Ω-Automata: A Coalgebraic Perspective on Regular ω-Languages

Ciancia, V.; Venema, Y.

DOI
10.4230/LIPIcs.CALCO.2019.5

Publication date
2019

Document Version
Final published version

Published in
8th Conference on Algebra and Coalgebra in Computer Science

Ω-Automata: A Coalgebraic Perspective on Regular ω-Languages

Vincenzo Ciancia
Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo" - Consiglio Nazionale delle Ricerche, Pisa, Italy
vincenzo.ciancia@isti.cnr.it

Yde Venema
Institute for Logic, Language and Computation, Universiteit van Amsterdam, Amsterdam, The Netherlands
`https://staff.fnwi.uva.nl/y.venema/`
y.venema@science.uva.nl

Abstract

In this work, we provide a simple coalgebraic characterisation of regular ω-languages based on languages of *lassos*, and prove a number of related mathematical results, framed into the theory of a new kind of automata called Ω-*automata*. In earlier work we introduced Ω-automata as two-sorted structures that naturally operate on *lassos*, pairs of words encoding ultimately periodic streams (infinite words). Here we extend the scope of these Ω-automata by proposing them as a new kind of acceptor for *arbitrary* streams. We prove that Ω-automata are expressively complete for the regular ω-languages. We show that, due to their coalgebraic nature, Ω-automata share some attractive properties with deterministic automata operating on finite words, properties that other types of stream automata lack. In particular, we provide a simple, coalgebraic definition of bisimilarity between Ω-automata that exactly captures language equivalence and allows for a simple minimization procedure. We also prove a coalgebraic Myhill-Nerode style theorem for lasso languages, and use this result, in combination with a closure property on stream languages called *lasso determinacy*, to give a characterization of regular ω-languages.

2012 ACM Subject Classification Theory of computation → Formal languages and automata theory; Theory of computation → Automata over infinite objects

Keywords and phrases ω-automata, regular ω-languages, coalgebra, streams, bisimilarity

Digital Object Identifier 10.4230/LIPIcs.CALCO.2019.5

Acknowledgements The authors wish to thank Oded Maler and Ludwig Staiger for useful discussions about right congruences and characterisations of ω-regularity.

1 Introduction

The theory of finite automata, seen as devices for classifying (possibly) infinite structures [11], combines a rich mathematical theory, dating back to the seminal work of Büchi, Rabin, and others, with a wide range of applications, in areas related to the verification and synthesis of systems that are not supposed to terminate. This applies in particular to automata operating on streams (infinite words): stream automata (or ω-automata), see [15] for a comprehensive reference. Stream automata can be classified in terms of their acceptance conditions (e.g. parity, Muller, Büchi), and come in deterministic, nondeterministic and alternating variants. With the exception of the weaker deterministic Büchi automata, these models all recognize the same class of stream languages (or ω-languages), viz., the *regular* ones.

Our perspective on stream automata and regular ω-languages will be coalgebraic. Universal Coalgebra [17] is a mathematical, category-based theory of evolving state-based systems such as streams, (infinite) trees, Kripke models, (probabilistic) transition systems, and many others. This approach combines simplicity with generality and wide applicability:

many features, including input, output, nondeterminism, probability, and interaction, can easily be encoded in the coalgebra type, which is formally an endofunctor T on some base category C (often Set, the category with sets as objects and functions as arrows). When it comes to the coalgebraic perspective on automata theory, the standard deterministic finite automata (DFAs) operating on *finite* words have been recognized as paradigmatic examples of coalgebras [16]: Standard coalgebraic concepts such as final coalgebras and bisimulations, feature very naturally in the theory of DFAs. For instance, language equivalence between DFA states is captured exactly by the notion of bisimilarity, an observation yielding both a coinductive procedure to decide language equivalence and an elegant method for minimizing automata; states of the final coalgebra also represent equivalence classes of the celebrated Myhill-Nerode theorem. These observations naturally raise the question whether automata operating on streams admit an attractive coalgebraic presentation as well.

Exactly this problem was addressed in our earlier paper [9]. Based on the well-known characterization of regular ω-languages by their ultimately periodic (UP) fragment, we focused on finite representations of UP streams called *lassos*: a lasso is a pair (u, v) of a finite word u and a finite nonempty word v, representing the UP stream uv^ω. This approach built on the work of Calbrix, Nivat & Podelski [7], who introduced certain DFAs operating on finite words of the form $u\$v$, where \$ is a special symbol separating the spoke u from the loop v of a lasso. The contribution of [9] was threefold. First, we introduced *two-sorted* automata operating on lassos directly, and we showed that these lasso automata share some nice properties with standard DFAs. Second, we presented these lasso automata as coalgebras, for a functor Ω on the category Set^2 of two-sorted sets with two-sorted functions. And third, we identified two properties, *coherence* and *circularity*, that characterize those Ω-coalgebras of which the recognized lasso language correspond to the UP fragment of a regular ω-language.

Where our discussion in [9] stayed at a fairly abstract level, and we only considered acceptance of lassos, the current paper shows how to make concrete use of Ω-coalgebras[1] as automata operating on arbitrary streams. For this purpose, we introduce a new kind of stream automaton by defining an Ω-*automaton* as a circular and coherent lasso automaton, and endowing these structures with a suitable notion of acceptance for streams.

Contribution

The technical results that we prove on these Ω-automata include the following:

- we prove that the property of being an Ω-automaton is decidable; that is, we show that it is decidable whether a given lasso automaton is circular and coherent (Theorem 18);
- we show that, with respect to expressive power, Ω-automata exactly capture the regular ω-languages (Theorem 22 and Corollary 24);
- we show that for Ω-automata, the natural (and coalgebraic) notion of bisimilarity exactly captures language equivalence (Theorem 25), and
- as a corollary we obtain a simple and natural minimization procedure for Ω-automata (Theorem 28), which is an instance of the well-known coalgebraic partition refinement;
- we prove a Myhill-Nerode theorem for lassos (Theorem 34), which is closely related to the work of Maler & Staiger [14], but has a coalgebraic component involving the final Ω-coalgebra;
- as a corollary of this, we give a new characterization of regular ω-languages (Theorem 37).

[1] In fact, the automata that we consider in this paper are a simplification of the ones defined in [9], see Remark 3. All results proved in [9] also apply in this setting, with only trivial modifications to the proofs.

Combining these observations with the results obtained in [9], we find that Ω-automata share many of the attractive properties of DFAs: bisimulation captures language equivalence and can be used for a minimization procedure, regular ω-languages correspond to the finitely generated subcoalgebras of the final coalgebra, Boolean operations (including complementation) on regular ω-languages can be implemented by straighforward operations on Ω-coalgebras, etc. This is in sharp contrast to the setting of standard devices such as Büchi or parity automata, where to the best of our knowledge no satisfactory notion of bisimilarity between automata has been defined.

The main point of this article, then, lies not so much in the above list of individual contributions, but in the picture we obtain by putting all technical results (from this paper and from [9]) *together*. The emerging picture shows that stream automata and (regular) ω-languages *do* fit in a coherent coalgebraic framework, and one that shares many of the attractive properties of DFAs and regular languages of finite words.

Related work. In between the publication of [9] and this work, some relevant results on coalgebraic interpretations of stream automata appeared in the literature. The research line of [20, 19] follows the so-called Kleisli approach to trace semantics of coalgebras. Here a system is a coalgebra in the Kleisli category associated with some monad T that encodes the branching style exhibited by the system. In such a Kleisli category the homsets are often equipped with a natural order relation, and the authors use this fact to characterize the behaviour of parity-style automata as an arrow that is a solution of some hierarchical equation system over this order. Proving the effectiveness of their definition, the authors are able to capture not only regular ω-languages, but also various forms of so-called *ω-regular behavioural equivalences* of a labelled graph, including *infinite trace semantics*, *tree languages*, and systems with both non-deterministic and probabilistic branching. In [6] similar results are developed for systems with so-called *internal moves* and the corresponding notion of *weak bisimilarity*.

