $\mathsf{G}' = (V'_\forall, V'_\exists, E', \Omega)$ *be parity games, let* $v \in V$*, and let* $f \colon \mathsf{G} \to \mathsf{G}'$ *be a bounded morphism. Then* $\exists$ *wins position* v *in* G *iff* $\exists$ *wins* $f(v)$ *in* G'.

We next provide a parity game characterization of formula satisfaction. This characterization highlights the close relationship between satisfaction of fixpoints and (non-)acceptance of runs in the tracking automaton A_χ (a nondeterministic parity automaton). Since the satisfiability game that we introduce in Section 6 will be based on the co-determinized tracking automaton B_χ, this relationship will be key in the correctness proof of the satisfiability game, specifically in the proof of the truth lemma (Lemma 4.12) and in the soundness proof (Lemma 5.4).

Definition 3.11 (Model checking games)**.** Given a coalgebra (C,ξ), the *model checking game* $\mathsf{G}_{\chi,(C,\xi)} = (V_\exists, V_\forall, E, \Omega)$ for χ over (C,ξ) is a parity game with sets of positions $V = V_\exists \cup V_\forall$ defined by

$$V_\exists = C \times \mathbf{F}_\exists \qquad\qquad V_\forall = (C \times \mathbf{F}_\forall) \cup (\mathcal{P}(C) \times \mathbf{F}).$$

Here, $\mathbf{F}_\exists$ consists of those formulae in $\mathbf{F}$ that are disjunctions, modal literals, fixpoint literals, or $\bot$, while $\mathbf{F}_\forall = \mathbf{F} \setminus \mathbf{F}_\exists$ consists of those formulae in $\mathbf{F}$ that are conjunctions or $\top$. The moves and priorities in the game are given by the following table (the ownership of positions is already defined, and mentioned in the table only for readability)

position	owner	set of allowed moves	priority
(x,ψ)	$\exists/\forall$	$\{(x,\phi) \in \{x\} \times \Delta(\psi,\tau) \mid \tau \in \mathsf{choices}\}$	$\alpha(\psi) - 1$
$(x,\heartsuit\psi)$	$\exists$	$\{(D,\psi) \in \mathcal{P}(C) \times \Delta(\heartsuit\psi,\kappa) \mid \kappa \in \mathsf{selections}, \xi(x) \in [\![\heartsuit]\!]D\}$	0
(D,ψ)	$\forall$	$\{(x,\psi) \mid x \in D\}$	0

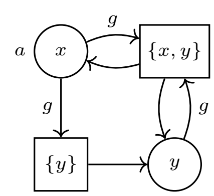
In the above table, the moves available to the players have been formulated in such a way that the mentioned relationship to the tracking automaton $\mathsf{A}_\chi = (\mathbf{F}, \Sigma, \Delta, q_{\mathsf{init}}, \alpha)$ becomes clear. In more detail, given an infinite play π in $\mathsf{G}_{\chi,(C,\xi)}$, the sequence of formulae ψ encountered on positions of the form (x,ψ) is a run ρ of A_χ on a word $w \in \Sigma^\omega$ extracted from π in an obvious manner (specifically, a move from (x,ψ) to (x,ϕ) where $\phi \in \Delta(\psi,\tau)$ adds τ to w, and a move from $(x,\heartsuit\psi)$ to (D,ψ) adds some $\kappa \in \mathsf{selections}$ such that $\heartsuit\psi \in \kappa$ to w). Even though w is not uniquely determined by π, we nevertheless refer to w as the word *induced* by π. Then, π is won by $\exists$ iff ρ is *non-accepting*.

The moves from states of the form (x,ψ) are more explicitly described as follows, depending on the shape of ψ. The existential player can move from $(x,\psi_1 \vee \psi_2)$ to (x,ψ_i) for any $i \in \{1,2\}$; each such move is witnessed by any $\tau \in \mathsf{choices}$ such that $\tau(\psi_1 \vee \psi_2) = \psi_i$. The universal player can move from $(x,\psi_1 \wedge \psi_2)$ to (x,ψ_i) for any $i \in \{1,2\}$. For fixpoint literals $\psi = \eta X.\,\phi$, the existential player moves from (x,ψ) to $(x,\phi[\psi/X])$; ownership of the position is purely formal in this case. For $\heartsuit\psi \in \mathbf{F}$, the existential player can move from $(x,\heartsuit\psi)$ to (D,ψ) for any set $D \subseteq C$ such that $\xi(x) \in [\![\heartsuit]\!]D$; each such move is witnessed by any $\kappa \in \mathsf{selections}$ such that $\heartsuit\psi \in \kappa$, hence $\Delta(\heartsuit\psi,\kappa) = \{\psi\}$. The universal player in turn can challenge satisfaction of ψ at any state $x \in D$ contained in the set D provided by the existential player by moving from (D,ψ) to (x,ψ). The definition of Ω implies that the only positions with non-zero priority are those of the form $(x,\eta X.\,\psi)$, with $\Omega(x,\eta X.\,\psi)$ being even if $\eta = \nu$, and odd if $\eta = \mu$.

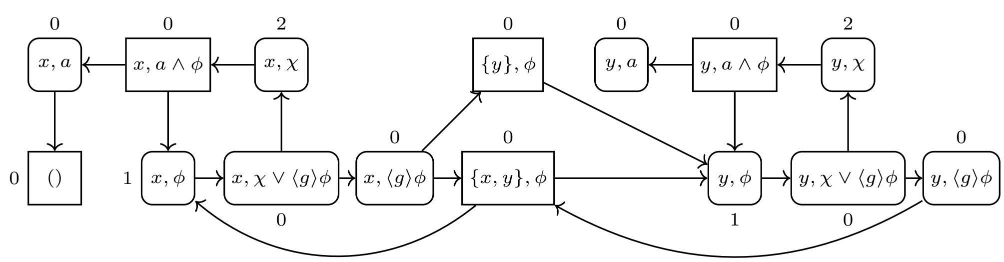
For technical purposes, we introduce a variant of the model checking game, the *subformula model checking game* $\mathsf{G}^{\mathsf{sub}}_{\chi,(C,\xi)}$ (the technical advantage of subformulae is that they allow for proofs by structural induction, a principle that we will employ in the correctness proof). The positions of $\mathsf{G}^{\mathsf{sub}}_{\chi,(C,\xi)}$ have the same shape as those of $\mathsf{G}_{\chi,(C,\xi)}$ except that subformulae occur in positions of $\mathsf{G}^{\mathsf{sub}}_{\chi,(C,\xi)}$ wherever elements of $\mathbf{F}$ occur in positions of $\mathsf{G}_{\chi,(C,\xi)}$. The ownership, priorities, and outgoing moves of positions are defined in $\mathsf{G}^{\mathsf{sub}}_{\chi,(C,\xi)}$ in the same way as in $\mathsf{G}_{\chi,(C,\xi)}$; in particular, for positions of the form (x,ψ), these data are defined by case distinction on the outermost connective of ψ like in $\mathsf{G}_{\chi,(C,\xi)}$, except for the following provisos: Given a variable X such that $\theta(X) = \eta X.\,\psi$, positions of the shape $(x,\eta X.\,\psi)$ or (x,X) receive priority $\Omega(x,\theta^*(X)) = \alpha(\theta^*(X)) - 1$ and belong to $\exists$, who has only one move, to (x,ψ).

Remark 3.12 (Higher-arity modalities)**.** This is the one place in the technical development where a comment is in order on how precisely the treatment of higher-arity modalities works: A *modal tracker* (ψ, i) consists of a formula $\psi = \heartsuit(\psi_1, \dots, \psi_n) \in \mathbf{F}$, where $\heartsuit \in \Lambda$ is an n-ary modality, and an index $i \in \{1, \dots, n\}$ indicating which argument position will be tracked; selections are then sets κ of modal trackers. The transition relation Δ of the tracking automaton A_χ is given on such ψ, κ by $\Delta(\psi, \kappa) = \{\psi_i \mid (\psi, i) \in \kappa\}$, i.e. the selection of arguments to be tracked introduces additional nondeterminism. In the model checking game $\mathsf{G}_{\chi,(C,\xi)}$, $\exists$ can move from (x, ψ) to any position of the form $((D_1, \psi_1), \dots, (D_n, \psi_n))$ (again of priority 0) such that $D_1, \dots, D_n \subseteq C$ and $\xi(x) \in [\![\heartsuit]\!]_X(D_1, \dots, D_n)$; that is, $\exists$ must provide a set of states for each argument of $\heartsuit$ in ψ. From $((D_1, \psi_1), \dots, (D_n, \psi_n))$, $\forall$ can then move to (y, ψ_i) if $y \in D_i$. In the correspondence between plays in $\mathsf{G}_{\chi,(C,\xi)}$ and runs of A_χ, the two subsequent moves from (x, ψ) to (y, ψ_i) then contribute a letter κ such that $(\psi, i) \in \kappa$ to the induced word w.

Example 3.13 (Model checking game)**.** We revisit the formula $\chi = \nu X. (a \wedge \mu Y. (X \vee \langle g \rangle Y))$ from Example 3.3 (the translation of the game logic formula $\langle (g^*)^\times \rangle a$), aiming to check its satisfaction over the neighbourhood model (C, ξ) shown below (with g assumed to be the only atomic game). States are depicted by circles, and neighbourhoods (that is, sets of states) are depicted by rectangles.



For instance, at x, Angel can force the game g into y (corresponding to $\{y\}$ being a g-neighbourhood at x), while Angel cannot influence what happens at y (whose only g-neighbourhood is the whole set $\{x, y\}$). Intuitively, the formula χ is satisfied at x since Angel can completely avoid playing g by stopping g^* immediately and satisfy a at x whenever, in $(g^*)^\times$, Demon decides to play g^* one more time. On the other hand, the formula is not satisfied at y since that state does not satisfy a and Angel cannot force the game out of y. The corresponding model checking game $\mathsf{G}_{\chi,(C,\xi)}$ is as follows, with rounded nodes belonging to $\exists$ and rectangles belonging to $\forall$; again, ϕ abbreviates the formula $\mu Y. (\chi \vee \langle g \rangle Y)$.



The game essentially consists of two copies of the automaton A_χ from Example 3.3; priorities in the game are also inherited from A_χ. The two copies of the automaton are linked by the positions $(\{y\}, \phi)$ and $(\{x, y\}, \phi)$ that encode the evaluation of the modality $\langle g \rangle \phi$. By definition of the model, we have $\xi(x)(g) = \{\{x, y\}, \{y\}\}$ and $\xi(y)(g) = \{\{x, y\}\}$. Recalling the predicate lifting $[\![\langle g \rangle]\!]$ for the monotone diamond from Example 2.1.2, we thus have $\xi(x) \in [\![\langle g \rangle]\!](D)$ for both $D = \{x, y\}$ and $D = \{y\}$ but $\xi(y) \in [\![\langle g \rangle]\!](D)$ only for $D = \{x, y\}$;

the moves to the central positions $(\{x, y\}, \phi)$ and $(\{y\}, \phi)$ are induced accordingly. The model checking game treats the propositional atom a as a nullary modality as per Remark 3.12: From positions of shape (z, a), player $\exists$ can move to $()$ (a 0-tuple of set/formula pairs), provided that a is satisfied at z. If this holds, then $\exists$ wins because $\forall$ has no moves at $()$; otherwise, $\exists$ loses, being stuck at (z, a). In the example, $\exists$ correspondingly wins (x, a) but loses (y, a).

Player $\exists$ wins the left-most five positions in this game by the strategy that always moves from position $(x, \chi \vee \langle g\rangle\phi)$ to position (x, χ); this enforces that plays of the game that start in one of these positions either get stuck in the position $(\emptyset, a)$ (which belongs to player $\forall$ but has no outgoing transitions and hence is won by player $\exists$), or forever follow the cycle through position (x, χ) and thereby visit priority 2 infinitely often. By the correctness of model checking games, this shows that x satisfies all formulae from $\mathbf{F}$ except $\langle g\rangle\phi$. All other positions in the game are won by player $\forall$ using the strategy that always moves from $(\{x, y\}, \phi)$ to (y, ϕ) and from $(y, a \wedge \phi)$ to (y, a); with this strategy, player $\forall$ can enforce that plays either get stuck in position (y, a) (lost by player $\exists$) or eventually take the bottom-right cycle forever, seeing priority 1 infinitely often. This shows that none of the formulae from $\mathbf{F}$ are satisfied at y.

We note that the two versions of the model checking games are bisimilar, and hence equivalent (Lemma 3.10):

Lemma 3.14. *The assignment $f(x, \psi) = (x, \theta^*(\psi))$, $f(D, \psi) = (D, \theta^*(\psi))$ defines a bounded morphism $f\colon \mathsf{G}^{\mathsf{sub}}_{\chi,(C,\xi)} \to \mathsf{G}_{\chi,(C,\xi)}$.*

Proof. It is clear that f preserves priorities and position ownership. We check the conditions on moves, restricting attention to the only cases where the games differ appreciably, viz., fixpoint literals and fixpoint variables. At such positions, however, the outgoing moves are uniquely determined in both games, so we just have to show that f preserves these moves.

So let X be a fixpoint variable, with $\theta(X) = \eta X.\, \psi$. The unique move from both (x, X) and $(x, \eta X.\, \psi)$ in $\mathsf{G}^{\mathsf{sub}}_{\chi,(C,\xi)}$ leads to (x, ψ). Since $\theta^*(X) = \theta^*(\eta X.\psi)$, (x, X) and $(x, \eta X.\, \psi)$ are mapped to the same position under f, and this position has the form $(x, \eta X.\psi')$ where ψ' is obtained from ψ by successively substituting free fixpoint variables Y with $\theta(Y)$ innermost first, but skipping the actual innermost variable X in ψ; so $\theta^*(\psi) = \theta^*(\psi[\eta X.\psi/X]) = \psi'[\eta X.\psi'/X])$. From $(x, \eta X.\psi')$, $\exists$'s unique move in $\mathsf{G}_{\chi,(C,\xi)}$ thus leads to $(x, \psi'[\eta X.\psi'/X]) = (x, \theta^*(\psi)) = f(x, \psi)$, as required. □

The model checking game $\mathsf{G}_{\chi,(C,\xi)}$ is very similar to the one considered by Cîrstea et al. [CKP+11b], one notable difference being that we do not assume guardedness. On the other hand, the subformula model checking game $\mathsf{G}^{\mathsf{sub}}_{\chi,(C,\xi)}$ resembles a game that was considered by Venema [Ven06] but which, again, assumes guardedness and moreover works with a version of the coalgebraic μ-calculus based on the coalgebraic cover modality [Mos99] instead of on predicate liftings. Indeed, the proof of correctness given by Cîrstea et al. is largely by reference to Venema's proof, an argument that is formally justified by our Lemma 3.14. Due to the mentioned guardedness issue, we opt to present a full correctness proof of our game, which largely follows the one given by Venema in that it makes do without (ordinal) timeouts as frequently used in the literature on the relational μ-calculus [SE89, NW96, BW18].

Theorem 3.15 (Correctness of the model checking game)**.** *Given a coalgebra (C, ξ), a state $x \in C$ and a formula $\psi \in \mathbf{F}$, we have $x \models \psi$ if and only if the existential player wins the position (x, ψ) in $\mathsf{G}_{\chi,(C,\xi)}$.*

Proof. We note first that the positions reachable from (x, ψ) in $\mathsf{G}_{\chi,(C,\xi)}$ are the same as in $\mathsf{G}_{\psi,(C,\xi)}$, so we can assume w.l.o.g. that ψ is the target formula, that is, $\psi = \chi$. By Lemmas 3.10 and 3.14, we can then replace $\mathsf{G}_{\chi,(C,\xi)}$ with $\mathsf{G}^{\mathsf{sub}}_{\chi,(C,\xi)}$, since $\theta^*(\chi) = \chi$. We will use structural induction on χ, and hence need to drop, only for purposes of this proof, the assumption that χ is closed. Of course, the game is then played over a pair $((C, \xi), i)$ where $i\colon \mathbf{V} \rightharpoonup \mathcal{P}(C)$ is a valuation of the fixpoint variables such that $i(X)$ is defined for all $X \in \mathsf{FV}(\chi)$; we correspondingly write $\mathsf{G}^{\mathsf{sub}}_{\chi,i}$ for the generalized game, eliding mention of (C, ξ) which remains unchanged throughout. We thus have new positions of the shape (x, X) or $(x, \neg X)$, which receive priority 0 and no outgoing moves. The ownership of these positions is defined by letting (x, X) be owned by $\exists$ if $x \notin i(X)$ and by $\forall$ otherwise, and correspondingly letting $(x, \neg X)$ be owned by $\exists$ if $x \in i(X)$ and by $\forall$ otherwise. Other than this, the game remains unchanged.

