Coalgebraic Trace Semantics for Büchi and Parity Automata*

Natsuki Urabe†[1], Shunsuke Shimizu[2], and Ichiro Hasuo[3]

1 **Department of Computer Science, The University of Tokyo, Japan and JSPS Research Fellow**
`urabenatsuki@is.s.u-tokyo.ac.jp`

2 **Department of Computer Science, The University of Tokyo, Japan**
`shunsuke@is.s.u-tokyo.ac.jp`

3 **Department of Computer Science, The University of Tokyo, Japan**
`ichiro@is.s.u-tokyo.ac.jp`

Abstract

Despite its success in producing numerous general results on state-based dynamics, the theory of *coalgebra* has struggled to accommodate the *Büchi acceptance condition*—a basic notion in the theory of automata for infinite words or trees. In this paper we present a clean answer to the question that builds on the "maximality" characterization of infinite traces (by Jacobs and Cîrstea): the accepted language of a Büchi automaton is characterized by two commuting diagrams, one for a *least* homomorphism and the other for a *greatest*, much like in a system of (least and greatest) fixed-point equations. This characterization works uniformly for the nondeterministic branching and the probabilistic one; and for words and trees alike. We present our results in terms of the *parity* acceptance condition that generalizes Büchi's.

1998 ACM Subject Classification F.1.1 Models of Computation

Keywords and phrases coalgebra, Büchi/parity/probabilistic/tree automaton

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2016.24

1 Introduction

Büchi Automata. *Automata* are central to theoretical computer science. Besides their significance in formal language theory and as models of computation, many *formal verification* techniques rely on them, exploiting their balance between expressivity and tractable complexity of operations on them. See e.g. [30, 12]. Many current problems in verification are about *nonterminating* systems (like servers); for their analyses, naturally, automata that classify *infinite* objects—such as infinite words and infinite trees—are employed.

The *Büchi acceptance condition* is the simplest nontrivial acceptance condition for automata for infinite objects. Instead of requiring finally reaching an accepting state ◎—which makes little sense for infinite words/trees—it requires accepting states visited *infinitely often*. This simple condition, too, has proved both expressive and computationally tractable: for the word case the Büchi condition can express any ω-regular properties; and the emptiness problem for Büchi automata can be solved efficiently by searching for a *lasso* computation.

* The authors are supported by Grants-in-Aid No. 24680001 & 15KT0012, JSPS.
† N.U. is supported by Grant-in-Aid for JSPS Fellows.

Coalgebras. Studies of automata and state-based transition systems in general have been shed a fresh *categorical* light in 1990's, by the theory of *coalgebra*. Its simple modeling of state-based dynamics—as a *coalgebra*, i.e. an arrow $c\colon X \to FX$ in a category $\mathbb{C}$—has produced numerous results that capture mathematical essences and provide general techniques. Among its basic results are: *behavior-preserving maps* as homomorphisms; a *final coalgebra* as a fully abstract domain of behaviors; *coinduction* (by finality) as definition and proof principles; a general span-based definition of *bisimulation*; etc. See e.g. [16, 22]. More advanced results are on: coalgebraic modal logic (see e.g. [7]); process algebras and congruence formats (see e.g. [18]); generalization of Kleene's theorem (see e.g. [24]); etc.

Büchi Automata, Coalgebraically. In the coalgebra community, however, two important phenomena in automata and/or concurrency have been known to be hard to model—many previous attempts have seen only limited success. One is *internal (τ-)transitions* and *weak (bi)similarity*; see e.g. recent [11]. The other one is the Büchi acceptance condition.

$$\begin{tikzcd} FX \arrow[r, "Ff"] & FY \\ X \arrow[u, "c"] \arrow[r, "f"'] & Y \arrow[u, "d"'] \end{tikzcd}$$

Here is a (sketchy) explanation why these two phenomena should be hard to model coalgebraically. The theory of coalgebra is centered around *homomorphisms* as behavior-preserving maps; see the diagram on the right. Deep rooted in it is the idea of *local matching* between one-step transitions in c and those in d. This is what fails in the two phenomena: in weak bisimilarity a one-step transition in c is matched by a possibly multi-step transition in d; and the Büchi acceptance condition—stipulating that accepting states are visited *infinitely often*, in the long run—is utterly *nonlocal*.

There have been some works that study Büchi acceptance conditions (or more general *parity* or *Muller* conditions) in coalgebraic settings. One is [5], where they rely on the lasso characterization of nonemptiness and use $\mathbf{Sets}^2$ as a base category. Another line is on *coalgebra automata* (see e.g. [31]), where however Büchi/parity/Muller acceptance conditions reside outside the realm of coalgebras.[1] Inspired by these works, and also by our work [14] on alternating fixed points and coalgebraic model checking, the current paper introduces a coalgebraic modeling of Büchi and parity automata based on *systems of fixed-point equations*.

Contributions. We present a clean answer to the question of "Büchi automata, coalgebraically," relying on the previous work on coalgebraic infinitary trace semantics [15, 6] and fixed-point equations [14]. Our modeling, hinted in (1), features: 1) accepting states as a *partition* of a state space; and 2) explicit use of μ and ν—for least/greatest fixed points—in diagrams. We state our results for the *parity* condition (that generalizes the Büchi one).

$$\boxed{\begin{tikzcd} \overline{F}X \arrow[r, "\shortmid" marking] \arrow[d, phantom, "=_\nu"] & \overline{F}Z \\ X \arrow[u, "c"] \arrow[r, "\mathrm{tr}^\infty(c)"', "\shortmid" marking] & Z \arrow[u, "\cong\, J\zeta"'] \end{tikzcd} \quad \text{in a Kleisli category } \mathcal{K}\ell(T)}
\Longrightarrow
\boxed{\begin{tikzcd} \overline{F}X \arrow[r, "\shortmid" marking] & \overline{F}Z \\ X_1 \arrow[u, "c_1"] \arrow[r, "\mathrm{tr}^{\mathrm{p}}(c_1)"', "\shortmid" marking] \arrow[ur, phantom, "=_\mu"] & Z \arrow[u, "\cong\, J\zeta"'] \end{tikzcd} \quad \begin{tikzcd} \overline{F}X \arrow[r, "\shortmid" marking] & \overline{F}Z \\ X_2 \arrow[u, "c_2"] \arrow[r, "\mathrm{tr}^{\mathrm{p}}(c_2)"', "\shortmid" marking] \arrow[ur, phantom, "=_\nu"] & Z \arrow[u, "\cong\, J\zeta"'] \end{tikzcd}} \qquad (1)$$

Characterization of languages under no (i.e. the trivial) acceptance condition [15, 6]

Under the Büchi acceptance condition, with $X_1 = \{\bigcirc\text{'s}\}$ and $X_2 = \{\circledcirc\text{'s}\}$

Our framework is generic: its leading examples are *nondeterministic* and (generative) *probabilistic* tree automata, with the Büchi/parity acceptance condition.

Our contributions are: 1) coalgebraic modeling of automata with the Büchi/parity conditions; 2) characterizing their accepted languages by diagrams with μ's and ν's (tr^{p} in (1));

[1] More precisely: a coalgebra automaton is an automaton (with Büchi/parity/Muller acceptance conditions) that *classifies* coalgebras (as generalization of words and trees). A coalgebra automaton itself is *not* described as a coalgebra; nor is its acceptance condition.

and 3) proving that the characterization indeed captures the conventional definitions. The last "sanity-check" proves to be intricate in the probabilistic case, and our proof—relying on previous [6, 23]—identifies the role of *final sequences* [32] in probabilistic processes.

With explicit μ's and ν's—that specify in *which* homomorphism, among *many* that exist, we are interested—we depart from the powerful reasoning principle of *finality* (existence of a unique homomorphism). We believe this is a necessary step forward, for the theory of coalgebra to take up long-standing challenges like the Büchi condition and weak bisimilarity. Our characterization (1)—although it is not so simple as the uniqueness argument by finality—seems useful, too: we have obtained some results on *fair simulation* notions between Büchi automata [28], following the current work.

Organization of the Paper. In Section 2 we provide backgrounds on: the coalgebraic theory of trace in a *Kleisli category* [15, 6] (where we explain the diagram on the left in (1)); and systems of fixed-point equations. In Section 3 we present a coalgebraic modeling of Büchi/parity automata and their languages. Coincidence with the conventional definitions is shown in Section 4 for the nondeterministic setting, and in Section 5 for the probabilistic one.

Most proofs are deferred to the appendices, that are found in [27].

Future Work. Here we are based on the coalgebraic theory of trace and simulation [21, 15, 13, 25]; it has been developed under the *trivial* acceptance condition (any run that does not diverge, i.e. that does not come to a deadend, is accepted). The current paper is about accommodating the Büchi/parity conditions in the *trace* part of the theory; for the *simulation* part we also have exploited the current results to obtain sound *fair simulation* notions for nondeterministic Büchi tree automata and probabilistic Büchi word automata [28].

On the practical side our future work mainly consists of proof methods for *trace/language inclusion*, a problem omnipresent in formal verification. *Simulations*—as one-step, local witnesses for trace inclusion—have been often used as a sound (but not necessarily complete) proof method that is computationally more tractable; with the observations in [28] we are naturally interested in them. Possible directions are: synthesis of *simulation matrices* between finite systems by linear programming, like in [26]; synthesis of simulations by other optimization techniques for program verification (where problem instances are infinite due to the integer type); and simulations as a proof method in interactive theorem proving.

