Completeness for coalgebraic fixpoint logic

Sebastian Enqvist*[1,2], Fatemeh Seifan†[1], and Yde Venema‡[1]

1 **ILLC, Universiteit van Amsterdam, The Netherlands**
2 **Department of Philosophy, Lund University, Sweden**

Abstract

We introduce an axiomatization for the coalgebraic fixed point logic which was introduced by Venema as a generalization, based on Moss' coalgebraic modality, of the well-known modal mu-calculus. Our axiomatization can be seen as a generalization of Kozen's proof system for the modal mu-calculus to the coalgebraic level of generality. It consists of a complete axiomatization for Moss' modality, extended with Kozen's axiom and rule for the fixpoint operators. Our main result is a completeness theorem stating that, for functors that preserve weak pullbacks and restrict to finite sets, our axiomatization is sound and complete for the standard interpretation of the language in coalgebraic models. Our proof is based on automata-theoretic ideas: in particular, we introduce the notion of consequence game for modal automata, which plays a crucial role in the proof of our main result. The result generalizes the celebrated Kozen-Walukiewicz completeness theorem for the modal mu-calculus, and our automata-theoretic methods simplify parts of Walukiewicz' proof.

1998 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases μ-calculus, coalgebra, coalgebraic modal logic, automata, completeness.

Digital Object Identifier 10.4230/LIPIcs.CSL.2016.23

1 Introduction

The modal μ-calculus (μML) was introduced Kozen [7] as a logic obtained by adding least and greatest fixpoint operators to modal logic. It is of great interest to computer science for expressing properties of processes such as termination and fairness. The power of the μ-calculus is also evident from a more theoretical perspective: adding fixpoint operators significantly increases the expressiveness of the formalism. In particular well-known temporal logics like LTL, CTL and CTL* can be defined in terms of the μ-calculus. A key result concerning the expressive strength of the μ-calculus is the Janin-Walukiewicz theorem [5], which states that a formula φ of monadic second-order logic (MSO) is equivalent to a μ-formula iff φ is invariant under bisimulation. Thus the μ-calculus seems a well-suited specification language, but there is a drawback: the μ-calculus is a complex formalism to work with.

A specific manifestation of this complexity lies in the axiomatization problem. In the same paper where he introduced the μ-calculus [7], Kozen also suggested an axiomatization and proved that his axiomatization is complete for a fragment of μML, consisting of the so-called *aconjunctive* formulas. The completeness of Kozen's axiomatization for the full language remained an open problem for many years, but eventually, Walukiewicz [18] provided a positive answer to this question. Regrettably his landmark result has remained a stand-alone

* sebastian.enqvist@fil.lu.se
† F.Seifan@uva.nl
‡ Y.Venema@uva.nl

theorem in the theory of (fixpoint) modal logic. This is largely due to the complexity of Walukiewicz' proof, which is based on an intricate mix of ideas from automata theory, game theory and logic (such as tableaux). It is the aim of our paper to clarify and generalize Walukiewicz' proof by focusing on ideas from automata theory and coalgebra.

Automata-theoretic methods lie on the heart of the theory of modal μ-calculus. In fact, most of the model-theoretic results on $\mu\mathtt{ML}$, such as D'Agostino and Hollenberg's characterization and uniform interpolation theorems [2] and the Janin-Walukiewicz theorem mentioned earlier, are based on Walukiewicz' result that formulas of $\mu\mathtt{ML}$ effectively correspond to what we shall call *disjunctive* automata: non-deterministic parity automata operating on Kripke models. This approach builds on a long tradition connecting logic and automata theory, going back to the seminal work of Büchi, Rabin and others. As examples we mention Rabins' decidability theorem [14], and the result by Büchi [1] showing that finite automata and monadic second order logic (MSO) have the same expressive power over infinite words.

Coalgebra enters this framework in a natural way. Recall from [15] that coalgebra uniformly generalizes state-based evolving systems such as streams, (infinite) trees, Kripke models, transition systems, and many others, by encoding the type of a dynamic system into a functor T. Starting with Moss' seminal paper [12], coalgebraic logics have been developed, and with Kripke structures constituting key examples of coalgebras, it should come as no surprise that most coalgebraic logics are some kind of modification or generalization of modal logic. Interestingly, Moss' logic centers around a coalgebraic generalization of the same cover modality ∇ that is the main operator featuring in Walukiewicz' completeness proof. Extending Moss' logic with fixpoint operators, Venema [17] introduced the coalgebraic fixpoint logic $\mu\mathtt{ML}_{\mathsf{T}}$, where T denotes the set functor encoding the coalgebra type. In the same paper Venema proved the existence of effective meaning preserving translations between formulas of $\mu\mathtt{ML}_{\mathsf{T}}$ and coalgebraic modal automata.

Here we address the completeness question for this coalgebraic fixpoint logic $\mu\mathtt{ML}_{\mathsf{T}}$. Note that a complete axiomatisation $\mathbf{M}$ for basic coalgebraic modal logic, i.e., the logic $\mu\mathtt{ML}_{\mathsf{T}}$ without the fixpoint operators, was given in [9]. Hence another way to view the result of this paper is that it extends the completeness result of [9] to the setting of fixpoint logic. In particular, our axiom system $\mathbf{K}$ is an extension of the system $\mathbf{M}$ with Kozen's axiom and derivation rule.

Just as in Walukiewicz' proof, we use translations between formulas of $\mu\mathtt{ML}_{\mathsf{T}}$ and automata. The difference is that we will be more radical and bring automata into the picture at an earlier stage. The main goal in Walukiewicz' proof strategy is to show that every formula of the μ-calculus *provably* implies a semantically equivalent disjunctive formula, that is, a modal μ-formula in a normal form corresponding to the disjunctive automata mentioned earlier. Here, we shall work with the full class of coalgebraic modal automata, for which an analogous result can be proved by much more elementary techniques. More specifically, our proof is based on translations between formulas of $\mu\mathtt{ML}_{\mathsf{T}}$ and coalgebraic automata (respectively denoted by $\mathbb{A}_\varphi$ for a formula $\varphi \in \mu\mathtt{ML}_{\mathsf{T}}$ and $\mathsf{tr}(\mathbb{A}) \in \mu\mathtt{ML}_{\mathsf{T}}$ for an automaton $\mathbb{A}$) for which we have the following proposition (with $\equiv_{\mathbf{K}}$ denoting provable equivalence with respect to system $\mathbf{K}$).

► **Proposition.** For every formula $\varphi \in \mu\mathtt{ML}_{\mathsf{T}}$, we have $\varphi \equiv_{\mathbf{K}} \mathsf{tr}(\mathbb{A}_\varphi)$.

This proposition takes us *half-way* towards Walukiewicz' result, and we address the remaining half of the distance by automata-theoretic techniques. In this way we can make the key concept of a *trace*, which is an essential but fairly informally discussed ingredient in Walukiewicz' proof, more explicit by developing a framework for 'managing' traces. Thus our machinery separates *dynamics* (coalgebra) from *combinatorics* (trace management) and this simplifies the proof in two ways. First, the coalgebraic perspective on automata uses a strictly controlled syntax and semantics via the so-called *one-step* framework, and second, it allows us to handle traces more explicitly. More in detail, our approach focus on two automata-related games: we will work with the *satisfiability game* of [3], which comprises a streamlined analogue of the logical notion of tableau, and we introduce the *consequence game* between two automata. Informally, the latter game, which resembles Walukiewicz' consequence game between tableaux, can be seen as a kind of implication game between the satisfiability games of two automata, revolving around establishing *structural connections* between the automata.

Making traces first-class citizens in our approach and bringing the trace theory to the surface, we will arrive at isolating classes of automata which allow a relatively simple trace management. The first automata that naturally appear are, once again, the *disjunctive automata*. These automata admit a *trivial* trace theory, in the sense that the matches of the satisfiability game of a disjunctive automaton involves a single trace only.

Bringing all these ideas together, as a key step in our completeness proof, we prove the following generalization of one of Walukiewicz' lemmas:

▶ **Theorem 1.1.** *For every formula $\varphi \in \mu\mathtt{ML}_{\mathsf{T}}$, there is a semantically equivalent disjunctive automaton $\mathbb{D}$ such that $\vdash_{\mathbf{K}} \varphi \to \mathsf{tr}(\mathbb{D})$.*

The proof of this theorem is by induction on the complexity of φ. In order to handle the induction steps we need to introduce a second class of special automata, namely *semi-disjunctive automata*, which roughly correspond to the *aconjunctive* formulas in Kozen's proof and the *weakly aconjunctive* formulas in the Walukiewicz' proof. These automata are more general than the disjunctive ones but still have a relatively *simple* (though not trivial) trace theory: the collection of *bad traces* in any match in the satisfiability game of a semi-disjunctive automaton is essentially finite. Finally, the completeness theorem itself is fairly straightforward corollary of Theorem 1.1.

Finally, we should mention that the completeness proof we provide here is still long and full of technical details, and for this reason we present this paper as an extended abstract. We hope that our 'deconstruction' of Walukiewicz' proof will contribute to a better understanding of the completeness theory of fixpoint logics. As a sample, in future work we will generalize this result to a wider coalgebraic context. As a first step we will use the results reported on here to provide a completeness result for *monotone fixpoint modal logic*.

Overview We first fix notation and terminology on Set-based functors and coalgebras. In section 3 we recall the definition of the coalgebraic fixpoint logic $\mu\mathtt{ML}_{\mathsf{T}}$ and we introduce the proof system **K**. Section 4 is concerned with the definition of the one-step framework and the coalgebraic modal automata corresponding to formulas of $\mu\mathtt{ML}_{\mathsf{T}}$. In this section we also define the satisfiability and consequence games for modal automata. Section 5 is devoted to the introduction of the disjunctive and semi-disjunctive automata. We discuss some of the closure properties of these automata and we state two of the main results of our paper, viz., Theorem 5.9 and Theorem 5.10. Finally, in section 6 we combine the results from the sections 4 and 5 in order to prove the completeness of the system **K** for $\mu\mathtt{ML}_{\mathsf{T}}$.

Acknowledgement The authors would like to thank the anonymous referees for many useful comments.

2 Preliminaries

In this section we settle on notation and terminology. For background on coalgebra the reader is referred to [15].