Next, for purposes of economizing one direction of the proof, we modify $\mathsf{G}^{\mathsf{sub}}_{\chi,i}$ to ensure symmetry between the players; we write $\mathsf{G}^{\mathsf{sub,sym}}_{\chi,i}$ for this symmetrized game. First, we reassign positions of the form $(x, \nu X.\,\psi)$ or (x, X), with X a ν-variable, to the universal player – since these positions have precisely one outgoing move, this change is clearly immaterial to how the game is played. Second, out of every pair $\heartsuit, \overline{\heartsuit}$ of dual modalities, we arbitrarily assign one to $\exists$ and the other to $\forall$, and we write $\heartsuit$ for modalities assigned to $\exists$, and $\overline{\heartsuit}$ for modalities assigned to $\forall$. Moreover, we rename the previous positions of the form (D, ψ) into $(D, \psi, \forall)$, and introduce additional positions $(D, \psi, \exists)$. Positions of the form $(x, \heartsuit\psi)$ still belong to $\exists$, with moves like before, into the renamed positions of the form $(D, \psi, \forall)$, which still belong to $\forall$. On the other hand, positions of the form $(x, \overline{\heartsuit}\psi)$ now belong to $\forall$, and $\forall$ can move to $(D, \psi, \exists)$ if $\xi(x) \notin [\![\heartsuit]\!]_C(\overline{D})$, where $\overline{D}$ denotes the complement of D in C. The new positions $(D, \psi, \exists)$ belong to $\exists$, who can move to (y, ψ) such that $y \in D$. The new sequences of moves $(x, \overline{\heartsuit}\psi) \xrightarrow{\forall} (D, \psi, \exists) \xrightarrow{\exists} (y, \psi)$ are, for purposes of $\exists$ winning the game, equivalent to the previous sequences $(x, \overline{\heartsuit}\psi) \xrightarrow{\exists} (D, \psi) \xrightarrow{\forall} (y, \psi)$: In either case, $\exists$ can force the game into a set U of positions of the form (y, ψ), necessarily of the form $U = D \times \{\psi\}$, iff $\xi(x) \in [\![\overline{\heartsuit}]\!]_C(D)$. To see this for the new moves $(x, \overline{\heartsuit}\psi) \xrightarrow{\forall} (D, \psi, \exists) \xrightarrow{\exists} (y, \psi)$, we reason as follows: $\exists$ can force the game into $D \times \{\psi\}$ iff $\forall$ cannot move to $(D', \psi, \exists)$ for any subset $D' \subseteq \overline{D}$ iff $\xi(x) \in [\![\heartsuit]\!]_C(\overline{D'})$ for all $D' \subseteq \overline{D}$ iff $\xi(x) \in [\![\overline{\heartsuit}]\!]_C(D)$, where the last step uses monotonicity of $\overline{\heartsuit}$. Thus, $\mathsf{G}^{\mathsf{sub}}_{\chi,i}$ and $\mathsf{G}^{\mathsf{sub,sym}}_{\chi,i}$ are equivalent in the sense that positions of the form (x, ψ) are won by the same player in either game.

By now, we have reduced the claim of the lemma to showing that in $\mathsf{G}^{\mathsf{sub,sym}}_{\chi,i}$,

(1) if $x \models \chi$, then $\exists$ wins (x, χ); and
(2) if $x \not\models \chi$, then $\forall$ wins (x, χ).

The symmetry of $\mathsf{G}^{\mathsf{sub,sym}}_{\chi,i}$ allows us to conclude (2) from (1), as follows: If $x \not\models \chi$, then $x \models \neg\chi$ (with $\neg\chi$ defined by taking negation normal forms as indicated in Section 2). By (1), $\exists$ wins the position $(x, \neg\chi)$ in the model checking game for $\neg\chi$. This game is now dual to the model checking game for χ in the sense that one is obtained from the other by swapping the positions of the players and dualizing the priorities (i.e. swapping priorities $2n$ and $2n - 1$); of course, positions $(x, \neg\psi)$ in the game for $\neg\chi$ correspond to positions (x, ψ) in the game for χ. We thus immediately obtain that $\forall$ wins (x, χ).

It remains to prove (1). We switch back to using the simpler game $\mathsf{G}^{\mathsf{sub}}_{\chi,i}$. As indicated above, we proceed by induction on χ; we treat i as universally quantified in the inductive claim. The cases for free fixpoint variables and Boolean operators ($\wedge$, $\vee$, $\top$, $\bot$) are trivial; we illustrate this on the case where $\chi = \chi_1 \wedge \chi_2$: If $x \in [\![\chi_1 \wedge \chi_2]\!]_i$, then $x \in [\![\chi_1]\!]_i$ and $x \in [\![\chi_2]\!]_i$. By induction, $\exists$ wins (x, χ_j) in $\mathsf{G}^{\mathsf{sub}}_{\chi_j,i}$ for $j = 1, 2$. In $\mathsf{G}^{\mathsf{sub}}_{\chi_1\wedge\chi_2,i}$, $\forall$ can move from (x, χ) to either (x, χ_1) or (x, χ_2), so $\exists$ wins by playing like in $\mathsf{G}^{\mathsf{sub}}_{\chi_1,,i}$ or $\mathsf{G}^{\mathsf{sub}}_{\chi_2,i}$, respectively. The remaining cases are as follows.

- $\chi = \heartsuit\chi_1$: By induction, $\exists$ wins by playing (D, χ_1) where $D = [\![\chi_1]\!]_i$.
- $\chi = \mu X.\, \chi_1$: It suffices to show that the set
$$W_1 := \{x \in C \mid \exists \text{ wins } (x, \chi) \text{ in } \mathsf{G}^{\mathsf{sub}}_{\chi,i}\} = \{x \in C \mid \exists \text{ wins } (x, \chi_1) \text{ in } \mathsf{G}^{\mathsf{sub}}_{\chi,i}\}$$
(where the second equality is immediate from the game rules) is a prefixpoint of the function defining $[\![\chi]\!]_i = [\![\mu X.\, \chi_1]\!]_i$ as a least prefixpoint, i.e. that
$$[\![\chi_1]\!]_{i'} \subseteq W_1 \quad \text{where } i' = i[X \mapsto W_1]. \tag{3.5}$$
So let $x \in [\![\chi_1]\!]_{i'}$. By induction, $\exists$ has a strategy s' in $\mathsf{G}^{\mathsf{sub}}_{\chi_1,i'}$ that wins (x, χ_1). We have to show that $\exists$ wins (x, χ_1) in $\mathsf{G}^{\mathsf{sub}}_{\chi,i}$, which differs from $\mathsf{G}^{\mathsf{sub}}_{\chi_1,i'}$ only in that it treats the fixpoint variable X as bound. The winning strategy works as follows:
  - From (x, χ_1), play according to s' until a position of the form (y, X) is encountered (if ever). This is possible because up to that point, there is no difference between $\mathsf{G}^{\mathsf{sub}}_{\chi,i}$ and $\mathsf{G}^{\mathsf{sub}}_{\chi_1,i'}$. If no position (y, X) is ever reached, then the play effectively takes place in $\mathsf{G}^{\mathsf{sub}}_{\chi_1,i'}$, and as such follows s'. It is thus won by $\exists$, since s' is winning.
  - If a position of the form (y, X) is reached, then $y \in i'(X) = W_1$, since s' is winning in $\mathsf{G}^{\mathsf{sub}}_{\chi_1,i'}$. The next position reached is (y, χ_1), so $\exists$ wins in $\mathsf{G}^{\mathsf{sub}}_{\chi_1,i}$ because $y \in W_1$.
- $\chi = \nu X.\, \chi_1$: Let $x \in [\![\chi]\!]_i$. We construct a strategy s that wins (x, χ) in $\mathsf{G}^{\mathsf{sub}}_{\chi,i}$ as follows. From (x, χ), the game proceeds to (x, χ_1). By fixpoint unfolding, we have
$$[\![\chi]\!]_i = [\![\chi_1]\!]_{i'} \quad \text{where } i' = i[x \mapsto [\![\chi]\!]_i]. \tag{3.6}$$
By induction, $\exists$ thus has a strategy s' that wins (x, χ_1) in $\mathsf{G}^{\mathsf{sub}}_{\chi_1,i'}$. We let s follow s' in $\mathsf{G}^{\mathsf{sub}}_{\chi,i}$ until a position of the form (y, X) is reached (exploiting like in the previous case that up to that point, the games $\mathsf{G}^{\mathsf{sub}}_{\chi_1,i'}$ and $\mathsf{G}^{\mathsf{sub}}_{\chi,i}$ do not differ). Since s' is winning in $\mathsf{G}^{\mathsf{sub}}_{\chi_1,i'}$, we then have $y \in i'(X) = [\![\chi]\!]_i = [\![\chi_1]\!]_{i'}$, so we can continue in the same manner after the game $\mathsf{G}^{\mathsf{sub}}_{\chi,i}$ automatically proceeds to (y, χ_1). To see that s is winning, we distinguish cases on a play π that follows s:
  - If from some point on, π no longer reaches positions of the form (y, X), then π has a suffix that is a winning play for $\exists$ from a position of the form (y, χ_1) in $\mathsf{G}^{\mathsf{sub}}_{\chi_1,i'}$, so $\exists$ wins π.
  - Otherwise, π infinitely often visits positions of the form (y, X). Thus, X is unfolded infinitely often. Intuitively speaking, since X is a ν-variable and $\nu X.\chi_1$ is the outermost fixpoint in the target formula (being the target formula itself), $\exists$ should therefore win the play according to the intention of the game, as long as the mechanism that replaces the direct comparison of inner vs. outer fixpoints (as used in the winning condition of the game considered by Venema [Ven06]) with the comparison of alternation depth works. Formally, we proceed as follows. We have to show that $\mathsf{ad}(Z) < \mathsf{ad}(X)$ for every μ-variable Z that is unfolded on π between two unfoldings of X (this implies $\alpha(\theta^*(Z)) < \alpha(\theta^*(X))$), so positions where Z is unfolded have lower priority than positions where X is unfolded). Let $Y_n, \dots, Y_1, Y_0 = X$ be the sequence of variables unfolded between two unfoldings of X, including the second (but not the first) unfolding of X

itself. We show by induction on $j \in \{0, \dots, n\}$ that for every j, there is a dependency chain, possibly of length 0, from Y_j to X (implying that $\mathsf{ad}(X) \geq \mathsf{ad}(Y_j)$, and that $\mathsf{ad}(X) > \mathsf{ad}(Y_j)$ if Y_j is a μ-variable). The induction base $j = 0$ is trivial. In the induction step for $j > 0$, we can assume that $Y_j \neq Y_k$ for $j > k$, since otherwise we are done by induction. The unfolding step from Y_j leads to a position of the form (y, ψ) where $\theta(Y_j) = \eta X.\psi$. Since a position (z, Y_{j-1}) is reached from (y, ψ) without interceding unfolding steps, Y_{j-1} is a subformula of ψ. If $Y_{j-1} \in \mathsf{FV}(\psi)$, then $Y_{j-1} \in \mathsf{FV}(\theta(Y_j))$ since $Y_{j-1} \neq Y_j$; that is, $Y_j \prec_{\mathsf{dep}} Y_{j-1}$, and we are done by induction. Otherwise, $\theta(Y_{j-1})$ is a subformula of ψ. By induction, there is a dependency chain $Y_{j-1} \prec_{\mathsf{dep}} Z \prec_{\mathsf{dep}} \dots \prec_{\mathsf{dep}} X$; in particular, $Z \in \mathsf{FV}(\theta(Y_{j-1}))$. If $Z = Y_j$, then we are done. Otherwise, $Z \in \mathsf{FV}(\theta(Y_j))$, i.e. $Y_j \prec_{\mathsf{dep}} Z$, and we are done. $\square$

Remark 3.16 (Fixpoint games)**.** By instantiation of the model checking game to a generalization of the monotone μ-calculus (Example 2.1.2), we obtain a general form of fixpoint games for monotone functions on powerset lattices, which in turn are an instance of fixpoint games over continuous lattices [BKMP19]. Details are as follows.

We need only the case without propositional atoms, whose mention we therefore elide in the following, and with only one atomic program that we keep implicit. On the other hand, we generalize to higher-arity neighbourhood frames and modalities: For $n \geq 0$, we define the *n-ary monotone neighbourhood functor* $\mathcal{M}_n$ (e.g. [SP10, MV15]) as taking a set X to the set of subsets of $(\mathcal{Q}X)^n$ that are upwards closed under componentwise subset inclusion. We use an n-ary modality $\Diamond$, which we interpret over $\mathcal{M}_n$ by the predicate lifting given by

$$[\![\Diamond]\!]_X(A_1, \dots, A_n) = \{\alpha \in \mathcal{M}_n X \mid (A_1, \dots, A_n) \in \alpha\}.$$

By transposition of arguments, a coalgebra $C \to \mathcal{M}_n C \subseteq \mathcal{Q}((\mathcal{Q}C)^n)$ can alternatively be seen as a map $g\colon (\mathcal{Q}C)^n \to \mathcal{Q}C$ that is monotone w.r.t. (componentwise) subset inclusion. Recall here that $\mathcal{Q}$ is the contravariant powerset functor; as we do not actually need the action on maps in the following, we will just write $\mathcal{P}(C)$ in place of $\mathcal{Q}C$. The semantics of a formula $\Diamond(\phi_1, \dots, \phi_n)$ in C under a valuation i is then equivalently given by $[\![\Diamond\phi]\!]_i = g([\![\phi_1]\!]_i, \dots, [\![\phi_1]\!]_i)$. That is, we can just see the monotone μ-calculus as an expression language for nested fixpoints over (higher-arity) monotone functions on $\mathcal{P}(C)$. In the corresponding instance of the model checking game on C, $\exists$ can move from a position $(x, \Diamond(\psi_1, \dots, \psi_n))$ to any tuple $((D_1, \psi_1), \dots, (D_n, \psi_n))$ such that $x \in g(D_1, \dots, D_n)$. We use these games in the fixpoint characterization of the satisfiability game (Lemma 6.3).

4. One-Step Satisfiability and Tableaux

In this section, we identify an embodiment of a model for χ in the shape of a subautomaton of the co-determinized tracking automaton B_χ that satisfies certain additional properties; we will use this concept as a stepping stone in the reduction of satisfiability checking to game solving, and, as usual, call such a witness for formula satisfaction a *tableau*. Specifically, such a subautomaton consists of those automaton nodes q for which there are states in the model that jointly satisfy all formulae from $l(q)$, and the automaton transitions in a tableau are required to witness satisfaction of those formulae; we formalize the structural property required for the satisfaction of modalities using the concept of *one-step satisfiability*. Then we show that every tableau carries a coalgebra structure that is *coherent* with its transitional structure and its labels; such coalgebras then satisfy a truth lemma implying satisfaction of the target formula. The proof of the truth lemma relies on the model checking game, and exploits that the latter relates closely to the nondeterministic tracking automaton A_χ.

We begin with considerations on the above-mentioned problem of *one-step satisfiability checking*, a functor-specific problem that in many instances can be solved in time singly exponential in $\mathsf{size}(\chi)$.

Definition 4.1 (One-step satisfiability problem [Sch07, SP08, MPS09])**.** Let V be a finite set of *propositional variables.* A *one-step pair* (γ, Θ) (over V) consists of a set $\Theta \subseteq \mathcal{P}(V)$ (understood as a disjunctive normal form over V, cf. Remark 4.2) and a set $\gamma \subseteq \Lambda(V)$, understood conjunctively, where we require that γ mentions every element of V precisely once. Correspondingly, we interpret $a \in V$ and γ over Θ by

$$\llbracket a \rrbracket_0^\Theta = \{u \in \Theta \mid a \in u\} \subseteq \Theta$$
$$\llbracket \gamma \rrbracket_1^\Theta = \textstyle\bigcap_{\heartsuit a \in \gamma} \llbracket \heartsuit \rrbracket_\Theta \llbracket a \rrbracket_0^\Theta \subseteq F\Theta.$$

We say that (γ, Θ) is *satisfiable* (over the functor F) if $\llbracket \gamma \rrbracket_1^\Theta \neq \emptyset$. The *strict one-step satisfiability problem* is to decide whether a given one-step pair (γ, Θ) is satisfiable; here, the qualification 'strict' refers to the measure of input size of the problem, which we take to be

$$\mathsf{size}(\gamma) := \sum_{\heartsuit a \in \gamma} (1 + \mathsf{size}(\heartsuit))$$

(in particular, $|\Theta|$, which may be exponential in $\mathsf{size}(\gamma)$, does not count towards the input size).