2 Preliminaries

2.1 Coalgebras in a Kleisli Category

We assume some basic category theory, most of which is covered in [16].

The conventional coalgebraic modeling of systems—as a function $X \to FX$—is known to capture *branching-time* semantics (such as bisimilarity) [16, 22]. In contrast accepted languages of Büchi automata (with nondeterministic or probabilistic branching) constitute *linear-time* semantics; see [29] for the so-called *linear time-branching time spectrum.*

For the coalgebraic modeling of such linear-time semantics we follow the "Kleisli modeling" tradition [21, 15, 13]. Here a system is parametrized by a monad T and an endofunctor F on **Sets**: the former represents the *branching type* while the latter represents the *(linear-*

time) transition type; and a system is modeled as a function of the type $X \to TFX$.[2]

A function $X \to TFX$ is nothing but an $\overline{F}$*-coalgebra* $X \nrightarrow \overline{F}X$ in the *Kleisli category* $\mathcal{K}\ell(T)$—where $\overline{F}$ is a suitable lifting of F. This means we can apply the standard coalgebraic machinery to linear-time behaviors, by changing the base category from **Sets** to $\mathcal{K}\ell(T)$.

A monad $T = (T, \eta, \mu)$ on a category $\mathbb{C}$ induces the *Kleisli category* $\mathcal{K}\ell(T)$. The objects of $\mathcal{K}\ell(T)$ are the same as $\mathbb{C}$'s; and for each pair X, Y of objects, the homset $\mathcal{K}\ell(T)(X, Y)$ is given by $\mathbb{C}(X, TY)$. An arrow $f \in \mathcal{K}\ell(T)(X, Y)$—that is $X \to TY$ in $\mathbb{C}$—is called a *Kleisli arrow* and is denoted by $f : X \nrightarrow Y$ for distinction. Given two successive Kleisli arrows $f : X \nrightarrow Y$ and $g : Y \nrightarrow Z$, their *Kleisli composition* is given by $\mu_Z \circ Tg \circ f : X \nrightarrow Z$ (where $\circ$ is composition in $\mathbb{C}$). This composition in $\mathcal{K}\ell(T)$ is denoted by $g \odot f$ for distinction. The *Kleisli inclusion* $J : \mathbb{C} \to \mathcal{K}\ell(T)$ is defined by $J(X) = X$ and $J(f) = \eta_Y \circ f : X \nrightarrow Y$.

In this paper we mainly use two combinations of T and F. The first is the *powerset monad* $\mathcal{P}$ and a polynomial functor on **Sets**; the second is the *(sub-)Giry monad* [10] $\mathcal{G}$ and a polynomial functor on **Meas**, the category of measurable spaces and measurable functions. The *Giry monad* [10] is commonly used for modeling (not necessarily discrete) probabilistic processes. We shall use its "sub" variant; a *subprobability measure* over $(X, \mathfrak{F}_X)$ is a measure μ such that $0 \le \mu(X) \le 1$ (we do not require $\mu(X) = 1$).

▶ **Definition 2.1** ($\mathcal{P}, \mathcal{G}$)**.** The *powerset monad* $\mathcal{P}$ on **Sets** is: $\mathcal{P}X = \{A \subseteq X\}$; $(\mathcal{P}f)(A) = \{f(x) \mid x \in A\}$; its unit is $\eta^{\mathcal{P}}_X(x) = \{x\}$; and its multiplication is $\mu^{\mathcal{P}}_X(M) = \bigcup_{A \in M} A$.

The *sub-Giry monad* is a monad $\mathcal{G} = (\mathcal{G}, \eta^{\mathcal{G}}, \mu^{\mathcal{G}})$ on **Meas** such that $\mathcal{G}(X, \mathfrak{F}_X) = (\mathcal{G}X, \mathfrak{F}_{\mathcal{G}X})$, where $\mathcal{G}X$ is the set of all *subprobability measures* on $(X, \mathfrak{F}_X)$, and $\mathfrak{F}_{\mathcal{G}X}$ is the smallest σ-algebra such that, for each $S \in \mathfrak{F}_X$, the function $\mathrm{ev}_S : \mathcal{G}X \to [0,1]$ defined by $\mathrm{ev}_S(P) = P(S)$ is measurable. Moreover, $\eta^{\mathcal{G}}_{(X,\mathfrak{F}_X)}(x)(S)$ is 1 if $x \in S$ and 0 otherwise (the *Dirac* distribution), and $\mu^{\mathcal{G}}_{(X,\mathfrak{F}_X)}(\Psi)(S) = \int_{\mathcal{G}(X,\mathfrak{F}_X)} \mathrm{ev}_S \, d\Psi$.

▶ **Definition 2.2** (polynomial functors on **Sets** and **Meas**)**.** A *polynomial functor* F on **Sets** is defined by the BNF notation $F ::= \mathrm{id} \mid A \mid F_1 \times F_2 \mid \coprod_{i \in I} F_i$. Here $A \in$ **Sets**.

A *(standard Borel) polynomial functor* F on **Meas** is defined by the BNF notation $F ::= \mathrm{id} \mid (A, \mathfrak{F}_A) \mid F_1 \times F_2 \mid \coprod_{i \in I} F_i$. Here I is countable, and we require each constant $(A, \mathfrak{F}_A) \in$ **Meas** be a *standard Borel space* (see e.g. [9]). The σ-algebra $\mathfrak{F}_{FX}$ associated to FX is defined as usual, with (co)product σ-algebras, etc. F's action on arrows is obvious.

A standard Borel polynomial functor shall often be called simply a *polynomial functor*.

The technical requirement of being standard Borel—meaning that it arises from a *Polish space* [9]—will be used in the probabilistic setting of Section 5; we follow [6, 23] in its use.

There is a well-known correspondence between a polynomial functor and a *ranked alphabet*—a set Σ with an *arity map* $|\_| : \Sigma \to \mathbb{N}$. In this paper a functor F (for the linear-time behavior type) is restricted to be polynomial; this essentially means that we are dealing with systems that generate *trees* over some ranked alphabet (with additional T-branching).

▶ **Definition 2.3** (Tree_Σ)**.** An *(infinitary)* Σ*-tree*, as in the standard definition, is a possibly infinite tree whose nodes are labeled with the ranked alphabet Σ and whose branching degrees are consistent with the arity of labels. The set of Σ-trees is denoted by Tree_Σ.

[2] Another eminent approach to coalgebraic linear-time semantics is the *Eilenberg-Moore* one (see e.g. [17, 1]): notably in the latter a system is expressed as $X \to FTX$. The Eilenberg-Moore approach can be seen as a categorical generalization of *determinization* or the *powerset construction*. It is however not clear how determinization serves our current goal (namely a coalgebraic modeling of the Büchi/parity acceptance conditions).

Table 1 Overview of existing results on coalgebraic trace semantics.

<table>
<tr><th>Semantics</th><th>Finite trace</th><th>Infinitary trace</th></tr>
<tr><td>Coalgebraic modeling</td><td>$$\begin{tikzcd} \overline{F}X \arrow[r, dashed, "\overline{F}(\mathsf{tr}(c))"] & \overline{F}A \\ X \arrow[u, "c"] \arrow[r, dashed, "\mathsf{tr}(c)"'] \arrow[ur, phantom, "="] & A \arrow[u, "\cong"', "J\alpha^{-1}"] \end{tikzcd} \quad \text{final} \qquad (3)$$</td><td>$$\begin{tikzcd} \overline{F}X \arrow[r, "\overline{F}(\mathsf{tr}^{\infty}(c))"] & \overline{F}Z \\ X \arrow[u, "c"] \arrow[r, "\mathsf{tr}^{\infty}(c)"'] \arrow[ur, phantom, "=_{\nu}"] & Z \arrow[u, "\cong"', "J\zeta"] \end{tikzcd} \quad \text{weakly final} \qquad (4)$$</td></tr>
<tr><td></td><td>Finality in $\mathcal{K}\ell(T)$ (Theorem 2.7)</td><td>(Weak finality + maximality) in $\mathcal{K}\ell(T)$ (Theorem 2.8)</td></tr>
</table>

▶ **Lemma 2.4.** *Let Σ be a ranked alphabet, and $F_\Sigma = \coprod_{\sigma\in\Sigma}(\_)^{|\sigma|}$ be the corresponding polynomial functor on* **Sets**. *The set* Tree_Σ *of (infinitary) Σ-trees carries a final F_Σ-coalgebra. The same holds in* **Meas**, *for countable Σ and the corresponding polynomial functor F_Σ.* ◀

We collect some standard notions and notations for such trees in Appendix A in [27].

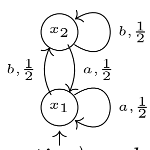
It is known [13, 25] that for $(\mathbb{C}, T) \in \{(\mathbf{Sets}, \mathcal{P}), (\mathbf{Meas}, \mathcal{G})\}$ and polynomial F on $\mathbb{C}$, there is a canonical *distributive law* [20] $\lambda\colon FT \Rightarrow TF$—a natural transformation compatible with T's monad structure. Such λ induces a functor $\overline{F}\colon \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$ that makes the diagram (2) commute.

$$\begin{tikzcd} \mathcal{K}\ell(T) \arrow[r, "\overline{F}"] & \mathcal{K}\ell(T) \\ \mathbb{C} \arrow[u, "J"] \arrow[r, "F"'] & \mathbb{C} \arrow[u, "J"'] \end{tikzcd} \qquad (2)$$

Using this *lifting* $\overline{F}$ of F from $\mathbb{C}$ to $\mathcal{K}\ell(T)$, an arrow $c : X \to TFX$ in $\mathbb{C}$—that is how we model an automaton—can be regarded as an *$\overline{F}$-coalgebra* $c\colon X \nrightarrow \overline{F}X$ in $\mathcal{K}\ell(T)$.

