Coalgebraic Trace Semantics for Büchi and Parity Automata

Natsuki Urabe, Shunsuke Shimizu, and Ichiro Hasuo

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

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.
© Natsuki Urabe, Shunsuke Shimizu and Ichiro Hasuo; licensed under Creative Commons License CC-BY
27th International Conference on Concurrency Theory (CONCUR 2016). Editors: Josée Desharnais and Radha Jagadeesan; Article No. 24; pp. 24:1–24:15
Leibniz International Proceedings in Informatics (LIPIcs) Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
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 $c: X \rightarrow FX$ in a category $C$—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 ($\tau$-)transitions and weak (bi)similarity; see e.g. recent [11]. The other one is the Büchi acceptance condition. 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 $\text{Sets}^2$ 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 $\mu$ and $\nu$—for least/greatest fixed points—in diagrams. We state our results for the parity condition (that generalizes the Büchi one).

| $\begin{array}{ccc} FX & \xrightarrow{c} & FZ \\
\downarrow & \cong & \downarrow \\
X & \xrightarrow{\text{tr}^\nu(c)} & Z \end{array}$ in a Kleisli category $\mathcal{K}(T)$ |
| Characterization of languages under no (i.e. the trivial) acceptance condition [15, 6] |
| $\Rightarrow$ |
| $\begin{array}{ccc} FX & \xrightarrow{c_1} & FZ \\
\downarrow & =_{\mu} & \downarrow \\
X_1 & \xrightarrow{\text{tr}^\mu(c_1)} & Z \\
\end{array}$ |
| $\begin{array}{ccc} FX & \xrightarrow{c_2} & FZ \\
\downarrow & =_{\nu} & \downarrow \\
X_2 & \xrightarrow{\text{tr}^\nu(c_2)} & Z \end{array}$ |

Under the Büchi acceptance condition, with $X_1 = \{\text{accepting states}\}$ and $X_2 = \{\text{nonaccepting states}\}$ (1)

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 $\mu$’s and $\nu$’s (tr$^\mu$ 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 \( \mu \)'s and \( \nu \)'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 \( X \rightarrow FX \)—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-
A function \( X \to TFX \) is nothing but an \( F \)-coalgebra \( X \to F X \) in the Kleisli category \( K\ell(T) \)—where \( F \) 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 \( \text{Sets} \) to \( K\ell(T) \).

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

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 \( \text{Sets} \); the second is the \( (sub-) \)Giry monad \( \mathcal{G} \) and a polynomial functor on \( \text{Meas} \), the category of measurable spaces and measurable functions. The Giry monad \( \mathcal{G} \) is commonly used for modeling (not necessarily discrete) probabilistic processes. We shall use its "sub" variant; a subprobability measure over \((X, \mathcal{F}_X)\) is a measure \( \mu \) such that \( 0 \leq \mu(X) \leq 1 \) (we do not require \( \mu(X) = 1 \)).

**Definition 2.1** \( \mathcal{P}, \mathcal{G} \). The powerset monad \( \mathcal{P} \) on \( \text{Sets} \) is \( \mathcal{P}X = \{A \subseteq X\}; \quad (\mathcal{P}f)(A) = \{f(x) \mid x \in A\} \); 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 \( \text{Meas} \) such that \( \mathcal{G}(X, \mathcal{F}_X) = (\mathcal{G}X, \mathcal{G}\mathcal{F}_X) \), where \( \mathcal{G}X \) is the set of all subprobability measures on \((X, \mathcal{F}_X)\), and \( \mathcal{G}\mathcal{F}_X \) is the smallest \( \sigma \)-algebra such that, for each \( S \in \mathcal{F}_X \), the function \( \text{ev}(x) : \mathcal{G}X \to [0,1] \) defined by \( \text{ev}_{S}(P) = P(S) \) is measurable. Moreover, \( \eta^\mathcal{G}_{X,\mathcal{F}_X}(x)(S) = 1 \) if \( x \in S \) and \( 0 \) otherwise (the Dirac distribution), and \( \mu^\mathcal{G}_{X,\mathcal{F}_X}(\Psi)(S) = \int_{S} \text{ev}_{\Psi} d\Psi \).