Remark 4.2 (One-step logic)**.** 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. In a more explicitly syntactic view, one will regard V as a set of propositional variables. One then sees that a one-step pair (γ, Θ) as above contains two layers: a purely propositional layer embodied in Θ, which postulates which propositional formulae over V are satisfiable (that is, we see $\Theta \subseteq \mathcal{P}(V)$ as a disjunctive normal form $\bigvee_{u \in \Theta}(\bigwedge_{a \in u} a \wedge \bigwedge_{a \in V \setminus u} \neg a)$); and a modal layer with nesting depth of modalities uniformly equal to 1, embodied in the set γ of modal literals, which specifies a constraint $\bigwedge_{\heartsuit a \in \gamma} \heartsuit a$ on an element of $F\Theta$. Under this perspective (and otherwise), it is trivial to note that satisfiability of a one-step pair (γ, Θ) is preserved under enlarging Θ, as this corresponds to weakening the propositional formula represented by Θ.

Example 4.3 (One-step satifiability)**.** We consider the one-step satisfiability problem for the logics from Example 2.1, omitting details on the (trivial) treatment of propositional atoms.

(1) For the *relational modal μ-calculus* (Example 2.1.1.), where $\Lambda = \{\Diamond, \Box\}$, the one-step satisfiability problem is to decide, for a given one-step pair (γ, Θ) over V, whether there is $A \in \llbracket \gamma \rrbracket_1^\Theta$, that is, a subset $A \in \mathcal{P}\Theta$ such that for each $\Diamond a \in \gamma$, there is $u \in A$ such that $a \in u$, and for each $\Box b \in \gamma$ and each $u \in A$, $b \in u$. Equivalently, one needs to check that for each $\Diamond a \in \gamma$ there is $u \in \Theta$ such that $a \in u$ and moreover $b \in u$ for all $\Box b \in \gamma$. To avoid quadratic complexity in $\mathsf{size}(\gamma)$, implement this check in two passes: In the first pass, go through all $\Box b \in \gamma$ and remove from Θ all u such that $b \notin u$; in the second pass, go through all $\Diamond a \in \gamma$ and check that there remains some u in Θ such that $a \in u$. Both passes can be done in time $\mathcal{O}(\mathsf{size}(\gamma) \cdot |V| \cdot |\Theta|) = \mathcal{O}(\mathsf{size}(\gamma) \cdot |V| \cdot 2^{|V|})$, showing that in this case the strict one-step satisfiability problem is in ExpTime. We note that this is all the work that will be required to instantiate our generic complexity bound (Theorem 6.5 below) to the relational μ-calculus, obtaining the known upper bound ExpTime for satisfiability checking [EJ99].

(2) For the *monotone modal μ-calculus* (Example 2.1.2.) with set $\mathcal{A}$ of atomic games, we have $\Lambda = \{\langle g\rangle, [g] \mid g \in \mathcal{A}\}$, again eliding propositional atoms for the sake of readability. It is an immediate property of the semantics of monotone modalities that in order to check that $[\![\gamma]\!]_1^\Theta \neq \emptyset$ for a given one-step pair (γ, Θ) over V, it suffices to check that whenever $\langle g\rangle a, [g]b \in \gamma$, then there is $u \in \Theta$ such that $a, b \in u$ [Var89, Proposition 3.8]. (Indeed, this criterion corresponds to the usual monotonicity rule – cf. [SP09, CKP11a] – under the correspondence between modal tableau rules and one-step satisfiability checking discussed in Remark 6.7 below.) This can clearly be done in time $\mathcal{O}(\mathsf{size}(\gamma)^2 \cdot |V| \cdot |\Theta|) = \mathcal{O}(\mathsf{size}(\gamma)^2 \cdot |V| \cdot 2^{|V|})$, showing that the strict one-step satisfiability problem for the monotone case is in ExpTime. We note that again this is all the work that is required to instantiate our generic complexity bound and obtain the known upper ExpTime bounds on satisfiability checking for game logic [PP03] and the monotone μ-calculus [CKP11a]; in fact, it appears that for the latter case, the result for the full unguarded logic is formally new (however, we note that it could alternatively be obtained by encoding the monotone μ-calculus into the relational μ-calculus in the same way as for game logic [Par83, PP03]).

(3) For the *graded μ-calculus* (Example 2.1.4.), the one-step satisfiability problem is to decide, for a one-step pair (γ, Θ), whether there is a multiset $\beta \in \mathcal{B}\Theta$ such that $\sum_{u\in\Theta\mid a\in u} \beta(u) > m$ for each $\langle m\rangle a \in \gamma$ and $\sum_{u\in\Theta\mid a\notin u} \beta(u) \leq m$ for each $[m]a \in \gamma$. The easiest way to see that the strict one-step satisfiability problem is in ExpTime is via a non-deterministic polynomial-space algorithm that goes through all $u \in \Theta$, guessing multiplicities $\beta(u) \in \{0, \dots, m+1\}$ where m is the greatest index of any diamond modality $\langle m\rangle$ that occurs in γ. This multiplicity is used to update $|V|$ counters that keep track of the total measure $\beta([\![a]\!]_0^\Theta)$ for $a \in V$; after updating the counters, the multiplicity is discarded, so that only polynomial space is used (for the counters). Once all multiplicities have been guessed, the algorithm verifies that $\beta \in [\![\gamma]\!]_1^\Theta$, using only the final counter values [KSV02, Lemma 1]. Essentially the same method works also for the *graded μ-calculus with polynomial inequalities* (Example 2.1.6) (and similar ideas have been used in work on Presburger modal logic [DL06]): In this setting, it still suffices to guess multiplicities up to $b+1$ where b is the largest index of any diamond modality $\langle p\rangle$ occurring in γ (indeed, if anything the bounds become smaller; e.g. for $\langle X_1^2 + X_2^2 - b\rangle$, it suffices to explore multiplicities up to $\lceil\sqrt{b}\rceil + 1$). Note that this argument does rely on the assumption that all coefficients of non-constant monomials are non-negative.

(4) For the *probabilistic μ-calculus with polynomial inequalities* (Example 2.1.5), we first observe, following [GJLS17, KPS22], that a small model property holds for the one-step logic: If a one-step pair (γ, Θ) over V is satisfiable, then $[\![\gamma]\!]_1^\Theta$ contains an element $(d, Q) \in \mathcal{D}\Theta \times \mathcal{P}(\mathsf{P})$ such that $d(u) > 0$ for only $|V|$-many $u \in \Theta$. This is seen as follows: Suppose that $(d_0, Q) \in [\![\gamma]\!]_1^\Theta$. Then for any $d \in \mathcal{D}\Theta$, to have $(d, Q) \in [\![\gamma]\!]_1^\Theta$ it suffices that $d([\![a]\!]_0^\Theta) = d_0([\![a]\!]_0^\Theta)$ for each $a \in V$. Since $d([\![a]\!]_0^\Theta) = \sum_{u\in[\![a]\!]_0^\Theta} d(u)$, this means that the numbers $y_u = d(u)$, for $u \in \Theta$, form a non-negative solution to the system of linear equations

$$\sum_{u\in[\![a]\!]_0^\Theta} y_u = d_0([\![a]\!]_0^u) \qquad \text{for } a \in V.$$

Since this system has a solution induced by d_0 itself, we obtain by the Carathéodory theorem (e.g. [Sch86]) that there is a solution with at most $|V|$ non-zero components. This allows us to solve the strict one-step satisfiability problem in non-deterministic polynomial space, and hence in exponential time, as follows: Guess $|V|$ elements $u \in \Theta$ that receive positive weight $d(u)$ in a solution d, and then check for satisfiability of the constraint on these $|V|$ real numbers that is embodied in γ. This constraint is a polynomially-sized system of polynomial inequalities, whose satisfiability can, by results of Canny [Can88], be checked in polynomial space.

Remark 4.4 (One-step polysize model property)**.** We say that the logic has the *one-step polysize model property (OSPMP)* if there is a polynomial p such that whenever a one-step pair (γ, Θ) over V is satisfiable, then $[\![\gamma]\!]_1^\Theta$ has an element of the form $Fi(t)$ where $i\colon \Theta_0 \to \Theta$ is the inclusion of a subset $\Theta_0 \subseteq \Theta$ such that $|\Theta_0| \leq p(|V|)$ [KPS22]. For instance, the arguments in Examples 4.3.1 and 4.3.4 show that the relational μ-calculus and the probabilistic μ-calculus (even with polynomial inequalities) both have the OSPMP. Similar arguments as for the probabilistic case show that the graded μ-calculus (even with polynomial inequalities) also has the OSPMP, using the integer Carathéodory theorem [ES06]; cf. [KPS22] for details. Our model construction below will establish that the OSPMP implies a polynomially branching model property (Remark 5.7).

Next, we present our notion of tableaux, which are partial subautomata of the co-determinized tracking automaton $\mathsf{B}_\chi = (D_\chi, \Sigma, \delta, q_{\mathsf{init}}, \Omega)$. We first fix some notation: To $\Xi \subseteq \mathsf{selections}$ and a node $q \in D_\chi$, we associate a one-step pair (γ_q, Θ_q^Ξ) over a set V_q of propositional variables, given by

$$\begin{aligned} V_q &= \{a_{\heartsuit\psi} \mid \heartsuit\psi \in l^A(q)\} \\ \gamma_q &= \{\heartsuit a_{\heartsuit\psi} \mid \heartsuit\psi \in l^A(q)\} \\ u_q^\kappa &= \{a_{\heartsuit\psi} \mid \heartsuit\psi \in \kappa \cap l^A(q)\} \quad \text{for } \kappa \in \mathsf{selections} \\ \Theta_q^\Xi &= \{u_q^\kappa \mid \kappa \in \Xi\} \end{aligned} \tag{4.1}$$

Thus, γ_q abstracts modalized formulae $\heartsuit\psi$ found in the label of q into $\heartsuit a_{\heartsuit\psi}$, and Θ_q^Ξ contains, for each $\kappa \in \Xi$, the set u_q^κ of variables $a_{\heartsuit\psi}$ such that $\heartsuit\psi \in l^A(q)$ and $\psi \in \Delta(\heartsuit\psi, \kappa)$, so that the non-deterministic tracking automaton A_χ tracks $\heartsuit\psi$ to ψ under κ. In the reading of Θ_q^Ξ suggested in Remark 4.2, Θ_q^Ξ is understood as the disjunction of the u_q^κ over all $\kappa \in \Xi$, with u_q^κ read as the conjunctive clause $\bigwedge_{\heartsuit\psi\in\kappa\cap l^A(q)} a_{\heartsuit\psi} \wedge \bigwedge_{\heartsuit\psi\in l^A(q)\setminus\kappa} \neg a_{\heartsuit\psi}$. We can similarly understand Ξ as a disjunctive normal form over atoms of the form $\heartsuit\psi \in l^A(q)$, and Θ_q^Ξ arises from Ξ by simply renaming these atoms into $a_{\heartsuit\psi}$. Notice also that the interpretation of $a_{\heartsuit\psi} \in V_q$ over Θ_q^Ξ as per Definition 4.1 is $[\![a]\!]_0^{\Theta_q^\Xi} = \{u_q^\kappa \mid \kappa \in \Xi, \heartsuit\psi \in \kappa\}$, a set which depends monotonically on Ξ. There is thus a balance to strike in selecting the set Ξ, which needs to be large enough to ensure satisfiability of the one-step pair (γ_q, Θ_q^Ξ), but on the other hand enlarging Ξ implies having to track more formulae.

Furthermore, given $\heartsuit\psi \in \mathbf{F}$ and $\Xi \subseteq \mathsf{selections}$, we write

$$\Xi/\heartsuit\psi = \{\kappa \in \Xi \mid \heartsuit\psi \in \kappa\}. \tag{4.2}$$

Definition 4.5 (Pre-tableaux and tableaux)**.** A *pre-tableau* $(W, \Sigma, \delta', q_{\mathsf{init}}, \Omega')$, or just (W, δ'), *for* χ consists of a set $W \subseteq D_\chi$ of nodes, a partial transition map $\delta'\colon W \times \Sigma \rightharpoonup W$, and a priority map $\Omega'\colon W \to \mathbb{N}$ such that the following conditions hold:

(1) $(W, \Sigma, \delta', q_{\mathsf{init}}, \Omega')$ is a *partial subautomaton* of B_χ. That is, the initial node q_{init} of B_χ is in W; $\delta'(q,\sigma) = \delta(q,\sigma)$ whenever $\delta'(q,\sigma)$ is defined for $q \in W, \sigma \in \Sigma$; and $\Omega'(q) = \Omega(q)$ for all $q \in W$. (Note that $\delta'(q,\sigma)$ may be undefined even when $\delta(q,\sigma) \in W$.)

(2) For all $q \in W$, we have $\bot \notin l^A(q)$, and there is a unique $\tau \in \mathsf{choices}$ such that $\delta'(q,\tau)$ is defined (it then equals $\delta(q,\tau)$ by (1)).

(3) For all $q \in W$, the one-step pair $(\gamma_q, \Theta_q^{\Xi(q)})$ (notation as per (4.1)) is one-step satisfiable, where

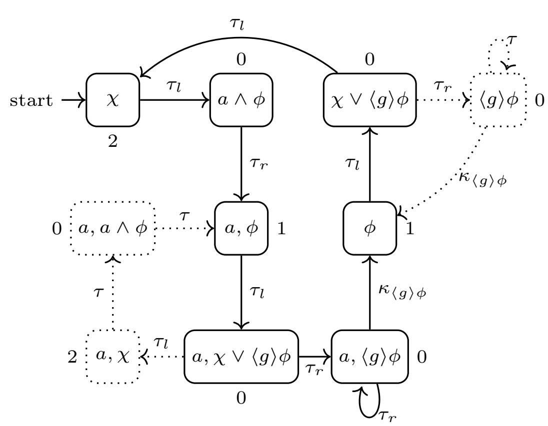
$$\Xi(q) = \{\kappa \in \mathsf{selections} \mid \kappa \subseteq l^A(q) \text{ and } \delta'(q,\kappa) \text{ is defined}\} \tag{4.3}$$

(i.e. the modal label of q is one-step satisfiable over the labels of the modal δ'-successors of q).

We refer to transitions in (W, δ') under letters in $\mathsf{choices}$ as *local transitions*, and under letters in $\mathsf{selections}$ as *modal transitions*. By (2), there is, from every $q \in W$, a unique *local* run, i.e. one consisting of only local transitions, which we denote by $\rho(q)$. A *tableau* is a pre-tableau in which every (infinite) run starting at $q_0 = q_{\mathsf{init}}$ is accepting.

Thus, a pre-tableau (W, δ') is obtained from B_χ by keeping just a single outgoing local transition at each node, and by removing some of the modal transitions in such a way that the remaining modal transitions still suffice for satisfaction of the modal literals in the label. In order for (W, δ') to be a tableau, we additionally require that all runs of this automaton are accepting (however, there may be infinite words over Σ on which (W, δ') does not have a run, and which are thus not accepted). Since the definition of u_q^κ as per (4.1) only deems a modal argument ψ to be satisfied in a κ-successor of q if ψ is tracked under κ, there is a balance to be struck in choosing the modal transitions to keep – as indicated, these need to suffice to satisfy all modal literals in the label, but every modal transition that is kept induces more tracking that may expose infinite deferral.

