A Coalgebraic Take on Regular and ω-Regular Behaviour for Systems with Internal Moves

Tomasz Brengos
Faculty of Mathematics and Information Science, Warsaw University of Technology, ul. Koszykowa 75, 00-662 Warszawa, Poland
t.brengos@mini.pw.edu.pl

Abstract

We present a general coalgebraic setting in which we define finite and infinite behaviour with Büchi acceptance condition for systems with internal moves. Since systems with internal moves are defined here as coalgebras for a monad, in the first part of the paper we present a construction of a monad suitable for modelling (in)finite behaviour. The second part of the paper focuses on presenting the concepts of a (coalgebraic) automaton and its (ω-) behaviour. We end the paper with coalgebraic Kleene-type theorems for (ω-) regular input. We discuss the setting in the context of non-deterministic (tree) automata and Segala automata.

2012 ACM Subject Classification Theory of computation Models of computation

Keywords and phrases coalgebras, regular languages, omega regular languages, automata, Büchi automata, silent moves, internal moves, monads, saturation

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2018.25

Acknowledgements I want to thank Marco Peressotti for his continuous support and feedback. I am grateful to the anonymous referees for valuable comments and remarks.

1 Introduction

Automata theory is one of the core branches of theoretical computer science and formal language theory. One of the most fundamental state-based structures considered in the literature is a non-deterministic automaton and its relation with languages. Non-deterministic automata with a finite state-space are known to accept *regular languages*, characterized as subsets of words over a fixed finite alphabet that can be obtained from the languages consisting of words of length less than or equal to one via a finite number of applications of three types of operations: union, concatenation and the Kleene star operation [22]. This result is known under the name of *Kleene theorem for regular languages* and readily generalizes to other types of finite input (see e.g. [31]).

$$R ::= \varnothing \mid a, a \in \Sigma_\varepsilon \mid R + R \mid R \cdot R \mid R^*$$

Figure 1 Reg. exp. grammar.

On the other hand, non-deterministic automata have a natural infinite semantics which is given in terms of infinite input satisfying the so-called Büchi acceptance condition (or *BAC* in short). The condition takes into account the terminal states of the automaton and requires them to be visited infinitely often. It is a common practise to use the term *Büchi automata* in order to refer to automata whenever their infinite semantics is taken into consideration. Although the standard type of infinite input of a Büchi automaton is the set of infinite words over a given alphabet, other types (e.g. trees) are also commonly studied [31]. The class of languages of infinite words accepted by Büchi automata can also be characterized akin to the characterization

input type	*Kleene theorem*	*where*
ω-words	$\bigcup_{i=1}^n R_i \cdot L_i^\omega$	$R_i, L_i =$ regular lang.
ω-trees	$T_0 \cdot [T_1 \dots T_n]^\omega$	$T_i =$ regular tree lang.

Figure 2 Kleene thm. for ω-regular input.

29th International Conference on Concurrency Theory (CONCUR 2018).
Editors: Sven Schewe and Lijun Zhang; Article No. 25; pp. 25:1–25:18
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

of regular languages. This result is known under the name of *Kleene theorem for ω-regular languages* and its variants hold for many input types (see e.g. [17, 31]). Roughly speaking, any language recognized by a Büchi automaton can be represented in terms of regular languages and the infinite iteration operator $(-)^\omega$. This begs the question of a unifying framework these systems can be put in and reasoned about on a more abstract level so that the analogues of Kleene theorems for (ω-)regular input are derived. The recent developments in the theory of coalgebra [11, 32, 35, 36] show that the coalgebraic framework may turn out to be suitable to achieve this goal.

A coalgebra $X \to FX$ is an abstract (categorical) representation of a single step of computation of a given process [18, 32]. The coalgebraic setting has already proved itself useful in modelling finite behaviour via least fixpoints (e.g. [8, 21, 35]) and infinite behaviour via greatest fixpoints of suitable mappings [12, 24]. The infinite behaviour with BAC can be modelled by a combination of the two [30, 36].

Our paper plans to revisit the coalgebraic framework of (in)finite behaviour from the perspective of *systems with internal moves*. A unifying theory of systems with internal steps has been part of the focus of the coalgebraic community in recent years [6–10, 35] and was mainly motivated by the research in finite behaviour of such systems. Intuitively, these systems have a special computation branch that is silent. This special branch, usually denoted by the letter τ or ε, is allowed to take several steps and in some sense remain neutral to the structure of a process. These systems arise in a natural manner in many branches of theoretical computer science, among which are process calculi [29] (labelled transition systems with τ-moves and their weak bisimulation) or automata theory (automata with ε-moves), to name only two. The approach from [8, 9] suggests that these systems should be defined as coalgebras whose type is a monad. This treatment allows for an elegant modelling of weak behavioural equivalences [9, 10] among which we find Milner's weak bisimulation [29]. Each coalgebra $\alpha : X \to TX$ becomes an endomorphism $\alpha : X \to X$ in the Kleisli category for the monad T and Milner's weak bisimulation on a labelled transition system α is defined to be a strong bisimulation on its *saturation* α^* which is the smallest LTS over the same state space satisfying $\alpha \leq \alpha^*$, $\mathsf{id} \leq \alpha^*$ and $\alpha^* \cdot \alpha^* \leq \alpha^*$ (where the composition and the order are given in the Kleisli category for the LTS monad) [8]. Hence, intuitively, α^* is the reflexive and transitive closure of α and is formally defined as the least fixpoint $\alpha^* = \mu x.(\mathsf{id} \vee x \cdot \alpha)$. Since a reflexive and transitive closure is understood as an accumulation of a *finite* number of compositions of the structure with itself, the concept of coalgebraic saturation is intrinsically related to *finite* behaviour of systems. A similar treatment of infinite behaviour (and/or their combination) in the context of systems with internal moves has not been considered so far.

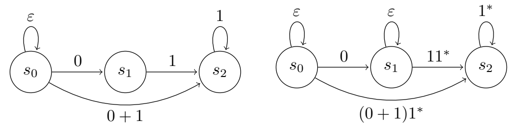


Figure 3 LTS with ε-moves and its saturation.

The aim of the paper. We plan to:

1. revisit non-deterministic (Büchi) automata and their behaviour in the coalgebraic context of systems with internal moves,
2. provide a type monad suitable for modelling (in)finite behaviour of general systems,
3. present a setting for defining (in)finite behaviour for abstract automata with silent moves,
4. state coalgebraic Kleene theorems for (ω-)regular behaviour.

The first point in the list is achieved by describing non-deterministic (Büchi) automata and their finite and infinite behaviour in terms of different coalgebraic (categorical) fixpoint constructions calculated in the Kleisli category for a suitable monad. Section 3 serves as a motivation for the framework presented later in Section 4 and Section 5.

Originally [20,35], coalgebras with internal moves were considered as systems $X \to TF_\varepsilon X$ for a monad T and an endofunctor F, where $F_\varepsilon \triangleq F + \mathcal{I}d$. The functor TF_ε could be embedded into the monad TF^*, where F^* is the free monad over F [8]. The monad TF^* is enough to model systems with internal moves and their finite behaviour [6,8,9]. However, it will prove itself useless in the context of infinite behaviour. Hence, by revisiting and tweaking the construction of TF^* from [8], Section 4 gives a general description of the monad TF^∞, the type functor TF_ε embeds into, which is used in the remaining part of the paper to model the combination of finite and infinite behaviour. Point (3) in the above list is achieved by using two fixpoint operators: the saturation operator $(-)^*$ and a new operator $(-)^\omega$ calculated in (a full subcategory of) the Kleisli category for a monad which admits infinite behaviour. The combination of $(-)^*$ and $(-)^\omega$ allows us to define infinite behaviour with BAC. Since we are mainly interested in finite state systems, all our results are presented in the context of the full subcategory of the Kleisli category whose objects are sets $\{1,\ldots,n\}$ for $n = 0,1,\ldots$, *a.k.a.* the Lawvere theory associated with the given monad. Kleene-type theorems of (4) are a direct consequence of the definition of finite and infinite behaviour with BAC using $(-)^*$ and $(-)^\omega$.

2 Basic notions

In this paper we assume the reader is familiar with basic category theory concepts like functor, (sub)monad, adjunction, Kleisli category, lifting of a functor to Kleisli category via distributive law, (initial) F-algebra, (final) F-coalgebra. For a thorough introduction to category theory the reader is referred to [28]. See also e.g. [7–9] for an extensive list of notions needed here.

Non-deterministic (Büchi) automata and their behaviour. Classically, a *nondeterministic automaton*, or simply *automaton*, is a tuple $\mathcal{Q} = (Q, \Sigma, \delta, q_0, \mathfrak{F})$, where Q is a finite set of *states*, Σ finite set called *alphabet*, $\delta : Q \times \Sigma \to \mathcal{P}(Q)$ a *transition function* and $\mathfrak{F} \subseteq Q$ set of *accepting states*. We write $q_1 \stackrel{a}{\to} q_2$ if $q_2 \in \delta(q_1, a)$. There are two standard types of semantics of automata: finite and infinite. The finite semantics, also known as the *language of finite words* of $\mathcal{Q}$, is defined as the set of all finite words $a_1 \ldots a_n \in \Sigma^*$ for which there is a sequence of transitions $q_0 \stackrel{a_1}{\to} q_1 \stackrel{a_2}{\to} q_2 \ldots q_{n-1} \stackrel{a_n}{\to} q_n$ which ends in an accepting state $q_n \in \mathfrak{F}$ [22]. The infinite semantics, also known as the ω*-language* of $\mathcal{Q}$, is the set of infinite words $a_1 a_2 \ldots \in \Sigma^\omega$ for which there is a run $r = q_0 \stackrel{a_1}{\to} q_1 \stackrel{a_2}{\to} q_2 \stackrel{a_3}{\to} q_3 \ldots$ for which the set of indices $\{i \mid q_i \in \mathfrak{F}\}$ is infinite, or in other words, the run r visits the set of final states $\mathfrak{F}$ infinitely often. Often in the literature, in order to emphasize that the infinite semantics is taken into consideration the automata are referred to as *Büchi automata* [31]. In our work we will consider (Büchi) automata without the initial state specified and define the (ω-)language in an automaton for any given state.

