# Lattice-Theoretic Progress Measures and Coalgebraic Model Checking

Ichiro Hasuo Shunsuke Shimizu
University of Tokyo, Japan
{ichiro,shunsuke}@is.s.u-tokyo.ac.jp

Corina Cîrstea
University of Southampton, UK
cc2@ecs.soton.ac.uk

## Abstract

In the context of formal verification in general and model checking in particular, *parity games* serve as a mighty vehicle: many problems are encoded as parity games, which are then solved by the seminal algorithm by Jurdzinski. In this paper we identify the essence of this workflow to be the notion of *progress measure*, and formalize it in general, possibly infinitary, lattice-theoretic terms. Our view on progress measures is that they are to nested/alternating fixed points what *invariants* are to safety/greatest fixed points, and what *ranking functions* are to liveness/least fixed points. That is, progress measures are combination of the latter two notions (invariant and ranking function) that have been extensively studied in the context of (program) verification.

We then apply our theory of progress measures to a general model-checking framework, where systems are categorically presented as coalgebras. The framework's theoretical robustness is witnessed by a smooth transfer from the branching-time setting to the linear-time one. Although the framework can be used to derive some decision procedures for finite settings, we also expect the proposed framework to form a basis for sound proof methods for some undecidable/infinitary problems.

***Categories and Subject Descriptors*** D.2.4 [*Software/Program Verification*]: Model checking; F.4.1 [*Mathematical Logic*]: Modal logic

***Keywords*** fixed-point logic, model checking, coalgebra

## 1. Introduction

### 1.1 Backgrounds

***Parity Games and Fixed-Point Logics*** For the purpose of formal verification where one aims at establishing that a system satisfies a certain property (called a *specification*), it is common to express: a model of the system as a state-based transition system such as an automaton or a Kripke structure; and a specification as a formula in some modal logic. For the latter, in particular, logics with *fixed-point operators*—such as LTL and CTL—serve well thanks to their remarkable expressivity [49]. The *modal $\mu$-calculus* (see e.g. [7, 38]) provides a clean syntax that incorporates the least and greatest fixed-point operators ($\mu$ and $\nu$) in a systematic manner.

Dealing with such fixed points is however a nontrivial task—this is especially the case when $\mu$'s and $\nu$'s are nested and they alternate. Many engineers find it challenging to express their intuition as a fixed-point formula; furthermore, many algorithms are first introduced for an alternation-free fragment and then later extended to the full fragment (see e.g. [18] and [19]).

For the purpose of analyses of fixed-point logics and designing algorithms for them, *parity games* have emerged as a very useful tool in the last decade or so. A parity game is played by two players $\mathsf{even}$ and $\mathsf{odd}$, on a board each position $x$ of which has a natural number $\mathsf{pr}(x) \in \omega$ called its *priority*. Notably its winning condition is the *parity condition*: the player $\mathsf{even}$ wins if the largest priority that occurs infinitely often in a given play (an infinite sequence of positions) is an even number. This condition—that may seem ad-hoc at first sight—turns out to be extremely useful for modeling nested and alternating $\mu$'s and $\nu$'s. It is in a sense a *combinatorial* presentation of an alternation between $\mu$'s and $\nu$'s.

The use of parity games has been boosted further by Jurdzinski's algorithm that efficiently determines the winner at each position of a parity game [33]. It exhibits a practical complexity that is exponential only in so-called the *alternation depth* of a parity game. It has then become a norm, in the context of fixed-point logics and algorithmic formal verification, to take the following *parity-game workflow*: it reduces a problem in question to the decision problem of some parity game, and then solves the latter by Jurdzinski's algorithm. A notable example is the model-checking problem for the modal $\mu$-calculus (see e.g. [59]).

The key ingredient of Jurdzinski's algorithm is what is called a *progress measure* (a notion originally from [35])—it can be understood as an extension of a *ranking function* (used e.g. for termination proofs) to a setting with nested $\mu$'s and $\nu$'s.

***Coalgebras and Coalgebraic Modal Logics*** On the other side of formal verification (namely system models), *coalgebra* has attracted attention as a categorical abstraction of state-based systems [31, 51] for more than a decade. An *$F$-coalgebra* is an arrow $c\colon X \to FX$ in some category $\mathbb{C}$, where $F\colon \mathbb{C} \to \mathbb{C}$ is an endofunctor. By changing $\mathbb{C}$ and $F$ a coalgebra instantiates to a variety of transition systems, such as Kripke structures, LTSs, Markov chains, tree automata, processes in the $\pi$-calculus, and so on. Abstracting away from specific choices of $\mathbb{C}$ and $F$ allows us to develop a uniform theory that applies to various systems. One notable success is a uniform definition of *bisimulation* that is independent from $\mathbb{C}$ and $F$. See [31, 51].

Along with the development of the theory of coalgebras, *coalgebraic modal logics* have been developed as languages suited for specifying about coalgebras (see e.g. [17]). Besides the approaches with Moss' *cover modality* [41] and *Stone-like dualities* [6], the one



*POPL'16*, January 20–22, 2016, St. Petersburg, FL, USA
ACM. 978-1-4503-3549-2/16/01...$15.00
http://dx.doi.org/10.1145/2837614.2837673

| properties | witnessed by |
|---|---|
| safety, gfp | invariants |
| liveness, lfp | ranking functions |
| *nested gfp's and lfp's* | winning strategies for parity games (if finitary), and *progress measures* (in general) |

**Table 1.** Progress measures = (invariants + ranking functions)

with *predicate liftings* [45] is widely adopted in the literature. The theory has since produced many uniform results about coalgebraic modal logics as specification languages. They are on: expressivity (i.e. that bisimilarity is captured) [36, 45]; sound and complete axiomatizations [45, 52]; satisfiability complexity [52]; cut elimination and interpolation [46, 47]; and so on.

Fixed-point operators in coalgebraic modal logics have been actively studied too. See e.g. [12, 13, 16, 29, 52, 53], and also [24, 58] where *coalgebraic automata* are studied as translations of $\mu$-calculus formulas. In particular, in [16], algorithms for the model-checking and satisfiability problems of a coalgebraic $\mu$-calculus are presented. These algorithms reduce the problems to parity games—this follows the common parity-game workflow that we already discussed. For satisfiability they also need a tableau system devised for this purpose.

### 1.2 Contributions

In this paper we scrutinize the aforementioned *parity-game workflow* of: reducing to a parity game, and solving by Jurdzinski's algorithm. We identify its essence in *progress measures*—a key notion in Jurdzinski's algorithm [33]—rather than in parity games themselves. This leads us to a lattice-theoretic *transfinite* notion of progress measure that works without any finiteness assumption, a restriction that is inevitable in the combinatorial notion of parity game. We then go on to develop a generic (and not necessarily finitary) framework for model checking, where system models and specifications also have generic presentations in the language of coalgebras and coalgebraic modal logics.

More specifically, our technical contributions are as follows.

***Lattice-Theoretic Progress Measure*** Taking an arbitrary complete lattice $L$ as a value domain (instead of a finite power $\mathbf{2}^m$ of $\mathbf{2} = \{\mathrm{tt}, \mathrm{ff}\}$), we present a lattice-theoretic characterization of solutions of recursive equations with (nested and alternating) greatest and least fixed-points. The characterization is by the notions of *prioritized ordinal* and *progress measure*—notions that are essentially generalization of what are in Jurdzinski's work [33]. Our general formalization allows one to use progress measures also in infinitary settings where we deal with infinite-state systems, quantitative verification (i.e. the set of truth values is infinite), or both.

One can also think of our progress measure as the combination of the common proof methods by: *invariants* for safety/gfp properties, and *ranking functions* for liveness/lfp properties (Table 1). These methods have been extensively studied especially in the field of *program verification*—where problems are inherently infinitary due to the `Integer` datatype—with an emphasis on automatic synthesis of invariants and ranking functions (see e.g. recent [5, 25]). Our current results therefore open the way to combining these automated synthesis techniques, and to obtaining automated proof methods for *nested lfp/gfp properties* (like the *response formula* $\mathsf{G}(p \to \mathsf{F}q)$ but much, much more). Once done its impact will be significant, since currently most automation attempts in the field focus on only safety or liveness, and not their combination.

We note that these results (in §2) are formulated solely in (rather elementary) lattice-theoretic terms, without any category theory. While their principal use in the current paper is in coalgebraic model checking, their application areas are expected to be widespread, in quantitative verification, program verification, and so on—by model checking and deductive methods alike.

***Progress Measure for Coalgebraic $\mu$-Calculus Model Checking*** We apply the notion of progress measure to model checking of a *coalgebraic modal $\mu$-calculus* $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$. Specifically, given a coalgebra $c\colon X \to FX$ (as a system model), a $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$-formula $\varphi$ (as a specification) and the domain $\Omega$ of truth values, we characterize the semantics $[\![\varphi]\!]_c\colon X \to \Omega$ of $\varphi$ over $c$ in terms of progress measures. Unlike the original definition of the semantics $[\![\varphi]\!]_c$ (that is highly nonlocal due to fixed-point operators), it can be checked locally whether given data constitute a progress measure.

The lattice-theoretic generality of our progress measure allows: a state space $X$ that is infinite; a domain $\Omega$ of truth values that is other than $\mathbf{2} = \{\mathrm{tt}, \mathrm{ff}\}$ (such as the unit interval $[0, 1]$); and so on. Furthermore, for its finitary special case, we derive a model-checking algorithm that is based on progress measures.

We expect our theoretical framework (general, possibly infinitary, in §4.1) to be a foundation on which various verification techniques—a candidate being an extension of the simulation-based method in [55]—can be formulated and proved sound.

Besides, our generic model-checking algorithm (in §4.2, as a finitary special case of the framework in §4.1) is a uniform algorithm that works for a variety of endofunctors $F$ and modalities over $F$ (normal modal logic over Kripke models, neighborhood frames, graded modal logic, coalition logic, and so on; see Example 3.3). Moreover, thanks to its concrete presentation with matrices, our algorithm should be easy to implement.

Currently it is not clear whether our algorithm in §4.2 competes with tailor-developed ones for a specific modal logic. However we believe our generic algorithm is at least worthwhile—much like a big part of the coalgebraic attempts towards abstraction and genericity, see §1—for the following reasons: 1) among the examples covered by our generic algorithm, not all enjoy tailor-developed algorithms; and 2) we believe our algorithm, though currently basic, can expose further "handles" for optimization. The latter means: in many parity game-based algorithms, the part of solving parity games is left as a blackbox; and in principle opening up a blackbox (like we do) should be good for optimization, possibly allowing for "shortcut fusion"-like optimization.

***Coalgebraic $\mu$-Calculus as a Linear-Time Logic*** In order to further demonstrate the theoretical robustness of our framework, we present an adaptation of the framework to *linear-time model checking*. In this case a system is a coalgebra $c\colon X \to \mathcal{P}FX$ (with additional nondeterministic branching represented by the powerset monad $\mathcal{P}$); and the question is whether there is an *infinitary trace* $z$ of $c$ starting from $x$ such that $z$ satisfies a $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$-formula $\varphi$.

It turns out that the combination with coalgebraic theory of traces and simulations (developed e.g. in [26, 30, 56]) allows a smooth transfer from the previous "branching-time" setting to the current linear-time one. The outcome is a uniform treatment of branching and (nondeterministic) linear-time logics—which does not seem to be achieved before despite the obvious efforts by the coalgebra community. This venture also needs a technical piece, namely the "pumping"-like result (Thm. 5.7) by Zorn's lemma.

Our technical contributions are: a progress measure-based characterization of linear-time model checking (where, again, whether given data is a valid progress measure or not can be checked locally); and a decision procedure for linear-time model checking (with the restriction that the state space $X$ is finite and the truth values are Boolean). The former solves the challenge, presented in [14], of a local characterization of linear-time semantics (called "step-wise semantics" in [14]) for coalgebraic fixed-point logics.

### 1.3 Future Work

There are a lot of further topics to study in our current venture to coalgebraic $\mu$-calculus. They include: implementation of our model-checking algorithms in §4.2 and §5.4 (the one in §4.2 should especially be easy because of the presentation by matrices); experiments, comparison with tailor-made algorithms and further optimization; satisfiability and small-model property; universal linear-time model checking (in this paper we study the existential one); synthesis; and $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$ as linear-time logic for systems with *probabilistic* branching. In particular we expect the last to be not hard, given the lattice-theoretic generality of the current results. It should also help that the coalgebraic theory of traces and simulations has been recently extended to the probabilistic setting [11, 56] (using the Giry monad over the category of measurable spaces). We can say we understand the mathematical structures therein fairly well: these studies suggest that the probabilistic setting is better-behaved than the nondeterministic setting, from a coalgebraic point of view. See [56] for further details.

Besides, our lattice-theoretic theory of nested fixed points allows progress measures (which we identify as the essence of parity games) to be applied to infinitary settings. We believe it will be useful for the following purposes. Working out these further applications is future work.

***Establishing an Alternating Fixed-Point in Theorem Proving*** In an infinitary setting (such as the state space $|X|$ and/or the truth domain $\Omega$ are infinite), the search space for our (infinitary) progress measures will be infinite, and hence is not amenable to algorithmic search. Even so, one could resort to human ingenuity to find one.

An advantage of a progress measure-based characterization of the semantics $[\![\varphi]\!]_c$ is, as we mentioned earlier, the validity of a progress measure can be checked *locally* in a straightforward manner. This is unlike the original definition of the semantics $[\![\varphi]\!]_c$ (see Def. 3.7) that involves highly nonlocal information like $V\colon X \to \Omega$. We believe this advantage will be especially useful when one works with fixed-point specifications in a *proof assistant*.

Due to the same advantage, our progress measure-based characterization might also form a basis of sound (but not necessarily complete) model-checking algorithms that rely e.g. on mathematical programming. This is much like in [55] where Kleisli simulations (whose existence is checked by linear programming and hence is PTIME) give a sound proof method for weighted language inclusion (an undecidable property).

***As a Tool in a Meta-Theory*** In *higher-order model checking* (see e.g. [37, 44, 54]), a *higher-order recursion scheme (HORS)* generates an infinite tree that is then model-checked against a modal $\mu$-formula. The generated tree is in general irrational—hence cannot be identified with a finite-state automaton. However it is shown [37, 44] that the model-checking is decidable; an algorithm operates directly with the HORS that generates the tree, but not with the tree itself. In this setting (and similar ones), we expect our infinitary progress measure to be a useful tool on the level of meta-theory, e.g. for showing the correctness of an algorithm.

We also envisage the use of our current results in lifting *(bi)simulation* notions for Büchi and parity automata (see e.g. [22]) to the coalgebraic level of abstraction and generality. In this direction we have obtained some preliminary results that characterize the accepted languages of Büchi/parity automata via coalgebras in a Kleisli category—results that will hopefully enable us to extend our coalgebraic theory of traces and simulations in [26, 30, 56] to Büchi/parity acceptance conditions. We also intend to study the relationship between our current work and *quantitative* extensions of parity games, a topic of extensive research efforts [8, 9].

### 1.4 Notations

Throughout the paper, the domain of truth values is denoted by $\Omega$ and is assumed to be a complete lattice, with its order denoted by $\sqsubseteq$, and its supremums and infimums denoted by $\bigsqcup$ and $\bigsqcap$. Typical examples of $\Omega$ are the set $\mathbf{2} = \{\mathrm{tt}, \mathrm{ff}\}$ of Boolean truth values, and the unit interval $[0, 1]$ for a quantitative notion of truth. In §2 we will use another complete lattice $L$; this will be instantiated by $L = \Omega^X$—where $X$ is the state space of the system in question—for the use in later sections. Since $\Omega$ is a complete lattice, any monotonic endofunction $f$ on $\Omega$ has the greatest and least fixed points $\nu f, \mu f$. The same holds for $L$ in place of $\Omega$.

We fix a countable set $\mathbf{Var}$ of *(fixed-point) variables*. It is ranged by $u, v, w, \ldots$. We let $\eta$ designate fixed-point operators in general; it is either $\mu$ or $\nu$. Confusion with a monad unit is unlikely.

The set of natural numbers is identified with the smallest infinite ordinal and denoted by $\omega$.

### 1.5 Organization of the Paper

In §2 we present our lattice-theoretic notion of progress measure and prove that it characterizes the solution of a system of fixed-point equations. In §3 we introduce our logic $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$—it is a coalgebraic modal logic with both greatest and least fixed-point operators $(\nu, \mu)$; it is parametrized not only by the set $\Lambda$ of predicate liftings (i.e. modalities) for a functor $F$, but also by the set $\Gamma$ of propositional connectives. In §4 we adapt progress measures in §2 to the purpose of $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$ model checking (against $F$-coalgebras), derive a model-checking algorithm and analyze its complexity. This framework is further adapted in §5 to (existential) linear-time model checking—where a system has additional nondeterministic branching. We present a decision procedure there.