Example 4.6 (Tableaux)**.** Recall the co-determinized tracking automaton B_χ for the monotone μ-calculus formula $\chi = \nu X. (a \wedge \mu Y. (X \vee \langle g \rangle Y))$ from Example 3.5. To avoid triviality, we slightly tweak the semantics to work with *serial* monotone neighbourhood frames, i.e. coalgebras for the functor $\mathcal{M}_s^{\mathcal{A}} \times \mathcal{P}(\mathsf{P})$ where $\mathcal{M}_s$ is the *serial monotone neighbourhood functor* $\mathcal{M}_s$ given by $\mathcal{M}_s(X) = \{N \in \mathcal{M}X \mid \emptyset \notin N \neq \emptyset\}$. (In a serial monotone neighbourhood frame, we thus cannot satisfy a formula $\Diamond_g \phi$ at a state c by just making the empty set a neighbourhood of c.) Below we show a partial automaton (obtained from B_χ by removing various transitions and nodes) that is a tableau for χ; for better comparison with the original automaton B_χ, we use dotted transitions and nodes to depict the parts of B_χ that have been removed (and are not considered to belong to the tableau). Recall that $\kappa_{\langle g \rangle \phi} = \{\langle g \rangle \phi\}$.



One easily verifies that this structure is indeed a tableau: First, it is clearly a subautomaton of B_χ. Moreover, every (reachable) node has exactly one outgoing local transition, and no node contains $\bot$ in its label. Again we skip the treatment of propositional atoms as modalities, and concentrate instead on one-step satisfiability in the bottom-right node, which for purposes of the subsequent discussion we denote as q. In q, the notation introduced in (4.1) and (4.3) instantiates as follows. We have $V_q = \{a_{\langle g\rangle\phi}\}$, $\gamma_q = \{\langle g\rangle a_{\langle g\rangle\phi}\}$, $\Xi(q) = \{\kappa_{\langle g\rangle\phi}\}$, and $\Theta_q^{\Xi(q)} = \{u_q^{\kappa_{\langle g\rangle\phi}}\} = \{\{a_{\langle g\rangle\phi}\}\}$. One-step satisfiability of the one-step pair $(\gamma_q, \Theta_q^{\Xi(q)}) = (\{\langle g\rangle a_{\langle g\rangle\phi}\}, \{\{a_{\langle g\rangle\phi}\}\})$ is obvious. Finally, every infinite run of this subautomaton either loops at the bottom-right node forever, or visits the initial node infinitely often; in either case, the maximal priority that is visited infinitely often is even, so the run is accepting.

We will see that there is a tableau for χ if and only if χ is satisfiable. We go on to show one direction of this statement ('only if') now; the other direction is a consequence of the results of Section 5 below. That is, we will construct a coalgebraic model of the target formula χ on a tableau for χ. As indicated above, the key property of such a coalgebra is *coherence* w.r.t. the tableau. We will build the coalgebraic model using only so-called *state nodes* of the tableau, defined next.

Definition 4.7 (Local runs, pre-tableau states)**.** Let (W, δ') be a pre-tableau. A node $q \in W$ is a *state node* if the local run $\rho(q)$ that starts at q is a cycle. We denote the set of state nodes of (W, δ') by $\mathsf{states}(W, \delta') \subseteq W$. For $q \in W$, we let $\lceil q\rceil$ denote the first state node (for definiteness) on $\rho(q)$ (possibly $\lceil q\rceil = q$), and extend this notation to sets of nodes, putting $\lceil V\rceil = \{\lceil q\rceil \mid q \in V\} \subseteq \mathsf{states}(W, \delta')$ for $V \subseteq W$.

Example 4.8 (Local runs, pre-tableau states)**.** The tableau from Example 4.6 provides just a single way to construct local runs (namely, by eventually staying forever in the bottom right node) and, consequently, contains just a single state node. Exploiting the fact that labels happen to be unique in the example, we refer to states by their labels. Then we have $\lceil W\rceil = \mathsf{states}(W, \delta') = \{\{a, \langle g\rangle\phi\}\}$. We can however, modify the tableau in this example and obtain an alternative tableau by taking the left (τ_l-)transition at $\{a, \chi \vee \langle g\rangle\phi\}$, instead of the right ($\tau_r$-)transition. In this alternative tableau, we then have state nodes $\{a, \phi\}$, $\{a, \chi \vee \langle g\rangle\phi\}$, $\{a, \chi\}$ and $\{a, a \wedge \phi\}$, all of which are part of the local run that loops through the bottom left cycle of the automaton. Then we have, e.g., $\lceil\{a \wedge \phi\}\rceil = \{a, \phi\}$ but $\lceil\{a, \chi\}\rceil = \{a, \chi\}$.

Remark 4.9 (Local runs, pre-tableau states)**.** Observe that the labels of nodes along the local run $\rho(q)$ become semantically stronger through the choice of disjuncts in disjunctions. In particular, the set of modal literals contained in the label grows monotonically along $\rho(q)$. At the same time, formulae may be syntactically lost from the label along steps of the local run; e.g. a disjunction may be replaced with a disjunct, a conjunction with both its conjuncts, and fixpoint literals may be unfolded. All (state) nodes on the local run of a state node are thus semantically equivalent, and contain the same modal literals, but otherwise may differ syntactically. We use the mechanism of local runs to avoid introducing a notion of non-modal entailment that combines propositional entailment and fixpoint unfolding.

Definition 4.10 (Coherence)**.** Let (W, δ') be a pre-tableau. A coalgebra structure ξ on $\mathsf{states}(W, \delta')$ is *coherent* (over (W, δ')) if for all $q \in \mathsf{states}(W, \delta')$ and all $\heartsuit\psi \in \mathbf{F}$,

$$\heartsuit\psi \in l^A(q) \text{ implies } \xi(q) \in [\![\heartsuit]\!]\lceil\delta'(q, \mathsf{selections}/\heartsuit\psi)\rceil.$$

Note that $\psi \in l^A(\delta'(q, \kappa))$ for every $\kappa \in \mathsf{selections}/\heartsuit\psi$, so ψ is semantically entailed by $l^A(\lceil\delta'(q, \kappa)\rceil)$. The converse, however, does not hold, i.e. even if $l^A(q')$ entails ψ, it need not be the case that q' is on the local run of some node in $\delta'(q, \mathsf{selections}/\heartsuit\psi)$. Requiring that $\xi(q) \in [\![\heartsuit]\!](\lceil\delta'(q, \mathsf{selections}/\heartsuit\psi)\rceil)$ in the above definition thus means that we insist that $\heartsuit\psi$ is satisfied considering only those successors of q to which ψ is tracked.

Due to property (3) of pre-tableaux, coherent coalgebra structures always exist:

Lemma 4.11 (Existence lemma)**.** *Let (W,δ') be a pre-tableau. Then there is a coherent coalgebra structure on* $\mathsf{states}(W,\delta')$*.*

Proof. Let $q \in \mathsf{states}(W,\delta')$ be a state node. Since (W,δ') is a pre-tableau, we can pick

$$t \in [\![\gamma_q]\!]_1^{\Theta_q^{\Xi(q)}},$$

in notation as per (4.1) and (4.3); in particular, $\Theta_q^{\Xi(q)} = \{u_q^\kappa \mid \kappa \subseteq l^A(q)$ and $\delta'(q,\kappa)$ is defined$\}$ where $u_q^\kappa = \{a_{\heartsuit\psi} \mid \heartsuit\psi \in \kappa\}$; so $\Theta_q^{\Xi(q)}$ abstracts the labels of the successors of q in the pretableau (W,δ'). Let $h\colon \Theta_q^{\Xi(q)} \to \Xi(q)$ be a section of the surjective map $\Xi(q) \to \Theta_q^{\Xi(q)}$, $\kappa \mapsto u_q^\kappa$ (so $u = u_q^{h(u)}$ for $u \in \Theta_q^{\Xi(q)}$), and put $\xi(q) = (Fg)(t)$ where $g\colon \Theta_q^{\Xi(q)} \to \mathsf{states}(W,\delta')$ is defined by $g(u) = \lceil \delta'(q,h(u)) \rceil$. We show that thus defined, ξ is coherent: Let $\heartsuit\psi \in l^A(q)$. Then $\heartsuit a_{\heartsuit\psi} \in \gamma_q$, so $t \in [\![\heartsuit]\!][\![a_{\heartsuit\psi}]\!]_0^{\Theta_q^{\Xi(q)}}$. By naturality and monotonicity of $[\![\heartsuit]\!]$, $\xi(q) \in [\![\heartsuit]\!]\lceil \delta'(q, \mathsf{selections}/\heartsuit\psi) \rceil$ follows once we show that

$$[\![a_{\heartsuit\psi}]\!]_0^{\Theta_q^{\Xi(q)}} \subseteq g^{-1}[\lceil \delta'(q, \mathsf{selections}/\heartsuit\psi) \rceil].$$

So let $u \in [\![a_{\heartsuit\psi}]\!]_0^{\Theta_q^{\Xi(q)}}$, that is, $a_{\heartsuit\psi} \in u$. By (4.3) and (4.1), $\delta'(q,h(u))$ is defined and $\heartsuit\psi \in h(u)$, so $\delta'(q,h(u)) \in \delta'(q,\mathsf{selections}/\heartsuit\psi)$ and hence $g(u) \in \lceil \delta'(q,\mathsf{selections}/\heartsuit\psi) \rceil$ as required. □

We finally show that a coherent coalgebra structure is indeed a model of χ:

Lemma 4.12 (Truth lemma)**.** *Let (W,δ') be a tableau, and let ξ be a coherent coalgebra structure on $V := \mathsf{states}(W,\delta')$. Then $\lceil q_{\mathsf{init}} \rceil \in [\![\chi]\!]$ in (V,ξ).*

Proof. By Theorem 3.15, it suffices to show that $\exists$ wins the position $(\lceil q_{\mathsf{init}} \rceil, \chi)$ in the model checking game $\mathsf{G}_{\chi,(V,\xi)}$. We define a history-dependent $\exists$-strategy s in $\mathsf{G}_{\chi,(V,\xi)}$, maintaining the invariant that if (q_n,ψ_n) is the n-th position of the shape (q',ψ') visited in the play, then

> there is $u_n \in W$ such that $\psi_n \in l^A(u_n)$ and q_n lies on the local run $\rho(u_n)$, and for $n > 0$ there is a word w_n such that $u_n = \delta'(u_{n-1},w_n)$ and $\psi_n \in \Delta(\psi_{n-1},w_n)$. Moreover,
> - If ψ_{n-1} has the form $\psi_{n-1} = \heartsuit\phi$, then w_{n-1} has the form $w_{n-1} = \tau_1,\dots,\tau_m,\kappa$ where $m \geq 0$, $\tau_1,\dots,\tau_m \in \mathsf{choices}$ and $\kappa \in \mathsf{selections}/\heartsuit\phi$. (Notice that this description of w_{n-1} already implies $\Delta(\psi_{n-1},w_{n-1}) = \Delta(\heartsuit\phi,w_{n-1}) = \{\phi\} = \{\psi_n\}$.)
> - Otherwise, w_{n-1} has the form $w_n = \tau$ where $\tau \in \mathsf{choices}$.

The history-dependence of s is caused by keeping the tableau node u_n in memory. The invariant holds initially, i.e. at $(q_0,\psi_0) = (\lceil q_{\mathsf{init}} \rceil, \chi)$, for $u_0 = q_{\mathsf{init}}$. We show next that $\exists$ can enforce the invariant at (q_{n+1},ψ_{n+1}) if it holds at (q_n,ψ_n). Since (W,δ') is a pre-tableau, we have a unique $\tau \in \mathsf{choices}$ such that $\delta'(u_i,\tau)$ is defined. We distinguish cases on ψ_n:

(1) $\psi_n = \bot$: By the invariant and the definition of pre-tableaux, this case does not occur.
(2) $\psi_n = \top$: $\exists$ wins immediately.
(3) $\psi_n = \phi_1 \wedge \phi_2$: Then (q_n,ψ_n) belongs to $\forall$, who moves to (q_{n+1},ψ_{n+1}) where $q_{n+1} = q_n$ and $\psi_{n+1} \in \{\phi_1,\phi_2\}$. The invariant is preserved by taking $u_{n+1} = \delta'(u_i,\tau)$.

(4) $\psi_n = \phi_1 \vee \phi_2$: We define s by letting $\exists$ move to $(q_{n+1}, \psi_{n+1}) = (q_n, \tau(\psi_n))$. Again, the invariant is preserved by taking $u_{n+1} = \delta'(u_i, \tau)$.
(5) $\psi_n = \eta x.\phi$: We define s by letting $\exists$ play the only available move, to $(q_{n+1}, \psi_{n+1}) = (q_n, \phi[\eta x.\phi/x])$; again, the invariant is preserved by taking $u_{n+1} = \delta'(u_i, \tau)$.
(6) $\psi_n = \heartsuit\phi$: It follows from the invariant that $\heartsuit\phi \in l^A(q_n)$, since $\heartsuit\phi$ is never processed along $\rho(u_n)$. Since ξ is coherent, we thus have $\xi(q_n) \in [\![\heartsuit]\!](D)$ for $D = \lceil\delta'(q_n, \mathsf{selections}/\heartsuit\phi)\rceil$, so we can define s by letting $\exists$ move to (D, ϕ). If $D = \emptyset$, then $\exists$ wins immediately. Otherwise, $\forall$ moves to some position (q_{n+1}, ϕ) (so $\psi_{n+1} = \phi$) such that $q_{n+1} \in D$, i.e. there is $\kappa \in \mathsf{selections}/\heartsuit\phi$ such that $q_{n+1} = \lceil\delta(q_n, \kappa)\rceil$. Since by the invariant, q_n lies on the local path $\rho(u_n)$, we have $\delta(q_n, \kappa) = \delta(u_n, w_n)$ for $w_n \in \Sigma^*$ of the required form $w_n = \tau_1, \dots, \tau_m, \kappa$ where $\tau_1, \dots, \tau_m \in \mathsf{choices}$, so the invariant is preserved by taking $u_{n+1} = \delta(q_n, \kappa)$. In particular, $\psi_{n+1} = \phi$ is in $l^A(u_{n+1})$ and in $\Delta(\psi_n, w_n)$ because $\heartsuit\phi \in \kappa$, and q_{n+1} is on the local path $\rho(u_{n+1})$.

We have to show that s is a winning strategy; since, as we have noted, the game never reaches a position of the form $(q, \bot)$, $\exists$ wins all finite plays, so it remains only to show that $\exists$ wins every infinite play π that follows s. By the invariant, π induces a word $w = w_0 w_1 \ldots \in \Sigma^\omega$, a run $\bar{\pi} = u_0, u_1, \ldots$ (with $u_0 = q_{\mathsf{init}}$) of the tableau (W, δ') on w, and a run ρ of the non-deterministic tracking automaton A_χ on w. Since (W, δ') is a tableau and w has an infinite run, w is accepted by (W, δ') and hence also by B_χ. Thus, w is rejected by A_χ; in particular, ρ is a non-accepting run of A_χ. Now ρ differs from the sequence of formulae occurring in π only by possible finite repetition of formulae of the form $\psi_n = \heartsuit\phi$, caused by A_χ looping on the choice functions occurring in w_n. As the winning objective in $\mathsf{G}_{\chi,(V,\xi)}$ is dual to acceptance in A_χ, which is unaffected by finite repetition of letters ('stuttering'), $\exists$ thus wins the play π. □

5. Satisfiability Games

We now introduce a generic game characterization of satisfiability in the coalgebraic μ-calculus (Definition 5.1), rooted in classical algorithmic treatments of the relational μ-calculus as well as in previous work on the coalgebraic μ-calculus [FLV10, CKP11a] (see Section 1 and Remark 6.8 for a detailed discussion). In the game, the existential player effectively attempts to establish existence of a tableau (Section 4) for the target formula χ. Like tableaux, the game thus involves the notion of one-step satisfiability (Definition 4.1). (We note that a similar condition appears in a previous notion of satisfiability game for the coalgebraic μ-calculus [FLV10], which however is otherwise markedly different from ours, cf. Remark 6.8. The notion of tableau game used in the algorithm based on complete sets of modal tableau rules [CKP11a] has a more similar shape to ours but appears slightly larger in that automata nodes that go into the model construction are additionally annotated with tableau sequents.) We prove correctness of the game by showing on the one hand that a tableau may indeed be extracted from a winning strategy of the existential player (*completeness*), and on the other hand that a winning strategy of the existential player can be extracted from a given model of the target formula χ (*soundness*), and indeed may be obtained from a winning strategy of the existential player in the corresponding model checking game.

