# EXPTIME TABLEAUX FOR THE COALGEBRAIC $\mu$-CALCULUS [*]

CORINA CÎRSTEA [a], CLEMENS KUPKE [b], AND DIRK PATTINSON [c]

[a] School of Electronics and Computer Science, University of Southampton
  *e-mail address*: cc2@ecs.soton.ac.uk

[b,c] Department of Computing, Imperial College London
  *e-mail address*: ckupke@doc.ic.ac.uk, dirk@doc.ic.ac.uk

ABSTRACT. The coalgebraic approach to modal logic provides a uniform framework that captures the semantics of a large class of structurally different modal logics, including e.g. graded and probabilistic modal logics and coalition logic. In this paper, we introduce the coalgebraic $\mu$-calculus, an extension of the general (coalgebraic) framework with fixpoint operators. Our main results are completeness of the associated tableau calculus and EXPTIME decidability for guarded formulas. Technically, this is achieved by reducing satisfiability to the existence of non-wellfounded tableaux, which is in turn equivalent to the existence of winning strategies in parity games. Our results are parametric in the underlying class of models and yield, as concrete applications, previously unknown complexity bounds for the probabilistic $\mu$-calculus and for an extension of coalition logic with fixpoints.

## 1. INTRODUCTION

The extension of a modal logic with operators for least and greatest fixpoints leads to a dramatic increase in expressive power [1]. The paradigmatic example is of course the modal $\mu$-calculus [14]. In the same way that the $\mu$-calculus extends the modal logic $K$, one can freely add fixpoint operators to any propositional modal logic, as long as modal operators are monotone. Semantically, this poses no problems, and the interpretation of fixpoint formulas can be defined in a standard way in terms of the semantics of the underlying modal logic.

This apparent simplicity is lost once we move from semantics to syntax: completeness and complexity even of the modal $\mu$-calculus are all but trivial [27, 7], and $\mu$-calculi arising from other monotone modal logics are largely unstudied, with the notable exception of the graded $\mu$-calculus [21]. Here, we improve on this situation, not by providing a new complexity result for a specific fixpoint logic, but by providing a generic and uniform treatment of modal fixpoint logics on the basis of *coalgebraic semantics*. This allows for a generic and uniform treatment of a large class of modal logics and replaces the investigation of a concretely given logic with the study of *coherence conditions* that mediate between the axiomatisation and the (coalgebraic) semantics. The

use of coalgebras conveniently abstracts the details of a concretely given class of models, which is replaced by the class of coalgebras for an (unspecified) endofunctor on sets. Specific choices for this endofunctor then yield specific model classes, such as the class of all Kripke frames or probabilistic transition systems. A property such as completeness or complexity of a specific logic is then automatic once the coherence conditions are satisfied. As it turns out, even *the same* coherence conditions that guarantee completeness and decidability of the underlying modal logic entail the same properties of the ensuing $\mu$-calculus. This immediately provides us with a number of concrete examples: as instances of the generic framework, we obtain not only the known EXPTIME bounds, both for the modal and the graded $\mu$-calculus [7, 21], but also previously unknown EXPTIME bounds for the probabilistic and monotone $\mu$-calculus, and for an extension of coalition logic [18] with fixpoint operators.

Our main technical results are a syntactical characterisation of satisfiability in terms of (non-) existence of closed tableaux and a game-theoretic characterisation of satisfiability that yields an EXPTIME upper bound for the satisfiability problem for guarded formulas. Along the way, we establish a small model theorem. Here, as usual, a formula is called guarded if every fixpoint variable occurs only within the scope of a modal operator. If we assume that every formula can be transformed into an equivalent guarded formula in polynomial time, our EXPTIME decidability result extends to the full coalgebraic $\mu$-calculus. This assumption is generally made in the literature on the modal $\mu$-calculus [15], but a recent paper [11] argues that in fact no algorithm is known that can perform the transformation in polynomial time. Therefore we formulate our EXPTIME-decidability result more restrictive than in [4]. We nevertheless conjecture that our tableau calculus can be used for proving EXPTIME-decidability for the full coalgebraic $\mu$-calculus.

We start by describing a parity game that characterizes model checking for the coalgebraic $\mu$-calculus. As in the model-checking game for the modal $\mu$-calculus (see e.g. [25]), we allow greatest and least fixpoints to be unfolded ad libitum. Truth of a formula in a particular state of a model then follows, if only greatest fixpoints are unfolded infinitely often on the top level along infinite paths, which is captured by a parity condition. The same technique is employed in the construction of tableaux, which we conceptualise as finite directed graphs: closed tableaux witness unsatisfiability of the root formula, provided that along any infinite tableau path one can construct an infinite sequence of formulas (a trace that tracks the evolution of formulas in a tableau) that violates the parity condition. In particular, closed tableaux are finitely represented proofs of the unsatisfiability of the root formula. Soundness of the tableau calculus is established by showing that a winning strategy in the model checking game precludes existence of a closed tableau. Decidability is then established with the help of tableau games, where the adversary chooses a tableau rule, and the player claiming satisfiability chooses one conclusion which effectively constructs a path in a tableau. In order to turn this tableau game into a parity game we combine the game board with the transition function of a deterministic parity word automaton. This automaton checks that on any given play, i.e., on any tableau path, there exists no trace that violates the parity condition. We prove adequacy of the tableau game by constructing a satisfying model from a winning strategy in the tableau game, which makes crucial use of the coherence conditions between the axiomatisation and the coalgebraic semantics. This allows us to determine satisfiability of a fixpoint formula by deciding the associated (parity) tableau game, and the announced EXPTIME upper bound for guarded formulas follows once we can ensure that legality of moves in the tableau game can be decided in exponential time.

**Related Work.** Our treatment is inspired by [17, 26, 24], but we note some important differences. In contrast to [17], we use parity games that directly correspond to tableaux, together with parity automata to detect bad traces. Moreover, owing to the generality of the coalgebraic framework, the model construction here needs to super-impose a coalgebra structure on the relation

induced by a winning strategy. This construction is necessarily different from [24], since we cannot argue by induction on modal rank in the presence of fixpoints. Coalgebraic fixpoint logics are also treated in [26], where an automata theoretic characterisation of satisfiability is presented. We add to this picture by providing complexity results and a complete tableau calculus. Moreover, we use standard syntax for modal operators, which allows us to subsume for instance the graded $\mu$-calculus that cannot be expressed in terms of the $\nabla$-operator used in *op.cit.*

## 2. THE COALGEBRAIC $\mu$-CALCULUS

To keep our treatment fully parametric in the underlying (modal) logic, we define the syntax of the coalgebraic $\mu$-calculus relative to a (fixed) modal similarity type, that is, a set $\Lambda$ of modal operators with associated arities. Throughout, we fix a denumerable set $\mathsf{V}$ of propositional variables. We will only deal with formulas in negation normal form and abbreviate $\overline{\Lambda} = \{\overline{\heartsuit} \mid \heartsuit \in \Lambda\}$ and $\overline{\mathsf{V}} = \{\overline{p} \mid p \in \mathsf{V}\}$. The arity of $\overline{\heartsuit} \in \overline{\Lambda}$ is the same as that of $\heartsuit$. The set $\mathcal{F}(\Lambda)$ of $\Lambda$-formulas is given by the grammar

$$A, B ::= p \mid A \vee B \mid A \wedge B \mid \heartsuit(A_1, \ldots, A_n) \mid \mu p.A \mid \nu p.A$$

where $p \in \mathsf{V} \cup \overline{\mathsf{V}}$, $\heartsuit \in \Lambda \cup \overline{\Lambda}$ is $n$-ary and $\overline{p}$ does not occur in $A$ in the last two clauses. The sets of free and bound variables of a formula are defined as usual, in particular $p$ is bound in $\mu p.A$ and $\nu p.A$. Negation $\overline{\phantom{-}} : \mathcal{F}(\Lambda) \to \mathcal{F}(\Lambda)$ is given inductively by $\overline{\overline{p}} = p$, $\overline{A \wedge B} = \overline{A} \vee \overline{B}$, $\overline{\heartsuit(A_1, \ldots, A_n)} = \overline{\heartsuit}(\overline{A_1}, \ldots, \overline{A_n})$ and $\overline{\mu p.A} = \nu p.\overline{A}[\overline{p} := p]$ and the dual clauses for $\vee$ and $\nu$. If $S$ is a set of formulas, then the collection of formulas that arises by prefixing elements of $S$ by one layer of modalities is denoted by $(\Lambda \cup \overline{\Lambda})(S) = \{\heartsuit(A_1, \ldots, A_n) \mid \heartsuit \in \Lambda \cup \overline{\Lambda} \ n\text{-ary}, A_1, \ldots, A_n \in S\}$. A *substitution* is a mapping $\sigma : V \to \mathcal{F}(\Lambda)$ and $A\sigma$ is the result of replacing all free occurrences of $p \in V$ in $A$ by $\sigma(p)$.

On the semantical side, parametricity is achieved by adopting coalgebraic semantics: formulas are interpreted over $T$-coalgebras, where $T$ is an (unspecified) endofunctor on sets, and we recover the semantics of a large number of logics in the form of specific choices for $T$. To interpret the modal operators $\heartsuit \in \Lambda$, we require that $T$ extends to a $\Lambda$-*structure* and comes with a predicate lifting, that is, a natural transformation of type $[\![\heartsuit]\!] : 2^n \to 2 \circ T^{\mathrm{op}}$ for every $n$-ary modality $\heartsuit \in \Lambda$, where $2 : \mathsf{Set} \to \mathsf{Set}^{\mathrm{op}}$ is the contravariant powerset functor. In elementary terms, this amounts to assigning a set-indexed family of functions $([\![\heartsuit]\!]_X : \mathcal{P}(X)^n \to \mathcal{P}(TX))_{X \in \mathsf{Set}}$ to every $n$-ary modal operator $\heartsuit \in \Lambda$ such that $(Tf)^{-1} \circ [\![\heartsuit]\!]_X(A_1, \ldots, A_n) = [\![\heartsuit]\!]_Y(f^{-1}(A_1), \ldots, f^{-1}(A_n))$ for all functions $f : Y \to X$. If $\heartsuit \in \Lambda$ is $n$-ary, we put $[\![\overline{\heartsuit}]\!]_X(A_1, \ldots, A_n) = (TX) \setminus [\![\heartsuit]\!]_X(X \setminus A_1, \ldots, X \setminus A_n)$. We usually denote a structure by the endofunctor $T$ and leave the definition of the predicate liftings implicit. A $\Lambda$-structure is *monotone* if, for all sets $X$ we have that $[\![\heartsuit]\!]_X(A_1, \ldots, A_n) \subseteq [\![\heartsuit]\!]_X(B_1, \ldots, B_n)$ whenever $A_i \subseteq B_i$ for all $i = 1, \ldots, n$.

In the coalgebraic approach, the role of frames is played by $T$-*coalgebras*, i.e. pairs $(X, \gamma)$ where $X$ is a set (of states) and $\gamma : X \to TX$ is a (transition) function. A $T$-*model* is a triple $(X, \gamma, h)$ where $(X, \gamma)$ is a $T$-coalgebra and $h : \mathsf{V} \to \mathcal{P}(X)$ is a valuation of the propositional variables that we implicitly extend to $\mathsf{V} \cup \overline{\mathsf{V}}$ by putting $h(\overline{p}) = X \setminus h(p)$. For a monotone $\Lambda$-structure $T$ and a $T$-model $M = (X, \gamma, h)$, the *truth set* $[\![A]\!]_M$ of a formula $A \in \mathcal{F}(\Lambda)$ w.r.t. $M$ is given inductively by

$$[\![p]\!]_M = h(p) \quad [\![\mu p.A]\!]_M = \mathsf{LFP}(A_p^M) \quad [\![\nu p.A]\!]_M = \mathsf{GFP}(A_p^M)$$

$$[\![\heartsuit(A_1, \ldots, A_n)]\!]_M = \gamma^{-1} \circ [\![\heartsuit]\!]_X([\![A_1]\!]_M, \ldots, [\![A_n]\!]_M)$$

where $\mathsf{LFP}(A_p^M)$ and $\mathsf{GFP}(A_p^M)$ are the least and greatest fixpoint of the monotone mapping $A_p^M :$ $\mathcal{P}(X) \rightarrow \mathcal{P}(X)$ defined by $A_p^M(U) = [\![A]\!]_{M'}$ where $M' = (X, \gamma, h')$ and $h'(q) = h(q)$ for $q \neq p$ and $h'(p) = U$. We write $M, x \models A$ if $x \in [\![A]\!]_M$ to denote that $A$ is satisfied at $x$. A formula $A \in \mathcal{F}(\Lambda)$ is *satisfiable* w.r.t. a given $\Lambda$-structure $T$ if there exists a $T$-model $M$ such that $[\![A]\!]_M \neq \emptyset$. The mappings $A_p^M$ are indeed monotone in case of a monotone $\Lambda$-structure, which guarantees the existence of fixpoints.

**Example 2.1.**     1. $T$-coalgebras $(X, \gamma : X \rightarrow \mathcal{P}(X))$ for $TX = \mathcal{P}(X)$ are Kripke frames. If $\Lambda = \{\Box\}$ for $\Box$ unary and $\overline{\Box} = \Diamond$, $\mathcal{F}(\Lambda)$ are the formulas of the modal $\mu$-calculus [14], and the structure $[\![\Box]\!]_X(U) = \{V \in \mathcal{P}(X) \mid V \subseteq U\}$ gives its semantics.

2. The syntax of the graded $\mu$-calculus [21] is given (modulo an index shift) by the similarity type $\Lambda = \{\langle n \rangle \mid n \geq 0\}$ where $\overline{\langle n \rangle} = [n]$, and $\langle n \rangle A$ reads as "$A$ holds in more than $n$ successors". In contrast to *op. cit.* we interpret the graded $\mu$-calculus over multigraphs, i.e. coalgebras for the functor $\mathsf{B}$

$$\mathsf{B}(X) = \{f : X \rightarrow \mathbb{N} \mid \mathsf{supp}(f) \text{ finite}\}$$

where $\mathsf{supp}(f) = \{x \in X \mid f(x) \neq 0\}$ is the support of $f$, that extends to a structure

$$[\![\langle n \rangle]\!]_X(U) = \{f \in \mathsf{B}(X) \mid \sum_{x \in U} f(x) > n\} \qquad \text{for } U \subseteq X.$$

Note that this semantics differs from the Kripke semantics for both graded modal logic [10] and the graded $\mu$-calculus. The change of the semantics is needed in order to fit graded modal logic into the coalgebraic framework, because in the standard semantics of graded modal logic we cannot interpret the modalities by natural transformations. Both types of semantics, however, induce the same satisfiability problem: image-finite Kripke frames are multigraphs where each edge has multiplicity one, and the unravelling of a multigraph can be turned into a Kripke frame by inserting the appropriate number of copies of each state. The transformations preserve satisfiability. The fact that the two types of semantics induce the same satisfiability problem makes use of the fact that the graded $\mu$-calculus has the tree-model property ([21]): a formula of the graded $\mu$-calculus is satisfiable on some Kripke frame iff it is satisfiable on a tree of finitely bounded branching degree. Alternatively, the fact that the two satisfiability problems are equivalent can be also obtained from the results in this paper by showing that the tableau calculus for the graded $\mu$-calculus is *sound* over the class of all Kripke frames.

3. The probabilistic $\mu$-calculus arises from the similarity type $\Lambda = \{\langle p \rangle \mid p \in [0,1] \cap \mathbb{Q}\}$ where $\overline{\langle p \rangle} = [p]$ and $\langle p \rangle \phi$ reads as "$\phi$ holds with probability at least $p$ in the next state". The semantics of the probabilistic $\mu$-calculus is given by the structure

$$\mathsf{D}(X) = \{\mu : X \rightarrow_f [0,1] \mid \sum_{x \in X} \mu(x) = 1\} \quad [\![\langle p \rangle]\!]_X(U) = \{\mu \in \mathsf{D}(X) \mid \sum_{x \in U} \mu(x) \geq p\}$$

where $U \subseteq X$ and $\rightarrow_f$ indicates maps with finite support. Coalgebras for $\mathsf{D}$ are precisely image-finite Markov chains, and the finite model property of the coalgebraic $\mu$-calculus that we establish later ensures that satisfiability is independent of image-finite semantics.

4. Formulas of coalition logic over a finite set $N$ of agents [18] arise via $\Lambda = \{[C] \mid C \subseteq N\}$, and are interpreted over game frames, i.e. coalgebras for the functor

$$\mathsf{G}(X) = \{(f, (S_i)_{i \in N}) \mid \prod_{i \in N} S_i \neq \emptyset, f : \prod_{i \in N} S_i \rightarrow X\}$$

which is a class-valued functor, which however fits with the subsequent development. We think of $S_i$ as the set of strategies for agent $i$ and $f$ is an outcome function. The formula $[C]A$ reads as "coalition $C$ can achieve $A$", which is captured by the lifting

$$[\![C]\!]_X(U) = \{(f, (S_i)_{i \in N}) \in \mathsf{G}(X) \mid \exists (s_i)_{i \in C} \forall (s_i)_{i \in N \setminus C} f((s_i)_{i \in N}) \in U\}$$

for $U \subseteq X$. The induced coalgebraic semantics is precisely the standard semantics of coalition logic, ie., the formula $[C]A$ holds at a state $x$ if all agents $i$ in the coalition $C$ can choose a strategy $s_i$ at $x$ such that, for all possible strategy choices of agents in $N \setminus C$ at position $x$, the play proceeds to a state $x'$ that satisfies property $A$.

5. Finally, the similarity type $\Lambda = \{\Box\}$ of monotone modal logic [2] has a single unary $\Box$ (we write $\overline{\Box} = \Diamond$) and interpret the ensuing language over monotone neighbourhood frames, that is, coalgebras for the functor / structure

$$\mathcal{M}(X) = \{Y \subseteq \mathcal{P}(\mathcal{P}(X)) \mid Y \text{ upwards closed}\} \quad [\![\Box]\!]_X(U) = \{Y \in \mathcal{M}(X) \mid U \in Y\}$$

for $U \subseteq X$ which recovers the standard semantics in a coalgebraic setting [12].
It is readily verified that all structures above are indeed monotone.

## 3. THE MODEL-CHECKING GAME

We start by characterising the satisfaction relation between states of a model and formulas of the coalgebraic $\mu$-calculus in terms of a two-player parity game that we call the *model checking game*. This characterisation will be the main technical tool for establishing soundness and completeness of an associated tableau calculus.

The game that we are about to describe generalises [25, Theorem 1, Chapter 6] to the coalgebraic setting, and is a variant of the game used in [6]. We begin by fixing our terminology concerning parity games.

A *parity game* played by $\exists$ (Éloise) and $\forall$ (Abélard) is a tuple $\mathcal{G} = (B_\exists, B_\forall, E, \Omega)$ where $B = B_\exists \cup B_\forall$ is the disjoint union of *positions* owned by $\exists$ and $\forall$, respectively, $E \subseteq B \times B$ indicates the allowed moves, and $\Omega : B \to \omega$ is a (parity) map with finite range. An infinite sequence $(b_0, b_1, b_2, \dots)$ of positions is called *bad* if $\max\{k \mid k = \Omega(b_i) \text{ for infinitely many } i \in \omega\}$ is odd.

A *play* in $\mathcal{G}$ is a finite or infinite sequence of positions $(b_0, b_1, \dots)$ with the property that $(b_i, b_{i+1}) \in E$ for all $i$, i.e. all moves are legal, and $b_0$ is the *initial position* of the play. A *full play* is either infinite, or a finite play ending in a position $b_n$ where $E[b_n] = \{b \in B \mid (b_n, b) \in E\} = \emptyset$, i.e. no more moves are possible. A finite play is lost by the player who cannot move, and an infinite play $(b_0, b_1, \dots)$ is lost by $\exists$ (and won by $\forall$) iff $(b_0, b_1, \dots)$ is bad.