Appendices to the current paper are found in the extended version [27]. Omitted proofs are there, too.

## 2. Progress Measures for Equational Systems

### 2.1 Prelude: (Unnested) Fixed Points, Invariants and Ranking Functions

In general, there are two different ways for characterizing (not nested) least/greatest fixed points (lfp's and gfp's). The first is the *Knaster-Tarski* one: the lfp is the least prefixed point; and the gfp is the greatest postfixed point. The second is the *Cousot-Cousot* one [20]: the lfp $\mu f$ of a monotone function $f\colon L \to L$ over a complete lattice $L$ is the (possibly transfinite) supremum of the chain $\bot \sqsubseteq f(\bot) \sqsubseteq f^2(\bot) \sqsubseteq \cdots$; similarly the gfp $\nu f$ is the infimum of $\top \sqsupseteq f(\top) \sqsupseteq \cdots$. Sometimes these chains are guaranteed to stabilize after $\omega$ steps, for example when $f$ satisfies suitable continuity conditions (the *Kleene* fixed-point theorem).

In this paper our principal interests will be finding *lower bounds* for fixed points; see Rem. 2.9 for system verification motivations. Among the last four characterizations (Knaster-Tarski and Cousot-Cousot, for each of lfp and gfp), what are suited for this purpose of ours are: the Cousot-Cousot one for lfp's; and the Knaster-Tarski one for gfp's (the other two only give us *upper bounds*). We explicitly note this fact for the record:

**Lemma 2.1** (lower bounds for fixed points)**.** *Let $L$ be a complete lattice and $f\colon L \to L$ be a monotone function.*

1. *For each ordinal $\alpha$ we have $f^\alpha(\bot) \sqsubseteq \mu f$. Here $f^\alpha(\bot)$ is defined by obvious induction: $f^{\alpha+1}(\bot) = f(f^\alpha(\bot))$ for a successor ordinal; and $f^\alpha(\bot) = \bigsqcup_{\beta<\alpha} f^\beta(\bot)$ for a limit ordinal.*
2. *For any $l \in L$, $l \sqsubseteq f(l)$ implies $l \sqsubseteq \nu f$.* □

We emphasize that this simple theoretical observation is what underlies the difference between the common proof methods for

*safety/gfp* properties and for *liveness/lfp* properties (Table 1). For the former (gfp's) one would seek for an *invariant*, that is, a post-fixed point $l$ such that $l \sqsubseteq f(l)$. For the latter (lfp's) one would typically synthesize a *ranking function*, an $\omega$-valued function that strictly decreases in each step. We formulate—also for the sake of some intuitions—the general principle behind the latter, focusing on $L = \mathbf{2}^X$.

**Definition 2.2.** Let $f\colon \mathbf{2}^X \to \mathbf{2}^X$ be a monotone function. A *ranking function* for $f$ is an ordinal- (or ♠, indicating "failure") valued function $\mathrm{rk}\colon X \to \mathrm{Ord} \amalg \{♠\}$ such that: 1) $\mathrm{rk}(x) \neq 0$ for each $x \in X$; 2) for each ordinal $\alpha$, $\{x \mid \mathrm{rk}(x) \leq \alpha + 1\} \subseteq f\big(\{x \mid \mathrm{rk}(x) \leq \alpha\}\big)$; and 3) for each limit ordinal $\alpha$, $\{x \mid \mathrm{rk}(x) \leq \alpha\} = \bigcup_{\beta<\alpha}\{x \mid \mathrm{rk}(x) \leq \beta\}$.

**Example 2.3.** Assume that $X$ is equipped with a transition relation $R \subseteq X \times X$ and we are interested in reachability to a subset $U \subseteq X$. We would then define $f$ by: $f(X') := U \cup \{x \mid \exists x'.\, xRx' \wedge x' \in X'\}$; this yields $f^\alpha(\bot)$ to be the set of states from which $U$ is reachable within $\alpha - 1$ steps. A prototypical ranking function is given by $\mathrm{rk}(x) :=$ (the distance from $x$ to $U$) $+ 1$.

**Lemma 2.4.** *In Def. 2.2, a ranking function* rk *for $f$ witnesses $\mu f$, the least fixed point of $f$. That is,* $\mathrm{rk}(x) \neq ♠$ *implies* $x \in \mu f$.

*Proof.* The following is easily shown by induction on an ordinal $\alpha$: for any $x \in X$ such that $\mathrm{rk}(x) = \alpha$, we have $x \in f^\alpha(\bot)$. The claim then follows from Lem. 2.1.1. □

**Remark 2.5.** Implicit in the above is a bijective correspondence—not unlike in *Stone-like dualities*—between:

- a ranking function $\mathrm{rk}\colon X \to \mathrm{Ord} \amalg \{♠\}$; and
- an *approximating sequence* $U_0 \subseteq U_1 \subseteq \cdots$ such that: 1) $U_0 = \bot = \emptyset$, 2) $U_{\alpha+1} \subseteq f(U_\alpha)$, and 3) $U_\alpha = \bigcup_{\beta<\alpha} U_\beta$ for any limit ordinal $\alpha$.

From the former to the latter we let $U_\alpha := \{x \mid \mathrm{rk}(x) \leq \alpha\}$; conversely we let $\mathrm{rk}(x) := \inf\{\alpha \mid x \in U_\alpha\}$.

### 2.2 Equational Systems

With the preparations in §2.1 for unnested fixed points, we set out to study nested and alternating ones. As a formalism of expressing them we prefer *equational systems*, to the (probably more common) modal $\mu$-calculus-like notations. Here we shall follow the accounts of similar notions in [19] and [2, §1.4].

**Definition 2.6** (equational system)**.** Let $L$ be a complete lattice. An *equational system* $E$ over $L$ is an expression of the form

$$u_1 =_{\eta_1} f_1(u_1, \ldots, u_m), \quad \ldots, \quad u_m =_{\eta_m} f_m(u_1, \ldots, u_m) \tag{1}$$

where: $u_1, \ldots, u_m$ are *variables*, $\eta_1, \ldots, \eta_m \in \{\mu, \nu\}$, and $f_1, \ldots, f_m\colon L^m \to L$ are monotone functions.

A variable $u_j$ is said to be a *$\mu$-variable* if $\eta_j = \mu$; it is a *$\nu$-variable* if $\eta_j = \nu$.

We say $u_i$ has a *bigger priority* than $u_j$ if $j < i$.

Note that, in the last definition, we have been vague about the distinction between a function $f_i$ as a semantical object and a syntactic symbol that denotes it.

It is straightforward to generalize the definition and allow different variables to take values in different complete lattices $L_1, \ldots, L_m$, and extend accordingly our technical developments below. We assume $L_1 = \cdots = L_m = L$ for ease of presentation.

The order of equations *matters* in an equational system like (1).[1] Intuitively, the system (1) is solved starting from the leftmost equation, where the remaining variables $u_2, \ldots, u_m$ are left as undetermined parameters. The *interim* solution of the leftmost equation (for $u_1$, in terms of $u_2, \ldots, u_m$) is then used in the second equation $u_2 =_{\eta_2} f_2(u_1, \ldots, u_m)$ to eliminate the occurrences of $u_1$ in its right-hand side. We continue this way; then solving the last (rightmost) equation would give us a *closed* (i.e. without any variables occurring in it) solution for $u_m$. Such closed solutions are then propagated from right to left in (1), finally giving a closed solution to each variable $u_i$.

The above intuitions can be put in the following precise terms.

**Definition 2.7** (solution)**.** The *solution* of an equational system (1) is defined as follows. For each $i \in [1, m]$ and $j \in [1, i]$, we define monotone functions

$$f_i^\ddagger\colon L^{m-i+1} \to L \quad \text{and} \quad l_j^{(i)}\colon L^{m-i} \to L$$

as follows, inductively on $i$. For the base case $i = 1$:

$$f_1^\ddagger(l_1, \ldots, l_m) := f_1(l_1, \ldots, l_m),$$
$$l_1^{(1)}(l_2, \ldots, l_m) := \eta_1\big[f_1^\ddagger(\_, l_2, \ldots, l_m)\colon L \to L\big].$$

In the last line we take the lfp or gfp (according to $\eta_1 \in \{\mu, \nu\}$) of the (monotone) function $f_1^\ddagger(\_, l_2, \ldots, l_m)\colon L \to L$.

For the step case, the function $f_{i+1}^\ddagger$ makes use of the $i$-th interim solutions $l_1^{(i)}, \ldots, l_i^{(i)}$ for the variables $u_1, \ldots, u_i$ obtained so far:

$$f_{i+1}^\ddagger(l_{i+1}, \ldots, l_m) :=$$
$$f_{i+1}\big(\, l_1^{(i)}(l_{i+1}, \ldots, l_m),\ \ldots,\ l_i^{(i)}(l_{i+1}, \ldots, l_m),\ l_{i+1}, \ldots, l_m\,\big).$$

We then let

$$l_{i+1}^{(i+1)}(l_{i+2}, \ldots, l_m) := \eta_{i+1}\big[\, f_{i+1}^\ddagger(\_, l_{i+2}, \ldots, l_m)\colon L \to L\,\big]$$

and use it to obtain the $(i+1)$-th interim solutions $l_1^{(i+1)}, \ldots, l_i^{(i+1)}$. That is, for each $j \in [1, i]$,

$$l_j^{(i+1)}(l_{i+2}, \ldots, l_m) := l_j^{(i)}\big(\, l_{i+1}^{(i+1)}(l_{i+2}, \ldots, l_m),\ l_{i+2}, \ldots, l_m\,\big) \tag{2}$$

Finally, the *solution* $(l_1^{\mathrm{sol}}, \ldots, l_m^{\mathrm{sol}}) \in L^m$ of the equational system (1) is defined by $(l_1^{\mathrm{sol}}, \ldots, l_m^{\mathrm{sol}}) := (l_1^{(m)}, \ldots, l_m^{(m)})$, where we identify a function $l_j^{(m)}\colon 1 \to L$ with an element of $L$.

It is easy to see that all the functions $f_i^\ddagger$ and $l_j^{(i)}$ involved here are monotone. That the solution uniquely exists is then guaranteed by the Knaster-Tarski theorem.

**Example 2.8.** As a simple example, consider an equational system $u_1 =_\mu u_2$, $u_2 =_\nu u_1$. Solving the first equation yields $u_1 = u_2$ (i.e. $l_1^{(1)}(l_2) = l_2$); using it to eliminate $u_1$ in the second equation, we obtain $u_2 =_\nu u_2$ (i.e. $f_2^\ddagger(l_2) = l_2$). We conclude $u_1 = u_2 = \top$ is the solution.

It is not hard to see that, if we change the order of the equations, the resulting system $u_2 =_\nu u_1$, $u_1 =_\mu u_2$ has a different solution $u_1 = u_2 = \bot$.

It is not hard to give a precise correspondence between equational systems and their modal $\mu$-calculus-like presentations. Each equation $u_j =_{\eta_j} f_j(u_1, \ldots, u_m)$ corresponds to a fixed-point formula $\eta_j u_j.\, f_j(u_1, \ldots, u_m)$; since an equational system like (1) is solved from left to right, the formula that corresponds to an equation on the left occurs *inside* the formula for an equation on the right. For example, if $m = 2$, the equational system (1) is presented as $\eta_2 u_2.\, f_2\big(\,(\eta_1 u_1.\, f_1(u_1, u_2))\,,\ u_2\,\big)$. In the light of such a correspondence to $\mu$-calculus-like formulas, the definition of bigger/smaller priorities in Def. 2.6 coincides with what is customary (an outside fixed-point operator has a bigger priority). A precise translation can be defined following [19]; see also Def. 3.5 later, in the special case of coalgebraic fixed-point logic.

[1] Here we follow the ordering convention in [2]. In [19] the order is reversed, and the rightmost equation is solved first.

**Remark 2.9** (aiming at lower bounds)**.** Assume that an equational system $E$ is given. For the purpose of system *verification*, one is typically not so much interested in its solution itself, as in a suitable *lower bound* of it. For a simple example consider the setting of Example 2.3, and assume that $X$, $R$ and $U$ are given as follows.


$U := \{x_0\}$.

A common question would be if $U$ is reachable from a specific state of our interest, say $x_3$. To *verify* it the ranking function

$$\mathrm{rk}(x_0) = 1, \quad \mathrm{rk}(x_i) = i + 1 \text{ for each } i \geq 1,$$
$$\mathrm{rk}(x_i) = \spadesuit \text{ for each } i < 0$$

suffices. This choice of a ranking function—while it gives a lower bound $\{x_0, x_1, \dots\} \subseteq \mu f$ of $\mu f$—does not witness e.g. $x_{-3} \in \mu f$ (that actually holds). This is not a problem because we are interested only in $x_3$.

This phenomenon (of only giving a lower bound) is the case with verification algorithms in general: they conduct "directed" searches from the states in question. Therefore in this paper we focus on characterizing *lower bounds* of the solution of an equational system. *Upper bounds*, in contrast, are useful in *refuting* that certain states have certain properties.

### 2.3 Progress Measures

We shall now characterize lower bounds of (nested and alternating) fixed points specified by an equational system. We use the technical notion of *progress measure*; it is a lattice-theoretic generalization of the notion of *parity progress measure* in [33], and hence is seen as a generalization of *winning strategies* for parity games, too. Roughly speaking, these are how one combines *invariants* (for gfp's) and *ranking functions* (for lfp's, see Table 1 and §2.1) in an intricate way so that priorities in alternation are respected.

Following Lem. 2.1 we approximate least fixed points by transfinite sequences starting from $\bot$. In general there are multiple $\mu$-variables in an equational system—we have one "counter" for each of them, and use their tuple that we call a *prioritized ordinal*. In particular, the definition of the preorder $\preceq_i$ between prioritized ordinals—derived from the one in [33] and defined for each variable $u_i$—lies in the technical core.

**Definition 2.10** (prioritized ordinal, $\preceq_i$)**.** Let $E$ be the equational system in (1), over a complete lattice $L$. Let us collect all those indices $i \in [1, m]$ for which $u_i$ is a $\mu$-variable in the equational system $E$, and arrange them so that $i_1 < \cdots < i_k$. That is,

$$\{i_1, \dots, i_k\} \;=\; \{i \in [1, m] \mid \eta_i = \mu \text{ in (1)}\}.$$

Then a *prioritized ordinal* for $E$ is a $k$-tuple $(\alpha_1, \dots, \alpha_k)$ of ordinals. Note that $k$ is the number of $\mu$-variables in $E$.

For each $i \in [1, m]$ we define a preorder $\preceq_i$ between prioritized ordinals—we call $\preceq_i$ the *$i$-th truncated lexicographic order*—as follows. Let $a \in [1, k]$ be such that

$$i_1 < \cdots < i_{a-1} < i \leq i_a < \cdots < i_k,$$

that is, $u_{i_a}$ is the $\mu$-variable with the smallest priority that is at least as big as that of $i$. Then we define

$$(\alpha_1, \dots, \alpha_k) \preceq_i (\alpha'_1, \dots, \alpha'_k)$$

if, between the *$i$-truncations* $(\alpha_a, \dots, \alpha_k)$ and $(\alpha'_a, \dots, \alpha'_k)$ of the prioritized ordinals, we have $(\alpha_a, \dots, \alpha_k) \preceq (\alpha'_a, \dots, \alpha'_k)$. Here the last $\preceq$ denotes the lexicographic extension of the usual order $\leq$ between ordinals, with the latter elements being the more significant. Note here that the $i$-truncation $(\alpha_a, \dots, \alpha_k)$ of $(\alpha_1, \dots, \alpha_k)$ is obtained by dropping the first elements that correspond to the $\mu$-variables with priorities smaller than that of $u_i$.

In case $\preceq_i$ holds in both ways we write $=_i$. Note that $=_i$ is in general coarser than the equality between prioritized ordinals (see Example 2.11). We define $(\alpha_1, \dots, \alpha_k) \prec_i (\alpha'_1, \dots, \alpha'_k)$ if $(\alpha_a, \dots, \alpha_k) \preceq_i (\alpha'_a, \dots, \alpha'_k)$ holds but $(\alpha_a, \dots, \alpha_k) =_i (\alpha'_a, \dots, \alpha'_k)$ fails.

**Example 2.11.** Let us consider the following example $E_0$ of an equational system:

$$u_1 =_\mu f_1(\vec{u}), \quad u_2 =_\nu f_2(\vec{u}), \quad u_3 =_\mu f_3(\vec{u}),$$
$$u_4 =_\mu f_4(\vec{u}), \quad u_5 =_\nu f_5(\vec{u}),$$

where $\vec{u}$ stands for $u_1, \dots, u_5$. A prioritized ordinal for this $E_0$ is a tuple $(\alpha_1, \alpha_2, \alpha_3)$ of ordinals, where the ordinals $\alpha_1$, $\alpha_2$ and $\alpha_3$ correspond to the $\mu$-variables $u_1$, $u_3$ and $u_4$, respectively.