Then the dynamics of $\mathcal{A}$—ignoring its initial and accepting states—is modeled as an $\overline{F}$-coalgebra $c : X \nrightarrow \overline{F}X$ in $\mathcal{K}\ell(\mathcal{P})$ where: $F = \{a, b\} \times (\_)$, $X = \{x_1, x_2\}$ and $c : X \to \mathcal{P}FX$ is the function $c(x_1) = c(x_2) = \{(a, x_1), (b, x_2)\}$. The information on initial and accepting states is redeemed later in

▶ Example 2.5. Let $\mathcal{M}$ be the Markov chain on the right. The dynamics of $\mathcal{M}$ is modeled as an $\overline{F}$-coalgebra $c : X \nrightarrow \overline{F}X$ in $\mathcal{K}\ell(\mathcal{G})$ where: $F = \{a, b\} \times (\_)$, $X = \{x_1, x_2\}$ with the discrete measurable structure, and $c\colon X \to \mathcal{G}FX$ is the (measurable) function defined by $c(x)\big\{(a, x_1)\big\} = c(x)\big\{(b, x_2)\big\} = 1/2$, and $c(x)\big\{(d, x')\big\} = 0$ for the other $(d, x') \in \{a, b\} \times X$.



Later we will equip Markov chains with accepting states and obtain *(generative) probabilistic Büchi automata*. Their probabilistic accepted languages will be our subject of study.

▶ Remark 2.6. Due to the use of the *sub*-Giry monad is that, in $f\colon X \nrightarrow Y$ in $\mathcal{K}\ell(\mathcal{G})$, the probability $f(x)(Y)$ can be smaller than 1. The missing $1 - f(x)(Y)$ is understood as that for *divergence*. In the nondeterministic case $f\colon X \nrightarrow Y$ in $\mathcal{K}\ell(\mathcal{P})$ *diverges* at x if $f(x) = \emptyset$.

This is in contrast with a system coming to halt generating a 0-ary symbol (such as ✓ in (5) later); this is deemed as *successful termination*.

2.2 Coalgebraic Theory of Trace

The above "Kleisli" coalgebraic modeling has produced some general results on: linear-time process semantics (called *trace semantics*); and *simulations* as witnesses of trace inclusion, generalizing the theory in [19]. Here we review the former; it underpins our developments later. A rough summary is in Table 1: typically the results apply to $T \in \{\mathcal{P}, \mathcal{D}, \mathcal{G}\}$—where $\mathcal{D}$ is the *subdistribution monad* on **Sets**, a discrete variant of $\mathcal{G}$—and polynomial F. In what follows we present these previous results in precise terms, sometimes strengthening the assumptions for the sake of presentation. The current paper's goal is to incorporate the Büchi acceptance condition in (the right column of) Table 1.

Firstly, *finite trace semantics*—linear-time behaviors that eventually terminate, such as the accepted languages of *finite* words for NFAs—is captured by finality in $\mathcal{K}\ell(T)$.

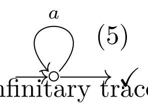
▶ **Theorem 2.7** ([13]). *Let $T \in \{\mathcal{P}, \mathcal{D}\}$ and F be a polynomial functor on* **Sets**. *An initial F-algebra $\alpha\colon FA \overset{\cong}{\Rightarrow} A$ in* **Sets** *yields a final $\overline{F}$-coalgebra in $\mathcal{K}\ell(T)$, as in (3) in Table 1.* ◀

The carrier A of an initial F-algebra in **Sets** is given by finite words/trees (over the alphabet that corresponds to F). The significance of Theorem 2.7 is that: for many examples, the unique homomorphism $\mathsf{tr}(c)$ induced by finality (3) captures the finite trace semantics of the system c. Here the word "finite" means that we collect only behaviors that eventually terminate.

What if we are also interested in *nonterminating* behaviors, like the infinite word $b^\omega = bbb\ldots$ accepted by the automaton in Example 2.5? There is a categorical characterization of such *infinitary trace semantics* too, although proper finality is now lost.

▶ **Theorem 2.8** ([15, 6, 25]). *Let $(\mathbb{C}, T) \in \{(\mathbf{Sets}, \mathcal{P}), (\mathbf{Meas}, \mathcal{G})\}$ and F be a polynomial functor on $\mathbb{C}$. A final F-coalgebra $\zeta\colon Z \overset{\cong}{\Rightarrow} FZ$ in $\mathbb{C}$ gives rise to a weakly final $\overline{F}$-coalgebra in $\mathcal{K}\ell(T)$, as in (4) in Table 1. Moreover, the coalgebra $J\zeta$ additionally admits the* greatest *homomorphism $\mathsf{tr}^\infty(c)$ with respect to the pointwise order $\sqsubseteq$ in the homsets of $\mathcal{K}\ell(T)$ (given by inclusion for $T = \mathcal{P}$, and by pointwise $\leq$ on subprobability measures for $T = \mathcal{G}$). That is: for each homomorphism f from c to $J\zeta$ we have $f \sqsubseteq \mathsf{tr}^\infty(c)$.* ◀

In many examples the greatest homomorphism $\mathsf{tr}^\infty(c)$ captures the infinitary trace semantics of the system c. (Here by *infinitary* we mean *both finite and infinite* behaviors.) For example, for the system (5) where ✓ denotes successful termination, its finite trace semantics is $\{\varepsilon, a, aa, \ldots\}$ whereas its infinitary trace semantics is $\{\varepsilon, a, aa, \ldots\} \cup \{a^\omega\}$. The latter is captured by the diagram (4), with $T = \mathcal{P}$ and $F = \{\checkmark\} + \{a\} \times (\_)$.



(5)

2.3 Equational Systems for Alternating Fixed Points

Nested, alternating greatest and least fixed points—as in a μ-calculus formula $\nu u_2.\mu u_1.(p \wedge u_2) \vee \Box u_1$—are omnipresent in specification and verification. For their relevance to the Büchi/parity acceptance condition one can recall the well-known translation of LTL formulas to Büchi automata and vice versa (see e.g. [30]). To express such fixed points we follow [8, 2] and use *equational systems*—we prefer them to the textual μ-calculus-like presentations.

▶ **Definition 2.9** (equational system). Let $L_1, \ldots, L_n$ be posets. An *equational system E* over $L_1, \ldots, L_n$ is an expression

$$u_1 =_{\eta_1} f_1(u_1, \ldots, u_n) \ , \quad \ldots, \quad u_n =_{\eta_n} f_n(u_1, \ldots, u_n) \tag{6}$$

where: $u_1, \ldots, u_n$ are *variables*, $\eta_1, \ldots, \eta_n \in \{\mu, \nu\}$, and $f_i\colon L_1 \times \cdots \times L_n \to L_i$ is a monotone function. A variable u_j is a *μ-variable* if $\eta_j = \mu$; it is a *ν-variable* if $\eta_j = \nu$.

The *solution* of the equational system E is defined as follows, under the assumption that L_i's have enough supremums and infimums. It proceeds as: 1) we solve the first equation to obtain an interim solution $u_1 = l_1^{(1)}(u_2, \ldots, u_n)$; 2) it is used in the second equation to eliminate u_1 and yield a new equation $u_2 =_{\eta_2} f_2^\ddagger(u_2, \ldots, u_n)$; 3) solving it again gives an interim solution $u_2 = l_2^{(2)}(u_3, \ldots, u_n)$; 4) continuing this way from left to right eventually eliminates all variables and leads to a closed solution $u_n = l_n^{(n)} \in L_n$; and 5) by propagating these closed solutions back from right to left, we obtain closed solutions for all of $u_1, \ldots, u_n$. A precise definition is found in Appendix B in [27].

It is important that the order of equations *matters*: for $(u =_\mu v, v =_\nu u)$ the solution is $u = v = \top$ while for $(v =_\nu u, u =_\mu v)$ the solution is $u = v = \bot$.

Whether a solution is well-defined depends on how "complete" the posets $L_1, \ldots, L_n$ are. It suffices if they are *complete lattices*, in which case every monotone function $L_i \to L_i$ has greatest/least fixed points (the *Knaster-Tarski* theorem). This is used in the nondeterministic setting: note that $\mathcal{P}Y$, hence the homset $\mathcal{K}\ell(\mathcal{P})(X,Y)$, are complete lattices.

▶ **Lemma 2.10.** *The system E (6) has a solution if each L_i is a complete lattice.* ◀

This does not work in the probabilistic case, since the homsets $\mathcal{K}\ell(\mathcal{G})(X,Y) = \mathbf{Meas}(X, \mathcal{G}Y)$ with the pointwise order—on which we consider equational systems—are not complete lattices. For example $\mathcal{G}Y$ lacks the greatest element in general; even if $Y = 1$ (when $\mathcal{G}1 \cong [0,1]$), the homset $\mathcal{K}\ell(\mathcal{G})(X,1)$ can fail to be a complete lattice. See Example B.2 in [27]. Our strategy is: 1) to apply the following *Kleene*-like result to the homset $\mathcal{K}\ell(\mathcal{G})(X,1)$; and 2) to "extend" fixed points in $\mathcal{K}\ell(\mathcal{G})(X,1)$ along a final F-sequence. See Section 5.1 later.