A *strategy* in $\mathcal{G}$ for a player $P \in \{\exists, \forall\}$ is a partial function that maps all plays that end in a position $b \in B_P$ of $P$ with $E[b] \neq \emptyset$ to a position $b' \in B$ such that $(b, b') \in E$. Intuitively, a strategy determines a player's next move, depending on the history of the game in all positions where the player can move. Given a strategy $s$ for player $P$ in $\mathcal{G}$ we say that a play $(b_0, \dots, b_i, \dots)$ of $\mathcal{G}$ is *played according to $s$* if for all proper prefixes $b_0 \dots b_i$ of $\pi$ with $b_i \in B_P$ we have $s(b_0 \dots b_i) = b_{i+1}$. A strategy for a player $P \in \{\exists, \forall\}$ is called *history-free* or *positional* if it only depends on the last position of a play. Formally, a *history-free strategy* for player $P \in \{\exists, \forall\}$ is a partial function $s : B_P \rightharpoonup B$ such that $s(b)$ is defined iff $E[b] \neq \emptyset$, in which case $(b, s(b)) \in E$. A play $(b_0, b_1, \dots)$ is *played according to $s$* if $b_{i+1} = s(b_i)$ for all $i$ with $b_i \in B_P$, and $s$ is a *winning strategy* from position $b \in B$ if $P$ wins all plays with initial position $b$ that are played according to $s$.

It is known that parity games are history-free determined [8, 16] and that winning regions can be decided in UP ∩ co-UP [13].

**Theorem 3.1.** [13] *At every position $b \in B_\exists \cup B_\forall$ in a parity game $\mathcal{G} = (B_\exists, B_\forall, E, \Omega)$ one of the players has a history-free winning strategy. Furthermore, for every $b \in B_\exists \cup B_\forall$, it can be determined in time $O\left(d \cdot m \cdot \left(\frac{n}{\lfloor d/2 \rfloor}\right)^{\lfloor d/2 \rfloor}\right)$ which player has a winning strategy from position $b$, where $n$, $m$ and $d$ are the size of $B$, $E$ and the range of $\Omega$, respectively.*

We will now introduce the model checking game as a parity game. The model checking game is played on pairs $(A, x)$ where $A$ is a formula and $x$ is a state, and (informally) $\forall$ tries to demonstrate that $x \not\models A$ whereas $\exists$ claims the opposite. The formulation of the game relies on formulas being *clean* (no variable occurs both free and bound, or is bound more than once) and *guarded* (bound variables only occur within the scope of modal operators). In the model checking game, we will only encounter a finite set of formulas, those that lie in the *closure* of the initial formula. The size of the closure will play a crucial role in our main complexity result because it yields an upper bound for the size of our tableau game that characterizes *satisfiability* of a formula. The formal definitions are as follows:

**Definition 3.2.** A set $\Gamma \subseteq \mathcal{F}(\Lambda)$ of formulas is *closed* if $B \in \Gamma$ whenever $B$ is a subformula of some $A \in \Gamma$ and $A[p := \eta p.A] \in \Gamma$ if $\eta p.A \in \Gamma$, where $\eta \in \{\mu, \nu\}$. The *closure* of $\Gamma$ is the smallest closed set $\mathrm{Cl}(\Gamma)$ for which $\Gamma \subseteq \mathrm{Cl}(\Gamma)$.

A formula $A \in \mathcal{F}(\Lambda)$ is *guarded* if, for all subformulas $\eta p.B$ of $A$, $p$ only appears in the scope of a modal operator within $B$, and $A$ is *clean* if the sets of free and bound variables of a formula are disjoint and if no two distinct occurrences of fixpoint operators in $A$ bind the same variable. A finite set of formulas $\Gamma$ is guarded if every element of $\Gamma$ is guarded and $\Gamma$ is clean if the formula $\bigwedge_{A \in \Gamma} A$ is clean.

In the model checking game, the unfolding of fixpoint formulas gives rise to infinite plays, and we have to ensure that all infinite plays that cycle on an outermost $\mu$-variable are lost by $\exists$ (who claims that the formula(s) under consideration are satisfied), as this would correspond to the infinite unfolding of a least fixpoint. This is where the parity map comes in: formulas of the form $\mu p.A$ are assigned odd priorities and, dually, $\nu A.p$ an even priority. To make sure that $\exists$ only looses those plays that cycle on the unfolding of an *outermost* $\mu$-variable, we require that the assignment of priorities is compatible with the subformula ordering.

**Definition 3.3.** A *parity map* for a finite, clean set of formulas $\Gamma$ is a function $\Omega : \mathrm{Cl}(\Gamma) \to \omega$ with finite range for which $\Omega(A) = 0$ unless $A$ is of the form $\eta p.B$, $\eta \in \{\mu, \nu\}$, $\Omega(A)$ is odd (even) iff $A$ is of the form $\mu p.B$ ($\nu p.B$), and $\Omega(\eta_1 p_1.B_1) \leq \Omega(\eta_2 p_2.B_2)$ whenever $\eta_1 p_1.B_1$ is a subformula of $\eta_2 p_2.B_2$, where $\eta_1, \eta_2 \in \{\mu, \nu\}$.

It is easy to see that every clean set of formulas admits a parity function.

**Lemma 3.4.** *If $\Gamma \subseteq \mathcal{F}(\Lambda)$ is finite and clean, then $\Gamma$ admits a parity function whose range is bounded by the cardinality of $\mathrm{Cl}(\Gamma)$.*

*Proof.* By induction on the well-founded ordering generated by

$$\Gamma, \Delta < \Gamma, A \text{ iff } A \notin \Delta \subseteq \mathsf{subf}(A)$$

where $\mathsf{subf}(A)$ denotes the subformulas of $A$. If $\Gamma$ contains a top-level conjunction, disjunction or propositional variable, then the claim follows by induction hypothesis. Now suppose that $\Gamma =$

$\mu p.A, \Gamma'$. By induction hypothesis, we obtain a parity function $\Omega' : \mathrm{Cl}(A, \Gamma') \to \omega$ that we may extend to a parity function $\Omega : \mathrm{Cl}(\Gamma) \to \omega$ by putting

$$\Omega(B) = \begin{cases} m & B = \mu p.A \\ \Omega'(B) & B \in \mathrm{Cl}(A, \Gamma') \\ 0 & \text{otherwise} \end{cases}$$

where $m$ is odd and $m > \Omega'(B)$ for all $B \in \mathrm{Cl}(A, \Gamma')$. The case $\Gamma = \nu p.A, \Gamma'$ can be treated in a similar fashion. □

Given a parity function, we can define the following parity game, the winning regions of which characterise satisfiability. We parametrise the model checking game in a *set* of formulas which will enable us to use it to prove soundness and completeness of the tableau calculus (which operates on sets of formulas) that we introduce later.

**Definition 3.5.** Suppose that $M = (X, \gamma, h)$ is a $T$-model, $\Gamma \subseteq \mathcal{F}(\Lambda)$ is finite, clean and guarded, and $\Omega$ is a parity map for $\Gamma$. The *model checking game* $\mathcal{MG}_\Gamma(M)$ is the parity game whose positions and admissible moves are given in the following table,

| Position: b | Player | Admissible moves: $E[b]$ |
|---|---|---|
| $(p, x), x \in h(p)$ | $\forall$ | $\emptyset$ |
| $(p, x), x \notin h(p)$ | $\exists$ | $\emptyset$ |
| $(\eta p.A(p), x)$ for $\eta \in \{\mu, \nu\}$ | $\exists$ | $\{(A[p = \eta p.A(p)], x)\}$ |
| $(A_1 \vee A_2, x)$ | $\exists$ | $\{(A_1, x), (A_2, x)\}$ |
| $(A_1 \wedge A_2, x)$ | $\forall$ | $\{(A_1, x), (A_2, x)\}$ |
| $(\heartsuit(A_1, \ldots, A_n), x)$ | $\exists$ | $\{(\heartsuit(A_1, \ldots, A_n), (U_1, \ldots, U_n)) \mid$ $U_1, \ldots, U_n \subseteq X, \gamma(x) \in [\![\heartsuit]\!]_X(U_1, \ldots, U_n)\}$ |
| $(\heartsuit(A_1, \ldots, A_n), (U_1, \ldots, U_n))$ | $\forall$ | $\{(A_i, x) \mid 1 \leq i \leq n, x \in U_i\}$ |

where $p \in \mathsf{V} \cup \overline{\mathsf{V}}$, $\heartsuit \in \Lambda \cup \overline{\Lambda}$, $A, A_1, \ldots, A_n \in \mathrm{Cl}(\Gamma)$ are $\Lambda$-formulas, $x \in X$ are states and $U_i \subseteq X$ are state sets. The parity function of $\mathcal{MG}_\Gamma(M)$ is given by $\Omega'(A, x) = \Omega(A)$ for $A \in \mathrm{Cl}(\Gamma)$ and $x \in X$, and $\Omega'(\_) = 0$ otherwise.

It is easy to see that any two parity functions for a given set of formulas induce the same winning region for both players. We therefore speak of *the* model checking game given by a set of formulas. Evidently, the model checking game is an extension of the boolean satisfiability game with fixpoints and modal operators. When the game reaches a fixpoint formula, that is, a position of type $(\eta p.A, x)$, this fixpoint is simply unfolded, and its nature (least or greatest fixpoint) and nesting depth of the formula fixpoint formula is recorded by the parity function. To show that a state $x$ satisfies a modal formula $\heartsuit(A_1, \ldots, A_n)$, $\exists$ needs to select sets $U_1, \ldots, U_n$ (that we think of as a subset of the truth sets of the $A_i$'s) so that the state $x$ is being mapped by $\gamma$ into the lifting of $U_1, \ldots, U_n$. Subsequently, $\forall$ may challenge this choice and select an index $1 \leq i \leq n$ and require that $\exists$ demonstrates that the formula $A_i$ is satisfied at an aribitrary element of $U_i$ (and thus corroborate that we may think of $U_i$ as the truth set of $A_i$). To prove that the model checking game characterises satisfiability, we make crucial use of monotonicity, as the $U_i$ under-approximate the truth sets of the $A_i$. The announced generalisation of [25, Theorem 1, Chapter 6] now takes the following form:

**Theorem 3.6.** *For $\Gamma$ finite, clean and guarded, a $T$-model $M = (X, \gamma, h)$, $A \in \mathrm{Cl}(\Gamma)$ and $x \in X$, $\exists$ has a winning strategy in $\mathcal{MG}_\Gamma(M)$ from position $(A, x)$ iff $M, x \models A$.*

*Proof.* The proof is by induction on $A$, and similar to the proof of adequacy of the game semantics for the coalgebraic $\mu$-calculus [26, Theorem 1]. It should be noted that the model-checking game in

*loc. cit.* has slightly diferent moves in positions that correspond to fixpoint formulas: in a position of the form $(\eta p.A(p), x)$, the only available choice is to move to $(A(p), x)$, and if a position of the form $(p, y)$ is reached later, then the only option is to move to $(A(p), y)$. However, one can show that both ways of treating fixpoint formulas in the model-checking game are equivalent. We only treat the case $A = \heartsuit(A_1, \ldots, A_n)$; all others are as in *loc. cit.* .

First suppose that $M, x \models \heartsuit(A_1, \ldots, A_n)$. By induction hypothesis, $\exists$ has a winning strategy from position $(A, x')$ if and only if $M, x' \models A$ for all subformulas $A$ of $\{A_1, \ldots, A_n\}$. These winning strategies can be extended to provide a winning strategy from $\heartsuit(A_1, \ldots, A_n)$ by stipulating that $\exists$ move to $(\heartsuit(A_1, \ldots, A_n), ([\![A_1]\!]_M, \ldots, [\![A_n]\!]_M)$. Now assume that $\exists$ has a winning strategy from position $(\heartsuit(A_1, \ldots, A_n), x)$ in $\mathcal{MG}_\Gamma(M)$ under which $\exists$ moves to position $(\heartsuit(A_1, \ldots, A_n), (U_1, \ldots, U_n))$ from position $(\heartsuit(A_1, \ldots, A_n), x)$. By induction hypothesis, we have that $x_i \models A_i$ for all $x_i \in U_i$ so that $U_i \subseteq [\![A_i]\!]_M$ and hence $\gamma(x) \in [\![\heartsuit]\!]([\![A_1]\!]_M, \ldots, [\![A_n]\!]_M)$ by monotonicity of $[\![\heartsuit]\!]$ whence $x \models \heartsuit(A_1, \ldots, A_n)$. $\qquad\square$

## 4. Tableaux for the coalgebraic $\mu$-calculus

In this section, we characterise *satisfiability* in terms of non-existence of *closed tableaux*. Given that our approach is parametric both in the model class over which we interpret formulas (embodied by the endofunctor) and the modal operators (given by the similarity type) that we use, our tableau system will be parametric in a set of modal tableau rules. Our tableaux will be constructed by applying the standard rules for deconstructing propositional connectives, the modal rules that are supplied as a parameter, and unfolding of fixpoints. To ensure soundness and completeness of the ensuing calculus, we need to ensure two properties:

(1) the supplied set of modal rules has to describe the model class in a sound and complete way
(2) topmost least fixpoints are only unfolded finitely often.

For the first property, we introduce *coherence conditions* between the proof rules and the semantics that will guarantee completeness. For the second property, we need to consider *traces* of formulas along the paths of the tableau and again use a parity function to determine whether outermost $\mu$s are unfolded only finitely often. As the unfolding of fixpoints may create infinite branches, we conceptualise a tableau as a graph. A closed tableau is then constructed according to the given rules so that outermost least fixpoints are unfolded infinitely many times along any path through the tableau.

We begin by describing the coherence conditions that will guarantee soundness and completeness of the modal rules. These rules describe the relationship between states and (coalgebraic) successors, are of a particularly simple form, and are formulated in terms of sequents.

**Definition 4.1.** A $\Lambda$-*tableau sequent*, or just *sequent*, is a finite set of $\Lambda$-formulas. We write $\mathsf{S}(\Lambda)$ for the set of $\Lambda$-sequents. If $\Gamma \in \mathsf{S}(\Lambda)$ we write $\mathsf{S}(\Gamma) = \{\Delta \in \mathsf{S}(\Lambda) \mid \Delta \subseteq \mathrm{Cl}(\Gamma)\}$ for the set of sequents over the closure of $\Gamma$.

We identify a formula $A \in \mathcal{F}(\Lambda)$ with the singleton set $\{A\}$, and write $\Gamma; \Delta = \Gamma \cup \Delta$ for the union of $\Gamma, \Delta \in \mathsf{S}(\Lambda)$ as before. Substitution extends to sequents via $\Gamma\sigma = \{A\sigma \mid A \in \Gamma\}$. A *monotone one-step tableau rule* for a similarity type $\Lambda$ is of the form

$$\frac{\Gamma_0}{\Gamma_1 \quad \ldots \quad \Gamma_n}$$

where $\Gamma_0 \in (\Lambda \cup \overline{\Lambda})(V)$ and $\Gamma_1, \ldots, \Gamma_n \subseteq V$ for some set $V \subseteq \mathsf{V}$ of propositional variables, every propositional variable occurs at most once in $\Gamma_0$ and all variables occurring in one of the $\Gamma_i$'s $(i > 0)$ also occur in $\Gamma_0$.

Monotone tableau rules do not contain negated propositional variables, which are not needed to axiomatise (the class of models induced by) monotone $\Lambda$ structures. The restriction on occurrences of propositional variables is unproblematic, as variables that occur in a conclusion but not in the premise and multiple occurrences of variables in the premise can always be eliminated. The set of one-step tableau rules is the only parameter in the construction of tableaux for coalgebraic fixpoint logics. The coherence conditions relate rule sets with the interpretation of modal operators purely on the level of properties of states (subsets of a set $X$) and properties of successors (subsets of $TX$).

**Definition 4.2.** Let $V \subseteq \mathsf{V}$ be a set of propositional variables. The *interpretation* of a propositional sequent $\Gamma \subseteq V \cup \overline{V}$ with respect to a set $X$ and a valuation $\tau : V \to \mathcal{P}(X)$ is given by $[\![\Gamma]\!]_{X,\tau} = \bigcap \{\tau(p) \mid p \in \Gamma\}$, and the interpretation $[\![\Gamma]\!]_{TX,\tau} \subseteq TX$ of a modalised sequent $\Gamma \subseteq (\Lambda \cup \overline{\Lambda})(V)$ is

$$[\![\Gamma]\!]_{TX,\tau} = \bigcap \{[\![\heartsuit]\!]_X(\tau(p_1), \ldots, \tau(p_n)) \mid \heartsuit(p_1, \ldots, p_n) \in \Gamma\}.$$

If $T$ is a $\Lambda$-structure, then a set $\mathsf{R}$ of monotone tableau rules for $\Lambda$ is *one-step tableau complete* (resp. *sound*) with respect to $T$ if $[\![\Gamma]\!]_{TX,\tau} \neq \emptyset$ if (only if) for all $\Gamma_0/\Gamma_1, \ldots, \Gamma_n \in \mathsf{R}$ and all $\sigma : V \to V$ with $\Gamma_0 \sigma \subseteq \Gamma$, there exists $1 \leq i \leq n$ such that $[\![\Gamma_i \sigma]\!]_{X,\tau} \neq \emptyset$, whenever $\Gamma \subseteq (\Lambda \cup \overline{\Lambda})(V)$ and $\tau : V \to \mathcal{P}(X)$.

Informally speaking, a set $\mathsf{R}$ of one-step tableau rules is one-step tableau complete if a modalised sequent $\Gamma$ is satisfiable whenever a rule that matches $\Gamma$ has a satisfiable conclusion. Some care has to be taken to ensure monotonicity of one-step rules in concrete examples, in particular for the graded and the probabilistic $\mu$-calculus. In order to obtain monotone rules for these logics, we need to insist that rule conclusions only contain prime implicants to avoid non-monotone occurrences of propositional variables. This ensures that we avoid a (non-monotone) conclusion consisting of e.g. $\Gamma; p$ and $\Gamma; \overline{p}$.

**Definition 4.3.** Suppose $I$ is a finite (index) set. A *prime implicant* of a boolean function $f : \{0, 1\}^I \to \{0, 1\}$ is a partial valuation $p : I \rightharpoonup \{0, 1\}$ with minimal domain of definition so that $f$ evaluates to 1 under all total extensions of $p$. Given a family $(p_i)_{i \in I}$ of propositional variables, every partial valuation $p : I \rightharpoonup \{0, 1\}$ (and hence every prime implicant) induces a sequent

$$\Gamma_p = \{p_i \mid p(i) = 1\} \cup \{\overline{p}_i \mid p(i) = 0\}.$$

Now consider $k \in \mathbb{Z}$, a family $(r_i)_{i \in I}$ of integers and a family $(p_i)_{i \in I}$ of propositional variables over the same index set. For $I = I_0 \cup I_1$, we let

$$\sum_{i \in I_0} r_i p_i + \sum_{i \in I_1} r_i \overline{p_i} < k = \{\Gamma_p \mid p \text{ prime implicant of } f\}$$

for the set of sequents induced by the prime implicants of the boolean function $f : \{0, 1\}^I \to \{0, 1\}$ defined by $f(v) = 1 \iff \sum_{i \in I_0} r_i v(i) + \sum_{i \in I_1} r_i(1 - v(i)) < k$.