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

A (standard Borel) polynomial functor \( F \) on \( \text{Meas} \) is defined by the BNF notation \( F ::= \text{id} \mid (A, \mathcal{F}_A) \mid F_1 \times F_2 \mid \bigsqcup_{i \in I} F_i \). Here \( I \) is countable, and we require each constant \( (A, \mathcal{F}_A) \in \text{Meas} \) be a standard Borel space (see e.g. [9]). The \( \sigma \)-algebra \( \mathcal{F}_X \) associated to \( FX \) is defined as usual, with (co)product \( \sigma \)-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 \( \Sigma \) with an arity map \( \underline{\text{arity}} : \Sigma \to \mathbb{N} \). 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** \( \text{Trees}_\Sigma \). An (infinitary) \( \Sigma \)-tree, as in the standard definition, is a possibly infinite tree whose nodes are labeled with the ranked alphabet \( \Sigma \) and whose branching degrees are consistent with the arity of labels. The set of \( \Sigma \)-trees is denoted by \( \text{Trees}_\Sigma \).

---

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>
<thead>
<tr>
<th>Semantics</th>
<th>Finite trace</th>
<th>Infinitary trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coalgebraic modeling</td>
<td>$T^X \xrightarrow{F(tr(c))} T^A \xrightarrow{\text{final}}$ (3)</td>
<td>$T^X \xrightarrow{F(tr^n(c))} T^Z \xrightarrow{\text{weakly}}$ (4)</td>
</tr>
<tr>
<td>Finality in $\mathcal{K}(T)$ (Theorem 2.7)</td>
<td>(Weak finality + maximality) in $\mathcal{K}(T)$ (Theorem 2.8)</td>
<td></td>
</tr>
</tbody>
</table>

Lemma 2.4. Let $\Sigma$ be a ranked alphabet, and $F_{\Sigma} = \coprod_{\Sigma \in \Sigma}^{|\Sigma|}$ be the corresponding polynomial functor on $\text{Sets}$. The set $\text{Tree}_\Sigma$ of (infinite) $\Sigma$-trees carries a final $F_{\Sigma}$-coalgebra. The same holds in $\text{Meas}$, for countable $\Sigma$ and the corresponding polynomial functor $F_{\Sigma}$.

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

It is known [13, 25] that for $(\mathcal{C}, T) \in \{(\text{Sets}, \mathcal{P}), (\text{Meas}, \mathcal{G})\}$ and polynomial $F$ on $\mathcal{C}$, there is a canonical distributional law [20] $\lambda: FT \xrightarrow{\lambda} K\mathcal{I}(T)$—a natural transformation compatible with $T$’s monad structure. Such $\lambda$ induces a functor $F^\mathcal{I}: K\mathcal{I}(T) \to K\mathcal{I}(T)$ that makes the diagram (2) commute.

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

Then the dynamics of $A$—ignoring its initial and accepting states—is modeled as an $F^\mathcal{I}$-coalgebra $c: X \to F^\mathcal{I}X$ in $K\mathcal{I}(\mathcal{P})$ 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 $F^\mathcal{I}$-coalgebra $c: X \to F^\mathcal{I}X$ in $K\mathcal{I}(\mathcal{G})$ where: $F = \{a, b\} \times \{\_\}$, $X = \{x_1, x_2\}$ with the discrete measurable structure, and $c: X \to \mathcal{G}FX$ is the (measurable) function defined by $c(x)\{\{a, x_1\}\} = c(x)\{\{b, x_2\}\} = 1/2$, and $c(x)\{d, x'\} = 0$ 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 $f: X \to Y$ in $K\mathcal{I}(\mathcal{G})$, 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 $f: X \to Y$ in $K\mathcal{I}(\mathcal{P})$ diverges at $x$ if $f(x) = \emptyset$.

This is in contrast with a system coming to halt generating a 0-ary symbol (such as $\checkmark$ 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 $\text{Sets}$, 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 $K\mathcal{I}(T)$.
The carrier $A$ of an initial $F$-algebra in $\text{Sets}$ 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 $\text{tr}(c)$ 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.

\textbf{Theorem 2.8} ([15, 6, 25]). Let $(\mathbb{C}, T) \in \{(\text{Sets}, \mathcal{P}), (\text{Meas}, \mathcal{G})\}$ and $F$ be a polynomial functor on $\mathbb{C}$. A final $F$-coalgebra $\zeta : Z \rightharpoonup FZ$ in $\mathbb{C}$ gives rise to a weakly final $\overline{F}$-coalgebra in $\mathbb{K}(T)$, as in (3) in Table 1. Moreover, the coalgebra $J_\zeta$ additionally admits the greatest homomorphism $\text{tr}^\omega(c)$ with respect to the pointwise order $\sqsubseteq$ in the homsets of $\mathbb{K}(T)$ (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 $f \sqsubseteq \text{tr}^\omega(c)$.