There are several other variants of input for non-deterministic Büchi automata known in the literature [17,31]. Here, we mention non-deterministic (Büchi) tree automaton, i.e. a tuple $(Q, \Sigma, \delta, \mathfrak{F})$, where $\delta : Q \times \Sigma \to \mathcal{P}(Q \times Q)$ and the rest is as before. The infinite semantics of this machine is the set of infinite binary trees with labels in Σ for which there is a run whose every branch visits $\mathfrak{F}$ infinitely often [17,31].

Coalgebras with internal moves and their type monads. As mentioned before coalgebras with internal moves were first introduced in the context of coalgebraic trace semantics as coalgebras of the type TF_ε for a monad T and an endofunctor F on C [20, 35]. If we take $F = \Sigma \times \mathcal{I}d$ then we have $TF_\varepsilon = T(\Sigma \times \mathcal{I}d + \mathcal{I}d) \cong T(\Sigma_\varepsilon \times \mathcal{I}d)$, where $\Sigma_\varepsilon \triangleq \Sigma + \{\varepsilon\}$. In [8] we showed that given some mild assumptions on T and F we may embed the functor TF_ε into the monad TF^*, where F^* is the free monad over F. In particular, if we apply this construction to $T = \mathcal{P}$ and $F = \Sigma \times \mathcal{I}d$ we obtain the monad $\mathcal{P}(\Sigma^* \times \mathcal{I}d)$ from Example 2.1 below. This construction is also revisited in this paper in Section 4. The trick of modelling the invisible steps via a monadic structure allows us *not* to specify the internal moves explicitly. Instead of considering TF_ε-coalgebras we consider T'-coalgebras for a monad T' on an arbitrary category.

The strategy of finding a suitable monad (for modelling the behaviour taken into consideration) will also be applied in this paper. Unfortunately, from the point of view of the infinite behaviour of coalgebras, considering systems of the type TF^* is not sufficient (see Section 3 for a discussion). Hence, in Section 4 we show how to obtain monads suitable for modelling infinite behaviour. Below, we list basic examples of monads considered in this paper. This list will be extended in sections to come.

▶ **Example 2.1.** The powerset endofunctor $\mathcal{P} : \mathsf{Set} \to \mathsf{Set}$ carries a monadic structure for which the category $\mathcal{K}l(\mathcal{P})$ consists of sets as objects and maps $f : X \to \mathcal{P}Y$ and $g : Y \to \mathcal{P}Z$ with the composition $g \cdot f : X \to \mathcal{P}Z$ defined as follows $g \cdot f(x) = \{z \in Z \mid z \in \bigcup g(f(x))\}$. The identity morphisms $\mathsf{id} : X \to \mathcal{P}X$ are given for any $x \in X$ by $\mathsf{id}(x) = \{x\}$. Now, for a set Σ the functor $\mathcal{P}(\Sigma^* \times \mathcal{I}d)$ carries a monadic structure whose composition in the Kleisli category is given as follows [8]. For $f : X \to \mathcal{P}(\Sigma^* \times Y)$ and $g : Y \to \mathcal{P}(\Sigma^* \times Z)$ we have $g \cdot f(x) = \{(\sigma_1\sigma_2, z) \mid x \xrightarrow{\sigma_1}_f y \xrightarrow{\sigma_2}_g z \text{ for some } y \in Y\}$. The identity morphisms in this category are $\mathsf{id} : X \to \mathcal{P}(\Sigma^* \times X)$ given by $\mathsf{id}(x) = \{(\varepsilon, x)\}$. Finally, let Σ^ω be the set of all infinite sequences of elements from Σ. The functor $\mathcal{P}(\Sigma^* \times \mathcal{I}d + \Sigma^\omega)$ carries a monadic structure whose Kleisli composition is the following. For $f : X \to \mathcal{P}(\Sigma^* \times Y + \Sigma^\omega)$ and $g : Y \to \mathcal{P}(\Sigma^* \times Z + \Sigma^\omega)$ the map $g \cdot f : X \to \mathcal{P}(\Sigma^* \times Z + \Sigma^\omega)$ is:

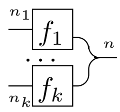
$$x \xrightarrow{\sigma}_{g\cdot f} z \iff \exists y \text{ s.t. } x \xrightarrow{\sigma_1}_f y \text{ and } y \xrightarrow{\sigma_2}_g z, \text{ where } \sigma = \sigma_1\sigma_2 \in \Sigma^*,$$
$$x \downarrow_{g\cdot f} v \iff x \downarrow_f v \text{ or } x \xrightarrow{\sigma}_f y,\ y \downarrow_g v' \text{ and } v = \sigma v' \in \Sigma^\omega.$$

In the above we write $x \xrightarrow{\sigma}_f y$ whenever $(\sigma, y) \in f(x)$ and $x \downarrow_f v$ if $v \in f(x)$ for $\sigma \in \Sigma^*$, $v \in \Sigma^\omega$. The identity morphisms in this category are the same as in the Kleisli category for the monad $\mathcal{P}(\Sigma^* \times \mathcal{I}d)$. The monadic structure of $\mathcal{P}(\Sigma^* \times \mathcal{I}d + \Sigma^\omega)$ arises as a consequence of a general construction of monads modelling (in)finite behaviour described in detail in Section 4.

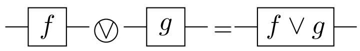
▶ **Example 2.2.** The subconvex distributions functor $\mathcal{CM}$ used to model Segala systems [33, 34] is defined as follows [16]. For any set X define $\mathcal{M}X$ to be the carrier of the free module for the semiring $[0, \infty)$ over X and put $\mathcal{CM}X = \{U \subseteq \mathcal{M}X \mid U = \overline{U} \text{ and } U \neq \varnothing\}$, where for $U \subseteq \mathcal{M}X$ we have $\overline{U} \triangleq \{\sum_{i=1}^n r_i \cdot u_i \mid u_i \in U, r_i \in [0, \infty) \;\&\; \sum_i r_i \leq 1\}$. For any map $f : X \to Y$ put $\mathcal{CM}(f) : \mathcal{CM}X \to \mathcal{CM}Y; U \mapsto \overline{\mathcal{M}f(U)}$. See also [8, 25] for a slightly different definition of $\mathcal{CM}$ and a more thorough discussion of this treatment. The functor can be equipped with a monadic structure which results in the Kleisli composition defined by: $g \cdot f(x) = \bigcup_{\phi \in f(x)} \sum_{y \in \mathsf{supp}\phi} \{\phi(y) \cdot \psi \mid \psi \in g(y)\} \in \mathcal{CM}Z$ for $x \in X$, $f : X \to \mathcal{CM}Y$ and $g : Y \to \mathcal{CM}Z$ [16].

Lawvere theories and categorical order enrichment. The primary interest of the theory of automata and formal languages focuses on automata over a *finite* state space. Hence, since we are interested in systems with internal moves (i.e. maps $X \to TX$ for a monad T), without any loss of generality we may focus our attention on coalgebras of the form $[n] \to T[n]$, where $[n] \triangleq \{1, \ldots, n\}$ with $n = 0, 1, \ldots$ for a Set-monad T. These morphisms are endomorphisms in a full subcategory of the Kleisli category for T known under the name of *Lawvere theory*. That is why we choose the setting of this paper to be Lawvere theories. Because we are interested in the coalgebraic essence of a Lawvere theory, we adopt the definition which is dual to the classical notion [27].

Formally, a *Lawvere theory*, or simply *theory*, is a category whose objects are natural numbers $n \geq 0$ such that each n is an n-fold coproduct of 1. For any element $i \in [n]$ let $i_n : 1 \to n$ denote the i-th coproduct injection and $[f_1, \ldots, f_k] : n_1 + \ldots + n_k \to n$ the cotuple of the family $\{f_l : n_l \to n\}_l$ depicted in the diagram on the right. The coprojection $n_i \to n_1 + \ldots + n_k$ into the i-th component of the coproduct will be denoted by $\mathsf{in}^{n_i}_{n_1+\ldots+n_k}$. Any morphism $k \to n$ of the form $[i^1_n, \ldots, i^k_n] : k \to n$ for $i^j \in [n]$ is called *base morphism* or *base map*. Finally, let $! : n \to 1$ be defined by $! \triangleq [1_1, 1_1, \ldots, 1_1]$. We say that a theory $\mathbb{T}'$ is a *subtheory* of $\mathbb{T}$ if there is a faithful functor $\mathbb{T}' \to \mathbb{T}$ which maps any object n onto itself. Any monad T on Set induces a theory $\mathbb{T}$ associated with it by restricting the Kleisli category $\mathcal{K}l(T)$ to objects $[n]$ for any $n \geq 0$. Conversely, for any theory $\mathbb{T}$ there is a Set based monad the theory is associated with (see e.g. [23] for details).



In order to establish the definition of the fixpoint operators $(-)^*$ and $(-)^\omega$ we require the Lawvere theory under consideration to be suitably order enriched. A category is said to be *order enriched*, or simply *ordered*, if each hom-set is a poset with the order preserved by the composition. It is $\vee$*-ordered* if all hom-posets admit arbitrary finite suprema. Note that, given such suprema exist, the composition in C does not have to distribute over them in general. We call such category *left distributive* (or *LD* in short) if $h \cdot (f \vee g) = h \cdot f \vee h \cdot g$. In this paper we will come across many left distributive categories that do not necessarily satisfy right distributivity. Still, however, all the examples taken into consideration satisfy a weaker form of right distributivity. To be precise, we say that a theory is *right distributive w.r.t. base morphisms* (or *bRD* in short) provided that $(f \vee g) \cdot j = f \cdot j \vee g \cdot j$ for any f, g and any base morphism j. We say that an order enriched category is ω-Cpo-enriched if any ascending ω-chain $f_1 \leq f_2 \leq \ldots$ of morphisms admits a supremum $\bigvee_i f_i$ which is preserved by the morphism composition. Finally, in an ordered category with finite coproducts we say that *cotupling preserves order* if $[f_1, f_2] \leq [g_1, g_2] \iff f_1 \leq g_1$ and $f_2 \leq g_2$ for any f_i, g_i with suitable domains and codomains.