We first present the definition of the satisfiability game, which like the model checking game (Section 3) takes the shape of a standard parity game. Recall that $\mathsf{B}_\chi = (D_\chi, \Sigma, \delta, q_{\mathsf{init}}, \Omega)$ is the co-determinized tracking automaton for the target formula χ, and comes with the labelling function $l^A : D_\chi \to \mathcal{P}(\mathbf{F})$.

Definition 5.1 (Satisfiability game)**.** The *satisfiability game* $\mathsf{G}_\chi = (V_\forall, V_\exists, E, v_0, \Omega')$ for χ is a parity game with sets

$$V_\exists = D_\chi \times \{0,1\} \qquad\qquad V_\forall = D_\chi \cup (D_\chi \times \mathcal{P}(\mathsf{selections}))$$

of positions and $v_0 = q_{\mathsf{init}}$. The moves and the priorities in G_χ are defined by the following table, where $q \in D_\chi$ and $\Xi \in \mathcal{P}(\mathsf{selections})$, and we write

$$\mathsf{selections}(q) = \{\kappa \in \mathsf{selections} \mid \kappa \subseteq l^A(q)\}.$$

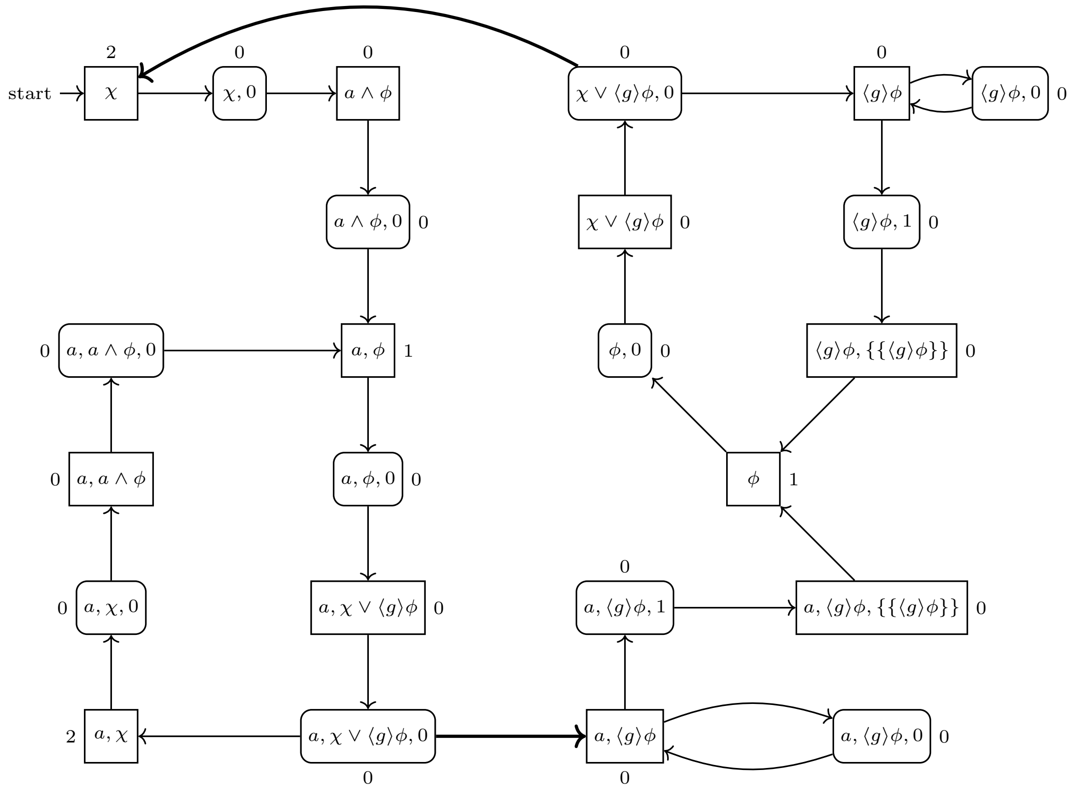
position	owner	set of allowed moves	priority
q	$\forall$	$\{(q,0),(q,1)\}$	$\Omega(q)$
$(q,0)$	$\exists$	$\{\delta(q,\tau) \in D_\chi \mid \tau \in \mathsf{choices}, \bot \notin l^A(q)\}$	0
$(q,1)$	$\exists$	$\{(q,\Xi) \mid \Xi \in \mathcal{P}(\mathsf{selections}(q)), (\gamma_q, \Theta_q^\Xi)$ is satisfiable$\}$	0
(q,Ξ)	$\forall$	$\{\delta(q,\kappa) \mid \kappa \in \Xi\}$	0

Recall here that the components of the one-step pair (γ_q, Θ_q^Ξ) are

$$\gamma_q = \{\heartsuit a_{\heartsuit\psi} \mid \heartsuit\psi \in l^A(q)\} \qquad \text{and}$$
$$\Theta_q^\Xi = \{u_q^\kappa = \{a_{\heartsuit\psi} \mid \heartsuit\psi \in l^A(q) \cap \kappa\} \mid \kappa \in \Xi\}.$$

Thus, G_χ is a parity game with $2n_0k'$ priorities (inherited from B_χ, cf. Section 3) and $|D_\chi|(3+2^{2^{n_0}})$ positions. As indicated above, the game is aimed at determining whether there exists a tableau for χ, and thus closely follows Definition 4.5. Specifically, when the play reaches a node $q \in D_\chi$, this indicates that the node needs to be included in the tableau, so $\forall$ may challenge either the propositional clause (2) or the modal clause (3) of the definition of pre-tableaux by moving to $(q,0)$ or to $(q,1)$, respectively. The admissibility conditions for the respective subsequent $\exists$-moves match the conditions given in the relevant clauses of Definition 4.5; in particular, $\exists$ loses $(q,0)$ if $\bot \in l^A(q)$. The winning condition of G_χ ensures the tableau property from Definition 4.5; that is, $\exists$ wins a play π iff the sequence of positions of the form $q \in D_\chi$ encountered on π is an accepting run of B_χ. We formally establish in Lemma 5.3 that $\exists$ winning the game really does guarantee existence of a tableau for χ.

Example 5.2 (Satisfiability game)**.** Recall the co-determinized tracking automaton B_χ for the monotone μ-calculus formula $\chi = \nu X.\,(a \wedge \mu Y.\,(X \vee \langle g\rangle Y))$ from Example 3.5 (with ϕ abbreviating $\mu Y.\,(X \vee \langle g\rangle Y)$). Like already in Example 4.6, we restrict the semantics to serial monotone neighbourhood frames. Below we show the satisfiability game G_χ for χ, constructed over B_χ. Again, rounded boxes indicate $\exists$-positions.



(In the figure, we have omitted positions of the form $(q, 1)$ where $l^A(q)$ does not contain any modal literal, such as $(\chi, 1)$ in the present example, which $\exists$ wins immediately by moving to $(q, \emptyset)$.) The bold arrows indicate an $\exists$-strategy s that wins the initial position χ in the game. As indicated above, such strategies induce tableaux; in this case, s induces (essentially) the tableau considered in Example 4.6 above.

As indicated above, we prove completeness of the game by showing that winning strategies for the existential player in the satisfiability game induce tableaux:

Lemma 5.3. *If the existential player wins the satisfiability game G_χ, then there is a tableau for χ.*

Proof. Let $s\colon V_\exists \to V$ be a history-free $\exists$-strategy that wins q_{init} in G_χ, and let W be the set of positions of the form $q \in D_\chi$ that are reachable in plays that follow s. We then define $\delta'\colon W \times \Sigma \rightharpoonup W$ as follows. Let $q \in W$. By the definition of the satisfiability game, there is $\tau \in \mathsf{choices}$ such that $s(q, 0) = \delta(q, \tau) \in W$ (in particular, $\bot \notin l^A(q)$); we put $\delta'(q, \tau) = \delta(q, \tau)$, and let $\delta'(q, \tau')$ be undefined for all other $\tau' \in \mathsf{choices}$. Similarly, $s(q, 1)$ has the form $s(q, 1) = (q, \Xi)$ where $\Xi \in \mathcal{P}(\mathsf{selections}(q))$, and we put $\delta'(q, \kappa) = \delta(q, \kappa)$ if $\kappa \in \Xi$, and let $\delta'(q, \kappa)$ be undefined otherwise, noting that $\delta(q, \kappa) \in W$ for $\kappa \in \Xi$ because $\forall$ can move to $\delta(q, \kappa)$ from $(q, \Xi) = s(q, 1)$. Then (W, δ') is a pre-tableau by construction. To show that (W, δ') is a tableau, let $q_{\mathsf{init}} = q_0, q_1, \ldots$ be a run of (W, δ') on a word $w = \sigma_0, \sigma_1, \ldots$. By construction of (W, δ'), this run induces a play π in G_χ that follows s (explicitly, if $\sigma_i \in \mathsf{choices}$, then the play has one intermediate position $(q_i, 0)$ between q_i and $q_{i+1} = s(q_i, 0)$, and if $\sigma_i \in \mathsf{selections}$, then the play has two intermediate positions $(q_i, 1)$ and $s(q_i, 1)$ between q_i and q_{i+1}). Since s is a winning strategy, π is won by $\exists$, which by the comments after

Definition 5.1 means that $q_0, q_1, \dots$ is an accepting run of B_χ, showing that (W, δ') is a tableau. □

It remains to prove soundness. As indicated above, we proceed by transforming a winning strategy in the model checking game into one in the satisfiability game, exploiting that the former is based on non-acceptance in the tracking automaton A_χ and the latter on acceptance in the co-determinized tracking automaton B_χ.

Lemma 5.4 (Soundness)**.** *Let χ be satisfiable. Then the existential player wins G_χ.*

Proof. Fix a coalgebra (C, ξ) and a state $x_0 \in C$ such that $x_0 \models \chi$. By Theorem 3.15, $\exists$ has a history-free strategy s' that wins (x_0, χ) in $\mathsf{G}_{\chi,(C,\xi)}$. We construct a history-dependent winning strategy s for $\exists$ in the satisfiability game G_χ that maintains the invariant that in positions $(q, b) \in V_\exists = D_\chi \times \{0, 1\}$, there is $x \in C$ such that

$$\text{for all } \psi \in l^A(q),\ s' \text{ wins } (x, \psi) \text{ in } \mathsf{G}_{\chi,(C,\xi)};$$

we call such an x a *realizer* of q. More precisely, we keep the realizer in memory, and in each step, we construct the new realizer from the previous one; this is why the strategy we construct is not history-free. We write $s((q, b), x)$ for the move recommended by s when in a position $(q, b) \in V_\exists$ with realizer x.

We can pick x_0 as the realizer of q_{init}, ensuring that the invariant holds initially since $l^A(q_{\mathsf{init}}) = \{\chi\}$ and s' wins (x_0, χ). To see that the existential player can maintain the invariant, let $(q, b) \in V_\exists = D_\chi \times \{0, 1\}$ and let $x \in C$ be a realizer of q. We distinguish cases on b.

Case $b = 0$: We define $\tau \in \mathsf{choices}$ as follows. For each disjunction $\psi = \psi_1 \vee \psi_2 \in l^A(q)$, s' wins (x, ψ) by the invariant, and we have $s'(x, \psi) = (x, \psi_i)$ for some $i \in \{1, 2\}$; of course, s' then wins (x, ψ_i). We put $\tau(\psi) = \psi_i$. For disjunctions $\psi \in \mathbf{F}$ not contained in $l^A(q)$, define $\tau(\psi)$ arbitrarily. We put $s((q, 0), x) = \delta(q, \tau)$: since the invariant implies in particular that $\bot \notin l^A(q))$, this is a valid move.

To establish the invariant at the new position $\delta(q, \tau)$, we pick the original realizer x of q as the new realizer of $\delta(q, \tau)$. We have to show that for all $\psi \in l^A(\delta(q, \tau))$, s' wins (x, ψ) in in $\mathsf{G}_{\chi,(C,\xi)}$. By the definition of the tracking automaton (Definition 3.2), such a ψ arises in one of the following ways:

- ψ is a disjunct of disjunction in $l^A(q)$. It was shown above that s' wins (x, ψ) in this case.
- ψ is a conjunct (w.l.o.g., the left one) of a formula $\psi \wedge \phi \in l^A(q)$. Then the universal player can move from $(x, \psi \wedge \phi)$ to (x, ψ) in $\mathsf{G}_{\chi,(C,\xi)}$. Since s' wins $(x, \psi \wedge \phi)$ by the invariant, s' also wins (x, ψ).
- $\psi = \psi_1[\eta X.\, \psi_1/X]$ for some $\eta X.\, \psi_1 \in l^A(q)$. By the invariant, s' wins $(x, \eta X.\, \psi_1)$ in $\mathsf{G}_{\chi,(C,\xi)}$, so s' also wins the unique next position (x, ψ).
- $\psi = \heartsuit\psi_1 \in l^A(q)$, Then s' wins (x, ψ) by the invariant.

Case $b = 1$: For each $\heartsuit\psi \in l^A(q)$, s' wins $(x, \heartsuit\psi)$ in $\mathsf{G}_{\chi,(C,\xi)}$ by the invariant. Then $s'(x, \heartsuit\psi)$ has the form

$$s'(x, \heartsuit\psi) = (D_{\heartsuit\psi}, \psi) \quad \text{where} \quad \xi(x) \in [\![\heartsuit]\!]_C(D_{\heartsuit\psi}). \tag{5.1}$$

Put

$$\Xi = \{\kappa \in \mathsf{selections} \mid \kappa \subseteq l^A(q) \text{ and } \textstyle\bigcap_{\heartsuit\psi\in\kappa} D_{\heartsuit\psi} \neq \emptyset\}.$$

For $\kappa \in \Xi$, fix an element $y_\kappa \in \bigcap_{\heartsuit\psi\in\kappa} D_{\heartsuit\psi}$. We put $s((q, 1), x) = (q, \Xi)$. The ensuing moves of the universal player then reach a position in $V_\exists$ of the form $(\delta(q, \kappa), b')$ where

$\kappa \in \Xi$. We pick y_κ as the realizer of $\delta(q,\kappa)$. To show that this ensures the invariant, let $\psi \in l^A(\delta(q,\kappa))$, that is, $\heartsuit\psi \in \kappa$ for some $\heartsuit \in \Lambda$; we have to show that s' wins (y_κ, ψ). But this follows from the fact that s' is a winning strategy and the universal player can move from $s'(x, \heartsuit\psi) = (D_{\heartsuit\psi}, \psi)$ to (y_κ, ψ), since $y_\kappa \in D_{\heartsuit\psi}$.

It remains to show that $((q,1),(q,\Xi))$ is a valid move in G_χ. Recalling that

$$\begin{aligned}V_q &= \{a_{\heartsuit\psi} \mid \heartsuit\psi \in l^A(q)\}\\ \gamma_q &= \{\heartsuit a_{\heartsuit\psi} \mid \heartsuit\psi \in l^A(q)\}\\ u_q^\kappa &= \{a_{\heartsuit\psi} \mid \heartsuit\psi \in \kappa \cap l^A(q)\}\\ \Theta_q^\Xi &= \{u_q^\kappa \mid \kappa \in \Xi\},\end{aligned}$$

we have to show that, as discussed after (4.1), Ξ is large enough to ensure that

$$[\![\gamma_q]\!]_1^H \neq \emptyset \tag{5.2}$$

where $H = \Theta_q^\Xi$; recall here that by definition, $[\![\gamma_q]\!]_1^H = \bigcap_{\heartsuit a_{\heartsuit\psi} \in \gamma_q} [\![\heartsuit]\!]_H [\![a_{\heartsuit\psi}]\!]_0^H$. We define a labelling $l\colon C \to H$ by

$$l(y) = \{a_{\heartsuit\psi} \in V_q \mid y \in D_{\heartsuit\psi}\}. \tag{5.3}$$

Here, we have to show that for $y \in C$, we indeed have $l(y) \in H$, i.e. we have to find $\kappa \in \Xi$ such that $l(y) = u_q^\kappa$. This will hold by definition for $\kappa := \{\heartsuit\psi \mid a_{\heartsuit\psi} \in l(y)\}$, once we show that $\kappa \in \Xi$. The latter means that $D := \bigcap_{a_{\heartsuit\psi} \in l(y)} D_{\heartsuit\psi} \neq \emptyset$; but $y \in D$ by definition of $l(y)$. We now establish (5.2) by showing that

$$Fl(\xi(x)) \in [\![\gamma_q]\!]_1^H.$$

So let $\heartsuit a_{\heartsuit\psi} \in \gamma_q$, that is, $\heartsuit\psi \in l^A(q)$; we have to show $Fl(\xi(x)) \in [\![\heartsuit]\!]_H [\![a_{\heartsuit\psi}]\!]_0^H$, which by naturality is equivalent to $\xi(x) \in [\![\heartsuit]\!]_C(l^{-1}[[\![a_{\heartsuit\psi}]\!]_0^H]) = [\![\heartsuit]\!]_C(D_{\heartsuit\psi})$; but this holds by (5.1).