In other words, the set of prime implicants of a boolean function corresponds to the reduced disjunctive normal form of the associated propositional formula. The notation $\sum_i r_i p_i < k$ introduced above allows us to read a linear inequality involving propositional variables as a set of sequents (that we will later use as the conclusion of a one-step rule). If we think of the propositional variables $p_i$ as denoting subsets $U_i$ of some set $X$, then the set of all points $x \in X$ that satisfy the inequality $\sum_i \mathbb{1}_{U_i}(x) \leq k$ is precisely the set of points that satisfies the induced collection of sequents. (We

write $\mathbb{1}_U : X \to \{0,1\}$ for the characteristic function of a subset $U \subseteq X$). For one-step rules formulated in terms of linear inequalities, we need this property to establish completeness.

**Lemma 4.4.** *Suppose $X$ is a set and $\tau : \mathsf{V} \to \mathcal{P}(X)$ is a valuation of propositional variables. Then $x$ satisfies one of the elements of $\sum_{i \in I_0} r_i p_i + \sum_{i \in I_1} r_i \overline{p_i} < k$ iff $\sum_{i \in I_0}^n r_i \mathbb{1}_{\tau(p_i)}(x) + \sum_{i \in I_1} r_i \mathbb{1}_{X \setminus \tau(p_i)}(x) < k$. That is,*

$$\sum_{i \in I_0} r_i \mathbb{1}_{\tau(p_i)}(x) + \sum_{i \in I_1} r_i \mathbb{1}_{X \setminus \tau(p_i)}(x) < k \iff x \in \bigcup \{ [\![\Gamma]\!]_{(X,\tau)} \mid \Gamma \in \sum_{i \in I_0} r_i p_i + \sum_{i \in I_0} r_i \overline{p_i} < k \}$$

*for all $x \in X$, whenever $r_1, \ldots, r_n, k \in \mathbb{Z}$.*

*Proof.* First suppose that $x \in \bigcup \{ [\![\Gamma]\!]_{(X,\tau)} \mid \Gamma \in \sum_{i \in I_0} r_i p_i + \sum_{i \in I_1} r_i \overline{p_i} < k \}$. Then there exists a prime implicant $p : I \rightharpoonup \{0,1\}$ of the function $f$ given by $f(v) = 1 \iff \sum_{i \in I_0} r_i v(i) + \sum_{i \in I_1} r_i (1 - v(i)) < k$ such that $x \in [\![\Gamma_p]\!]_{(X,\tau)}$. Then the function $c : I_0 \cup I_1 \to \{0,1\}$ given by

$$c(i) := \begin{cases} 1 & \text{if } x \in \tau(p_i) \text{ and } i \in I_0 \\ 1 & \text{if } x \notin \tau(p_i) \text{ and } i \in I_1 \\ 0 & \text{otherwise.} \end{cases}$$

extends $p$ and therefore $f(c) = 1$ whence

$$\sum_{i \in I_0} r_i \mathbb{1}_{\tau(p_i)}(x) + \sum_{i \in I_1} r_i \mathbb{1}_{X \setminus \tau(p_i)}(x) = \sum_{i \in I_0} r_i c(i) + \sum_{i \in I_1} r_i c(i) < k.$$

Now suppose that $\sum_{i \in I_0} r_i \mathbb{1}_{\tau(p_i)}(x) + \sum_{i \in I_1} r_i \mathbb{1}_{X \setminus \tau(p_i)}(x) < k$ and consider the valuation

$$v(i) = \begin{cases} 1 & \text{if } x \in \tau(p_i), i \in I_0 \\ 1 & \text{if } x \notin \tau(p_i), i \in I_1 \\ 0 & \text{otherwise.} \end{cases}$$

We have that $f(v) = 1$ and therefore obtain a prime implicant $p : I \rightharpoonup \{0,1\}$ of $f$ such that $v$ extends $p$ and $x \in [\![\Gamma_p]\!]_{(X,\tau)}$. $\qquad\square$

This finishes our discussion of prime implicants and we are ready to have a look at several examples. We use the following one-step rules to axiomatise the model classes introduced in Example 2.1.

**Example 4.5.** (1) The (standard) modal logic of Kripke frames is axiomatised by all the instances of

$$(K)\frac{\Diamond p_0; \Box p_1; \ldots; \Box p_n}{p_0; p_1; \ldots; p_n}$$

where $n \geq 0$.

(2) the set of one-step rules associated with graded modal logic (and the graded $\mu$-calculus, interpreted over finitely branching multigraphs) can be axiomatised by the rule schema

$$(G)\frac{\langle k_1 \rangle p_1; \ldots; \langle k_n \rangle p_n; [l_1]q_1; \ldots; [l_m]q_m}{\sum_{j=1}^m s_j \overline{q_j} - \sum_{i=1}^n r_i p_i < 0}$$

where $m, n \geq 0$ and $r_i, s_j \in \mathbb{N} \setminus \{0\}$ and $\sum_{i=1}^n r_i(k_i + 1) \geq 1 + \sum_{j=1}^m s_j l_j$.

(3) The set of rules associated to the probabilistic $\mu$-calculus comprises all instances of

$$(P)\frac{\langle a_1 \rangle p_1; \ldots; \langle a_n \rangle p_n; [b_1]q_1; \ldots; [b_m]q_m}{\sum_{j=1}^m s_j \overline{q_j} - \sum_{i=1}^n r_i p_i < k}$$

where $m, n \geq 0$, $r_i, s_j \in \mathbb{N} \setminus \{0\}$ and $\sum_{i=1}^n r_i a_i - \sum_{j=1}^m s_j b_j \leq k$ if $n > 0$ and $-\sum_{j=1}^m s_j b_j < k$ if $n = 0$.

(4) For coalition logic, we have all instances of

$$(C_1)\frac{[C_1]p_1;\ldots;[C_n]p_n}{p_1;\ldots;p_n} \qquad (C_2)\frac{[C_1]p_1;\ldots;[C_n]p_n;\overline{[D]}q;\overline{[N]}r_1;\ldots;\overline{[N]}r_m}{p_1;\ldots;p_n;q;r_1;\ldots;r_m}$$

where again $m, n \geq 0$. Both rules are subject to the side condition that the $C_i$ are disjoint. For $(C_2)$ we moreover require $C_i \subseteq D$.

(5) Finally, the rule set associated to monotone modal logic contains the single rule

$$(M)\frac{\Box p;\Diamond q}{p;q}.$$

In the rule schemas $(G)$ and $(P)$, we note that $\sum_i r_i a_i < k$ is a set of (propositional) sequents, and therefore qualifies as the conclusion of a tableau rule. To ensure monotonicity, we have to ensure that no literal appears negatively. This is a direct consequence of the following:

**Lemma 4.6.** *Suppose that $p$ is a prime implicant of the boolean function $f : \{0,1\}^I \to \{0,1\}$ given by $f(v) = 1$ iff $\sum_{i \in I} r_i v_i < k$, where $(r_i)_{i \in I}$ is a sequence of nonzero integers. Then $p(i) = 1$ or undefined whenever $r_i < 0$ and analogously, $p(i) = 0$ or undefined whenever $r_i > 0$. In particular, all instances of $(G)$ and $(P)$ are monotone.*

*Proof.* We only demonstrate the first item, the second is analogous. Suppose, for a contradiction, that $p(i) = 0$ and $r_i < 0$. Then, removing $i$ from the domain of definition of $p$ yields a function $q : I \rightharpoonup \{0,1\}$ such that all total extensions $e$ of $q$ still satisfy $f(e) = 1$, contradicting the minimality of $p$. $\square$

It is easy to see that every $\Lambda$-structure admits a one-step sound and complete set of one-step tableau rules. While this demonstrates that our approach is applicable to all conceivable $\Lambda$-structures, the challenge of finding a tractable representation of the rule set remains, which is crucial for a complexity analysis. An adaptation of [22, Theorem 17] to the setting of monotone tableau rules shows that one-step complete rule sets always exist.

**Proposition 4.7.** *Every monotone $\Lambda$-structure admits a one-step tableau sound and one-step tableau complete set of monotone tableau rules.*

*Proof.* Suppose that $T$ is a monotone $\Lambda$-structure. We show that there exists a set $\mathsf{R}$ of monotone tableau rules so that $\mathsf{R}$ is one-step tableau sound and one-step tableau complete for $T$, essentially by showing that the set of all monotone one-step sound rules is indeed one-step complete. We let $\mathsf{R}$ consist of all monotone tableau rules $\Gamma_0/\Gamma_1,\ldots,\Gamma_n$ that satisfy

$$\llbracket\Gamma_1\rrbracket_{X,\tau} = \cdots = \llbracket\Gamma_n\rrbracket_{X,\tau} = \emptyset \implies \llbracket\Gamma_0\rrbracket_{TX,\tau} = \emptyset$$

for all sets $X$ and valuations $\tau : V \to \mathcal{P}(X)$. We claim that $\mathsf{R}$ is one-step tableau sound and one-step tableau complete.

First, for one-step tableau soundness, suppose that $\tau : V \to \mathcal{P}(X)$ is given and $\llbracket\Gamma\rrbracket_{TX,\tau} \neq \emptyset$ for some $\Gamma \subseteq (\Lambda \cup \overline{\Lambda})(V)$. For $\Gamma_0/\Gamma_1,\ldots,\Gamma_n \in \mathsf{R}$ and a renaming $\sigma : V \to V$ such that $\Gamma_0\sigma \subseteq \Gamma$, we have to show that $\llbracket\Gamma_i\sigma\rrbracket_{X,\tau} \neq \emptyset$ for some $1 \leq i \leq n$. Assume, for a contradiction, that $\llbracket\Gamma_i\sigma\rrbracket_{X,\tau} = \emptyset$ for all $1 \leq i \leq n$. Then, for $\tau'(p) = \tau(\sigma(p))$ we have $\llbracket\Gamma_i\rrbracket_{X,\tau'} = \emptyset$ for all $1 \leq i \leq n$ so that $\llbracket\Gamma_0\sigma\rrbracket_{TX,\tau} = \llbracket\Gamma_0\rrbracket_{TX,\tau'} = \emptyset$, contradicting $\llbracket\Gamma_0\sigma\rrbracket_{TX,\tau} \supseteq \llbracket\Gamma\rrbracket_{TX,\tau} \neq \emptyset$.

For one-step tableau completeness, we directly show the contrapositive. Assume that $\llbracket\Gamma\rrbracket_{TX,\tau} = \emptyset$ for some set $X$ and some valuation $\tau : V \to \mathcal{P}(X)$. We show that, in this case, there exists $\Gamma_0/\Gamma_1\ldots,\Gamma_n \in \mathsf{R}$ and $\sigma : V \to V$ such that $\Gamma_0\sigma \subseteq \Gamma$ and $\llbracket\Gamma_i\sigma\rrbracket_{X,\tau} = \emptyset$.

So suppose that $[\![\Gamma]\!]_{TX,\tau} = \emptyset$ and consider the set

$$S = \{\Delta \subseteq V_\Gamma \mid [\![\Delta]\!]_{X,\tau} = \emptyset\}$$

where $V_\Gamma$ denotes the set of propositional variables occurring in $\Gamma$. If we let $S = \{\Gamma_1, \ldots, \Gamma_n\}$, it suffices to show that $\Gamma/\Gamma_1, \ldots, \Gamma_n \in \mathsf{R}$. So suppose $\rho : V \to \mathcal{P}(Y)$ is a valuation such that $[\![\Gamma_1]\!]_{Y,\rho} = \cdots = [\![\Gamma_n]\!]_{Y,\rho} = \emptyset$. We show that $[\![\Gamma]\!]_{TY,\rho} = \emptyset$. To this effect, we claim that there exists a function $f : Y \to X$ such that $y \in \rho(p) \implies f(y) \in \tau(p)$ for all $p \in V_\Gamma$. For if not, there exists $y \in Y$ for which a suitable $f(y)$ cannot be found, i.e. for all $x \in X$ we may find $p_x \in V_\Gamma$ such that $x \notin \tau(p_x)$ but $y \in \rho(p_x)$. For the sequent $\Delta = \{p_x \mid x \in X\}$ we then obtain $[\![\Delta]\!]_{X,\tau} = \emptyset$ whence $\Delta \in S$ but $y \in [\![\Delta]\!]_{Y,\rho}$, contradicting $[\![\Gamma_i]\!]_{Y,\rho} = \emptyset$ for all $i = 1, \ldots, n$.

By construction, the function $f$ satisfies $\rho(p) \subseteq f^{-1}(\tau(p))$ for all $p \in V_\Gamma$, which gives, by monotonicity of the $\Lambda$-structure $T$, that

$$[\![\Gamma]\!]_{TY,\rho} \subseteq [\![\Gamma]\!]_{TY,f^{-1}\circ\tau} = (Tf)^{-1}([\![\Gamma]\!]_{TX,\tau}) = \emptyset$$

as required, where the second equality is by naturality of predicate liftings. $\quad\square$

In the examples, we can find concrete (and tractable) representations of one-step complete rule sets.

**Proposition 4.8.** *The rule sets introduced in Example 4.5 are both one-step tableau sound and one-step tableau complete with respect to the corresponding structures defined in Example 2.1.*

*Proof.* It is straightforward to see that a set of monotone rules is one-step tableau complete iff the set of proof rules arising by negating and swapping premise and conclusion is one-step sound and strictly one-step complete in the sense of [24], where soundness and completeness is established for the dual rule sets. The case of graded and probabilistic modal logic additionally requires to invoke Lemma 4.4 together with Lemma 3.18 of *op. cit.* $\quad\square$

We now introduce the set of tableau rules that we are using to axiomatise the coalgebraic $\mu$-calculus. As to be expected, these rules are parametric in a set of one-step rules, and we will instantiate our results to the logics introduced in Example 2.1 with help of the previous proposition. Along with the tableau rules, we also introduce rule blueprints and rule representations that will aid us in the definition of paths through a tableau later on.

**Definition 4.9.** The set $\mathsf{TR}$ of *tableau rules* induced by a set $\mathsf{R}$ of one-step rules contains the propositional and fixpoint rules, the modal rules $(\mathsf{m})$ and the axiom $(\mathsf{rule})$ below:

$$(\wedge)\frac{\Gamma; A \wedge B}{\Gamma; A; B} \quad (\vee)\frac{\Gamma; A \vee B}{\Gamma; A \quad \Gamma; B} \quad (\mathsf{f})\frac{\Gamma; \eta p.A}{\Gamma; A[p := \eta p.A]} \quad (\mathsf{m})\frac{\Gamma_0\sigma, \Delta}{\Gamma_1\sigma \ldots \Gamma_n\sigma} \quad (\mathsf{Ax})\frac{}{\Gamma, A, \overline{A}}$$

Here $\Gamma_0/\Gamma_1 \ldots \Gamma_n \in \mathsf{R}$ and $\sigma : \mathsf{V} \to \mathcal{F}(\Lambda)$ is a substitution satisfying $\sharp(\Gamma_0) = \sharp(\Gamma_0\sigma)$ where $\sharp$ denotes cardinality. The formulas $A \wedge B$, $A \vee B$ and $\eta p.A$ are called *principal* in the rules $(\wedge)$, $(\vee)$ and $(\mathsf{f})$. A *rule blueprint* is of the form $A \wedge B$, $A \vee B$, $\eta p.A$, $(A, \overline{A})$ or $(r, \sigma)$, where $r \in \mathsf{R}$ and $\sigma : V_0 \to \mathcal{F}(\Lambda)$ is a substitution satisfying $\sharp(\Gamma_0) = \sharp(\Gamma_0\sigma)$ and $V_0 \subseteq \mathsf{V}$ is the set of variables occurring in $r$. We write $\mathcal{B}(\mathsf{R})$ for the set of rule blueprints over the set $\mathsf{R}$ of one-step rules. A *rule representation* is a tuple $(\Gamma, \flat)$ where $\Gamma \in \mathsf{S}(\Lambda)$ and $\flat$ is a rule blueprint that satisfies

- $\flat \in \Gamma$ if $\flat$ is of the form $A \wedge B$, $A \vee B$ or $\eta p.A$
- $A, \overline{A} \in \Gamma$ if $\flat = (A, \overline{A})$
- $\Gamma_0\sigma \subseteq \Gamma$ if $\flat = (r, \sigma)$ and $r = \Gamma_0/\Gamma_1 \ldots \Gamma_n$.

Each rule representation $(\Gamma, \flat)$ induces a tableau rule $\rho(\Gamma, \flat) \in \mathsf{TR}$ given by

$$\rho(\Gamma, A \wedge B) = \frac{\Gamma}{A, B, \Gamma'} \qquad\qquad \rho(\Gamma, A \vee B) = \frac{\Gamma}{A, \Gamma' \quad B, \Gamma'}$$

$$\rho(\Gamma, \eta p.A) = \frac{\Gamma}{A[p := \eta p.A], \Gamma'} \qquad\qquad \rho(\Gamma, (r, \sigma)) = \frac{\Gamma}{\Gamma_1 \sigma \ldots \Gamma_n \sigma}$$

$$\rho(\Gamma, (A, \overline{A})) = \frac{\Gamma}{\quad}$$

where $\Gamma' = \Gamma \setminus \{\flat\}$ in the first three clauses, and $r = \Gamma_0 / \Gamma_1 \ldots \Gamma_n$ in the fourth clause.

The restriction $\sharp(\Gamma_0 \sigma) = \sharp(\Gamma_0)$ on instances of one-step rules ensures that the substitution does not identify literals in the premise of a one-step rule, which implies that only finitely many modal rules are applicable to any sequent. Similarly, because of the restriction $\sharp(\Gamma_0 \sigma) = \sharp(\Gamma_0)$ on substitutions and on the size of the domain of such substitutions in rule representations, it is also easy to see that for any $\Gamma \in \mathsf{S}(\Lambda)$ the set of rule representations $(\Gamma, \flat)$ is finite. This will enable us to deduce decidability, and indeed complexity bounds later. We will, however, need to require that the set of modal rules is contraction closed in order to ensure completeness of the restricted calculus.

**Remark 4.10.** Alternatively, we could also prove soundness and completeness for the tableau calculus without the restriction $\sharp(\Gamma_0 \sigma) = \sharp(\Gamma_0)$ and without requiring contraction closure for the set of modal rules. In this case, in order to obtain decidability, we would have to require contraction-closedness of the modal rules. This is essential for proving that we can restrict the calculus to (finitely many) instances $(r, \sigma)$ of modal rules with non-identifying substitutions $\sigma$.

Our definition of rule blueprints and rule representations may seem a bit bureaucratic at first sight, so some comments are in order. If we understand a tableau as a two-player game where $\forall$ plays a tableau rule and $\exists$ selects a conclusion, the winning condition for $\exists$ stipulates that least fixpoints are not unfolded infinitely often. This condition is formalised in terms of the evolution of formulas along a path in a tableau, which in turn necessitates that we can re-construct the rules applied to tableau nodes. This is achieved by annotating each tableau node with a rule blueprint. Together with the node label, the blueprint forms a rule representation which in turn induces a rule. We use this mechanism for two reasons:

- for propositional rules and the fixpoint rule, the rule blueprint records the principal formula, that we need to track to define traces later. Moreover, we can distinguish between the different conclusions of the induced rule, and
- for modal rules, the rule blueprint is an unsubstituted one-step rule, which allows us to track (unsubstituted) propositional variables, which is again needed for the definition of traces.

The usefulness of the blueprints and rule representations will become clearer in Definition 4.12 where we define the set of traces through a tableau path. We are now ready to introduce the notion of tableau that we will use throughout the paper. As fixpoint rules generate infinite paths, we formalise tableaux as finite, rooted graphs. As a consequence, *closed* tableaux are finitely represented proofs of the unsatisfiability of the root formula.