▶ **Lemma 2.11.** *The equational system E (6) has a solution if: each L_i is both a pointed ω-cpo and a pointed ω^{op}-cpo; and each f_i is both ω-continuous and ω^{op}-continuous.* ◀

In Appendix B in [27] we have additional lemmas on "homomorphisms" of equational systems and preservation of solutions. They play important roles in the proofs of the later results.

3 Coalgebraic Modeling of Parity Automata and Its Trace Semantics

Here we present our modeling of Büchi/parity automata. We shall do so axiomatically with parameters $\mathbb{C}$, T and F—much like in Section 2.1–2.2. Our examples cover: both nondeterministic and probabilistic branching; and automata for trees (hence words as a special case).

▶ Assumptions 3.1. In what follows a monad T and an endofunctor F, both on $\mathbb{C}$, satisfy:
- The base category $\mathbb{C}$ has a final object 1 and finite coproducts.
- The functor F has a final coalgebra $\zeta \colon Z \to FZ$ in $\mathbb{C}$.
- There is a *distributive law* $\lambda \colon FT \Rightarrow TF$ [20], hence $F : \mathbb{C} \to \mathbb{C}$ is lifted to $\overline{F} \colon \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$. See (2).
- For each $X, Y \in \mathcal{K}\ell(T)$, the homset $\mathcal{K}\ell(T)(X,Y)$ carries an order $\sqsubseteq_{X,Y}$ (or simply $\sqsubseteq$).
- Kleisli composition $\odot$ and cotupling $[\_,\_]$ are monotone with respect to the order $\sqsubseteq$. The latter gives rise to an order isomorphism $\mathcal{K}\ell(T)(X_1 + X_2, Y) \cong \mathcal{K}\ell(T)(X_1, Y) \times \mathcal{K}\ell(T)(X_2, Y)$, where $+$ is inherited along a left adjoint $J \colon \mathbb{C} \to \mathcal{K}\ell(T)$.
- $\overline{F} : \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$ is *locally monotone*: for $f, g \in \mathcal{K}\ell(T)(X,Y)$, $f \sqsubseteq g$ implies $\overline{F}f \sqsubseteq \overline{F}g$.

▶ Example 3.2. The category **Sets**, the powerset monad $\mathcal{P}$ (Definition 2.1) and a polynomial functor F on **Sets** (Definition 2.2) satisfy Assumption 3.1. Here for $X, Y \in \mathcal{K}\ell(\mathcal{P})$, an order $\sqsubseteq_{X,Y}$ is defined by: $f \sqsubseteq g$ if $f(x) \subseteq g(x)$ for each $x \in X$.

▶ Example 3.3. The category **Meas**, the sub-Giry monad $\mathcal{G}$ (Definition 2.1) and a polynomial functor F on **Meas** (Definition 2.2) satisfy Assumption 3.1. For $X, Y \in \mathcal{K}\ell(\mathcal{G})$, a natural order $\sqsubseteq_{(X,\mathfrak{F}_X),(Y,\mathfrak{F}_Y)}$ is defined by: $f \sqsubseteq g$ iff $f(x)(A) \leq g(x)(A)$ (in $[0,1]$) for each $x \in X$ and $A \in \mathfrak{F}_Y$.

3.1 Coalgebraic Modeling of Büchi/Parity Automata

The Büchi and parity acceptance conditions have been big challenges to the coalgebra community, because of their *nonlocal* and *asymptotic* nature (see Section 1). One possible modeling is to take the distinction between ◯ vs. ◎—or different priorities in the parity case—as *state labels*. This is much like in the established coalgebraic modeling of deterministic automata as $2 \times (\_)^{\Sigma}$-coalgebras (see e.g. [22, 16]). Here the set 2 tells if a state is accepting or not.

A key to our current modeling, however, is that accepting states should rather be specified by a *partition* $X = X_1 + X_2$ of a state space, with $X_1 = \{$◯'s$\}$ and $X_2 = \{$◎'s$\}$. This idea smoothly generalizes to *parity* conditions, too, by $X_i = \{\text{states of priority } i\}$. Equipping such partitions to coalgebras (with explicit initial states, as in Section 2.2) leads to the following.

Henceforth we state results for the parity condition, with Büchi being a special case.

▶ **Definition 3.4** (parity (T, F)-system)**.** A *parity* (T, F)*-system* is given by a triple $\mathcal{X} = \big((X_1, \ldots, X_n),\ c\colon X \nrightarrow \overline{F}X,\ s\colon 1 \nrightarrow X\big)$ where n is a positive integer, and:

- $(X_1, \ldots, X_n)$ is an n-tuple of objects in $\mathbb{C}$ for *states* (with their *priorities*), and we define $X = X_1 + \cdots + X_n$ (a coproduct in $\mathbb{C}$);
- $c\colon X \nrightarrow \overline{F}X$ is an arrow in $\mathcal{K}\ell(T)$ for *dynamics*; and
- $s\colon 1 \nrightarrow X$ is an arrow in $\mathcal{K}\ell(T)$ for *initial states*.

For each $i \in [1, n]$ we define $c_i\colon X_i \nrightarrow \overline{F}X$ to be the restriction $c \circ \kappa_i\colon X_i \nrightarrow \overline{F}X$ along the coprojection $\kappa_i\colon X_i \hookrightarrow X$, in case the maximum priority is $n = 2$, a parity (T, F)-system is referred to as a *Büchi* (T, F)*-system*.

3.2 Coalgebraic Trace Semantics under the Parity Acceptance Condition

On top of the modeling in Definition 3.4 we characterize *accepted languages*—henceforth referred to as *trace semantics*—of parity (T, F)-systems. We use systems of fixed-point equations; this naturally extends the previous characterization of infinitary traces (i.e. under the trivial acceptance conditions) by maximality (Theorem 2.8; see also (1)).

▶ **Definition 3.5** (trace semantics of parity (T, F)-systems)**.** Let $\mathcal{X} = \big((X_1, \ldots, X_n), c, s\big)$ be a parity (T, F)-system. It induces the following equational system $E_{\mathcal{X}}$, where $\zeta\colon Z \xrightarrow{\cong} FZ$ is a final coalgebra in $\mathbb{C}$ (see Assumption 3.1). The variable u_i ranges over the poset $\mathcal{K}\ell(T)(X_i, Z)$.

$$E_{\mathcal{X}} \;\coloneqq\; \left[\begin{array}{llll} u_1 & =_{\mu} & (J\zeta)^{-1} \odot \overline{F}[u_1, \ldots, u_n] \odot c_1 & \in \mathcal{K}\ell(T)(X_1, Z) \\ u_2 & =_{\nu} & (J\zeta)^{-1} \odot \overline{F}[u_1, \ldots, u_n] \odot c_2 & \in \mathcal{K}\ell(T)(X_2, Z) \\ & \vdots & & \\ u_n & =_{\eta_n} & (J\zeta)^{-1} \odot \overline{F}[u_1, \ldots, u_n] \odot c_n & \in \mathcal{K}\ell(T)(X_n, Z) \end{array}\right]$$

Here $\eta_i = \mu$ if i is odd and $\eta_i = \nu$ if i is even. The functions in the equations are seen to be monotone, thanks to the monotonicity assumptions on cotupling, $\overline{F}$ and $\odot$ (Assumption 3.1).

We say that (T, F) constitutes a *parity trace situation*, if $E_{\mathcal{X}}$ has a solution for any parity (T, F)-system $\mathcal{X}$, denoted by $\mathsf{tr}^{\mathrm{p}}_1(\mathcal{X})\colon X_1 \nrightarrow Z, \ldots, \mathsf{tr}^{\mathrm{p}}_n(\mathcal{X})\colon X_n \nrightarrow Z$. The composite

$$\mathsf{tr}^{\mathrm{p}}(\mathcal{X}) \;\coloneqq\; \Big(1 \xrightarrow{s} X = X_1 + X_2 + \cdots + X_n \xrightarrow{[\mathsf{tr}^{\mathrm{p}}_1(\mathcal{X}), \mathsf{tr}^{\mathrm{p}}_2(\mathcal{X}), \ldots, \mathsf{tr}^{\mathrm{p}}_n(\mathcal{X})]} Z\Big)$$

is called the *trace semantics* of the parity (T, F)-system $\mathcal{X}$.

If $\mathcal{X}$ is a Büchi (T, F)-system, the equational system $E_{\mathcal{X}}$—with their solutions $\mathsf{tr}^{\mathrm{p}}_1(\mathcal{X})$ and $\mathsf{tr}^{\mathrm{p}}_2(\mathcal{X})$ in place—can be expressed as the following diagrams (with explicit μ and ν). See (1).

$$\begin{tikzcd} FX \arrow[r, "\overline{F}[\mathrm{tr^p}(c_1),\mathrm{tr^p}(c_2)]"] \arrow[d, phantom, ""] & FZ \\ X_1 \arrow[u, "c_1"] \arrow[r, "\mathrm{tr^p}(c_1)"'] \arrow[ur, phantom, "=_\mu"] & Z \arrow[u, "\cong J\zeta"'] \end{tikzcd} \qquad \begin{tikzcd} FX \arrow[r, "\overline{F}[\mathrm{tr^p}(c_1),\mathrm{tr^p}(c_2)]"] & FZ \\ X_2 \arrow[u, "c_2"] \arrow[r, "\mathrm{tr^p}(c_2)"'] \arrow[ur, phantom, "=_\nu"] & Z \arrow[u, "\cong J\zeta"'] \end{tikzcd} \tag{7}$$

4 Coincidence with the Conventional Definition: Nondeterministic

The rest of the paper is devoted to showing that our coalgebraic characterization (Definition 3.5) indeed captures the conventional definition of accepted languages. In this section we study the nondeterministic case; we let $\mathbb{C} = \mathbf{Sets}$, $T = \mathcal{P}$, and F be a polynomial functor.