General In this paper we work with two categories: the category Set with sets as objects and functions between sets as arrows. The composition of two functions $f : X \to Y$ and $g : Y \to Z$ is written as $g \circ f$. For a function $f : X \to Y$ and a set $X' \subseteq X$ we define the restriction of f to X' as $f{\restriction_{X'}} : X' \to Y,\ x \mapsto f(x)$. The next category is the category Rel with sets as objects and relations as arrows. The composition of relations $R : X \nrightarrow Y$ and $S : Y \nrightarrow Z$ is written as $R\,;S$. Given a function f we use the same symbol f to refer to the graph of f.

Coalgebra A coalgebra (over Set) for a functor $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$, also called T-coalgebra, is a pair (X, σ) where S is a set (of states) and $\sigma : X \to \mathsf{T}X$ is a function (the transition structure). A T-coalgebra morphism from a T-coalgebra (X_1, σ_1) to a T-coalgebra (X_2, σ_2) is a function $f : X_1 \to X_2$ such that $\mathsf{T}f \circ \sigma_1 = \sigma_2 \circ f$. The prime examples of coalgebras for modal logicians are Kripke frames and Kripke models. Bisimulations between Kripke structures also have their natural coalgebraic generalization: a relation Z between the carrier sets of two coalgebras is a bisimulation if for all $(x_1, x_2) \in Z$, the pair $(\sigma_1(x_1), \sigma_2(x_2))$ belongs to the *lifting* $\overline{\mathsf{T}}Z$ of relation Z.

▶ **Definition 2.1.** Let T be a set functor. Given a binary relation Z between two sets X_1 and X_2, we define the relation $\overline{\mathsf{T}}Z : \mathsf{T}X_1 \nrightarrow \mathsf{T}X_2$ as follows:

$$\overline{\mathsf{T}}Z := \{((\mathsf{T}\pi_1)\rho, (\mathsf{T}\pi_2)\rho) \mid \rho \in \mathsf{T}Z\}$$

where $\pi_i : Z \to X_i$ for $i = 1, 2$ are the projection maps.

In this paper we will confine attention to set functors that preserve finite sets (that is, $\mathsf{T}X$ is a finite set if X is finite) and weak pullbacks. The latter property, which plays an important role in the theory of coalgebras, is defined as follows:

▶ **Definition 2.2.** A functor T preserves weak pullbacks if it maps every weak pullback (P, p_1, p_2) of maps $f_1 : X_1 \to Y$ and $f_2 : X_2 \to Y$ onto a weak pullback $(\mathsf{T}P, \mathsf{T}p_1, \mathsf{T}p_2)$ of $\mathsf{T}f_1 : \mathsf{T}X_1 \to \mathsf{T}Y$ and $\mathsf{T}f_2 : \mathsf{T}X_2 \to \mathsf{T}Y$.

▶ **Convention 2.3.** Throughout this paper we fix a functor T that preserves weak pullbacks and finite sets. We also assume that T preserves inclusions and finite intersections, but this is without loss of generality (see Convention 2.7. of [11]).

The following fact lists the properties of relation lifting $\overline{\mathsf{T}}$ that we use in our paper. For proofs we refer to [12] and references therein.

▶ **Fact 2.4.** *Let T be a set functor that preserves inclusions and weak pullbacks. Then relation lifting $\overline{\mathsf{T}}$*
(1) extends T: $\overline{\mathsf{T}}f = \mathsf{T}f$;
(2) is monotone: $R \subseteq Q$ implies $\overline{\mathsf{T}}R \subseteq \overline{\mathsf{T}}Q$;
(3) commutes with taking restrictions: $\overline{\mathsf{T}}(R{\restriction_{X \times X'}}) = (\overline{\mathsf{T}}R){\restriction_{\mathsf{T}X \times \mathsf{T}X'}}$;
(4) preserves composition: $\overline{\mathsf{T}}(R\,;Q) = \overline{\mathsf{T}}R\,;\overline{\mathsf{T}}Q$ and converse: $\overline{\mathsf{T}}(R^\circ) = (\overline{\mathsf{T}}R)^\circ$.

Lifting of special relations, like the membership relation, is used to define notions that will be used in the next section.

We first define the notion of *Base*.

▶ **Definition 2.5.** Given a functor T we define its subfunctor T_ω to be the functor that maps a set X to $\mathsf{T}_\omega X = \bigcup\{\mathsf{T}X' \mid X' \subseteq X,\ X' \text{ is finite }\}$. Then for every set X we define the function

$$Base_X : \mathsf{T}_\omega X \to \mathcal{P}_\omega X,\ \alpha \mapsto \bigcap\{X' \subseteq X \mid \alpha \in \mathsf{T}X'\}.$$

One may prove that $Base_X(\alpha)$ is the smallest set $U \in \mathcal{P}_\omega X$ such that $\alpha \in \mathsf{T}_\omega U$.

▶ **Definition 2.6.** Given a set X, we let $\in_X \subseteq X \times \mathcal{P}X$ denote the membership relation, restricted to X. We define the maps $\lambda^{\mathsf{T}}_X : \mathsf{T}\mathcal{P}X \to \mathcal{P}\mathsf{T}X$ by

$$\lambda^{\mathsf{T}}_X(\Phi) := \{\alpha \in \mathsf{T}X \mid \alpha\ \overline{\mathsf{T}}{\in_X}\ \Phi\}$$

and call members of $\lambda^{\mathsf{T}}_X(\Phi)$ *lifted members* of Φ. An object $\Phi \in \mathsf{T}\mathcal{P}X$ is a *redistribution of* $\Gamma \in \mathcal{P}\mathsf{T}X$ if $\Gamma \subseteq \lambda^{\mathsf{T}}_X(\Phi)$. In case $\Gamma \in \mathcal{P}_\omega \mathsf{T}_\omega X$, we call a redistribution Φ *slim* if $\Phi \in \mathsf{T}_\omega \mathcal{P}_\omega(\bigcup_{\alpha\in\Gamma} Base(\alpha))$. The set of all slim redistributions of Γ is denoted as $SRD(\Gamma)$.

Slim redistributions will be later used in order to define an axiom system for the coalgebraic fixpoint logic.

3 Coalgebraic fixpoint logic

3.1 Syntax and semantics

In this section we recall from [17] the syntax and semantics of a version of coalgebraic fixpoint logic which is based on Moss modality.

▶ **Definition 3.1.** We fix an infinite set of propositional variables. The language $\mu\mathtt{ML}_{\mathsf{T}}$ of coalgebraic fixpoint formulas is defined by the following grammar:

$$\varphi ::= \bot \mid \top \mid p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \nabla\alpha \mid \neg\varphi \mid \mu p.\varphi \mid \nu p.\varphi$$

where p belongs to the set of propositional variables, and $\alpha \in \mathsf{T}_\omega(\mu\mathtt{ML}_{\mathsf{T}})$. There is a restriction on the formation of the formulas $\mu p.\varphi$ and $\nu p.\varphi$, namely, no occurrence of p in φ may be in the scope of an odd number of negations. We denote by $\mu\mathtt{ML}_{\mathsf{T}}(\mathtt{X})$ the set of formulas with free variables from set $\mathtt{X}$, and as a convention we usually use letters $p, q, r, \ldots$ to denote bound variables and $x, y, z, \ldots$ for free variables of formulas. [1]

Readers who worry about the well-definedness of the inductive clause for ∇ may observe that since T_ω is a *finitary* functor, what we are saying is simply that for any finite set X of formulas, any object $\alpha \in \mathsf{T}_\omega(X)$ is a formula as well. In fact, any $\alpha \in \mathsf{T}_\omega(\mu\mathtt{ML}_{\mathsf{T}})$ belongs to the set $\mathsf{T}_\omega(Base(\alpha))$, and we will call the formulas in $Base(\alpha)$ the *immediate subformulas* of the formula $\nabla\alpha$.

To introduce the semantics of $\mu\mathtt{ML}_{\mathsf{T}}$ we first define the notion of a T-model over a set $\mathtt{X}$ of propositional letters.

▶ **Definition 3.2.** A T-model $\mathbb{S} = (S, \sigma, m)$ is a T-coalgebra (S, σ) together with a marking $m : S \to \mathcal{P}\mathtt{X}$. It will be convenient to think of a T-model $\mathbb{S}$ as a coalgebra $\mathbb{S} = (S, \sigma_m)$ for functor $\mathsf{T}_{\mathtt{X}}$ defined by $\mathsf{T}_{\mathtt{X}}S := \mathcal{P}\mathtt{X} \times \mathsf{T}S$ where $\sigma_m : S \to \mathsf{T}_{\mathtt{X}}S$ is given by map (m, σ). It is obvious that any marking $m : S \to \mathcal{P}\mathtt{X}$ induces a valuation $V_m : \mathtt{X} \to \mathcal{P}S$ mapping p to the set $\{s \in S \mid p \in m(s)\}$.

[1] For a precise definition of the notions *scope* and *occurrence*, we inductively define the construction tree of a formula, where the children of a node labeled $\nabla\alpha$ are the formulas in $Base(\alpha)$. For the definition of free and bound variables see [17].

Using the relation lifting $\overline{\mathsf{T}}$ we define the semantics for the language $\mu\mathtt{ML}_{\mathsf{T}}(\mathtt{X})$ on T-models. Since apart from the nabla modality, the definition of the satisfaction relation $\Vdash_{V_m}$ is exactly the same as it is for the μ-calculus, here we only give the definition for the nabla modality [2]:

$$s \Vdash_{V_m} \nabla\alpha \text{ iff } (\sigma_m(s), \alpha) \in \overline{\mathsf{T}}(\Vdash_{V_m}).$$

See [11] for more details.

The standard modal μ-calculus is the logic $\mu\mathtt{ML}_{\mathcal{P}}$ for the power set functor $\mathcal{P}$. In this case ∇ is known as the *cover modality*. It can be expressed using $\Box$ and $\Diamond$:

$$\nabla A = \Box \bigvee A \wedge \bigwedge \Diamond A;$$

where $\Diamond A$ denotes the set $\{\Diamond a \mid a \in A\}$. Conversely we have:

$$\Diamond a = \nabla\{a, \top\} \quad \text{and} \quad \Box a = \nabla\emptyset \vee \{a\}.$$

It is not difficult to see that in this case $\mathcal{P}_{\mathtt{X}}$-coalgebras are standard Kripke models.