▶ **Example 2.3.** The primary interest of the next section of this paper lies in the theories LTS and LTS^ω which are defined to be the theories that arise from the Kleisli categories of the monads $\mathcal{P}(\Sigma^* \times \mathcal{I}d)$ and $\mathcal{P}(\Sigma^* \times \mathcal{I}d + \Sigma^\omega)$ respectively. Both theories are order-enriched with the hom-set ordering given by $f \leq g \iff f(i) \subseteq g(i)$ for any $i \in [n]$. It is easy to see that the hom-posets of LTS and LTS^ω are complete lattices, both theories are ω-Cpo-enriched and satisfy LD and bRD. Moreover, cotupling $[-, -]$ in LTS and LTS^ω preserves order.

3 Non-deterministic (Büchi) automata, coalgebraically

The purpose of this section is to give motivations for the development of the abstract theory done in the remainder of the paper. Here, we will focus on finite non-deterministic (Büchi) automata and their (in)finite behaviour from the perspective of the theories LTS and LTS^ω.

Without any loss of generality we may only consider automata over the state space $[n]$ for some natural number n. Any non-deterministic automaton with ε-moves $([n], \Sigma_\varepsilon, \delta, \mathfrak{F})$ may be modelled as a $\mathcal{P}(\Sigma_\varepsilon \times \mathcal{I}d + 1)$−coalgebra $[n] \to \mathcal{P}(\Sigma_\varepsilon \times [n] + 1)$ [32]. However, as it has been already noted in [36], from the point of view of infinite behaviour with BAC it is more useful to extract the information about the final states of the automaton and do not encode it into the transition map as above. Instead, we consider the given automaton as a pair $(\alpha, \mathfrak{F})$ where $\alpha : [n] \to \mathcal{P}(\Sigma_\varepsilon \times [n])$ is defined by $\alpha(i) = \{(a, j) \mid j \in \delta(a, i)\}$ and consider the map:

$$\mathfrak{f}_{\mathfrak{F}} : [n] \to \mathcal{P}(\Sigma_\varepsilon \times [n]); i \mapsto \begin{cases} \{(\varepsilon, i)\} & \text{if } i \in \mathfrak{F}, \\ \varnothing & \text{otherwise.} \end{cases}$$

The purpose of $\mathfrak{f}_{\mathfrak{F}}$ is to encode the set of accepting states with an endomorphism in the same Kleisli category in which the transition α is an endomorphism. Now, we have all the necessary ingredients to revisit finite and infinite behaviour (with BAC) of non-deterministic automata from the perspective of the theory LTS^ω.

Finite behaviour. Consider $\alpha^* : n \to n$ to be an endomorphism in LTS (or LTS^ω) given by $\alpha^* = \mu x.(\mathsf{id} \vee x \cdot \alpha) = \bigvee_{n \in \omega} \alpha^n$, where the order is as in Example 2.3. We have [8]:

$$\alpha^*(i) = \{(\sigma, j) \mid i \overset{\sigma}{\Longrightarrow} j\},$$

where $\overset{\sigma}{\Longrightarrow} \triangleq (\overset{\varepsilon}{\to})^* \circ \overset{a_1}{\to} \circ (\overset{\varepsilon}{\to})^* \circ \ldots (\overset{\varepsilon}{\to})^* \circ \overset{a_n}{\to} (\overset{\varepsilon}{\to})^*$ for $\sigma = a_1 \ldots a_n$, $a_i \in \Sigma$ and $\overset{\varepsilon}{\Longrightarrow} \triangleq (\overset{\varepsilon}{\to})^*$. Now, let us recall the definition of $!$ in any theory $\mathbb{T}$. In particular, when $\mathbb{T} = \mathsf{LTS}, \mathsf{LTS}^\omega$ the map $! : [n] \to \mathcal{P}(\Sigma_\varepsilon \times [1])$ satisfies $!(i) = \{(\varepsilon, 1)\}$. Finally, consider the morphism $! \cdot \mathfrak{f}_{\mathfrak{F}} \cdot \alpha^* : n \to 1$ in LTS (or LTS^ω)) which is explicitly given by:

$$! \cdot \mathfrak{f}_{\mathfrak{F}} \cdot \alpha^*(i) = \{(\sigma, 1) \mid \sigma \in \Sigma^* \text{ such that } i \overset{\sigma}{\Longrightarrow} j \text{ and } j \in \mathfrak{F}\}.$$

Since $\mathcal{P}(\Sigma^* \times [1]) \cong \mathcal{P}(\Sigma^*)$, the set $! \cdot \mathfrak{f}_{\mathfrak{F}} \cdot \alpha^*(i)$ represents the set of all finite words accepted by the state i in the automaton $([n], \Sigma_\varepsilon, \delta, \mathfrak{F})$.

Infinite behaviour with BAC. Note that both theories LTS and LTS^ω are complete and, hence (by Tarski-Knaster theorem), come equipped with an operator which assigns to any endomorphism $\beta : n \to n$ the morphism $\beta^\omega : n \to 0$ defined as the greatest fixpoint of $\lambda x.x \cdot \beta$. For α the map $\alpha^\omega : [n] \to \mathcal{P}(\Sigma^* \times \varnothing) = \{\varnothing\}$ is unique in LTS with $\alpha^\omega(i) = \varnothing$. However, if we compute α^ω in LTS^ω the result will be different. Indeed, we have the following.

▶ **Theorem 3.1.** *Let $\beta : [n] \to \mathcal{P}(\Sigma^* \times [n])$ be a transition map with no silent moves. Then $\beta^\omega : [n] \to \mathcal{P}(\Sigma^* \times \varnothing + \Sigma^\omega) = \mathcal{P}(\Sigma^\omega)$ in LTS^ω is given by:*

$$\beta^\omega(i) = \{\sigma_1\sigma_2 \ldots \in \Sigma^\omega \mid i \overset{\sigma_1}{\to} i_1 \overset{\sigma_2}{\to} i_2 \ldots \text{ for some } i_k \in [n] \text{ and } \sigma_k \in \Sigma^* \setminus \{\varepsilon\}\}.$$

Hence, if we, for now, assume that $\alpha : [n] \to \mathcal{P}(\Sigma_\varepsilon \times [n])$ has no silent transitions then by the above theorem: $\alpha^\omega(i) = \{a_1 a_2 \ldots \in \Sigma^\omega \mid i \overset{a_1}{\to} i_1 \overset{a_2}{\to} i_2 \ldots \text{ for some } i_k \in [n]\}$. We will use the operation $(-)^\omega$ in LTS^ω to extract the information about the ω-language of $(\alpha, \mathfrak{F})$. However, we need one last ingredient. Let us define $\alpha^+ \triangleq \alpha^* \cdot \alpha$ and note

$$\alpha^+(i) = \{(\sigma, j) \mid i \overset{a_1}{\to} i_1 \ldots \overset{a_k}{\to} i_k \text{ in } \alpha \text{ and } \sigma = a_1 \ldots a_k \text{ for } k \geq 1\}.$$

Finally, consider the morphism $(\mathfrak{f}_{\mathfrak{F}} \cdot \alpha^+)^\omega : n \to 0$ in LTS^ω. In order to see the explicit formula for $(\mathfrak{f}_{\mathfrak{F}} \cdot \alpha^+)^\omega$ let us first note that the endomorphism $\mathfrak{f}_{\mathfrak{F}} \cdot \alpha^+ : [n] \to \mathcal{P}(\Sigma^* \times [n])$ has

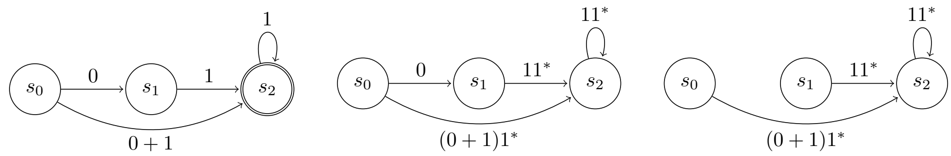


Figure 4 A non-deterministic automaton $(\alpha, \mathfrak{F})$ and the maps α^+ and $\mathfrak{f}_{\mathfrak{F}} \cdot \alpha^+$.

no silent moves and $\mathfrak{f}_{\mathfrak{F}} \cdot \alpha^+(i) = \{(\sigma, j) \mid i \xrightarrow{\sigma} j \text{ in } \alpha^+ \text{ and } j \in \mathfrak{F}\}$. Therefore, by Theorem 3.1, the map $(\mathfrak{f}_{\mathfrak{F}} \cdot \alpha^+)^\omega : [n] \to \mathcal{P}(\Sigma^\omega)$ satisfies:

$$(\mathfrak{f}_{\mathfrak{F}} \cdot \alpha^+)^\omega(i) = \text{ the } \omega\text{-language of } i \text{ in the Büchi automaton represented by } (\alpha, \mathfrak{F}).$$

This property suggests a general approach towards modelling (ω-)behaviours of abstract (coalgebraic) automata that we will develop in the sections to come.

▶ Remark. Note that throughout this paragraph we assumed the map α to have no ε-transitions. It may not be instantly clear why. It turns out that ε moves are problematic for the infinite behaviour operator $(-)^\omega$ defined as above. Indeed, in order to see this consider two finite languages $A, B \subseteq \{a, b\}^*$ defined by $A = \{\varepsilon, ab\}$ and $B = \{ab\}$. These languages can be viewed as endomorphisms $\alpha, \beta : 1 \to 1$ in LTS^ω given by $\alpha, \beta : [1] \to \mathcal{P}(\Sigma^* \times [1])$, where $\alpha(1) \triangleq \{(\varepsilon, 1), (ab, 1)\}$ and $\beta(1) \triangleq \{(ab, 1)\}$. Note that α has a silent loop, β has no silent transitions and both maps $\alpha^*, \beta^* : 1 \to 1$ satisfy $\alpha^*(1) = \{((ab)^n, 1) \mid n \geq 0\} = \beta^*(1)$. However, $\alpha^\omega \neq \beta^\omega$ in LTS^ω. Indeed, by Theorem 3.1, $\beta^\omega(1) = \{abababab\ldots\}$ but $\alpha^\omega(1) = \mathcal{P}(\{a, b\}^\omega)$ is the set of *all* infinite words over $\{a, b\}$. The latter holds, since $\mathsf{id} \leq \alpha$ in LTS^ω and the greatest fixpoint of $\lambda x.x \cdot \alpha$ is the greatest morphism $\top : 1 \to 0$ in the given theory as $\top = \top \cdot \mathsf{id} \leq \top \cdot \alpha \leq \top$. The identity $\alpha^\omega = \mathcal{P}(\{a, b\}^\omega)$ seems to be unintuitive considering the fact that in many classical works on Büchi automata (e.g. [31]) $A^\omega = B^\omega = \{abababab\ldots\}$. These papers use a slightly incompatible definition of the language operator $(-)^\omega : \mathcal{P}(\Sigma^*) \to \mathcal{P}(\Sigma^\omega)$ which explicitly removes ε from the argument set. Since it would be difficult to devise such an operator on a more abstract categorical level, we decide to keep with $\nu x.x \cdot \beta$ as the definition of β^ω and bear in mind this minor incompatibility with the classical work.