This concludes the proof that s is a strategy, that is, yields legal moves in G_χ. It remains to show that s is winning. In showing that $\exists$ can maintain the invariant, we have in particular shown that $\exists$ never gets stuck, and hence wins all finite plays. So let π be an infinite play that starts at v_0 and follows s; we have to show that $\exists$ wins π. As noted after Definition 5.1, π gives rise to a run r of B_χ on a word $w \in \Sigma^\omega$. In more detail, the play π consists of concatenated subplays π_i, either of the shape $q_i, (q_i, 0), q_{i+1}$ where $q_{i+1} = \delta(q_i, \sigma_i)$ for some $\sigma_i \in \mathsf{choices}$, or of the shape $q_i, (q_i, 1), (q_i, \Xi), q_{i+1}$ where $q_{i+1} = \delta(q_i, \sigma_i)$ for some $\sigma_i \in \Xi$. In this notation, $r = q_0, q_1, \dots$, where $q_0 = q_{\mathsf{init}}$, is the run of B_χ on $w = \sigma_0, \sigma_1, \dots$. Again as noted after Definition 5.1, the winning objective of the existential player in G_χ is $w \in L(\mathsf{B}_\chi)$. Since B_χ complements A_χ, we show equivalently that every run $\rho = \psi_0, \psi_1, \dots$ of A_χ on the word w, where $\psi_0 = \chi$, is non-accepting. From ρ, we obtain a play π' of the model checking game $\mathsf{G}_{\chi,(C,\xi)}$ that starts at (x_0, χ) and follows s', hence is won by $\exists$, and moreover induces w in the sense discussed after Definition 3.11. Specifically, the positions of the form (x, ψ) visited by π' are precisely $(x_0, \psi_0), (x_1, \psi_1), \dots$ where x_i is the realizer of q_i according to the invariant; interceding moves to positions of the form (D, ψ) with $D \subseteq C$ are determined by s'. Since as noted after Definition 3.11, the winning objective of $\exists$ in the model checking game is non-acceptance of the associated run of A_χ, this implies that ρ is non-accepting. $\square$

The results so far are tied up as follows:

Theorem 5.5 (Soundness and completeness)**.** *The following are equivalent:*

(1) *The existential player wins the position q_{init} in G_χ.*

(2) *There is a tableau for* χ*.*
(3) *The formula* χ *is satisfiable.*

Proof. The implication (1) $\implies$ (2) is Lemma 5.3. We have shown the implication (3) $\implies$ (1) in Lemma 5.4. We prove (2) $\implies$ (3): Let (W, δ') be a tableau for χ. By the existence lemma (Lemma 4.11), there is a coherent coalgebra built over (W, δ'), which by the truth lemma (Lemma 4.12) is a model for χ. $\square$

Our model construction in the proof of Lemma 4.11 moreover yields the same bound on minimum model size as in earlier work on the coalgebraic μ-calculus [CKP11a, FLV10]:

Corollary 5.6 (Small-model property)**.** *Let* χ *be a satisfiable coalgebraic* μ*-calculus formula, with parameters* n_0, k *and* $k' = \lfloor (k+1)/2 \rfloor + 1$ *as in the running notation. Then* χ *has a model of size* $\mathcal{O}(((nk')!)^2) \in 2^{\mathcal{O}(nk \log n)}$*.*

Remark 5.7 (Polynomially branching models)**.** In addition to having an exponentially bounded number of states (Corollary 5.6), the models (V, ξ) constructed in the above completeness proof are also polynomially branching, provided that the logic has the one-step polysize model property, which holds in all our running examples (Remark 4.4). By this we mean that there is a polynomial p such that for every $q \in V$, there is a subset $V_0 \subseteq V$ such that $|V_0| \leq p(n)$ and $\xi(q)$ has the form $\xi(q) = Fi(t)$ where $i\colon V_0 \to V$ is the subset inclusion. This property is immediate from the construction of coherent coalgebras in the proof of the existence lemma (Lemma 4.11), in which $\xi(q)$ is obtained from a model of a one-step pair over $\mathbf{F}$. 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.

6. Lazy Game Solving for the Coalgebraic μ-Calculus

We proceed to show that the satisfiability game introduced in the previous section can be solved in singly exponential time (under mild assumptions on the complexity of the underlying one-step satisfiability problem). To this end, we introduce a satisfiability checking algorithm that solves the game *on-the-fly* (that is, in a lazy fashion), and analyse the runtime of the algorithm. As mentioned above, the obstacle to be overcome here is that the game is doubly exponentially large, specifically has singly exponentially many positions owned by the existential player but doubly exponentially many owned by the universal player. We deal with this issue by a characterization of the existential player's winning region as a nested fixpoint that lives on an exponential-sized subset of the game positions (those corresponding directly to states in the co-determinized tracking automaton B_χ), with interceding moves absorbed into the definition of the function whose fixpoint is computed (Lemma 6.3). The satisfiability checking algorithm may then be understood as computing this fixpoint, respectively determining whether the root position of the game belongs to the fixpoint.

We recall that $q_{\mathsf{init}} \in D_\chi$ is the initial node of the co-determinized tracking automaton B_χ. The algorithm expands B_χ step by step starting from q_{init}; the expansion step adds nodes according to all possible choice functions and all selections of modalities in an unexpanded node q. The order of expansion can be chosen freely, e.g. by heuristic methods. Optional intermediate game solving steps can be used judiciously to realize on-the-fly solving.

Algorithm (Satisfiability checking). To decide satisfiability of the input formula χ, initialize the sets of *unexpanded* and *expanded* nodes, $U = \{q_{\mathsf{init}}\}$ and $Q = \emptyset$, respectively.

(1) Expansion: Choose some unexpanded node $q \in U$, remove q from U, and add q to Q. Add all nodes in the sets $\{\delta(q,\tau) \in D_\chi \mid \tau \in \mathsf{choices}\} \setminus Q$ and $\{\delta(q,\kappa) \in D_\chi \mid \kappa \in \mathsf{selections}, \kappa \subseteq l^A(q)\} \setminus Q$ to U.
(2) Optional solving: Compute $\mathsf{win}^\exists_Q$ and/or $\mathsf{win}^\forall_Q$. If $q_{\mathsf{init}} \in \mathsf{win}^\exists_Q$, then return 'satisfiable', if $q_{\mathsf{init}} \in \mathsf{win}^\forall_Q$, then return 'unsatisfiable'.
(3) If $U \neq \emptyset$, then continue with Step 1.
(4) Final game solving: Compute $\mathsf{win}^\exists$. If $q_{\mathsf{init}} \in \mathsf{win}^\exists$, then return 'satisfiable', otherwise return 'unsatisfiable'.

Before analysing the run time behaviour of the algorithm, we first show how to compute the sets $\mathsf{win}^\exists_Q$ and $\mathsf{win}^\forall_Q$ in singly exponential time. Put

$$N = 2n_0k'$$

with k' as per (3.1), the number of priorities in B_χ (cf. Section 3). We define N-ary set functions f_Q and g_Q that compute one-step (tn)satisfiability w.r.t. their argument sets. These functions essentially encode short sequences of moves in G_χ leading from one node in D_χ to the next.

Definition 6.1 (Small-step game solving functions)**.** For sets $Q \subseteq D_\chi$ and $X_1, \dots, X_N \subseteq Q$, we put

$$\begin{aligned} f_Q(X_1,\dots,X_N) =& \{q \in Q \mid (\gamma_q, \Theta_q^{\Xi(X_{\Omega(q)})}) \text{ is satisfiable}, \bot \notin l^A(q) \text{ and} \\ & \exists \tau \in \mathsf{choices}.\, \delta(q,\tau) \in X_{\Omega(q)}\} \\ g_Q(X_1,\dots,X_N) =& \{q \in Q \mid (\gamma_q, \Theta_q^{\Xi(\overline{X_{\Omega(q)}})}) \text{ is not satisfiable}, \bot \in l^A(q) \text{ or} \\ & \forall \tau \in \mathsf{choices}.\, \delta(q,\tau) \in X_{\Omega(q)}\}, \end{aligned}$$

where $\Xi(X) = \{\kappa \in \mathsf{selections} \mid \kappa \subseteq l^A(q) \text{ and } \delta(q,\kappa) \in X\}$ and $\overline{X} = D_\chi \setminus X$ for $X \subseteq D_\chi$.

Note how f_Q propagates winning positions for $\exists$ in G_χ, checking whether $\exists$ has a response to both immediate next $\forall$-moves from $q \in D_\chi$ (to $(q,0)$ or $(q,1)$), while g_Q propagates winning positions for $\forall$, checking that $\forall$ wins by moving to either $(q,0)$ or $(q,1)$.

The time required for small-step game solving steps thus depends on the time complexity of the one-step satisfiability problem. In Lemma 6.4, we correspondingly give an estimate of the overall time complexity of the satisfiability checking algorithm under the assumption that the strict one-step satisfiability problem is in ExpTime.

Next we characterize the winning regions $\mathsf{win}^\exists_Q$ and $\mathsf{win}^\forall_Q$ by fixpoint expressions over $\mathcal{P}(D_\chi)$, using the small-step game solving functions f_Q and g_Q, respectively.

Definition 6.2 (Fixpoint descriptions of winning regions)**.** Given a set $Q \subseteq D_\chi$, we put

$$\mathbf{E}_Q = \eta_N X_N. \dots \eta_1 X_1. f_Q(\mathbf{X}) \qquad\qquad \mathbf{A}_Q = \overline{\eta_N} X_N \dots \overline{\eta_1} X_1. g_Q(\mathbf{X}),$$

where $\mathbf{X} = X_1, \dots, X_N$ is a vector of variables X_i ranging over subsets of Q, where $\eta_i = \mu$ for odd i, $\eta_i = \nu$ for even i, and where $\overline{\nu} = \mu$ and $\overline{\mu} = \nu$.

We will show that this fixpoint characterization is indeed correct, that is, that

$$\mathbf{E}_Q = \mathsf{win}^\exists_Q \quad \text{and} \quad \mathbf{A}_Q = \mathsf{win}^\forall_Q$$

for $Q \subseteq D_\chi$. As the sets $\mathbf{E}_Q$ and $\mathbf{A}_Q$ grow monotonically with Q, and since clearly $\mathbf{A}_{D_\chi}$ is the complement of $\mathbf{E}_{D_\chi}$, it suffices to prove that the winning region $\mathsf{win}^\exists$ in G_χ coincides with the set $\mathbf{E} := \mathbf{E}_{D_\chi}$.

Lemma 6.3. *For all $q \in D_\chi$, we have $q \in \mathbf{E}$ if and only if the existential player wins the position q in the satisfiability game G_χ.*

Proof. The fixpoint

$$\mathbf{E} = \eta_N X_N. \dots \eta_1 X_1. f_{D_\chi}(X_1, \dots, X_N)$$

on D_χ may, as discussed in Remark 3.16, be seen as described by a formula $\eta_N X_N. \dots \eta_1 X_1. \heartsuit(X_1, \dots, X_N)$ in a generalized form of the monotone μ-calculus, and thus is characterized by the corresponding instance of the subformula model checking game. The simple structure of the fixpoint allows for further simplification of the game. First, all positions of the form (q, ψ) where ψ is not a fixpoint literal or a fixpoint variable (so the next move might not be uniquely determined) have $\psi = \heartsuit(X_1, \dots, X_N)$; in particular, all such positions belong to $\exists$. Second, positions (q', X_l) reached from such a position after $\exists$'s move and a subsequent $\forall$-move automatically proceed to $(q', \heartsuit(X_1, \dots, X_N))$, with l being the maximal priority visited on the way. We thus eliminate the intermediate positions, and rename positions $(q, \heartsuit(X_1, \dots, X_N))$ into just q; similarly, we omit formula annotations on subsets of D_χ played by $\exists$ in modal moves. We write $\mathsf{G}_\mathbf{E}$ for the simplified form of the game, which is summarized by the following table:

position	owner	set of allowed moves	priority
q	$\exists$	$\{(A_1, \dots, A_N) \in \mathcal{P}(D_\chi)^N \mid q \in f_{D_\chi}(A_1, \dots, A_N)\}$	0
$(A_1, \dots, A_N)$	$\forall$	$\{(q, X_l) \mid l \in \{1, \dots, N\}, q \in A_l\}$	0
(q, X_l)	$\exists$	$\{q\}$	l

By correctness of the model checking game (Theorem 3.15), the claim is thus reduced to showing that $\exists$ wins q in $\mathsf{G}_\mathbf{E}$ iff $\exists$ wins q in G_χ. We say that $\exists$ can *force* a set $U \subseteq \{0, \dots, N\} \times D_\chi$ in position q in one of the games if $\exists$ has a strategy ensuring that $\exists$ does not lose by getting stuck and that the pair (j, q') consisting of the next position $q' \in D_\chi$ reached in the play (if any; $\forall$ might still get stuck) and the maximal priority j encountered on the way to q', including the priority of q but excluding the priority of q', lies in U. Since every infinite play in $\mathsf{G}_\mathbf{E}$ or G_χ infinitely often visits positions in D_χ, it suffices to show that at every $q \in D_\chi$, $\exists$ can force the same sets U in either of the games.

For one direction, suppose that $\exists$ can force $U \subseteq \{0, \dots, N\} \times D_\chi$ at q in $\mathsf{G}_\mathbf{E}$ by moving to $(A_1, \dots, A_N)$; in particular, $q \in f_{D_\chi}(A_1, \dots, A_N)$. In G_χ, $\exists$ then enforces U at q as follows.

- First, suppose that $\forall$ moves from q to $(q, 0)$. Since $q \in f_{D_\chi}(A_1, \dots, A_N)$, we have $\bot \notin l^A(q)$, and there is $\tau \in \mathsf{choices}$ such that $\delta(q, \tau) \in A_{\Omega(q)}$. Thus, $\exists$ can move from $(q, 0)$ to $\delta(q, \tau)$ in G_χ, the highest priority encountered on the way from q to $\delta(q, \tau)$ being $\Omega(q)$. The pair $(\Omega(q), \delta(q, \tau))$ is in U as required, since in $\mathsf{G}_\mathbf{E}$, $\forall$ can move from $(A_1, \dots, A_N)$ to $(\delta(q, \tau), X_{\Omega(q)})$, which has priority $\Omega(q)$, and the game then automatically proceeds to $\delta(q, \tau)$.
- Second, suppose that $\forall$ moves to $(q, 1)$. Since $q \in f_{D_\chi}(A_1, \dots, A_N)$, we have that the one-step pair $(\gamma_q, \Theta_q^{\Xi(A_{\Omega(q)})})$ is satisfiable, recalling that $\Xi(A_{\Omega(q)}) = \{\kappa \in \mathsf{selections} \mid \kappa \subseteq l^A(q) \text{ and } \delta(q, \kappa) \in A_{\Omega(q)}\}$; so $\exists$ can move to $(q, \Xi(A_{\Omega(q)}))$. After the next move by $\forall$, we thus end up in $\delta(q, \kappa)$ for some $\kappa \in \Xi(A_{\Omega(q)})$, with the highest priority encountered on the

way being $\Omega(q)$. Since $\kappa \in \Xi(A_{\Omega(q)})$, we have $\delta(q,\kappa) \in A_{\Omega(q)}$, so by the same analysis as in the previous case, the pair $(\Omega(q), \delta(q,\kappa))$ is in U as required.