It holds that $(\omega, 2, 2) \preceq_1 (0, 3, 2)$. To see that, since $u_1$ is with the smallest priority, we have to check $(\omega, 2, 2) \preceq (0, 3, 2)$. This holds; recall that $\preceq$ is the lexicographic order with the latter being the more significant. We can similarly see that:

$$(\omega, 2, 2) \prec_2 (0, 3, 2), \qquad (\omega, 2, 2) \prec_3 (0, 3, 2),$$
$$(\omega, 2, 2) =_4 (0, 3, 2), \quad \text{and} \qquad (\omega, 2, 2) =_5 (0, 3, 2).$$

Note here that the 3-, 4- and 5-truncations of $(\omega, 2, 2)$ and $(0, 3, 2)$ are: $(2, 2)$ and $(3, 2)$; $(2)$ and $(2)$; and $()$ and $()$, respectively.

In the following definition, the element $p_i(\alpha_1, \dots, \alpha_k) \in L$ is understood as the "$(\alpha_1, \dots, \alpha_k)$-th approximation" of the solution for the variable $u_i$ in the equational system (1).

**Definition 2.12** (progress measure for an equational system)**.** Assume the same setting as in Def. 2.10, with $E$ being the equational system (1) and $i_1 < \cdots < i_k$ enumerating the indices of all the $\mu$-variables.

A *progress measure* $p$ for $E$ is given by a tuple

$$p \;=\; \big( (\overline{\alpha_1}, \dots, \overline{\alpha_k}), \big( p_i(\alpha_1, \dots, \alpha_k) \big)_{i, \alpha_1, \dots, \alpha_k} \big)$$

that consists of:

- the *maximum prioritized ordinal* $(\overline{\alpha_1}, \dots, \overline{\alpha_k})$; and
- the *approximants* $p_i(\alpha_1, \dots, \alpha_k) \in L$, defined for each $i \in [1, m]$ and each prioritized ordinal $(\alpha_1, \dots, \alpha_k)$ such that $\alpha_1 \leq \overline{\alpha_1}, \dots, \alpha_k \leq \overline{\alpha_k}$.

The approximants $p_i(\alpha_1, \dots, \alpha_k)$ are subject to:

1. **(Monotonicity)** Let $i \in [1, m]$ (hence $u_i$ is either a $\mu$- or $\nu$-variable). Then
$$(\alpha_1, \dots, \alpha_k) \preceq_i (\alpha'_1, \dots, \alpha'_k) \text{ implies } p_i(\alpha_1, \dots, \alpha_k) \sqsubseteq p_i(\alpha'_1, \dots, \alpha'_k).$$
2. **($\mu$-variables, base case)** Let $a \in [1, k]$. Then $\alpha_a = 0$ implies $p_{i_a}(\alpha_1, \dots, \alpha_a, \dots, \alpha_k) = \bot$. (Note the correspondence between: the subscript $i_a$ of $p_{i_a}$; and the counter $\alpha_a$ that is assumed to be 0.)
3. **($\mu$-variables, step case)** Let $a \in [1, k]$, and let $(\alpha_1, \dots, \alpha_a + 1, \dots, \alpha_k)$ be a prioritized ordinal such that its $a$-th counter $\alpha_a + 1$ is a successor ordinal. Then, regarding the approximant $p_{i_a}(\alpha_1, \dots, \alpha_{a-1}, \alpha_a + 1, \alpha_{a+1}, \dots, \alpha_k)$, there exist ordinals $\beta_1, \dots, \beta_{a-1}$ such that
$$p_{i_a}(\alpha_1, \dots, \alpha_{a-1}, \alpha_a + 1, \alpha_{a+1}, \dots, \alpha_k) \sqsubseteq f_{i_a} \begin{pmatrix} p_1(\beta_1, \dots, \beta_{a-1}, \alpha_a, \alpha_{a+1}, \dots, \alpha_k), \\ \dots, \\ p_m(\beta_1, \dots, \beta_{a-1}, \alpha_a, \alpha_{a+1}, \dots, \alpha_k) \end{pmatrix} \quad (3)$$
and $\beta_1 \leq \overline{\alpha_1}, \dots, \beta_{a-1} \leq \overline{\alpha_{a-1}}$. Recall here that $f_{i_a}$ is a function in the system (1).
Cond. (3) originates from the definition $f^{\alpha+1}(\bot) = f(f^\alpha(\bot))$ in Lem. 2.1.1; a notable difference here is that the counters with

smaller priorities (i.e. from the first to the $(a-1)$-th) can be modified arbitrarily.

4. **($\mu$-variables, limit case)** Let $a \in [1,k]$, and let $(\alpha_1,\dots,\alpha_k)$ be a prioritized ordinal such that its $a$-th counter $\alpha_a$ is a limit ordinal. Then, regarding the approximant $p_{i_a}(\alpha_1,\dots,\alpha_k)$, we have
$$p_{i_a}(\alpha_1,\dots,\alpha_a,\dots,\alpha_k) \sqsubseteq \bigsqcup_{\beta<\alpha_a} p_{i_a}(\alpha_1,\dots,\beta,\dots,\alpha_k)\ . \tag{4}$$
5. **($\nu$-variables)** Let $i \in [1,m] \setminus \{i_1,\dots,i_k\}$ (i.e. $u_i$ is a $\nu$-variable in the system (1)); let $a \in [1,k]$ such that
$$i_1 < \cdots < i_{a-1} < i < i_a < \cdots < i_k.$$
Let $(\alpha_1,\dots,\alpha_k)$ be a prioritized ordinal. Then, regarding the approximant $p_i(\alpha_1,\dots,\alpha_k)$, there exist ordinals $\beta_1,\dots,\beta_{a-1}$ such that
$$p_i(\alpha_1,\dots,\alpha_{a-1},\alpha_a,\dots,\alpha_k) \sqsubseteq f_i\begin{pmatrix} p_1(\beta_1,\dots,\beta_{a-1},\alpha_a,\dots,\alpha_k), \\ \dots, \\ p_m(\beta_1,\dots,\beta_{a-1},\alpha_a,\dots,\alpha_k)\end{pmatrix} \tag{5}$$
and $\beta_1 \le \overline{\alpha_1},\dots,\beta_{a-1} \le \overline{\alpha_{a-1}}$.
This condition is somewhat similar to Cond. 3 above: it comes from the condition $l \sqsubseteq f(l)$ in Lem. 2.1.2; and much like in Cond. 3, the counters with smaller priorities can be modified arbitrarily.

**Theorem 2.13** (correctness of progress measures)**.** *Let $E$ be the equational system in (1) over $L$, and $(l_1^{\mathrm{sol}},\dots,l_m^{\mathrm{sol}})$ be its solution (Def. 2.7).*

1. ***(Soundness)** A progress measure gives a lower bound of the solution. That is, assume $\big((\overline{\alpha_1},\dots,\overline{\alpha_k}),\ \big(p_i(\alpha_1,\dots,\alpha_k)\big)_{i,\overrightarrow{\alpha}}\big)$ is a progress measure. Then for each $i \in [1,m]$ we have*
$$p_i(\overline{\alpha_1},\dots,\overline{\alpha_k}) \sqsubseteq l_i^{\mathrm{sol}}.$$
2. ***(Completeness)** There exists a progress measure that achieves the optimal, that is, $\big((\overline{\alpha_1},\dots,\overline{\alpha_k}),\ \big(p_i(\alpha_1,\dots,\alpha_k)\big)_{i,\overrightarrow{\alpha}}\big)$ such that*
$$p_i(\overline{\alpha_1},\dots,\overline{\alpha_k}) = l_i^{\mathrm{sol}}$$
*for each $i \in [1,m]$.*
*Moreover, such an "optimal" progress measure can be chosen so that the ordinals in its maximum prioritized ordinal $(\overline{\alpha_1},\dots,\overline{\alpha_k})$ are suitably bounded, in the following sense. Let $\mathsf{ascCL}(L)$ be the ordinal defined by the supremum of the length of any (possibly transfinite) strictly ascending chain in $L$. Then $\overline{\alpha_a} \le \mathsf{ascCL}(L)$ for each $a \in [1,k]$.* □

In the item 2, the bound $\mathsf{ascCL}(L)$ is generally better than the bound by the size $|L|$ of the complete lattice $L$. For example, in case $L = \mathbf{2}^X$ (where $\mathbf{2} = \{\mathrm{tt},\mathrm{ff}\}$ and $X$ is a set), $\mathsf{ascCL}(L) = |X|$ while $|L| = 2^{|X|}$.

We will need the following relaxation in establishing a correspondence to Jurdzinski's notion of parity progress measure [33].

**Definition 2.14** (extended progress measure for equational systems)**.** Assume the setting of Def. 2.12. An *extended progress measure $p$* for $E$ is the same as a progress measure, except that Cond. 2 of Def. 2.12 is replaced by the following:

2'. Let $a \in [1,k]$. Then $\alpha_a = 0$ implies either $p_{i_a}(\alpha_1,\dots,\alpha_k) = \bot$, or there exists a prioritized ordinal $(\alpha_1',\dots,\alpha_k')$ such that $(\alpha_1',\dots,\alpha_k') \prec_{i_a} (\alpha_1,\dots,\alpha_k)$ and $p_{i_a}(\alpha_1,\dots,\alpha_k) \sqsubseteq p_{i_a}(\alpha_1',\dots,\alpha_k')$.

**Proposition 2.15.** *An extended progress measure is still sound in the sense of Thm. 2.13.1.* □

In Appendix A found in the extended version [27], as a sanity check, we present a correspondence between our notion of progress measure (Def. 2.12) and Jurdzinski's parity progress measure [33]. Jurdzinski's formalization follows that of ranking functions, while ours here is based on approximation sequences $p(0) \sqsubseteq p(1) \sqsubseteq \cdots$ in the lattice $L = \mathbf{2}^X$. The relationship between the two is much like in Rem. 2.5.

**Example 2.16.** For a simple example following the spirit of Example 2.3, let us consider a set $X$ and a transition relation $R \subseteq X \times X$, and introduce a "modal operator" $\Box\colon \mathbf{2}^X \to \mathbf{2}^X$ by $\Box(X') := \{x \in X \mid \forall x'.\, xRx' \Rightarrow x' \in X'\}$.

We now fix a subset $F \subseteq X$, and consider the following equational system over $L = \mathbf{2}^X$.
$$u_1 =_\mu (F \cap u_2) \cup \Box u_1, \qquad u_2 =_\nu u_1. \tag{6}$$
The system corresponds to the $\mu$-calculus formula $\nu u_2.\mu u_1.\,(F \cap u_2) \cup \Box u_1$, and it is not hard to see—possibly relying on the Knaster-Tarski and Cousot-Cousot characterizations, see §2.1—that the solution for $u_2$ is the set of states *any infinite path from which visits $F$ infinitely often*.

For this specific system (6), a progress measure (Def. 2.12) is given by data $\big(\overline{\alpha},\ \big(p_1(\alpha)\big)_{\alpha\le\overline{\alpha}},\ \big(p_2(\alpha)\big)_{\alpha\le\overline{\alpha}}\big)$ subject to suitable conditions. Some simplifications are possible, exploiting that in (5) (and elsewhere) counters with smaller priorities can be modified arbitrarily. We see, after this simplification, that a *progress measure* for the equational system (6) is given by
$$p_1(0) \subseteq p_1(1) \subseteq \cdots \subseteq p_1(\overline{\alpha}) \quad \text{and} \quad p_2,$$
all being subsets of $X$, such that: 1) $p_1(0) = \emptyset$; 2) $p_1(\alpha+1) \subseteq (F \cap p_2) \cup \Box p_1(\alpha)$; 3) $p_1(\alpha) = \bigcup_{\beta<\alpha} p_1(\beta)$ for a limit ordinal $\alpha$; and 4) $p_2 \subseteq p_1(\overline{\alpha})$. This "witnesses" the solution of (6), i.e. $x \in p_2$ implies that any infinite path from $x$ visits $F$ infinitely often.

## 3. Coalgebraic $\mu$-Calculus $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$

From this section on we apply the theory developed in §2 to a coalgebraic $\mu$-calculus $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$. In the current section, as a preparation, we introduce the logic $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$: its syntax, semantics, and a translation to equational systems (so that the results in §2 apply).

### 3.1 Coalgebraic Preliminaries

We start with a minimal set of coalgebraic preliminaries. For further backgrounds on coalgebras see e.g. [31, 51]; and see e.g. [3, 39] for categorical preliminaries. From now to §4 we fix the base category to be the one $\mathbf{Sets}$ of sets and functions.

Let $F$ be an endofunctor on $\mathbf{Sets}$. An *$F$-coalgebra* is a function $c\colon X \to FX$, where $X$, $F$ and $c$ are intuitively understood as a state space, a behavior type and a transition structure, respectively. Therefore an $F$-coalgebra is "a transition system of the behavior type $F$." Some examples are presented later in Example 3.3.

Given two coalgebras $c\colon X \to FX$ and $d\colon Y \to FY$ for the same functor, a *coalgebra homomorphism* from $c$ to $d$ is a function $f\colon X \to Y$ such that the above diagram (7) commutes. In many examples of $F$, the notion of homomorphism expresses a natural definition of *behavior-preserving map*. Conversely, it is common in the theory of coalgebras that the notion of *$F$-behavioral equivalence* is defined using homomorphism (namely via cospans, see [31]).
$$\begin{array}{ccc} FX & \xrightarrow{Ff} & FY \\ {\scriptstyle c}\uparrow & & \uparrow{\scriptstyle d} \\ X & \xrightarrow{f} & Y \end{array} \tag{7}$$

Furthermore, many functors $F$ allow a "classifying coalgebra"—one that contains every possible $F$-behavior without redundancy. This is categorically captured by *finality* of a coalgebra. Precisely, a coalgebra $\zeta\colon Z \to FZ$ is *final* if, for any
$$\begin{array}{ccc} FX & \xrightarrow{F(\mathsf{beh}(c))} & FZ \\ {\scriptstyle c}\uparrow & & \uparrow{\scriptstyle \mathrm{final}\,\zeta} \\ X & \xrightarrow[\mathsf{beh}(c)]{} & Z \end{array} \tag{8}$$

coalgebra $c\colon X \to FX$ there exists a unique homomorphism $\mathsf{beh}(c)\colon X \to Z$, as shown in (8). This way we understand the carrier $Z$ of a final coalgebra to be *the set of all $F$-behaviors*; and the map $\mathsf{beh}(c)$ induced by finality (as in (8)) as the *behavior map*.

### 3.2 $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$: Syntax

It is common in the study of coalgebraic modal logics that the set of modalities is parametrized. In our logic $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$, moreover, we parametrize propositional connectives too. This allows us to accommodate unconventional connectives that occur in quantitative setting, like the *truncated sum* in the Łukasiewicz $\mu$-calculus [40] and the *average* operator in e.g. [1].

**Definition 3.1** ($\Lambda$, $\Gamma$)**.** A *modal signature* $\Lambda$ over $F$ is a ranked alphabet $\Lambda = (\Lambda_n)_{n\in\omega}$. An element $\lambda \in \Lambda_n$ is an *$n$-ary modality*, and we write $|\lambda|$ for its arity $n$.

We assume that a modal signature comes with its *interpretation*. Assigned to each $\lambda \in \Lambda$ is a natural transformation $[\![\lambda]\!]$, whose components are functions

$$[\![\lambda]\!]_X \;:\; \big(\Omega^X\big)^{|\lambda|} \longrightarrow \Omega^{FX} \quad , \quad \text{natural in } X,$$

and a component $[\![\lambda]\!]_X$ must be monotone with respect to (pointwise extensions of) the order $\sqsubseteq$ of the domain $\Omega$ of truth values. Such $[\![\lambda]\!]$ is commonly called a *(monotone) predicate lifting* [28, 45].

Similarly, a *propositional signature* is a ranked alphabet $\Gamma$ where each $\gamma \in \Gamma$ is called a *propositional connective*. Unlike a modal signature, each $\gamma \in \Gamma$ is interpreted by a function $[\![\gamma]\!]\colon \Omega^{|\gamma|} \to \Omega$; we require that $[\![\gamma]\!]$ be monotone.

In what follows we will often write $\lambda$ and $\gamma$ for $[\![\lambda]\!]$ and $[\![\gamma]\!]$.

**Definition 3.2** ($\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$)**.** The language of our *coalgebraic modal logic* $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$ over $\Gamma$ and $\Lambda$ is given by the following set of formulas.

$$\varphi, \varphi_i \;::= u \mid \boxdot_\gamma(\varphi_1, \dots, \varphi_{|\gamma|}) \mid \heartsuit_\lambda(\varphi_1, \dots, \varphi_{|\lambda|}) \mid \\ \mu u.\, \varphi \mid \nu u.\, \varphi$$

Here $u \in \mathbf{Var}$ is a (fixed-point) variable. The notations $\boxdot_\gamma$ (for $\gamma \in \Gamma$) and $\heartsuit_\lambda$ (for $\lambda \in \Lambda$) are to distinguish propositional connectives (the former) from modalities (the latter).