In many examples the greatest homomorphism $\text{tr}^\omega(c)$ 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 $\sqrt{2}$ 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 $F = \{\sqrt{2}\} + \{a\} \times (\omega)$.

\subsection{Equational Systems for Alternating Fixed Points}

\textit{Nested, alternating} greatest and least fixed points—as in a $\mu$-calculus formula $\nu u_2.\mu u_1. (p \land u_2) \lor \Box u_1$—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 \textit{equational systems}—we prefer them to the textual $\mu$-calculus-like presentations.

\textbf{Definition 2.9} (equational system). Let $L_1, \ldots, L_n$ be posets. An \textit{equational system} $E$ over $L_1, \ldots, L_n$ is an expression

$$u_1 =_{\mathfrak{f}_1} f_1(u_1, \ldots, u_n), \ldots, u_n =_{\mathfrak{f}_n} f_n(u_1, \ldots, u_n)$$

(6)

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

The \textit{solution} of the equational system $E$ is defined as follows, under the assumption that $L_i$‘s have enough suprema and inifmums. It proceeds as: 1) we solve the first equation to obtain an interim solution $u_1 = f_1^{(1)}(u_2, \ldots, u_n)$; 2) it is used in the second equation to eliminate $u_1$ and yield a new equation $u_2 =_{u_2} f_2^{(1)}(u_2, \ldots, u_n)$; 3) solving it again gives an interim solution $u_2 = f_2^{(2)}(u_3, \ldots, u_n)$; 4) continuing this way from left to right eventually eliminates all variables and leads to a closed solution $u_n = f_n^{(n)}(u_n) \in L_n$; 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 \textit{matters}: for $(u =_{\mu} v, v =_{\nu} u)$ the solution is $u = v = T$ while for $(v =_{\nu} u, u =_{\mu} v)$ the solution is $u = v = \bot$. 

\begin{itemize}
\item \textbf{Theorem 2.7} ([13]). Let $T \in \{\mathcal{P}, \mathcal{D}\}$ and $F$ be a polynomial functor on $\text{Sets}$. An initial $F$-algebra $\alpha : FA \rightharpoonup A$ in $\text{Sets}$ yields a final $\overline{F}$-coalgebra in $\mathbb{K}(T)$, as in (3) in Table 1.
\end{itemize}
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 non-deterministic setting: note that $\mathcal{P}Y$, hence the homset $\mathcal{K}(\mathcal{P})(X,Y)$, are complete lattices.

\textbf{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 $\mathcal{K}(\mathcal{G})(X,Y) = \text{Meas}(X,\mathcal{G}Y)$ 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 $\mathcal{K}(\mathcal{G})(X,1)$ 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 $\mathcal{K}(\mathcal{G})(X,1)$; and 2) to “extend” fixed points in $\mathcal{K}(\mathcal{G})(X,1)$ along a final $F$-sequence. See Section 5.1 later.

\textbf{Lemma 2.11.} The equational system $E$ (6) has a solution if: each $L_i$ is both a pointed $\omega$-cpo and a pointed $\omega^{op}$-cpo; and each $f_i$ is both $\omega$-continuous and $\omega^{op}$-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.

\section{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 $\mathcal{C}$, $T$ 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).

\textbf{Assumptions 3.1.} In what follows a monad $T$ and an endofunctor $F$, both on $\mathcal{C}$, satisfy:

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

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

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

\subsection{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 \( 2 \times (\_)^{2} \)-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 \( X_{1} = \{ \bigcirc \}'s \) and \( X_{2} = \{ \odot \}'s \). 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 \( \mathcal{X} = ( (X_{1}, \ldots, X_{n}), c: X \rightarrow \mathcal{F}X, s: 1 \rightarrow X ) \) where \( n \) is a positive integer, and:

\( (X_{1}, \ldots, X_{n}) \) is an \( n \)-tuple of objects in \( \mathcal{C} \) for states (with their priorities), and we define \( X = X_{1} + \cdots + X_{n} \) (a coproduct in \( \mathcal{C} \));

\( c: X \rightarrow \mathcal{F}X \) is an arrow in \( \mathcal{K}(T) \) for dynamics; and

\( s: 1 \rightarrow X \) is an arrow in \( \mathcal{K}(T) \) for initial states.