Why systems with internal moves? In the light of the above remark the reader may get the (wrong) impression that putting systems with internal moves into the context of infinite behaviour with BAC may seem rather *ad hoc*. To add to this, the need for categorical modelling of infinite behaviour for systems with silent steps is *not* sufficiently justified by the classical literature on the topic, where such systems rarely occur in practice (*conf.* [31]). However, as mentioned before, since putting systems with internal steps into the context is, in fact, extending the given setting to the setting of coalgebras $X \to TX$ whose type T is a monad, the main profit from this approach is the access to a simple and powerful language of the Kleisli category for the monad T. It allows us to abstract away from several "unnecessary" details and focus on the core properties. Hopefully, this paper demonstrates that the access to the language justifies the extension of the setting, as it makes it possible to formulate new results and provide their simple proofs which, in our opinion, would be tedious without such extension.

Büchi automata with non-standard input and beyond. As mentioned in Section 2, there are variants of non-deterministic (Büchi) automata that accept other types of input (e.g. binary trees). In general, given a functor $F : \mathsf{Set} \to \mathsf{Set}$ we define a non-deterministic (Büchi)

F-automaton as a pair $(\alpha, \mathfrak{F})$, where $\alpha : [n] \to \mathcal{P}F[n]$ (or $\alpha : [n] \to \mathcal{P}F_\varepsilon[n]$- to model systems with internal moves) and $\mathfrak{F} \subseteq [n]$. A natural question that arises is the following: are we able to build a setting in which we can reason about the (in)finite behaviour of systems for arbitrary non-deterministic Büchi F-automata (or even more generally, for systems of the type TF (or TF_ε) for a monad T)? If so, then is it possible to generalize the Kleene theorem for ($\omega-$)regular languages stated in the introduction to a coalgebraic level? We will answer these questions positively in the next sections.

4 Monads for (in)finite behaviour

Let C be a category which admits binary coproducts. We denote the coproduct operator by $+$ and the coprojection into the first and the second component of a coproduct by inl and inr respectively. Moreover, let $F : \mathsf{C} \to \mathsf{C}$ be a functor.

The purpose of this section is to present a monad the functor TF_ε embeds into that will prove itself sufficient to model the combination of finite and infinite behaviour (akin to the monad $\mathcal{P}(\Sigma^* \times \mathcal{I}d + \Sigma^\omega)$ for the functor $\mathcal{P}(\Sigma_\varepsilon \times \mathcal{I}d)$). At first we list basic facts needed in the remainder of this section. In Subsection 4.2 we revisit the construction of the monad TF^* from [8]. Here, however, we show how it can be obtained by composing a different pair of adjunctions. Finally, we give a description of the definition of TF^∞ suitable for modelling (in)finite behaviour. In what follows, in this section we assume:

- (T, μ, η) is a monad on C and $F : \mathsf{C} \to \mathsf{C}$ lifts to $\mathcal{K}l(T)$ via a dist. law $\lambda : FT \implies TF$,
- there is an initial $F(-) + X$-algebra for any object X and a terminal F-coalgebra $\zeta : F^\omega \to FF^\omega$.

4.1 Preliminaries

Existence of the initial $F(-) + X$-algebra $i_X : FF^*X + X \to F^*X$ (i.e. $i_X \circ \mathsf{inl}$ is the free F-algebra over X) for any object X yields an adjoint situation $\mathsf{C} \rightleftarrows \mathsf{Alg}(F)$, where the left adjoint is the free algebra functor which assigns to any object X the free algebra $i_X \circ \mathsf{inl} : FF^*X \to F^*X$ over it. The right adjoint is the forgetful functor which assigns to any F-algebra its carrier and is the identity on morphisms. This adjunction yields the monad $F^* : \mathsf{C} \to \mathsf{C}$ which assigns to any object X the carrier of the free F-algebra over X.

▶ **Example 4.1.** For any set Σ and X the initial $\Sigma \times \mathcal{I}d + X$-algebra is given by the morphism $i_X : \Sigma \times \Sigma^* \times X + X \to \Sigma^* \times X$, where $i_X(a, (\sigma, x)) = (a\sigma, x)$ and $i_X(x) = (\varepsilon, x)$.

Now we recall basic definitions and properties of Bloom F-algebras [1] which will be used to introduce monads for infinite behaviour in the next subsection. A pair $(a : FA \to A, (-)^\dagger)$ is called *Bloom F-algebra* provided that for any F-coalgebra $e : X \to FX$ the map $e^\dagger : X \to A$ satisfies:

$$\begin{array}{ccc} X & \xrightarrow{e^\dagger} & A \\ e\big\downarrow & & \big\uparrow a \\ FX & \xrightarrow[Fe^\dagger]{} & FA \end{array} \quad \text{and} \quad \begin{array}{ccc} X & \xrightarrow{h} & Y \\ e\big\downarrow & & \big\downarrow f \\ FX & \xrightarrow[Fh]{} & FY \end{array} \text{ implies } \begin{array}{ccc} X & \xrightarrow{h} & Y \\ & e^\dagger\searrow \quad \swarrow f^\dagger & \\ & A & \end{array}$$

By a *homomorphism* between Bloom algebras $(a : FA \to A, (-)^\dagger)$ and $(b : FB \to B, (-)^\ddagger)$ we mean a map $h : A \to B$ which is an F-algebra homomorphism from a to b and which additionally preserves the solution, i.e. $e^\dagger \circ h = e^\ddagger$. The category of Bloom algebras and homomorphisms between them is denoted by $\mathsf{Alg}_B(F)$. We assume that $\mathsf{Alg}(F)$ has binary coproducts which are denoted by $\oplus$. We have the following theorem.

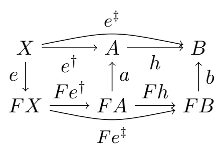
▶ **Theorem 4.2.** *[1] The pair* $(\zeta^{-1} : FF^\omega \to F^\omega, [[-]])$*, where* $[[-]]$ *assigns to* $e : X \to FX$ *the unique coalgebra homomorphism* $[[e]] : X \to F^\omega$ *between* e *and* ζ*, is an initial object in* $\mathsf{Alg}_B(F)$*. Moreover,* $i_X \circ \mathsf{inl} \oplus \zeta^{-1}$ *is the free Bloom algebra over* X*.*

▶ Remark. Let $F^\infty : \mathsf{C} \to \mathsf{C}$ be defined as the composition of the left and right adjoints $\mathsf{C} \rightleftarrows \mathsf{Alg}_B(F)$ respectively, where the left adjoint is the free Bloom algebra functor and the right adjoint is the forgetful functor. The functor F^∞ carries a monadic structure which extends F^*. Indeed, by Th. 4.2, the monad F^* is a submonad of F^∞ (via the transformation induced by the coprojection into the first component of $i_X \circ \mathsf{inl} \oplus \zeta^{-1}$ in $\mathsf{Alg}(F)$). The construction of the free Bloom algebra from the above theorem indicates that F^∞ is a natural extension of F^* encompassing infinite behaviours of the final F-coalgebra. By abusing the notation slightly, we can write $F^\infty = F^* \oplus F^\omega$. The functor F_ε is a subfunctor of F^* [8, Lemma 4.12] and hence, by the above, also of F^∞. In the following sections this will let us turn any coalgebra $X \to TFX$ or $X \to TF_\varepsilon X$ into a system $X \to TF^\infty X$ and, by doing so, allow us to model their (in)finite behaviour.

▶ **Example 4.3.** The terminal $\Sigma \times \mathcal{I}d$-coalgebra is $\zeta : \Sigma^\omega \to \Sigma \times \Sigma^\omega; a_1a_2\ldots \mapsto (a_1, a_2a_3\ldots)$. The coproduct of $a : \Sigma \times A \to A$ and $b : \Sigma \times B \to B$ in $\mathsf{Alg}(F)$ is $a \oplus b : \Sigma \times (A + B) \to A + B; (\sigma, x) \mapsto$ **if** $x \in A$ **then** $a(\sigma, x)$ **else** $b(\sigma, x)$. Hence, the free Bloom algebra over X is: $\Sigma \times (\Sigma^* \times X + \Sigma^\omega) \to \Sigma^* \times X + \Sigma^\omega$, where $(a, (\sigma, x)) \mapsto (a\sigma, x)$ and $(a, a_1a_2\ldots) \mapsto aa_1a_2\ldots$.

Let $(a : FA \to A, (-)^\dagger)$ be a Bloom algebra, $b : FB \to B$ an F-algebra and $h : A \to B$ a homomorphism between F-algebras a and b. Then there is a unigue assignment $(-)^\ddagger$ which turns $(b : FB \to B, (-)^\ddagger)$ into a Bloom algebra and h into a Bloom algebra homomorphism and it is defined as follows [1]: for $e : X \to FX$ the map $e^\ddagger : X \to B$ is $e^\ddagger \triangleq h \circ e^\dagger$.

4.2 Lifting monads to algebras



Take an F-algebra $a : FA \to A$ and define $\bar{T}(a) \triangleq FTA \stackrel{\lambda_A}{\to} TFA \stackrel{Ta}{\to} TA$. If $h : A \to B$ is a homomorphism of algebras a and $b : FB \to B$ we put $\bar{T}(h) = T(h)$. $\bar{T} : \mathsf{Alg}(F) \to \mathsf{Alg}(F)$ is a functor for which the morphism $\eta_A : A \to TA$ is an F-algebra homomorphism from $a : FA \to A$ to $\bar{T}(a) : FTA \to TA$. Moreover, $\mu_A : T^2A \to TA$ is a homomorphism from $\bar{T}^2(a)$ to $\bar{T}(a)$ (see [4] for details). A direct consequence of this construction is the following.