Attractive about this perspective is the modularity of the coalgebraic approach, which permits results to be generalised to various interpretations of the notion "(ω-regular) behaviour", and its clear link to the research area of *process calculi* and the categorical modelling of their behavioural equivalences; however, such abstract representations are not directly exploitable for the verification of properties related to such equivalences. In contrast, our approach is framed in the logical-algorithmic view of automata, and therefore aimed at simple, finite representations, and algorithms such as minimization (see Section 4.4) or Boolean operations (defined in [9], including complementation) that are also a typical ingredient of model checkers. This does *not* mean that our presentation lacks generality; for instance, changing the base category is an interesting possibility to explore (consider e.g. [8], defining a class of *nominal* omega-regular languages, notably also characterised by a form of ultimately periodic behaviour, see Theorem 7 therein).

To mention some closely related work in a different direction, a well-known algorithmic application in language theory is *automata learning* [1]. It is noteworthy that, in fact, a class of finite-state machines [2], similarly inspired by [14], has been used by Angluin and Fisman for learning ω-regular languages [3], and exploited by an independent group in a tool that obtained best-in-class results in terms of computational efficiency [12]. We believe that the coalgebraic perspective brings in some unique improvements on its own; for instance, minimization and Myhill-Nerode style theorems are a standard consequence of the theory; furthermore, the theory of coalgebras opens up to the possibility of generalising the presented constructions, and derive new ones (more on this in the Conclusions).

2 Preliminaries

Assuming that the reader is familiar with the theory of automata operating on (in)finite words [15], we fix some notation and terminology.

Given sets X, Y and Z, we define Y^X as the function space of maps from X to Y, and using currying we may identify the sets $Z^{X\times Y}$ and $(Z^Y)^X$.

Throughout this paper we fix a finite alphabet C consisting of symbols or colors. We write C^* (C^+) for the set of finite (respectively, finite nonempty) *words* over C. The empty word is denoted as ϵ, and the length of a word u as $|u|$. With ω denoting the set of natural numbers, C^ω is the collection of *streams* (*infinite words*) over C. The binary operation of concatenation between finite and (in)finite words is denoted by juxtaposition. For $u \in C^+$, we let u^ω denote the stream that repeats u ω many times, and for $(v_i)_{i\in\omega}$ in C^+ we write $\vec{v}$ for the stream $v_0v_1\cdots$. A stream α is *ultimately periodic* if it is of the form uv^ω for some $u \in C^*$ and $v \in C^+$. A *language* is a set of finite words; an *ω-language* or *stream language* is a set of streams. The *ultimately periodic fragment* $UP(L)$ of a stream language L is the set of its ultimately periodic members.

A *transition function* on a set X is a map of the form $\rho : X \times C \to X$ (or, equivalently, $\rho : X \to X^C$). Given such a map, we inductively define the functions $\widehat{\rho} : X \times C^* \to X$ and $\rho^\circ : X \times C^* \to \mathcal{P}X$ by putting $\widehat{\rho}(x, \epsilon) := x$, $\widehat{\rho}(x, cu) := \widehat{\rho}(\rho(x, c), u)$, resp. $\rho^\circ(x, \epsilon) := \varnothing$, $\rho^\circ(x, cu) := \{\rho(x, c)\} \cup \rho^\circ(\rho(x, c), u)$. In words, $\widehat{\rho}(p, u)$ denotes the state reached by a transition system after, starting at state p, it has processed the word u; and $\rho^\circ(p, u)$ is the set of states passed along the way, after leaving p. We often write $x \xrightarrow{c}_\rho y$ for $\rho(x, c) = y$ and $x \overset{u}{\Rightarrow}_\rho y$ for $\widehat{\rho}(x, u) = y$, omitting the subscript if ρ is understood.

Our concept of an automaton will not include an initial state; rather, we define an *initialized* or *pointed* automaton to be a pair $(\mathbb{A}, a)$ such that a is a state of the automaton $\mathbb{A}$. Given a transition map $\rho : X \times C \to X$, a state $x \in X$, and a stream α, we let $Inf(x, \alpha)$ denote the set of states in X traversed infinitely often when following α starting from x. A (deterministic) *parity* automaton is a triple $\mathbb{P} = (P, \rho, \Pi)$ such that P is a finite set of states, $\rho : P \times C \to P$ is a transition map, and $\Pi : P \to \omega$ is a priority function. An initialized parity automaton $(\mathbb{P}, p)$ accepts a stream α if $\max(\Pi[Inf(p, \alpha)])$ is even. The sets of words/nonempty words/streams recognized by an initialized automaton $(\mathbb{A}, a)$ (of the appropriate kind) are denoted as $L(\mathbb{A}, a)/L^+(\mathbb{A}, a)/Streams(\mathbb{A}, a)$. We shall generally use the symbol $\rhd$ for acceptance. A stream language is *regular* if it is recognized by some initialized parity automaton.

3 Ω-coalgebras as lasso automata

3.1 Basics

As mentioned in the introduction, regular stream languages are determined by their ultimately periodic fragments. This motivates our interest in finite representations of ultimately periodic streams: lassos.

▶ **Definition 1.** *A* lasso *is a pair $(u, v) \in C^* \times C^+$, representing the stream uv^ω. The words u and v are respectively the* spoke *and* loop *of the lasso. We call two lassos (u_0, v_0) and (u_1, v_1)* bisimilar*, notation: $(u_0, v_0) \leftrightarrows (u_1, v_1)$, if they represent the same stream, i.e., if $u_0v_0^\omega = u_1v_1^\omega$.*

Continuing the program of Calbrix, Nivat & Podelski [7] in a coalgebraic direction, in [9] we introduced Ω-coalgebras as automata operating on lassos.

▶ **Definition 2.** *An* Ω-coalgebra *is a structure* $\mathbb{A} = (P, X, \rho, \xi, \sigma, F)$ *such that* P *and* X *are sets of* spoke *and* loop states*, respectively; both* $\rho : P \to P^C$ *and* $\xi : X \to X^C$ *are transition functions;* $\sigma : P \to X^C$ *is the* switch *map; and* $F \subseteq X$ *is a set of* accepting states*. A* lasso automaton *is a finite* Ω*-coalgebra.*

The loop automaton *of* $\mathbb{A}$ *is the DFA* $\mathbb{A}_\ell := (P \uplus X, \sigma{:}\xi, F)$*, where* $\sigma{:}\xi : (P \uplus X) \to (P \uplus X)^C$ *is defined as* σ *on* P *and as* ξ *on* X*. For* $p \in P$*, we define Loop*(p) *as the set of finite words accepted by* $(\mathbb{A}_\ell, p)$*.*

Think of $\mathbb{A} = (P, X, \rho, \xi, \sigma, F)$ as a spoke part (P, ρ) and a loop part (X, ξ, F) that are connected by the switch function $\sigma : P \times C \to X$. Clearly this "loop part" is a DFA in its own right, just like "the" loop automaton $\mathbb{A}_\ell$; observe that (X, ξ, F) is a subautomaton of the loop automaton $\mathbb{A}_\ell$, and that by construction, any initialized automaton $(\mathbb{A}_\ell, p)$ with $p \in P$ will accept nonempty words only.

▶ Remark 3. In fact, Definition 2 is a simplification of the one given in [9], where the switching function has type $\sigma : P \to X$, and the loop part is a finite automaton designed to operate on nonempty words.