We first have to check that Definition 3.5 makes sense. Existence of enough fixed points is obvious because $\mathcal{K}\ell(\mathcal{P})(X_i, Z)$ is a complete lattice (Lemma 2.10). See also Example 3.2.

▶ **Theorem 4.1.** *$T = \mathcal{P}$ and a polynomial F constitute a parity trace situation (Definition 3.5).* ◀

Here is the conventional definition of automata [12].

▶ **Definition 4.2** (NPTA). A *nondeterministic parity tree automaton* (NPTA) is a quadruple

$$\mathcal{X} \;=\; \big(\,(X_1,\dots,X_n),\, \Sigma,\, \delta\colon X \to \mathcal{P}\big(\textstyle\coprod_{\sigma\in\Sigma} X^{|\sigma|}\big),\, s \in \mathcal{P}X\,\big)\,,$$

where $X = X_1 + \cdots + X_n$, each X_i is the set of *states* with the *priority* i, Σ is a ranked alphabet (with the arity map $|\_|\colon \Sigma \to \mathbb{N}$), δ is a *transition function* and s is the set of *initial states*.

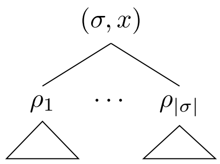
The accepted language of an NPTA $\mathcal{X}$ is conventionally defined in the following way. Here we are sketchy due to the lack of space; precise definitions are in Appendix A in [27].

A (possibly infinite) $(\Sigma \times X)$-labeled tree ρ is a *run* of an NPTA $\mathcal{X} = (\vec{X}, \Sigma, \delta, s)$ if: for each node with a label (σ, x), it has $|\sigma|$ children and we have $\big(\sigma, (x_1, \dots, x_{|\sigma|})\big) \in \delta(x)$ where $x_1, \dots, x_{|\sigma|}$ are the X-labels of its children. For a pedagogical reason we do not require the root X-label to be an initial state. A run ρ of an NPTA $\mathcal{X}$ is *accepting* if any infinite branch π of the tree ρ satisfies the parity acceptance condition (i.e. $\max\{i \mid \pi$ visits states in X_i infinitely often$\}$ is even). The sets of runs and accepting runs of $\mathcal{X}$ are denoted by $\mathrm{Run}_{\mathcal{X}}$ and $\mathrm{AccRun}_{\mathcal{X}}$, respectively.

The function $\mathrm{rt}\colon \mathrm{Run}_{\mathcal{X}} \to X$ is defined to return the root X-label of a run. For each $X' \subseteq X$, we define $\mathrm{Run}_{\mathcal{X},X'}$ by $\{\rho \in \mathrm{Run}_{\mathcal{X}} \mid \mathrm{rt}(\rho) \in X'\}$; the set $\mathrm{AccRun}_{\mathcal{X},X'}$ is similar. The map $\mathrm{DelSt}\colon \mathrm{Run}_{\mathcal{X}} \to \mathrm{Tree}_\Sigma$ takes a run, removes all X-labels and returns a Σ-tree.

▶ **Definition 4.3** ($\mathrm{Lang}(\mathcal{X})$ for NPTAs). Let $\mathcal{X}$ be an NPTA. Its *accepted language* $\mathrm{Lang}(\mathcal{X})$ is defined by $\mathrm{DelSt}(\mathrm{AccRun}_{\mathcal{X},s})$.

The following coincidence result for the nondeterministic setting is fairly straightforward. A key is the fact that accepting runs are characterized—among all possible runs—using an equational system that is parallel to the one in Definition 3.5.



(8)

▶ **Lemma 4.4.** *Let $\mathcal{X} = (\vec{X}, \Sigma, \delta, s)$ be an NPTA, and $l_1^{\mathrm{sol}}, \dots, l_n^{\mathrm{sol}}$ be the solution of the following equational system, whose variables $u_1, \dots, u_n$ range over $\mathcal{P}(\mathrm{Run}_{\mathcal{X}})$.*

$$u_1 \;=_{\eta_1}\; \Diamond_{\mathcal{X}}(u_1 \cup \cdots \cup u_n) \cap \mathrm{Run}_{\mathcal{X},X_1}\,, \quad \dots\,, \quad u_n \;=_{\eta_n}\; \Diamond_{\mathcal{X}}(u_1 \cup \cdots \cup u_n) \cap \mathrm{Run}_{\mathcal{X},X_n} \tag{9}$$

Here: $\Diamond_{\mathcal{X}} : \mathcal{P}(\mathrm{Run}_{\mathcal{X}}) \to \mathcal{P}(\mathrm{Run}_{\mathcal{X}})$ *is given by* $\Diamond_{\mathcal{X}} R \coloneqq \big\{\big((\sigma, x), (\rho_1, \dots, \rho_{|\sigma|})\big) \in \mathrm{Run}_{\mathcal{X}} \,\big|\, \sigma \in \Sigma, x \in X, \rho_i \in R\big\}$ *(see the figure (8) above);* $X = X_1 + \cdots + X_n$*; and* η_i *is* μ *(for odd* i*) or* ν *(for even* i*). Then the* i*-th solution* l_i^{sol} *coincides with* $\mathrm{AccRun}_{\mathcal{X},X_i}$. ◀

We shall translate the above result to the characterization of accepted trees (Lemma 4.5). In its proof (that is deferred to the appendix in [27]) Lemma B.3—on homomorphisms of equational systems—plays an important role.

▶ **Lemma 4.5.** *Let* $\mathcal{X} = (\vec{X}, \Sigma, \delta, s)$ *be an NPTA, and let* $l_1'^{\mathrm{sol}}, \dots, l_n'^{\mathrm{sol}}$ *be the solution of the following equational system, where* u_i' *ranges over the complete lattice* $\big(\mathcal{P}(\mathrm{Tree}_\Sigma)\big)^{X_i}$*:*

$$u_1' \;=_{\eta_1}\; \Diamond_\delta([u_1', \dots, u_n']) \restriction X_1 \;, \quad \dots \;, \quad u_n' \;=_{\eta_n}\; \Diamond_\delta([u_1', \dots, u_n']) \restriction X_n \;. \tag{10}$$

Here η_i *is* μ *(for odd* i*) or* ν *(for even* i*);* $(\_) \restriction X_i \colon \big(\mathcal{P}(\mathrm{Tree}_\Sigma)\big)^X \to \big(\mathcal{P}(\mathrm{Tree}_\Sigma)\big)^{X_i}$ *denotes domain restriction; and the function* $\Diamond_\delta \colon \big(\mathcal{P}(\mathrm{Tree}_\Sigma)\big)^X \to \big(\mathcal{P}(\mathrm{Tree}_\Sigma)\big)^X$ *is given by*

$$(\Diamond_\delta T)(x) \;\coloneqq\; \big\{\big(\sigma, (\tau_1, \dots, \tau_{|\sigma|})\big) \,\big|\, \big(\sigma, (x_1, \dots, x_{|\sigma|})\big) \in \delta(x), \tau_i \in T(x_i)\big\} \;.$$

Then we have a coincidence $l_i'^{\mathrm{sol}} = \mathrm{DelSt}'(\mathrm{AccRun}_{\mathcal{X},X_i})$*, where the function* $\mathrm{DelSt}' \colon \mathcal{P}(\mathrm{Run}_{\mathcal{X}}) \to (\mathcal{P}(\mathrm{Tree}_\Sigma))^X$ *is given by* $\mathrm{DelSt}'(R)(x) \coloneqq \mathrm{DelSt}(\{\rho \in R \mid \mathrm{rt}(\rho) = x\})$*. Recall that* rt *returns a run's root* X*-label.* ◀

▶ **Theorem 4.6** (coincidence, in the nondeterministic setting)**.** *Let* $\mathcal{X} = ((X_1, \dots, X_n), \Sigma, \delta, s)$ *be an NPTA, and* $F_\Sigma = \coprod_{\sigma \in \Sigma} (\_)^{|\sigma|}$ *be the polynomial functor on* **Sets** *that corresponds to* Σ*. Then* $\mathcal{X}$ *is identified with a parity* $(\mathcal{P}, F_\Sigma)$*-system; moreover* $\mathrm{Lang}(\mathcal{X})$ *(in the conventional sense of Definition 4.3) coincides with the coalgebraic trace semantics* $\mathsf{tr}^{\mathrm{p}}(\mathcal{X})$ *(Definition 3.5). Note here that* Tree_Σ *carries a final* F_Σ*-coalgebra (Lemma 2.4).*

Proof. We identify $\mathcal{X}$ with the $(\mathcal{P}, F_\Sigma)$-system $\big((X_1, \dots, X_n), \delta \colon X \nrightarrow \overline{F_\Sigma} X, s \colon 1 \nrightarrow X\big)$, and let $1 = \{\bullet\}$. The equational system $E_{\mathcal{X}}$ in Definition 3.5 is easily seen to coincide with (9) in Lemma 4.5. The claim is then shown as follows, exploiting the last coincidence.