▶ **Theorem 4.4.** *[4] The triple* $(\bar{T}, \bar{\mu}, \bar{\eta})$*, where for* $a : FA \to A$ *we put* $\bar{\mu}_a : \bar{T}^2(a) \to \bar{T}(a)$*;* $\bar{\mu}_a = \mu_A$ *and* $\bar{\eta}_a : a \to \bar{T}(a); \bar{\eta}_a = \eta_A$ *is a monad on* $\mathsf{Alg}(F)$*.*

The above theorem together with the assumption of existence of an arbitrary free F-algebra in $\mathsf{Alg}(F)$ leads to a pair of adjoint situations in Fig. 5. Since the composition of adjoint situations is an adjoint situation this yields a monadic structure on the functor $TF^* : \mathsf{C} \to \mathsf{C}$.

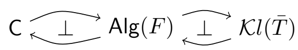


Figure 5

▶ **Example 4.5.** An example of this phenomenon is given by the monad $\mathcal{P}(\Sigma^* \times \mathcal{I}d)$ from Example 2.1 where in the above we set $T = \mathcal{P}$ and $F = \Sigma \times \mathcal{I}d$. This monad has already been described e.g. in [8], but it arose as a consequence of the composition of a different pair of adjunctions.

Monads on Bloom algebras. Above we gave a recipe for a general construction of a monadic structure on the functor TF^*. As witnessed in [6, 8], this monad is suitable to model coalgebras and their weak bisimulations and weak finite trace semantics (i.e. their finite behaviour). Our primary interest is in modelling infinite behaviour and this monad will prove itself insufficient. The purpose of this subsection is to show how to tweak the middle category from Fig. 5 so that the monad obtained from the composition of two adjunctions is suitable to our needs.

Let $(a : FA \to A, (-)^\dagger)$ be a Bloom algebra and define $\bar{T}_B((a : FA \to A, (-)^\dagger)) \triangleq (\bar{T}(a) : FTA \to TA, (-)^\ddagger)$, where for any $e : X \to FX$ the map $e^\ddagger$ is given by $\eta_A \circ e^\dagger$. Since $\eta_A : A \to TA$ is a homomorphism between $a : FA \to A$ and $\bar{T}(a) : FTA \to TA$ the pair $(\bar{T}(a), (-)^\ddagger)$ is a Bloom algebra. For a pair of Bloom algebras $(a : FA \to A, (-)^\dagger)$ and $(b : FB \to B, (-)^\ddagger)$ and a Bloom algebra homomorphism $h : A \to B$ between them put $\bar{T}_B(h) = T(h)$. This defines a functor $\bar{T}_B : \mathsf{Alg}_B(F) \to \mathsf{Alg}_B(F)$. Analogously to the previous subsection we have the following direct consequence of the construction.

▶ **Theorem 4.6.** *The triple $(\bar{T}_B, \bar{\mu}^B, \bar{\eta}^B)$ is a monad on $\mathsf{Alg}_B(F)$, where for any Bloom algebra $(a : FA \to A, (-)^\dagger)$ the $(a, (-)^\dagger)$-components of the transformations $\bar{\mu}^B$ and $\bar{\eta}^B$ are $\bar{\mu}^B_{(a,(-)^\dagger)} : \bar{T}^2_B(a, (-)^\dagger) \to \bar{T}_B(a, (-)^\dagger)$; $\bar{\mu}^B_{(a,(-)^\dagger)} = \mu_A$ and $\bar{\eta}^B_{(a,(-)^\dagger)} : (a, (-)^\dagger) \to \bar{T}_B(a, (-)^\dagger)$ with $\bar{\eta}^B_{(a,(-)^\dagger)} = \eta_A$.*

Hence, we have the following two adjoint situations: $\mathsf{C} \underset{\perp}{\rightleftarrows} \mathsf{Alg}_B(F) \underset{\perp}{\rightleftarrows} \mathcal{K}l(\bar{T}_B)$. These adjunctions impose a monadic structure on $TF^\infty : \mathsf{C} \to \mathsf{C}$. The monad $\mathcal{P}(\Sigma^* \times \mathcal{I}d + \Sigma^\omega)$ from Example 2.1 arises from the composition of the above adjoint situations (see also Example 4.3). It is important to note that since any Set-based monad T is strong, the functor $\Sigma \times \mathcal{I}d : \mathsf{Set} \to \mathsf{Set}$ always lifts to a functor on the Kleisli category for T. If we additionally assume T is a commutative monad then this is, in fact, true for any *polynomial functor* $F : \mathsf{Set} \to \mathsf{Set}$ [21], i.e. a functor defined by the grammar $F \triangleq \Sigma \in \mathsf{Set} \mid \mathcal{I}d \mid F \times F \mid \sum F$.

▶ **Example 4.7.** Let $F = \Sigma \times \mathcal{I}d^2$. Then $F^\infty = T_\Sigma(-)$ is a functor which assigns to any set X the set of complete binary trees (i.e. every node has either two children or no children) with inner nodes taking values in Σ and finitely many leaves, all taken from X [1]. This yields a monadic structure on $\mathcal{P}F^\infty = \mathcal{P}T_\Sigma$, where the Kleisli composition for $f : X \to \mathcal{P}T_\Sigma Y$ and $g : Y \to \mathcal{P}T_\Sigma Z$ is $g \cdot f : X \to \mathcal{P}T_\Sigma Z$ with $g \cdot f(x)$ being a set of trees obtained from trees in $f(x) \subseteq T_\Sigma Y$ by replacing any occurence of the leaf $y \in Y$ with a tree from $g(y) \subseteq T_\Sigma Z$. Let TTS^ω denote the theory associated with $\mathcal{P}T_\Sigma$. It is a simple exercise to prove that this category is order enriched with the order $f \leq g$ defined by $f(i) \subseteq g(i)$ for any $i \in [n]$ being complete, and that it is LD, ω-Cpo-enriched, and bRD.

▶ **Example 4.8.** For $T = \mathcal{CM}$ and $F = \Sigma \times \mathcal{I}d$ we get the monad $\mathcal{CM}(\Sigma^* \times \mathcal{I}d + \Sigma^\omega)$. The composition $\cdot$ for $f : X \to \mathcal{CM}(\Sigma^* \times Y + \Sigma^\omega)$ and $g : Y \to \mathcal{CM}(\Sigma^* \times Z + \Sigma^\omega)$ in its Kleisli category is as follows. If $\sum_{i=1}^n r_i \cdot (\sigma_i, y_i) + \sum_{i=n+1}^{n+k} r_i \cdot v_{i-n} \in f(x)$ and $\sum_{i=1}^{n_j} r_i^j \cdot (\sigma_i^j, z_i^j) + \sum_{i=n_j+1}^{n_j+k_j} r_i^j \cdot v_{i-n_j}^j \in g(y_j)$ for $j = 1, \ldots, n$, where $\sigma_i, \sigma_i^j \in \Sigma^*$ and $v_i, v_i^j \in \Sigma^\omega$, then the expression

$$\sum_{i=1}^n \left(\sum_{l=1}^{n_i} r_i \cdot r_l^i \cdot (\sigma_i \sigma_l^i, z_i^l) + \sum_{l=n_i+1}^{n_i+k_i} r_i \cdot r_l^i \cdot \sigma_i v_{l-n_i}^i \right) + \sum_{i=n+1}^{n+k} r_i \cdot v_{i-n}$$

is a member of the set $g \cdot f(x)$. The theory associated to this monad will be denoted by SGL^ω. It is order enriched with $f \leq g$ whenever $f(i) \subseteq g(i)$ for any i. For an arbitrary family of morphisms f_i their supremum $\bigvee_i f_i$ exists and is given by $\bigvee_i f_i(j) = \overline{\bigcup_i f_i(j)}$. Hence, the theory is complete with the infima $\bigwedge_i f_i(j) = \bigcap_i f_i(j)$. It is also LD, ω-Cpo-enriched, and bRD (the proof of this statement is analogous to the proof that the Kleisli categories for $\mathcal{CM}$ or $\mathcal{CM}(\Sigma^* \times \mathcal{I}d)$ have these properties [8, 9, 16] and, hence, is omitted).

5 Abstract (Büchi) automata and their behaviour

The purpose of this section is to generalize the concepts from Section 3 to an arbitrary theory with a suitable ordering. We start with the definition of an automaton for a theory $\mathbb{T}$.

▶ **Definition 5.1.** A $\mathbb{T}$-*automaton* or simply *automaton* is a pair $(\alpha, \mathfrak{F})$, where $\alpha : n \to n$ is an arbitrary endomorphism called *transition morphism* and $\mathfrak{F} \subseteq [n]$.

In order to define finite and infinite behaviour of $(\alpha, \mathfrak{F})$ we require the theory to satisfy more assumptions. An order enriched theory $\mathbb{T}$ is called *complete saturation theory* (or *CST* in short) provided that:

- **i** hom-posets are complete lattices,
- **ii** it is ω-Cpo-enriched, LD & bRD
- **iii** bottom maps 0 satisfy $f \cdot 0 = 0$ for any f,
- **iv** cotupling preserves the order.

From now on in this section we assume that $\mathbb{T}$ is a complete saturation theory. Note that the definition of a $\mathbb{T}$-automaton was stated in a more general framework. However, the finite and infinite behaviour of $(\alpha, \mathfrak{F})$ will be only considered for complete saturation theories.

▶ Remark. The assumption about completeness of the order, although a strong assumption, will guarantee existence of two types of fixpoints, namely $(-)^*$ and $(-)^\omega$. The former fixpoint operator was thoroughly studied in [7–10] in the context of coalgebraic weak bisimulation. Although it can be defined in an arbitrary completely ordered category, it requires left distributivity to be expressive enough [9] and ω-Cpo-enrichment to be calculated in terms of countable joins. Right distributivity w.r.t. the base morphisms is a technical assumption that is crucial in the proofs of theorems to come. This is a weak assumption as already discussed in [9, Lemma 3.25]. The bottom maps 0 provide us with a natural annihilator thanks to which given a set $\mathfrak{F} \subseteq [n]$ we can encode it as an endomorphism $\mathfrak{f}_{\mathfrak{F}} : n \to n$ defined as the cotuple of i_n's and 0_n^1's depending on whether the given coordinate is a member of $\mathfrak{F}$ or not. Finally, the last assumption guarantees that the order plays well with the coproduct.