▶ Remark 4. Although the main point of the paper is to show how stream automata nicely fit a coalgebraic framework and our entire approach is coalgebraic in spirit, we decided to keep the categorical machinery at a minimum. The coalgebraic definition is phrased as follows. Our base category is Set^2, the category of two-sorted sets with two-sorted functions. For the definition of the endofunctor $\Omega : \mathsf{Set}^2 \to \mathsf{Set}^2$, it will be convenient to use the functors $\mathsf{E}_C := (-)^C$ and $\mathsf{D}_C := 2 \times (-)^C$ of, respectively, (C-)transition functions and (C-)DFAs. Now, given an object (X_0, X_1) in Set^2, we define

$$\Omega(X_0, X_1) := (\mathsf{E}_C X_0 \times \mathsf{E}_C X_1, \mathsf{D}_C X_1).$$

For the action of Ω on arrows, consider a pair $f = (f_0, f_1)$ of functions $f_i : X_i \to Y_i$, then Ωf is the pair of functions $(\mathsf{E}_C f_0 \times \mathsf{E}_C f_1, \mathsf{D}_C f_1)$. With this definition, the Ω-coalgebras of Definition 2 and 6 are coalgebras for the functor Ω indeed; to see this, observe that a coalgebra for the functor Ω consists of two sets X_0 and X_1, and three maps $h : X_0 \to X_0^C$, $h' : X_0 \to X_1^C$, and $h'' : X_1 \to 2 \times X_1^C$. An Ω-coalgebra $(P, X, \rho, \xi, \sigma, F)$ is rendered in such a form by letting $X_0 = P$, $X_1 = X$, $h = \rho$, $h' = \xi$, and deriving h'' from σ and F. As the construction is similar to that of [9], we leave the details as an exercise for the reader.

▶ **Definition 5.** *Where* $\mathbb{A} = (P, X, \rho, \xi, \sigma, F)$ *is an* Ω*-coalgebra, and* $p \in P$ *is a spoke state, we say that a lasso* (u, v) *is* accepted *by* $\mathbb{A}$ *at* p*, notation:* $\mathbb{A}, p \rhd (u, v)$*, if* $\widehat{\sigma{:}\xi}(\widehat{\rho}(p, u), v)) \in F$*, and we write Lassos*$(\mathbb{A}, p)$ *to denote the lasso language* recognized *by* $(\mathbb{A}, p)$*, that is, the set of all lassos accepted by Lassos*$(\mathbb{A}, p)$*. Finally, a lasso language is called* regular *if it is the language recognized by a pointed finite* Ω*-coalgebra, i.e., a lasso automaton.*

Intuitively, an Ω-coalgebra operates on a lasso (u, v) by first processing the spoke u, then switching to the loop automaton and processing the loop v. It accepts the lasso iff the resulting state is accepting. That is, $(\mathbb{A}, p)$ accepts (u, v) iff $v \in Loop(\widehat{\rho}(p, u))$.

▶ **Definition 6.** *Let* $\mathbb{A} = (P, X, \rho, \xi, \sigma, F)$ *and* $\mathbb{A}' = (P', X', \rho', \xi', \sigma', F')$ *be two* Ω*-coalgebras. An* Ω*-morphism from* $\mathbb{A}$ *to* $\mathbb{A}'$ *is a pair* $h = (h_0, h_1)$ *of maps* $h_0 : P \to P'$*,* $h_1 : X \to X'$ *such that, for all* $p \in P, x \in X$ *and* $c \in C$ *we have*

(M1) $h_0(\rho(p, c)) = \rho'(h_0 p, c)$*,*
(M2) $h_1(\xi(x, c)) = \xi'(h_1 x, c)$*,*
(M3) $h_1(\sigma(p, c)) = \sigma'(h_0 p, c)$*, and*

(M4) $x \in F$ *iff* $h_1 x \in F'$.
We usually write h *for* h_i.

A spoke state p *of* $\mathbb{A}$ generates *a subcoalgebra* $\mathbb{A}_p$ *of* $\mathbb{A}$ *that is based on the sets of spoke and loop states that are* reachable *from* p *(in the obvious sense, using* ρ, σ, *and* ξ*).*

As a manifestation of the coalgebraic nature of our structures, we discuss below how the natural concept of equivalence induced by Ω-morphisms can be captured by a notion of bisimilarity.

▶ **Definition 7.** *A* bisimulation *between two* Ω*-coalgebras* $\mathbb{A}$ *and* $\mathbb{A}'$ *is a pair* $Z = (Z_0, Z_1)$ *of relations* $Z_0 \subseteq P \times P'$, $Z_1 \subseteq X \times X'$ *such that, for all* $(p, p') \in Z_0$ *and* $(x, x') \in Z_1$, *and all* $c \in C$ *we have*
(B1) $(\rho(p, c), \rho'(p', c)) \in Z_0$,
(B2) $(\xi(x, c), \xi'(x', c)) \in Z_1$,
(B3) $(\sigma(p, c), \sigma'(p', c)) \in Z_1$, *and*
(B4) $x \in F$ *iff* $x' \in F'$.

Two pointed coalgebras $(\mathbb{A}, p)$ *and* $(\mathbb{A}', p')$ *are* bisimilar, *notation:* $\mathbb{A}, p \leftrightarrows \mathbb{A}', p'$, *if there is a bisimulation linking* p *and* p'.

The following characterization of bisimilarity using morphisms holds for a wide range of coalgebras; in coalgebraic terms, it says that for the functor Ω, the notions of bisimilarity and behavioral equivalence coincide. The proposition follows from categorical properties of the functor Ω, but also has a straightforward direct proof.

▶ **Proposition 8.** *Let* $(\mathbb{A}, p)$ *and* $(\mathbb{A}', p')$ *be pointed* Ω*-coalgebras. Then* $(\mathbb{A}, p) \leftrightarrows (\mathbb{A}', p')$ *iff there is a* Ω*-coalgebra* $\mathbb{B}$ *and* Ω*-morphisms* $h : \mathbb{A} \to \mathbb{B}$ *and* $h' : \mathbb{A}' \to \mathbb{B}$ *such that* $hp = h'p'$.

The following proposition on bisimilarity will be needed later on; we omit the proof which follows a routine coalgebra argument.

▶ **Proposition 9.** *Let* $\mathbb{A}$ *and* $\mathbb{A}'$ *be two* Ω*-coalgebras. Then*
(1) *the collection of bisimulations between* $\mathbb{A}$ *and* $\mathbb{A}'$ *forms a complete lattice, of which the join is given by union;*
(2) *the relation* $\leftrightarrows$ *itself is the largest bisimulation between* $\mathbb{A}$ *and* $\mathbb{A}'$;
(3) *if* $\mathbb{A} = \mathbb{A}'$, *the relation* $\leftrightarrows$ *is an equivalence relation.*

The key observation is that bisimilarity *exactly* captures lasso equivalence. This was first shown in [9]; by looking at the explicit definition of bisimilarity given in Definition 7, the proof is a simple extension to the two-sorted setting of the classical result that in classical DFAs operating on *finite* words, bisimilarity coincides with language equivalence (see [16]).

▶ **Fact 10.** [9] *Any pair of pointed* Ω*-coalgebras* $(\mathbb{A}, p)$ *and* $(\mathbb{A}', p')$ *satisfy*

$$\mathbb{A}, p \leftrightarrows \mathbb{A}', p' \text{ iff } \mathit{Lassos}(\mathbb{A}, p) = \mathit{Lassos}(\mathbb{A}', p'). \tag{1}$$

▶ **Example 11.** Fixing the alphabet $C = \{a, b\}$, we define the Ω*-coalgebra* $\mathbb{A} = (P, X, \rho, \xi, \sigma, F)$ where $P = \{1, 2, 3\}$, $X = \{4, 5, 6\}$, $\rho = \{(1, a) \mapsto 1, (1, b) \mapsto 2, (2, a) \mapsto 1, (2, b) \mapsto 3, (3, a) \mapsto 2, (3, b) \mapsto 3\}$, $\xi = \{(4, a) \mapsto 4, (4, b) \mapsto 6, (5, b) \mapsto 5, (5, a) \mapsto 6, (6, a) \mapsto 6, (6, b) \mapsto 6\}$, $\sigma = \{(p, a) \mapsto 4, (p, b) \mapsto 5\}$ for each $p \in P$, and $F = \{4, 5\}$. By construction, for all $p \in P$, we have $\mathit{Lassos}(\mathbb{A}, p) = \{(u, v) \mid u \in C^\star \wedge v \in (\{a^+\} \cup \{b^+\})\}$. Consider the Ω-coalgebra $\mathbb{A}' = (\{1\}, X, \rho', \sigma', \xi, F)$, where ρ' and σ' are the restriction of ρ and σ, respectively, to the singleton $\{1\}$. By Fact 10, $\mathbb{A}, 1 \leftrightarrows \mathbb{A}, 2 \leftrightarrows \mathbb{A}, 3$. The situation is witnessed by the Ω-morphism $h : \mathbb{A} \to \mathbb{A}'$, which identifies all the states in P by mapping them to the state 1. Furthermore, for all $p \in P$, we have $\mathbb{A}, p \leftrightarrows \mathbb{A}', 1$.