3.2 Derivation system K

Our derivation system **K** is the extension of the complete derivation system **M** for Moss' finitary logic [9][8] with rules and axioms for the fixpoint operators.

▶ **Definition 3.3.** The derivation system **K** which is uniformly parametric in the functor T is given by the following axioms and derivation rules, together with any complete set of axioms and rules for classical propositional logic. The rules and axioms of **K** are given in Table 1.

$(\Delta 1)$	$\dfrac{\{\varphi \to \psi \mid (\varphi, \psi) \in Z\}}{\nabla\alpha \to \nabla\beta}(\alpha, \beta) \in \overline{\mathsf{T}}Z$
$(\Delta 2)$	$\bigwedge\{\nabla\alpha \mid \alpha \in \Gamma\} \to \bigvee\{\nabla(\mathsf{T}\bigwedge)(\Phi) \mid \Phi \in \mathit{SRD}(\Gamma)\}$
$(\Delta 3)$	$\nabla(\mathsf{T}\bigvee)(\Phi) \to \bigvee\{\nabla\alpha \mid \alpha \overline{\mathsf{T}}{\in} \Phi\}$
(A_f)	$\varphi(\mu x.\varphi(x)) \to \mu x.\varphi(x)$
(R_f)	$\dfrac{\varphi(\psi) \to \psi}{\mu x.\varphi \to \psi}$

Table 1 Rules and axioms of the system **K**

The axioms $(\Delta 2)$ and $(\Delta 3)$ are governing the interaction of ∇ with conjunctions and disjunctions respectively and can be seen as *modal distributive laws*. Here we see conjunction and disjunction ($\bigwedge$ and $\bigvee$) as maps from $\mathcal{P}_\omega(\mu\mathtt{ML}_{\mathsf{T}})$ to $\mu\mathtt{ML}_{\mathsf{T}}$, so we can apply T to them and get maps $\mathsf{T}\bigwedge$ and $\mathsf{T}\bigvee$. The rule $(\Delta 1)$ can be read as a congruence and monotonicity rule in one. It has a side condition expressing that it may only be applied when the set of premisses

[2] Strictly speaking the clause for nabla in Definition 3.2 is not stated in a correct recursive way, since it makes use of the unrestricted satisfaction relation $\Vdash_{V_m}$ that has yet to be defined. We can only suppose that $\Vdash_{V_m}\!|_{S\times Base(\alpha)}$ is already defined. The actual recursive definition is that $s \Vdash_{V_m} \nabla\alpha$ iff $(\sigma(s), \alpha) \in \overline{\mathsf{T}}(\Vdash_{V_m}\!|_{S\times Base(\alpha)})$. One can apply Fact 2.4 item (3) to prove that this definition is equal to the clause given above.

is indexed by a relation Z such that (α, β) belongs to the lifted relation $\overline{\mathsf{T}}Z$. (A_f) and (R_f) are the standard axiom and rule for pre-fixpoint.

The notions of *derivability* with respect to this system is standard. If there is a derivation of the formula φ we write $\vdash_{\mathbf{K}} \varphi$. We write $\varphi \equiv_{\mathbf{K}} \psi$ in the case that both $\vdash_{\mathbf{K}} \varphi \to \psi$ and $\vdash_{\mathbf{K}} \psi \to \varphi$ hold. A formula φ is $\mathbf{K}$-*consistent* or simply *consistent* if $\varphi \to \bot$ is not derivable in $\mathbf{K}$.

▶ **Example 3.4.** In the case of the power set functor ($\mathsf{T} = \mathcal{P}$) the axioms ($\Delta 2$) and ($\Delta 3$) look as follows:

$$\bigwedge\{\nabla\alpha \mid \alpha \in \Gamma\} \to \bigvee\{\nabla\{\bigwedge\beta \mid \beta \in \Phi\} \mid \bigcup\Gamma = \bigcup\Phi \text{ and } \alpha \cap \beta \neq \emptyset \text{ for all } \alpha \in \Gamma, \beta \in \Phi\}$$

$$\nabla\{\bigvee\beta \mid \beta \in \Phi\} \to \bigvee\{\nabla\alpha \mid \alpha \subseteq \bigcup\Phi \text{ and } \alpha \cap \beta \neq \emptyset \text{ for all } \beta \in \Phi\}$$

4 Coalgebraic modal automata

In this section, we lift the well-known relation between automata and logic to a coalgebraic level. We first recall the notions of one-step syntax and semantics.

4.1 One-step logic

Given a set A, we define the set $\mathtt{Latt}(A)$ of lattice terms over A through the following grammar:

$$\pi ::= \bot \mid \top \mid a \mid \pi \wedge \pi \mid \pi \vee \pi,$$

where $a \in A$. Given two sets $\mathtt{X}$ and A, we define the set $\mathtt{1ML}_{\mathsf{T}}(\mathtt{X}, A)$ of *modal one-step formulas* over A with respect to $\mathtt{X}$ inductively by

$$\alpha ::= \bot \mid \top \mid p \mid \neg p \mid \nabla\beta \mid \alpha \wedge \alpha \mid \alpha \vee \alpha,$$

with $p \in \mathtt{X}$ and $\beta \in \mathsf{T}_\omega\mathtt{Latt}(A)$.

Any valuation $V : A \to \mathcal{P}S$ can be extended to a meaning function $[\![-]\!]^0_V : \mathtt{Latt}(A) \to \mathcal{P}S$ in the usual way. We write $S, s \Vdash^0_V \varphi$ to indicate $s \in [\![\varphi]\!]^0_V$. The meaning function $[\![-]\!]^0_V$ induces a map $[\![-]\!]^1_V : \mathtt{1ML}_{\mathsf{T}}(\mathtt{X}, A) \to \mathcal{P}\mathsf{T}_{\mathtt{X}}S$ interpreting one-step formulas as subsets of $\mathsf{T}_{\mathtt{X}}S$. Before giving the definition of $[\![-]\!]^1_V$ we recall that every $\tau \in \mathsf{T}_{\mathtt{X}}S$ is of the form $(\mathtt{Y}, \tau') \in \mathcal{P}\mathtt{X} \times \mathsf{T}S$.

Going back to the map $[\![-]\!]^1_V$, it has the usual clauses for conjunction and disjunction, and the following clauses for the propositional letters and the modal operator:

- $\tau = (\mathtt{Y}, \tau') \in [\![p]\!]^1_V$ iff $p \in \mathtt{Y}$
- $\tau = (\mathtt{Y}, \tau') \in [\![\neg p]\!]^1_V$ iff $p \notin \mathtt{Y}$
- $\tau = (\mathtt{Y}, \tau') \in [\![\nabla\beta]\!]^1_V$ iff $(\tau', \beta) \in \overline{\mathsf{T}}(\Vdash^0_V)$

We write $S, \tau \Vdash^1_V \varphi$ to indicate $\tau \in [\![\varphi]\!]^1_V$, and refer to this relation as the *one-step semantics*.

For technical reasons, we need the following *binary* version of the modal distributive law for one-step conjunctions.

▶ **Proposition 4.1.** *Given $\alpha_1, \alpha_2 \in \mathsf{T}_\omega\mathtt{Latt}(A)$ the following holds:*

$$\nabla\alpha_1 \wedge \nabla\alpha_2 \equiv_K \bigvee\{\nabla(\mathsf{T}\wedge)\alpha \mid \alpha \in \mathsf{T}(Base(\alpha_1) \times Base(\alpha_2)) \text{ and } \mathsf{T}\pi_i(\alpha) = \alpha_i \text{ for } i \in \{1, 2\}\}$$

where the conjunction on the right hand side is a map $\wedge : \mathtt{Latt}(A) \times \mathtt{Latt}(A) \to \mathtt{Latt}(A)$.

To prove this result one can use properties of weak pullback preserving functors and show that these formulas are semantically equivalent and then from the one-step completeness result of [9] derive that they are provably equivalent.

4.2 Modal automata

▶ **Definition 4.2.** A *(coalgebraic) modal* $\mathtt{X}$*-automaton* is a quadruple $\mathbb{A} = (A, \Theta, \Omega, a_I)$ such that A is a finite set of *states*, $\Theta : A \to \mathtt{1ML}_{\mathsf{T}}(\mathtt{X}, A)$ is the *transition map* of $\mathbb{A}$, $\mathtt{X}$ is the set of free variables of formulas in the range of the transition map Θ, $\Omega : A \to \omega$ is the *priority map* of $\mathbb{A}$ and a_I is the *initial state*. We define the notions of free and positive occurrences of x in $\mathbb{A}$ in the obvious way.

▶ **Definition 4.3.** The *(directed) graph* of $\mathbb{A}$ is the structure $(A, E_{\mathbb{A}})$, where $aE_{\mathbb{A}}b$ if a occurs in $\Theta(b)$, and we let $\lhd_{\mathbb{A}}$ denote the transitive closure of $E_{\mathbb{A}}$. If $a \lhd_{\mathbb{A}} b$ we say that a is *active* in b. We write $a \bowtie_{\mathbb{A}} b$ if $a \lhd_{\mathbb{A}} b$ and $b \lhd_{\mathbb{A}} a$. A *cluster* of $\mathbb{A}$ is a cell of the equivalence relation generated by $\bowtie_{\mathbb{A}}$; a cluster C is *degenerate* if it is of the form $C = \{a\}$ with $a \not\bowtie_{\mathbb{A}} a$. Given a state a of $\mathbb{A}$, we write $\eta_a = \mu$ if $\Omega(a)$ is odd, and $\eta_a = \nu$ if $\Omega(a)$ is even, and we call state a a η_a-state. The sets of μ- and ν-states are denoted with A^μ and A^ν, respectively. For a state a we denote by C_a the unique cluster of $\mathbb{A}$ to which a belongs.