For any endomorphism $\alpha : n \to n$ in $\mathbb{T}$ define $\alpha^*, \alpha^+ : n \to n$ and $\alpha^\omega : n \to 0$ by:

$$\alpha^* \triangleq \mu x.(\mathsf{id} \vee x \cdot \alpha), \quad \alpha^+ \triangleq \alpha^* \cdot \alpha \text{ and } \alpha^\omega \triangleq \nu x.x \cdot \alpha.$$

In a complete saturation theory we have $\alpha^* = \bigvee_{n<\omega}(\mathsf{id} \vee \alpha)^n$ [8] and $\alpha^\omega = \bigwedge_{\kappa \in \mathsf{Ord}}(\lambda x.x \cdot \alpha)^\kappa \top$, where $\top : n \to 0$ is the greatest element of $\mathbb{T}(n, 0)$ and $(\lambda x.x \cdot \alpha)^\kappa$ is defined by the transfinite induction by $(\lambda x.x \cdot \alpha)^{\kappa+1} = (\lambda x.x \cdot \alpha)(\lambda x.x \cdot \alpha)^\kappa$ for a successor ordinal $\kappa + 1$ and $(\lambda x.x \cdot \alpha)^\kappa = \bigwedge_{\lambda<\kappa}(\lambda x.x \cdot \alpha)^\lambda$ for a limit ordinal κ.

▶ **Theorem 5.2.** *For any $\alpha, \beta : n \to n$ we have:*

1. $\mathsf{id}^* = \mathsf{id}$, $\mathsf{id} \le \alpha^*$ *and* $\alpha^* \cdot \alpha^* = \alpha^*$,
2. $(\alpha \cdot \beta)^\omega = (\beta \cdot \alpha)^\omega \cdot \beta$,
3. $(\alpha^n)^\omega = \alpha^\omega$ *for any* $n > 0$,
4. $\alpha^\omega = (\alpha^+)^\omega$.

▶ **Definition 5.3.** *Finite behaviour* (ω*-behaviour*) $||(\alpha, \mathfrak{F})|| : n \to 1$ (resp. $||(\alpha, \mathfrak{F})||_\omega : n \to 0$) of an automaton $(\alpha, \mathfrak{F})$ is defined by:

$$||(\alpha, \mathfrak{F})|| \triangleq ! \cdot \mathfrak{f}_{\mathfrak{F}} \cdot \alpha^* \text{ and } ||(\alpha, \mathfrak{F})||_\omega \triangleq (\mathfrak{f}_{\mathfrak{F}} \cdot \alpha^+)^\omega.$$

Finite (ω-)behaviour of a state $i_n : 1 \to n$ of $(\alpha, \mathfrak{F})$ is $||(\alpha, \mathfrak{F})|| \cdot i_n$ (resp. $||(\alpha, \mathfrak{F})||_\omega \cdot i_n$).

▶ **Example 5.4.** The theories LTS^ω, TTS^ω and SGL^ω are complete saturation theories. As we have already seen in Section 3, the finite and ω-behaviour of LTS^ω-automata coincides with the classical notions. The same can be easily proven to be true for TTS^ω (see e.g. [31] for classical definitions in the theory of tree automata). According to our knowledge, *Segala automata* or, in other words, SGL^ω-automata, have not been considered in the computer science literature so far. See Subsection 5.1 for a discussion on these systems in the context of expressivity in language theory.

▶ Remark. So far in the coalgebraic literature, finite behaviour of systems was introduced in terms of the finite trace [6,26,35]. In the order enriched setting for systems with internal moves for which the type functor encodes accepting states, finite trace is given by $\alpha^\dagger = \mu x.x \cdot \alpha$ [7]. However, from the point of our setting, the terminal states are not part of the transition. In this case we can consider the *exception monad* $\mathcal{I}d + 1$ on any theory $\mathbb{T}$, denote its associated theory by $\widehat{\mathbb{T}}$, and encode any $\mathbb{T}$-automaton $(\alpha, \mathfrak{F})$ as a $\widehat{\mathbb{T}}$-endomorphism $\widehat{\alpha} : n \to n$ (or equivalently $\mathbb{T}$-morphism $\widehat{\alpha} : n \to n+1$) defined by $\widehat{\alpha} = \mathsf{in}^n_{n+1} \cdot \alpha \vee \widehat{\mathfrak{f}_{\mathfrak{F}}}$, where $\widehat{\mathfrak{f}_{\mathfrak{F}}} : n \to n+1$ is a morphism in $\mathbb{T}$ given by $\widehat{\mathfrak{f}_{\mathfrak{F}}}(i) =$ **if** $i \in \mathfrak{F}$ **then** $n+1$ **else** 0. It is a simple exercise to prove that, given the assumptions of this section, $\widehat{\alpha}^\dagger = ||(\alpha, \mathfrak{F})||$. Therefore, our definition of finite behaviour via $(-)^*$ coincides with the trace definition in an ordered category [6].

Kleene theorems. The prominent role in the theory of non-deterministic automata is played by regular languages. Using the nomenclature of Section 3 these languages are given by $! \cdot \mathfrak{f} \cdot \alpha^* \cdot i_n : 1 \to 1$ for an LTS^ω automaton $(\alpha, \mathfrak{F})$ in which we have $\alpha : [n] \to \mathcal{P}(\Sigma_\varepsilon \times [n])$. The set of regular languages, denoted by $\mathfrak{Reg}(1,1)$, is known to be closed under the language composition, finite union and Kleene star operation. These three operations are exactly the composition, finite joins and the saturation of morphisms $1 \to 1$ in the theory LTS^ω. Moreover, $\mathfrak{Reg}(1,1)$ is the smallest set of languages containing the empty language, single letter languages and being closed under the three operations. This classical result is known under the name of *Kleene theorem for regular languages* [22]. A similar theorem can be proven for automata that accept non-sequential data types, e.g. trees [17,31]. However for tree automata the result is slightly more involved as the set $\mathfrak{Reg}(1,1)$ of regular tree languages is closed under a more complex type of composition, namely the composition of regular tree languages with multiple variables. To be more precise, if $\mathfrak{Reg}(1,p)$ denotes the set of regular tree languages whose leaves may end in variables from $\{1,\ldots,p\}$, then the morphism $[r_1,\ldots,r_p] \cdot r$ is a member of $\mathfrak{Reg}(1,1)$ for any $r \in \mathfrak{Reg}(1,p)$ and $r_i \in \mathfrak{Reg}(1,1)$. These observations are generalized to the coalgebraic level below. As a direct consequence of this treatment we get a characterization of ω-regular behaviours.

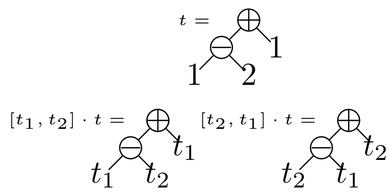


Figure 6 Tree composition with inner nodes in $\{+,-\}$ and variables in $\{1,2\}$.

Let $T = (T,\mu,\eta)$ be a monad on Set and F a Set-endofunctor satisfying the assumptions of Section 4. This allows us to consider the monad TF^∞ and the theory $\mathbb{T}_{TF^\infty}$ associated with it. We say that a map $\alpha : m \to n$ in $\mathbb{T}_{TF^\infty}$ is a *(T, F_ε)-map* if $\alpha : [m] \to TF_\varepsilon[n]$ in Set (it is a well defined notion as F_ε is a subfunctor of F^∞). Note that by the definition of F^∞ the family of (T,F_ε)-maps contains all base maps of $\mathbb{T}_{TF^\infty}$ and is closed under cotupling and the composition with base morphisms (it follows by the definition of the monadic structure of F^∞, TF^∞ and Remark in Subsection 4.1). $\mathbb{T}_{TF^\infty}$-automata whose transition maps are (T,F_ε)-maps will be referred to as *(T,F_ε)-automata.* In this paragraph we assume that $\mathbb{T}_{TF^\infty}$ is a CST and:

- (T, F_ε)-maps are closed under taking arbitrary suprema (hence, also contain 0's),
- $0 \cdot \alpha = 0$ for any (T, F_ε)-map α.

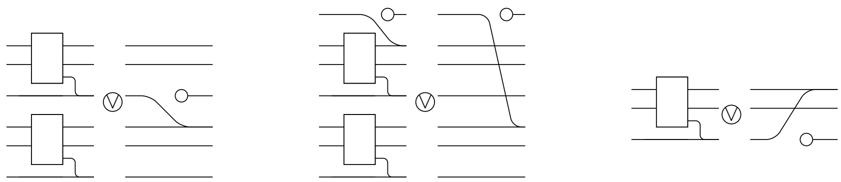
As a direct consequence of these assumptions and since id is a base morphism we get that $0 \cdot \alpha^* = \alpha^* \cdot 0 = 0$ for any (T, F_ε)-map α which is a $\mathbb{T}_{TF^\infty}$-endomorphism. We define the set of *regular morphisms* $m \to p$ by:

$$\mathfrak{Reg}(m,p) \triangleq \{j' \cdot \mathfrak{f}_{\mathfrak{F}} \cdot \alpha^* \cdot j \mid (\alpha : n \to n, \mathfrak{F}) \text{ is a } (T, F_\varepsilon)\text{-aut. and } j : m \to n, j' : n \to p \text{ are base maps}\}.$$

The set of regular morphisms $\mathfrak{Reg}(1,p)$ will be often referred to as the set of *regular trees with variables in* p. Note that $\mathfrak{Reg}(1,1)$ is exactly the set of finite behaviours of states in (T, F_ε)-automata. A regular morphism $r \in \mathfrak{Reg}(m,p)$ is said to be in *normal form* (NF) if it is given by $r = [0_p^n, \mathsf{id}_p] \cdot [\alpha, \mathsf{in}_{n+p}^p]^* \cdot \mathsf{in}_{n+p}^m$ for a (T, F_ε)-map $\alpha : n \to n + p$ and $m \leq n$. The following lemma states that all regular morphisms can be given in their normal forms and that they can be obtained from regular trees via cotupling.