3.2 From parity automata to lasso automata

Given a Büchi automaton $\mathbb{B}$, Calbrix, Nivat & Podelski [7] constructed a DFA recognizing the finite-word language $\{u\$v \mid \mathbb{B} \rhd uv^\omega\}$, where \$ is a new symbol (i.e., not in C). Here we give a similar construction for parity automata.

▶ **Definition 12.** *Let $\mathbb{P} = (P, \rho, \Pi)$ be a parity automaton and let p be a state of $\mathbb{P}$. The DFA $\mathbb{X}_p := (X, \xi, F_p)$ is based on the state space $X := (P \times N)^P$, where $N := \mathsf{Ran}(\Pi)$ is the range of Π. To define its transition map ξ, consider an arbitrary state $t \in X$ and think of t as the pairing of $t^0 : P \to P$ and $t^1 : P \to N$. Now define*

$$\xi(t)(c) := \lambda q.\Big(\rho(t^0 q, c) \ , \ \max(t^1 q, \Pi(\rho(t^0 q, c))) \Big)$$

For the set F_p of accepting states, define, for an arbitrary state $t \in X$, the sequence $(p_i^t)_{i\in\omega}$ by putting $p_0^t := p$, $p_{i+1}^t := t^0 p_i^t$, and put

$$F_p := \{t \in X \mid \max(t^1[Inf((p_i^t)_{i\in\omega})]) \text{ is even } \},$$

where $Inf((p_i^t)_{i\in\omega})$ is the set of $p \in P$ occurring as p_i^t for infinitely many i. Finally, we define $s_I^p \in X$ as the initial state $s_I^p := \lambda q.(q, 0)$.

Intuitively, the initialized DFA $(\mathbb{X}_p, s_I^p)$ consumes a word v by following it in parallel, starting from each state of $\mathbb{P}$. Moreover, in each of these parallel runs the automaton collects the maximum priority of the traversed states. This explains the carrier and the transition map of the automaton $\mathbb{X}_p$. For its set of accepting states, first note that F_p is the only part of $\mathbb{X}_p$ depending on p. The idea behind its definition is that for a word $v \in C^+$, the state $t := \widehat{\xi}(s_I^p, v)$ encodes essential information on the run $(\mathbb{P}, p)$ on the stream v^ω. In particular, we have $p_i^t = \widehat{\rho}(p, v^i)$, for all i. Analyzing the way in which the map $t^1 : P \to N$ keeps track of maximal priorities along finite runs, we may then show that $\max(t^1[Inf((p_i^t)_{i\in\omega})])$ corresponds to the maximal priority that one encounters in the run of $(\mathbb{P}, p)$ on v^ω.

The key observation on this automaton is that $(\mathbb{X}_p, s_I^p)$ recognizes the looping language of $(\mathbb{P}, p)$, that is, for all $v \in C^+$:

$$(\mathbb{X}_p, s_I^p) \text{ accepts } v \text{ iff } (\mathbb{P}, p) \text{ accepts } v^\omega.$$

▶ **Definition 13.** *Let $\mathbb{P} = (P, \rho, \Pi)$ be a parity automaton. Recalling that by definition, P is finite, we define the lasso automaton $\mathbb{A}_\mathbb{P} := (P, X, \rho, \xi, \sigma, F)$ by letting (X, ξ, F) be the coproduct (disjoint union) of the family $\{\mathbb{X}_p \mid p \in P\}$ of DFAs, and putting $\sigma(p, c) := \xi(s_I^p, c)$.*

▶ **Fact 14.** [9] *Let $(\mathbb{P}, p)$ be an initialized parity automaton. For any lasso $(u, v) \in C^* \times C^+$:*

$$(\mathbb{A}_\mathbb{P}, p) \text{ accepts } (u, v) \text{ iff } (\mathbb{P}, p) \text{ accepts } uv^\omega. \tag{2}$$

▶ **Example 15.** Using the alphabet $C = \{a, b\}$, consider the parity automaton $\mathbb{P} = (P, \rho, \Pi)$ where P and ρ come from Example 11, and $\Pi = \{1 \mapsto 0, 3 \mapsto 0, 2 \mapsto 1\}$. It is easily seen that, no matter what the initial state is, the language accepted by the automaton is $\{\alpha \subseteq C^\omega \mid \exists i \in \omega . \exists c \in C . \forall j \geq i . \alpha_i = c\}$, that is, those streams that have a tail consisting of an infinite repetition of one symbol. The reader should note that there is no single-state parity automaton accepting the same language, as a single-state automaton may either accept C^ω or the empty language. However, either by direct construction, or by Fact 14, for all $p \in P$, it can be shown that $\mathbb{A}_\mathbb{P}, p \leftrightarrows \mathbb{A}', 1$, where $\mathbb{A}'$ comes from Example 11, in turn.

3.3 Coherence & Circularity

The language recognized by a lasso automaton $(\mathbb{A}, p)$ does not necessarily correspond to the ultimately periodic fragment of a regular ω-language. A necessary condition for the latter is that $\mathit{Lassos}(\mathbb{A}, p)$ is *invariant under lasso bisimilarity*: $(u, v) \leftrightarrows (u', v')$ implies that $(u, v) \in \mathit{Lassos}(\mathbb{A}, p)$ iff $(u', v') \in \mathit{Lassos}(\mathbb{A}, p)$. In [9] we proved that this condition is also sufficient, and we characterized it by the properties of coherence and circularity.

▶ **Definition 16.** *A regular language L is* circular *if $v \in L \Leftrightarrow v^k \in L$, for all $k > 0$ and $v \in C^+$. An initialized DFA $(\mathbb{A}, a)$ is* circular *if $L(\mathbb{A}, a)$ is circular. A lasso automaton $\mathbb{A}$ is* circular *if each $\mathit{Loop}(p)$ is circular, and* coherent *if $cu \in \mathit{Loop}(p) \Leftrightarrow uc \in \mathit{Loop}(\rho(p, c))$, for every spoke state p, $u \in C^*$ and $c \in C$.*

▶ **Fact 17.** [9] *For any lasso automaton $\mathbb{A} = (P, X, \rho, \xi, \sigma, F)$ the following are equivalent:*
(1) *$\forall p \in P$. $\mathit{Lassos}(\mathbb{A}, p) = \{(u, v) \mid uv^\omega \in L\}$ for some regular ω-language L;*
(2) *$\forall p \in P$. $\mathit{Lassos}(\mathbb{A}, p)$ is bisimulation invariant;*
(3) *$\mathbb{A}$ is circular and coherent.*

Motivated by Fact 17, in the sequel we will largely confine our attention to circular and coherent lasso automata. This explains the importance of the following result.

▶ **Theorem 18.** *It is decidable whether a given lasso automaton is circular and coherent.*

Finally, we note that by Fact 10, the class of circular and coherent lasso automata is closed under taking surjective Ω-morphisms.

4 Ω-automata

In this section, we look at Ω-automata as acceptors of *streams*. As we shall see, this notion of acceptance coincides with the one of parity automata (Theorem 22) and is invariant under Ω-morphisms (Theorem 23). The main goal of this section is to show that, unlike the standard types of stream automata, Ω-automata admit a natural notion of bisimilarity that *exactly* captures language equivalence (Theorem 25). Finally, as a corollary of these results we obtain a simple and natural minimization procedure for Ω-automata (Theorem 28).