**Example 3.3.** Examples of predicate lifting-based coalgebraic logics abound.

1. *Standard (normal) modal logic* is obtained by taking $F = \mathcal{P}(\mathsf{AP}) \times \mathcal{P}(\_)$ (with $\mathcal{P}$ the covariant powerset functor and $\mathsf{AP}$ a set of atomic propositions), $\Omega = \mathbf{2}$, $\Gamma = \{\mathsf{tt}, \mathsf{ff}, \wedge, \vee\}$ with the usual interpretations, and $\Lambda = \mathsf{AP} \cup \{\Box, \Diamond\}$ with

   $[\![p]\!]_X\colon 1 \to \mathbf{2}^{FX},\ * \mapsto \big\{\, (U, Y) \mid p \in U \,\big\}$ (where $p \in \mathsf{AP}$),
   $[\![\Box]\!]_X\colon \mathbf{2}^X \to \mathbf{2}^{FX},\ (V \subseteq X) \mapsto \big\{\, (U, Y) \,\big|\, Y \subseteq V \,\big\}$,
   $[\![\Diamond]\!]_X\colon \mathbf{2}^X \to \mathbf{2}^{FX},\ (V \subseteq X) \mapsto \big\{\, (U, Y) \,\big|\, Y \cap V \neq \emptyset \,\big\}$.

   Atomic propositions are thus identified with 0-ary modalities, as is standard in coalgebraic modal logic (see e.g. [52]).
2. *Hennessy-Milner logic* is obtained by taking $F = (\mathcal{P}(\_))^A$ (with $A$ a set of labels), $\Omega$ and $\Gamma$ as before, and $\Lambda = \{[a], \langle a\rangle\}$ with associated predicate liftings

   $[\![[a]]\!]_X : \mathbf{2}^X \to \mathbf{2}^{FX},\ Y \mapsto \{f \in \mathcal{P}(X)^A \mid f(a) \subseteq Y\}$,
   $[\![\langle a\rangle]\!]_X : \mathbf{2}^X \to \mathbf{2}^{FX},\ Y \mapsto \{f \in \mathcal{P}(X)^A \mid f(a) \cap Y \neq \emptyset\}$.
3. *Monotone neighborhood logic* [10] is obtained by taking $FX = \{Y \in \mathcal{P}(\mathcal{P}(X)) \mid Y \text{ is upward-closed}\}$, $\Omega$ and $\Gamma$ as before and $\Lambda = \{\Box\}$, with an associated predicate lifting

   $[\![\Box]\!] : \mathbf{2}^X \to \mathbf{2}^{FX},\ (U \subseteq X) \mapsto \{Y \in \mathcal{P}(\mathcal{P}(X)) \mid U \in Y\}$.
4. *Graded modal logic* [23] is obtained by taking $FX = (\omega + 1)^X$, $\Omega$ and $\Gamma$ as before, and $\Lambda$ consisting of graded modalities $\Box_k$ ("for all but $k$ successors") and $\Diamond_k$ ("for more than $k$ successors") for $k \in \omega$, with associated predicate liftings

   $[\![\Box_k]\!] : \mathbf{2}^X \to \mathbf{2}^{FX},\ Y \mapsto \{f \in FX \mid \sum_{x\notin Y} f(x) \leq k\}$,
   $[\![\Diamond_k]\!] : \mathbf{2}^X \to \mathbf{2}^{FX},\ Y \mapsto \{f \in FX \mid \sum_{x\in Y} f(x) > k\}$.
5. Our approach also covers the *coalition logic* [48], interpreted over *game frames*—these are coalgebras of the (class-valued) functor

   $$FX = \{(S_1, \dots, S_N, f) \mid \emptyset \neq S_i \in \mathbf{Sets}, f : \prod_{i\in N} S_i \to X\}$$

   with $N$ a set of agents, tuples $(S_1, \dots, S_N)$ capturing agent strategies, and functions $f : \prod_{i\in N} S_i \to X$ modeling the outcomes of strategy choices for the agents. The modalities $[C]$, with $C \subseteq N$ a *coalition*, arise from predicate liftings $[\![[C]]\!]_X : \mathbf{2}^X \to \mathbf{2}^{FX}$ given by $[\![[C]]\!]_X(Y) = \{(S_1, \dots, S_N, f) \in FX \mid \exists \sigma_C \in S_C.\forall \sigma_{\overline{C}} \in S_{\overline{C}}.f(\sigma_C, \sigma_{\overline{C}}) \in Y\}$, where $S_C = \prod_{i\in C} S_i$, $\overline{C} = N \setminus C$, and $(\sigma_C, \sigma_{\overline{C}})$ is defined as expected.
6. Here is an example that would yield the usual "linear-time logic" like LTL (i.e. formulas are interpreted over infinite words), in the setting of §5. Take $F = \mathcal{P}(\mathsf{AP}) \times (\_)$, $\Omega = \mathbf{2}$, $\Gamma = \{\mathsf{tt}, \mathsf{ff}, \wedge, \vee\}$ and $\Lambda = \mathsf{AP} \cup \{\mathsf{X}\}$, with the predicate liftings

   $[\![p]\!]_X\colon 1 \to \mathbf{2}^{FX},\ \ * \mapsto \big\{\, (U, x) \in \mathcal{P}(\mathsf{AP}) \times X \,\big|\, p \in U \,\big\}$,
   $[\![\mathsf{X}]\!]_X\colon \mathbf{2}^X \to \mathbf{2}^{FX},\ (V \subseteq X) \mapsto \big\{\, (U, x) \,\big|\, x \in V \,\big\}$.

In addition to the above $\mathbf{2}$-valued logics—that are also accounted for by other coalgebraic approaches to modal logic (see e.g. [16])—our approach additionally covers many-valued logics. For example, the Łukasiewicz logic of [40] can be recovered by taking $F = \mathcal{PD}$, where $\mathcal{D} : \mathbf{Sets} \to \mathbf{Sets}$ is the *probability distribution functor* defined by $\mathcal{D}X = \{\mu : X \to [0, 1] \mid \sum_x \mu(x) = 1\}$, $\Omega = [0, 1]$, $\Gamma = \{\sqcup, \oplus\}$ and $\Lambda = \{\Diamond\}$, with interpretations $[\![\sqcup]\!], [\![\oplus]\!] : [0, 1] \times [0, 1] \to [0, 1]$ given by

$$[\![\sqcup]\!]_X(p, q) = \max(p, q), \qquad [\![\oplus]\!]_X(p, q) = \min(1, p + q)$$

and predicate lifting $[\![\Diamond]\!]_X : [0, 1]^X \to [0, 1]^{FX}$ given by

$$[\![\Diamond]\!]_X(p)(f) = \max_{\mu\in f} \sum_x \mu(x)p(x).$$

Another many-valued logic that is covered by our framework is logics with *future discounting* [1, 21, 43]. A basic fragment is given as follows: take $F = \mathcal{P}(\mathsf{AP}) \times (\_)$, $\Omega = [0, 1]$, $\Gamma = \{\mathsf{tt}, \mathsf{ff}, \wedge, \vee\}$ and $\Lambda = \mathsf{AP} \cup \{\mathsf{X}\}$, with the predicate liftings

$$[\![p]\!]_X\colon 1 \to [0, 1]^{FX},\ [\![p]\!]_X(*)(U, x) = \begin{cases} 1 & \text{if } p \in U \\ 0 & \text{otherwise,} \end{cases}$$

$$[\![\mathsf{X}]\!]_X\colon [0, 1]^X \to [0, 1]^{FX},\ d \mapsto \big[\, (U, x) \mapsto \frac{1}{2} \cdot d(x) \,\big].$$

Note the factor $\frac{1}{2}$ that discounts the value of truth in the next step.

### 3.3 Equational Presentation

In this paper we favor working with equational presentations of $\mu$-calculus formulas. Furthermore, for simplicity, we shall present $\mu$-calculus formulas as *simple* equational systems, meaning that each right-hand side is of depth at most 1.

**Definition 3.4** (simple $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$-equational system)**.** A *simple $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$-equational system* is an expression of the form

$$u_1 =_{\eta_1} \varphi_1, \quad \dots, \quad u_m =_{\eta_m} \varphi_m \tag{9}$$

where: $\eta_i \in \{\mu, \nu\}$; $u_1, \ldots, u_m \in \mathbf{Var}$ are fixed-point variables; and $\varphi_1, \ldots, \varphi_m$ are *simple* $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$*-formulas* of the form

$$u_i, \quad \boxdot_\gamma(u_{i_1}, \ldots, u_{i_{|\gamma|}}) \quad \text{or} \quad \heartsuit_\lambda(u_{i_1}, \ldots, u_{i_{|\lambda|}}).$$

We make a further requirement that, in case $\eta_i = \mu$, the corresponding equation is of the form $u_i =_\mu u_j$ for some $j \in [1, m]$. This inessential requirement simplifies our subsequent exposition.

A simple $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$-equational system (9) is *closed* if all the variables that occur in $\varphi_1, \ldots, \varphi_m$ are among $u_1, \ldots, u_m$.

Note that, much like in §2, the order of equations in (9) matters—the equations are solved from left to right, i.e. priorities increases as one goes from left to right.

Translation of $\mu$-calculus formulas into equational systems is standard; so is translation in the other direction.

**Definition 3.5** (translation)**.** For each $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$-formula $\varphi$, its *equational presentation* $E_\varphi$ is defined by the following induction. Here $u_\varphi \in \mathbf{Var}$ denotes the variable on the left-hand side of the last equation in the equational system $E_\varphi$, $E_1; \ldots; E_k$ denotes the concatenation of equational systems $E_1, \ldots, E_k$, and the variable $v$ in each clause is chosen to be a fresh one.

$$\begin{aligned}
E_u &:= \big(v =_\nu u\big) \\
E_{\boxdot_\gamma(\varphi_1,\ldots,\varphi_n)} &:= \big(E_{\varphi_1}; \ldots; E_{\varphi_n}; v =_\nu \boxdot_\gamma(u_{\varphi_1}, \ldots, u_{\varphi_n})\big) \\
E_{\heartsuit_\lambda(\varphi_1,\ldots,\varphi_n)} &:= \big(E_{\varphi_1}; \ldots; E_{\varphi_n}; v =_\nu \heartsuit_\lambda(u_{\varphi_1}, \ldots, u_{\varphi_n})\big) \\
E_{\eta u.\,\varphi} &:= \big(E_\varphi; u =_\eta u_\varphi\big) \qquad \text{where } \eta \in \{\mu, \nu\}
\end{aligned}$$

The choice of $=_\nu$ in the first three clauses is arbitrary from the semantical viewpoint: changing it into $=_\mu$ yields the same semantics. It is however beneficial from the algorithmic and presentational viewpoints—in particular the resulting system enjoys the requirement in Def. 3.4 (that a $\mu$-equation is of the form $u_i =_\mu u_j$).

It is straightforward to see that a closed formula $\varphi$ yields a closed equational system $E_\varphi$.

Conversely, given a simple $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$-equational system $E$ like in (9), we define its *formulaic presentation* $\varphi_E$ by induction on the number $m$ of equations. If $m = 1$ then an equation $u_1 =_{\eta_1} \varphi_1$ becomes the formula $\eta_1 u_1.\, \varphi_1$. For the step case, let $E'$ be obtained by dropping the first equation, that is,

$$E' = \big(u_2 =_{\eta_2} \varphi_2, \quad \ldots, \quad u_m =_{\eta_m} \varphi_m\big).$$

Then we define $\varphi_E$ to be the result of replacing $u_1$ in $\varphi_{E'}$ with $\eta_1 u_1.\, \varphi_1$. That is,

$$\varphi_E := \varphi_{E'}\big[\,\eta_1 u_1.\varphi_1 \,/\, u_1\,\big].$$

The two translations are mutually inverse—not necessarily syntactically, but the semantics is preserved. See Prop. 3.10 later. Therefore, in what follows, we do not distinguish a $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$-formula $\varphi$ and its equational presentation $E_\varphi$. Both will be denoted by $\varphi$.

**Example 3.6.** Let $\Gamma = \{\wedge, \vee\}$ and $\Lambda = \mathsf{AP} \cup \{\mathsf{X}\}$ (from Example 3.3). The $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$-formula $\nu u.\, \mu v.\, ((p \vee \mathsf{X} v) \wedge \mathsf{X} u)$ gets translated into the simple $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$-equational system

$$\begin{aligned}
&u_1 =_\mu p, \;\; u_2 =_\mu v, \;\; u_3 =_\mu \mathsf{X} u_2, \;\; u_4 =_\mu u_1 \vee u_3, \\
&u_5 =_\mu u, \;\; u_6 =_\mu \mathsf{X} u_5, \;\; u_7 =_\mu u_4 \wedge u_6, \;\; v =_\mu u_7, \;\; u =_\nu v,
\end{aligned}$$

under Def. 3.5. The translation in the other direction gives rise to a complicated formula which, however, is easily seen to be equivalent to the original formula under (obviously sound) simplifications like $\mu u_1.\, p$ into $p$.

### 3.4 $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$: Semantics

Formulas of $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$ are interpreted over $F$-coalgebras (see §3.1). The following inductive interpretation is a standard one; it follows the tradition of coalgebraic modal logic [13, 16, 24, 52] as well as that of fixed-point logics [38].

**Definition 3.7** (semantics of $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$ formulas)**.** Let $\Gamma$ and $\Lambda$ be propositional and modal signatures in Def. 3.1, and Let $c\colon X \to FX$ be a coalgebra. A formula $\varphi$ of $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$—with free variables $u_1, \ldots, u_m$—is assigned its *denotation* over $c$; it is given by a function

$$[\![\varphi]\!]_c : \big(\Omega^X\big)^m \longrightarrow \Omega^X$$

that is defined inductively in the following way. Here $\vec{V}$ is short for $V_1, \ldots, V_m$, where $V_i\colon X \to \Omega$.

$$\begin{aligned}
[\![u_i]\!]_c(\vec{V})(x) &:= V_i(x), \\
[\![\boxdot_\gamma(\varphi_1, \ldots, \varphi_n)]\!]_c(\vec{V})(x) &:= \\
&\qquad \gamma\big([\![\varphi_1]\!]_c(\vec{V})(x), \ldots, [\![\varphi_n]\!]_c(\vec{V})(x)\big), \\
[\![\heartsuit_\lambda(\varphi_1, \ldots, \varphi_n)]\!]_c(\vec{V})(x) &:= \\
&\qquad \Big(\lambda_X\big([\![\varphi_1]\!]_c(\vec{V}), \ldots, [\![\varphi_n]\!]_c(\vec{V})\big)\Big)\big(c(x)\big), \\
[\![\mu u.\, \varphi]\!]_c(\vec{V})(x) &:= \big(\mu\big([\![\varphi]\!]_c(\vec{V}, _)\colon \Omega^X \to \Omega^X\big)\big)(x), \\
[\![\nu u.\, \varphi]\!]_c(\vec{V})(x) &:= \big(\nu\big([\![\varphi]\!]_c(\vec{V}, _)\colon \Omega^X \to \Omega^X\big)\big)(x).
\end{aligned}$$

Recall that $\gamma\colon \Omega^n \to \Omega$ and $\lambda_X\colon (\Omega^X)^n \to \Omega^{FX}$ are assumed to be given (Def. 3.1). In the last two clauses it is assumed, by suitably rearranging variables, that the bound variable $u$ is the last one $u_m$ among the free variables $u_1, \ldots, u_m$ of $\varphi$. The necessary fixed points of the function $[\![\varphi]\!]_c(\vec{V}, \_)\colon \Omega^X \to \Omega^X$ are guaranteed by the Knaster-Tarski theorem, since $\Omega$ (and hence $\Omega^X$) is a complete lattice and the function $[\![\varphi]\!]_c(\vec{V}, \_)$ is easily seen to be monotone.

**Lemma 3.8.** *Let $f$ be a coalgebra homomorphism from $c\colon X \to FX$ to $d\colon Y \to FY$, as in (7). For each closed $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$-formula $\varphi$ and each $x \in X$, we have $[\![\varphi]\!]_c(x) = [\![\varphi]\!]_d(f(x))$.* □

As discussed in §3.3 we favor working with equational presentation of formulas. We shall therefore define their semantics, too.

**Definition 3.9** (semantics of simple $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$-equational systems)**.** Let $E$ be a simple $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$-equational system

$$u_1 =_{\eta_1} \varphi_1, \;\ldots,\; u_m =_{\eta_m} \varphi_m \tag{10}$$

from Def. 3.4; assume that it is closed. Let $c\colon X \to FX$ be an $F$-coalgebra.