**Definition 4.11.** A *tableau* for a clean, guarded sequent $\Gamma \in \mathsf{S}(\Lambda)$ is a finite, directed, rooted and labelled graph $(N, K, R, \ell, \alpha)$ where $N$ is the set of nodes, $K \subseteq N \times N$ is the set of edges, $R$ is the root node and $\ell : N \to \mathsf{S}(\Gamma)$ is a labelling function such that $\ell(R) = \Gamma$ and $\alpha : N \rightharpoonup \mathcal{B}(\mathsf{R})$ is a partial function (that we think of as an annotation) satisfying

- $\alpha(n)$ is defined iff there exists a tableau rule with premise $\ell(n)$ iff $K(n) \neq \emptyset$.
- if $\Gamma_0 / \Gamma_1 \ldots \Gamma_n = \rho(\ell(n), \alpha(n))$ then $\{\Gamma_1, \ldots, \Gamma_n\} = \{\ell(n') \mid n' \in K(n)\}$

where $K(n) = \{n' \mid (n, n') \in K\}$ and $\rho$ is as in Definition 4.9.

In other words, tableaux are sequent-labelled graphs where a rule has to be applied at a node if the node label matches a rule premise, and no rule may be applied otherwise. The purpose of the annotation $\alpha$ is to record which rule (if any) has been applied at a particular node. To keep track of whether least fixpoints are unfolded infinitely often, we record the *unsubstituted* one-step rule (together with a substitution) at modal nodes, as we need to track the evolution of formulas along one-step rules, where propositional variables may become identified by a substitution. Moreover, it may be the case that two different one-step rules generate the same rule instance: both rules $\heartsuit p/p$ and $\heartsuit p, \heartsuit q/p$ generate the instance $\heartsuit A/A$. As we will be required to traverse infinite loops in a tableaux to ensure that only greatest fixponits are unfolded infinitely often, we need to ensure that the identity of a rule does not change when nodes are encountered multiple times.

The reader might wonder why we make a distinction between nodes in a tableau and their labels. The technical reason for this is that we need to run an automaton in parallel to the tableau, so that the same sequent may be associated with different automata states (see the definition of the tableau game in Section 5). Informally speaking, we have to allow for enough paths through a tableau to ensure completeness. We can view a tableau as a strategy of $\forall$ in this tableau game, where $\forall$ tries to prove that a given sequent is not satisfiable. Accordingly, a closed tableau will correspond to a winning strategy for him in the tableau game. An identification of nodes and sequents in a tableau would mean that the corresponding strategy of $\forall$ in the tableau game would only depend on the set of formulas with which a position of $\forall$ is labeled. We cannot guarantee, however, that $\forall$ has a winning strategy of this special kind, even if he has some winning strategy. Therefore, in order to be able to represent arbitrary strategies of $\forall$ in the tableau game as tableaux, we have to have the possibility to distinguish between nodes and their labels. The only restriction we make is that the tableau graph is finite, i.e. we only consider strategies of $\forall$ with bounded memory.

Our goal is to show that a formula $A \in \mathcal{F}(\Lambda)$ is satisfiable iff no tableau for $A$ ever closes. In a setting without fixpoints, a tableau is closed iff all leaves are labelled with axioms. Here we also need to consider infinite paths, and ensure that only greatest fixpoints are unfolded infinitely often at the top level of an infinite path. As in [17], this necessitates to consider the set of *traces* through a given tableau. Informally, a trace records the evolution (by application of tableau rules) of a single formula through a tableau. Formally, we associate binary relations with tableau rules, and traces arise by sequencing these relations.

**Definition 4.12.** Suppose that $\mathsf{T} = (N, K, R, \ell, \alpha)$ is a tableau for $\Gamma$. A *path* through $\mathsf{T}$ is a finite or infinite sequence

$$\pi : n_0 \xrightarrow{c_0} n_1 \xrightarrow{c_1} n_2 \ldots$$

where $n_0 = R$, $n_{i+1} \in K(n_i)$ and $c_i \in \mathbb{N}$ satisfying that $\ell(n_{i+1})$ is the $c_i$-th conclusion of the rule represented by $(\ell(n_i), \alpha(n_i))$. A path is called *complete* if it is infinite or if it ends at a node $n \in N$ with $K[n] = \emptyset$.

A *trace* through a path $\pi$ is a finite or infinite sequence of formulas $(A_0, A_1, \ldots)$ such that $A_i \in \ell(n_i)$ and $(A_i, A_{i+1}) \in \mathsf{Tr}(\ell(n_i), \alpha(n_i), c_i)$ where the relations $\mathsf{Tr}(\Gamma, \flat, i) \subseteq \mathcal{F}(\Lambda) \times \mathcal{F}(\Lambda)$ are given as follows:

- $\mathsf{Tr}(\Gamma, A_1 \wedge A_2, 1) = \{(A_1 \wedge A_2, A_1), (A_1 \wedge A_2, A_2)\} \cup \mathsf{Diag}(\Gamma \setminus \{A \wedge B\})$
- $\mathsf{Tr}(\Gamma, A_1 \vee A_2, i) = \{(A_1 \vee A_2, A_i)\} \cup \mathsf{Diag}(\Gamma \setminus \{A_1 \vee A_2\})$ for $i = 1, 2$.
- $\mathsf{Tr}(\Gamma, \eta p.A, 1) = \{(\eta p.A, A[p := \eta p.A])\} \cup \mathsf{Diag}(\Gamma \setminus \{\eta p.A\})$
- $\mathsf{Tr}(\Gamma, (r, \sigma), i) = \{(\heartsuit(p_1, \ldots, p_n)\sigma, p_j\sigma) \mid \heartsuit(p_1, \ldots, p_n) \in \Gamma_0, p_j \in \Gamma_i\}$ where $r = \Gamma_0/\Gamma_1 \ldots \Gamma_n$.

Here $\mathsf{Diag}(X) = \{(x, x) \mid x \in X\}$ is the diagonal on a set $X$. The triples $(\Gamma, \flat, i)$ where $(\Gamma, \flat)$ is a rule representation, $i \in \mathbb{N}$ and $\rho(\Gamma, \flat)$, the rule represented by $(\Gamma, \flat)$, has at least $i$ conclusions,

are called *trace tiles*. Finally, a tableau $\mathsf{T}$ with root node labelled by $\Gamma$ is *closed*, if the end node of all finite paths through $\mathsf{T}$ of maximal length that starts in the root node is labelled with a tableau axiom, and every infinite path starting in the root node carries at least one bad trace with respect to a parity function $\Omega$ for $\Gamma$.

Informally, a path through a tableau is a sequence of nodes, together with the information which rule has been applied to nodes, and we cannot have a path that ends in a node to which $(\mathsf{Ax})$ was applied. As for the construction of tableaux, the construction of traces requires that we pick the same conclusion every time a node is traversed. While in the instance $A \vee B, A, B / A, B$ of $(\vee)$, both conclusions are identified, they are not equivalent from the point of view of traces, as the 'left' conclusion continues the trace from $A \vee B$ to $A$ whereas the right conclusion continues the same trace to $B$. This difficulty does not arise in [17] where tableaux are formalised as sibling-ordered trees, and the rule blueprints used here serve essentially the same purpose. The traces through a path are calculated using the so-called trace tiles. A trace tile records which rule has been applied in a node that is visited by the path and through which of the successors of the node that path is continuing. It should be noted that for $\Gamma \in \mathsf{S}(\Lambda)$ the set

$$\Sigma_\Gamma = \{(\Delta, \flat, i) \mid (\Delta, \flat, i) \text{ is a trace tile and } \Delta \in \mathsf{S}(\Gamma)\}$$

is finite because, as remarked after Definition 4.9, for each $\Delta \in \mathsf{S}(\Gamma)$ there are only finitely many rule representations $(\Gamma, \flat)$. We stress this fact, because later on we will use $\Sigma_\Gamma$ as alphabet of the parity automaton that is essential for the definition of our tableau game.

**Example 4.13.** Assume that we have a tableau where $(\vee)$ has been applied at the node labelled with $A \vee \mu p.B; C$ and $(\mathsf{f})$ has been applied at the node labelled with $\mu p.\Diamond B; C$. (We identify nodes and their labels here for simplicity.) Then the path

$$A \vee \mu p.B; C \xrightarrow{\ 2\ } \mu p.\Diamond B; C \xrightarrow{\ 1\ } \Diamond B[p := \mu p.B]; C \dots$$

supports the traces $(A \vee \mu p.\Diamond B, \mu p.\Diamond B, \Diamond B[p := \mu p.B], \dots)$ and $(C, C, C, \dots)$. Note that there is no trace on this path that starts with $A$.

We now continue the development of the general theory and first establish soundness of the tableau calculus: satisfiable sequents cannot have closed tableaux. This relies on Theorem 3.6, as a winning strategy for $\exists$ in the model checking game can be used to construct a path through any tableau that carries a bad trace.

**Theorem 4.14.** *Let $\mathsf{R}$ be a one-step tableau complete set of monotone rules for the modal similarity type $\Lambda$, and let $\Gamma \in \mathsf{S}(\Lambda)$ be clean and guarded. If $\Gamma$ is satisfiable in some model $M = (X, \gamma, h)$, then no closed tableau for $\Gamma$ exists.*

*Proof.* Consider a model $M = (X, \gamma, h)$ and $x \in X$ such that $M, x \models \Gamma$ and let $\mathsf{T} = (N, K, R, \ell, \alpha)$ be a tableau for $\Gamma$. As $M, x \models \Gamma$, Theorem 3.6 implies that $\exists$ has a history-free winning strategy $g$ in $\mathcal{MG}_\Gamma = \mathcal{MG}_\Gamma(M)$ from all positions $(B, x)$ of the game board with $B \in \Gamma$. We now establish that there exists a complete path and an associated sequence of model states satisfying the formulas on this path that can be contracted to a play in the model checking game. More precisely, we establish the existence of a path $\pi = n_0 c_0 n_1 c_1 \dots n_l c_l \dots$ through $\mathsf{T}$ and a sequence $\chi = x_0 x_1 \dots x_l \dots$ of states that satisfy

  (i) $n_0 = R$ and $x_0 = x$ and $M, x_i \models \ell(n_i)$ whenever $n_i$ is defined,
 (ii) for each trace $\tau = B_0 B_1 \dots B_l \dots$ through $\pi$ there exists a play $(A_0, y_0)(A_1, y_1) \dots$ (where we do not record the positions that have subsets of the model as second component) that is played according to $g$ and there is an increasing sequence $0 = s_0 < s_1 < \dots$ of indices

such that $B_{s_1} B_{s_2} \cdots = A_0 A_1 \ldots$ and $y_0 y_1 \cdots = x_{s_0} x_{s_1} \ldots$ where $B_i = B_{s_j}$ and $x_i = x_{s_j}$ whenever $s_j \leq i < s_{j+1}$.

Once this claim is established, it follows that $\mathsf{T}$ cannot be closed: consider the path $\pi$ just constructed. If $\pi$ is finite, the label $\Delta$ of the last node of $\pi$ cannot be a tableau axiom, as $\Delta$ is satisfiable by construction. In case $\pi$ is infinite, every trace $\tau$ through $\pi$ induces a $\mathcal{MG}_\Gamma$-play that is played according to $\exists$'s winning strategy $g$ which implies that $\tau$ is not bad. Taken together, this shows that $\mathsf{T}$ cannot be closed, so it remains to establish the claim.

We construct the required path $\pi$ and the sequence $\chi$ of states in a step-by-step fashion, starting at the root of the tableau and at the state $x$, i.e. we put $n_0 = R$ and $x_0 = x$. So suppose that a path $\pi = n_0 c_0 \ldots c_{j-1} n_j$ and a sequence of model states $x_0 \ldots x_j$ satisfying (i) and (ii) above have already been constructed, and $\pi$ is not yet complete. We distinguish cases on the rule $r = \rho(\ell(n_j), \alpha(n_j))$ applied at (the last) node $n_j$.

We begin with the case where $r = \Delta; D_1 \vee D_2 / \Delta; D_1 \quad \Delta; D_2$ is an instance of the disjunction rule. In this case, we can find tableau nodes $m_1$ and $m_2$ with $\ell(m_1) = \Delta, D_1$ and $\ell(m_2) = \Delta, D_2$ and $K(n_j) \supseteq \{m_1, m_2\}$. Suppose that $g(D_1 \vee D_2, x_j) = (D_i, x_j)$ for $i \in \{1, 2\}$. We put $c_j = i$, $n_{j+1} = m_i$ and $x_{j+1} = x_j$. Then the extended path $\pi' = \pi c_j n_{j+1}$ and $x_0 x_1 \ldots x_j x_{j+1}$ satisfy condition (i) of our claim. Obviously we have $x_{j+1} \models \Delta$. Furthermore, $x_{j+1} \models D_i$ as $(D_i, x_{j+1})$ is a winning position of $\exists$ in $\mathcal{MG}_\Gamma$. Thus, as $\ell(n_{j+1}) = \Delta; D_i$, we have $x_{j+1} \models \ell(n_{j+1})$ as required. To see that (ii) also holds, consider a trace $\tau' = B_0 \ldots B_j B_{j+1}$ through $\pi'$ and let $P = (A_0, y_0)(A_1, y_1) \ldots (B_j, y_k)$ be the partial play of $\mathcal{MG}_\Gamma$ that is associated to $\tau = B_0 \ldots B_j$ and that is played according to $g$. If $B_j \neq D_1 \vee D_2$ we have $B_j = B_{j+1}$ and $P$ can be chosen as the corresponding $\mathcal{MG}_\Gamma$-play for $\tau'$. Otherwise, if $B_j = D_1 \vee D_2$, we have $B_{j+1} = D_i$ and we extend $P$ to $(A_0, y_0)(A_1, y_1) \ldots (D_1 \vee D_2, y_k)(D_i, y_{k+1})$ with $y_{k+1} = y_k$. This $\mathcal{MG}_\Gamma$-play now satisfies condition (ii) of our claim.

The cases where $r$ is an instance of the conjunction or fixpoint rules are similar (even easier, as these rules only have one conclusion). So suppose that $r$ is an instance of a modal rule. That is, $r = \rho(\ell(n_j), \alpha(n_j))$ with $\alpha(n_j) = (r, \sigma)$ for some rule $\Delta / \Delta_1, \cdots, \Delta_s$ with $\Delta \sigma \subseteq \ell(n_j)$ and $K(n_j) \supseteq \{m_1, \ldots, m_s\}$ with $\ell(m_i) = \Delta_i \sigma$ for $i \in \{1, \ldots, s\}$. We define a valuation $\tau : V_\Delta \to \mathcal{P}(X)$ on the set $V_\Delta$ of variables occurring in $\Delta$ by stipulating that $\tau(p) = U_k$ where the (unique) occurrence of $p = p_k$ is in the formula $\heartsuit(p_1, \ldots, p_r) \in \Delta$ and $g(D, x_j) = (D, (U_1, \ldots, U_r))$ with $D = \heartsuit(\sigma(p_1), \ldots, \sigma(p_r))$. As $g$ is winning for $\exists$ in $\mathcal{MG}_\Gamma$ at position $(D', x_j)$ for all $D' \in \Delta \sigma$, it follows that $\gamma(x_j) \in [\![\Delta]\!]_{TX,\tau}$, which implies that $[\![\Delta]\!]_{TX,\tau} \neq \emptyset$.

By one-step tableau completeness, $[\![\Delta_i]\!]_{X,\tau} \neq \emptyset$ for some $i \in \{1, \ldots, s\}$. We now extend $\pi$ to a path $\pi' = \pi m_i$ and let $x_{j+1}$ be an arbitrary element of $[\![\Delta_i]\!]_{X,\tau}$. Now consider a trace $\tau'$ through $\pi'$ that ends in some formula $A$ with $A = \sigma(p_A)$ for some $p_A \in \Delta_i$. Then, by Definition 4.12, $\tau'$ is of the form $\tau A$ where $\tau$ is a trace through $\pi$ ending in a formula of the form $B = \heartsuit(p_1, \ldots, p_n)\sigma$, $\heartsuit(p_1, \ldots, p_n) \in \Delta$, and $p_A = p_k$ for some $k \in \{1, \ldots, n\}$.

By assumption, there exists a corresponding $\mathcal{MG}_\Gamma$-play, played according to $g$, that ends in position $(\heartsuit(p_1, \ldots, p_n)\sigma, x_j)$. This play can be extended by $\exists$ moving to $g(\heartsuit(p_1, \ldots, p_n)\sigma, x_j) = (\heartsuit(p_1, \ldots, p_n)\sigma, (U_1, \ldots U_n))$. We extend this play letting $\forall$ move to $(A, x_{j+1})$. The latter move is legitimate as $\sigma(p_k) = A$ and because $x_{j+1} \in [\![\Delta_i]\!]_{X,\tau} = \bigcap_{p \in \Delta_i} \tau(p) \subseteq \tau(p_k) \in \{U_1, \ldots, U_n\}$. It remains to note that for every formula $A' \in \Delta_i$ there exists a trace through $\pi'$ that ends in $A'$, and therefore also a possibly partial $\mathcal{MG}_\Gamma$-play according to $\exists$'s winning strategy $g$ ending at $(A', x_{j+1})$. This implies that for all $A' \in \Delta_i$, $(A', x_{j+1})$ is a winning position for $\exists$ in $\mathcal{MG}_\Gamma$, and hence $M, x_{j+1} \models A'$ for all $A' \in \Delta_i$. This finishes the proof of the claim and hence that of the theorem. $\qquad \square$

**Example 4.15.** Consider the following formula of the coalitional $\mu$-calculus

$$[C]\nu X.(p \wedge \overline{[N]}X) \wedge [D]\mu Y.(\overline{p} \vee [D]Y)$$

stating that "coalition $C$ can achieve that, from the next stage onwards, $p$ holds irrespective of the strategies used by other agents, and coalition $D$ can ensure (through suitable strategies used in the long term) that $\overline{p}$ holds after some finite number of steps". Here, we assume that $C, D \subseteq N$ are such that $C \cap D = \emptyset$. Define a parity map $\Omega$ for the above formula by $\Omega(\nu X.(p \wedge \overline{[N]}X)) = 2$, $\Omega(\mu Y.(\overline{p} \vee [D]Y)) = 1$ and $\Omega(A) = 0$ otherwise. The unsatisfiability of this formula is witnessed by the following closed tableau:

$$
\begin{array}{c}
[C]B \wedge [D]A \\
\hline
[C]B \,;\, [D]A \\
\hline
B \,;\, A \\
\hline
p \wedge \overline{[N]}B \,;\, A \\
\hline
p \wedge \overline{[N]}B \,;\, \overline{p} \vee [D]A \\
\hline
p \,;\, \overline{[N]}B \,;\, \overline{p} \vee [D]A \\
\end{array}
$$

$$\underline{p \,;\, \overline{[N]}B \,;\, \overline{p}} \qquad\qquad p \,;\, \overline{[N]}B \,;\, [D]A$$

where $B = \nu X.(p \wedge \overline{[N]}X)$, $A = \mu Y.(\overline{p} \vee [D]Y)$ and where we omitted the annotation $\alpha$ because in this case $\alpha$ can be easily deduced from the structure of the tableau. For example, the annotation for the root node is equal to $[C]B \wedge [D]A$ and for the child of the root the annotation is equal to $(\frac{[C]p;[D]q}{p;q}, \sigma)$ where $\sigma : \{p, q\} \to \{A, B\}$ is a substitution with $\sigma(p) = B$ and $\sigma(q) = A$.

Any finite path through this tableau ends in an axiom, and the only infinite path contains the trace

$$[C]B \wedge [D]A, \; [D]A, \; A, \; \overline{A, \; \overline{p} \vee [D]A, \; \overline{p} \vee [D]A, \; [D]A, \; A}$$

where the overlined sequence is repeated ad infinitum. This trace is bad with respect to $\Omega$, as $\Omega(A) = 1$ and $A$ is the only fixpoint formula that occurrs infinitely often.