▶ **Lemma 5.5.** *The following equality is true:* $\mathfrak{Reg}(m,p) = \{[r_1, \ldots, r_m] \mid r_i \in \mathfrak{Reg}(1,p)\} = \{[0_p^n, \mathsf{id}_p] \cdot [\alpha, \mathsf{in}_{n+p}^p]^* \cdot \mathsf{in}_{n+p}^m \mid \alpha : n \to n + p$ *is a* (T, F_ε)*-map and* $m \leq n\}$.

The next results (Lem. 5.6 and Th. 5.7) show, in particular, that regular morphisms with suitable domains and codomains are closed under composition, finite joins and saturation operation. The constructions used in the proofs of the results below are simple generalization of classical constructions of non-deterministic automata with ε-moves used in proving that concatenation/finite union/Kleene star of regular languages is regular (see e.g. [22]). Hence, in our opinion, it can be considered a computer science folklore which presents itself very aesthetically in terms of the string diagram calculus. Note that for classical regular languages it was enough to consider the case where the normal form $[0, \mathsf{id}_p] \cdot [\alpha, \mathsf{in}^p]^* \cdot \mathsf{in}^m$ of the expressions satisfied $m = p = 1$ (i.e. one initial and one final state). These constructions can be summarized by the following three diagrams.



$r_1 \cdot r_2$ $\qquad$ $r_1 \vee r_2$ $\qquad$ r^*

▶ **Lemma 5.6.** *The identity maps in* $\mathbb{T}_{TF^\infty}$ *are regular morphisms. Moreover, regular morphisms are closed under the composition from* $\mathbb{T}_{TF^\infty}$.

Let $\mathfrak{Reg}(T,F)$ be the category whose objects are the same as the objects of $\mathbb{T}_{TF^\infty}$ and whose hom-sets are $\mathfrak{Reg}(m,n)$ with the composition taken from $\mathbb{T}_{TF^\infty}$. By the above lemmas this definition is proper and, moreover, $\mathfrak{Reg}(T,F)$ is a theory. It is order enriched with the order from $\mathbb{T}_{TF^\infty}$. Moreover, the following statement holds.

▶ **Theorem 5.7** (Kleene thm. for regular behaviour)**.** $\mathfrak{Reg}(T,F)$ *is an ordered theory which:*

(a) *contains all* (T, F_ε)*-maps,*

(b) *admits finite suprema and each hom-set contains the bottom element,*

(c) *endomorphisms are closed under* $(-)^*$.

If $\mathfrak{Rat}(T,F)$ *is the smallest subtheory of* $\mathbb{T}_{TF^\infty}$ *satisfying (a)-(c) then* $\mathfrak{Rat}(T,F) = \mathfrak{Reg}(T,F)$.

Finally, put $\omega\mathfrak{Rat}(T,F) \triangleq \{[r_1,\ldots,r_m]^\omega \cdot r \mid r \in \mathfrak{Rat}(1,m), r_i \in \mathfrak{Rat}(1,m) \text{ for } m<\omega\}$ and

$$\omega\mathfrak{Reg}(T,F) \triangleq \{||(\alpha,\mathfrak{F})||_\omega \cdot i_m : 1 \to 0 \mid (\alpha,\mathfrak{F}) \text{ is } \mathbb{T}_{TF^\infty}\text{-aut. with } (T,F_\varepsilon)\text{-map } \alpha : m \to m\}.$$

▶ **Theorem 5.8** (Kleene thm. for ω-regular behaviour)**.** *We have* $\omega\mathfrak{Rat}(T,F) = \omega\mathfrak{Reg}(T,F)$.

5.1 Behaviours v. languages

The purpose of this subsection is to define *languages* for $\mathbb{T}_{TF^\infty}$-automata. Unlike behaviours, languages are simply subsets of $F^\infty 1$. As we will see below there is a natural way to introduce such languages for abstract automata. The theory presented in this subsection is motivated by (ω-)languages of probabilistic (ω-)automata [3] defined using a construction akin to the one presented below. However, since fully probabilistic automata are not considered in our paper (see the summary section for details), we will focus our attention on SGL^ω-automata and the languages they generate. We will show that the classes of (ω-)regular languages of these machines coincide with the class of (ω-)regular languages in the classical sense.

Let T be a functor on Set and consider the transformation $\tau : T \implies \mathcal{P}$ whose X-component is defined by $\tau_X(t) = \bigcap\{Y \subseteq X \mid t \in TY\}$. If T preserves preimages and infinite intersections then the transformation is natural [19]. Here, we assume it is the case.

▶ **Example 5.9.** For $T = \mathcal{CM}$ the transformation τ is given by:

$$\tau_X(U) = \bigcup\{\{x_1,\ldots,x_n\} \mid r_1\cdot x_1 + \ldots + r_n \cdot x_n \in U \text{ for } r_i > 0\} \text{ for } U \in \mathcal{CM}X.$$

It is a simple exercise to prove that $\tau : \mathcal{CM} \implies \mathcal{P}$ is a natural transformation.

Let T and F and $\mathcal{P}$ and F satisfy the assumptions of Section 4. Then the transformation $\tau : T \implies \mathcal{P}$ imposes an assignment τ_{F^∞} between theories $\mathbb{T}_{TF^\infty}$ and $\mathbb{T}_{\mathcal{P}F^\infty}$ given by: $\tau_{F^\infty}(n) \triangleq n$ and $\tau_{F^\infty}(f : m \to TF^\infty m) \triangleq \tau_{F^\infty n} \circ f$. Assume that both $\mathbb{T}_{TF^\infty}$ and $\mathbb{T}_{\mathcal{P}F^\infty}$ are complete saturation theories. Given a $\mathbb{T}_{TF^\infty}$-automaton $(\alpha : n \to n, \mathfrak{F})$ and $i \in [n]$, we define its *language* (resp. ω*-language*) by $\mathcal{L}(\alpha,\mathfrak{F},i) \triangleq \tau_{F^\infty}(||(\alpha,\mathfrak{F})|| \cdot i_n)$ and $\mathcal{L}^\omega(\alpha,\mathfrak{F},i) \triangleq \tau_{F^\infty}(||(\alpha,\mathfrak{F})||_\omega \cdot i_n)$. Now, the sets of *regular* and ω*-regular languages* for $\mathbb{T}_{TF^\infty}$ are $\mathfrak{LReg}(T,F) \triangleq \{(\mathcal{L}(\alpha,\mathfrak{F},i) \mid \alpha \text{ is a } (T,F_\varepsilon)\text{-map}\}$ and

$$\omega\mathfrak{LReg}(T,F) \triangleq \{\mathcal{L}^\omega(\alpha,\mathfrak{F},i) \mid \alpha \text{ is a } (T,F_\varepsilon)\text{-map}\}.$$

▶ **Theorem 5.10.** *If* $\tau_{F^\infty} : \mathbb{T}_{TF^\infty} \to \mathbb{T}_{\mathcal{P}F^\infty}$ *is a functor which preserves cotupling, preserves* 0*'s, finite suprema and suprema of* ω*-chains then* $\mathfrak{LReg}(T,F) \subseteq \mathfrak{Reg}(\mathcal{P},F)(1,1)$*. Moreover, if* $\tau_{F^\infty}(\beta^\omega) = \tau_{F^\infty}(\beta)^\omega$ *for any* $\mathbb{T}_{TF^\infty}$*-endomorphism* β *of the form* $\mathfrak{f}_{\mathfrak{F}} \cdot \alpha^+$ *for a* (T,F_ε)*-map* α *then* $\omega\mathfrak{LReg}(T,F) \subseteq \omega\mathfrak{Reg}(\mathcal{P},F)$*.*

▶ **Example 5.11.** The assignment $\mathsf{SGL}^\omega \to \mathsf{LTS}^\omega$ induced by the natural transformation from Example 5.9 satisfies the assumptions of the first part of Th. 5.10. Additionally, it preserves the assumptions of the second part of this statement, and, hence, from the point of view of regular and ω-regular languages Segala automata are equally expressive as the non-deterministic (Büchi) automata.

6 Summary, future and related work

The purpose of this paper was to develop a coalgebraic (categorical) framework to reason about abstract automata and their finite and infinite behaviours satisfying BAC. We achieved this goal by constructing a monad suitable to handle the types of behaviours we were interested in and defining them in the right setting. A natural and direct consequence of this treatment was Theorem 5.7 and Theorem 5.8, i.e. the coalgebraic characterization of regular and ω-regular behaviour. These two results are the main reason why the primary interest of this paper is the Set-based finite structures. Note that several definitions and properties of Section 5 generalize to systems whose type monad is over a different category than Set (in this case an *automaton* should be simply defined as a pair of endomorphisms in the given Kleisli category).

Seemingly, the main restrictions of this framework are hidden behind the assumptions in the definition of a complete saturation theory. However, many of these axioms can be relaxed. For instance, in case of lack of left distributivity we may use a construction from our previous work [9] which embeds suitably ordered categories into left distributive ones. Secondly, the assumption about completeness of the order may be replaced with the assumption about existence of $(-)^*$ and $(-)^\omega$ satisfying the desired properties (note that the theory $\mathfrak{Reg}$ defined in Section 5 is not necessarily complete, yet finite joins, $(-)^*$ and $(-)^\omega$ are well defined).

Future work. We plan that the next step from here will be to put fully probabilistic automata into our framework, as this type of machines and their ω-languages play a significant role in infinite language theory [2]. Probabilistic systems have been successfully put into the saturation and weak bisimulation framework by embedding the category these systems are described in, into a category which admits left distributivity [9].

Given our natural characterization of coalgebraic ω-regular languages we ask if it is possible to characterize it in an *algebraic* way in terms of a preimage of a subset of a finite algebraic structure. Especially, considering the fact that by Th. 5.2 the pair of hom-sets $(\mathbb{T}(n,n),\mathbb{T}(n,0))$ equipped with suitable operations resembles a Wilke algebra used in the algebraic characterization of these languages (see e.g. [31] for details).