▶ **Definition 4.4.** Let $\mathbb{A} = (A, \Theta, \Omega, a_I)$ be a modal $\mathtt{X}$-automaton and let $\mathbb{S} = (S, \sigma, m)$ be a T-model. The associated acceptance game $\mathcal{A}(\mathbb{A}, \mathbb{S})$ is the parity game given by Table 2:

Position	Player	Admissible moves
$(a, s) \in A \times S$	$\exists$	$\{V : A \to \mathcal{P}S \mid \sigma_m(s) \in [\![\Theta(a)]\!]^1_V\}$
V	$\forall$	$\{(b, t) \in A \times S \mid t \in V(b)\}$

Table 2 Acceptance Game

The loser of a finite match is the player who got stuck; the winner of an infinite match is $\exists$ if the greatest parity that appears infinitely often in the match is even, and it is $\forall$ if this parity is odd. A pointed coalgebra $(\mathbb{S}, s_I)$ is *accepted* by the automaton $\mathbb{A}$ if (a_I, s_I) is a winning position for player $\exists$ in $\mathcal{A}(\mathbb{A}, \mathbb{S})$. We denote the *language recognized by* $\mathbb{A}$ with $L(\mathbb{A})$.

▶ **Fact 4.5.** *There are effective meaning preserving translations from coalgebraic fixpoint formulas to modal automata and vice versa [17].*

4.3 Satisfiability and consequence game

In this subsection we introduce two of our main tools, viz., the satisfiability and consequence game associated with modal automata. Before we can turn to the definition of these games we need some preliminary notions.

▶ **Definition 4.6.** Fix a set A. We let $A^\sharp$ denote the set of binary relations over A, that is, $A^\sharp := \mathcal{P}(A \times A)$. A *trace* through a finite word $R_1R_2R_3...R_k$ over $A^\sharp$ is a finite A-word $a_0a_1a_2...a_k$ such that $a_iR_{i+1}a_{i+1}$ for all $i < k$. A trace through an $A^\sharp$-stream $R_1R_2R_3...$ is an A-stream $a_0a_1a_2...$ such that $a_iR_{i+1}a_{i+1}$ for all $i < \omega$. Given a parity map $\Omega : A \to \omega$, with $NBT(A, \Omega)$ we denote the set of $A^\sharp$-streams $R_1R_2R_3...$ that contains no *bad trace*, that is, no trace $a_0a_1a_2...$ such that the greatest parity occurring infinitely often is odd.

The satisfiability game $\mathcal{S}(\mathbb{A})$ for a modal automaton $\mathbb{A}$ as introduced in [3] is a two-player graph game played by $\exists$ and $\forall$. We want this game to be such that $\exists$ has a winning strategy in $\mathcal{S}(\mathbb{A})$ iff there is a pointed coalgebra $\mathbb{S}$ that is accepted by $\mathbb{A}$. The idea behind the satisfiability game is to make a simultaneous projection of all matches of the acceptance game on this pointed coalgebra. More in particular, every match of the satisfiability game can be seen as a *bundle* of matches of the acceptance game for the automaton $\mathbb{A}$ on $\mathbb{S}$. To gather some intuition, suppose that a pointed coalgebra $\mathbb{S}$ is given and assume that $\exists$ has a winning strategy in the acceptance game for $\mathbb{A}$ and $\mathbb{S}$. Now we will see how $\exists$ can get a winning strategy in $\mathcal{S}(\mathbb{A})$.

Intuitively a basic position of $\mathcal{S}(\mathbb{A})$ should be represented as a subset B of A. We may associate with such a *macro-state* B a point $s \in S$ such that $\exists$ has to deal with positions (b, s) in the acceptance game, for all $b \in B$. For each $t \in S$ and for each $b \in B$, we define the set A^b_t as the collection of states $b' \in A$ such that (b', t) is a possible basic position in the acceptance game following the basic position (b, s). Since B is a macro-state, we define $A_t := \bigcup\{A^b_t \mid b \in B\}$. Hence, each such a set is a potential next combination of states in A that $\exists$ has to be able to handle simultaneously. In this set-up $\exists$'s move would be based on the set $\{A_t \mid t \in S\}$. Now it is up to $\forall$ to choose a set from this collection, moving to the next macro-state.

With this definition of the game, a match of $\mathcal{S}(\mathbb{A})$ corresponds to a sequence $\rho := B_0B_1B_2...$ of basic positions, which are subsets of A. Now to clarify whether ρ is won by $\exists$ we could naively say that $\exists$ wins if there is no bad trace $b_0b_1b_2...$ in ρ. However, if there is such a bad trace $b_0b_1b_2...$, this would only be a problem if it actually corresponds to a match of the acceptance game. Up to now we know that each b_i occurs in *some* match of the acceptance game, but there is no way to know whether $b_0b_1b_2...$ is the projection of an actual match of the acceptance game. This shows that defining the game based on subsets of A doesn't work properly. A solution to this problem is to replace the subset B by a relation $R \in A^\sharp$. The range of R would play the same role as B. This helps us to remember which traces are relevant, when we define the winning condition.

In the following we give the definition of satisfiability game and in Proposition 4.9 we state that as aimed for, $\exists$ has a winning strategy in $\mathcal{S}(\mathbb{A})$ iff there is a pointed coalgebra that is accepted by $\mathbb{A}$.

We first consider the one-step models based on the set $A^\sharp$ of binary relations over A.

▶ **Definition 4.7.** The *natural a-valuation* $V_a : A \to \mathcal{P}A^\sharp$ is given by

$$V_a : b \mapsto \{R \in A^\sharp \mid (a, b) \in R\}.$$

For $\alpha \in \mathsf{T}_{\mathtt{X}}A^\sharp$ and $\varphi \in \mathtt{1ML}_{\mathsf{T}}(\mathtt{X}, A)$, we write $\alpha \Vdash^1_a \varphi$ to denote that $A^\sharp, \alpha \Vdash^1_{V_a} \varphi$, and we define $[\![\varphi]\!]^1_a := \{\alpha \in \mathsf{T}_{\mathtt{X}}A^\sharp \mid \alpha \Vdash^1_a \varphi\}$.

▶ **Definition 4.8.** The satisfiability game associated with a modal $\mathtt{X}$-automaton $\mathbb{A} = (A, \Theta, \Omega, a_I)$ is denoted by $\mathcal{S}(\mathbb{A})$ and given by Table 3:

Position	Player	Admissible moves
$R \in A^\sharp$	$\exists$	$\bigcap_{a \in \mathsf{ran}(R)} [\![\Theta(a)]\!]^1_a$
$(\mathtt{Y}, \alpha) \in \mathsf{T}_{\mathtt{X}}(A^\sharp)$	$\forall$	$\{R \mid R \subseteq R' \text{ for some } R' \in Base(\alpha)\}$

Table 3 Satisfiability Game

Unless specified otherwise, we assume $\{(a_I, a_I)\}$ to be the starting position of $\mathcal{S}(\mathbb{A})$. An infinite match $R_1\alpha_1R_2\alpha_2R_3...$ is winning for $\exists$ if $R_1R_2R_3... \in NBT(A, \Omega)$.

▶ **Proposition 4.9.** *[Adequacy] Let $\mathbb{A}$ be a modal automaton. Then $\exists$ has a winning strategy in $\mathcal{S}(\mathbb{A})$ iff the language recognized by $\mathbb{A}$ is non-empty [3].*

As announced in our abstract and introduction, an important role in our approach is played by the *consequence game* $\mathcal{C}(\mathbb{A}, \mathbb{A}')$ associated with two automata $\mathbb{A}$ and $\mathbb{A}'$, which is played by two players I and II. One may think of player II trying to show that automaton $\mathbb{A}$ *implies* $\mathbb{A}'$ by establishing a close structural connection between the two automata, and of player I trying to show this does not hold.

Matches of the consequence game $\mathcal{C}(\mathbb{A}, \mathbb{A}')$ are tightly linked to the matches of the satisfiability games $\mathcal{S}(\mathbb{A})$ and $\mathcal{S}(\mathbb{A}')$ and this connection extends to the definition of the winning conditions of $\mathcal{C}(\mathbb{A}, \mathbb{A}')$ in terms of winning conditions of $\mathcal{S}(\mathbb{A})$ and $\mathcal{S}(\mathbb{A}')$. In fact the consequence $\mathcal{C}(\mathbb{A}, \mathbb{A}')$ is reminiscent of games defined by Santocanale, which go back to the literature on game semantics for linear logic (see [16] and references therein).

To describe the game we consider a match of $\mathcal{C}(\mathbb{A}, \mathbb{A}')$. Each round of this match consists of three moves. At the start of the round, at a basic position $(R, R') \in A^\sharp \times A'^\sharp$, player I picks a local model $\alpha \in \mathsf{T}_\mathsf{X} A^\sharp$ for formulas given by the range of R, as if she was player $\exists$ in the satisfiability game $\mathcal{S}(\mathbb{A})$. Second, player II transforms this one-step model into a model for formulas given by the range of R', inducing a move for $\exists$ in the satisfiability game $\mathcal{S}(\mathbb{A}')$. More precisely, player II provides a map $f : A^\sharp \to A'^\sharp$ turning α to a model for R'. The admissibility of this move reveals the essentially coalgebraic nature of the game, using the fact that T is actually a functor, i.e., it operates on arrows (that are, functions) as well as on objects (sets). More specfically, player II's move f is addmissible if the model α', that we obtain by applying the map $\mathsf{T}_\mathsf{X} f$ to the model α, is a model for R'. Player I then finishes the round by picking an element from the graph of map f as the next basic position.

▶ **Definition 4.10.** The *consequence game* $\mathcal{C}(\mathbb{A}, \mathbb{A}')$ between modal automata $\mathbb{A} = (A, \Theta, \Omega, a_I)$ and $\mathbb{A}' = (A', \Theta', \Omega', a'_I)$ is given by the following table.