## 5. THE TABLEAU GAME

We now introduce the tableau game associated to a clean and guarded sequent $\Gamma$, and use it to characterise the (non-)existence of closed tableaux in terms of winning strategies in the tableau game. For the entire section, we fix a modal similarity type $\Lambda$ and a set R of monotone tableau rules that is both one-step sound and complete. The idea underlying the tableau game is that $\forall$ intends to construct a closed tableau for a given set of formulas $\Gamma$, while $\exists$ wants to demonstrate that any tableau constructed by $\forall$ contains a path $\pi$ that violates the closedness condition. As infinite plays of the tableau game correspond to paths through a tableau, an infinite play should be won by $\exists$ if it does not carry a bad trace, that is, outermost least fixpoints are only unfolded finitely often. To be able to see this tableau game as a parity game, we therefore need a mechanism to detect bad traces, and we employ parity word automata for this task. Board positions in the ensuing tableau game will therefore be sequent / automata state pairs, with the priority of a board position being determined by the parity function of the automaton. In particular, this will ensure that winning strategies of $\exists$ in the tableau game do not generate bad traces. We start our discussion of the tableau game by recalling some basic notions concerning parity word automata.

**Definition 5.1.** Let $\Sigma$ be a finite alphabet. A *non-deterministic parity $\Sigma$-word automaton* is a quadruple $\mathbb{A} = (Q, a_I, \delta : Q \times \Sigma \to \mathcal{P}(Q), \Omega)$ where $Q$ is the set of states of $\mathbb{A}$, $a_I \in Q$ is

the initial state, $\delta$ is the transition function, and $\Omega : Q \to \omega$ is a (parity) function. Given an infinite word $\gamma = c_0 c_1 c_2 c_3 \dots$ over $\Sigma$, a run of $\mathbb{A}$ on $\gamma$ is a sequence $\rho = a_0 a_1 a_2 \dots \in Q^\omega$ such that $a_0 = a_I$ and for all $i \in \omega$ we have $a_{i+1} \in \delta(a_i, c_i)$. A run $\rho$ is *accepting* if $\rho$ is not a bad sequence with respect to $\Omega$. We say that $\mathbb{A}$ *accepts* an infinite $\Sigma$-word $\gamma$ if there exists an accepting run $\rho$ of $\mathbb{A}$ on $\gamma$. Finally we call $\mathbb{A}$ *deterministic* if $\delta(a, c)$ is a one-element set for all $(a, c) \in Q \times \Sigma$.

In other words, a parity word automaton is deterministic if its transition function has type $Q \times \Sigma \to Q$. To develop the tableau game, we use parity word automata over trace tiles (cf. Definition 4.12) to detect the existence of bad traces through infinite plays. We now establish the existence of such automata, together with a bound on both the state set and the range of the parity function.

**Lemma and Definition 5.2.** Let $\Gamma \in \mathsf{S}(\Lambda)$ be a clean, guarded sequent, and let $\Sigma_\Gamma$ denote the set of trace tiles $(\Delta, \flat, i)$ with $\Delta \in \mathsf{S}(\Gamma)$. There exists a deterministic parity $\Sigma_\Gamma$-word automaton $\mathbb{A}_\Gamma = (Q_\Gamma, a_\Gamma, \delta_\Gamma, \Omega')$ such that $\mathbb{A}_\Gamma$ accepts an infinite sequence $(t_0, t_1, \dots) \in \Sigma_\Gamma^\infty$ of trace tiles iff there is no sequence of formulas $(A_0, A_1, \dots)$ with $(A_i, A_{i+1}) \in \mathsf{Tr}(t_i)$ which is a bad trace with respect to a parity function for $\Gamma$. Moreover, the index of $\mathbb{A}$ and the cardinality of $Q$ are bounded by $p(|\mathrm{Cl}(\Gamma)|)$ and $2^{p(|\mathrm{Cl}(\Gamma)|)}$ for a polynomial $p$, respectively. Such an automaton $\mathbb{A}$ is called a $\Gamma$-*parity automaton*.

*Proof.* We start by constructing a non-deterministic parity automaton that accepts $w = t_0 t_1 t_2 \dots \in \Sigma_\Gamma^\omega$ iff $w$ does contain a sequence $A_0 A_1 \dots \in \mathrm{Cl}(\Gamma)^\omega$ that is bad w.r.t. $\Omega$ and satisfies $(A_i, A_{i+1}) \in \mathsf{Tr}(t_i)$ for all $i \in \mathbb{N}$. We put $Q' = \mathrm{Cl}(\Gamma) \cup \{a_I\}$ where we assume that $a_I \notin \mathrm{Cl}(\Gamma)$ and define $\delta' : Q' \times \Sigma_\Gamma \to \mathcal{P}(Q')$ by $\delta'(a_I, t) = \bigcup_{A \in \Gamma} \mathsf{Tr}(t)(A) \subseteq \mathrm{Cl}(\Gamma)$ and $\delta'(B, t) = \mathsf{Tr}(t)(B)$ for $B \in \mathrm{Cl}(\Lambda)$ and $t \in \Sigma_\Gamma$. If we put $\Omega''(a_I) = 0$ and $\Omega''(B) = \Omega(B) + 1$ where $\Omega$ is a parity function for $\Gamma$, the automaton $\mathbb{A}' = (Q', a_I, \delta', \Omega'')$ accepts a word $w$ if $w$ *does* contain a bad trace starting in some $B \in \Gamma$. We now transform $\mathbb{A}'$ into an equivalent deterministic parity automaton $\mathbb{A}'_d$ by means of the Safra construction to obtain an automaton of size $2^{O(nk \log(nk))}$ whose parity function has a range of order $\mathcal{O}(nk)$ where $n = |Q'| + 1$ and $k$ is the cardinality of the range of $\Omega$ (cf. [19, 20]). The automaton $\mathbb{A}_\Gamma$ is then obtained by complementing $\mathbb{A}_d$ which can be done by changing the parity function, and neither increases the size nor the index of the automaton. This implies the claim as the cardinality $k$ of the range of $\Omega$ is bounded by the size $n$ of the state set $n$ of the initial automaton. $\square$

We thus arrive at the following notion of tableau game, where $\Gamma$-parity automata are used to detect bad traces.

**Definition 5.3.** Let $\Gamma \in \mathsf{S}(\Lambda)$ be clean and guarded, and let $\mathbb{A} = (Q, a_\Gamma, \delta, \Omega)$ be a $\Gamma$-parity automaton. We denote the set of tableau rules $\Gamma_0/\Gamma_1, \dots, \Gamma_n \in \mathsf{TR}$ for which $\Gamma_0 \in \mathsf{S}(\Gamma)$ by $\mathsf{TR}_\Gamma$ and write $\mathcal{B}(\Gamma)$ for the set of rule blueprints $\flat$ such that

* $\flat \in \mathrm{Cl}(\Gamma)$ if $\flat \in \mathcal{F}(\Lambda)$ and $A \in \mathrm{Cl}(\Gamma)$ if $\flat = (A, \overline{A})$
* $\Gamma_0 \sigma \in \mathsf{S}(\Gamma)$ if $\flat = (r, \sigma)$ and $r = \Gamma_0/\Gamma_1 \dots \Gamma_n$.

The $\Gamma$-*tableau game* is the parity game $\mathcal{G}_\Gamma = (B_\exists, B_\forall, E, \Omega')$ where $B_\forall = \mathsf{S}(\Gamma) \times Q$, $B_\exists = \mathsf{S}(\Gamma) \times \mathcal{B}(\Gamma) \times Q$ and the relation $E \subseteq B_\forall \times B_\exists \cup B_\exists \times B_\forall$ that defines the allowed moves is given by $(b_1, b_2) \in E$ if either

* $b_1 = (\Delta, a) \in B_\forall$, $b_2 = (\Delta, \flat, a)$ and $(\Delta, \flat)$ is a rule representation
* $b_1 = (\Delta, \flat, a)$, $b_2 = (\Delta', a')$ and there exists $i \in \mathbb{N}$ such that $\Delta'$ is the $i$-th conclusion of the rule represented by $(\Delta, \flat)$ and $a' = \delta(a, (\Delta, \flat, i))$.

The parity function $\Omega' : (B_\exists \cup B_\forall) \to \omega$ of $\mathcal{G}_\Gamma$ is given by $\Omega'(\Delta, a) = \Omega(a)$ if $(\Delta, a) \in B_\forall$ and $\Omega'(\Delta, \flat, a) = 0$.

If not explicitly stated otherwise, we will only consider $\mathcal{G}_\Gamma$-plays that start at $(\Gamma, a_\Gamma)$ where $a_\Gamma$ is the initial state of the automaton $\mathbb{A}$. In particular, we say that a player has a winning strategy in $\mathcal{G}_\Gamma$ if (s)he has a winning strategy in $\mathcal{G}_\Gamma$ at position $(\Gamma, a_\Gamma)$.

The easier part of the correspondence between satisfiability and winning strategies in $\mathcal{G}_\Gamma$ is proved by constructing a closed tableau based on a winning strategy for $\forall$. To show that this tableau is indeed closed, we need to show that every infinite path carries at least one bad trace, which follows from the fact that $\forall$ wins in the tableau game. To make this formal, we consider a notion of path and trace also relative to plays in the tableau game.

**Definition 5.4.** For a $\mathcal{G}_\Gamma$-play

$$\pi = (\Gamma^0, a_0)(\Gamma^0, \flat_0, a_0)(\Gamma^1, a_1)(\Gamma^1, \flat_1, a_1) \dots (\Gamma^l, a_l)(\Gamma^l, \flat_l, a_l) \dots$$

a sequence $\pi' = \Gamma^0 c_0 \Gamma^1 c_1 \dots \Gamma^l c_l \dots$ of sequents and natural numbers is an *underlying path* of $\pi$ if $t_i = (\Gamma^i, \flat_i, c_i)$ is a trace tile and $\delta(a_i, t_i) = a_{i+1}$ for all $i \in \mathbb{N}$. A sequence of formulas $\alpha = A_0 A_1 A_2 \dots \in \mathcal{F}(\Lambda)^\infty$ is a *trace* through $\pi$ if there exists an underlying path $\pi' = \Gamma^0 c_0 \Gamma^1 c_1 \Gamma^2 \dots$ of $\pi$ such that $(A_i, A_{i+1}) \in \mathsf{Tr}(\Gamma^i, \flat_i, c_i)$ for all $i \in \mathbb{N}$.

An underlying path of a $\mathcal{G}_\Gamma$-play is very similar to the notion of a tableau path. This is due to the correspondence between tableaux and strategies of $\forall$ in the tableau game. This correspondence is crucial in the proof of the following theorem.

**Theorem 5.5.** *Let $\Gamma \in \mathsf{S}(\Lambda)$ be clean and guarded. If $\forall$ has a winning strategy in $\mathcal{G}_\Gamma$, then $\Gamma$ has a closed* $\mathsf{TR}$*-tableau.*

*Proof.* Suppose that $\forall$ has a winning strategy $f$ in $\mathcal{G}_\Gamma$ at position $(\Gamma, a_\Gamma)$. As $\mathcal{G}_\Gamma$ is a parity game we can assume that $\forall$'s strategy is history-free, i.e. it can be encoded as a partial function $f : \mathsf{S}(\Gamma) \times Q \rightharpoonup \mathsf{S}(\Gamma) \times \mathcal{B}(\Gamma) \times Q$. In order to prove the claim we are going to define a closed tableau $\mathsf{T} = (N, K, R, \ell, \alpha)$ for $\Gamma$. We define $N$ to be the set of positions in $\mathsf{S}(\Gamma) \times Q$ for which $f$ is a winning strategy (in particular, this entails that $f$ is defined at all positions in $N$). Obviously we have $(\Gamma, a_\Gamma) \in N$ and we put $R = (\Gamma, a_\Gamma)$. The labelling function on $N$ is the first projection map, i.e. $\ell(\Delta, a) = \Delta$ for all $(\Delta, a) \in N \subseteq \mathsf{S}(\Gamma) \times Q$.

For all $(\Delta, a) \in N$ the set of $K$-successors is defined using $\forall$'s strategy by putting $K(\Delta, a) = \{(\Delta', a') \mid (\Delta', a') \in E(f(\Delta, a))\}$ where $E(f(\Delta, a))$ is the set of possible moves of $\exists$ at $f(\Delta, a)$. Finally we define the annotation $\alpha$ of $\mathsf{T}$ by putting $\alpha(\Delta, a) = \pi_2(f(\Delta, a))$ where $\pi_2 : \mathsf{S}(\Gamma) \times \mathcal{B}(\Gamma) \times Q \to \mathcal{B}(\Gamma)$ denotes the second projection map.

It is an easy consequence of the definition of the tableau game that $\mathsf{T}$ is a well-defined tableau. We now show that $\mathsf{T}$ is a closed tableau. To this aim consider first a finite complete path $\pi = (\Gamma_0, a_0) c_0 (\Gamma_1, a_1) c_1 \cdots c_{n-1}(\Gamma_n, a_n)$ through $\mathsf{T}$ with $(\Gamma_0, a_0) = (\Gamma, a_\Gamma)$. This gives rise to a $\mathcal{G}_\Gamma$-play of the form

$$(\Gamma_0, a_0)(\Gamma_0, \flat_0, a_0)(\Gamma_1, a_1)(\Gamma_1, \flat_1, a_1)(\Gamma_2, a_2) \dots (\Gamma_n, a_n)$$

that is played according to $\forall$'s winning strategy $f$. In order to see this, note that for all $0 \le i < n$ we have $(\Gamma_{i+1}, a_{i+1}) \in E(f(\Gamma_i, a_i))$, i.e. $(\Gamma_{i+1}, a_{i+1})$ is a legal answer to $\forall$'s move at $(\Gamma_i, a_i)$ if $\forall$ is playing according to his strategy $f$. Since $\pi$ was assumed to be complete, and since $\forall$ has a winning strategy at the last position $(\Gamma_n, a_n)$ of the corresponding $\mathcal{G}_\Gamma$-play, it follows that $\exists$ cannot move in the position obtained by $\forall$ playing according to his strategy at $(\Gamma_n, a_n)$. This can only be the case if $\forall$ moves to $(\Gamma_n, (A, \overline{A}), a_n)$ at $(\Gamma_n, a_n)$ for some $A \in \mathcal{F}(\Lambda)$, which in turn is only possible if $\Gamma_n$ is a tableau axiom.

Consider now an infinite path $\pi = (\Gamma_0, a_0)c_0(\Gamma_1, a_1)c_1(\Gamma_2, a_2) \ldots$ through $\mathsf{T}$ with $(\Gamma_0, a_0) = (\Gamma, a_\Gamma)$. As in the previous case, this induces an infinite $\mathcal{G}_\Gamma$-play $P$ of the form

$$P = (\Gamma_0, a_0)(\Gamma_0, \flat_0, a_0)(\Gamma_1, a_1)(\Gamma_1, \flat_1, a_1)(\Gamma_2, a_2) \ldots (\Gamma_n, a_n) \ldots$$

that is played according to $\forall$'s winning strategy $f$. By the definition of the game board of $\mathcal{G}_\Gamma$, this means that the infinite sequence $\rho = a_0 a_1 a_2 \ldots \in Q^\omega$ can be seen as a run of $\mathbb{A}_\Gamma$ on

$$w = (\Gamma_0, \flat_0, c_0)(\Gamma_1, \flat_1, c_1)(\Gamma_2, \flat_2, c_2) \ldots \in \Sigma_\Gamma^\omega.$$

By assumption $f$ was winning for $\forall$ and therefore $P$ does not satisfy the parity condition $\Omega'$ of $\mathcal{G}_\Gamma$. This implies that $\rho = a_\Gamma a_1 a_2 \ldots \in Q^\omega$ does not fulfil the parity condition $\Omega$ of the automaton $\mathbb{A}_\Gamma$. In other words, as $\rho$ is the run of $\mathbb{A}_\Gamma$ on $w$, there must be a sequence $\beta = B_0 B_1 B_2 \ldots \in \mathrm{Cl}(\Gamma)^\omega$ such that $(B_i, B_{i+1}) \in \mathsf{Tr}(\Gamma_i, \flat_i, c_i)$ that is bad w.r.t. $\Omega$. In other words, $\beta$ is also a trace through the path $\pi$, which implies that there exists a trace through $\pi$ that is bad w.r.t. $\Omega$ as required. This finishes the proof that $\mathsf{T}$ is closed. $\qquad\square$

The converse of the above theorem is established later as Theorem 5.18. The challenge there is to construct a model for $\Gamma$ based on a winning strategy for $\exists$ in the $\Gamma$-tableau game. As we only allow substitution instances of modal (one-step) rules that do not duplicate literals (we require that substitutions do not decrease the cardinality of premises in one-step rules in Definition 4.9), we need to require that the set of tableau rules to be closed under contraction.

**Definition 5.6.** A set $\mathsf{R}$ of monotone one-step rules is *closed under contraction*, if for all rules $\Gamma_0/\Gamma_1, \ldots, \Gamma_n \in \mathsf{R}$ and all $\sigma : \mathsf{V} \to \mathsf{V}$, there exists a rule $\Delta_0/\Delta_1, \ldots, \Delta_k \in \mathsf{R}$ and a renaming $\tau : V \to V$ such that $A\tau = B\tau$ for $A, B \in \Delta_0$ implies that $A = B$, $\Delta_0 \tau \subseteq \Gamma_0 \sigma$ and, for each $1 \le i \le n$, there exists $1 \le j \le k$ such that $\Gamma_i \sigma \subseteq \Delta_j \tau$.

In other words, instances of one-step rules which duplicate literals in the premise may be replaced by instances for which this is not the case.

**Remark 5.7.** Every monotone $\Lambda$-structure admits a one-step tableau sound and one-step tableau complete set of monotone tableau rules that is closed under contraction. This follows from the fact that the set of one-step rules from the proof of Proposition 4.7 is closed under contraction: Consider a rule $\Gamma_0/\Gamma_1, \ldots, \Gamma_n \in \mathsf{R}$ and any renaming $\sigma$. Then, by the definition of the set of one-step rules $\mathsf{R}$ in Prop. 4.7 we can easily show that $\Delta_0/\Delta_1, \ldots, \Delta_n \in \mathsf{R}$ with $\Delta_i = \Gamma_i \sigma$ for $i = 0, \ldots, n$. Closure under contraction follows from the fact that $\Delta_0/\Delta_1, \ldots, \Delta_n$ together with $\tau = \mathrm{id}_\mathsf{V}$ satisfy the conditions of Definition 5.6.

Under the condition of closure under contraction (cf. Remark 4.10), we prove:

**Theorem 5.8.** *Suppose that $\Gamma \in \mathsf{S}(\Lambda)$ is clean and guarded and $\mathsf{R}$ is one-step tableau complete and contraction closed. If $\exists$ has a winning strategy in $\mathcal{G}_\Gamma$, then $\Gamma$ is satisfiable in a model of size $\mathcal{O}(2^{p(n)})$ where $n$ is the cardinality of $\mathrm{Cl}(\Gamma)$ and $p$ is a polynomial.*

The proof of Theorem 5.8 constructs a model for $\Gamma$ out of the game board of $\mathcal{G}_\Gamma$ using a winning strategy $f$ for $\exists$ in $\mathcal{G}_\Gamma$. We use one-step tableau completeness to impose a $T$-coalgebra structure on those $\forall$-positions in $\mathcal{G}_\Gamma$ that are reachable through $f$-conform $\mathcal{G}_\Gamma$-plays, with the resulting coalgebra satisfying the truth lemma. We then equip this $T$-coalgebra with a valuation that makes $\Gamma$ satisfiable in the resulting model. While our construction shares some similarities with the shallow model construction of [24], it is by no means a simple adaptation of *op. cit.*, as we are dealing with fixpoint formulas and thus cannot employ induction over the modal rank of formulas to construct satisfying models. Our proof of satisfiability is also substantially different from the corresponding proof for

the modal $\mu$-calculus (cf. [17]) – we show satisfiability by directly deriving a winning strategy for $\exists$ in the model-checking game from a winning strategy of $\exists$ in the tableau game.