Then $E$ and $c$ together induce an equational system $E_c$ (in the sense of Def. 2.6) over the complete lattice $L = \Omega^X$—this is by identifying a simple formula $\varphi_i$ on a right-hand side with the function $[\![\varphi_i]\!]_c\colon (\Omega^X)^m \to \Omega^X$ defined in Def. 3.7.

Finally, solving $E_c$ as in Def. 2.7 yields a solution $(l_1^{\mathrm{sol}}, \ldots, l_m^{\mathrm{sol}})$ that is an element of $(\Omega^X)^m$. The last component $l_m^{\mathrm{sol}}$ is referred to as the *semantics* of the simple $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$-equational system $E$ over the coalgebra $c$.

The two semantics—the direct one, and the one via equational presentation—coincide, as expected.

**Proposition 3.10.** *Let $\varphi$ be a closed $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$-formula, and $c\colon X \to FX$ be a coalgebra. Consider its equational presentation $E_\varphi$ (a simple $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$-equational system, Def. 3.5). Then the semantics of $E_\varphi$ over $c$—in the sense of Def. 3.9, i.e. the solution of the equational system $E_{\varphi,c}$ over $\Omega^X$—coincides with $[\![\varphi]\!]_c$ from Def. 3.7.*

*Proof.* Straightforward by induction. □

## 4. $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$ Model Checking against $F$-Coalgebras

Let us turn to the model-checking problem of the modal logic $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$ against $F$-coalgebras. Later in §5 we study model checking

against coalgebras with additional nondeterministic branching—i.e. there the logic $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$ is thought of as a "linear-time" logic. In contrast, here $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$ is a "branching-time" logic, in the sense that there is no additional branching to be abstracted away.

Prop. 3.10, together with Thm. 2.13, already gives us a characterization of the semantics $[\![\varphi]\!]_c$ in terms of progress measures. In this section we shall rephrase it to yet another form, called *MC progress measure*, that is easier to manipulate. Using it we present our main technical results, namely a generic model-checking algorithm (Algorithm 1) and its complexity (Thm. 4.13).

The following correspondence for (polyadic) modalities—that is not unlike in the Yoneda lemma—will be used in the following developments.

**Lemma 4.1** ($\lambda^{\langle j_1,\dots,j_n\rangle}, \tilde{\lambda}$)**.** *Let $\lambda$ be a natural transformation, given by arrows $\lambda_X : (\Omega^X)^n \to \Omega^{FX}$ that are natural in $X$. (This is the setting in Def. 3.1, where $\lambda$ is an $n$-ary modality). Let $m \in \omega$ and $j_1, \dots, j_n \in [1, m]$. These data induce an arrow*

$$\lambda^{\langle j_1,\dots,j_n\rangle} \colon F(\Omega^m) \to \Omega$$

*by $\lambda^{\langle j_1,\dots,j_n\rangle} := \lambda_{\Omega^m}(\pi_{j_1}, \dots, \pi_{j_n})$. Recall that $\lambda_{\Omega^m}$ is of type $(\Omega^{\Omega^m})^n \to \Omega^{F(\Omega^m)}$, and $\pi_j : \Omega^m \to \Omega$.*

*Moreover, let us define $\tilde{\lambda} \colon F(\Omega^n) \to \Omega$ by $\tilde{\lambda} := \lambda^{\langle 1,2,\dots,n\rangle} = \lambda_{\Omega^n}(\pi_1, \dots, \pi_n)$, where $\pi_1, \dots, \pi_n \colon \Omega^n \to \Omega$. Then we have*

$$\begin{array}{ccc} F(\Omega^m) & & \\ {\scriptstyle F\langle \pi_{j_1},\dots,\pi_{j_n}\rangle}\downarrow & \searrow^{\lambda^{\langle j_1,\dots,j_n\rangle}} & \\ F(\Omega^n) & \xrightarrow{\tilde{\lambda}} & \Omega \ . \end{array}$$

□

### 4.1 MC Progress Measure

We start with customizing the lattice-theoretic notion of progress measure (Def. 2.12) to one that is tailored to $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$ model checking. For reuse in later sections, the definition is separated into the transition-irrelevant part (which we call *pre-progress measure*), and the full definition.

**Definition 4.2** (pre-progress measure, pPM)**.** Let $F \colon \mathbf{Sets} \to \mathbf{Sets}$ be a functor. Let $\varphi$ be a $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$-formula—where $\Lambda$ is a modal signature over $F$—that is identified with a simple equational system $u_1 =_{\eta_1} \varphi_1, \dots, u_m =_{\eta_m} \varphi_m$ as in §3.3. Let $i_1 < \cdots < i_k$ enumerate the indices of all the $\mu$-variables.

A *pre-progress measure (pPM) $p$* for $\varphi$ is given by a tuple

$$p \ = \ \big( (\overline{\alpha_1}, \dots, \overline{\alpha_k}), \big( p_i(\alpha_1, \dots, \alpha_k) \big)_{i,\alpha_1,\dots,\alpha_k} \big)$$

that consists of:

- the *maximum prioritized ordinal* $(\overline{\alpha_1}, \dots, \overline{\alpha_k})$; and
- the *approximants* $p_i(\alpha_1, \dots, \alpha_k) \in \Omega$, defined for each $i \in [1, m]$ and each prioritized ordinal $(\alpha_1, \dots, \alpha_k)$ such that $\alpha_1 \le \overline{\alpha_1}, \dots, \alpha_k \le \overline{\alpha_k}$.

The approximants $p_i(\alpha_1, \dots, \alpha_k)$ are subject to:

1. **(Monotonicity)** Let $i \in [1, m]$ (hence $u_i$ is either a $\mu$- or $\nu$-variable). Then $(\alpha_1, \dots, \alpha_k) \preceq_i (\alpha'_1, \dots, \alpha'_k)$ implies $p_i(\alpha_1, \dots, \alpha_k) \sqsubseteq p_i(\alpha'_1, \dots, \alpha'_k)$.
2. **($\mu$-variables, base case)** Let $a \in [1, k]$. Then $\alpha_a = 0$ implies either: $p_{i_a}(\alpha_1, \dots, \alpha_k) = \bot$, or there exists a prioritized ordinal $(\alpha'_1, \dots, \alpha'_k)$ such that $(\alpha'_1, \dots, \alpha'_k) \prec_{i_a} (\alpha_1, \dots, \alpha_k)$ and $p_{i_a}(\alpha_1, \dots, \alpha_k) \sqsubseteq p_{i_a}(\alpha'_1, \dots, \alpha'_k)$. (Note here that this condition mirrors Cond. 2' of Def. 2.14, rather than Cond. 2 of Def. 2.12. Prop. 2.15 justifies doing so.)
3. **($\mu$-variables, step case)** Let $a \in [1, k]$, and let $(\alpha_1, \dots, \alpha_a + 1, \dots, \alpha_k)$ be a prioritized ordinal such that its $a$-th counter $\alpha_a + 1$ is a successor ordinal. Consider the approximant $p_{i_a}(\alpha_1, \dots, \alpha_a + 1, \dots, \alpha_k)$. Since $u_{i_a}$ is a $\mu$-variable, by a requirement in Def. 3.4 the corresponding equation is of the form $u_{i_a} =_\mu u_{i'}$ for some $i' \in [1, m]$. We require that there exist ordinals $\beta_1, \dots, \beta_{a-1}$ such that

$$p_{i_a}(\alpha_1, \dots, \alpha_a + 1, \dots, \alpha_k) \sqsubseteq p_{i'}(\beta_1, \dots, \beta_{a-1}, \alpha_a, \dots, \alpha_k)$$

and $\beta_1 \le \overline{\alpha_1}, \dots, \beta_{a-1} \le \overline{\alpha_{a-1}}$.
4. **($\mu$-variables, limit case)** Let $a \in [1, k]$, and let $(\alpha_1, \dots, \alpha_k)$ be a prioritized ordinal such that its $a$-th counter $\alpha_a$ is a limit ordinal. We require

$$p_{i_a}(\alpha_1, \dots, \alpha_a, \dots, \alpha_k) \sqsubseteq \bigsqcup_{\beta < \alpha_a} p_{i_a}(\alpha_1, \dots, \beta, \dots, \alpha_k) \ .$$

5. **($\nu$-variables)** Let $i \in [1, m] \setminus \{i_1, \dots, i_k\}$ (i.e. $u_i$ is a $\nu$-variable in the system (1)); let $a \in [1, k]$ such that

$$i_1 < \cdots < i_{a-1} < i < i_a < \cdots < i_k.$$

Let $(\alpha_1, \dots, \alpha_k)$ be a prioritized ordinal. We require the following on the approximant $p_i(\alpha_1, \dots, \alpha_k)$:

(a) **(RHS is a variable)** If the formula $\varphi_i$ in the $i$-th equation $u_i =_\nu \varphi_i$ is a variable $u_{i'}$ (for some $i' \in [1, m]$), then there exist ordinals $\beta_1, \dots, \beta_{a-1}$ such that

$$p_{i_a}(\alpha_1, \dots, \alpha_a, \dots, \alpha_k) \sqsubseteq p_{i'}(\beta_1, \dots, \beta_{a-1}, \alpha_a, \dots, \alpha_k)$$

and $\beta_1 \le \overline{\alpha_1}, \dots, \beta_{a-1} \le \overline{\alpha_{a-1}}$.

(b) **(RHS is a propositional formula)** If the formula $\varphi_i$ is a propositional formula $\boxdot_\gamma(u_{j_1}, \dots, u_{j_n})$, then there exist ordinals $\beta_1, \dots, \beta_{a-1}$ such that

$$p_i(\alpha_1, \dots, \alpha_a, \dots, \alpha_k) \sqsubseteq [\![\gamma]\!] \begin{pmatrix} p_{j_1}(\beta_1, \dots, \beta_{a-1}, \alpha_a, \dots, \alpha_k), \\ \dots, \\ p_{j_n}(\beta_1, \dots, \beta_{a-1}, \alpha_a, \dots, \alpha_k) \end{pmatrix}$$

and $\beta_1 \le \overline{\alpha_1}, \dots, \beta_{a-1} \le \overline{\alpha_{a-1}}$.

Let $\alpha$ be an ordinal. The collection of all pre-progress measures for a formula $\varphi$, whose maximum prioritized ordinal $(\overline{\alpha_1}, \dots, \overline{\alpha_k})$ satisfies $\overline{\alpha_i} = \alpha$ for each $i \in [1, k]$, shall be denoted by $\mathsf{pPM}_{\varphi,\alpha}$.

Recall that $\Omega$ is the complete lattice of truth values. In the definition of $\mathsf{pPM}_{\varphi,\alpha}$, the explicit bound by $\alpha$ is there so that the collection $\mathsf{pPM}_{\varphi,\alpha}$ is a (small) set.

Comparing the previous definition with Def. 2.12 of progress measures, what are missing here are the treatment of modal formulas $\heartsuit_\lambda(u_{j_1}, \dots, u_{j_n})$ in Cond. 5—this is precisely the case where the transition structure of the coalgebra in question becomes relevant. In the current setting of $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$ as a "branching-time" logic, this case is taken care of in the following way. MC stands for "model checking."

**Definition 4.3** (MC progress measure)**.** Assume the setting of Def. 2.12, and let $c \colon X \to FX$ be a coalgebra in $\mathbf{Sets}$. An *MC progress measure* for $\varphi$ over $c$ is given by:

- some ordinal $\alpha$, called the *maximum ordinal*, and
- a function $Q \colon X \to \mathsf{pPM}_{\varphi,\alpha}$,

that are subject to the following condition.

5(c) **($\nu$-variables, RHS is a modal formula)** Let $x \in X$ and $p := Q(x)$ be a pre-progress measure for $\varphi$. Let $i \in [1, m]$ and assume the setting of Cond. 5 of Def. 4.2 (i.e. $u_i$ is a $\nu$-variable), and further that the formula $\varphi_i$ is a modal formula: $\varphi_i = \heartsuit_\lambda(u_{j_1}, \dots, u_{j_n})$.

Now consider the approximant $p_i(\alpha_1, \dots, \alpha_a, \dots, \alpha_k) \in \Omega$ of $p$. We require there exist ordinals $\beta_1, \dots, \beta_{a-1}$ such that

$$p_i(\alpha_1, \dots, \alpha_a, \dots, \alpha_k) \sqsubseteq \mathsf{PT}_{\heartsuit_\lambda(u_{j_1}, \dots, u_{j_n})}(\beta_1, \dots, \beta_{a-1}, \alpha_a, \dots, \alpha_k)\big((FQ \circ c)(x)\big), \tag{11}$$

and $\beta_1 \le \alpha, \dots, \beta_{a-1} \le \alpha$.

Note that $(FQ \circ c)(x) \in F(\mathsf{pPM}_{\varphi,\alpha})$ since $X \xrightarrow{c} FX \xrightarrow{FQ} F(\mathsf{pPM}_{\varphi,\alpha})$. For each $(\alpha'_1, \dots, \alpha'_k)$, the function

$$\mathsf{PT}_{\heartsuit_\lambda(u_{j_1}, \dots, u_{j_n})}(\alpha'_1, \dots, \alpha'_k) \;:\; F(\mathsf{pPM}_{\varphi,\alpha}) \to \Omega$$

in (11) is defined as follows. (The name $\mathsf{PT}$ comes from "predicate transformer.")

$$\mathsf{PT}_{\heartsuit_\lambda(u_{j_1}, \dots, u_{j_n})}(\alpha'_1, \dots, \alpha'_k) := \Big[F(\mathsf{pPM}_{\varphi,\alpha}) \xrightarrow{F(\mathsf{ev}(\alpha'_1, \dots, \alpha'_k))} F(\Omega^m) \xrightarrow{\lambda^{\langle j_1, \dots, j_n\rangle}} \Omega\Big], \tag{12}$$

where $\lambda^{\langle j_1, \dots, j_n\rangle}$ is from Lem. 4.1, and the function

$$\mathsf{ev}(\alpha'_1, \dots, \alpha'_k) \colon \mathsf{pPM}_{\varphi,\alpha} \to \Omega^m$$

is defined by "fixing a prioritized ordinal," that is,

$$\mathsf{ev}(\vec{\alpha'})(p) := \big(p_1(\vec{\alpha'}), \dots, p_m(\vec{\alpha'})\big) \in \Omega^m.$$

The composite in the definition of $\mathsf{PT}_{\heartsuit_\lambda(u_{j_1}, \dots, u_{j_n})}(\vec{\alpha'})$ in (12) might seem exotic, but the definition here is in fact a straightforward adaptation of the common interpretation of modal formulas in coalgebraic logics. Recall the interpretation of a modal formula $\heartsuit_\lambda(\varphi_1, \dots, \varphi_n)$ in Def. 3.7, that is also the standard one in the literature (see e.g. [52]). Then it is not hard—by naturality of $\lambda$, much like in the proof of Thm. 4.4—that this standard definition of $[\![\heartsuit_\lambda(\vec{\varphi})]\!]_c$ is equivalent to the following, where $[\![\varphi_i]\!]_c \colon X \to \Omega$ are the interpretations of the constituent subformulas (for $i \in [1, n]$), and $\tilde{\lambda}$ is from Lem. 4.1.

$$[\![\heartsuit_\lambda(\vec{\varphi})]\!]_c \;=\; \big(X \xrightarrow{c} FX \xrightarrow{F\langle [\![\varphi_i]\!]_c\rangle_i} F(\Omega^n) \xrightarrow{\tilde{\lambda}} \Omega\big).$$

This indeed resembles the right-hand side of (11), namely

$$\big(X \xrightarrow{c} FX \xrightarrow{F(\mathsf{ev}(\vec{\alpha'}) \circ Q)} F(\Omega^m) \xrightarrow{\lambda^{\langle j_1, \dots, j_n\rangle}} \Omega\big)(x).$$

**Theorem 4.4** (correctness of MC progress measure)**.** *Assume the setting of Def. 4.3. In particular, the formula $\varphi$ is translated to an equational system with $m$ variables.*

1. ***(Soundness)** Let $Q$ be an MC progress measure (with the maximum ordinal $\alpha$), $x \in X$ and $p := Q(x)$. Then*
$$p_m(\alpha, \dots, \alpha) \sqsubseteq [\![\varphi]\!]_c(x),$$
*where $[\![\varphi]\!]_c \colon X \to \Omega$ is from Def. 3.7.*
2. ***(Completeness)** There exists an MC progress measure $Q$ that achieves the optimal. That is, an MC progress measure $Q$ such that $(Q(x))_m(\alpha, \dots, \alpha) = [\![\varphi]\!]_c(x)$ for each $x \in X$. Moreover, $Q$ can be chosen so that its maximum ordinal $\alpha$ is $\alpha = \mathsf{ascCL}(\Omega^X)$, where $\mathsf{ascCL}(\Omega^X)$ is the length of the longest strictly ascending chain in $\Omega^X$ (see Thm. 2.13.2).* □

### 4.2 Algorithms

Here we shall further translate the notion of MC progress measure (Def. 4.3) to a Jurdzinski-style presentation; the latter shall be called a *matrix progress measure*. The correspondence is an extension of the one in Appendix A (found in the extended version [27]); see also Rem. 2.5. We shall then devise a model-checking algorithm based on matrix progress measures. Thanks to the concrete presentation with matrices, we believe its implementation is a fairly straightforward task.