For each \( i \in [1, n] \) we define \( c_{i}: X_{i} \rightarrow \mathcal{F}X \) to be the restriction \( c \circ \kappa_{i}: X_{i} \rightarrow \mathcal{F}X \) along the coprojection \( \kappa_{i}: X_{i} \hookrightarrow X \), 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 \( \mathcal{X} = ( (X_{1}, \ldots, X_{n}), c, s ) \) be a parity \((T, F)\)-system. It induces the following equational system \( E_{\mathcal{X}} \), where \( \zeta: Z \Rightarrow FZ \) is a final coalgebra in \( \mathcal{C} \) (see Assumption 3.1). The variable \( u_{i} \) ranges over the poset \( \mathcal{K}(T)(X_{1}, Z) \).

\[
E_{\mathcal{X}} = \begin{cases}
    u_{1} =_{\mu} (J\zeta)^{-1} \circ \mathcal{F}[u_{1}, \ldots, u_{n}] \circ c_{1} & \in \mathcal{K}(T)(X_{1}, Z) \\
    u_{2} =_{\nu} (J\zeta)^{-1} \circ \mathcal{F}[u_{1}, \ldots, u_{n}] \circ c_{2} & \in \mathcal{K}(T)(X_{2}, Z) \\
    \vdots \\
    u_{n} =_{\eta_{i}} (J\zeta)^{-1} \circ \mathcal{F}[u_{1}, \ldots, u_{n}] \circ c_{n} & \in \mathcal{K}(T)(X_{n}, Z)
\end{cases}
\]

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, \( \mathcal{F} \) 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 \( \text{tr}_{p}^{\mathcal{X}}(\mathcal{X}): X_{1} \Rightarrow Z, \ldots, \text{tr}_{n}^{\mathcal{X}}(\mathcal{X}): X_{n} \Rightarrow Z \). The composite

\[
\text{tr}^{p}(\mathcal{X}) = \left( 1 \xrightarrow{X = X_{1} + X_{2} + \cdots + X_{n}} \frac{\text{tr}_{p}^{\mathcal{X}}(\mathcal{X}), \text{tr}_{2}^{\mathcal{X}}(\mathcal{X}), \ldots, \text{tr}_{n}^{\mathcal{X}}(\mathcal{X})}{Z} \right)
\]

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 \( \text{tr}_{p}^{\mathcal{X}}(\mathcal{X}) \) and \( \text{tr}_{p}^{\mathcal{X}}(\mathcal{X}) \) in place—can be expressed as the following diagrams (with explicit \( \mu \) and \( \nu \)). See (1).
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 $C = \text{Sets}$, $T = \mathcal{P}$, 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 $\mathcal{K}(\mathcal{P})(X, Z)$ 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

$$\mathcal{X} = (X_1, \ldots, X_n, \Sigma, \delta: X \rightarrow \mathcal{P}(\bigoplus_{\sigma \in \Sigma} X^{\sigma}), s \in \mathcal{P} X)$$

where $X = X_1 + \cdots + X_n$, each $X_i$ is the set of states with the priority $i$, $\Sigma$ is a ranked alphabet (with the arity map $|\cdot|: \Sigma \rightarrow \mathbb{N}$), $\delta$ 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 $\rho$ is a run of an NPTA $\mathcal{X} = (X, \Sigma, \delta, s)$ if: for each node with a label $(\sigma, x)$, it has $|\sigma|$ children and we have $(\sigma, (x_1, \ldots, x_{|\sigma|})) \in \delta(x)$ where $x_1, \ldots, x_{|\sigma|}$ 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 $\rho$ of an NPTA $\mathcal{X}$ is accepting if any infinite branch $\pi$ of the tree $\rho$ satisfies the parity acceptance condition (i.e., $\max\{i | \pi \text{ visits states in } X_i \text{ infinitely often} \}$ is even). The sets of runs and accepting runs of $\mathcal{X}$ are denoted by $\text{Run}_{\mathcal{X}}$ and $\text{AccRun}_{\mathcal{X}}$, respectively.