Position	Player	Admissible moves
$(R, R') \in A^\sharp \times A'^\sharp$	I	$\bigcap_{a \in \mathsf{ran}(R)} [\![\Theta(a)]\!]^1_a$
$(\alpha, R') \in \mathsf{T}_\mathsf{X} A^\sharp \times A'^\sharp$	II	$\{f : A^\sharp \to A'^\sharp \mid \mathsf{T}_\mathsf{X} f(\alpha) \in \bigcap_{b \in \mathsf{ran}(R')} [\![\Theta'(b)]\!]^1_b\}$
$f : A^\sharp \to A'^\sharp$	I	$\{(R, R') \mid f(R) = R'\}$

Table 4 Consequence Game

As we already mentioned, pair of the form (R, R') in the above definition will be called a *basic position* of the consequence game. Similar to the satisfiability game, our standard assumption is that $(\{(a_I, a_I)\}, \{(a'_I, a'_I)\})$ is the starting position of $\mathcal{C}(\mathbb{A}, \mathbb{A}')$. We declare player I to be the winner of an infinite match $(R_1, R'_1)(R_2, R'_2)(R_2, R'_2)...$ if there exists a bad trace on the $\mathbb{A}'$-side, i.e. through $R'_1 R'_2 R'_3...$ but no bad trace on the $\mathbb{A}$-side i.e. through $R_1 R_2 R_3....$ In all other cases player II is the winner. Whenever II has a winning strategy in $\mathcal{C}(\mathbb{A}, \mathbb{A}')$ we say that $\mathbb{A}'$ is a *game consequence* of $\mathbb{A}$ and denote this fact with $\mathbb{A} \vDash_\mathcal{C} \mathbb{A}'$.

▶ **Proposition 4.11.** *Given automata $\mathbb{A}$ and $\mathbb{A}'$ we have that $\mathbb{A} \vDash_\mathcal{C} \mathbb{A}'$ implies $L(\mathbb{A}) \subseteq L(\mathbb{A}')$.*

Below Lemma 4.17 we shall see a counter-example to the converse of this proposition.

4.4 Formulas and automata

There are a few different methods for transforming a formula of the μ-calculus into an equivalent parity automaton [19][4]. The method used in [4] first constructs a tableau for the formula, which is then transformed into an automaton. Here, we shall instead use a direct translation from formulas to automata given by induction on the complexity of a formula [16]. We denote the translation of a formula φ by $\mathbb{A}_\varphi$. As a consequence of adequacy we get the following:

▶ **Fact 4.12.** *Given a formula φ and its translation $\mathbb{A}_\varphi$ we have that φ is satisfiable if and only if $\exists$ has a winning strategy in $\mathcal{S}(\mathbb{A})$ [3].*

For the opposite direction, we use a translation $\mathtt{tr}$ from automata to formulas. Unfortunately due to space limits we will not go through the definition of this translation, which is obtained using similar methods applied in [17]. The point about the translation $\mathtt{tr}$ is that it behaves well with respect to the notion of provability and enables us to apply proof-theoretic concepts, such as consistency, to automata. The key observation of this subsection is the following proposition.

▶ **Proposition 4.13.** *For every formula φ, we have $\varphi \equiv_{\mathbf{K}} \mathtt{tr}(\mathbb{A}_\varphi)$.*

We will define the translation from formulas to automata by induction on the complexity of a formula. We omit the base step and inductive cases of disjunction, conjunction and negation, check [6] and [10] for details. For formulas of the form $\varphi = \nabla\alpha$ the automaton $\mathbb{A}_\varphi$ is defined by using formulas in $Base(\alpha)$ and applying T to the translation map.

For fixpoint formulas of the form $\varphi = \eta x.\alpha$ with $\eta \in \{\nu, \mu\}$, the construction of $\eta x.\mathbb{A}$ starts by putting $\mathbb{A}$ into a suitable shape denoted by $\mathbb{A}^x$. The key observation about $\mathbb{A}^x$ is that the free variable x in $\mathbb{A}$ becomes in a certain sense *guarded* in $\mathbb{A}^x$. Since we do not allow variables to appear guarded in the one-step formulas in the image of the transition map of an automaton, we need to introduce a new state $\underline{x}$ that we use to represent the variable x. For the construction of $\mathbb{A}^x$ we will use the fact that for every automaton $\mathbb{A}$ with a free variable x and any state $a \in A$, there are formulas θ_0^a and θ_1^a in which x does not appear, such that $\Theta(a) \equiv_{\mathbf{K}} (x \wedge \theta_0^a) \vee \theta_1^a$.
The following definition explains this auxiliary automaton:

▶ **Definition 4.14.** Let $\mathbb{A}$ be a modal automaton in which the variable x is free, and assume without loss of generality, that the priority map Ω is injective, and the smallest priority in the image of Ω is greater than 0. Pick a new state $\underline{x} \notin A$. Then we define the automaton $\mathbb{A}^x = (A^x, \Theta^x, a_I^x, \Omega^x)$ as follows:

- $A^x = (A \times \{0,1\}) \cup \{\underline{x}\}$. We write (a,i) as a_i, for $i \in \{0,1\}$.
- $\Theta^x(a_0) = \theta_0^a[\kappa]$, $\Theta^x(a_1) = \theta_1^a[\kappa]$ and $\Theta^x(\underline{x}) = x$,
- $\Omega^x(a_i) = \Omega(a)$, $\Omega^x(\underline{x}) = 0$ and $a_I^x = (a_I)_1$.

Here, κ is defined to be the substitution $a \mapsto (\underline{x} \wedge a_0) \vee a_1$ for every a.

We are now ready to define fixpoint operations on automata:

▶ **Definition 4.15.** The automaton $\mu x.\mathbb{A} = (A', \Theta', \Omega', a_I')$ is defined by setting $A' = A^x$, $\Theta'(a_i) = \Theta^x(a_i)$ for $a \in A$, $\Theta'(\underline{x}) = \theta_1^{a_I}[\kappa]$, $a_I' = \underline{x}$, $\Omega'(a_i) = \Omega^x(a_i)$ and $\Omega^x(\underline{x}) = 2 \cdot \mathtt{max}(\Omega^x[A^x]) + 1$. The automaton $\nu x.\mathbb{A} = (A', \Theta', \Omega', a_I')$ is defined in the same way, except that $\Theta'(\underline{x}) = \theta_0^{a_I}[\kappa] \vee \theta_1^{a_I}[\kappa]$ and $\Omega^x(\underline{x}) = 2 \cdot \mathtt{max}(\Omega^x[A^x]) + 2$.

In the following we define the notion of substitution for modal automata.

▶ **Definition 4.16.** Let $\mathbb{A} = (A, \Theta_A, \Omega_A, a_I)$ and $\mathbb{B} = (B, \Theta_B, \Omega_B, b_I)$ be modal automata over the languages P and $P \setminus \{x\}$, respectively. Assume that $\mathbb{A}$ is positive in x. We define the modal $P \setminus \{x\}$-automaton $\mathbb{A}[\mathbb{B}/x]$ as the structure $(D, \Theta_D, \Omega_D, d_I)$, where $D := A \uplus B$, $d_I := a_I$, and the transition map Θ_D is given by

$$\Theta_D(d) := \begin{cases} \Theta_A(d)[\Theta_B(b_I)/x] & \text{if } d \in A \\ \Theta_B(d) & \text{if } d \in B, \end{cases}$$

and $\Omega_D := \Omega_A \uplus \Omega_B$.

The next lemma lists some of the properties of the automaton $\mu x.\mathbb{A}$:

▶ **Lemma 4.17.** *Given modal automata $\mathbb{A}$ with x free, we have*
(1) $\mathsf{tr}(\mu x.\mathbb{A}) \equiv_{\mathbf{K}} \mu x.\mathsf{tr}(\mathbb{A}^x)$*;*
(2) $\mathbb{A}^x[\mu x.\mathbb{A}/x] \vDash_{\mathcal{C}} \mu x.\mathbb{A}$*.*

In passing we note that if in the second item of the above lemma we replace $\mathbb{A}^x$ with $\mathbb{A}$, then it is generally not the case that $\mathbb{A}[\mu x.\mathbb{A}/x] \vDash_{\mathcal{C}} \mu x.\mathbb{A}$. This is a counter-example to the converse of Proposition 4.11.

The following lemma summarizes properties of the translation $\mathtt{tr}$ which are needed to prove Proposition 4.13.

▶ **Lemma 4.18.** *There exists a translation* $\mathtt{tr}$ *from modal automata to the formulas of* $\mu\mathtt{ML}_{\mathsf{T}}$ *and operators* $\wedge, \vee, \neg$ *and* ∇ *on automata such that the following claims hold:*
(1) For all $\mathbb{A}, \mathbb{B}$*,* $\mathsf{tr}(\mathbb{A} \wedge \mathbb{B}) \equiv_{\mathbf{K}} \mathsf{tr}(\mathbb{A}) \wedge \mathsf{tr}(\mathbb{B})$*,* $\mathsf{tr}(\mathbb{A} \vee \mathbb{B}) \equiv_{\mathbf{K}} \mathsf{tr}(\mathbb{A}) \vee \mathsf{tr}(\mathbb{B})$ *and* $\mathsf{tr}(\neg\mathbb{A}) \equiv_{\mathbf{K}} \neg\mathsf{tr}(\mathbb{A})$*;*
(2) Given automata $\mathbb{A}_1, \mathbb{A}_2, ..., \mathbb{A}_n$ *and* $\alpha \in \mathsf{T}\{\mathbb{A}_1, \mathbb{A}_2, ..., \mathbb{A}_n\}$ *then* $\mathsf{tr}(\nabla\alpha) \equiv_{\mathbf{K}} \nabla(\mathsf{T}\mathsf{tr})(\alpha)$*;*
(3) For every automaton $\mathbb{A}$ *with* x *free in* $\mathbb{A}$ *and* $\eta \in \{\nu, \mu\}$ *we have* $\mathsf{tr}(\eta x.\mathbb{A}) \equiv_{\mathbf{K}} \eta x.\mathsf{tr}(\mathbb{A})$*;*
(4) For all $\mathbb{A}, \mathbb{B}$ *with* x *free in* $\mathbb{A}$*, we have* $\mathsf{tr}(\mathbb{A}[\mathbb{B}/x]) \equiv_{\mathbf{K}} \mathsf{tr}(\mathbb{A})[\mathsf{tr}(\mathbb{B})/x]$*.*

5 Disjunctive and semi-disjunctive automata

Generally, the combinatorics of the trace graph(s) associated with the satisfiability and the consequence games are rather involved. As mentioned in the introduction, an important role in our proof of the Kozen-Walukiewicz theorem is played by two kinds of special automata that allow somewhat simpler trace graphs: *disjunctive* and *semi-disjunctive* automata. The conditions on these automata can be nicely expressed in terms of restrictions on the one-step language.