We now turn to the details of the proof of Theorem 5.8. Throughout the proof, we assume that $\Gamma \in \mathsf{S}(\Lambda)$ is a clean, guarded sequent and $f : \mathsf{S}(\Gamma) \times \mathcal{B}(\Gamma) \times Q \rightharpoonup \mathsf{S}(\Gamma) \times Q$ is a history-free winning strategy for $\exists$ in $\mathcal{G}_\Gamma$. The construction of a supporting Kripke frame for a model of $\Gamma$ is based on $\forall$-positions of $\mathcal{G}_\Gamma$ where only modal rules can be applied. This is formalised through the notion of *atomic sequent*.

**Definition 5.9.** A $\Lambda$-formula is *atomic* if it is either a propositional variable $p \in \mathsf{V}$, a negated propositional variable $\overline{p} \in \overline{\mathsf{V}}$, or a formula of the form $\heartsuit(A_1, \ldots, A_n)$ or $\overline{\heartsuit}(A_1, \ldots, A_n)$. A sequent $\Delta \in \mathsf{S}(\Lambda)$ is *atomic* if all its elements are atomic. We write $\mathsf{At}(\Gamma)$ for the set of atomic sequents in $\mathsf{S}(\Gamma)$, and call a $\mathcal{G}_\Gamma$-position $(\Delta, a) \in B_\forall$ *atomic* if $\Delta$ is atomic.

The state set of the satisfying model that we are about to construct are the atomic $\mathcal{G}_\Gamma$-positions $(\Delta, a)$ that are reachable from $(\Gamma, a_\Gamma)$ through $\mathcal{G}_\Gamma$-play that is played according to $f$. As the propositional rules are invertible, we may assume that $\forall$ applies them in any fixed, given order. This simplifies the model construction as it implies – together with $\exists$'s strategy – that every sequent is unfolded to an atomic sequent in a unique way. Fixing the order in which $\forall$ applies propositional rules can be seen as a strategy, that we call propositional:

**Definition 5.10.** A *propositional strategy* for $\forall$ in the tableau game $\mathcal{G}_\Gamma$ is a function

$$g : \mathsf{S}(\Gamma) \setminus \mathsf{At}(\Gamma) \rightarrow \mathcal{B}(\Gamma)$$

such that $(\Delta, g(\Delta))$ is a rule representation for all $\Delta \in \mathsf{S}(\Gamma) \setminus \mathsf{At}(\Gamma)$. A $\mathcal{G}_\Gamma$-play is *played according to $g$* if $\forall$ moves at any position of the form $(\Delta, a) \in (\mathsf{S}(\Gamma) \setminus \mathsf{At}(\Gamma)) \times Q$ that occurs in the play to the position $(\Delta, g(\Delta), a)$.

For the remainder of this section we fix a propositional strategy $g$ for $\forall$. As announced informally in the beginning, this dictates that plays proceed to atomic positions in a unique way, and in fact induces a function from arbitrary positions to atomic ones in the tableau game.

**Lemma and Definition 5.11.** Let $f$ be a strategy for $\exists$ in $\mathcal{G}_\Gamma$. For any position $(\Delta, a) \in \mathsf{S}(\Gamma) \times Q$ there exists precisely one position $(\Delta', a') \in \mathsf{At}(\Gamma) \times Q$ and one partial $\mathcal{G}_\Gamma$-play

$$(\Delta, a), \cdots, (\Delta', a')$$

that is played according to $f$ and $g$ and which does not contain an instance of a modal rule. We let $\sigma_f : \mathsf{S}(\Gamma) \times Q \rightarrow \mathsf{At}(\Gamma) \times Q$ be the function given by $\sigma_f(\Delta, a) = (\Delta', a')$.

For the construction of a satisfying model for $\Gamma$ we are going to define a relation on the set of atomic positions of $\mathcal{G}_\Gamma$ where two atomic positions are related if the second position is selected by $\exists$'s strategy in response to $\forall$ playing a modal rule. In the case of Kripke frames, this relation would already define the satisfying model, but in the general case, we need to impose a coalgebra structure on top of this relation in a coherent way. To achieve this, we single out specific states (the $A$-successors) that we take as under-approximation of the semantics of a formula $A$. Informally speaking, an $A$-successor of an atomic state arises by $\forall$ playing a modal rule, and $\exists$ selecting a conclusion containing $A$ that is then reduced to another atomic position. Formally, we introduce the notions of $A$-children (conclusions selected by $\exists$ that contain $A$) and $A$-successors (reductions of $A$-children to atomic form), both relative to a strategy for $\exists$.

**Definition 5.12.** Suppose that $f$ is a history-free strategy of $\exists$ in $\mathcal{G}_\Gamma$, and let $(\Delta, a) \in \mathsf{At}(\Gamma)$. A position $(\Delta', a') \in \mathsf{S}(\Gamma) \times Q$ is an $A$-*child* of $(\Delta, a)$ along $f$ if $A \in \Delta'$ and $(\Delta', a') = f(\Delta, \flat, a)$

where $((\Delta, a), (\Delta, \flat, a))$ is a legal move of $\forall$ in $\mathcal{G}_\Gamma$. We put

$$\mathsf{Chld}_f(A, \Delta, a) = \{(\Delta', a') \in \mathsf{S}(\Gamma) \times Q \mid (\Delta', a') \text{ } A\text{-child of } (\Delta, a) \text{ along } f\}$$

and write $\mathsf{Chld}_f(\Delta, a)$ for the collection of all $A$-children of $(\Delta, a)$ along $f$. An atomic position $(\Delta'', a'')$ is an $A$-*successor* of $(\Delta, a)$ along $f$ if $(\Delta'', a'') = \sigma_f(\Delta', a')$ for some $A$-child $(\Delta', a')$ of $(\Delta, a)$ along $f$. This is denoted by

$$\mathsf{Suc}_f(A, \Delta, a) = \{(\Delta'', a'') \in \mathsf{At}(\Gamma) \times Q \mid (\Delta'', a'') \text{ } A\text{-successor of } (\Delta, a) \text{ along } f\}$$

and we write $\mathsf{Suc}_f(\Delta, a) = \bigcup_{A \in \mathrm{Cl}(\Gamma)} \mathsf{Suc}_f(A, \Delta, a)$ for the collection of all $A$-successors of $(\Delta, a)$.

In other words, an atomic position $(\Delta'', a'')$ is a successor of $(\Delta, a)$ if it is reachable from $(\Delta, a)$ by a play that is played according to $\exists$'s strategy $f$ and the (fixed) propositional strategy $g$ that involves precisely one modal rule. The position $(\Delta'', a'')$ is an $A$-successor of $(\Delta, a)$ if the conclusion of this modal rule that is picked by $f$ contains the formula $A$. This allows us to introduce *coherent* coalgebra structures, i.e. those structures on atomic positions that satisfy the truth lemma.

**Definition 5.13.** Suppose that $f$ is a history-free strategy for $\exists$ in $\mathcal{G}_\Gamma$ and let

$$Y = \{(\Delta, a) \in \mathsf{At}(\Gamma) \times Q \mid \sigma_f(\Gamma, a_I) \to^* (\Delta, a)\}$$

where for $(\Delta, a), (\Delta', a') \in \mathsf{At}(\Gamma) \times Q$, $(\Delta, a) \to (\Delta', a')$ if $(\Delta', a') \in \mathsf{Suc}_f(\Delta, a)$. A coalgebra structure $\gamma : Y \to TY$ on $Y$ is called *coherent* if

$$\gamma(\Delta, a) \in [\![\heartsuit]\!]_Y(\mathsf{Suc}_f(A_1, \Delta, a), \dots, \mathsf{Suc}_f(A_n, \Delta, a))$$

whenever $\heartsuit(A_1, \dots, A_n) \in \Delta$. A valuation $h : \mathsf{V} \to \mathcal{P}(Y)$ is *coherent* if $(\Delta, a) \in h(p)$ whenever $p \in \Delta$.

In other words, the carrier of a coherent coalgebra is the set of atomic positions that are reachable from the initial position via $\exists$'s strategy $f$, and the coalgebra structure is so that we can establish the truth lemma, together with monotonicity of the modal operators: the $A$-successors of an atomic position contain an element of the disjunctive normal form of $A$ and hence serve as an under-approximation of the truth-set of $A$. We note that a position cannot be both an $A$-successor and an $\overline{A}$-successor of the same position.

**Lemma 5.14.** *Let $f$ be a history-free winning strategy for $\exists$ in $\mathcal{G}_\Gamma$ and let $(\Delta_1, a_1)$ and $(\Delta_2, a_2)$ be atomic $\mathcal{G}_\Gamma$-positions such that $f$ is a winning strategy for $\exists$ at $(\Delta_1, a_1)$. Then for all formulas $A$ we have*

$$(\Delta_2, a_2) \in \mathsf{Suc}_f(A, \Delta_1, a_1) \quad \text{implies that} \quad (\Delta_2, a_2) \notin \mathsf{Suc}_f(\overline{A}, \Delta_1, a_1).$$

*Proof.* Suppose for a contradiction that $(\Delta_2, a_2) \in \mathsf{Suc}_f(A, \Delta_1, a_1)$ and $(\Delta_2, a_2) \in \mathsf{Suc}_f(\overline{A}, \Delta_1, a_1)$ for some formula $A$. Then, by the definition of $\mathsf{Suc}_f$, there must exist $(\Delta', a')$ and $(\Delta'', a'')$ in $\mathsf{S}(\Gamma) \times Q$ such that $A \in \Delta'$, $\overline{A} \in \Delta''$ and $\sigma_f(\Delta', a') = \sigma_f(\Delta'', a'') = (\Delta_2, a_2)$. A straightforward induction argument shows that in this case there must exist a formula $B$ such that $B, \overline{B} \in \Delta_2$. Therefore $(\Delta_2, a_2)$ is a winning position for $\forall$. But this contradicts the fact that there exists a $\mathcal{G}_\Gamma$-play from $(\Delta_1, a_1)$ to $(\Delta_2, a_2)$ played according to $f$, and our assumption that $f$ is winning at $(\Delta_1, a_1)$. $\square$

We now show that if $\exists$ has a winning strategy $f$ in the tableau game for $\Gamma$, then a coherent model for $\Gamma$ exists. This is where contraction closure is needed as the application of modal rules may not identify elements in the premise of a rule.

**Proposition 5.15.** *Every history-free winning strategy $f : \mathsf{S}(\Gamma) \times \mathcal{B}(\Gamma) \times Q \rightharpoonup \mathsf{S}(\Gamma) \times Q$ for $\exists$ in $\mathcal{G}_\Gamma$ induces a coherent model $(Y, \gamma, h)$.*

*Proof.* We follow Definition 5.13 and put $Y = \{(\Delta, a) \in \mathsf{At}(\Gamma) \mid \sigma_f(\Gamma, a_I) \rightarrow^* (\Delta, a)\}$ where $\rightarrow$ is as in the definition, and we define a coherent valuation $h : \mathsf{V} \to Y$ by $h(p) = \{(\Delta, a) \in Y \mid p \in \Delta\}$. It remains to be seen that we can define $\gamma : Y \to TY$ coherently. It is a consequence of Lemma 5.14 and of the fact that $f$ is a winning strategy for $\exists$ in $\mathcal{G}_\Gamma$ that for all $(\Delta_1, a_1), (\Delta_2, a_2) \in Y$ we have

$$(\Delta_2, a_2) \in \mathsf{Suc}_f(A, \Delta_1, a_1) \quad \text{implies} \quad (\Delta_2, a_2) \notin \mathsf{Suc}_f(\overline{A}, \Delta_1, a_1). \tag{5.1}$$

Now suppose for a contradiction that there is no $\gamma : Y \to TY$ such that $(Y, \gamma)$ is a coherent coalgebra structure for $\Gamma$. Then there exists some $(\Delta, a) \in Y$ such that we cannot find a $t \in TY$ that satisfies the condition in Definition 5.13. Consider the set of formulas

$$\begin{aligned}\Theta = \{&\heartsuit(p_{A_1}, \ldots, p_{A_n}) \mid \heartsuit(A_1, \ldots, A_n) \in \Delta\} \\ \cup \ &\{\overline{\heartsuit}(p_{A_1}, \ldots, p_{A_n}) \mid \overline{\heartsuit}(A_1, \ldots, A_n) \in \Delta\}\end{aligned} \tag{5.2}$$

where for any formulas of the form $\heartsuit(A_1, \ldots, A_n)$ or $\overline{\heartsuit}(A_1, \ldots, A_n)$ in $\Delta$ we associate a unique propositional variable $p_{A_i}$ to the formula $A_i$, for $i \in \{1, \ldots, n\}$. Let $V_\Theta$ be the set of propositional variables occurring in $\Theta$. We define a valuation $\tau : V_\Theta \to \mathcal{P}(\mathsf{Suc}_f(\Delta, a))$ by putting $\tau(p_A) = \mathsf{Suc}_f(A, \Delta, a)$.

Using our assumption on $(\Delta, a)$ it is not difficult to see that $[\![\Theta]\!]_{T\mathsf{Suc}_f(\Delta,a),\tau} = \emptyset$. Therefore one-step tableau completeness implies that there exists a rule $\Gamma_0/\Gamma_1 \cdots \Gamma_n$ and a substitution $\sigma : V \to V$ such that $\Gamma_0\sigma \subseteq \Theta$ and $[\![\Gamma_i\sigma]\!]_{\mathsf{Suc}_f(\Delta,a),\tau} = \emptyset$ for all $i \in \{1, \ldots, n\}$. Because of contraction closure of R we can assume w.l.o.g. that $\sharp(\Gamma_0\sigma) = \sharp(\Gamma_0)$.

On the other hand, for $\eta : V_\Theta \to \mathcal{F}(\Lambda)$ with $\eta(p_A) = A$, we clearly have $\Gamma_0\sigma\eta \subseteq \Delta$ with $\sharp(\Gamma_0\sigma\eta) = \sharp(\Gamma_0)$, and thus $\forall$ can move in the tableau game from position $(\Delta, a)$ to the position $(\Delta, (\Gamma_0/\Gamma_1 \cdots \Gamma_n, \eta \circ \sigma), a)$. Now $\exists$ moves to some $(\Gamma_j\sigma\eta, a'')$ with $j \in \{1, \ldots, n\}$ according to her winning strategy $f$. Therefore we have $(\Gamma_j\sigma\eta, a'') \in \mathsf{Chld}_f(\Delta, a)$. Furthermore, the play can be continued according to $\exists$'s strategy $f$ until the atomic position $(\Delta', a') = \sigma_f(\Gamma_j\sigma\eta, a'')$ is reached. By definition we have

$$(\Delta', a') \in \mathsf{Suc}_f(B\eta, \Delta, a) \text{ for all } B \in \Gamma_j\sigma. \tag{5.3}$$

It now follows that $(\Delta', a') \in [\![B]\!]_{\mathsf{Suc}_f(\Delta,a),\tau}$ for all $B \in \Gamma_j\sigma$. To see this, consider an arbitrary formula $B \in \Gamma_j\sigma$. By the definition of $\Theta$ and the fact that $\Gamma_0\sigma \subseteq \Theta$ we have that $\Gamma_j\sigma$ consists of atoms only. Therefore $B = p_A$ for some formula $A$. By (5.3), we know that $(\Delta', a') \in \mathsf{Suc}_f(p_A\eta, \Delta, a) = \mathsf{Suc}_f(A, \Delta, a)$, and therefore $(\Delta', a') \in [\![B]\!]_{\mathsf{Suc}_f(\Delta,a),\tau}$. As $B$ was an arbitrary element of $\Gamma_j\sigma$ we obtain $(\Delta', a') \in [\![B]\!]_{\mathsf{Suc}_f(\Delta,a),\tau}$ for all $B \in \Gamma_j\sigma$, which contradicts the fact that $[\![\Gamma_j\sigma]\!]_{\mathsf{Suc}_f(\Delta,a),\tau} = \emptyset$. This concludes the proof. $\square$

We can now take a history-free winning strategy $f$ for $\exists$ in the tableau game and show that the induced coherent model $Y$ satisfies the initial sequent. This is achieved by converting the strategy $f$ (in the tableau game) to a strategy $\tilde{f}$ in the model checking game over $Y$. Satisfiability then follows as soon as we establish that $\mathcal{MG}_\Gamma$-plays that are played according to $\tilde{f}$ correspond to traces through $\mathcal{G}_\Gamma$-plays that are played according to $f$.

**Lemma 5.16.** *Let $f$ be a history-free winning strategy for $\exists$ in $\mathcal{G}_\Gamma$, let $\mathbb{Y} = (Y, \gamma, h)$ be the coherent model induced by $f$, and consider a position $(A_0, (\Delta_0, a_0))$ in $\mathcal{MG}_\Gamma(\mathbb{Y})$ with $(\Delta_0, a_0) = \sigma_f(\Gamma, a_I)$ and $A_0 \in \Delta_0$. Then $\exists$ has a strategy $\tilde{f}$ in $\mathcal{MG}_\Gamma(\mathbb{Y})$ at $(A_0, (\Delta_0, a_0))$ such that for any (possibly infinite) sequence $(A_0, (\Delta_0, a_0))(A_1, (\Delta_1, a_1)) \ldots (A_n, (\Delta_n, a_n)) \ldots$ that can be extended to an $\tilde{f}$-conform $\mathcal{MG}_\Gamma(\mathbb{Y})$-play by inserting positions of the form $(\heartsuit(B_1, \ldots, B_n), (U_1, \ldots, U_n))$ we have*

(1) *there exists a (possibly infinite) $\mathcal{G}_\Gamma$-play $\pi$ and a trace $\tau = B_0, B_1, \ldots, B_r, \ldots$ through $\pi$ (cf. Def. 5.4), such that*

  (a) *$\pi$ contains a sub-sequence of $\forall$-positions of the form*

$$(\Delta_0', a_0'), (\Delta_1', a_1'), \ldots, (\Delta_n', a_n'), \ldots$$

   *with $\sigma_f(\Delta_i', a_i') = (\Delta_i, a_i)$ and $\Delta_i' \ni A_i$ for each $i \geq 0$*

  (b) *$\tau$ is contractable to $A_0, A_1, \ldots, A_n, \ldots$, that is, there exists an increasing sequence $0 = s_0 < s_1 < \ldots$ of indices such that $A_0 A_1 \ldots = B_{s_0} B_{s_1} \ldots$, where $B_i = B_{s_j}$ whenever $s_j \leq i < s_{j+1}$, for $j = 0, 1, \ldots$.*

(2) *for all $\mathcal{MG}_\Gamma(\mathbb{Y})$-positions of the form $(A, (\Delta, a))$ occurring in $\pi$, with $A$ atomic, we have $A \in \Delta$.*

*Proof.* We define the strategy $\tilde{f}$ for $\exists$ in $\mathcal{MG}_\Gamma(\mathbb{Y})$ starting at position $(A_0, (\Delta_0, a_0))$ by showing how to extend each partial, $\tilde{f}$-conform $\mathcal{MG}_\Gamma(\mathbb{Y})$-play starting in $(A_0, (\Delta_0, a_0))$ and ending in an $\exists$-position $b = (B, (\Delta, a))$ with a position $b'$, such that $(b, b')$ is a valid move for $\exists$ in $\mathcal{MG}_\Gamma(\mathbb{Y})$. We will show later that each such partial play determines a partial $\mathcal{G}_\Gamma$-play starting in $(\Delta_0, a_0)$ and ending in some $(\Delta', a') \in \mathsf{S}(\Gamma) \times Q$ with $\sigma_f(\Delta', a') = (\Delta, a)$ and $\Delta' \ni B$. At this point, we assume the above, and base our definition of $\exists$'s strategy solely on $(B, (\Delta, a))$ and $(\Delta', a')$. We define $\exists$'s move in $(B, (\Delta, a))$ by case analysis on $B$:

**Case $B = B_1 \vee B_2$:** Then $\sigma_f(\Delta', a') = (\Delta, a)$ together with $\Delta' \ni B$ ensure the existence of a $\mathcal{G}_\Gamma$-play of the form

$$(\Gamma_0, d_0)(\Gamma_0, \flat_0, d_0) \ldots (\Gamma_{k-1}, \flat_{k-1}, d_{k-1})(\Gamma_k, d_k)$$

with $(\Gamma_0, d_0) = (\Delta', a')$, $\Gamma_j \notin \mathsf{At}(\Gamma)$ for $0 \leq j < k$ and $(\Gamma_k, d_k) = (\Delta, a)$, that is played according to $f$ and $g$, such that $\flat_j = B_1 \vee B_2$ for some $0 \leq j < k$. Let $\Gamma_0 c_0 \ldots c_{k-1} \Gamma_k$ be an underlying path of the above $\mathcal{G}_\Gamma$-play. Then, $c_j \in \{1, 2\}$, and we define $\exists$'s move at position $(B, (\Delta, a))$ of $\mathcal{MG}_\Gamma(\mathbb{Y})$ to be to the position $(B_{c_j}, (\Delta, a))$. Moreover, we note for future reference that the tableau node $(\Gamma_{j+1}, d_{j+1})$ satisfies $\sigma_f(\Gamma_{j+1}, d_{j+1}) = (\Delta, a)$ and $\Gamma_{j+1} \ni B_{c_j}$.