$$\begin{aligned}\mathsf{tr}^{\mathrm{p}}(\mathcal{X}) &= [\mathsf{tr}^{\mathrm{p}}_1(\mathcal{X}), \dots, \mathsf{tr}^{\mathrm{p}}_n(\mathcal{X})] \odot s(\bullet) \qquad \text{by Definition 3.5}\\ &= [\mathrm{DelSt}'(\mathrm{AccRun}_{\mathcal{X},X_1}), \dots, \mathrm{DelSt}'(\mathrm{AccRun}_{\mathcal{X},X_n})](s)\\ &= \mathrm{DelSt}(\mathrm{AccRun}_{\mathcal{X},s}) = \mathrm{Lang}(\mathcal{X}) \qquad \text{by Definition 4.3.}\end{aligned}$$ ◀

5 Coincidence with the Conventional Definition: Probabilistic

In the probabilistic setting the coincidence result is much more intricate. Even the well-definedness of parity trace semantics (Definition 3.5) is nontrivial: the posets $\mathcal{K}\ell(\mathcal{G})(X_i, Z)$ of our interest are not complete lattices, and they even lack the greatest element $\top$. Therefore neither of Lemmas 2.10–2.11 ensures a solution of $E_{\mathcal{X}}$ in Definition 3.5. As we hinted in Section 2.3 our strategy is: 1) to apply the Lemma 2.11 to the homset $\mathcal{K}\ell(\mathcal{G})(X, 1)$; and 2) to "extend" fixed points in $\mathcal{K}\ell(\mathcal{G})(X, 1)$ along a final F-sequence. Implicit in the proof details below, in fact, is a correspondence between: abstract categorical arguments along a final sequence; and concrete operational intuitions on probabilistic parity automata.

In this section we let $\mathbb{C} = \mathbf{Meas}$, $T = \mathcal{G}$ (Definition 2.1), and F be a polynomial functor.

▶ Remark 5.1. The class of probabilistic systems of our interest are *generative* (as opposed to *reactive*) ones. Their difference is eminent in the types of transition functions:

$$\begin{array}{llllll} X \longrightarrow \mathcal{G}(A \times X) & \text{(word)} & X \longrightarrow \mathcal{G}(\coprod_{\sigma \in \Sigma} X^{|\sigma|}) & \text{(tree)} & \text{for generative;}\\ X \longrightarrow (\mathcal{G}X)^A & \text{(word)} & X \longrightarrow \prod_{\sigma \in \Sigma} \mathcal{G}(X^{|\sigma|}) & \text{(tree)} & \text{for reactive.}\end{array}$$

A generative system (probabilistically) chooses which character to *generate*; while a reactive one *receives* a character from the environment. Reactive variants of probabilistic tree automata have been studied e.g. in [4], following earlier works like [3] on reactive probabilistic word automata. Further discussion is in Appendix C.1 in [27].

5.1 Trace Semantics of Parity $(\mathcal{G}, F)$-Systems is Well-Defined

In the following key lemma—that is inspired by the observations in [6, 23, 25]—a typical usage is for $X_A = X_1 + \cdots + X_i$ and $X_B = X_{i+1} + \cdots + X_n$.

▶ **Lemma 5.2.** *Let $\mathcal{X} = ((X_1, \ldots, X_n), s, c)$ be a parity $(\mathcal{G}, F)$-system, and suppose that we are given a partition $X = X_A + X_B$ of $X := X_1 + \cdots + X_n$.*

We define a function $\Gamma\colon \mathcal{K}\ell(\mathcal{G})(X, Z) \to \mathcal{K}\ell(\mathcal{G})(X, 1)$ by $\Gamma(g) = J!_Z \odot g$, where $!\colon Z \to 1$ is the unique function of the type. Its variants $\Gamma_A : \mathcal{K}\ell(\mathcal{G})(X_A, Z) \to \mathcal{K}\ell(\mathcal{G})(X_A, 1)$ and $\Gamma_B : \mathcal{K}\ell(\mathcal{G})(X_B, Z) \to \mathcal{K}\ell(\mathcal{G})(X_B, 1)$ are defined similarly.

For arbitrary $g_B\colon X_B \nrightarrow Z$, we define $\mathfrak{G}^{g_B}$ and $\mathfrak{H}^{g_B}$ as the following sets of "fixed points":

$$\mathfrak{G}^{g_B} := \left\{ \begin{array}{c} g_A\colon \\ X_A \nrightarrow Z \end{array} \;\middle|\; \begin{array}{ccc} \overline{F}X & \xrightarrow{\overline{F}[g_A, g_B]} & \overline{F}Z \\ {\scriptstyle c_A}\uparrow & = & \downarrow{\scriptstyle J\zeta^{-1}} \\ X_A & \xrightarrow[g_A]{} & Z \end{array} \right\} \text{ and } \mathfrak{H}^{g_B} := \left\{ \begin{array}{c} h_A\colon \\ X_A \nrightarrow 1 \end{array} \;\middle|\; \begin{array}{ccc} \overline{F}X & \xrightarrow{\overline{F}[h_A, \Gamma_B(g_B)]} & \overline{F}1 \\ {\scriptstyle c_A}\uparrow & = & \downarrow{\scriptstyle J!_{F1}} \\ X_A & \xrightarrow[h_A]{} & 1 \end{array} \right\} \quad (11)$$

Then Γ_A restricts to a function $\mathfrak{G}^{g_B} \to \mathfrak{H}^{g_B}$. Moreover, the restriction is an order isomorphism, with its inverse denoted by $\Delta^{g_B} : \mathfrak{H}^{g_B} \xrightarrow{\cong} \mathfrak{G}^{g_B}$. ◀

In the proof of the last lemma (deferred to the appendix in [27]), the inverse Δ^{g_B} is defined by "extending" $h_A\colon X_A \nrightarrow 1$ to $X_A \nrightarrow Z$, along the final F-sequence $1 \leftarrow F1 \leftarrow \cdots$ (more precisely: the image of the sequence under the Kleisli inclusion $J\colon \mathbf{Meas} \to \mathcal{K}\ell(\mathcal{G})$).

We are ready to prove existence of $E_{\mathcal{X}}$'s solution (Definition 3.5).

▶ **Lemma 5.3.** *Assume the same setting as in Lemma 5.2. We define $\Phi_{\mathcal{X}}\colon \mathcal{K}\ell(\mathcal{G})(X, Z) \nrightarrow \mathcal{K}\ell(\mathcal{G})(X, Z)$ and $\Psi_{\mathcal{X}}\colon \mathcal{K}\ell(\mathcal{G})(X, 1) \nrightarrow \mathcal{K}\ell(\mathcal{G})(X, 1)$, respectively, by*

$$\Phi_{\mathcal{X}}(g) \;:=\; J\zeta^{-1} \odot \overline{F}g \odot c \quad \text{and} \quad \Psi_{\mathcal{X}}(h) \;:=\; J!_{F1} \odot \overline{F}h \odot c\,;$$

these are like the diagrams in (11), except that the latter are parametrized by X_A, X_B, g_B. Now consider the following equational systems, where: $\eta_i = \mu$ if i is odd and $\eta_i = \nu$ if i is even; u_i ranges over $\mathcal{K}\ell(\mathcal{G})(X_i, Z)$; and u'_i ranges over $\mathcal{K}\ell(\mathcal{G})(X_i, 1)$.

$$E = \left[\begin{array}{c} u_1 =_{\eta_1} \Phi_{\mathcal{X}}([u_1, \ldots, u_n]) \odot \kappa_1 \\ \vdots \\ u_n =_{\eta_n} \Phi_{\mathcal{X}}([u_1, \ldots, u_n]) \odot \kappa_n \end{array} \right] \quad E' = \left[\begin{array}{c} u'_1 =_{\eta_1} \Psi_{\mathcal{X}}([u'_1, \ldots, u'_n]) \odot \kappa_1 \\ \vdots \\ u'_n =_{\eta_n} \Psi_{\mathcal{X}}([u'_1, \ldots, u'_n]) \odot \kappa_n \end{array} \right] \quad (12)$$

We claim that the equational systems have solutions $(l_1^{\mathrm{sol}}, \ldots, l_n^{\mathrm{sol}})$ and $(l_1'^{\mathrm{sol}}, \ldots, l_n'^{\mathrm{sol}})$; and moreover, we have $\Gamma(\mathsf{tr}^{\mathrm{p}}(\mathcal{X})) = \Gamma([l_1^{\mathrm{sol}}, \ldots, l_n^{\mathrm{sol}}]) = [l_1'^{\mathrm{sol}}, \ldots, l_n'^{\mathrm{sol}}]$. ◀

▶ **Theorem 5.4.** *$T = \mathcal{G}$ and a polynomial F constitute a parity trace situation (Definition 3.5).* ◀

▶ Remark 5.5. The process-theoretic interpretation of the isomorphism $\mathfrak{G}^{g_B} \cong \mathfrak{H}^{g_B}$ is interesting. Let us set $X_A = X$ and $X_B = \emptyset$ for simplicity. The greatest element on the left is the *infinitary trace semantics* (i.e. accepted languages under the trivial acceptance condition), as in Theorem 2.8 (cf. Table 1). The corresponding greatest element on the right—a function $h_A\colon X_A \to \mathcal{G}1 \cong [0,1]$—assigns to each state $x \in X$ the probability with which a run from x *does not diverge* (recall from Remark 2.6 that the *sub*-Giry monad $\mathcal{G}$ allows divergence probabilities). The accepted language under the parity condition is in general an element of $\mathfrak{G}^{g_B}$ that is neither greatest nor least; the corresponding element in $\mathfrak{H}^{g_B}$ assigns to each state the probability with which it generates a *accepting* run (over any Σ-tree).