▶ **Definition 5.1.** Let $\mathbb{A}$ be a modal automaton and let C be a cluster of $\mathbb{A}$. An element $a \in C$ is called a *maximal even element* of C if it has the maximal priority in C, and this priority is even. A relation $R \in A^\sharp$ is *thin with respect to* $\mathbb{A}$ and a if:

(1) for all $b \in A$ with aRb we have $a \lhd_{\mathbb{A}} b$;
(2) for all $b_1, b_2 \in A$ with $b_1, b_2 \in R[a] \cap C_a$, either $b_1 = b_2$ or one of b_1 and b_2 is a maximal even element of C_a.

We call R $\mathbb{A}$-thin or simply thin, if it is thin with respect $\mathbb{A}$ and all $a \in A$.

A motivating observation about thin relations is the following.

▶ **Fact 5.2.** *For a stream* $\rho = R_1R_2R_3...$ *of thin relations there exists a finite collection* F *of traces on* ρ *such that any trace* t *on* ρ *is bad if and only if there is some* $t' \in F$ *cofinally equal to* t*.*

We are now ready to define *disjunctive* and *semi-disjunctive* automata.

▶ **Definition 5.3.** Given sets $\mathtt{X}$ and A we define the sets $\mathtt{LitC}(X)$ and $\mathtt{1ML}^d_{\mathsf{T}}(\mathtt{X}, A)$ by respectively:

$$\pi ::= \bot \mid \top \mid p \wedge \pi \mid \neg p \wedge \pi$$

and

$$\alpha ::= \bot \mid \pi \wedge \nabla\beta \mid \alpha \vee \alpha,$$

where $\pi \in \mathtt{LitC}(\mathtt{X})$ and $\beta \in \mathsf{T}A$. Elements of $\mathtt{1ML}^d_{\mathsf{T}}(\mathtt{X}, A)$ are called *one-step disjunctive formulas* and a modal automaton $\mathbb{A} = (A, \Theta, \Omega, a_I)$ is *disjunctive* if $\Theta(a)$ belongs to $\mathtt{1ML}^d_{\mathsf{T}}(\mathtt{X}, A)$ for all $a \in A$.

▶ **Definition 5.4.** Let $\mathbb{A}$ be a modal automaton an let C be a cluster of $\mathbb{A}$. The set of *(zero-step) C-safe conjunctions*, denoted by $\mathtt{Conj}^C_0(A)$ contains formulas of the form $\bigwedge B$ with $B \subseteq A$, such that for all $b_1 \neq b_2 \in B \cap C$, either b_1 or b_2 is a maximal even element of C. The grammar

$$\alpha ::= \bot \mid \pi \wedge \nabla\gamma \mid \alpha \vee \alpha,$$

where $\pi \in \mathtt{LitC}(\mathtt{X})$ and $\gamma \in \mathsf{T}\mathtt{Conj}^C_0(A)$, defines the set $\mathtt{1ML}^{s(C)}_{\mathsf{T}}(\mathtt{X}, A)$ of *one-step C-safe formulas*. We call a one-step formula α *semi-disjunctive with respect to* $a \in A$ if α is a C_a-safe formula. A modal automaton $\mathbb{A} = (A, \Theta, \Omega, a_I)$ is *semi-disjunctive* if $\Theta(a)$ is semi-disjunctive with respect to a for all $a \in A$.

The key property of these automata is that the matches of the satisfiability and consequence game are of a relatively simple shape. For disjunctive (respectively semi-disjunctive) automata, without loss of generality, we can always assume that these matches contain only functional (respectively thin) relations. This property allows us to work with a variant of the satisfiability game which is called *thin* satisfiability game.

▶ **Definition 5.5.** The *thin* satisfiability game $\mathcal{S}_{thin}(\mathbb{A})$ is the variant of the satisfiability game $\mathcal{S}(\mathbb{A})$ where $\forall$'s choice of moves is restricted to *thin* relations. A winning strategy for $\forall$ in $\mathcal{S}_{thin}(\mathbb{A})$ is called a *thin refutation* of $\mathbb{A}$.

Note that for an arbitrary automaton $\mathbb{A}$ it is not always the case that $\mathcal{S}(\mathbb{A})$ and $\mathcal{S}_{thin}(\mathbb{A})$ are equivalent, but for disjunctive and semi-disjunctive automata it holds:

▶ **Proposition 5.6.** *Given a semi-disjunctive automaton $\mathbb{A}$, then $\exists$ ($\forall$, respectively) has a winning strategy in $\mathcal{S}(\mathbb{A})$ if and only if $\exists$ ($\forall$, respectively) has a winning strategy in $\mathcal{S}_{thin}(\mathbb{A})$.*

In our paper we will use the construction given by [10, Definition 3.10] to transform a modal automaton $\mathbb{A}$ into a disjunctive automaton denoted by $\mathsf{sim}(\mathbb{A})$[3]. The proof of equivalence of $\mathbb{A}$ and $\mathsf{sim}(\mathbb{A})$ amounts to a coalgebraic version of the simulation theorem [13] for modal automata [10]. Roughly speaking, the idea of the proof is to define $\mathsf{sim}(\mathbb{A})$ via a variation of the power set construction such that a match of the acceptance game of $\mathsf{sim}(\mathbb{A})$ corresponds to $\exists$ simultaneously playing various matches of the acceptance game $\mathbb{A}$.

▶ **Fact 5.7.** *Let $\mathbb{A}$ be a modal automaton. Then $\mathsf{sim}(\mathbb{A})$ is a disjunctive automaton satisfying $L(\mathbb{A}) = L(\mathsf{sim}(\mathbb{A}))$.*

▶ **Proposition 5.8.** *The map $\mathsf{sim}(\cdot)$ assigns to each modal automaton $\mathbb{A}$ a disjunctive automaton $\mathsf{sim}(\mathbb{A})$ such that:*
(1) $\mathbb{A}$ and $\mathsf{sim}(\mathbb{A})$ are semantically equivalent;
(2) $\mathbb{A} \vDash_{\mathcal{C}} \mathsf{sim}(\mathbb{A})$.

The rest of this section is devoted to discuss other properties of disjunctive and semi-disjunctive automata which will later on be used to prove our main technical result, viz., Theorem 1.1. In the following theorem we show how to use the fact that an automaton $\mathbb{D}$ is a game consequence of an automaton $\mathbb{A}$, to find a thin refutation in the satisfiability game for $\mathbb{A} \wedge \neg\mathbb{D}$.

[3] Note that the approach in [10] does not explicitly use the ∇ operator but the automata in [10] can be seen as a notational variant of the ones employed here.

▶ **Theorem 5.9.** *Let $\mathbb{A}$ be a semi-disjunctive automaton and $\mathbb{D}$ be an arbitrary modal automaton such that $\mathbb{A} \models_{\mathcal{C}} \mathbb{D}$. Then $\mathbb{A} \wedge \neg\mathbb{D}$ has a thin refutation.*

This theorem is an automata-theoretic version of Lemma 36 from [18], one of the key lemmas of Walukiewicz' completeness proof, and at the same time it generalizes that result in two ways; first, our coalgebraic approach extends the result from the power set functor $\mathcal{P}$ to a set functor T, second, we prove the result for an arbitrary automaton $\mathbb{D}$ instead for a disjunctive one.

Proof sketch of Theorem 5.9. To fix notation, let $\mathbb{A} = (A, \Theta, \Omega, a_I)$ and $\mathbb{D} = (D, \Theta', \Omega', d_I)$. We recall that the transition map of the automaton $\neg\mathbb{D}$ is defined by taking boolean duals of the formulas assigned by the transition map of $\mathbb{D}$, while the priority map is defined by simply raising all priorities by 1.

Assume that $\mathbb{A} \models_{\mathcal{C}} \mathbb{D}$. Without loss of generality we can assume that II has a winning strategy involving thin relations only. To show that the automaton $\mathbb{A} \wedge \neg\mathbb{D}$ has a thin refutation, we will define a thin winning strategy χ for $\forall$ in $\mathcal{S}_{thin}(\mathbb{A} \wedge \neg\mathbb{D})$. Given a χ-guided partial match in $\mathcal{S}(\mathbb{A} \wedge \neg\mathbb{D})$ with basic positions $R_0 R_1 R_2 ... R_n$. Our aim is to introduce a response R_{n+1} for $\forall$ to every possible move γ by $\exists$, such that:
(i) R_{n+1} is a legitimate move, i.e., $R_{n+1} \subseteq R'$ for some $R' \in Base(\gamma)$;
(ii) $\mathsf{ran}(R_{n+1}) \cap D$ is a singleton;
(iii) R_{n+1} is thin.

We shall also maintain the induction hypothesis that for every χ-guided partial match $R_0, ..., R_n$ there is a shadow-match in the consequence game, guided by the winning strategy for player II, of the form $(S_0, S'_0)(S_1, S'_1)...(S_n, S'_n)$ where, for each i, we have $R_i \cap A^2 = S_i$ and $R_i \cap D^2 \subseteq S'_i$.

Going to the details of how we maintain these condition, we claim that:

▶ **Claim.** There is some $S \in Base(\gamma)$ and some $c \in D$ with $(d, c) \in f(S \cap A^2) \cap (S \cap D^2)$, where $f : A^\sharp \to D^\sharp$ is dictated by Player II's winning strategy in $\mathcal{C}(\mathbb{A}, \mathbb{D})$.

Proof is an exercise in one-step coalgebraic logic.

With this claim in place, we define the next move for $\forall$ by picking the relation $(S \cap A^2) \cup \{(d, c)\}$, where $S \in Base(\gamma)$ and $c \in D$ are as described in the claim, so that $(d, c) \in f(S \cap A^2) \cap (S \cap D^2)$. Then we prove that this a legitimate move for $\forall$ and the shadow-match which is then extended by the pair $(S \cap A^2, f(S \cap A^2))$ also satisfies condition (i)-(iii). Thin-ness of the strategy χ defined in this way follows from semi-disjunctivity of the automaton $\mathbb{A}$. Finally we show that every infinite χ-guided match ρ contains a bad trace. Note that any infinite match contains a unique trace in D, which will also be a trace on the right side of the shadow-match in the consequence game. If this trace is *not* bad given the priorities assigned to states in $\neg\mathbb{D}$, then it must be bad as a trace in $\mathbb{D}$ since parities are swapped in $\neg\mathbb{D}$. So there must be a bad trace on the left side of the shadow-match in the consequence game, since this shadow-match was guided by the winning strategy of II. But since every such trace corresponds to a trace through A in the satisfiability game, we see that either the unique trace through D in ρ is bad or there is some bad trace through A in ρ. In either case, there must be some bad trace in ρ, so $\forall$ wins. ◀

The next Theorem is a version of another key lemma in Walukiewicz' proof for modal automata, viz., [18, Lemma 39].