For the converse direction, suppose that $\exists$ can force $U \subseteq \{0,\dots,N\} \times D_\chi$ at q in G_χ by moving to $\delta(q,\tau)$ in case $\forall$ moves to $(q,0)$ and to (q,Ξ) in case $\forall$ moves to $(q,1)$, where $\tau \in \mathsf{choices}$ and $\Xi \in \mathcal{P}(\mathsf{selections}(q))$. In particular, we then have $\bot \notin l^A(q)$ and the one-step pair (γ_q, Θ_q^Ξ) is satisfiable. We claim that in $\mathsf{G}_\mathbf{E}$, $\exists$ then forces U by moving to $(\emptyset,\dots,\emptyset, A_{\Omega(q)}, \emptyset,\dots,\emptyset)$ (with $A_{\Omega(q)}$ in position $\Omega(q)$) where

$$A_{\Omega(q)} = \{\delta(q,\tau)\} \cup \delta(q,\Xi).$$

We note that this is a legal move, as $q \in f_{D_\chi}(\emptyset,\dots,\emptyset, A_{\Omega(q)}, \emptyset,\dots,\emptyset)$ by construction of $A_{\Omega(q)}$ (and Remark 4.2). In $\mathsf{G}_\mathbf{E}$, $\forall$ then necessarily moves to a position $(q', X_{\Omega(q)})$ where $q' \in A_{\Omega(q)}$, and the game then automatically proceeds to q', with the maximal priority encountered on the way being $\Omega(q)$. We distinguish cases on q':

- If $q' = \delta(q,\tau)$, then $(\Omega(q), q') \in U$ as required, since $\delta(q,\tau)$ is $\exists$'s reply to $(q,0)$ forcing U in G_χ.
- Otherwise, $q' = \delta(q,\kappa)$ for some $\kappa \in \Xi$. Since (q,Ξ) is $\exists$'s reply to $(q,0)$ forcing U in G_χ, and $\forall$ can move from (q,Ξ) to $\delta(q,\kappa)$ in G_χ, we again have $(\Omega(q), q') \in U$ as required. □

Having shown how the sets $\mathsf{win}_Q^\exists$ and $\mathsf{win}_Q^\forall$ can be computed by evaluating fixpoint expressions over $\mathcal{P}(D_\chi)$, we next analyse the run time behaviour of the introduced algorithm.

Lemma 6.4 (Time analysis)**.** *If the strict one-step satisfiability problem is decidable in time $t(n)$, then the above satisfiability checking algorithm runs in time $\mathcal{O}(((2n_0k')!)^{2c} \cdot t(n))$ for some constant c if no optional game solving steps are performed, with n, k' being the parameters of the target formula as per Section 3.*

(The run time with optional game solving steps is still singly exponential; in view of the fact that exponential run time of some fixed strategy on intermediate game solving suffices to obtain the ExpTime bound on satisfiability checking, we restrict to the case without optional game solving for the sake of simplicity.)

Proof. The loop of the algorithm expands the co-determinized tracking automaton node by node and hence is executed at most $|D_\chi| \in \mathcal{O}(((n_0k')!)^2)$ times. A single expansion step can be implemented in time $\mathcal{O}(2^{n_0})$ since in both propositional and modal expansion steps, at most 2^{n_0} new nodes are added, corresponding to the maximal possible number of choice functions and matching selections, respectively. By Lemma 6.3, the final solving step can be performed by computing a fixpoint of nesting depth N of the function f over $\mathcal{P}(D_\chi)^N$. A single computation of $f(\mathbf{X})$ for a tuple $\mathbf{X} \in \mathcal{P}(D_\chi)^N$ can be implemented in time $\mathcal{O}(|D_\chi| \cdot (t(n) + 2^{n_0})) = \mathcal{O}(((n_0k')!)^2 \cdot (t(n_0) + 2^{n_0}))$ by going through all elements q of D_χ, calling the one-step satisfiability checker on $l^A(q) \cap \Lambda(\mathbf{F})$, and verifying the existence of a suitable choice letter. Since we have $N = \mathcal{O}(\log |D_\chi|)$, it follows from recent work on the computation of nested fixpoints [HS21] that these fixpoints can be computed in time $\mathcal{O}((n_0k')!^{2c} \cdot t(n_0))$, where $c = 5$. (Classical methods for computing nested fixpoints [LBC+94, Sei96] are exponential in the nesting depth, which however still leads to a singly exponential overall time bound, computed explicitly in the conference version of the paper [HS19].) Thus the complexity of the whole algorithm is dominated by the complexity of the final game solving step, and adheres to the claimed asymptotic bound. □

Relying on the correctness of satisfiability games as shown in Section 5 above, we obtain the following results.

Theorem 6.5 (Exponential-time upper bound)**.** *If the strict one-step satisfiability problem of a coalgebraic logic is in* ExpTime*, then the satisfiability problem of the corresponding coalgebraic μ-calculus is in* ExpTime.

Since as discussed in Remark 6.7 below, 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 [CKP11a, HSE16, HSD18]; however, it covers additional example logics for which no suitable tableau rules are known. In particular, by Example 4.3, we have:

Proposition 6.6. *The satisfiability problems of the following logics are in* ExpTime:

(1) the standard μ-calculus,
(2) the monotone μ-calculus (including its fragment game logic),
(3) the graded μ-calculus,
(4) the (two-valued) probabilistic μ-calculus,
(5) the graded μ-calculus with polynomial inequalities,
(6) the (two-valued) probabilistic μ-calculus with polynomial inequalities.

Remark 6.7 (Modal tableau rules)**.** As indicated in the introduction, previous generic algorithms for the coalgebraic μ-calculus [CKP$^+$11b] employ tractable sets of modal tableau rules [SP09] in place of one-step satisfiability checking. This method roughly works as follows. A *(monotone) modal tableau rule* over a finite set W of propositional variables (local to the rule) has the form ϕ/ψ where ψ is a propositional formula over W, given in disjunctive normal form as a subset of $\mathcal{P}(W)$, and ϕ is a finite subset of $\Lambda(W)$, representing a finite conjunction, that mentions every variable in W exactly once. Given a set V, a *rule match* to a finite subset γ of $\Lambda(V)$ is a pair $(\phi/\psi, \iota)$ consisting of a rule ϕ/ψ and a substitution $\iota\colon W \to V$ that acts injectively on ϕ (i.e. for $\heartsuit a, \heartsuit b \in \phi$, $\iota(a) = \iota(b)$ implies $a = b$) such that $\phi\iota \subseteq \gamma$, where we write application of the substitution ι in postfix notation. In the notation of the present paper, a set $\mathcal{R}$ of modal tableau rules is *one-step tableau sound and complete* if the following condition holds for each one-step pair (γ, Θ) over V: The pair (γ, Θ) is satisfiable iff $\psi\iota \cap \Theta \neq \emptyset$ for each rule match $(\phi/\psi, \iota)$ as above to γ; note that applying the substitution $\iota\colon W \to V$ to $\psi \subseteq \mathcal{P}(W)$ yields a propositional formula $\psi\iota$ over V that is represented as a subset of $\mathcal{P}(V)$.

A rule set $\mathcal{R}$ is *exponentially tractable* if rule matches can be encoded as strings in such a way that every rule match to a given finite set $\gamma \subseteq \Lambda(V)$ has a code of polynomial size in $\mathsf{size}(\gamma)$, and moreover (i) it can be decided in exponential time whether a given code actually encodes a rule match to γ and (ii) the conclusion $\psi\iota$ of a rule match $(\phi/\psi, \iota)$ can be computed from its code in exponential time. Using an exponentially tractable rule set, we can decide the strict one-step satisfiability problem in exponential time: Given a one-step pair (γ, Θ), go through all codes of possible rule matches, filtering for actual matches $(\phi/\psi, \iota)$, and check for each such match that $\psi\iota \cap \Theta \neq \emptyset$. Thus, the present approach applies more generally than the approach via modal tableau rules. See also [KPS22] for a more detailed discussion of the relationship.

Tractable sets of tableau rules for the graded μ-calculus and the Presburger μ-calculus have been claimed in previous work [SP09, KP10]. However, these rule sets have since turned out to be incomplete. Indeed the rule sets are very similar to rule sets for real-valued systems, and remain sound over an evident real-valued relaxation of the semantics, an observation from which concrete examples showing incompleteness are obtained rather immediately both for the Presburger [KPS22, Remark 3.8] and for the graded case [GHH+23, Appendix of extended version].

Remark 6.8 (Λ-automata)**.** As indicated in Section 1, the satisfiability game considered by Fontaine et al. [FLV10] actually checks emptiness of so-called Λ-automata, into which formulae of the coalgebraic μ-calculus can be translated with only polynomial blow-up. Non-emptiness of a Λ-automaton $\mathbb{A}$ with set A of states is checked using a game $Sat(\mathbb{A})$ that has pairs of automata states from A as $\exists$-positions, and sets of binary relations on A as $\forall$-positions. The winning condition of $Sat(\mathbb{A})$ is regular but not parity, so winning strategies in $Sat(\mathbb{A})$ need to depend on memory states $m \in M$ from an exponential-sized set M. The model construction then has states being pairs (v, m) consisting of an $\exists$-position v and a memory state $m \in M$ (while our model construction uses states from the co-determinized tracking automaton). As we indicate in the introduction, it does not seem likely that our approach to solving a doubly-exponential-sized satisfiability game in singly exponential time will in general transfer to $Sat(\mathbb{A})$ (for the case where a tractable set of tableau rules is known, $Sat(\mathbb{A})$ has been reformulated to be solvable in exponential time [FLV10, Section 5]), as none of the two types of positions in $Sat(\mathbb{A})$ has the right size (there are doubly exponentially many $\forall$-positions and polynomially many $\exists$-positions). Also, our fixpoint description relies on the fact that our game has a parity winning condition, and it is not clear how it would transfer to a regular game.

Remark 6.9 (One-step satisfiability and fusion)**.** The criterion of Theorem 6.5 is stable under fusion of logics; that is: Suppose that the strict one-step satisfiability problems for logics with disjoint modal similarity types Λ_i interpreted over functors F_i, for $i = 1, 2$, are both in ExpTime. Then the strict one step satisfiability problem of the fusion (Remark 2.5), with modal similarity type $\Lambda = \Lambda_1 \cup \Lambda_2$ interpreted over $F = F_1 \times F_2$, is in ExpTime as well. To see this, just note that a one-step pair (γ, Θ) over V in the fusion is satisfiable over $F = F_1 \times F_2$ iff for $i = 1, 2.$ the one-step pair $(\gamma \cap \Lambda_i(V), \Theta)$ over V is satisfiable over F_i.

Thus, we obtain by Theorem 6.5 that the satisfiability problem of any combination of the logics mentioned in Proposition 6.6 is in ExpTime; for instance, this holds for the logic of Markov decision processes described in Example 2.1.7.

7. 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. Our method does not require guardedness of fixpoint variables. Prominent examples include the graded μ-calculus, the monotone μ-calculus and its fragment game logic, 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 on-the-fly solving in the spirit of global caching algorithms. Moreover, we have obtained a polynomial bound on minimum branching width in models for all example logics mentioned above.

References

[AHK02] Rajeev Alur, Thomas Henzinger, and Orna Kupferman. Alternating-time temporal logic. *J. ACM*, 49:672–713, 2002. doi:10.1145/585265.585270.

[AHS90] Jiří Adámek, Horst Herrlich, and George Strecker. *Abstract and concrete categories*. Wiley, 1990. Also: Reprints in Theory and Applications of Categories 17 (2006), 1–507. URL: http://www.tac.mta.ca/tac/reprints/articles/17/tr17.pdf.

[AM89] Peter Aczel and Nax Paul Mendler. A final coalgebra theorem. In David H. Pitt, David E. Rydeheard, Peter Dybjer, Andrew M. Pitts, and Axel Poigné, editors, *Category Theory and Computer Science, CTCS 1989*, volume 389 of *LNCS*, pages 357–365. Springer, 1989. doi:10.1007/BFB0018361.

[BdRV01] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2001. doi:10.1017/CBO9781107050884.

[BFL15] Florian Bruse, Oliver Friedmann, and Martin Lange. On guarded transformation in the modal μ-calculus. *Log. J. IGPL*, 23(2):194–216, 2015. doi:10.1093/jigpal/jzu030.

[BKMP19] Paolo Baldan, Barbara König, Christina Mika-Michalski, and Tommaso Padoan. Fixpoint games on continuous lattices. *Proc. ACM Program. Lang.*, 3(POPL):26:1–26:29, 2019. doi:10.1145/3290339.

[BS07] Julian C. Bradfield and Colin Stirling. Modal mu-calculi. In Patrick Blackburn, J. F. A. K. van Benthem, and Frank Wolter, editors, *Handbook of Modal Logic*, volume 3 of *Studies in logic and practical reasoning*, pages 721–756. North-Holland, 2007. doi:10.1016/S1570-2464(07)80015-2.

[BSdV04] Falk Bartels, Ana Sokolova, and Erik P. de Vink. A hierarchy of probabilistic system types. *Theor. Comput. Sci.*, 327(1-2):3–22, 2004. doi:10.1016/J.TCS.2004.07.019.

[BW18] Julian Bradfield and Igor Walukiewicz. The mu-calculus and model checking. In Edmund Clarke, Thomas Henzinger, Helmut Veith, and Roderick Bloem, editors, *Handbook of Model Checking*, pages 871–919. Springer, 2018. doi:10.1007/978-3-319-10575-8.

[Can88] John F. Canny. Some algebraic and geometric computations in PSPACE. In Janos Simon, editor, *Symposium on Theory of Computing, STOC 1988*, pages 460–467. ACM, 1988. doi:10.1145/62212.62257.

[CC79] Patrick Cousot and Radhia Cousot. Constructive versions of Tarski's fixed point theorems. *Pacific J. Math.*, 81(1):43–57, 1979. doi:10.2140/pjm.1979.82.43.

[CIN05] Rance Cleaveland, S. Purushothaman Iyer, and Murali Narasimha. Probabilistic temporal logics via the modal mu-calculus. *Theor. Comput. Sci.*, 342(2-3):316–350, 2005. doi:10.1016/J.TCS.2005.03.048.

[CK16] Souymodip Chakraborty and Joost-Pieter Katoen. On the satisfiability of some simple probabilistic logics. In Martin Grohe, Eric Koskinen, and Natarajan Shankar, editors, *Logic in Computer Science, LICS 2016*, pages 56–65. ACM, 2016. doi:10.1145/2933575.2934526.

[CKP11a] Corina Cîrstea, Clemens Kupke, and Dirk Pattinson. EXPTIME tableaux for the coalgebraic μ-calculus. *Log. Meth. Comput. Sci.*, 7, 2011. doi:10.2168/LMCS-7(3:3)2011.

[CKP+11b] Corina Cîrstea, Alexander Kurz, Dirk Pattinson, Lutz Schröder, and Yde Venema. Modal logics are coalgebraic. *Comput. J.*, 54(1):31–41, 2011. doi:10.1093/COMJNL/BXP004.

[CKW18] Sjoerd Cranen, Jeroen J. A. Keiren, and Tim A. C. Willemse. Parity game reductions. *Acta Informatica*, 55(5):401–444, 2018. doi:10.1007/s00236-017-0301-x.

[DG08] Anuj Dawar and Erich Grädel. The descriptive complexity of parity games. In Michael Kaminski and Simone Martini, editors, *Computer Science Logic, CSL 2008*, volume 5213 of *LNCS*, pages 354–368. Springer, 2008. doi:10.1007/978-3-540-87531-4.

[DL06] Stéphane Demri and Denis Lugiez. Presburger modal logic is PSPACE-complete. In Ulrich Furbach and Natarajan Shankar, editors, *Automated Reasoning, IJCAR 2006*, volume 4130 of *LNCS*, pages 541–556. Springer, 2006. doi:10.1007/11814771.

[DV02] Giovanna D'Agostino and Albert Visser. Finality regained: A coalgebraic study of Scott-sets and multisets. *Arch. Math. Log.*, 41(3):267–298, 2002. doi:10.1007/S001530100110.

[EJ91] E.A. Emerson and C.S. Jutla. Tree automata, mu-calculus and determinacy. In *[1991] Proceedings 32nd Annual Symposium of Foundations of Computer Science*, pages 368–377, 1991. doi:10.1109/SFCS.1991.185392.

[EJ99] E. Allen Emerson and Charanjit S. Jutla. The complexity of tree automata and logics of programs. *SIAM J. Comput.*, 29(1):132–158, 1999. doi:10.1137/S0097539793304741.

[EKRS17] Javier Esparza, Jan Kretínský, Jean-François Raskin, and Salomon Sickert. From LTL and limit-deterministic Büchi automata to deterministic parity automata. In Axel Legay and Tiziana Margaria, editors, *Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2017*, volume 10205 of *LNCS*, pages 426–442. Springer, 2017. doi:10.1007/978-3-662-54577-5.