5.2 Probabilistic Parity Tree Automata and Its Languages

▶ **Definition 5.6** (PPTA)**.** A *(generative) probabilistic parity tree automaton (PPTA)* is

$$\mathcal{X} \;=\; \big(\,(X_1,\dots,X_n),\,\Sigma,\,\delta\colon X \to \mathcal{G}\big(\textstyle\coprod_{\sigma\in\Sigma} X^{|\sigma|}\big),\, s \in \mathcal{G}X\,\big) \;\;,$$

where $X = X_1 + \cdots + X_n$, each X_i is a countable set and Σ is a countable ranked alphabet. The subdistribution s over X is for the choice of *initial states.*

In Definition 5.6 the size restrictions on X and Σ are not essential: restricting to discrete σ-algebras, however, makes the following arguments much simpler.

We shall concretely define accepted languages of PPTAs, continuing Section 4 and deferring precise definitions to Appendix A in [27]. This is mostly standard; a reactive variant is found in [4].

▶ **Definition 5.7** (Tree_Σ and $\mathrm{Run}_\mathcal{X}$)**.** Let Σ be a ranked alphabet; Tree_Σ is the set of Σ-trees. A finite $(\Sigma\cup\{*\})$-labeled tree λ, with its branching degrees compatible with the label arities, is called a *partial* Σ*-tree*. Here the new symbol $*$ (“continuation”) is deemed to be 0-ary. The *cylinder set* associated to λ, denoted by $\mathrm{Cyl}_\Sigma(\lambda)$, is the set of (non-partial) Σ-trees that have λ as their prefix (in the sense that a subtree is replaced by $*$). The (smallest) σ-algebra on Tree_Σ generated by the family $\{\mathrm{Cyl}_\Sigma(\lambda) \mid \lambda \text{ is a partial } \Sigma\text{-tree}\}$ will be denoted by $\mathfrak{F}_\Sigma$.

A *run* of a PPTA $\mathcal{X}$ with state space X is a (possibly infinite) $(\Sigma \times X)$-labeled tree whose branching degrees are compatible with the arities of Σ-labels. $\mathrm{Run}_\mathcal{X}$ denotes the set of runs. The measurable structure $\mathfrak{F}_\mathcal{X}$ on $\mathrm{Run}_\mathcal{X}$ is defined analogously to $\mathfrak{F}_\Sigma$: a *partial run* ξ of $\mathcal{X}$ is a suitable $(\Sigma\cup\{*\})\times X$-labeled tree; it generates a *cylinder set* $\mathrm{Cyl}_\mathcal{X}(\xi) \subseteq \mathrm{Run}_\mathcal{X}$; and these cylinder sets generate the σ-algebra $\mathfrak{F}_\mathcal{X}$. Finally, the set $\mathrm{AccRun}_\mathcal{X}$ of *accepting runs* consists of all those runs all branches of which satisfy the (usual) *parity acceptance condition* (namely: $\max\{i \mid \pi \text{ visits states in } X_i \text{ infinitely often}\}$ is even).

The following result is much like [4, Lemma 36] and hardly novel.

▶ **Lemma 5.8.** *The set* $\mathrm{AccRun}_\mathcal{X}$ *of accepting runs is an* $\mathfrak{F}_\mathcal{X}$*-measurable subset of* $\mathrm{Run}_\mathcal{X}$*.* ◀

In the following $\mathrm{NoDiv}_\mathcal{X}(x)$ is the probability with which an execution from x does not diverge: since we use the *sub*-Giry monad (Definition 5.6), a PPTA can exhibit divergence.

▶ **Definition 5.9** ($\mu^{\mathrm{Run}}_\mathcal{X}$ over $\mathrm{Run}^{\mathcal{G}}_\mathcal{X}$)**.** Let $\mathcal{X} = ((X_1,\dots,X_n),\Sigma,\delta,s)$ be a PPTA.

Firstly, for each $k \in \mathbb{N}$, let $\mathrm{NoDiv}_{\mathcal{X},k}\colon X \to [0,1]$ (“no divergence in k steps”) be defined inductively by: $\mathrm{NoDiv}_{\mathcal{X},0}(x) \coloneqq 1$ and

$$\mathrm{NoDiv}_{\mathcal{X},k+1}(x) \coloneqq \sum_{(\sigma,(x_1,\dots,x_{|\sigma|}))\in\coprod_{\sigma\in\Sigma} X^{|\sigma|}} \delta(x)\big(\sigma,(x_1,\dots,x_{|\sigma|})\big)\cdot \textstyle\prod_{i\in[1,|\sigma|]} \mathrm{NoDiv}_{\mathcal{X},k}(x_i)\,.$$

We define $\mathrm{NoDiv}_\mathcal{X}(x) \coloneqq \bigwedge_{k\in\mathbb{N}} \mathrm{NoDiv}_{\mathcal{X},k}(x)$.

Secondly we define a subprobability measure $\mu_{\mathcal{X}}^{\mathrm{Run}}$ over $\mathrm{Run}_{\mathcal{X}}$. It is given by

$$\mu_{\mathcal{X}}^{\mathrm{Run}}(\mathrm{Cyl}_{\mathcal{X}}(\xi)) \coloneqq s\bigl(\mathrm{rt}(\xi)\bigr)\cdot P_{\mathcal{X}}(\xi) \quad \text{for each partial run } \xi\text{, where } P_{\mathcal{X}}(\xi) \text{ is given by}$$

$$P_{\mathcal{X}}(\xi) \coloneqq \begin{cases} \mathrm{NoDiv}_{\mathcal{X}}(x) & \text{if } \xi = \bigl((*,x)\bigr); \\ \delta(x)\bigl(\sigma, \bigl(\mathrm{rt}(\xi_1),\dots,\mathrm{rt}(\xi_{|\sigma|})\bigr)\bigr)\cdot \prod_{i\in[1,|\sigma|]} P_{\mathcal{X}}(\xi_i) & \text{if } \xi = \bigl((\sigma,x),(\xi_1,\dots,\xi_{|\sigma|})\bigr). \end{cases} \tag{13}$$

The above extends to a measure thanks to Carathéodory's theorem. See Lemma C.3 in [27].

Thirdly we introduce a measure $\mu_{\mathcal{X}}^{\mathrm{Tree}}$ over Tree_{Σ} ("which trees are generated by what probabilities"). It is a *push-forward measure* of $\mu_{\mathcal{X}}^{\mathrm{Run}}$ along $\mathrm{DelSt}\colon \mathrm{Run}_{\mathcal{X}} \to \mathrm{Tree}_{\Sigma}$:

$$\mu_{\mathcal{X}}^{\mathrm{Tree}}(\mathrm{Cyl}_{\Sigma}(\lambda)) \;\coloneqq\; \mu_{\mathcal{X}}^{\mathrm{Run}}\left(\mathrm{DelSt}^{-1}(\mathrm{Cyl}_{\Sigma}(\lambda)) \cap \mathrm{AccRun}_{\mathcal{X}}\right) \quad \text{for each partial } \Sigma\text{-tree } \lambda. \tag{14}$$

Since X is countable DelSt is easily seen to be measurable. Finally, the *accepted language* $\mathrm{Lang}(\mathcal{X}) \in \mathcal{G}(\mathrm{Tree}_{\Sigma})$ of $\mathcal{X}$ is defined by $\mu_{\mathcal{X}}^{\mathrm{Tree}}$ in the above.

5.3 Coincidence between Conventional and Coalgebraic Languages

▶ **Lemma 5.10.** *Let $\mathcal{X} = ((X_1,\cdots,X_n),\Sigma,\delta,s)$ be a PPTA with $X = \coprod_i X_i$, and $\Psi'_{\mathcal{X}}$ be*

$$\Psi'_{\mathcal{X}}\colon [0,1]^X \to [0,1]^X,\ \Psi'_{\mathcal{X}}(p)(x) \coloneqq \textstyle\sum_{(\sigma,x_1,\dots,x_{|\sigma|})\in\coprod_\sigma X^{|\sigma|}} \delta(x)(\sigma,(x_1,\dots,x_{|\sigma|}))\cdot\prod_{i\in[1,|\sigma|]} p(x_i).$$

Let us define $\mu_{\mathcal{X},x}^{\mathrm{Tree}} \coloneqq \mu_{\mathcal{X}(x)}^{\mathrm{Tree}}$ where $\mathcal{X}(x)$ is the PPTA obtained from $\mathcal{X}$ by changing its initial distribution s into the Dirac distribution δ_x; $\mu_{\mathcal{X},x}^{\mathrm{Run}}$ is similar. We define $\mathrm{AccProb}_{\mathcal{X}}\colon X \to [0,1]$—it assigns to each state the probability of generating an accepting run—by $\mathrm{AccProb}_{\mathcal{X}}(x) \coloneqq \mu_{\mathcal{X},x}^{\mathrm{Run}}(\mathrm{AccRun}_{\mathcal{X}})$.