**Case $B = \heartsuit(B_1, \ldots, B_n)$:** We define $\exists$'s move at position $(B, (\Delta, a))$ of $\mathcal{MG}_\Gamma(\mathbb{Y})$ to be to the position $(B, (U_1, \ldots, U_n))$ with

$$U_j \;=\; \mathsf{Suc}_f(B_j, \Delta, a)$$

for $j = 1, \ldots, n$. To justify this move, we must show that $\gamma(\Delta, a) \in [\![\heartsuit]\!]_Y(U_1, \ldots, U_n)$. But this follows from Definition 5.13.

This defines a strategy for $\exists$ as there is no choice for $\exists$ at all other positions $(B, (\Delta, a))$ in $\mathcal{MG}_\Gamma(\mathbb{Y})$. Now consider a (possibly infinite) $\mathcal{MG}_\Gamma(\mathbb{Y})$-play of the form

$$(A_0, (\Delta_0, a_0)), (A_1, (\Delta_1, a_1)), \ldots, (A_n, (\Delta_n, a_n)), \ldots$$

played according to the previously defined strategy. We shall construct a $\mathcal{G}_\Gamma$-play $\pi$ and an underlying path $\pi'$ of $\pi$ with an associated trace $\tau$, with the required properties. In particular, the construction of $\pi$ will supply a sequence of $\mathcal{G}_\Gamma$-positions $(\Delta_0', a_0'), (\Delta_1', a_1'), \ldots$ to be used in defining $\exists$'s moves.

To begin with, note that by assumption on $(A_0, (\Delta_0, a_0))$ we have $\sigma_f(\Gamma, a_\Gamma) = (\Delta_0, a_0)$ and $A_0 \in \Gamma$. Hence, we let $(\Delta_0', a_0') = (\Gamma, a_\Gamma)$ be the first position of $\pi$, let $\Gamma$ be the first position of $\pi'$, and let $\tau_0 = A_0 \in \Gamma$.

Now assume that $\pi$, $\pi'$ and $\tau$ have been constructed up to a position $(\Delta_i', a_i')$, with $\sigma_f(\Delta_i', a_i') = (\Delta_i, a_i)$ and $\Delta_i' \ni A_i$. We extend the partial $\mathcal{G}_\Gamma$-play $\pi$ with a segment starting in $(\Delta_i', a_i')$ and ending in some $(\Delta_{i+1}', a_{i+1}')$, with $\sigma_f(\Delta_{i+1}', a_{i+1}') = (\Delta_{i+1}, a_{i+1})$ and $\Delta_{i+1}' \ni A_{i+1}$. Here $(\Delta_{i+1}, a_{i+1})$ represents the position obtained as a result of $\exists$ moving in $(\Delta_i, a_i)$, based on the additional information provided by $(\Delta_i', a_i')$, according to the strategy defined earlier. At the same time, we extend the underlying path $\pi'$ of $\pi$ with a segment $\Delta_i' \ldots \Delta_{i+1}'$, and the trace $\tau$ with a segment $A_i, \ldots, A_i, A_{i+1}$. These constructions are carried out by case analysis on $A_i$.

**Case $A_i = A_i^1 \vee A_i^2$:** Here, the $\mathcal{MG}_\Gamma(\mathbb{Y})$-move from $(A_i, (\Delta_i, a_i))$ to $(A_{i+1}, (\Delta_{i+1}, a_{i+1}))$ is an $\exists$-move played according to the strategy defined earlier. The definition of this move was based on a $\mathcal{G}_\Gamma$-play of the form

$$(\Gamma_0, d_0)(\Gamma_0, \flat_0, d_0) \ldots (\Gamma_{k-1}, \flat_{k-1}, d_{k-1})(\Gamma_k, d_k)$$

with $(\Gamma_0, d_0) = (\Delta_i', a_i')$, $\Gamma_l \notin \mathsf{At}(\Gamma)$ for $0 \le l < k$ and $(\Gamma_k, d_k) = (\Delta_i, a_i)$, played according to $f$ and $g$, with an underlying path $\Gamma_0 c_0 \ldots c_{k-1} \Gamma_k$, such that there exists $0 \le j < k$ with $(A_i, A_i) \in \mathsf{Tr}(\Gamma_l, \flat_l, c_l)$ for $0 \le l < j$ and $(A_i, A_i^{c_j}) \in \mathsf{Tr}(\Gamma_j, \flat_j, c_j)$. Moreover, this definition guarantees that we have $\sigma_f(\Gamma_{j+1}, d_{j+1}) = (\Delta_{i+1}, a_{i+1}) = (\Delta_i, a_i)$. We now put $(\Delta_{i+1}', a_{i+1}') = (\Gamma_{j+1}, d_{j+1})$, and extend the play $\pi$ to $(\Gamma_0, \flat_0, d_0) \ldots (\Gamma_j, \flat_j, d_j)(\Gamma_{j+1}, d_{j+1})$, the underlying path $\pi'$ with $c_0 \ldots c_j \Gamma_{j+1}$, and the trace $\tau$ with $A_i, \ldots, A_i, A_i^{c_j}$.

**Case $A_i = A_i^1 \wedge A_i^2$:** This time, the move from $(A_i, (\Delta_i, a_i))$ to $(A_{i+1}, (\Delta_{i+1}, a_{i+1}))$ is a $\forall$-move, with $A_{i+1} = A_i^l$ for some $l \in \{1, 2\}$ and $(\Delta_{i+1}, a_{i+1}) = (\Delta_i, a_i)$. Since $A_i \in \Delta_i'$ and $\sigma_f(\Delta_i', a_i') = (\Delta_i, a_i)$, it follows that there exist a $\mathcal{G}_\Gamma$-play of the form

$$(\Gamma_0, d_0)(\Gamma_0, \flat_0, d_0) \ldots (\Gamma_{k-1}, \flat_{k-1}, d_{k-1})(\Gamma_k, d_k)$$

with $(\Gamma_0, d_0) = (\Delta_i', a_i')$, $\Gamma_l \notin \mathsf{At}(\Gamma)$ for $0 \le l < k$ and $(\Gamma_k, d_k) = (\Delta_i, a_i)$, played according to $f$ and $g$, such that $\flat_j = A_i^1 \wedge A_i^2$ for some $0 \le j < k$, and an underlying path $\Gamma_0 c_0 \ldots c_{k-1} \Gamma_k$ of this $\mathcal{G}_\Gamma$-play that satisfies $(A_i, A_i) \in \mathsf{Tr}(\Gamma_h, \flat_h, c_h)$ for $0 \le h < j$ and $(A_i, A_i^l) \in \mathsf{Tr}(\Gamma_j, A_i^1 \wedge A_i^2, c_j)$. From the latter we obtain $l = c_j$. We then let $(\Delta_{i+1}', a_{i+1}')$ be given by $(\Gamma_{j+1}, d_{j+1})$, and note that $\sigma_f(\Delta_{i+1}', a_{i+1}') = (\Delta_{i+1}, a_{i+1}) = (\Delta_i, a_i)$ and $\Delta_{i+1}' = \Gamma_{j+1} \ni A_i^{c_j} = A_i^l = A_{i+1}$. It is therefore possible for us to extend the play $\pi$ with the sequence $(\Gamma_0, \flat_0, d_0) \ldots (\Gamma_j, \flat_j, d_j)(\Gamma_{j+1}, d_{j+1})$, the underlying path $\pi'$ with $c_0 \ldots c_j \Gamma_{j+1}$, and the trace $\tau$ with $A_i, \ldots, A_i, A_i^l$.

**Case $A_i = \heartsuit(B_1, \ldots, B_n)$:** The move from $(A_i, (\Delta_i, a_i))$ to $(A_{i+1}, (\Delta_{i+1}, a_{i+1}))$ thus incorporates an $\exists$-move played according to the strategy defined earlier, followed by a $\forall$-move. Again, from $A_i \in \Delta_i'$ and $\sigma_f(\Delta_i', a_i') = (\Delta_i, a_i)$ we obtain a $\mathcal{G}_\Gamma$-play of the form

$$(\Gamma_0, d_0)(\Gamma_0, \flat_0, d_0) \ldots (\Gamma_{k-1}, \flat_{k-1}, d_{k-1})(\Gamma_k, d_k)$$

with $(\Gamma_0, d_0) = (\Delta_i', a_i')$, $\Gamma_l \notin \mathsf{At}(\Gamma)$ for $0 \le l < k$ and $(\Gamma_k, d_k) = (\Delta_i, a_i)$, played according to $f$ and $g$, that has an underlying path $\Gamma_0 c_0 \ldots c_{k-1} \Gamma_k$ such that $(A_i, A_i) \in \mathsf{Tr}(\Gamma_j, \flat_j, c_j)$ for $0 \le j < k$. Also, by definition of $\exists$'s move in $(A_i, (\Delta_i, a_i))$ we obtain $A_{i+1} = B_j$ and $(\Delta_{i+1}, a_{i+1}) \in \mathsf{Suc}_f(B_j, \Delta_i, a_i)$ for some $j \in \{1, \ldots, n\}$. It follows that there exists a position $(\Delta'', a'')$ such that $(\Delta'', a'') \in \mathsf{Chld}_f(\Delta_i, a_i)$, $B_j \in \Delta''$ and $\sigma_f(\Delta'', a'') = (\Delta_{i+1}, a_{i+1})$. We then let $(\Delta_{i+1}', a_{i+1}')$ be given by $(\Delta'', a'')$. Moreover, from $(\Delta_{i+1}', a_{i+1}') \in \mathsf{Chld}_f(\Delta_i, a_i)$ it follows that $\forall$ can move in $\mathcal{G}_\Gamma$ from $(\Delta_i, a_i)$ to some $(\Delta_i, \flat, a_i)$ with $f(\Delta_i, \flat, a_i) = (\Delta_{i+1}', a_{i+1}')$. Since $\exists$'s move at position $(\Delta_i, \flat, a_i)$ was legal, this now yields $c \in \mathbb{N}$ such that $\Delta_{i+1}'$ is the $c$-th

conclusion of the rule represented by $(\Delta_i, \flat)$. This together with $B_j \in \Delta'_{i+1}$ yield $(A_i, B_j) \in \mathsf{Tr}(\Delta_i, \flat, c)$. It is now possible to extend the play $\pi$ with

$$(\Gamma_0, \flat_0, d_0) \ldots (\Gamma_{k-1}, \flat_{k-1}, d_{k-1})(\Gamma_k, d_k)(\Delta_i, \flat, a_i)(\Delta'_{i+1}, a'_{i+1}) \,,$$

the underlying path $\pi'$ with $c_0 \ldots c_{k-1} \Gamma_k c \Delta'_{i+1}$, and the trace $\tau$ with $A_i, \ldots, A_i, B_j$.

**Case** $A_i = \eta X.A$, $\eta \in \{\mu, \nu\}$**:** The move from $(A_i, (\Delta_i, a_i))$ to $(A_{i+1}, (\Delta_{i+1}, a_{i+1}))$ consists of unfolding the fixpoint variable $X$, that is, $A_{i+1} = A[X := \eta X.A]$ and $(\Delta_{i+1}, a_{i+1}) = (\Delta_i, a_i)$. Again, $\Delta'_i \ni \eta X.A$ together with $\sigma_f(\Delta'_i, a'_i) = (\Delta_i, a_i)$ yield a $\mathcal{G}_\Gamma$-play of the form

$$(\Gamma_0, d_0)(\Gamma_0, \flat_0, d_0) \ldots (\Gamma_{k-1}, \flat_{k-1}, d_{k-1})(\Gamma_k, d_k)$$

with $(\Gamma_0, d_0) = (\Delta'_i, a'_i)$, $\Gamma_l \notin \mathsf{At}(\Gamma)$ for $0 \leq l < k$ and $(\Gamma_k, d_k) = (\Delta_i, a_i)$, played according to $f$ and $g$, such that $\flat_j = \eta X.A$ for some $0 \leq j < k$, and an underlying path $\Gamma_0 c_0 \ldots c_{k-1} \Gamma_k$ of this $\mathcal{G}_\Gamma$-play that satisfies $(A_i, A_i) \in \mathsf{Tr}(\Gamma_h, \flat_h, c_h)$ for $0 \leq h < j$ and $(A_i, A_{i+1}) \in \mathsf{Tr}(\Gamma_j, \flat_j, c_j)$. We now let $(\Delta'_{i+1}, a'_{i+1})$ be given by $(\Gamma_{j+1}, d_{j+1})$, and note that $\sigma_f(\Delta'_{i+1}, a'_{i+1}) = (\Delta_{i+1}, a_{i+1}) = (\Delta_i, a_i)$ and $\Delta'_{i+1} \ni A[X := \eta X.A]$. It is therefore possible to extend the play $\pi$ with $(\Gamma_0, \flat_0, d_0) \ldots (\Gamma_j, \flat_j, d_j)(\Gamma_{j+1}, d_{j+1})$, the underlying path $\pi'$ with $c_0 \ldots c_j \Gamma_{j+1}$, and the trace $\tau$ with $A_i, \ldots, A_i, A[X := \eta X.\phi]$.

To show the second property of the $\mathcal{MG}_\Gamma(\mathbb{Y})$-play

$$(A_0, (\Delta_0, a_0)), (A_1, (\Delta_1, a_1)), \ldots, (A_n, (\Delta_n, a_n)), \ldots$$

we note that $\sigma_f(\Delta'_i, a'_i) = (\Delta_i, a_i)$ together with $A_i$ atomic and $A_i \in \Delta'_i$ yield $A_i \in \Delta_i$, for $i = 0, 1, \ldots$. $\qquad\square$

Finally, we prove satisfiability of $\Gamma$ in $\mathbb{Y}$ by showing that the strategy resulting from Lemma 5.16 is a winning strategy for $\exists$ in $\mathcal{MG}_\Gamma(\mathbb{Y})$.

**Theorem 5.17.** *Let* $f : \mathsf{S}(\Gamma) \times \mathcal{B}(\Gamma) \times Q \rightharpoonup \mathsf{S}(\Gamma) \times Q$ *be a history-free winning strategy for* $\exists$ *in* $\mathcal{G}_\Gamma$, *and let* $\mathbb{Y} = (Y, \gamma, h)$ *be the corresponding model of a coherent coalgebra structure* $(Y, \gamma)$ *for* $\Gamma$. *Then,* $\mathbb{Y}, (\Delta, a) \models A$ *for all states* $(\Delta, a) \in \sigma_f(\Gamma, a_\Gamma)$ *and all formulas* $A \in \Gamma$.

*Proof.* Let $(\Delta_0, a_0) \in Y$ be such that $\sigma_f(\Gamma, a_\Gamma) = (\Delta_0, a_0)$, and let $A_0 \in \Gamma$. Thus, $(A_0, (\Delta_0, a_0))$ is an initial position of $\mathcal{MG}_\Gamma(\mathbb{Y})$. Let $\tilde{f}$ be the strategy for $\exists$ at $(A_0, (\Delta_0, a_0))$ in $\mathcal{MG}_\Gamma(\mathbb{Y})$ provided by Lemma 5.16. We show that $\mathbb{Y}, (\Delta_0, a_0) \models A_0$ by showing that $\exists$ wins all $\mathcal{MG}_\Gamma(\mathbb{Y})$-plays that start at position $(A_0, (\Delta_0, a_0))$ and are played according to $\tilde{f}$.

Consider such a play, and assume first that it is finite. Let $(A, (\Delta, a))$ be its last position of type $\mathsf{Cl}(\Gamma) \times (\mathsf{S}(\Gamma) \times Q)$. Thus, the last position of the play is either $(A, (\Delta, a))$ itself, or a $\forall$-position of type $(\heartsuit(B_1, \ldots, B_n), (U_1, \ldots, U_n))$, with $U_i = \emptyset$ for $i = 1, \ldots, n$. In either case, $A$ is atomic (otherwise the play would not be complete). We distinguish the following cases:

(1) $A = p$ for some propositional variable $p$. By coherence of the valuation, we have $p \in \Delta$, and therefore by the definition of $\mathbb{Y}$ we have $(\Delta, a) \in h(p)$, which implies that $(p, (\Delta, a))$ is a winning position for $\exists$.

(2) $A = \overline{p}$. Similar to the previous case.

(3) $A = \heartsuit(B_1, \ldots, B_n)$. According to the definition of $\exists$'s strategy $\tilde{f}$, the last position of the play must be a $\forall$-position of type $(\heartsuit(B_1, \ldots, B_n), (U_1, \ldots, U_n))$ with $U_i = \emptyset$ for $i = 1, \ldots, n$ (as $\exists$ can always play in positions of type $(\heartsuit(B_1, \ldots, B_n), (\Delta, a))$). Thus, $(A, (\Delta, a))$ is a winning position for $\exists$.

It therefore follows that $\exists$ wins all finite $\mathcal{MG}_\Gamma(\mathbb{Y})$-plays that start at $(A_0, (\Delta_0, a_0))$ and are played according to $\tilde{f}$. Now consider an infinite $\mathcal{MG}_\Gamma(\mathbb{Y})$-play starting at $(A_0, (\Delta_0, a_0))$ and played according to $\tilde{f}$, and let $\pi$ be the infinite $\mathcal{G}_\Gamma$-play and $\tau$ be the associated trace through $\pi$ provided by

Lemma 5.16. It follows from the statement of the lemma that $\tau$ is contractable to the sequence of formulas appearing in the given $\mathcal{MG}_\Gamma(\mathbb{Y})$-play. Since the strategy $f$ was winning for $\exists$ in $\mathcal{G}_\Gamma$, it follows that any trace through $\pi$, and therefore also $\tau$, satisfies the parity condition of $\mathcal{G}_\Gamma$. As a result, the parity condition of $\mathcal{MG}_\Gamma(\mathbb{Y})$ is satisfied by the given infinite $\mathcal{MG}_\Gamma(\mathbb{Y})$-play, which is thus won by $\exists$.                                                                       □

Theorem 5.8 now follows from Theorem 5.17 and the observation that the sizes of both $Q$ and $\mathsf{S}(\Gamma)$ are bounded by an exponential in the size of $\mathrm{Cl}(\Gamma)$ (by Lemma 5.2 and respectively the definition of $\mathsf{S}(\Gamma)$).