[ES06] Friedrich Eisenbrand and Gennady Shmonin. Carathéodory bounds for integer cones. *Oper. Res. Lett.*, 34(5):564–568, 2006. doi:10.1016/J.ORL.2005.09.008.

[ESV17] Sebastian Enqvist, Fatemeh Seifan, and Yde Venema. An expressive completeness theorem for coalgebraic modal mu-calculi. *Log. Methods Comput. Sci.*, 13(2), 2017. doi:10.23638/LMCS-13(2:14)2017.

[ESV19] Sebastian Enqvist, Fatemeh Seifan, and Yde Venema. Completeness for μ-calculi: A coalgebraic approach. *Ann. Pure Appl. Log.*, 170(5):578–641, 2019. doi:10.1016/j.apal.2018.12.004.

[FHM90] Ronald Fagin, Joseph Y. Halpern, and Nimrod Megiddo. A logic for reasoning about probabilities. *Inf. Comput.*, 87(1/2):78–128, 1990. doi:10.1016/0890-5401(90)90060-U.

[FL13] Oliver Friedmann and Martin Lange. Deciding the unguarded modal μ-calculus. *J. Appl. Non-Classical Log.*, 23:353–371, 2013. doi:10.1080/11663081.2013.861181.

[FLL13] Oliver Friedmann, Markus Latte, and Martin Lange. Satisfiability games for branching-time logics. *Log. Methods Comput. Sci.*, 9, 2013. doi:10.2168/LMCS-9(4:5)2013.

[FLV10] Gaëlle Fontaine, Raul Andres Leal, and Yde Venema. Automata for coalgebras: An approach using predicate liftings. In Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide, and Paul Spirakis, editors, *Automata, Languages and Programming, ICALP 2010*, volume 6199 of *LNCS*, pages 381–392. Springer, 2010. doi:10.1007/978-3-642-14162-1.

[GHH+23] Oliver Görlitz, Daniel Hausmann, Merlin Humml, Dirk Pattinson, Simon Prucker, and Lutz Schröder. COOL 2 – A generic reasoner for modal fixpoint logics (system description). In Brigitte Pientka and Cesare Tinelli, editors, *Automated Deduction, CADE 2023*, volume 14132 of *LNAI*, pages 1–14. Springer, 2023. Extended version available as arXiv eprint 2305.11015. doi:10.1007/978-3-031-38499-8_14.

[GJLS17] Víctor Gutiérrez-Basulto, Jean Christoph Jung, Carsten Lutz, and Lutz Schröder. Probabilistic description logics for subjective uncertainty. *J. Artif. Intell. Res.*, 58:1–66, 2017. doi:10.1613/jair.5222.

[GN13] Rajeev Goré and Linh Anh Nguyen. Exptime tableaux for *ALC* using sound global caching. *J. Autom. Reason.*, 50(4):355–381, 2013. doi:10.1007/S10817-011-9243-0.

[GTW02] Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *LNCS*. Springer, 2002. doi:10.1007/3-540-36387-4.

[GW09] Rajeev Goré and Florian Widmann. Sound global state caching for $\mathcal{ALC}$ with inverse roles. In Martin Giese and Arild Waaler, editors, *Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX 2009*, volume 5607 of *LNCS*, pages 205–219. Springer, 2009. doi:10.1007/978-3-642-02716-1_16.

[HK97] Michael Huth and Marta Kwiatkowska. Quantitative analysis and model checking. In *Logic in Computer Science, LICS 1997*, pages 111–122. IEEE, 1997. doi:10.1109/LICS.1997.614940.

[HP06] Thomas A. Henzinger and Nir Piterman. Solving games without determinization. In Zoltán Ésik, editor, *Computer Science Logic, CSL 2006*, volume 4207 of *LNCS*, pages 395–410. Springer, 2006. doi:10.1007/11874683_26.

[HS15] Daniel Hausmann and Lutz Schröder. Global caching for the flat coalgebraic μ-calculus. In Fabio Grandi, Martin Lange, and Alessio Lomuscio, editors, *Temporal Representation and Reasoning, TIME 2015*, pages 121–143. IEEE, 2015. doi:10.1109/TIME.2015.15.

[HS19] Daniel Hausmann and Lutz Schröder. Optimal satisfiability checking for arithmetic μ-calculi. In Mikolaj Bojanczyk and Alex Simpson, editors, *Foundations of Software Science and Computation Structures, FOSSACS 2019*, volume 11425 of *LNCS*, pages 277–294. Springer, 2019. doi:10.1007/978-3-030-17127-8_16.

[HS21] Daniel Hausmann and Lutz Schröder. Quasipolynomial computation of nested fixpoints. In Jan Friso Groote and Kim Guldstrand Larsen, editors, *Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2021*, volume 12651 of *LNCS*, pages 38–56. Springer, 2021. doi:10.1007/978-3-030-72016-2_3.

[HSD18] Daniel Hausmann, Lutz Schröder, and Hans-Peter Deifel. Permutation games for the weakly aconjunctive μ-calculus. In Dirk Beyer and Marieke Huisman, editors, *Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2018*, volume 10806 of *LNCS*, pages 361–378. Springer, 2018. doi:10.1007/978-3-319-89963-3_21.

[HSE16] Daniel Hausmann, Lutz Schröder, and Christoph Egger. Global caching for the alternation-free coalgebraic μ-calculus. In Josée Desharnais and Radha Jagadeesan, editors, *Concurrency Theory, CONCUR 2016*, volume 59 of *LIPIcs*, pages 34:1–34:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICS.CONCUR.2016.34.

[KKV01] Valerie King, Orna Kupferman, and Moshe Vardi. On the complexity of parity word automata. In Furio Honsell and Marino Miculan, editors, *Foundations of Software Science and Computation Structures, FOSSACS 2001*, volume 2030 of *LNCS*, pages 276–286. Springer, 2001. doi:10.1007/3-540-45315-6_18.

[KMV20] Clemens Kupke, Johannes Marti, and Yde Venema. Size matters in the modal μ-calculus. *CoRR*, abs/2010.14430, 2020. URL: https://arxiv.org/abs/2010.14430, arXiv:2010.14430.

[KMV22] Clemens Kupke, Johannes Marti, and Yde Venema. Succinct graph representations of μ-calculus formulas. In Florin Manea and Alex Simpson, editors, *Computer Science Logic, CSL 2022*, volume 216 of *LIPIcs*, pages 29:1–29:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.CSL.2022.29.

[Koz83] Dexter Kozen. Results on the propositional μ-calculus. *Theor. Comput. Sci.*, 27:333–354, 1983. doi:10.1016/0304-3975(82)90125-6.

[KP10] Clemens Kupke and Dirk Pattinson. On modal logics of linear inequalities. In Lev Beklemishev, Valentin Goranko, and Valentin Shehtman, editors, *Advances in Modal Logic, AiML 2010*, pages 235–255. College Publications, 2010. URL: http://www.aiml.net/volumes/volume8/Kupke-Pattinson.pdf.

[KPS15] Clemens Kupke, Dirk Pattinson, and Lutz Schröder. Reasoning with global assumptions in arithmetic modal logics. In Adrian Kosowski and Igor Walukiewicz, editors, *Fundamentals of Computation Theory, FCT 2015*, volume 9210 of *LNCS*, pages 367–380. Springer, 2015. doi:10.1007/978-3-319-22177-9.

[KPS22] Clemens Kupke, Dirk Pattinson, and Lutz Schröder. Coalgebraic reasoning with global assumptions in arithmetic modal logics. *ACM Trans. Comput. Log.*, 23(2):11:1–11:34, 2022. doi:10.1145/3501300.

[KSV02] Orna Kupferman, Ulrike Sattler, and Moshe Vardi. The complexity of the graded μ-calculus. In Andrei Voronkov, editor, *Automated Deduction, CADE 02*, volume 2392 of *LNCS*, pages 423–437. Springer, 2002. doi:10.1007/3-540-45620-1_34.

[Lav76] R. Laver. Well-quasi-orders and sets of finite sequences. *Math. Proc. Cambridge Philos. Soc.*, 79:1–10, 1976. doi:10.1017/S030500410005204X.

[LBC+94] David E. Long, Anca Browne, Edmund M. Clarke, Somesh Jha, and Wilfredo R. Marrero. An improved algorithm for the evaluation of fixpoint expressions. In *Computer Aided Verification, CAV 1994*, volume 818 of *LNCS*, pages 338–350. Springer, 1994. doi:10.1007/3-540-58179-0_66.

[LSWZ15] Wanwei Liu, Lei Song, Ji Wang, and Lijun Zhang. A simple probabilistic extension of modal mu-calculus. In Qiang Yang and Michael Wooldridge, editors, *International Joint Conference on Artificial Intelligence, IJCAI 2015*, pages 882–888. AAAI Press, 2015. URL: http://ijcai.org/proceedings/2015.

[Mar75] Donald A. Martin. Borel determinacy. *Annals of Mathematics*, 102(2):363–371, 1975. doi:10.2307/1971035.

[MH84] Satoru Miyano and Takeshi Hayashi. Alternating finite automata on ω-words. *Theor. Comput. Sci.*, 32:321–330, 1984. doi:10.1016/0304-3975(84)90049-5.

[Mio11] Matteo Mio. Probabilistic modal μ-calculus with independent product. In Martin Hofmann, editor, *Foundations of Software Science and Computational Structures, FOSSACS 2011*, volume 6604 of *LNCS*, pages 290–304. Springer, 2011. doi:10.1007/978-3-642-19805-2.

[MM07] Annabelle McIver and Carroll Morgan. Results on the quantitative μ-calculus qmμ. *ACM Trans. Comput. Log.*, 8, 2007. doi:10.1145/1182613.1182616.

[Mos99] Lawrence Moss. Coalgebraic logic. *Ann. Pure Appl. Log.*, 96(1-3):277–317, 1999. doi:10.1016/S0168-0072(98)00042-6.

[MPS09] Robert Myers, Dirk Pattinson, and Lutz Schröder. Coalgebraic hybrid logic. In Luca de Alfaro, editor, *Foundations of Software Science and Computational Structures, FOSSACS 2009*, volume 5504 of *LNCS*, pages 137–151. Springer, 2009. doi:10.1007/978-3-642-00596-1.

[MV15] Johannes Marti and Yde Venema. Lax extensions of coalgebra functors and their logic. *J. Comput. Syst. Sci.*, 81(5):880–900, 2015. doi:10.1016/j.jcss.2014.12.006.

[NW96] Damian Niwinski and Igor Walukiewicz. Games for the μ-calculus. *Theor. Comput. Sci.*, 163:99–116, 1996. doi:10.1016/0304-3975(95)00136-0.

[Par83] Rohit Parikh. Propositional game logic. In *Foundations of Computer Science, FOCS 1983*. IEEE Computer Society, 1983. doi:10.1109/SFCS.1983.47.

[Par85] Rohit Parikh. The logic of games and its applications. *Ann. Discrete Math.*, 24:111–140, 1985. doi:10.1016/S0304-0208(08)73078-0.

[Pat03] Dirk Pattinson. Coalgebraic modal logic: Soundness, completeness and decidability of local consequence. *Theor. Comput. Sci.*, 309:177–193, 2003. doi:10.1016/S0304-3975(03)00201-9.

[Pat04] Dirk Pattinson. Expressive logics for coalgebras via terminal sequence induction. *Notre Dame J. Formal Logic*, 45:19–33, 2004. doi:10.1305/NDJFL/1094155277.

[Pau01] Marc Pauly. *Logic for social software*. PhD thesis, University of Amsterdam, 2001. URL: https://eprints.illc.uva.nl/id/eprint/2029/.

[Pit07] Nir Piterman. From nondeterministic Büchi and Streett automata to deterministic parity automata. *Log. Meth. Comput. Sci.*, 3, 2007. doi:10.2168/LMCS-3(3:5)2007.

[PP03] Marc Pauly and Rohit Parikh. Game logic – an overview. *Stud. Log.*, 75(2):165–182, 2003. doi:10.1023/A:1027354826364.

[PS10] Dirk Pattinson and Lutz Schröder. Cut elimination in coalgebraic logics. *Inf. Comput.*, 208(12):1447–1468, 2010. doi:10.1016/j.ic.2009.11.008.

[RdV99] Jan Rutten and Erik de Vink. Bisimulation for probabilistic transition systems: a coalgebraic approach. *Theor. Comput. Sci.*, 221:271–293, 1999. doi:10.1016/S0304-3975(99)00035-3.

[Rut00] Jan Rutten. Universal coalgebra: A theory of systems. *Theor. Comput. Sci.*, 249:3–80, 2000. doi:10.1016/S0304-3975(00)00056-6.

[Saf88] Shmuel Safra. On the complexity of omega-automata. In *Foundations of Computer Science, FOCS 1988*, pages 319–327. IEEE Computer Society, 1988. doi:10.1109/SFCS.1988.21948.

[Sch86] Alexander Schrijver. *Theory of linear and integer programming*. Wiley Interscience, 1986.

[Sch07] Lutz Schröder. A finite model construction for coalgebraic modal logic. *J. Log. Algebr. Prog.*, 73:97–110, 2007. doi:10.1016/J.JLAP.2006.11.004.

[Sch08] Lutz Schröder. Expressivity of coalgebraic modal logic: The limits and beyond. *Theor. Comput. Sci.*, 390(2-3):230–247, 2008. doi:10.1016/j.tcs.2007.09.023.

[SE89] Robert S. Streett and E. Allen Emerson. An automata theoretic decision procedure for the propositional mu-calculus. *Inf. Comput.*, 81(3):249–264, 1989.

[Sei96] Helmut Seidl. Fast and Simple Nested Fixpoints. *Inf. Process. Lett.*, 59:303–308, 1996. doi:10.1016/0020-0190(96)00130-5.

[SP08] Lutz Schröder and Dirk Pattinson. Shallow models for non-iterative modal logics. In Andreas Dengel, Karsten Berns, Thomas Breuel, Frank Bomarius, and Thomas Roth-Berghofer, editors, *Advances in Artificial Intelligence, KI 2008*, volume 5243 of *LNCS*, pages 324–331. Springer, 2008. doi:10.1007/978-3-540-85845-4.

[SP09] Lutz Schröder and Dirk Pattinson. PSPACE bounds for rank-1 modal logics. *ACM Trans. Comput. Log.*, 10(2):13:1–13:33, 2009. doi:10.1145/1462179.1462185.

[SP10] Lutz Schröder and Dirk Pattinson. Rank-1 modal logics are coalgebraic. *J. Log. Comput.*, 20:1113–1147, 2010. doi:10.1093/logcom/exn096.

[SP11] Lutz Schröder and Dirk Pattinson. Modular algorithms for heterogeneous modal logics via multi-sorted coalgebra. *Math. Struct. Comput. Sci.*, 21:235–266, 2011. doi:10.1017/S0960129510000563.

[SV14] Sven Schewe and Thomas Varghese. Determinising parity automata. In Erzsébet Csuhaj-Varjú, Martin Dietzfelbinger, and Zoltán Ésik, editors, *Mathematical Foundations of Computer Science, MFCS 2014*, volume 8634 of *LNCS*, pages 486–498. Springer, 2014. doi:10.1007/978-3-662-44522-8.

[SV18] Lutz Schröder and Yde Venema. Completeness of flat coalgebraic fixpoint logics. *ACM Trans. Comput. Log.*, 19(1):4:1–4:34, 2018. doi:10.1145/3157055.

[Var89] M.Y. Vardi. On the complexity of epistemic reasoning. In *[1989] Proceedings. Fourth Annual Symposium on Logic in Computer Science*, pages 243–252, 1989. doi:10.1109/LICS.1989.39179.

[Ven04] Yde Venema. Automata and fixed point logics for coalgebras. In Jirí Adámek and Stefan Milius, editors, *Coalgebraic Methods in Computer Science, CMCS 2004*, volume 106 of *ENTCS*, pages 355–375. Elsevier, 2004. doi:10.1016/j.entcs.2004.02.038.

[Ven06] Yde Venema. Automata and fixed point logic: A coalgebraic perspective. *Inf. Comput.*, 204(4):637–678, 2006. doi:10.1016/j.ic.2005.06.003.

[Wil01] Thomas Wilke. Alternating tree automata, parity games, and modal μ-calculus. *Bull. Belg. Math. Soc.*, 8(2):359–391, 2001. doi:10.36045/bbms/1102714178.