4.1 Ω-coalgebras as stream automata

In the previous section we saw that a lasso language corresponds to the ultimately periodic fragment of a regular ω-language if and only if it is the language recognized by an initialized, circular and coherent lasso automaton. This suggests that circular and coherent Ω-coalgebras might be used directly as stream automata, and inspires the following definition.

▶ **Definition 19.** *An* Ω-automaton *is a circular and coherent lasso automaton.*

Following ideas in Calbrix, Nivat & Podelski [7], we now define acceptance of streams.

▶ **Definition 20.** *Let $\mathbb{A} = (P, X, \rho, \xi, \sigma, F)$ be an Ω-automaton, and let q be a spoke state of $\mathbb{A}$. We say that a stream α is* accepted *by $(\mathbb{A}, q)$, notation: $(\mathbb{A}, q) \rhd \alpha$, if there are a finite word u, a sequence $(v_i)_{i \in \omega}$ of finite, non-empty words, a state $p \in P$ and an accepting state $z \in F$ such that $\alpha = u\vec{v}$, $q \overset{u}{\Rightarrow}_\rho p$ and $p \overset{v_i}{\Rightarrow}_\rho p$, $p \overset{v_i}{\Rightarrow}_{\sigma;\xi} z$ for each $i \in \omega$. The set of streams accepted by an initialized Ω-automaton $(\mathbb{A}, p)$ is denoted as $\mathit{Streams}(\mathbb{A}, p)$.*

▶ Remark 21. Where this notion of acceptance may seem somewhat odd at first sight, it can be related to that of well-known stream automata. A *successful run* of $(\mathbb{A}, q)$ on a stream α consists of

- a run of the spoke part of $\mathbb{A}$ on a (finite) initial segment u of α (i.e., $\alpha = u\beta$ for some stream β), leading to a spoke state p, followed by
- an infinite run of the *product structure* of (P, ρ) and $\mathbb{A}_\ell$ on the remaining stream β, where for some accepting state $z \in F$, the product automaton infinitely often makes a *silent step* from (p, z) to (p, p):

$$(p,p) \overset{v_0}{\Longrightarrow}_{\rho\times(\sigma:\xi)} (p,z) \overset{\epsilon}{\rightsquigarrow} (p,p) \overset{v_1}{\Longrightarrow}_{\rho\times(\sigma:\xi)} (p,z) \overset{\epsilon}{\rightsquigarrow} (p,p) \overset{v_1}{\Longrightarrow}_{\rho\times(\sigma:\xi)} \cdots$$

Based on this observation, it is not difficult to show (but rather tedious to spell out in detail) that Ω-automata can be seen as rather special Büchi automata, where the nondeterminism is restricted to (1) a unique jump from an initial part of the automaton to a final part, and (2) some very specific silent steps. From this perspective, that is, with Ω-automata taken as a subclass of Büchi automata with silent steps, it is remarkable that the theory of Ω-automata is so well-behaved when we consider bisimilarity etc. This good behavior may be explained by the observation that seen as lasso automata, Ω-coalgebra are completely deterministic.

4.2 Adequacy

The following theorem states that, when it comes to recognizing stream languages, Ω-automata are as least as expressive as more standard models like parity automata.

▶ **Theorem 22** (Adequacy). *Let $(\mathbb{P}, q)$ be an initialized deterministic parity automaton. Then $Streams(\mathbb{P}, q) = Streams(\mathbb{A}_\mathbb{P}, q)$.*

4.3 Language equivalence as bisimilarity

▶ **Theorem 23** (Invariance). *Let $h : \mathbb{A} \to \mathbb{A}'$ be an Ω-morphism between two Ω-automata.*

$$Streams(\mathbb{A}, q) = Streams(\mathbb{A}', hq) \tag{3}$$

for any spoke state q of $\mathbb{A}$.

▶ **Corollary 24.** *Let $(\mathbb{A}, q)$ be an Ω-automaton. Then $Streams(\mathbb{A}, q)$ is an ω-regular language and $Lassos(\mathbb{A}, q) = \{(u, v) \mid uv^\omega \in Streams(\mathbb{A}, q)\}$.*

The next result shows that, unlike well-known types of stream automata such as Büchi, Muller or parity automata, Ω-automata share a fundamental property with deterministic automata operating on finite words.

▶ **Theorem 25** (Language equivalence as bisimilarity). *Let $(\mathbb{A}, q)$ and $(\mathbb{A}', q')$ be two initialized Ω-automata. Then*

$$\mathbb{A}, q \leftrightarrows \mathbb{A}', q' \text{ iff } Streams(\mathbb{A}, q) = Streams(\mathbb{A}', q'). \tag{4}$$

▶ **Example 26.** Continuing from Example 15, observe that the Ω-coalgebras $\mathbb{A}$ and $\mathbb{A}'$ are actually circular and coherent, and therefore Ω-automata. The stream language that these coalgebras accept, from any initial state, is the same as the one accepted by the parity automaton $\mathbb{P}$, from any initial state. However, in the realm of Ω-automata, by virtue of the associated notion of bisimilarity, there is a canonical representative for the class of all pointed Ω-automata accepting the stream language of $\mathbb{P}$ (from any state, as they all accept

the same language). It should not be difficult to guess that the canonical representative is $(\mathbb{A}', 1)$, up-to isomorphism. A formal proof needs just to show that the three states in X accept different languages (of finite words). In Section 4.4 we shall discuss computation of such a representative by partition refinement.

4.4 Minimal Ω-automaton

The minimization problem for stream automata is much harder than that for deterministic finite automata operating on finite words. In particular, regular ω-languages generally do not have a *unique* minimal automaton [18], and to the best of our knowledge, nice minimization procedures are only available for restricted classes of automata [18, 13].

This is different in the setting of Ω-automata. As a corollary of Theorem 25 we obtain a simple and natural minimization procedure for Ω-automata, linking the final coalgebra to the minimal automaton: Theorem 28. A *partition refinement* algorithm is a standard consequence of the coalgebraic framework (see [9] for a more detailed explanation).

▶ **Definition 27.** *Let $\mathbb{A} = (P, X, \rho, \xi, \sigma, F)$ be an Ω-automaton, and recall from Proposition 8 that $\leftrightarrows$ is an equivalence relation on P and on X. We denote the equivalence class of a state a with $\overline{a}$. Since $\leftrightarrows$ is itself a bisimulation relation, the following is a correct definition of an Ω-coalgebra structure on the $\leftrightarrows$-cells:*

$$\begin{array}{lcll} \overline{\rho}(\overline{p}) & := & \overline{\rho p}, & \\ \overline{\xi}(\overline{x}) & := & \overline{\xi x}, & \\ \overline{\sigma}(\overline{p})(c) & := & \overline{\sigma(p)(c)}, & \text{for all } c, \\ \overline{F} & := & \{\overline{x} \mid x \in F\}. & \end{array}$$

We denote the resulting Ω-automaton by $\mathbb{A}_{/\leftrightarrows}$.

The following theorem shows that $\mathbb{A}_{/\leftrightarrows}$ is a *minimal* automaton recognizing the languages of $\mathbb{A}$.

▶ **Theorem 28.** *Let $(\mathbb{A}, p)$ be an initialized Ω-automaton, with $L := \mathit{Streams}(\mathbb{A}, p)$. Then*

(1) *$\mathit{Streams}(\mathbb{A}_{/\leftrightarrows}, \overline{p}) = L$;*

(2) *For any initialized Ω-automaton $(\mathbb{A}', p')$ such that $\mathit{Streams}(\mathbb{A}', p') = L$, there is an Ω-morphism $h : \mathbb{A}'_{p'} \to \mathbb{A}_{/\leftrightarrows}$ such that $h(p') = \overline{p}$.*

In passing we note that the Theorems 22 and 28 yield a minimization procedure for parity automata (and other standard stream automata) as well; this procedure transforms any parity automaton, not into a minimal parity automaton, but into a canonically obtained minimal Ω-automaton.