The function $\text{rt}: \text{Run}_{\mathcal{X}} \rightarrow X$ is defined to return the root $X$-label of a run. For each $X' \subseteq X$, we define $\text{Run}_{\mathcal{X}, X'}$ by $\{\rho \in \text{Run}_{\mathcal{X}} | \text{rt}(\rho) \in X'\}$; the set $\text{AccRun}_{\mathcal{X}, X'}$ is similar.

The map $\text{DelSt}: \text{Run}_{\mathcal{X}} \rightarrow \text{Tree}_{\mathcal{X}}$ takes a run, removes all $X$-labels and returns a $\Sigma$-tree.

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

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.

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

\begin{equation}
\begin{aligned}
\begin{array}{c}
\rho_1 \quad \cdots \quad \rho_{|\sigma|}
\end{array}
\end{aligned}
\end{equation}

\begin{equation}
\begin{aligned}
\rho_1(\cdots)_{|\sigma|} = q_1 \land \rho_1(\cdots)_{|\sigma|} = q_n \land (\rho_1(\cdots)_{|\sigma|} = q_1 \land \cdots \land q_n) \land \text{Run}_{\mathcal{X}, X_1} \land \cdots \land \text{Run}_{\mathcal{X}, X_n}
\end{aligned}
\end{equation}
Here: $\diamond_X : P(\text{Run}_X) \to P(\text{Run}_X)$ is given by $\diamond_X R := \{((\sigma, x), (\rho_1, \ldots, \rho_{|\sigma|})) \in \text{Run}_X \mid \sigma \in \Sigma, x \in X, \rho_i \in R\}$ (see the figure (8) above); $X = X_1 + \cdots + X_n$; and $\eta$ is $\mu$ (for odd $i$) or $\nu$ (for even $i$). Then the $i$-th solution $I^\text{sol}_i$ coincides with $\text{AccRun}_{X,X_i}$.

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 $\mathcal{X} = (\bar{X}, \Sigma, \delta, s)$ be an NPTA, and let $I^\text{sol}_1, \ldots, I^\text{sol}_n$ be the solution of the following equational system, where $u_i'$ ranges over the complete lattice $(P(\text{Tree}_\Sigma))^X$:

$$u_1' =_{\eta_1} \diamond_\delta ([u_1', \ldots, u_n']) \uparrow X_1, \ldots, u_n' =_{\eta_n} \diamond_\delta ([u_1', \ldots, u_n']) \uparrow X_n.$$  \hspace{1cm} (10)

Here $\eta_i$ is $\mu$ (for odd $i$) or $\nu$ (for even $i$); $\langle \_ \rangle \uparrow X_i : (P(\text{Tree}_\Sigma))^X \to (P(\text{Tree}_\Sigma))^X_i$ denotes domain restriction; and the function $\diamond_\delta : (P(\text{Tree}_\Sigma))^X \to (P(\text{Tree}_\Sigma))^X$ is given by

$$(\diamond_\delta T)(x) := \{ (\sigma, (\tau_1, \ldots, \tau_{|\sigma|})) \mid (\sigma, (x_1, \ldots, x_{|\sigma|})) \in \delta(x), \tau_i \in T(x_i) \}.$$  \hspace{1cm} (9)

Then we have a coincidence $I^\text{sol}_i = \text{DelSt}'(\text{AccRun}_{X,X_i})$, where the function $\text{DelSt}' : P(\text{Run}_X) \to (P(\text{Tree}_\Sigma))^X$ is given by $\text{DelSt}'(R)(x) := \text{DelSt}(\{ \rho \in R \mid rt(\rho) = x \})$. Recall that $rt$ returns a run’s root $X$-label.

Theorem 4.6 (coincidence, in the nondeterministic setting). Let $\mathcal{X} = ((X_1, \ldots, X_n), \Sigma, \delta, s)$ be an NPTA, and $F_\Sigma = \bigsqcup_{i \in \Omega} (\_)^{\Sigma_i}$ be the polynomial functor on $\text{Sets}$ that corresponds to $\Sigma$. Then $\mathcal{X}$ is identified with a parity $(P, F_\Sigma)$-system; moreover $\text{Lang}(\mathcal{X})$ (in the conventional sense of Definition 4.3) coincides with the coalgebraic trace semantics $\text{tr}^P(\mathcal{X})$ (Definition 3.5). Note here that $\text{Tree}_\Sigma$ carries a final $F_\Sigma$-coalgebra (Lemma 2.4).