▶ **Theorem 5.10.** *Let $\mathbb{A}$ and $\mathbb{B}$ be arbitrary modal automata, and let x be a free propositional variable of $\mathbb{B}$. Then we have $\mathbb{B}[\mathsf{sim}(\mathbb{A})/x] \models_{\mathcal{C}} \mathbb{B}[\mathbb{A}/x]$.*

Proof sketch of Theorem 5.10. Starting with notation, let $\mathbb{A} = (A, \Theta_A, \Omega_A, a_I)$, $\mathbb{B} = (B, \Theta_B, \Omega_B, b_I)$ and $\mathsf{sim}(\mathbb{A}) = \mathbb{D} = (D, \Theta_D, \Omega_D, d_I)$. From the proof of Proposition 5.8 we get a map $G : D \to A^\sharp$ that *reflects traces* in the sense that if $G(d_i)_{i\in\omega} \in (A^\sharp)^\omega$ contains a bad $\mathbb{A}$-trace, then $(d_i)_{i\in\omega}$ is itself a bad $\mathbb{D}$-trace. By the proof of Proposition 5.8 part (2) this map G encodes a particularly simple winning strategy for player II in the consequence game $\mathcal{C}(\mathbb{D}, \mathbb{A})$. The trick of proving Theorem 5.10 is to turn this winning strategy encoded by G into a new winning strategy for player II in $\mathcal{C}(\mathbb{B}[\mathbb{D}/x], \mathbb{B}[\mathbb{A}/x])$.

Let us first clarify why the proof is not so straightforward. To see where the difficulties lie, consider an arbitrary infinite match $\rho = (R_n, R'_n)_{n\in\omega}$ of the consequence game for $\mathbb{B}[\mathbb{D}/p]$ and $\mathbb{B}[\mathbb{A}/p]$. Given the shape of these two automata, we may assume that traces on $\rho_l := R_0R_1\ldots$ consist of either a $\mathbb{B}$-stream, or of a finite $\mathbb{B}$-trace followed by an infinite $\mathbb{D}$-trace, and that traces on $\rho_r := R'_0R'_1\ldots$ are either a $\mathbb{B}$-stream or composed of a finite $\mathbb{B}$-trace and an $\mathbb{A}$-trace. Our purpose will be to associate with each ρ_r-trace

$\tau = b_0b_1\ldots b_na_{n+1}a_{n+2}a_{n+3}\ldots$, a ρ_l-trace $\tau_l = b_0b_1\ldots b_nd_{n+1}d_{n+2}d_{n+3}\ldots$, such that we can use the trace reflection on the $\mathbb{D}$- and $\mathbb{A}$-tail of τ and τ_l, respectively.

For this purpose we will define, for each partial match leading to final position (R_n, R'_n), a map $g_n : \mathsf{ran}_A R'_n \to \mathsf{ran}_D R_n$. Intuitively, for $a \in A$, $g_n(a)$ represents a $d \in D$ that 'implies' a, more precisely $a \in \mathsf{ran}G(g_n a)$. Ideally, we would like to show that the τ-tail $(a_i)_{i>n}$ is in fact a trace on the $A^\sharp$-stream $(G(g_ia_i))_{i>n}$, while $(g_ia_i)_{i>n}$ is an ρ_l-trace so that the trace reflection applies indeed.

Unfortunately, this is too good to be true, due to complications that are caused by $\mathbb{A}$-traces *merging*: the point is that *trace jumps* may occur, that is, situations where for some pair $(a, a') \in R'_{j+1}$ it does not hold that $(g_ja, g_{j+1}a') \in R_{j+1}$. Our solution to this problem will be to ensure that every ρ_r-trace can suffer only *finitely many* trace jumps. Thus, what we *can* show is that any $\mathbb{A}$-trace $a_0a_1\cdots$ has a *tail* $a_ka_{k+1}a_{k+2}\cdots$ which is a trace on $G(g_ka_k)G(g_{k+1}a_{k+1})G(g_{k+2}a_{k+2})\cdots$ This suffices to prove that if there is a bad trace on ρ_r, then there is also a bad trace on ρ_l, so that player II indeed wins the match ρ.

The *tool* that we employ to guarantee this consists of a well-founded ordering of the collection of those ρ_l-traces that arrive to the $\mathbb{D}$-part of the automaton $\mathbb{B}[\mathbb{D}/p]$. The definition of this ordering crucially uses the disjunctivity of $\mathbb{D}$.

We define strategy χ by a simultaneous induction on the length of a partial χ-match $\rho = (R_0, R'_0), \ldots, (R_n, R'_n)$. Using a well-founded order on the set of traces, and as we already mentioned, we define a map $g_n : \mathsf{ran}_A R'_n \to \mathsf{ran}_D R_n$ and a map $F_n : (B \cup D)^\sharp \to (B \cup A)^\sharp$. We let the F-maps determine player II's strategy in the following sense. Suppose that in the mentioned partial match ρ, player I legitimately picks an element $(\Upsilon, \alpha) \in \mathsf{T}(B \cup D)^\sharp$. Then player II's response will be the map $F_n \restriction_{Base(\alpha)}$.

Inductively we maintain the following conditions.

($*$) $F_{n-1}R_n = R'_n$, for all $n > 0$;
(†1) $R'_n \cap (B \times B) \subseteq R_n$;
(†2) $R'_n \cap (B \times A) \subseteq \bigcup_{d\in D}\{(b, a) \mid (b, d) \in R_n \text{ and } a \in \mathsf{ran}G(d)\}$;
(†3) $R'_n \cap (A \times A) \subseteq \bigcup\{G(d) \mid d \in \mathsf{ran}R_n \cap D\}$;
(††) $a \in \mathsf{ran}G(g_n(a))$ for all $a \in \mathsf{ran}R'_n \cap A$.

These conditions enable us to keep track of the shape of $\mathbb{B}[\mathbb{D}/p]$- and $\mathbb{B}[\mathbb{A}/p]$-traces.

Applying the tool that we developed for *trace management* we can extend the match and define g_{n+1} and thus F_{n+1} in such a way that conditions ($*$)-(††) remain true for one more round. We finish the proof by showing that the following claims hold:

▶ **Claim (1).** The moves for player II prescribed by the strategy χ are legitimate.

Proof of this claim is straightforward according to properties of one-step coalgebraic logic.

▶ **Claim** (2). Suppose ρ is an infinite χ-guided match with basic positions

$$(R_0, R_0')(R_1, R_1')(R_2, R_2') \ldots .$$

If there is a bad trace on $R_0'R_1'R_2'$..., there is also a bad trace on $R_0R_1R_2$....

Proof is based on the actual definition of maps g_n and trace reflection property of map G. ◀

The final lemma of this section summarizes some of the closure properties of (semi)-disjunctive automata.

▶ **Lemma 5.11.** *Let $\mathbb{A}$ and $\mathbb{B}$ be modal automata. Then we have:*
(1) if $\mathbb{A}$ is disjunctive, then it is also semi-disjunctive;
(2) if $\mathbb{A}$ and $\mathbb{B}$ are disjunctive, then so is $\mathbb{A} \vee \mathbb{B}$;
(3) if $\mathbb{A}$ and $\mathbb{B}$ are semi-disjunctive, then so are $\mathbb{A} \vee \mathbb{B}$ and $\mathbb{A} \wedge \mathbb{B}$;
(4) if $\mathbb{A}$ is semi-disjunctive and $\mathbb{B}$ is disjunctive, then $\mathbb{A}[\mathbb{B}/x]$ is semi-disjunctive;
(5) $\nu x.\mathbb{A}$ and $\mathbb{A}^x$ are semi-disjunctive in case $\mathbb{A}$ is disjunctive.

For the clauses (3)-(5) of this lemma we need to involve the modal distribution laws in order to make sure that all constructed automata have the right syntactic shape.

6 Completeness

In this section we give an overview of the completeness proof for $\mu\mathsf{ML}_\mathsf{T}$ with respect to the derivation system $\mathbf{K}$. In [7] Kozen proved the completeness of his proof system for a fragment of the modal μ-calculus: he showed that for *aconjunctive* formulas consistency implies satisfiability. The following lemma can be seen as an automata-theoretic version of Walukiewicz' rendering of this result.

▶ **Lemma 6.1.** *Given an automaton $\mathbb{A}$, if $\mathsf{tr}(\mathbb{A})$ is consistent, then $\exists$ has a winning strategy in the thin satisfiability game for $\mathbb{A}$.*

We now turn to the proof of our main technical result, viz., Theorem 1.1, as we discussed in the introduction. To prove this theorem, recall from Proposition 4.13 that every formula $\varphi \in \mu\mathsf{ML}_\mathsf{T}$ is provably equivalent to the translation of the modal automaton $\mathbb{A}_\varphi$. Thus in particular we have $\vdash_\mathbf{K} \varphi \to \mathsf{tr}(\mathbb{A}_\varphi)$. We now want to apply the automata-theoretic machinery that we developed in previous sections, to strengthen this result, showing that for any formula φ there is a *disjunctive* automaton $\mathbb{D}_\varphi$ such that $\vdash_\mathbf{K} \varphi \to \mathsf{tr}(\mathbb{D}_\varphi)$. The following lemma shows that whenever φ is the translation of a semi-disjunctive automaton this result can be proved.