▶ **Example 29.** Continuing from Example 26, the Ω-coalgebra $\mathbb{A}'$ is, up to isomorphism, the minimal representative of the coalgebra $\mathbb{A}$. Note that the three states $\{1, 2, 3\}$ are quotiented by the unique morphism h (from Example 11).

5 A final Ω-coalgebra and a Myhill-Nerode Theorem

5.1 Final Ω-coalgebra

In the theory of Universal Coalgebra, an important role is played by final coalgebras. Recall that an object z is final in a category C if for every object there is a unique arrow to z. Final coalgebras do not exist for every functor, but when they exist, they usually encode a

natural notion of *behavior* related to the functor. For instance, in the theory of DFAs, a final coalgebra is provided by the 'language automaton', in which the states are languages (of finite words), the transition structure is given by the derivatives ($L_c := \{u \in C^* \mid cu \in L\}$), and a language/state is accepting iff it contains the empty word. This language coalgebra has many nice properties and can be used to prove many fundamental properties of regular languages [16]. In this section we will see that the category of Ω-coalgebras admits a final coalgebra, and we will use this coalgebra to give a Myhill-Nerode theorem for regular lasso languages, and a related characterization for regular ω-languages.

▶ **Definition 30.** *With defining*

$$\begin{array}{lcl} Z_0 & := & \mathcal{P}(C^* \times C^+), \\ Z_1 & := & \mathcal{P}C^*, \\ \zeta_0(L,c) & := & \{(u,v) \in C^* \times C^+ \mid (cu,v) \in L\}, \\ \zeta_1(M,c) & := & \{v \in C^* \mid cv \in M\}, \\ \sigma(L,c) & := & \{v \in C^* \mid (\epsilon, cv) \in L\}, \\ F & := & \{M \in \mathcal{P}C^* \mid \epsilon \in F\}, \end{array}$$

we obtain the Ω-coalgebra $\mathbb{Z} := (Z_0, Z_1, \zeta_0, \zeta_1, \sigma, F)$.

Observe that the loop part of this Ω-coalgebra is given by the final coalgebra for DFAs that we just mentioned; in particular, its carrier is based on the set of all languages of finite words. The carrier of the spoke part is given by the set $\mathcal{P}(C^* \times C^+)$ of all lasso languages. The exact relation of this Ω-coalgebra with the set of all stream languages remains to be investigated in more detail, but note that the relation of *lasso determinacy* (Definition 35) will play an important role here.

A very useful property of the structure $\mathbb{Z}$ is that any lasso language L coincides with the set of lassos that it accepts, seen as a state of $\mathbb{Z}$.

▶ **Theorem 31.** *Let L be a lasso language. Then for any lasso (u,v) we have*

$$\mathbb{Z}, L \rhd (u,v) \text{ iff } (u,v) \in L. \tag{5}$$

As a corollary, the relation $\leftrightarrows$ restricts to the identity relation on $\mathbb{Z}$.

▶ **Theorem 32.** *$\mathbb{Z}$ is final in the category of Ω-coalgebras and Ω-morphisms. That is, for every Ω-coalgebra $\mathbb{A}$ there is a unique Ω-morphism $h : \mathbb{A} \to \mathbb{Z}$.*

5.2 A Myhill-Nerode Theorem for lasso languages

Rutten provided a nice coalgebraic perspective on the Myhill-Nerode theorem for regular languages of finite words, identifying the congruence classes of the Myhill-Nerode equivalence relation with states in the final coalgebra [16]. A similar result holds for lasso languages.

▶ **Definition 33.** *Let L be a lasso language. Define the equivalence relation $\equiv_0$ on C^* such that $u_0 \equiv_0 u_1$ iff for all lassos (u,v) it holds that $(u_0u, v) \in L \iff (u_1u, v) \in L$. Define a family of binary relations $\equiv_{[u]}$ on C^+, indexed by the set of $\equiv_0$-cells, such that $v_0 \equiv_{[u]} v_1$ iff for all $w \in C^*, u_0, u_1 \in [u]$ we have $(u_0, v_0w) \in L \iff (u_1, v_1w) \in L$. Finally, let*

$$(u_0, v_0) \equiv_L (u_1, v_1) \text{ iff } u_0 \equiv_0 u_1 \text{ and } v_0 \equiv_{[u_i]} v_1$$

define a relation $\equiv_L$ on lassos.

It is obvious that $\equiv_L$ is an equivalence relation, and here we have arrived at our coalgebraic Myhill-Nerode theorem for lassos. It refers to the generated subcoalgebra of a state/language L in the final coalgebra, see Definition 6. Recall that a lasso language is regular if it is the set of all lassos that are accepted by a pointed lasso automaton.

▶ **Theorem 34.** *The following are equivalent, for any lasso language L:*
(1) *L is regular;*
(2) *L generates a finite subcoalgebra in $\mathbb{Z}$;*
(3) *the relation $\equiv_L$ has finite index.*

5.3 A characterization theorem for stream languages

Attempts at finding congruences that characterise ω-regularity date back to the earliest works in the theory of ω-languages, such as Arnold [4], who isolates the *syntactic congruence* of a regular ω-regular language $L \subseteq C^\omega$ as the coarsest congruence on C^* that recognizes L (in some precisely defined manner). An interesting open question was to identify a congruence which is of finite index if and only if a given stream language is regular. Maler and Staiger [14] approached this problem via so-called *families of right congruences* (FORCs), and proved that a stream language L is ω-regular if and only if there exists a finite FORC that recognises it. Furthermore, the paper identified a necessary and sufficient characterisation of those regular ω-languages that are accepted by a minimal state automaton, derived from Arnold's syntactic congruence.

While moving in a similar direction, we are able to simplify the matter somewhat. The definition of our lasso relation $\equiv_L$ is reminiscent to that of a FORC, as it is dependent on equivalence classes of a right congruence on finite words.[2] Using the relation $\equiv_L$, we were able to provide an *exact* characterisation of regular ω-languages. Our characterization, Theorem 37, involves a property, *lasso determinacy*, that is somewhat related to (but not the same as) Arnold's saturation property. One may also look at lasso determinacy as an infinitary version of the *pumping* property of regular languages of finite words.

▶ **Definition 35.** *A stream language L is* lasso determined*, or has the property* LD*, if for every infinite sequence $(v_i)_{i\in\omega}$ of nonemtpy words there is an infinite set $Y \subseteq \omega$ such that*

$$\vec{v} \in L \iff (v_0 \cdots v_j)(v_{j+1} \cdots v_k)^\omega \in L.$$

for all $j, k \in Y$ with $j < k$.

It is not difficult to verify that all regular ω-languages are lasso determined. The following property justifies the terminology.

▶ **Proposition 36.** *Let L, L' be two stream languages with the property LD. If $Lassos(L) = Lassos(L')$ then $L = L'$.*

▶ **Theorem 37.** *The following are equivalent, for any stream language L:*
(1) *L is regular;*
(2) *L is lasso determined and $Lassos(L)$ generates a finite subcoalgebra in $\mathbb{Z}$;*
(3) *L is lasso determined and the relation $\equiv_L$ has finite index.*

---

[2] However, $\equiv_L$ does not exactly correspond to a FORC as it lacks the condition $v_0 \equiv_{[u_i]} v_1 \implies u_0 v_0 \equiv_0 u_1 v_1$.

6 Conclusions

The main point of this paper was to argue that (a slight variation of) the two-sorted lasso automaton we introduced in [9] provides an interesting framework for recognizing regular stream languages as well. For this purpose, we presented Ω-automata as the class of finite, circular and coherent lasso automata, and we defined a notion of acceptance for streams. Throughout the paper we used the fact that these structures are coalgebras for an endofunctor Ω on the category Set^2 of two-sorted sets and functions. The advantage of this coalgebraic presentation of stream automata is that bisimilarity and minimization are easily obtained using the general theory of Universal Coalgebra. Using another standard feature of the coalgebraic framework, namely, final coalgebras, we proved a Myhill-Nerode theorem for lasso automata that extends the basic framework of Rutten [16] to lassos. Then, involving a property we called *lasso determinacy*, we obtained a characterization of regular ω-languages. These results provide additional motivation for and in some sense complete the work of Maler & Staiger [14] on two-sorted congruences.