**Assumption 4.5.** Throughout §4.2 we focus on the Boolean setting (i.e. $\Omega = \mathbf{2}$), and restrict the state space $X$ of the coalgebra $c \colon X \to FX$ (as a system model) to be finite. This is a reasonable assumption because we aim at a concrete algorithm. In view of Thm. 4.4.2, in employing the theoretical machinery developed so far, all the ordinals that occur can be restricted to finite (since $\mathsf{ascCL}(\mathbf{2}^X) = |X|$ is finite).

Furthermore, we restrict the propositional signature $\Gamma$ to $\Gamma_n := \{\bigwedge_n, \bigvee_n\}$, where $\bigwedge_n$ and $\bigvee_n$ are the $n$-ary conjunction and disjunction operators with obvious interpretations. This signature of $\Gamma$ is functionally complete in the current monotonic Boolean setting: any other propositional connective $\gamma \colon \mathbf{2}^n \to \mathbf{2}$ can be encoded by

$$\bigvee\{\bigwedge\{l_{i_1}, \dots, l_{i_k}\} \,|\, l_{i_1} = \cdots = l_{i_k} = \mathrm{tt} \Rightarrow \gamma(l_1, \dots, l_n) = \mathrm{tt}\}.$$

**Definition 4.6** (prioritized ordinal matrix, POM)**.** Assume the setting of Def. 4.2. A *prioritized ordinal matrix* is an $m \times k$ matrix

$$\begin{bmatrix} \alpha_1^{(1)} & \cdots & \alpha_k^{(1)} \\ \vdots & \ddots & \vdots \\ \alpha_1^{(m)} & \cdots & \alpha_k^{(m)} \end{bmatrix},$$

where each entry $\alpha_a^{(i)}$ is either

- an ordinal, or
- the symbol ♠ for "failure."

It is required that, if any entry $\alpha_a^{(i)}$ is ♠ then all entries on the same row is ♠, that is, $\alpha_1^{(i)} = \cdots = \alpha_k^{(i)} = ♠$.

The set of all POMs, such that all the ordinals therein are no bigger than $\alpha$, is denoted by $\mathsf{POM}_\alpha$.

A POM is therefore an $m$-tuple of prioritized ordinals, where some prioritized ordinals can be replaced by ♠. Its $i$-th row will be a prioritized ordinal for the $i$-th variable $u_i$. In view of the monotonicity conditions (in Def. 2.12 and 4.2) and the definition of $\preceq_i$ (Def. 2.10), we can see that some first elements in a row (precisely: those which correspond to $\mu$-variables with a smaller priority than $u_i$) do not make any difference. Such entries can safely be denoted by $*$ ("arbitrary"). An example is shown in the above: it is a POM for an equational system with 5 variables, in which $u_1, u_3, u_4$ are $\mu$-variables and $u_2, u_5$ are $\nu$-variables. We shall however restrict use of $*$ for providing intuitions; it does not appear in the technical developments.

$$\begin{array}{c} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{array} \begin{bmatrix} 3 & 5 & 2 \\ * & 1 & 0 \\ ♠ & ♠ & ♠ \\ * & * & 4 \\ * & * & * \end{bmatrix}$$

In the current section (§4.2) where $X$ is assumed to be a finite set, it is not needed to allow any ordinal as an entry of a POM (Def. 4.6). Natural numbers will just suffice.

**Definition 4.7** (matrix progress measure, MPM)**.** Assume the setting of Def. 4.3. A *matrix progress measure (MPM)* for $\varphi$ over $c$, with a maximum ordinal $\alpha$, is a function $R \colon X \to \mathsf{POM}_\alpha$ that satisfies the following conditions. Let $x \in X$ be arbitrary, and consider $R(x) \in \mathsf{POM}_\alpha$.

2. **($\mu$-variables, base case)** Let $a \in [1, k]$ and consider the corresponding $\mu$-variable $u_{i_a}$. Assume $\alpha_1^{(i_a)} \neq ♠$. Then we must have $(R(x))^{(i_a)} \succ_{i_a} (0, 0, \dots, 0)$. Note that the $i_a$-th row $(R(x))^{(i_a)}$ of $R(x)$ is a prioritized ordinal, and recall $\succ_{i_a}$ from Def. 2.10. Note also that the required inequality is strict.
(That is, a row in $R(x)$ that corresponds to a $\mu$-variable must not be $(*, \dots, *, 0, \dots, 0)$.)
3. **($\mu$-variables, step case)** Let $a \in [1, k]$ and consider the corresponding $\mu$-variable $u_{i_a}$. Assume $\alpha_1^{(i_a)} \neq ♠$. Let $u_{i_a} =_\mu u_{i'}$ (where $i' \in [1, m]$) be the corresponding equation in $\varphi$. If

$i' \leq i_a$, then we must have $(R(x))^{(i_a)} \succ_i (R(x))^{(i')}$. Note the inequality is strict.

4. **($\mu$-variables, limit case)** Let $a \in [1,k]$, and consider the corresponding $\mu$-variable $u_{i_a}$. Then $(R(x))^{(i_a)}_a$ must not be a limit ordinal. (This condition is vacuous when $\alpha$ is finite.)
5. **($\nu$-variables)** Let $i \in [1,m] \setminus \{i_1, \dots, i_k\}$ (i.e. $u_i$ is a $\nu$-variable). Assume that $\alpha_1^{(i)} \neq \spadesuit$. Let $u_i =_\nu \varphi_i$ be the corresponding equation in $\varphi$.
   (a) **(RHS is a variable)** If the formula $\varphi_i$ is a variable $u_{i'}$ (for some $i' \in [1,m]$). Then $(R(x))^{(i)} \succeq_i (R(x))^{(i')}$.
   (b) **(RHS is a propositional formula)** Recall that we have restricted propositional connectives to $\bigwedge$ and $\bigvee$ (Assumption 4.5). If $\varphi_i = \bigwedge(u_{i_1}, \dots, u_{i_n})$ then we require all of
   $$(R(x))^{(i)} \succeq_i (R(x))^{(i_1)}, \dots, (R(x))^{(i)} \succeq_i (R(x))^{(i_n)} \tag{13}$$
   to hold. If $\varphi_i = \bigvee(u_{i_1}, \dots, u_{i_n})$ then we require at least one of (13) to hold.
   (c) **(RHS is a modal formula)** Assume that the formula $\varphi_i$ is a modal formula $\varphi_i = \heartsuit_\lambda(u_{j_1}, \dots, u_{j_n})$. Let $\vec{\alpha} = (\alpha_1, \dots, \alpha_k) := (R(x))^{(i)}$. Consider the following composite $h \colon X \to \mathbf{2}$:
   $$h := \left( \begin{array}{c} X \xrightarrow{c} FX \xrightarrow{FR} F(\mathsf{POM}_\alpha) \xrightarrow{F(\mathsf{ev}'(\vec{\alpha}))} \\ F(\mathbf{2}^m) \xrightarrow{\lambda^{\langle j_1, \dots, j_n \rangle}} \mathbf{2} \end{array} \right), \tag{14}$$
   where $\mathsf{ev}'(\vec{\alpha}) \colon \mathsf{POM}_\alpha \to \mathbf{2}^m$ is defined by
   $$\Big(\mathsf{ev}'(\vec{\alpha})\big((\beta_j^{(i)})_{i,j}\big)\Big)_{i'} = \mathrm{tt} \overset{\text{def.}}{\iff} \vec{\alpha} \succeq_{i'} (\beta_1^{(i')}, \dots, \beta_k^{(i')})$$
   for each $i' \in [1,m]$. We require that $h(x) = \mathrm{tt}$.

Again, much like for Cond. 5(c) in Def. 4.3, the composite in (14) is understood as an analogue of the usual interpretation of modal formulas in coalgebraic logics (cf. Def. 3.7).

**Theorem 4.8** (correctness of MPM)**.** *Assume the setting of Def. 4.3.*

1. ***(Soundness)** If there exists an MPM $R \colon X \to \mathsf{POM}_\alpha$ such that $(R(x))^{(m)}_k \neq \spadesuit$, then $[\![\varphi]\!]_c(x) = \mathrm{tt}$.*
2. ***(Completeness)** There is an optimal MPM $R_0 \colon X \to \mathsf{POM}_{|X|}$ such that: $[\![\varphi]\!]_c(x) = \mathrm{tt}$ if and only if $(R_0(x))^{(m)}_k \neq \spadesuit$.* □

We follow [33] and present an algorithm that looks for the optimal MPM. See Algorithm 1. There we use the following functions.

**Definition 4.9** ($\max_{\preceq_i}, \min_{\preceq_i}$)**.** In Algorithm 1, the function $\max_{\preceq_i}$ takes a set of prioritized ordinals (and possibly $(\spadesuit, \dots, \spadesuit)$) and returns a prioritized ordinal such that: the first irrelevant entries (due to priorities smaller than that of $u_i$) are set to 0; and the rest is the maximum (with the lexicographic order, the latter the more significant) among the corresponding suffixes of the prioritized ordinals given as input. In case the input set contains $(\spadesuit, \dots, \spadesuit)$, then the output is $(\spadesuit, \dots, \spadesuit)$ too.

For example, in the setting of Example 2.11,

$$\max_{\preceq_3}\{(1,2,3),(3,4,1)\} = (0,2,3)$$

where the first element of each sequence is irrelevant.

The function $\min_{\preceq_i}$ is defined similarly, by: truncating the first irrelevant elements, choosing the smallest one in the lexicographic order, and padding the missing elements with 0. The output is $(\spadesuit, \dots, \spadesuit)$ in case the input set contains nothing other than $(\spadesuit, \dots, \spadesuit)$.

The functions $\max_{\preceq_i}$ and $\min_{\preceq_i}$ can be efficiently implemented: if the input is the set of $N$ prioritized ordinals then the time complexity is $O(Nk)$.

**Algorithm 1** An algorithm for $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$ model checking, in the setting of Def. 4.3 and Assumption 4.5. Here $R(x,i)$ denotes the prioritized ordinal $\big(R(x,i,1), \dots, R(x,i,k)\big)$. Note that on lines 16, 19 and 23, $u_i$ is necessarily a $\nu$-variable.

**Input:** A $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$-formula $\varphi$ presented as an equational system $u_1 =_{\eta_1} \varphi_1, \dots, u_m =_{\eta_m} \varphi_m$ where $u_{i_1}, \dots, u_{i_k}$ are $\mu$-variables, and a coalgebra $c \colon X \to FX$
**Output:** $[\![\varphi]\!]_c \in \mathbf{2}^X$

1: **for** each $x \in X$, $i \in [1,m]$ and $j \in [1,k]$ **do** ▷ initialization
2: $R(x,i,j) := 0$
3: **end for**
4: **for** each $a \in [1,k]$ **do** ▷ Cond. 2
5: $R(x, i_a, a) := 1$
6: **end for**
7: **repeat** ▷ the main loop
8: **for** each $x \in X$ and $i \in [1,m]$ **do**
9: **if** $u_i$ is a $\mu$-variable, $i = i_a$ and $\varphi_i = u_{i'}$ **then** ▷ Cond. 3
10: $R(x,i) := \max_{\preceq_i}\{\, R(x,i),$ ▷ cf. Def. 4.9
11: $\big(R(x,i',1), \dots, R(x,i',a)+1, \dots, R(x,i',k)\big)\,\}$
12: **end if**
13: **if** $u_i$ is a $\nu$-variable and $\varphi_i = u_{i'}$ **then** ▷ Cond. 5(a)
14: $R(x,i) := \max_{\preceq_i}\{\, R(x,i),\ R(x,i')\,\}$
15: **end if**
16: **if** $\varphi_i = \bigwedge(u_{j_1}, \dots, u_{j_n})$ **then** ▷ Cond. 5(b), the $\bigwedge$-case
17: $R(x,i) := \max_{\preceq_i}\{\, R(x,i),\ R(x,j_1), \dots, R(x,j_n)\,\}$
18: **end if**
19: **if** $\varphi_i = \bigvee(u_{j_1}, \dots, u_{j_n})$ **then** ▷ Cond. 5(b), the $\bigvee$-case
20: $R(x,i) := \max_{\preceq_i}\{\, R(x,i),$
21: $\min_{\preceq_i}\{\, R(x,j_1), \dots, R(x,j_n)\,\}\,\}$
22: **end if**
23: **if** $\varphi_i = \heartsuit_\lambda(u_{j_1}, \dots, u_{j_n})$ **then** ▷ Cond. 5(c)
24: $R(x,i) := \max_{\preceq_i}\{\, R(x,i),\ \mathsf{PT}_i^{\mathrm{M}}(x)\,\}$ ▷ cf. Def. 4.10
25: **end if**
26:
27: **for** each $j \in [1,k]$ **do**
28: **if** $R(x,i,j) > |X|$ **then** ▷ $u_i$ has seen to be false at $x$
29: $R(x,i) := (\spadesuit, \dots, \spadesuit)$
30: **end if**
31: **end for**
32: **end for**
33: **until** no change is made
34: **return** $\{x \in X \mid R(x,m,k) \neq \spadesuit\}$

**Definition 4.10** ($\mathsf{PT}_i^{\mathrm{M}}$)**.** In Algorithm 1, the function $\mathsf{PT}_i^{\mathrm{M}}$ takes a state $x \in X$ and returns

$$\mathsf{PT}_i^{\mathrm{M}}(x) := \min_{\preceq_i}\Big\{\, \vec{\alpha} \in |X|^k \;\Big|\; \big(\lambda^{\langle j_1, \dots, j_n \rangle} \circ F\big(\mathsf{ev}'(\vec{\alpha}) \circ R\big) \circ c\big)(x) = \mathrm{tt} \,\Big\} \tag{15}$$

where the composite is from (14) and $R \colon X \to \mathsf{POM}_\alpha$ is given by the current values of $\big(R(x,i,j)\big)_{x,i,j}$ in the algorithm.

The complexity of $\mathsf{PT}_i^{\mathrm{M}}$ depends greatly on the choice of a functor $F$ and a predicate lifting $\lambda$. A uniform and brute-force algorithm for $\mathsf{PT}_i^{\mathrm{M}}$ is possible, however, by enumerating all $\vec{\alpha} \in |X|^k$ from the smaller ones with respect to $\preceq_i$, and checking for each $\vec{\alpha}$ whether the condition in (15) is satisfied. The worst-case complexity is $O(km^2|X|^{k+1} + C|X|^k)$ with some constant $C$, on the assumption that the value

$$\big(\lambda^{\langle j_1, \dots, j_n \rangle} \circ F(\mathsf{ev}'(\vec{\alpha})) \circ FR \circ c\big)(x) \quad \text{that appear in (15)}$$

is computed in time $O(km^2|X| + C)$. The last assumption is derived as follows: the computation of $\mathsf{ev}'(\vec{\alpha})$ is in $O(km)$; hence the computation of $F(\mathsf{ev}'(\vec{\alpha}))$ is in $O(km|X|)$; that of $\lambda^{\langle j_1, \dots, j_n \rangle}$ is in $O(m)$ (exploiting Lem. 4.1); and the other components like $c$ and application of $F$ have only a constant contribution $C$ to the complexity.

**Remark 4.11.** Most $F$ and $\Lambda$ allow much better complexity of $\mathsf{PT}_i^{\mathrm{M}}$. For example, the choice $F = \mathcal{P}(\mathsf{AP}) \times (\_)$ and $\lambda = \mathsf{X}$ (the next-time modality) in Example 3.3.6 (that will yield a logic like LTL in §5), the function $\mathsf{PT}_i^{\mathrm{M}}$ picks up the prioritized ordinal $R(x', i)$ of the successor $x$ and truncates its first irrelevant elements to 0. This can be done in time $O(k)$. More generally, often it is possible to "propagate backwards" by computing $\{t \in F(\Omega^m) \mid \lambda^{\langle j_1,\dots,j_n\rangle}(t) = \mathrm{tt}\}$, for which a *one-step complete* set of deduction rules can be used (see e.g. [16]). Such optimizations by deduction rules are left as future work.

**Theorem 4.12.** *Algorithm 1 indeed returns* $[\![\varphi]\!]_c$. □

The following complexity result is derived from an analysis of Algorithm 1. Recall that it assumes a brute-force algorithm for $\mathsf{PT}_i^{\mathrm{M}}$ (Def. 4.10); fixing $F$ and $\Lambda$ will allow further optimization. See Rem. 4.11. It nevertheless achieves a complexity that is exponential only in $k$. This is much like the most known complexity results for model-checking (see e.g. [19, 59])—note that $k$ bounds the alternation depth of a formula $\varphi$.