Putting everything together, we obtain a complete characterisation of satisfiability in the coalgebraic $\mu$-calculus.

**Theorem 5.18.** *Suppose that* $\Gamma \in \mathsf{S}(\Lambda)$ *is a clean, guarded sequent and* $\mathsf{R}$ *is one-step tableau complete and contraction closed. Then* $\Gamma$ *is satisfiable iff no tableau for* $\Gamma$ *is closed iff* $\exists$ *has a winning strategy in the tableau game* $\mathcal{G}_\Gamma$.

As a by-product, we obtain the following small model property.

**Corollary 5.19.** *A satisfiable, clean and guarded formula* $A$ *is satisfiable in a model of size* $\mathcal{O}(2^{p(n)})$ *where* $n$ *is the cardinality of* $\mathrm{Cl}(A)$ *and* $p$ *is a polynomial.*

*Proof.* The statement follows immediately from Theorems 4.14, 5.5 and 5.8 together with the determinacy of two player parity games.                                                                       □

## 6. COMPLEXITY

We now show that – subject to a mild condition on the rule set – the satisfiability problem for guarded formulas of the coalgebraic $\mu$-calculus is decidable in exponential time. By Theorem 5.18, the satisfiability problem is reducible to the existence of winning strategies in parity games. Given any guarded sequent $\Gamma$, we thus construct a parity game of exponential size (measured in the size of $\Gamma$), the parity function of which has polynomial range (again measured relative to the size of $\Gamma$). This will ensure EXPTIME-decidability if we can decide legal moves in this game in exponential time. According to Definition 5.3, the game board consists of the disjoint union of

- $\mathsf{S}(\Gamma) \times Q$ (the positions owned by $\forall$) where $Q$ is the state set of a $\Gamma$-parity automaton and $\mathsf{S}(\Gamma)$ are the sequents that we can form in the closure of $\Gamma$, and
- $\mathsf{S}(\Gamma) \times \mathcal{B}(\Gamma) \times Q$ where $\mathcal{B}(\Gamma)$ are the blueprints of rules with premise in $\mathsf{S}(\Gamma)$.

We know that the state set $Q$ of the $\Gamma$-parity automaton is exponential in the size of $\mathrm{Cl}(\Gamma)$ by Lemma 5.2 and it is easy to see that $\mathsf{S}(\Gamma)$ is exponentially bounded. The crucial step for obtaining an overall exponential bound is thus the ability to treat rule blueprints. While this is simple for many logics (where it is easy to see one only has exponentially many applicable rule/substitution pairs that are of polynomial size), more care is needed for the rules of the probabilistic and the graded $\mu$-calculus. The main difficulty lies in the fact that the conclusions of these rules (Example 4.5) are sets of sequents that may be exponentially large. On the other hand, the conclusions can be represented by (small) linear inequalities, as in fact we did in Example 4.5 for presentational purposes, and leads to an obvious solution. Instead of representing rule blueprints associated with modal rules directly, we use a coding of modal rules that can be decided efficiently, to obtain an exponential representation of the game board. This approach has been used previously in [24] to give PSPACE-bounds for coalgebraic logics, and we will refer to *op.cit.* for some of the technical points.

In order to be able to speak about the complexity of the satisfiability problem in a meaningful way, we begin by formalising the notion of size of formulas and sequents. To do this, we assume that the underlying similarity type $\Lambda$ is equipped with a size measure $s : \Lambda \to \mathbb{N}$ and measure the size of a formula $A$ in terms of the number of subformulas counted with multiplicities, adding $s(\heartsuit)$ for every occurrence of a modal operator $\heartsuit \in \Lambda$ in $A$. In the examples, we code numbers in binary, that is, $s(\langle k \rangle) = s([k]) = \lceil \log_2 k \rceil$ for the graded $\mu$-calculus and $s(\langle p/q \rangle) = s([p/q]) = \lceil \log_2 p \rceil + \lceil \log_2 q \rceil + 1$ for the probabilistic $\mu$-calculus, and $s([a_1, \ldots, a_k]) = 1$ for coalition logic. Note that in the latter case, the overall number of agents is fixed, so there will only be finitely many coalitions which allows us to assign unit size to every operator. The definition of size is extended to sequents by $\mathsf{size}(\Gamma) = \sum_{A \in \Gamma} \mathsf{size}(A)$ for $\Gamma \in \mathsf{S}(\Lambda)$ and $\mathsf{size}(\{\Gamma_1, \ldots, \Gamma_n\}) = \sum_{i=1}^{n} \mathsf{size}(\Gamma_i)$ for sets of sequents.

We continue by discussing the mechanism to encode rule blueprints that we did describe informally at the beginning of this section. In order to obtain an exponential bound, we require that blueprints of modal rules can be encoded by strings of polynomial length. In order to have a uniform treatment, we make the following definition.

**Definition 6.1.** Suppose that $\Gamma \in \mathsf{S}(\Lambda)$. A set $\mathsf{R}$ of one-step rules is *exponentially tractable* if there is an alphabet $\Sigma$ and a polynomial $p$ such that every $\flat = (r, \sigma)$ with $r = \Gamma_0/\Gamma_1 \ldots \Gamma_n$ can be encoded as a string of length $\leq p(\mathsf{size}(\Gamma_0 \sigma))$ and the relations

$$R_1 = \{(\Delta, (\Gamma_0/\Gamma_1 \ldots \Gamma_n, \sigma)) \mid \Gamma_0 \sigma \subseteq \Delta\}$$

and

$$R_2 = \{((\Delta, \flat), \Delta') \mid \Delta' \text{ is } i\text{-th conclusion of } \rho(\Delta, \flat)\}$$

are decidable in EXPTIME (modulo this coding) for all $i \in \mathbb{N}$.

Exponential tractability gives an upper bound on the size of the board of the tableau game and the complexity of both the parity function and the relation determining legal moves. The proof of this result requires the following auxiliary lemmas thate establish bounds on the closure of the root sequent, and the size of the sequents in the closure, respectively.

**Lemma 6.2.** *Suppose $A \in \mathcal{F}(\Lambda)$. Then $|\mathrm{Cl}(A)| \leq \mathsf{size}(A)$.*

*Proof.* By induction on the structure of $A$ where the only non-trivial case is $A = \eta p.B$ for $\eta \in \{\mu, \nu\}$. To establish the claim, we show that $D = \{C[p := \eta p.A] \mid C \in \mathrm{Cl}(B)\}$ is closed. This implies that $\mathrm{Cl}(A) \subseteq D$ and the claim follows from the induction hypothesis. $\square$

**Lemma 6.3.** *If $\Gamma \in \mathsf{S}(\Lambda)$ and $\Delta \in \mathsf{S}(\Gamma)$ then $\mathsf{size}(\Delta) \leq \mathsf{size}(\Gamma)^3$.*

*Proof.* The closure of $\Gamma$ has at most $\mathsf{size}(\Gamma)$ many elements, each of which may be larger than $\mathsf{size}(\Gamma)$ as a result of substituting $\mu p.A$ for $p$ in $A$ if $\mu p.A \in \Gamma$. The result follows as this can happen at most $\mathsf{size}(\Gamma)$-many times. $\square$

We can now formulate, and prove, the annonced encoding of the tableau game as follows.

**Lemma 6.4.** *Suppose that $\mathsf{R}$ is exponentially tractable. Then every position in the tableau game $\mathsf{G}_\Gamma = (B_\exists, B_\forall, E, \Omega)$ of $\Gamma \in \mathsf{S}(\Lambda)$ can be represented by a string of polynomial length in $\mathsf{size}(\Gamma)$. Under this coding, the relation $(b, b') \in E$ is decidable in exponential time.*

*Proof.* We know that the state set $A$ of the parity automaton $A$ associated with $\mathsf{G}_\Gamma$ is exponential in $\mathsf{size}(\Gamma)$, hence every $a \in A$ can be represented by a string of polynomial length in $\mathsf{size}(\Gamma)$.

As we are now working with the encoding of the game board we think of the automaton as operating on encodings of rule blueprints rather than on the rule blueprints itself. More precisely,

we run the automaton not on trace tiles $(\Delta, \flat, i)$ but on encoded trace tiles $(\mathrm{code}(\Delta), \mathrm{code}(\flat), i)$ where $\mathrm{code}(\Delta)$ is the given encoding of sequents in $\mathsf{S}(\Gamma)$ and $\mathrm{code}(\flat)$ is the encoding of $\flat = (r, \sigma)$ according to Definition 6.1 if $\flat$ encodes a modal rule or $\mathrm{code}(\flat)$ is the principal formula of the (non-modal) rule represented by $\flat$ otherwise.

Every element of the set $\mathsf{S}(\Gamma)$ can be encoded by a string of polynomial length in $\mathsf{size}(\Gamma)$ by Lemma 6.3. Thus every position $(\Delta, a)$ of $B_\exists$ can be encoded by a string of polynomial length.

By exponential tractability, every rule blueprint $\flat$ can be encoded as a string of polynomial length, leading premise, leading to a polynomial bound on the size of the positions $(\Delta, \flat, a)$ of $B_\forall$.

To see that $E$ is decidable in exponential time, note that it follows from exponential tractability that the moves of $\forall$ from $(\Delta, a)$ to $(\Delta, \flat, b)$ are decidable in EXPTIME by Definition of tractability. To ensure EXPTIME decidablity of a move from $(\Delta, \flat, a)$ to $(\Delta', a')$ where $\flat$ is a blueprint of a modal rule, note that the rule represented by $(\Delta, \flat)$ has at most exponentially many conclusions (measured in the size of $\Delta$), and as we can check whether $\Delta'$ is the $i$-th conclusion of $\rho(\Delta, \flat)$ in exponential time, we conclude that $E$ is decidable in EXPTIME overall. $\qquad\square$

We now obtain an EXPTIME upper bound for satisfiability.

**Corollary 6.5.** *Suppose $T$ is a monotone $\Lambda$-structure and $\mathsf{R}$ is exponentially tractable, contraction closed and one-step tableau complete for $T$. Then the problem of deciding whether $\exists$ has a winning strategy in the tableau game for a clean, guarded sequent $\Gamma \in \mathsf{S}(\Lambda)$ is in EXPTIME. As a consequence, the same holds for satisfiability of any guarded formula $A \in \mathcal{F}(\Lambda)$.*

*Proof.* The first assertion follows from Lemma 6.4 as the problem of deciding the winner in a parity game is exponential only in the size of the parity function of the game (Theorem 3.1) which is polynomial in the size of $\Gamma$ (Lemma 5.2). The second statement now follows with the help of Theorem 5.18. $\qquad\square$

**Example 6.6.** It is easy to see that the rule sets for the modal $\mu$-calculus, the coalitional $\mu$-calculus and the monotone $\mu$-calculus are exponentially tractable, as the number of conclusions of each one-step rule is bounded. To establish exponential tractability for the rule sets for the graded and probabilistic $\mu$-calculus, we argue as in [24] where tractability of the (dual) proof rules has been established. We encode a rule with premise $\sum_{i=1}^{n} r_i a_i < k$ as $(r_1, a_1, \ldots, r_n, a_n, k)$ and Lemma 6.16 of *op. cit.* provides a polynomial bound on the size of the solutions for the linear inequalities that combine conclusion and side condition of both the $(G)$ and $(P)$-rule. Exponential tractability follows, once we agree on a fixed order on the set of prime implicants. In all cases, contraction closure is immediate.

## 7. Conclusions

In this paper, we have introduced the coalgebraic $\mu$-calculus that provides a generic and uniform framework for modal fixpoint logics. The calculus takes three parameters:

- an endofunctor $T : \mathsf{Set} \to \mathsf{Set}$ that defines the class of $T$-coalgebras over which the calculus is interpreted
- a collection $\Lambda$ of modal operators that defines the syntax of the calculus, and
- the interpretation of the modal operators over $T$-coalgebras, which is given by predicate liftings for $T$.

In this general setting, our main results are soundness and completeness of of the calculus and EXPTIME decidability of the satisfiability problem for guarded formulas. Technically, completeness

was achieved by tracking the evolution of fixpoint formulas in a tableau, and for a closed tableau we require that an outermost least fixpoint is unfolded along every infinite branch. To detect these infinite unfoldings of least fixpoints, we use a parity automaton that we run in parallel with the tableau, so that the existence of closed tableaux can be characterised by winning strategies in a parity game that is played on pairs consisting of a sequent and an automaton state. Our treatment borrows from by [17] and [26], but there are some important differences. In contrast to [17], we use parity games that directly correspond to tableaux, together with parity automata to detect bad traces. Moreover, our model construction super-imposes a coalgebra structure on the relation induced by a winning strategy for $\exists$. This model construction is substantially more involved than that given in [24], since we cannot argue in terms of modal rank in the presence of fixpoints. Compared with [26] (where no complexity results are presented), we use standard syntax for modal operators, which allows us to subsume for instance the graded $\mu$-calculus that cannot be expressed in terms of the $\nabla$-operator used in *op. cit.*. By instantiating the generic approach to specific logics, that is, by providing instances of the endofunctor $T$, the set $\Lambda$ of modal operators and the one-step rules R, we

- reproduce the complexity bound for the modal $\mu$-calculus [9], together with the completeness of a slight variant of the tableau calculus presented in [17],
- lead to a new proof of the known EXPTIME bound for the graded $\mu$-calculus [21],
- establish previously unknown EXPTIME bounds for the probabilistic $\mu$-calculus, for coalition logic with fixpoints and for the monotone $\mu$-calculus.

We note that these bounds are tight for all logics except possibly the monotone $\mu$-calculus, as the modal $\mu$-calculus can be encoded into all other logics. Given that the coalgebraic framework is inherently compositional [6, 3, 5, 23], our results also apply to (coalgebraic) logics that arise by combining various features, such as strategic games and quantitative uncertainty.

As mentioned before we would like to stress that we established the EXPTIME bound only for the guarded formulas of the above listed logics. Under the frequently used assumption that one can transform an arbitrary formula into an equivalent guarded one in polynomial or even linear time, we could extend our results to the full logics. In particular, note that in [21] precisely this assumption has been used for the graded $\mu$-calculus. For the modal $\mu$-calculus a tableau-based EXPTIME-procedure that works for arbitrary formulas as input has been presented recently in [11]. After careful inspection of our calculus we conjecture that our tableau calculus is also sound and complete for arbitrary formulas and formula sequents. We have to leave the details of the substantially more complicated completeness proof for this general case as future work.

## REFERENCES

[1] J. C. Bradfield. On the expressivity of the modal mu-calculus. In C. Puech and R. Reischuk, editors, *Proc. STACS 1996*, volume 1046 of *Lecture Notes in Computer Science*, pages 479–490. Springer, 1996.

[2] B. Chellas. *Modal Logic*. Cambridge, 1980.

[3] C. Cîrstea. A compositional approach to defining logics for coalgebras. *Theoret. Comput. Sci.*, 327:45–69, 2004.

[4] C. Cîrstea, C. Kupke, and D. Pattinson. EXPTIME tableaux for the coalgebraic $\mu$-calculus. In *Proceeding of Computer Science Logic, CSL 09*, volume 5771 of *LNCS*, pages 179–193, 2009.

[5] C. Cirstea and D. Pattinson. Modular proof systems for coalgebraic logics. *Theoretical Computer Science*, 388:83–108, 2007.

[6] C. Cîrstea and M. Sadrzadeh. Modular Games for Coalgebraic Fixed Point Logics. In J. Adámek and C. Kupke, editors, *Coalgebraic Methods in Computer Science (CMCS'2008)*, volume 203 of *ENTCS*, 2008.

[7] E. Emerson and C. Jutla. The complexity of tree automata and logics of programs. In *Proc. FOCS 1988*, pages 328–337. IEEE, 1988.

[8] E. Emerson and C. Jutla. Tree automata, mu-calculus and determinacy. In *Proceedings of the 32nd IEEE Symposium on Foundations of Computer Science (FoCS'91)*, pages 368–377. IEEE Computer Society Press, 1991.

[9] E. A. Emerson and C. S. Jutla. The complexity of tree automata and logics of programs. *SIAM J. Comput.*, 29(1):132–158, 1999.

[10] K. Fine. In so many possible worlds. *Notre Dame J. Formal Logic*, 13:516–520, 1972.

[11] O. Friedmann and M. Lange. The modal $\mu$-calculus caught off guard. In *Proceedings of the 20th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX)*, 2011. To appear.

[12] H. H. Hansen and C. Kupke. A coalgebraic perspective on monotone modal logic. In J. Adámek and S. Milius, editors, *Coalgebraic Methods in Computer Science*, volume 106 of *ENTCS*, pages 121–143. Elsevier, 2004.

[13] M. Jurdziński. Small Progress Measures for Solving Parity Games. In *Proceedings of the 17th Annual Symposium on Theoretical Aspects of Computer Science, STACS*, volume 1770 of *LNCS*, pages 290–301, 2000.

[14] D. Kozen. Results on the propositional $\mu$-calculus. *Theoret. Comput. Sci.*, 27:333–354, 1983.

[15] O. Kupferman, M. Vardi, and P. Wolper. An automata-theoretic approach to branching-time model checking. *Journal of the ACM*, 47(2):312–360, 2000.

[16] A. Mostowski. Games with forbidden positions. Technical Report 78, Instytut Matematyki, Uniwersytet Gdański, Poland, 1991.

[17] D. Niwinski and I. Walukiewicz. Games for the mu-calculus. *Theor. Comput. Sci.*, 163(1&2):99–116, 1996.

[18] M. Pauly. A modal logic for coalitional power in games. *J. Logic Comput.*, 12(1):149–166, 2002.

[19] N. Piterman. From nondeterministic Büchi and Streett automata to deterministic parity automata. In *Proceedings of the Twentyfirst Annual IEEE Symposium on Logic in Computer Science (LICS 2006)*, pages 255–264. IEEE Computer Society, 2006.

[20] S. Safra. On the complexity of $\omega$-automata. In *Proc. 29th IEEE Symposium on the Foundations of Computer Science*, pages 319–327, 1988.

[21] U. Sattler, O. Kupferman, and M. Y. Vardi. The complexity of the graded mu-calculus. In *Proc. CADE 2002*, volume 2392 of *LNCS*, pages 423–437. Springer, 2002.

[22] L. Schröder. A finite model construction for coalgebraic modal logic. In L. Aceto and A. Ingólfsdóttir, editors, *Foundations Of Software Science And Computation Structures*, volume 3921 of *LNCS*, pages 157–171. Springer, 2006.

[23] L. Schröder and D. Pattinson. Compositional algorithms for heterogeneous modal logics. In *Proc. ICALP 2007*, LNCS, 2007.

[24] L. Schröder and D. Pattinson. PSPACE bounds for rank-1 modal logics. *ACM Trans. Compl Log.*, 2(10), 2008. to appear.

[25] C. Stirling. *Modal and Temporal Properties of Processes*. Texts in Computer Science. Springer, 2001.

[26] Y. Venema. Automata and fixed point logics: a coalgebraic perspective. *Inform. Comput.*, 204(4):637–678, 2006.

[27] I. Walukiewicz. Completeness of Kozen's axiomatisation of the propositional $\mu$-calculus. *Inf. Comput.*, 157(1-2):142–182, 2000.