Of the many interesting directions to take from here we mention a few. First, not only Ω-automata but in particular the objects they operate on (lassos, streams) admit a very simple and natural coalgebraic presentation. It would be interesting to explore and exploit this connection further – in particular, we are interested in truly coalgebraic characterizations of regular ω-languages. In a subsequent publication we hope to report on such a characterization, which reveals the coalgebraic nature of the property of lasso determinacy and involves a pumping property for lasso languages. Second, a very interesting and useful coalgebraic concept is that of *coinduction*. This definition and proof principle has many applications in the theory of DFAs. It would be worthwhile to find and study manifestations of coinduction in the world of stream automata as well, also patterned after [20, 19, 6]. Third, given the two-sorted nature of our framework, it seems natural to explore its connections with the theory of *Wilke algebras* [21]. Finally, recent work on *coalgebraic automata learning* [5] could shed further light on the link between our framework and the research line on learning of omega-regular languages; a first step in this direction would be a mathematically precise comparison between the automata model presented in [2] and our coalgebraic variant.

Using Category Theory for modelling abstract notions entails the possibility to generalise results by changing the base category that is used. In this respect, our paper can be used as a starting point for further development of *nominal* omega-regular languages [8]. Finally, it would be interesting to study our work from the categorical perspective of [20, 19, 6]. As a starting point one could try to rephrase our acceptance definition for streams (Definition 20) in the approach of [19].

References

1 Dana Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75(2):87–106, Elsevier, 1987. doi:10.1016/0890-5401(87)90052-6.

2 Dana Angluin, Udi Boker, and Dana Fisman. Families of dfas as acceptors of ω-regular languages. *Logical Methods in Computer Science*, Volume 14, Issue 1, February 2018. doi:10.23638/LMCS-14(1:15)2018.

3 Dana Angluin and Dana Fisman. Learning regular omega languages. In *Algorithmic Learning Theory*, volume 8776 of *Lecture Notes in Computer Science*, pages 125–139. Springer International Publishing, 2014. doi:10.1007/978-3-319-11662-4_10.

4 André Arnold. A syntactic congruence for rational ω-languages. *Theoretical Computer Science*, 39:333–335, Elsevier, 1985. doi:10.1016/0304-3975(85)90148-3.

5 Simone Barlocco, Clemens Kupke, and Jurriaan Rot. Coalgebra learning via duality. In *Foundations of Software Science and Computation Structures*, volume 11425 of *Lecture Notes in Computer Science*, pages 62–79. Springer International Publishing, 2019. doi:10.1007/978-3-030-17127-8_4.

6 Tomasz Brengos. A coalgebraic take on regular and omega-regular behaviour for systems with internal moves. In *International Conference on Concurrency Theory*, volume 118 of *LIPIcs*, pages 25:1–25:18. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018. doi:10.4230/LIPIcs.CONCUR.2018.25.

7 Hugues Calbrix, Maurice Nivat, and Andreas Podelski. Ultimately periodic words of rational ω-languages. In *Mathematical Foundations of Programming Semantics*, volume 802 of *Lecture Notes in Computer Science*, pages 554–566. Springer, 1993. doi:10.1007/3-540-58027-1_27.

8 Vincenzo Ciancia and Matteo Sammartino. A class of automata for the verification of infinite, resource-allocating behaviours. In *Trustworthy Global Computing*, volume 8902 of *Lecture Notes in Computer Science*, pages 97–111. Springer Berlin Heidelberg, 2014. doi:10.1007/978-3-662-45917-1_7.

9 Vincenzo Ciancia and Yde Venema. Stream automata are coalgebras. In *Coalgebraic Methods in Computer Science*, volume 7399 of *Lecture Notes in Computer Science*, pages 90–108. Springer, 2012. doi:10.1007/978-3-642-32784-1_6.

10 Szilárd Fazekas. Powers of regular languages. In Volker Diekert and Dirk Nowotka, editors, *Developments in Language Theory*, volume 5583 of *Lecture Notes in Computer Science*, pages 221–227. Springer, 2009. doi:10.1007/978-3-642-02737-6_17.

11 E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logic, and Infinite Games*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002.

12 Yong Li, Yu-Fang Chen, Lijun Zhang, and Depeng Liu. A novel learning algorithm for büchi automata based on family of dfas and classification trees. In *Tools and Algorithms for the Construction and Analysis of Systems, Part I*, volume 10205 of *Lecture Notes in Computer Science*, pages 208–226, 2017. doi:10.1007/978-3-662-54577-5_12.

13 Christof Löding. Efficient minimization of deterministic weak ω-automata. *Information Processing Letters*, 79:105–109, Elsevier, 2001. doi:10.1016/S0020-0190(00)00183-6.

14 Oded Maler and Ludwig Staiger. On syntactic congruences for ω-languages. *Theoretical Computer Science*, 183:93–112, Elsevier, 1997. doi:10.1007/3-540-56503-5_58.

15 Dominique Perrin and Jean-Éric Pin. *Infinite Words: Automata, Semigroups, Logic and Games*. Elsevier, 2004.

16 J. Rutten. Automata and coinduction (an exercise in coalgebra). In *International Conference on Concurrency Theory*, Lecture Notes in Computer Science, pages 194–218. Springer, 1998. doi:10.1007/BFb0055624.

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

18 Ludwig Staiger. Finite-state ω-languages. *Journal of Computer and System Sciences*, 27:434–448, Elsevier, 1983. doi:10.1016/0022-0000(83)90051-X.

19 Natsuki Urabe and Ichiro Hasuo. Categorical büchi and parity conditions via alternating fixed points of functors. In *Coalgebraic Methods in Computer Science, Revised Selected Papers*, volume 11202 of *Lecture Notes in Computer Science*, pages 214–234. Springer, 2018. doi:10.1007/978-3-030-00389-0_12.

20 Natsuki Urabe, Shunsuke Shimizu, and Ichiro Hasuo. Coalgebraic trace semantics for buechi and parity automata. In *International Conference on Concurrency Theory*, 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.

21 Thomas Wilke. An algebraic theory for regular languages of finite and infinite words. *International Journal of Algebra and Computation*, 03(04):447–489, 1993. doi:10.1142/S0218196793000287.

A Proofs

Proof of Theorem 18. Let $\mathbb{A}$ be lasso automaton. Decidability of coherence is straightforward, since it can be checked in terms of the equivalence of various pairs of initialized DFAs that can easily be constructed from the loop automaton of $\mathbb{A}$.

For circularity, it suffices to consider initialized DFAs. A routine argument shows that a language is circular iff $Pow(L) = L = Root(L)$, where the *power* and *root* of L are defined by $Pow(L) := \{u^k \mid u \in L, k \geq 1\}$ and $Root(L) := \{u \mid u^k \in L \text{ for some } k \geq 1\}$. The decidability of the equation, $L = Pow(L)$, was established by Fazekas [10].

The problem, whether $L(\mathbb{A}, a) = Root(L(\mathbb{A}, a_I))$ for a given initialized DFA $(\mathbb{A}, a_I)$, can be solved by first defining an initialized DFA $(\mathbb{A}', i)$ that accepts $Root(L)$ and then checking language equivalence with L. So consider a DFA (A, δ, F) and a state $a_I \in A$. Let $\mathbb{A}' := (A^A, \theta, G)$ be the DFA where θ is given pointwise: $\theta(f, c) := \lambda a.\delta(fa, c)$. For the definition of G, define, for an arbitrary $g \in A^A$, the set $A_g := \{g^n a_I \mid n \geq 1\}$ – since A is finite, this set can be computed in at most $|A|$ steps. Put $G := \{g \in A^A \mid A_g \cap F \neq \varnothing\}$.