**Theorem 4.13** (complexity)**.** *In the setting of Def. 4.3 and Assumption 4.5, the model-checking problem can be decided in time*

$$O\big(\, m^2(km^2|X| + C)|X|^{k+2}(|X|+1)^k \,\big). \qquad \square$$

A straightforward optimization is possible: each iteration of the inner loop (lines 8–32) tests all $(x, i)$; this is unnecessary. Algorithm 1 is presented as it is, however, since the correspondence to Def. 4.7 is clearer. It should be possible also to improve the complexity so that it is exponential to the alternation depth, instead of to the number $k$ of $\mu$-operators, of the given formula $\varphi$.

## 5. Coalgebraic $\mu$-Calculus $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$ as a Nondeterministic Linear-Time Logic

In this section we adapt the previous results to the setting where we think of $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$ as a (nondeterministic) *linear-time* logic, that is, where a system in question exhibits nondeterministic branching over transitions of type $F$. Such a system is represented as a function $c\colon X \to \mathcal{P}FX$.

Our main results here are: 1) categorical characterization of the truth value of a linear-time logic formula using progress measures (Thm. 5.6); 2) a "smallness" result that cuts down the search spaces for linear-time model checking (Thm. 5.7); and 3) a decision procedure (Thm. 5.9) that depends on the smallness result.

### 5.1 Coalgebraic Preliminaries

In what follows we will be dealing with coalgebras of the type $c\colon X \to \mathcal{P}FX$, where $F\colon \mathbf{Sets} \to \mathbf{Sets}$ (that is like in §3.1) is understood as the type of *linear-time behaviors*, and $\mathcal{P}$ is the powerset monad.

This is a common setting taken in the coalgebraic studies of *trace semantics*. The use of a monad $T$ in a coalgebra $c\colon X \to TFX$ with "$T$-branching over $F$-linear time behaviors" originates in [50], and is subsequently adopted e.g. in [11, 26, 30, 34, 56].[2] The formalization in the current paper most closely follows that in [56]. We shall again present minimal preliminaries to this *Kleisli approach* to coalgebraic trace semantics. See e.g. [26, 56] for further details; for monads and Kleisli categories see [39].

A *monad* $T$ on $\mathbf{Sets}$ is an endofunctor equipped with natural transformations $\eta^T\colon \mathrm{id} \Rightarrow T$ (*unit*) and $\mu^T\colon T \circ T \Rightarrow T$ (*multiplication*) that are subject to certain "monoid" commutative diagrams. In our current example of the powerset monad $\mathcal{P}$, its unit $\eta^{\mathcal{P}}$ is the singleton map and its multiplication $\mu^{\mathcal{P}}$ is given by union. In the class of examples of $T$ that are relevant to us, the unit turns an element into "a branching with a unique choice"; and the multiplication "suppresses" two transitions into one (see [26]).

The *Kleisli category* $\mathcal{K}\ell(T)$ has sets as its objects, and an arrow $X \mathrel{+\!\!\to} Y$ in $\mathcal{K}\ell(T)$ is given by a function $X \to TY$. It becomes a category using the monad structure of $T$. For example, given two successive arrows $f\colon X \mathrel{+\!\!\to} Y$ and $g\colon Y \mathrel{+\!\!\to} Z$ in $\mathcal{K}\ell(T)$, its composition $g \odot f\colon X \mathrel{+\!\!\to} Z$ is given by the composite $X \xrightarrow{f} TY \xrightarrow{Tg} T(TZ) \xrightarrow{\mu_Z} TZ$ of functions. It is also easy to see that we have the so-called *Kleisli inclusion* functor $J\colon \mathbf{Sets} \to \mathcal{K}\ell(T)$ by $JX = X$ and $Jf = \eta^T \circ f$.

Note that we used the symbols $\mathrel{+\!\!\to}$ and $\odot$ (as opposed to $\to$ and $\circ$) for constructs in $\mathcal{K}\ell(T)$, for distinction. In what follows we stick to this convention.

Note that for our example of $T = \mathcal{P}$, the Kleisli category $\mathcal{K}\ell(\mathcal{P})$ is nothing but the category $\mathbf{Rel}$ of sets and binary relations. We will however stick to $\mathcal{K}\ell(\mathcal{P})$, hoping that the theory will be transported to other monads (such as the Giry monad on $\mathbf{Meas}$, for probabilistic branching).

The following is our current notion of system model. For technical reasons, we impose certain conditions on $F$. These conditions are common ones and imposed also in [26, 30, 34].

**Definition 5.1** (nondeterministic $F$-coalgebra)**.** Let $F\colon \mathbf{Sets} \to \mathbf{Sets}$ be a functor, such that the following hold.

1. A final coalgebra $\zeta\colon Z \xrightarrow{\cong} FZ$ exists in $\mathbf{Sets}$.
2. The functor $F$ comes with a distributive law $\xi\colon F\mathcal{P} \Rightarrow \mathcal{P}F$ over the powerset monad $\mathcal{P}$ (which, as is well-known [30], induces a lifting $\overline{F}\colon \mathcal{K}\ell(\mathcal{P}) \to \mathcal{K}\ell(\mathcal{P})$ of $F$).

A *nondeterministic $F$-coalgebra* is $c\colon X \to \mathcal{P}FX$ in $\mathbf{Sets}$, that is, an arrow $c\colon X \mathrel{+\!\!\to} \overline{F}X$ in the Kleisli category $\mathcal{K}\ell(\mathcal{P})$.

Examples of such functors are polynomial functors inductively generated by

$$F, F_i \;::=\; \mathrm{id} \mid A \mid F_1 \times F_2 \mid \coprod_{i\in I} F_i$$

where $A$ is a constant functor that takes any set to $A \in \mathbf{Sets}$. See e.g. [26, 56] for further details on Cond. 1–2.

In view of §3.1, in the current setting, we can identify a state $z$ of a final coalgebra $\zeta\colon Z \xrightarrow{\cong} FZ$ with a (possibly infinite, long-term) *linear-time behavior* of the type $F$. For example, when $F = \mathcal{P}(\mathsf{AP}) \times (\_)$ (Example 3.3.6), a final coalgebra is carried by the set $Z = (\mathcal{P}(\mathsf{AP}))^\omega$ of infinite streams of subsets of $\mathsf{AP}$. Such streams are commonly called *computations* in the context of model checking.

The following result [30] allows us to characterize, in categorical terms, the set of possible (linear-time) $F$-behaviors of a nondeterministic $F$-coalgebra.[3] The same holds in a probabilistic setting, too; see e.g. [56].

[2] Another common coalgebraic formalization of linear-time semantics is via *determinization*, and uses Eilenberg-Moore categories (as opposed to Kleisli) as base categories. See e.g. [4, 32]. Adapting the current model-checking framework to this Eilenberg-Moore approach seems hard: fixed-point specifications are usually interpreted over *infinitary* traces such as infinite words; and this makes determinization, the core of the Eilenberg-Moore approach, much more complicated (like Büchi word automata become Rabin automata, see e.g. [57]).

[3] In papers like [26] coalgebraic *finite* trace semantics is studied. Here "finite" means linear-time behaviors that eventually come to halt within finitely many steps; and the set of finite $F$-behaviors is identified with the carrier of an *initial $F$-algebra* in $\mathbf{Sets}$ (as opposed to a final $F$-coalgebra).

**Proposition 5.2** (coalgebraic infinitary[4] trace semantics [30])**.** *Let $F\colon \mathbf{Sets} \to \mathbf{Sets}$ be a functor that satisfies the conditions in Def. 5.1; and $c\colon X \to \mathcal{P}FX$ be a nondeterministic $F$-coalgebra. Consider the diagram*

$$\begin{array}{ccc} \overline{F}X & \xrightarrow{\overline{F}f} & \overline{F}Z \\ {\scriptstyle c}\uparrow\!\dagger & & \cong\uparrow\!\dagger{\scriptstyle J\zeta} \\ X & \xrightarrow[f]{} & Z \end{array} \qquad \text{in } \mathcal{K}\ell(\mathcal{P}); \tag{16}$$

*then: 1) there exists at least one function $f\colon X \to \mathcal{P}Z$ that makes the diagram commute; and 2) among such $f$, there exists the greatest one with respect to (the pointwise extension of) the inclusion order in $\mathcal{P}Z$. The greatest one shall be denoted by $\mathrm{tr}(c)\colon X \to \mathcal{P}Z$ and called the* (infinitary) trace semantics *of $c$. Moreover, an element $z \in \mathrm{tr}(c)(x)$—identified with a single linear-time behavior over time—is referred to as an* infinitary trace *of $c$ from $x$.*

We note that the definition of $\mathrm{tr}(c)$ in Prop. 5.2 amounts to the following: $\mathrm{tr}(c)$ is the greatest fixed point of the monotone function

$$\Psi\colon \mathcal{K}\ell(\mathcal{P})(X,Z) \to \mathcal{K}\ell(\mathcal{P})(X,Z), \quad f \mapsto (J\zeta)^{-1} \odot \overline{F}f \odot c \tag{17}$$

where $\odot$ denotes composition of arrows in $\mathcal{K}\ell(\mathcal{P})$.

It has been observed that, for many examples of the functor $F$, the greatest homomorphism $\mathrm{tr}(c)$ in Prop. 5.2 indeed captures the set of all possible linear-time behaviors. See e.g. [11] and [56, Appendix A.2].

### 5.2 $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$ as a Linear-Time Logic

We take a modal language $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$ whose modal signature $\Lambda$ is over $F$. Hence a $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$-formula $\varphi$ specifies a property of $F$-behaviors, where the latter are identified with elements $z \in Z$ of a final coalgebra $\zeta\colon Z \xrightarrow{\cong} FZ$. See §3.1.

**Definition 5.3** (semantics of the logic $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$ over nondeterministic $F$-coalgebra)**.** Let $\varphi$ be a closed $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$-formula, and $c\colon X \to \mathcal{P}FX$ be a nondeterministic $F$-coalgebra. The *denotation* of $\varphi$ over $c$ is given by a function $[\![\varphi]\!]_c\colon X \to \mathcal{P}(\Omega)$ defined by

$$[\![\varphi]\!]_c := \big(X \overset{\mathrm{tr}(c)}{\longrightarrow\!\!\!\!\!\rightarrow} Z \overset{J([\![\varphi]\!]_\zeta)}{\longrightarrow\!\!\!\!\!\rightarrow} \Omega\big)$$

where: $\mathrm{tr}(c)$ is the infinitary trace semantics of $c$ (Prop. 5.2); $[\![\varphi]\!]_\zeta$ is the denotation of $\varphi$ over the (proper) $F$-coalgebra $\zeta\colon Z \to FZ$ defined in Def. 3.7; and $J\colon \mathbf{Sets} \to \mathcal{K}\ell(\mathcal{P})$ is the Kleisli inclusion functor (§5.1).

Given a nondeterministic $F$-coalgebra $c$ and its state $x$, a typical question is whether some (or all) of its linear-time behaviors satisfy a formula $\varphi$. This problem is the *existential* (or *universal*) model-checking problem, respectively. In the current paper we focus on existential model checking.

**Example 5.4.** Take the combination of $F, \Omega, \Gamma$ and $\Lambda$ in Example 3.3.6. A Kripke structure can then be thought of as a nondeterministic $F$-coalgebra.[5] Recall that a final coalgebra is carried by the set $(\mathcal{P}(\mathsf{AP}))^\omega$ of computations; in this case the infinitary trace semantics $\mathrm{tr}(c)\colon X \to \mathcal{P}\big((\mathcal{P}(\mathsf{AP}))^\omega\big)$ is precisely the map that carries each state $x \in X$ to the set of computations that arise from the paths from $x$.

[4] Note that "infinitary" does not mean that a behavior is necessarily of an infinite length. For example, if $F = \{\checkmark\} + A \times (\_)$, a final $F$-coalgebra is carried by the set $Z = A^* + A^\omega$ of all words over $A$ of finite or infinite length. All words (finite or infinite) are deemed to be "infinitary" traces.

[5] A Kripke structure is most naturally modeled by a function $c'\colon X \to \mathcal{P}(\mathsf{AP}) \times \mathcal{P}X$. This gives rise to a function $c\colon X \to \mathcal{P}\big(\mathcal{P}(\mathsf{AP}) \times X\big)$ in an obvious way that turns state-labels into transition-labels, namely $c(x) = \{((\pi_1 \circ c')(x), x') \mid x' \in (\pi_2 \circ c')(x)\}$.

A $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$-formula $\varphi$ is interpreted over elements of a final coalgebra, i.e. computations. Overall, Def. 5.3 in this setting yields the set of truth values that $\varphi$ can take, ranging over all the possible computations $z \in \mathrm{tr}(c)(x)$ that start from the given state $x \in X$.

### 5.3 (Existential) Linear-Time Model-Checking for $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$

We shall follow essentially the same path as in §4.1. We shall use precisely the same notion of pre-progress measure (Def. 4.2). The additional compatibility condition with the dynamic structure of the system in question is different reflecting the difference between the systems in question ($X \to FX$ in $\mathbf{Sets}$, or $X \nrightarrow FX$ in $\mathcal{K}\ell(\mathcal{P})$).

The following is a counterpart of Def. 4.3; LT is for *linear-time*.

**Definition 5.5** (LTMC progress measure)**.** Let $\varphi$ be a $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$-formula, identified with a simple $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$-equational system $u_1 =_{\eta_1} \varphi_1, \ \ldots, \ u_m =_{\eta_m} \varphi_m$. Let $c\colon X \to \mathcal{P}FX$ be a nondeterministic $F$-coalgebra (with some conditions on $F$; see Def. 5.1). An *LTMC progress measure* for $\varphi$ over $c$ is given by a tuple $(\alpha, Y \xrightarrow{q} FY, r, s)$ of:

- some ordinal $\alpha$,
- an $F$-coalgebra $q\colon Y \to FY$, and
- functions $r\colon Y \to \mathsf{pPM}_{\varphi,\alpha}$ and $s\colon Y \to X$

that are subject to the following condition. Let $y \in Y$.

5(c) **($\nu$-variables, RHS is a modal formula)** In the setting of Cond. 5 of Def. 4.2, assume further that the formula $\varphi_i$ is a modal formula: $\varphi_i = \heartsuit_\lambda(u_{j_1}, \ldots, u_{j_n})$.
Consider the approximant $p_i(\alpha_1, \ldots, \alpha_a, \ldots, \alpha_k) \in \Omega$ of $p := r(y)$. There must exist ordinals $\beta_1, \ldots, \beta_{a-1}$ such that

$$p_{i_a}(\alpha_1, \ldots, \alpha_a, \ldots, \alpha_k) \sqsubseteq \mathsf{PT}_{\heartsuit_\lambda(u_{j_1}, \ldots, u_{j_n})}(\beta_1, \ldots, \beta_{a-1}, \alpha_a, \ldots, \alpha_k)\big((Fr \circ q)(y)\big), \tag{18}$$

and $\beta_1 \le \alpha, \ldots, \beta_{a-1} \le \alpha$.

6. **(Compatibility with $c$)** For each $y \in Y$ we have $(Fs \circ q)(y) \in c(x)$. That is diagrammatically:

$$\begin{array}{ccc} \overline{F}Y & \xrightarrow{\overline{F}Js} & \overline{F}X \\ {\scriptstyle Jq}\uparrow\!\dagger & \subseteq & \uparrow\!\dagger{\scriptstyle c} \\ Y & \xrightarrow[Js]{} & X \end{array} \quad \text{in } \mathcal{K}\ell(\mathcal{P}), \tag{19}$$

where $Jq\colon Y \to \mathcal{P}FY$ is given by $(Jq)(y) = \{q(y)\}$ (§5.1).

**Theorem 5.6** (correctness of LTMC progress measure)**.** *Assume the setting of Def. 5.5. In particular, the formula $\varphi$ is translated to an equational system with $m$ variables.*

1. ***(Soundness)*** *Let $(\alpha, Y \xrightarrow{q} FY, r, s)$ be an LTMC progress measure. Let $y \in Y$ be an arbitrary state, $x := s(y)$ (a state of the coalgebra $c$) and $p := r(y)$ (a pre-progress measure). Then there exists an infinitary trace $z \in \mathrm{tr}(c)(x)$ of $x$ such that $p_m(\alpha, \ldots, \alpha) \sqsubseteq [\![\varphi]\!]_\zeta(z)$. Here $[\![\varphi]\!]_\zeta\colon Z \to \Omega$ is from Def. 3.7.*
2. ***(Completeness)*** *Let $x \in X$, and $z \in \mathrm{tr}(c)(x)$ be an infinitary trace from $x$. There is an LTMC progress measure $(\alpha, Y \xrightarrow{q} FY, r, s)$ and some $y \in Y$ such that $s(y) = x$, $\mathsf{beh}(q)(y) = z$ and $p_m(\alpha, \ldots, \alpha) = [\![\varphi]\!]_\zeta(z)$ where $p := r(y)$. Here $\mathsf{beh}(q)$ is the behavior map induced by finality (8).* □

The completeness result in the last theorem is not totally satisfactory, especially from an algorithmic point of view. The question is the size of an LTMC progress measure: in the proof we used $Y \subseteq X \times Z$, but this can be very large—$Z$ is an uncountable set for most common functors $F$. Fortunately we have the following theorem that cuts down the set $Y$ from $X \times Z$ to $X \times \mathsf{pPM}_{\varphi,\alpha}$ (that is potentially much smaller, especially when $\Omega = \mathbf{2}$).