Related work. The first coalgebraic take on ω-languages was presented in [11], where authors put deterministic Muller automata with Muller acceptance condition into the framework. Our work is related to a more recent paper [36], where Urabe *et al.* give a coalgebraic framework for modelling behaviour with Büchi acceptance condition for (T,F)-systems. The main ingredient of their work is a solution to a system of equations which uses least and greatest fixpoints. This is done akin to Park's [30] classical characterization of ω-languages via a system of equations. In our paper we also use least and greatest fixpoints, however, the operators we consider are the two natural types of operators $(-)^* = \mu x.\mathsf{id} \vee x \cdot (-)$ and $(-)^\omega = \nu x.x \cdot (-)$ which generalize the language operators $(-)^*$ and $(-)^\omega$ known from the classical theory of regular and ω-regular languages. By calculating everything in the Kleisli category for the given monad and by using the aforementioned operators we simplify the language considerably. This allows us to state and prove Kleene-type theorems for (ω-)regular input which was not achieved in [36] and (in our opinion) would be difficult to obtain in that setting. To summarize, the major differences between our work and [36] are the following:

- we use the setting of systems with internal moves (i.e. coalgebras over a monad) to discuss infinite behaviour with BAC,
- the infinite behaviour with BAC is calculated in terms of a simple expression which uses $(-)^*$ and $(-)^\omega$ in the Kleisli category,
- we provide the definition of a finite behaviour of a system (using $(-)^*$) and build a bridge between regular and ω-regular behaviours on a coalgebraic level in terms of the Kleene theorem.

Abstract finite automata have already been considered in the computer science literature in the context of Lawvere iteration theories with analogues of Kleene theorems stated and proven (see e.g. [5, 13–15]). Some of these results seem to be presented in a more general setting than ours, using a slightly different language than ours (*conf.* Theorem 5.7 and e.g. [5, Theorem 1.4]). We decided to state Theorem 5.7 the way we did, in order to make a direct generalization of the classical Kleene theorem for regular input and to give a coalgebraic interpretation which is missing in [5, 13–15]. We should also mention that the infinite behaviour with BAC was defined in *loc. cit.* only for a very specific type of theories (i.e. the matricial theories over an algebra with an infinite iteration operator), which do not encompass e.g. non-deterministic Büchi tree automata and their infinite tree languages.

References

1. Jirí Adámek, Mahdieh Haddadi, and Stefan Milius. Corecursive algebras, corecursive monads and bloom monads. *Logical Methods in Computer Science*, 10(3), 2014. `doi:10.2168/LMCS-10(3:19)2014`.
2. Christel Baier and Marcus Grosser. Recognizing omega-regular languages with probabilistic automata. In *Proceedings of the 20th Annual IEEE Symposium on Logic in Computer Science*, LICS '05, pages 137–146, Washington, DC, USA, 2005. IEEE Computer Society. `doi:10.1109/LICS.2005.41`.
3. Christel Baier, Marcus Grösser, and Nathalie Bertrand. Probabilistic omega-automata. *J. ACM*, 59(1):1:1–1:52, 2012. `doi:10.1145/2108242.2108243`.
4. Jon Beck. Distributive laws. In B. Eckmann, editor, *Seminar on Triples and Categorical Homology Theory*, pages 119–140, Berlin, Heidelberg, 1969. Springer Berlin Heidelberg.
5. Stephen Bloom and Zoltán Ésik. *Iteration Theories. The Equational Logic of Iterative Processes.* Monographs in Theoretical Computer Science. Springer, 1993.
6. Filippo Bonchi, Stefan Milius, Alexandra Silva, and Fabio Zanasi. Killing epsilons with a dagger: A coalgebraic study of systems with algebraic label structure. *Theoretical Computer Science*, 604:102–126, 2015. `doi:10.1016/j.tcs.2015.03.024`.
7. Tomasz Brengos. On coalgebras with internal moves. In Marcello M. Bonsangue, editor, *Proc. CMCS*, Lecture Notes in Computer Science, pages 75–97. Springer, 2014. `doi:10.1007/978-3-662-44124-4_5`.
8. Tomasz Brengos. Weak bisimulation for coalgebras over order enriched monads. *Logical Methods in Computer Science*, 11(2):1–44, 2015. `doi:10.2168/LMCS-11(2:14)2015`.
9. Tomasz Brengos, Marino Miculan, and Marco Peressotti. Behavioural equivalences for coalgebras with unobservable moves. *Journal of Logical and Algebraic Methods in Programming*, 84(6):826–852, 2015. `doi:10.1016/j.jlamp.2015.09.002`.
10. Tomasz Brengos and Marco Peressotti. A Uniform Framework for Timed Automata. In Josée Desharnais and Radha Jagadeesan, editors, *27th International Conference on Concurrency Theory (CONCUR 2016)*, volume 59 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 26:1–26:15, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. `doi:10.4230/LIPIcs.CONCUR.2016.26`.
11. Vincenzo Ciancia and Yde Venema. Stream automata are coalgebras. In *Proc. CMCS*, volume 7399 of *Lecture Notes in Computer Science*, pages 90–108, 2012. `doi:10.1007/978-3-642-32784-1_6`.
12. Corina Cîrstea. Generic infinite traces and path-based coalgebraic temporal logics. *Electr. Notes Theor. Comput. Sci.*, 264(2):83–103, 2010. `doi:10.1016/j.entcs.2010.07.015`.
13. Zoltán Ésik and Tamás Hajgató. Iteration grove theories with applications. In *Proc. Algebraic Informatics*, volume 5725 of *Lecture Notes in Computer Science*, pages 227–249. Springer, 2009. `doi:10.1007/978-3-642-03564-7_15`.

14 Zoltán Ésik and Werner Kuich. *A Unifying Kleene Theorem for Weighted Finite Automata*, pages 76–89. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. doi:10.1007/978-3-642-19391-0_6.

15 Zoltán Ésik and Werner Kuich. Modern automata theory, 2013. URL: http://www.dmg.tuwien.ac.at/kuich/.

16 Sergey Goncharov and Dirk Pattinson. Coalgebraic weak bisimulation from recursive equations over monads. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Proc. ICALP*, volume 8573 of *Lecture Notes in Computer Science*, pages 196–207. Springer, 2014. doi:10.1007/978-3-662-43951-7_17.

17 Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata Logics, and Infinite Games: A Guide to Current Research*, page 392. Springer-Verlag New York, Inc., New York, NY, USA, 2002.

18 H. Peter Gumm. *Elements of the general theory of coalgebras.* LUATCS 99, Rand Afrikaans University, 1999.

19 H. Peter Gumm. From t-coalgebras to filter structures and transition systems. In José Luiz Fiadeiro, Neil Harman, Markus Roggenbach, and Jan Rutten, editors, *Algebra and Coalgebra in Computer Science*, pages 194–212, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. doi:doi.org/10.1007/11548133_13.

20 Ichiro Hasuo. Generic forward and backward simulations. In *Prof. CONCUR*, volume 4137 of *Lecture Notes in Computer Science*, pages 406–420, 2006. doi:10.1007/11817949_27.

21 Ichiro Hasuo, Bart Jacobs, and Ana Sokolova. Generic trace semantics via coinduction. *Logical Methods in Computer Science*, 3(4), 2007. doi:10.2168/LMCS-3(4:11)2007.

22 John E. Hopcroft, Rajeev Motwani, Rotwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages and Computability.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2000.

23 Martin Hyland and John Power. The category theoretic understanding of universal algebra: Lawvere theories and monads. *Electronic Notes in Theoretical Computer Science*, 172:437–458, 2007. doi:10.1016/j.entcs.2007.02.019.

24 Bart Jacobs. Trace semantics for coalgebras. *Electr. Notes Theor. Comput. Sci.*, 106:167–184, 2004. doi:/doi.org/10.1016/j.entcs.2004.02.031.

25 Bart Jacobs. Coalgebraic trace semantics for combined possibilistic and probabilistic systems. In *Proc. CMCS*, Electronic Notes in Theoretical Computer Science, pages 131–152, 2008. doi:10.1016/j.entcs.2008.05.023.

26 Bart Jacobs, Alexandra Silva, and Ana Sokolova. Trace semantics via determinization. In *Proc. CMCS*, volume 7399 of *Lecture Notes in Computer Science*, pages 109–129, 2012. doi:10.1007/978-3-642-32784-1_7.

27 F. W. Lawvere. Functorial semantics of algebraic theories. *Proc. Nat. Acad. Sci. U.S.A.*, 50:869–872, 1963. doi:10.1073/pnas.50.5.869.

28 Saunders Mac Lane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics*. Springer-Verlang New York, 1978. doi:10.1007/978-1-4757-4721-8.

29 Robin Milner. *Communication and Concurrency.* Prentice-Hall, 1989.

30 David Park. Concurrency and automata on infinite sequences. In Peter Deussen, editor, *Theoretical Computer Science*, pages 167–183, Berlin, Heidelberg, 1981. Springer Berlin Heidelberg.

31 Jean-Eric Pin and Dominique Perrin. *Infinite Words: Automata, Semigroups, Logic and Games*, page 538. Elsevier, 2004.

32 Jan J. M. M. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science*, 249(1):3–80, 2000. doi:10.1016/S0304-3975(00)00056-6.

33 Roberto Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems.* PhD thesis, Massachusetts Institute of Technology, 1995.

34 Roberto Segala and Nancy Lynch. Probabilistic simulations for probabilistic processes. In *Proc. CONCUR*, volume 836 of *Lecture Notes in Computer Science*, pages 481–496, 1994. doi:10.1007/978-3-540-48654-1_35.

35 Alexandra Silva and Bram Westerbaan. A coalgebraic view of ϵ-transitions. In Reiko Heckel and Stefan Milius, editors, *Proc. CALCO*, volume 8089 of *Lecture Notes in Computer Science*, pages 267–281. Springer, 2013. doi:10.1007/978-3-642-40206-7_20.

36 Natsuki Urabe, Shunsuke Shimizu, and Ichiro Hasuo. Coalgebraic Trace Semantics for Buechi and Parity Automata. In Josée Desharnais and Radha Jagadeesan, editors, *27th International Conference on Concurrency Theory (CONCUR 2016)*, volume 59 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 24:1–24:15, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.CONCUR.2016.24.