Consider the following equational system, where u'_i ranges over $\mathcal{K}\ell(\mathcal{G})(X_i,1)$, and $(\_)\restriction X_i$ denotes domain restriction.

$$u'_1 =_{\eta_1} \Psi'_{\mathcal{X}}([u'_1,\cdots,u'_n])\restriction X_1, \quad \dots, \quad u_n =_{\eta_n} \Psi'_{\mathcal{X}}([u'_1,\cdots,u'_n])\restriction X_n$$

We claim: 1) the system has a solution $l_1'^{\mathrm{sol}},\dots,l_n'^{\mathrm{sol}}$; and 2) $[l_1'^{\mathrm{sol}},\dots,l_n'^{\mathrm{sol}}] = \mathrm{AccProb}_{\mathcal{X}}$. ◀

Its proof (in [27]) relies on Lemma B.4 on homomorphisms of equational systems.

▶ **Theorem 5.11** (coincidence, in the probabilistic setting)**.** *Let $\mathcal{X} = ((X_1,\dots,X_n),\Sigma,\delta,s)$ be a PPTA, and $X = X_1+\cdots+X_n$, and F_Σ be the polynomial functor on* **Meas** *that corresponds to Σ. Then $\mathcal{X}$ is identified with a parity $(\mathcal{G},F_\Sigma)$-system; moreover its coalgebraic trace semantics $\mathsf{tr}^{\mathrm{p}}(\mathcal{X})$ (Definition 3.5) coincides with the (probabilistic) language $\mathrm{Lang}(\mathcal{X})$ concretely defined in Definition 5.9. Precisely: $\mathsf{tr}^{\mathrm{p}}(\mathcal{X})(\bullet)(U) = \mathrm{Lang}(\mathcal{X})(U)$ for any measurable subset U of Tree_Σ, where $\bullet$ is the unique element of 1 in $\mathsf{tr}^{\mathrm{p}}(\mathcal{X})\colon 1 \to \mathcal{G}(\mathrm{Tree}_\Sigma)$.* ◀

Acknowledgments. Thanks are due to Corina Cîrstea, Kenta Cho, Bartek Klin, Tetsuri Moriya and Shota Nakagawa for useful discussions; and to the anonymous referees for their comments.

References

1 Jirí Adámek, Filippo Bonchi, Mathias Hülsbusch, Barbara König, Stefan Milius, and Alexandra Silva. A coalgebraic perspective on minimization and determinization. In *Proc. FoSSaCS'12*, volume 7213 of *LNCS*, pages 58–73. Springer, 2012. doi:10.1007/978-3-642-28729-9_4.

2 André Arnold and Damian Niwiński. *Rudiments of μ-Calculus*, volume 146 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 2001. `doi:10.1016/S0049-237X(01)80001-X`.

3 Christel Baier and Marcus Größer. Recognizing omega-regular languages with probabilistic automata. In *Proc. LICS'05*, pages 137–146. IEEE Computer Society, 2005. URL: `http://dx.doi.org/10.1109/LICS.2005.41`, `doi:10.1109/LICS.2005.41`.

4 Arnaud Carayol, Axel Haddad, and Olivier Serre. Randomization in automata on infinite trees. *ACM Trans. Comp. Logic*, 15(3):24:1–24:33, 2014. `doi:10.1145/2629336`.

5 Vincenzo Ciancia and Yde Venema. Stream automata are coalgebras. In *Selected Papers of CMCS'12*, volume 7399 of *LNCS*, pages 90–108. Springer, 2012. `doi:10.1007/978-3-642-32784-1_6`.

6 Corina Cîrstea. Generic infinite traces and path-based coalgebraic temporal logics. *Electr. Notes in Theor. Comp. Sci.*, 264(2):83–103, 2010. `doi:10.1016/j.entcs.2010.07.015`.

7 Corina Cîrstea, Alexander Kurz, Dirk Pattinson, Lutz Schröder, and Yde Venema. Modal logics are coalgebraic. *Comp. Journ.*, 54(1):31–41, 2011. `doi:10.1093/comjnl/bxp004`.

8 Rance Cleaveland, Marion Klein, and Bernhard Steffen. Faster model checking for the modal mu-calculus. In *Proc. CAV'92*, volume 663 of *LNCS*, pages 410–422. Springer, 1992. `doi:10.1007/3-540-56496-9_32`.

9 Ernst-Erich Doberkat. *Stochastic Coalgebraic Logic*. Monographs in Theoretical Computer Science. An EATCS Series. Springer, 2009. `doi:10.1007/978-3-642-02995-0`.

10 Michèle Giry. Categorical aspects of topology and analysis. In *A categorical approach to probability theory, an Intl. Conference at Carleton University, 1981, Proceedings*, volume 915 of *Lect. Notes in Math.*, pages 68–85. Springer, 1982. `doi:10.1007/BFb0092872`.

11 Sergey Goncharov and Dirk Pattinson. Coalgebraic weak bisimulation from recursive equations over monads. In *Proc. ICALP'14, Part II*, volume 8573 of *LNCS*, pages 196–207. Springer, 2014. `doi:10.1007/978-3-662-43951-7_17`.

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

13 Ichiro Hasuo, Bart Jacobs, and Ana Sokolova. Generic trace semantics via coinduction. *Logical Methods in Comp. Sci.*, 3(4):11:1–11:36, 2007. `doi:10.2168/LMCS-3(4:11)2007`.

14 Ichiro Hasuo, Shunsuke Shimizu, and Corina Cîrstea. Lattice-theoretic progress measures and coalgebraic model checking. In *Proc. POPL'16*, pages 718–732. ACM, 2016. `doi:10.1145/2837614.2837673`.

15 Bart Jacobs. Trace semantics for coalgebras. *Electr. Notes in Theor. Comp. Sci.*, 106:167–184, 2004. `doi:10.1016/j.entcs.2004.02.031`.

16 Bart Jacobs. Introduction to coalgebra. Towards mathematics of states and observations. Draft of a book (ver. 2.0), available online, 2012. URL: `http://www.cs.ru.nl/B.Jacobs/CLG/JacobsCoalgebraIntro.pdf`.

17 Bart Jacobs, Alexandra Silva, and Ana Sokolova. Trace semantics via determinization. *J. Comp. & Syst. Sci.*, 81(5):859–879, 2015. `doi:10.1016/j.jcss.2014.12.005`.

18 Bartek Klin. Bialgebraic methods and modal logic in structural operational semantics. *Inf. & Comp.*, 207(2):237–257, 2009. `doi:10.1016/j.ic.2007.10.006`.

19 Nancy Lynch and Frits Vaandrager. Forward and backward simulations. *Inf. & Comp.*, 121(2):214–233, 1995. `doi:10.1006/inco.1995.1134`.

20 Philip S. Mulry. Lifting theorems for Kleisli categories. In *Proc. MFPS'93*, volume 802 of *LNCS*, pages 304–319. Springer, 1994. `doi:10.1007/3-540-58027-1_15`.

21 John Power and Hayo Thielecke. Environments, continuation semantics and indexed categories. In *Proc. TACS'97*, volume 1281 of *LNCS*, pages 391–414. Springer, 1997. `doi:10.1007/BFb0014560`.

22 Jan J. M. M. Rutten. Universal coalgebra: a theory of systems. *Theor. Comp. Sci.*, 249(1):3–80, 2000. `doi:10.1016/S0304-3975(00)00056-6`.

23 Christoph Schubert. Terminal coalgebras for measure-polynomial functors. In *Proc. TAMC'09*, volume 5532 of *LNCS*, pages 325–334. Springer, 2009. `doi:10.1007/978-3-642-02017-9_35`.

24 Alexandra Silva. A short introduction to the coalgebraic method. *ACM SIGLOG News*, 2(2):16–27, April 2015. `doi:10.1145/2766189.2766193`.

25 Natsuki Urabe and Ichiro Hasuo. Coalgebraic infinite traces and kleisli simulations. In Lawrence S. Moss and Pawel Sobocinski, editors, *Proc. CALCO'15*, volume 35 of *LIPIcs*, pages 320–335. Schloss Dagstuhl, 2015. `doi:10.4230/LIPIcs.CALCO.2015.320`.

26 Natsuki Urabe and Ichiro Hasuo. Quantitative simulations by matrices. *Inf. & Comp.*, 2016. In press. `doi:10.1016/j.ic.2016.03.007`.

27 Natsuki Urabe, Shunsuke Shimizu, and Ichiro Hasuo. Coalgebraic trace semantics for Büchi and parity automata. arXiv preprint, 2016.

28 Natsuki Urabe, Shunsuke Shimizu, and Ichiro Hasuo. Fair simulation for nondeterministic and probabilistic Büchi automata: a coalgebraic perspective. *CoRR*, abs/1606.04680, 2016. URL: `http://arxiv.org/abs/1606.04680`.

29 R.J. van Glabbeek. The linear time – branching time spectrum I: The semantics of concrete, sequential processes. In J.A. BergstraA. PonseS.A. Smolka, editor, *Handbook of Process Algebra*, chapter 1, pages 3–99. Elsevier, 2001. `doi:10.1016/B978-044482830-9/50019-9`.

30 Moshe Y. Vardi. An automata-theoretic approach to linear temporal logic. In *Logics for Concurrency, the of 8th Banff Higher Order Workshop, 1995, Proceedings*, volume 1043 of *LNCS*, pages 238–266. Springer, 1995. `doi:10.1007/3-540-60915-6_6`.

31 Yde Venema. Automata and fixed point logic: A coalgebraic perspective. *Inf. & Comp.*, 204(4):637–678, 2006. `doi:10.1016/j.ic.2005.06.003`.

32 James Worrell. On the final sequence of a finitary set functor. *Theor. Comp. Sci.*, 338(1-3):184–199, 2005. `doi:10.1016/j.tcs.2004.12.009`.