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

$$\text{tr}^P(\mathcal{X}) = [\text{DelSt}'(\text{AccRun}_{X,X_1}), \ldots, \text{DelSt}'(\text{AccRun}_{X,X_n})](s)$$

$$= \text{DelSt}(\text{AccRun}_{X,X}) = \text{Lang}(\mathcal{X})$$  \hspace{1cm} (9)

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 $K^\ell(\mathcal{G})(X, Z)$ 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 $E_X$ in Definition 3.5. As we hinted in Section 2.3 our strategy is: 1) to apply the Lemma 2.11 to the homset $K^\ell(\mathcal{G})(X, 1)$; and 2) to “extend” fixed points in $K^\ell(\mathcal{G})(X, 1)$ 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 $\mathcal{G} = \text{Meas}$, $T = \mathcal{G}$ (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:

$$X \to \mathcal{G}(A \times X) \quad \text{(word)} \quad X \to \mathcal{G}(\prod_{\sigma \in \Sigma} X^{[\sigma]}) \quad \text{(tree)}$$  \hspace{1cm} for generative;

$$X \to (GX)^A \quad \text{(word)} \quad X \to \prod_{\sigma \in \Sigma} \mathcal{G}(X^{[\sigma]}) \quad \text{(tree)}$$  \hspace{1cm} for reactive.
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 \(X = ((X_1, \ldots, X_n), s, c)\) 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 \(\Gamma : \mathcal{Kl}(\mathcal{G})(X, Z) \rightarrow \mathcal{Kl}(\mathcal{G})(X, 1)\) by \(\Gamma(g) = !Z \circ g\), where \(! : Z \rightarrow 1\) is the unique function of the type. Its variants \(\Gamma_A : \mathcal{Kl}(\mathcal{G})(X_A, Z) \rightarrow \mathcal{Kl}(\mathcal{G})(X_A, 1)\) and \(\Gamma_B : \mathcal{Kl}(\mathcal{G})(X_B, Z) \rightarrow \mathcal{Kl}(\mathcal{G})(X_B, 1)\) are defined similarly.

For arbitrary \(g_B : X_B \rightarrow Z\), we define \(S^{\eta_B}\) and \(\bar{S}^{\eta_B}\) as the following sets of “fixed points”:

\[
\begin{align*}
0_n, X_A \rightarrow Z \quad \begin{cases}
  \frac{F X}{c g} = \frac{F Z}{g} \quad \text{and} \quad \frac{h_A}{X_A} = 1 \quad \frac{F X}{c g} = \frac{F Z}{g} \quad \text{and} \quad \frac{h_A}{X_A} = 1
\end{cases}
\end{align*}
\]

Then \(\Gamma_A\) restricts to a function \(S^{\eta_B} \rightarrow \bar{S}^{\eta_B}\). Moreover, the restriction is an order isomorphism, with its inverse denoted by \(\Delta^{\eta_B} : \bar{S}^{\eta_B} \cong S^{\eta_B}\).

In the proof of the last lemma (deferred to the appendix in [27]), the inverse \(\Delta^{\eta_B}\) is defined by “extending” \(h_A : X_A \rightarrow 1\) to \(X_A \rightarrow Z\), along the final \(1\)–sequence \(1 \leftarrow F 1 \leftarrow \cdots\) (more precisely: the image of the sequence under the Kleisli inclusion \(J : \text{Meas} \rightarrow \mathcal{Kl}(\mathcal{G})\)).

We are ready to prove existence of \(E_X\)’s solution (Definition 3.5).

**Lemma 5.3.** Assume the same setting as in Lemma 5.2. We define \(\Phi_X : \mathcal{Kl}(\mathcal{G})(X, Z) \rightarrow \mathcal{Kl}(\mathcal{G})(X, 1)\) and \(\Psi_X : \mathcal{Kl}(\mathcal{G})(X, 1) \rightarrow \mathcal{Kl}(\mathcal{G})(X, 1)\), respectively, by

\[
\Phi_X(g) := J^{-1} \circ F g \circ c \quad \text{and} \quad \Psi_X(h) := J_1 F_1 \circ F h \circ c;
\]