▶ **Lemma 6.2.** *Let $\mathbb{A}$ be any semi-disjunctive automaton. Then $\vdash_\mathbf{K} \mathsf{tr}(\mathbb{A}) \to \mathsf{tr}(\mathsf{sim}(\mathbb{A}))$.*

Proof. By Proposition 5.8 there is a winning strategy for player II in the consequence game $\mathcal{C}(\mathbb{A}, \mathsf{sim}(\mathbb{A}))$. Since $\mathbb{A}$ is semi-disjunctive, it follows from Proposition 5.9 that there is a winning strategy for $\forall$ in the thin satisfiability game for $\mathbb{A} \wedge \neg\mathsf{sim}(\mathbb{A})$. We get $\vdash_\mathbf{K} \neg\mathsf{tr}(\mathbb{A} \wedge \neg\mathsf{sim}(\mathbb{A}))$ by lemma 6.1. Then from clauses (1) of Lemma 4.18 we have $\vdash_\mathbf{K} \neg(\mathsf{tr}(\mathbb{A}) \wedge \neg\mathsf{tr}(\mathsf{sim}(\mathbb{A})))$, which means that $\vdash_\mathbf{K} \mathsf{tr}(\mathbb{A}) \to \mathsf{tr}(\mathsf{sim}(\mathbb{A}))$ as required. ◀

Proof of Theorem 1.1. The proof is given by induction on the complexity of formula φ. We assume without loss of generality that φ is in negation normal form and inductively omit the cases of literals, disjunctions and the modal operator. For conjunctions, given formulas α, α' we have disjunctive automata $\mathbb{D} \equiv \alpha$ and $\mathbb{D}' \equiv \alpha'$ such that $\vdash_{\mathbf{K}} \alpha \to \mathsf{tr}(\mathbb{D})$ and $\vdash_{\mathbf{K}} \alpha' \to \mathsf{tr}(\mathbb{D}')$. By the first clause of Lemma 4.18 we get $\vdash_{\mathbf{K}} \alpha \wedge \alpha' \to \mathsf{tr}(\mathbb{D} \wedge \mathbb{D}')$. But $\mathbb{D} \wedge \mathbb{D}'$ is semi-disjunctive by the third clause of Lemma 5.11, and we can apply Lemma 6.2 to obtain the desired conclusion.

Finally we turn to the fixpoint operators. Given a formula $\varphi = \nu x.\alpha(x)$ and let x be a free variable of α. Inductively there is a disjunctive automaton $\mathbb{A}$ for α such that $\vdash_{\mathbf{K}} \alpha(x) \to \mathsf{tr}(\mathbb{A})$. Since $\mathbb{A}$ is disjunctive, by the last clause of Lemma 5.11 $\nu x.\mathbb{A}$ is semi-disjunctive, so it suffices to show that $\vdash_{\mathbf{K}} \nu x.\alpha(x) \to \mathsf{tr}(\nu x.\mathbb{A})$. By clause (3) of Lemma 4.18, it suffices to prove that $\vdash_{\mathbf{K}} \nu x.\alpha(x) \to \nu x.\mathsf{tr}(\mathbb{A})$, and this clearly follows from our assumption that $\vdash_{\mathbf{K}} \alpha(x) \to \mathsf{tr}(\mathbb{A})$.

Now we consider the crucial case where $\varphi = \mu x.\alpha(x)$. By the induction hypothesis there is a semantically equivalent disjunctive automaton $\mathbb{A}$ for $\alpha(x)$ such that $\vdash_{\mathbf{K}} \alpha(x) \to \mathsf{tr}(\mathbb{A})$. Let $\mathbb{D} := \mathsf{sim}(\mu x.\mathbb{A})$. This automaton is clearly semantically equivalent to φ. We want to show that:

$$\vdash_{\mathbf{K}} \mu x.\mathsf{tr}(\mathbb{A}) \to \mathsf{tr}(\mathbb{D}),$$

from which the result follows since $\vdash_{\mathbf{K}} \varphi \to \mu x.\mathsf{tr}(\mathbb{A})$. By clause (1) of Lemma 4.17 together with clause (3) of Lemma 4.18 we get $\mu x.\mathsf{tr}(\mathbb{A}) \equiv_{\mathbf{K}} \mu x.\mathsf{tr}(\mathbb{A}^x)$, so it in fact suffices to prove:

$$\vdash_{\mathbf{K}} \mu x.\mathsf{tr}(\mathbb{A}^x) \to \mathsf{tr}(\mathbb{D}).$$

Hence by the fixpoint rule it suffices to prove that:

$$\vdash_{\mathbf{K}} \mathsf{tr}(\mathbb{A}^x)[\mathsf{tr}(\mathbb{D})/x] \to \mathsf{tr}(\mathbb{D}).$$

But using clause (4) of Lemma 4.18 we get

$$\mathsf{tr}(\mathbb{A}^x)[\mathsf{tr}(\mathbb{D})/x] \equiv_{\mathbf{K}} \mathsf{tr}(\mathbb{A}^x[\mathbb{D}/x]),$$

so it suffices to prove $\vdash_{\mathbf{K}} \mathsf{tr}(\mathbb{A}^x[\mathbb{D}/x]) \to \mathsf{tr}(\mathbb{D})$, or equivalently:

$$\vdash_{\mathbf{K}} \neg(\mathsf{tr}(\mathbb{A}^x[\mathbb{D}/x]) \wedge \neg\mathsf{tr}(\mathbb{D})).$$

We can now apply the clauses (2) and (3) of Lemma 4.18 to see that this is equivalent to $\vdash_{\mathbf{K}} \neg\mathsf{tr}(\mathbb{A}^x[\mathbb{D}/x] \wedge \neg\mathbb{D})$, and by Lemma 6.1 it therefore suffices to prove that $\forall$ has a winning strategy in the thin satisfiability game for the automaton $\mathbb{A}^x[\mathbb{D}/x] \wedge \neg\mathbb{D}$. Note that by clause (5) of Lemma 5.11 $\mathbb{A}^x$ is semi-disjunctive since $\mathbb{A}$ is disjunctive. Now since $\mathbb{D}$ is disjunctive and $\mathbb{A}^x$ is semi-disjunctive, from Lemma 5.11 clause (4) it follows that $\mathbb{A}^x[\mathbb{D}/x]$ is semi-disjunctive too. Hence, by Theorem 5.9 the required conclusion follows if we can show that $\mathbb{A}^x[\mathbb{D}/x] \vDash_{\mathcal{C}} \mathbb{D}$. But from Lemma 4.17 and Proposition 5.8 we get by transitivity of game consequance:

$$\mathbb{A}^x[\mu x.\mathbb{A}/x] \vDash_{\mathcal{C}} \mu x.\mathbb{A} \vDash_{\mathcal{C}} \mathsf{sim}(\mu x.\mathbb{A}) = \mathbb{D}.$$

so it suffices to show that

$$\mathbb{A}^x[\mathbb{D}/x] \vDash_{\mathcal{C}} \mathbb{A}^x[\mu x.\mathbb{A}/x].$$

But this is an instance of Theorem 5.10, and so we are done. ◀

Finally we see how Theorem 1.1 implies completeness.

▶ **Theorem 6.3 (Completeness).** *Every consistent formula $\varphi \in \mu\mathtt{ML}_{\mathsf{T}}$ is satisfiable.*

Proof. Given a consistent formula φ, by Theorem 1.1 there exists a semantically equivalent disjunctive automaton $\mathbb{D}$ such that $\mathsf{tr}(\mathbb{D})$ is consistent too. Now by Lemma 6.1 $\exists$ has a winning strategy in $\mathcal{S}_{thin}(\mathbb{D})$. But $\mathbb{D}$ is disjunctive and hence semi-disjunctive, and so by Proposition 5.6 $\exists$ also has a winning strategy in $\mathcal{S}(\mathbb{D})$. ◀

References

1 J. Richard Büchi. Weak second-order arithmetic and finite automata. *Mathematical Logic Quarterly*, 6(1-6):66–92, 1960.

2 Giovanna D'Agostino and Marco Hollenberg. Logical questions concerning the μ-calculus: Interpolation, lyndon and łoś-tarski. *Journal of Symbolic Logic*, 65(1):310–332, 2000.

3 Gäelle Fontaine, Raul Leal, and Yde Venema. Automata for coalgebras: An approach using predicate liftings. In *Automata, Languages and Programming: 37th International Colloquium ICALP'10*, pages 381–392, 2010.

4 David Janin and Igor Walukiewicz. Automata for the modal μ-calculus and related results. In *20th Symposium on Mathematical Foundations of Computer Science MFCS'95*, volume 969, pages 552–562, 1995.

5 David Janin and Igor Walukiewicz. On the expressive completeness of the propositional mu-calculus with respect to monadic second order logic. In *7th International Conference on Concurrency Theory CONCUR '96*, pages 263–277, 1996.

6 Christian Kissig and Yde Venema. Complementation of coalgebra automata. In *3rd Conference on Algebra and Coalgebra in Computer Science CALCO'09*, pages 81–96, 2009.

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

8 Clemens Kupke, Alexander Kurz, and Yde Venema. Completeness of the finitary Moss logic. In *9th Conference on Advances in Modal Logic AiML'08*, pages 193–217, 2008.

9 Clemens Kupke, Alexander Kurz, and Yde Venema. Completeness for the coalgebraic cover modality. *Logical Methods in Computer Science*, 8:1–76, 2012.

10 Clemens Kupke and Yde Venema. Coalgebraic automata theory: Basic results. *Logical Methods in Computer Science*, 4(4), 2008.

11 Johannes Marti, Fatemeh Seifan, and Yde Venema. Uniform interpolation for coalgebraic fixpoint logic. In *6th Conference on Algebra and Coalgebra in Computer Science CALCO'15*, pages 238–252, 2015.

12 Larry Moss. Coalgebraic logic. *Annals of Pure and Applied Logic*, 96:277–317, 1999. (Erratum published *Ann.P.Appl.Log.* 99:241–259, 1999).

13 David E. Muller and Paul E. Schupp. Simulating alternating tree automata by nondeterministic automata: New results and new proofs of the theorems of Rabin, McNaughton and Safra. *Theoretical Computer Science*, 141(1-2):69–107, 1995.

14 Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 141:1–35, 1969.

15 J.J.M.M. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science*, 249(1):3–80, 2000.

16 Luigi Santocanale. Free μ-lattices. *Journal of Pure and Applied Algebra*, 168(2-3):227–264, 2002.

17 Yde Venema. Automata and fixed point logic: a coalgebraic perspective. *Information and Computation*, 204:637–678, 2006.

18 Igor Walukiewicz. Completeness of Kozen's axiomatisation of the propositional μ-calculus. *Information and Computation*, 157:142–182, 2000.

19 Thomas Wilke. Alternating tree automata, parity games, and modal μ-calculus. *Bulletin of the Belgian Mathematical Society - Simon Stevin*, 8(2):359–391, 2001.