**Theorem 5.7** (small LTMC progress measure)**.** *Assume the setting of Def. 5.5, and let $x \in X$. For any infinitary trace $z \in \mathrm{tr}(c)(x)$, there exists an LTMC progress measure $(\alpha, Y \xrightarrow{q} FY, r, s)$ and some $y \in Y$ such that: $s(y) = x$, and $p_m(\alpha, \dots, \alpha) = [\![\varphi]\!]_\zeta(z)$ where $p := r(y)$. Moreover $(\alpha, Y \xrightarrow{q} FY, r, s)$ can be chosen so that: $Y \subseteq X \times \mathsf{pPM}_{\varphi,\alpha}$; and $r = \pi_2$ and $s = \pi_1$.* □

Our proof of the last theorem comes in a *pumping* flavor. In it, since the relevant set is possibly infinite, we resort to Zorn's lemma.

### 5.4 Decision Procedure

We exploit the previous results and derive a decision procedure for linear-time $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$-model checking. We make the following assumption; its justification is discussed shortly.

**Assumption 5.8.** In what follows we assume that the satisfiability problem of $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$ (against $F$-coalgebras) is decidable.

Moreover we assume the *small model property*: for each satisfiable $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$-formula $\varphi$, we can compute a natural number $N_\varphi \in \omega$ such that there exists an $F$-coalgebra that satisfies $\varphi$ the size of whose state space is no greater than $N_\varphi$. That is: there exists a coalgebra $\varepsilon\colon E \to FE$, its state $e \in E$ and an MC progress measure $Q\colon E \to \mathsf{pPM}_{\varphi,\alpha}$ (Def. 4.3) such that $Q(e)(\alpha, \dots, \alpha) = \mathrm{tt}$ and $|E| \leq N_\varphi$. It is moreover guaranteed by Thm. 4.4.2 that we can take $\alpha := N_\varphi$.

Finally, we assume that $F$ preserves finiteness, that is, $FB$ is finite if $B$ is finite.

Assumption 5.8 is a mild one. For example, [16] shows that the assumption holds when the logic $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$ comes with a one-step complete, contraction-closed and exponentially-tractable set of deductive rules. These conditions hold in well-known modal logics, including (the fixed-point extensions of) the normal modal logic $\mathsf{K}$, and monotone modal logic (Example 3.3). Of more relevance here is the fact that the assumption holds for (coalgebras of) polynomial functors $F$ (with suitable finiteness requirements), which are the ones typically used to specify linear time behavior; modalities and deductive rules for such functors can be modularly derived from their structure, using an approach similar to that of [15], and proving the tractability of the set of rules is straightforward in this case.

It also seems that the framework in §4 can be adapted to satisfiability check and hence to the small-model property. Due to lack of space we do not do so in the current paper and just assume the small model property.

**Theorem 5.9** (linear-time $\mathbf{C}\boldsymbol{\mu}_{\Gamma,\Lambda}$-model checking is decidable)**.** *Assume the setting of Def. 5.5. Assume further that: $\Omega = \mathbf{2}$; and $X$ is a finite set. Then it is decidable whether there exists an infinitary trace $z \in \mathrm{tr}(c)(x)$ such that $[\![\varphi]\!]_\zeta(z) = \mathrm{tt}$.* □

## Acknowledgments

We thank Kenta Cho, Tetsuri Moriya, Shota Nakagawa, Jurriaan Rot, and Natsuki Urabe for useful discussions. I.H. and S.S. are supported by Grants-in-Aid No. 24680001 & 15KT0012, JSPS; C.C. was supported by a Royal Society International Exchanges Grant (IE131642).

## References

[1] S. Almagor, U. Boker, and O. Kupferman. Formalizing and reasoning about quality. In F. V. Fomin, R. Freivalds, M. Z. Kwiatkowska, and D. Peleg, editors, *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part II*, volume 7966 of *Lecture Notes in Computer Science*, pages 15–27. Springer, 2013.

[2] A. Arnold and D. Niwiński. *Rudiments of $\mu$-Calculus*. Studies in Logic and the Foundations of Mathematics. Elsevier, Amsterdam, 2001.

[3] S. Awodey. *Category Theory*. Oxford Logic Guides. Oxford Univ. Press, 2006.

[4] F. Bartels. *On generalised coinduction and probabilistic specification formats. Distributive laws in coalgebraic modelling*. PhD thesis, Free Univ. Amsterdam, 2004.

[5] A. M. Ben-Amram and S. Genaim. Complexity of Bradley-Manna-Sipma lexicographic ranking functions. In D. Kroening and C. S. Pasareanu, editors, *Computer Aided Verification - 27th International Conference, CAV 2015, San Francisco, CA, USA, July 18-24, 2015, Proceedings, Part II*, volume 9207 of *Lecture Notes in Computer Science*, pages 304–321. Springer, 2015.

[6] M. M. Bonsangue and A. Kurz. Duality for logics of transition systems. In V. Sassone, editor, *FoSSaCS*, volume 3441 of *Lect. Notes Comp. Sci.*, pages 455–469. Springer, 2005.

[7] J. Bradfield and C. Stirling. Modal mu-calculi. In P. Blackburn, J. van Benthem, and F. Wolter, editors, *Handbook of Modal Logic*, volume 3 of *Studies in Logic and Practical Reasoning*, chapter 12. Elsevier, 2006.

[8] K. Chatterjee and L. Doyen. Energy parity games. *Theor. Comput. Sci.*, 458:49–60, 2012.

[9] K. Chatterjee, M. Jurdzinski, and T. A. Henzinger. Quantitative stochastic parity games. In J. I. Munro, editor, *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2004, New Orleans, Louisiana, USA, January 11-14, 2004*, pages 121–130. SIAM, 2004.

[10] B. F. Chellas. *Modal Logic*. Cambridge University Press, 1980.

[11] C. Cîrstea. Maximal traces and path-based coalgebraic temporal logics. *Theor. Comput. Sci.*, 412(38):5025–5042, 2011.

[12] C. Cîrstea. Model checking linear coalgebraic temporal logics: An automata-theoretic approach. In A. Corradini, B. Klin, and C. Cîrstea, editors, *Algebra and Coalgebra in Computer Science - 4th International Conference, CALCO 2011, Winchester, UK, August 30 - September 2, 2011. Proceedings*, volume 6859 of *Lecture Notes in Computer Science*, pages 130–144. Springer, 2011.

[13] C. Cîrstea. A coalgebraic approach to linear-time logics. In Muscholl [42], pages 426–440.

[14] C. Cîrstea. Canonical coalgebraic linear time logics. In *Proc. 6th International Conference on Algebra and Coalgebra in Computer Science (CALCO 2015)*, 2015. To appear.

[15] C. Cîrstea and D. Pattinson. Modular construction of complete coalgebraic logics. *Theor. Comput. Sci.*, 388(1-3):83–108, 2007.

[16] C. Cîrstea, C. Kupke, and D. Pattinson. EXPTIME tableaux for the coalgebraic $\mu$-calculus. In E. Grädel and R. Kahle, editors, *CSL*, volume 5771 of *Lecture Notes in Computer Science*, pages 179–193. Springer, 2009.

[17] C. Cîrstea, A. Kurz, D. Pattinson, L. Schröder, and Y. Venema. Modal logics are coalgebraic. *Comput. J.*, 54(1):31–41, 2011.

[18] R. Cleaveland and B. Steffen. A linear-time model-checking algorithm for the alternation-free modal mu-calculus. *Formal Methods in System Design*, 2(2):121–147, 1993.

[19] R. Cleaveland, M. Klein, and B. Steffen. Faster model checking for the modal mu-calculus. In G. von Bochmann and D. K. Probst, editors, *Computer Aided Verification, Fourth International Workshop, CAV '92, Montreal, Canada, June 29 - July 1, 1992, Proceedings*, volume 663 of *Lecture Notes in Computer Science*, pages 410–422. Springer, 1992.

[20] P. Cousot and R. Cousot. Constructive versions of Tarski's fixed point theorems. *Pacific Journal of Mathematics*, 82(1):43–57, 1979.

[21] L. de Alfaro, T. A. Henzinger, and R. Majumdar. Discounting the future in systems theory. In J. C. M. Baeten, J. K. Lenstra, J. Parrow, and G. J. Woeginger, editors, *Automata, Languages and Programming, 30th International Colloquium, ICALP 2003, Eindhoven, The Nether-*

*lands, June 30 - July 4, 2003. Proceedings*, volume 2719 of *Lect. Notes Comp. Sci.*, pages 1022–1037. Springer, 2003.

[22] K. Etessami, T. Wilke, and R. A. Schuller. Fair simulation relations, parity games, and state space reduction for büchi automata. *SIAM J. Comput.*, 34(5):1159–1175, 2005.

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

[24] G. Fontaine, R. A. Leal, and Y. Venema. Automata for coalgebras: An approach using predicate liftings. In S. Abramsky, C. Gavoille, C. Kirchner, F. Meyer auf der Heide, and P. G. Spirakis, editors, *ICALP (2)*, volume 6199 of *Lecture Notes in Computer Science*, pages 381–392. Springer, 2010.

[25] P. Garg, C. Löding, P. Madhusudan, and D. Neider. ICE: A robust framework for learning invariants. In A. Biere and R. Bloem, editors, *Computer Aided Verification - 26th International Conference, CAV 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 18-22, 2014. Proceedings*, volume 8559 of *Lecture Notes in Computer Science*, pages 69–87. Springer, 2014.

[26] I. Hasuo, B. Jacobs, and A. Sokolova. Generic trace semantics via coinduction. *Logical Methods in Comp. Sci.*, 3(4:11), 2007.

[27] I. Hasuo, S. Shimizu, and C. Cîrstea. Lattice-theoretic progress measures and coalgebraic model checking. Extended version with appendices, available at `http://arxiv.org/`, 2015.

[28] C. Hermida and B. Jacobs. Structural induction and coinduction in a fibrational setting. *Inf. & Comp.*, 145:107–152, 1998.

[29] B. Jacobs. The temporal logic of coalgebras via Galois algebras. *Math. Struct. in Comp. Sci.*, 12:875–903, 2002.

[30] B. Jacobs. Trace semantics for coalgebras. In J. Adámek and S. Milius, editors, *Coalgebraic Methods in Computer Science*, volume 106 of *Elect. Notes in Theor. Comp. Sci.* Elsevier, Amsterdam, 2004.

[31] B. Jacobs. Introduction to coalgebra. Towards mathematics of states and observations. Draft of a book (ver. 2.0), available online, 2012.

[32] B. Jacobs, A. Silva, and A. Sokolova. Trace semantics via determinization. *J. Comput. Syst. Sci.*, 81(5):859–879, 2015.

[33] M. Jurdzinski. Small progress measures for solving parity games. In H. Reichel and S. Tison, editors, *STACS*, volume 1770 of *Lecture Notes in Computer Science*, pages 290–301. Springer, 2000.

[34] H. Kerstan and B. König. Coalgebraic trace semantics for continuous probabilistic transition systems. *Logical Methods in Computer Science*, 9(4), 2013.

[35] N. Klarlund and D. Kozen. Rabin measures and their applications to fairness and automata theory. In *Proceedings of the Sixth Annual Symposium on Logic in Computer Science (LICS '91), Amsterdam, The Netherlands, July 15-18, 1991*, pages 256–265. IEEE Computer Society, 1991.

[36] B. Klin. Coalgebraic modal logic beyond **Sets**. In *MFPS XXIII*, volume 173, pages 177–201. Elsevier, Amsterdam, 2007.

[37] N. Kobayashi and C.-H. L. Ong. A type system equivalent to the modal mu-calculus model checking of higher-order recursion schemes. In *LICS*, pages 179–188. IEEE Computer Society, 2009.

[38] D. Kozen. Results on the propositional $\mu$-calculus. *Theor. Comp. Sci.*, 27(3):333–354, 1983.

[39] S. Mac Lane. *Categories for the Working Mathematician*. Springer, Berlin, 2nd edition, 1998.

[40] M. Mio. Upper-expectation bisimilarity and Łukasiewicz $\mu$-calculus. In Muscholl [42], pages 335–350.

[41] L. S. Moss. Coalgebraic logic. *Ann. Pure & Appl. Logic*, 96(1-3): 277–317, 1999. *Erratum* in *Ann. Pure & Appl. Logic*, 99(1-3):241–259, 1999.

[42] A. Muscholl, editor. *Foundations of Software Science and Computation Structures - 17th International Conference, FOSSACS 2014, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, Grenoble, France, April 5-13, 2014, Proceedings*, volume 8412 of *Lecture Notes in Computer Science*, 2014. Springer.

[43] S. Nakagawa and I. Hasuo. Near-optimal scheduling for LTL with future discounting. In *Trustworthy Global Computing - 10th International Symposium, TGC 2015*, Lecture Notes in Computer Science. Springer, 2015. to appear.

[44] C. L. Ong. On model-checking trees generated by higher-order recursion schemes. In *21th IEEE Symposium on Logic in Computer Science (LICS 2006), 12-15 August 2006, Seattle, WA, USA, Proceedings*, pages 81–90. IEEE Computer Society, 2006.

[45] D. Pattinson. Coalgebraic modal logic: soundness, completeness and decidability of local consequence. *Theor. Comput. Sci.*, 309(1-3):177–193, 2003.

[46] D. Pattinson. The logic of exact covers: Completeness and uniform interpolation. In *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013*, pages 418–427. IEEE Computer Society, 2013.

[47] D. Pattinson and L. Schröder. Admissibility of cut in coalgebraic logics. *Electr. Notes Theor. Comput. Sci.*, 203(5):221–241, 2008.

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

[49] A. Pnueli. The temporal logic of programs. In *Found. Comp. Sci.*, pages 46–57. IEEE, 1977.

[50] J. Power and D. Turi. A coalgebraic foundation for linear time semantics. In *Category Theory and Computer Science*, volume 29 of *Elect. Notes in Theor. Comp. Sci.* Elsevier, Amsterdam, 1999.

[51] J. J. M. M. Rutten. Universal coalgebra: a theory of systems. *Theor. Comp. Sci.*, 249:3–80, 2000.

[52] L. Schröder and D. Pattinson. Pspace bounds for rank-1 modal logics. *ACM Trans. Comput. Log.*, 10(2), 2009.

[53] L. Schröder and Y. Venema. Flat coalgebraic fixed point logics. In P. Gastin and F. Laroussinie, editors, *CONCUR*, volume 6269 of *Lect. Notes Comp. Sci.*, pages 524–538. Springer, 2010.

[54] T. Tsukada and C. L. Ong. Compositional higher-order model checking via $\omega$-regular games over böhm trees. In T. A. Henzinger and D. Miller, editors, *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*, pages 78:1–78:10. ACM, 2014.

[55] N. Urabe and I. Hasuo. Generic forward and backward simulations III: quantitative simulations by matrices. In P. Baldan and D. Gorla, editors, *CONCUR 2014 - Concurrency Theory - 25th International Conference, CONCUR 2014, Rome, Italy, September 2-5, 2014. Proceedings*, volume 8704 of *Lecture Notes in Computer Science*, pages 451–466. Springer, 2014. Best paper award.

[56] N. Urabe and I. Hasuo. Coalgebraic infinite traces and Kleisli simulations. In *Proc. 6th International Conference on Algebra and Coalgebra in Computer Science (CALCO 2015)*, Leibniz International Proceedings in Informatics, 2015. To appear; extended version available at `http://arxiv.org/abs/1505.06819`.

[57] M. Y. Vardi. An automata-theoretic approach to linear temporal logic. In F. Moller and G. M. Birtwistle, editors, *Banff Higher Order Workshop*, volume 1043 of *Lecture Notes in Computer Science*, pages 238–266. Springer, 1995.

[58] Y. Venema. Automata and fixed point logic: A coalgebraic perspective. *Inf. Comput.*, 204(4):637–678, 2006.

[59] T. Wilke. Alternating tree automata, parity games, and modal $\mu$-calculus. *Bull. Belg. Math. Soc. Simon Stevin*, 8(2):359–391, 2001.