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

\[
E = \begin{bmatrix}
  u_1 = \eta_1 \quad \Phi_X([u_1, \ldots, u_n]) \circ \kappa_1 \\
  \vdots \\
  u_n = \eta_n \quad \Phi_X([u_1, \ldots, u_n]) \circ \kappa_n
\end{bmatrix} \quad E' = \begin{bmatrix}
  u'_1 = \eta_1 \quad \Psi_X([u'_1, \ldots, u'_n]) \circ \kappa_1 \\
  \vdots \\
  u'_n = \eta_n \quad \Psi_X([u'_1, \ldots, u'_n]) \circ \kappa_n
\end{bmatrix}
\]

We claim that the equation systems have solutions \((l_1^{\text{sol}}, \ldots, l_n^{\text{sol}})\) and \((l'_1^{\text{sol}}, \ldots, l'_n^{\text{sol}})\); and moreover, we have \(\Gamma(\text{tr}^1(X)) = \Gamma([l_1^{\text{sol}}, \ldots, l_n^{\text{sol}}]) = [l'_1^{\text{sol}}, \ldots, l'_n^{\text{sol}}]\).

**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 \(S^{\eta_B} \cong \bar{S}^{\eta_B}\) 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 $h_A : X_A \rightarrow \mathcal{G}1 \cong [0,1]$—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 $\mathcal{G}^{\mathbb{N}}$ that is neither greatest nor least; the corresponding element in $\mathcal{H}^{\mathbb{N}}$ assigns to each state the probability with which it generates a accepting run (over any $\Sigma$-tree).

### 5.2 Probabilistic Parity Tree Automata and Its Languages

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

\[ \mathcal{X} = \{ (X_1, \ldots, X_n), \Sigma, \delta : X \rightarrow \mathcal{G}(\prod_{s \in \Sigma} X^{[s]}), s \in \mathcal{G}X \} , \]

where $X = X_1 + \cdots + X_n$, each $X_i$ is a countable set and $\Sigma$ 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 $\Sigma$ are not essential: restricting to discrete $\sigma$-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$\Sigma$ and Run$\mathcal{X}$).** Let $\Sigma$ be a ranked alphabet; Tree$\Sigma$ is the set of $\Sigma$-trees. A finite ($\Sigma \cup \{ * \}$)-labeled tree $\lambda$, with its branching degrees compatible with the label arities, is called a partial $\Sigma$-tree. Here the new symbol $*$ ("continuation") is deemed to be 0-ary. The cylinder set associated to $\lambda$, denoted by $\text{Cyl}_\Sigma(\lambda)$, is the set of (non-partial) $\Sigma$-trees that have $\lambda$ as their prefix (in the sense that a subtree is replaced by $*$). The (smallest) $\sigma$-algebra on Tree$\Sigma$ generated by the family $\{ \text{Cyl}_\Sigma(\lambda) \mid \lambda$ is a partial $\Sigma$-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 $\Sigma$-labels. Run$\mathcal{X}$ denotes the set of runs. The measurable structure $\mathfrak{F}_X$ on Run$\mathcal{X}$ is defined analogously to $\mathfrak{F}_\Sigma$: a partial run $\xi$ of $\mathcal{X}$ is a suitable $(\Sigma \cup \{ * \}) \times X$-labeled tree; it generates a cylinder set $\text{Cyl}_\mathcal{X}(\xi) \subseteq \text{Run}_\mathcal{X}$; and these cylinder sets generate the $\sigma$-algebra $\mathfrak{F}_X$. Finally, the set 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 AccRun$\mathcal{X}$ of accepting runs is an $\mathfrak{F}_X$-measurable subset of Run$\mathcal{X}$.

In the following 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 ($\mu^{\mathcal{X}}$ run over Run$^\Sigma_\mathcal{X}$).** Let $\mathcal{X} = ( (X_1, \ldots, X_n), \Sigma, \delta, s)$ be a PPTA.

Firstly, for each $k \in \mathbb{N}$, let NoDiv$\mathcal{X}_{\cdot k} : X \rightarrow [0,1]$ ("no divergence in $k$ steps") be defined inductively by: NoDiv$\mathcal{X}_{\cdot 0}(x) := 1$ and

\[
\text{NoDiv}_{\cdot k+1}(x) := \sum_{(\sigma, (x_1, \ldots, x_{|\sigma|})) \in \prod_{s \in \Sigma} X^{[s]} \quad |\sigma| \leq |\sigma|} \delta(x)(\sigma, (x_1, \ldots, x_{|\sigma|})) \cdot \prod_{i \in [1, |\sigma|]} \text{NoDiv}_{\cdot k}(x_i).
\]

We define NoDiv$\mathcal{X}(x) := \bigwedge_{k \in \mathbb{N}} \text{NoDiv}_{\cdot k}(x)$.
We claim: 1) the system has a solution \( s \) of \( P(\xi) \) for each partial run \( \xi \), where \( P(\xi) \) is given by

\[
P(\xi) = \begin{cases} 
\text{NoDiv}_X(x) & \text{if } \xi = (\langle \sigma \rangle, x); \\
\delta(x)(\sigma, (rt(\xi_1), \ldots, rt(\xi_\ell))) \cdot \prod_{i \in [1, \ell]} P(\xi_i) & \text{if } \xi = (\langle \sigma, x \rangle, (\xi_1, \ldots, \xi_\ell)).
\end{cases}
\]  

(13)

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

Finally, we introduce a measure \( \mu_X^{\text{Tree}}(\text{Cyl}_X(\lambda)) := \mu_X^{\text{Run}}(\text{DelSt}^{-1}(\text{Cyl}_X(\lambda)) \cap \text{AccRun}_X) \) for each partial \( \Sigma \)-tree \( \lambda \).

Since \( X \) is countable \( \text{DelSt} \) is easily seen to be measurable. Finally, the accepted language \( \text{Lang}(\chi) \in G(\text{Tree}_X) \) of \( X \) is defined by \( \mu_X^{\text{Tree}} \) in the above.

### 5.3 Coincidence between Conventional and Coalgebraic Languages

#### Lemma 5.10

Let \( X = ((X_1, \ldots, X_n), \Sigma, \delta, s) \) be a PPTA with \( X = \prod_i X_i \), and \( \Psi_X' : [0, 1]^X \to [0, 1]^X \), \( \Psi_X''(p)(x) := \sum_{(\sigma, x_1, \ldots, x_{|\sigma|}) \in \prod_i X_i^{|\sigma|}} \delta(x)(\sigma, (x_1, \ldots, x_{|\sigma|})) \cdot \prod_{i \in [1, |\sigma|]} P(x_i). \)

Let us define \( \mu_X^{\text{Tree}, X} := \mu_X^{\text{Tree}}|_{\text{tree}(X)} \) where \( X(x) \) is the PPTA obtained from \( X \) by changing its initial distribution \( s \) into the Dirac distribution \( \delta_x \); \( \mu_X^{\text{Run}} \) is similar. We define \( \text{AccProb}_X : \chi \to [0, 1] \) — it assigns to each state the probability of generating an accepting run — by \( \text{AccProb}_X(x) := \mu_X^{\text{Run}}(\text{AccRun}_X). \)

Consider the following equational system, where \( u_i' \) ranges over \( \chi(\mathcal{G})(X_1, 1) \), and \( (_) \mid X_i \) denotes domain restriction.

\[
\begin{align*}
u_1' &= _{\eta_1} \Psi_X'([u_1', \ldots, u_n']) \mid X_1, \ldots, u_n' = _{\eta_n} \Psi_X'([u_1', \ldots, u_n']) \mid X_n
\end{align*}
\]

We claim: 1) the system has a solution \( v_1^{\text{sol}}, \ldots, v_n^{\text{sol}} \); and 2) \( [v_1^{\text{sol}}, \ldots, v_n^{\text{sol}}] = \text{AccProb}_X. \)

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

#### Theorem 5.11

[Coincidence, in the probabilistic setting.]

Let \( X = ((X_1, \ldots, X_n), \Sigma, \delta, s) \) be a PPTA, and \( X = X_1 + \cdots + X_n \), and \( \mathcal{F}_X \) be the polynomial functor on \( \text{Meas} \) that corresponds to \( \Sigma \). Then \( X \) is identified with a parity \( (\mathcal{G}, \mathcal{F}_X) \)-system; moreover its coalgebraic trace semantics \( \text{tr}^P(X) \) (Definition 3.5) coincides with the (probabilistic) language \( \text{Lang}(X) \) concretely defined in Definition 5.9. Precisely: \( \text{tr}^P(X)(\bullet)(U) = \text{Lang}(X)(U) \) for any measurable subset \( U \) of \( \text{Tree}_X \), where \( \bullet \) is the unique element of \( 1 \) in \( \text{tr}^P(X) : 1 \to \mathcal{G}(\text{Tree}_X) \).

### 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

Coalgebraic Trace Semantics for Büchi and Parity Automata