Let u an arbitrary finite word, and define $g_u := \widehat{\theta}(i, u)$. By construction we obtain for all $a \in A$ that $g_u a = \widehat{\delta}(a, u)$, so that $A_{g_u} = \{\widehat{\delta}(a_I, u^n) \mid n \geq 1\}$. Now consider the following chain of equivalences:

$$
\begin{aligned}
\mathbb{A}', i \rhd u \text{ iff } & g_u = \widehat{\theta}(i, u) \in G && \text{(definition of acceptance)}\\
\text{iff } & A_{g_u} \cap F \neq \varnothing && \text{(definition } G\text{)}\\
\text{iff } & \widehat{\delta}(a_I, u^k) \in F, \text{ some } k \geq 1 && (A_{g_u} = \{\widehat{\delta}(a_I, u^n) \mid n \geq 1\})\\
\text{iff } & u^k \in L = L(\mathbb{A}, a_I), \text{ some } k \geq 1 && \text{(definition of acceptance)}\\
\text{iff } & u \in Root(L) && \text{(definition } Root\text{)}
\end{aligned}
$$

From this it follows that $L(\mathbb{A}', i) = Root(L(\mathbb{A}, a_I))$, as required. ◀

Proof of Theorem 22. Let $\mathbb{P} = (P, \rho, \Pi)$ be a parity automaton, let $\mathbb{A}_\mathbb{P} = (P, X, \rho, \xi, \sigma, F)$ be its associated lasso automaton, and let N denote the range of Π. Fix a state $q \in P$.

For the inclusion $Streams(\mathbb{P}, q) \subseteq Streams(\mathbb{A}_\mathbb{P}, q)$, assume that $\mathbb{P}, q \rhd \alpha$ for some stream α. Without loss of generality we may assume that α can be split as $\alpha = u\vec{v}$, such that, with $p := \widehat{\rho}(q, u)$, we have $p \overset{v_i}{\Longrightarrow}_\rho p$ and $Inf(q, \alpha) = \rho^\circ(p, v_i)$, for each $i \in \omega$. Define $v_{ij} := v_i \cdots v_{j-1}$ for $i, j \in \omega$ with $i < j$. It follows by construction of $\mathbb{A}_\mathbb{P}$ that

$$\widehat{\sigma{:}\xi}(p, v_{ij}) \in F \text{ for each } i, j \text{ with } i < j. \tag{6}$$

Note that the equivalence relation on $\omega_< = \{(i, j) \in \omega^2 \mid i < j\}$ given by $(i, j) \sim (k, l)$ if $\widehat{\sigma{:}\xi}(p, v_{ij}) = \widehat{\sigma{:}\xi}(p, v_{kl})$, has finite index. It then follows by Ramsey's Theorem that there is an infinite subset $Y \subseteq \omega$, and a unique $z \in F$ such that $\widehat{\sigma{:}\xi}(p, v_{ij}) = z$, for each pair $i, j \in Y$ with $i < j$.

Enumerate $Y = \{k_0, k_1, \ldots\}$ with $k_0 < k_1 < \cdots$, and define $u' := v_{0k_0}$, for each $i \in \omega$, $w_i := v_{k_i k_{i+1}}$. It is obvious from these definitions that

$$\alpha = uu'\vec{w}. \tag{7}$$

In addition, we claim that

$$q \overset{uu'}{\Longrightarrow}_\rho p \tag{8}$$

and that

$$p \overset{w_i}{\Longrightarrow}_\rho p \text{ and } p \overset{w_i}{\Longrightarrow}_{\sigma:\xi} z, \text{ for all } i \in \omega. \tag{9}$$

For the proof of (8) and (9), observe that $p \xRightarrow{v_j}_\rho p$ for each j, so that we find $p \xRightarrow{u'}_\rho p$ and $p \xRightarrow{w_i}_\rho p$ for each i on the basis of u' and each w_i being finite concatenations of v_j's. From this we immediately obtain the first statement in (9), but also (8) because $q \xRightarrow{u}_\rho p \xRightarrow{u'}_\rho p$. The remaining, second, statement in (9) follows directly from the assumption on z and the definition of the w_j.

Finally, it follows directly from (7), (8) and (9) that α is accepted by $(\mathbb{A}_P, q)$.

For the inclusion $\mathit{Streams}(\mathbb{A}_\mathbb{P}, q) \subseteq \mathit{Streams}(\mathbb{P}, q)$, assume that $(\mathbb{A}_\mathbb{P}, q)$ accepts some given stream α. Then by definition we can split this stream as $\alpha = u\vec{v}$, and find states $p \in P$ and $t \in F$ such that $q \xRightarrow{u}_\rho p$ and $p \xRightarrow{v_i}_\rho p$, $p \xRightarrow{v_i}_{\sigma:\xi} t$ for each $i \in \omega$. Fix some $i \in \omega$. By definition of $\mathbb{A}_\mathbb{P}$ it follows from $p \xRightarrow{v_i}_{\sigma:\xi} t$ that $s_I^p \xRightarrow{v_i}_\xi t$ in $\mathbb{X}_p$, where $\mathbb{X}_p$, s_I^p and ξ are as in Definition 12. But since it is straightforward to verify from $p \xRightarrow{v_i}_\rho p$ that $\rho^\circ(p, v_i)$ consists of all states traversed on the cycle $p \xRightarrow{v_i}_\rho p$, it follows that $t \in (P \times N)^P$ satisfies $t(p) = (p, \max(\Pi[\rho^\circ(p, v_i)]))$. Since t is accepting it follows that $\max(\Pi[\rho^\circ(p, v_i)])$ is even. Since this holds for each $i \in \omega$, it easily follows that $(\mathbb{P}, q)$ accepts α. ◀

Proof of Theorem 23. Fix $\mathbb{A} = (P, X, \rho, \xi, \sigma, F)$ and $\mathbb{A} = (P', X', \rho', \xi', \sigma', F')$, and let $h : \mathbb{A} \to \mathbb{A}'$ be an Ω-morphism. The proof of the inclusion $\mathit{Streams}(\mathbb{A}, q) \subseteq \mathit{Streams}(\mathbb{A}', q')$ is routine and left to the reader.

For the opposite inclusion, assume that $(\mathbb{A}', hq)$ accepts some stream α. Let $u, (v_i)_{i\in\omega}$, $p' \in P'$ and $z' \in F'$ bear witness to this fact, in the sense that $\alpha = u\vec{v}$, $hq \xRightarrow{u}_{\rho'} p'$ and $p' \xRightarrow{v_i}_{\rho'} p'$, $p' \xRightarrow{v_i}_{\sigma':\xi'} z'$ for each $i \in \omega$.

Define $p_0 := \widehat{\rho}(q, u)$ and $p_{i+1} := \widehat{\rho}(p_i, v_i)$ for $i > 0$. Observe that $hp_i = p'$ for all $i \in \omega$. Since P is finite, so is the set $Q := \{p_i \mid i \in \omega\}$, and hence, some element p of Q is traversed infinitely often in the path $p_0 \xRightarrow{v_0}_\rho p_1 \xRightarrow{v_1}_\rho p_2 \cdots$. In other words, there is an infinite subset $K = \{k_0, k_1, \ldots\} \subseteq \omega$ with $k_0 < k_1 < \ldots$ such that $p_0 \xRightarrow{v_0 \cdots v_{k_0-1}}_\rho p$ and $p \xRightarrow{v_{k_i} \cdots v_{k_{i+1}-1}}_\rho p$ for all $i \in \omega$. Define $u' := v_0 \cdots v_{k_0-1}$ and $w_i := v_{k_i} \cdots v_{k_{i+1}-1}$, then we have $\alpha = uu'\vec{w}$, $q \xRightarrow{uu'}_\rho p$ and $p \xRightarrow{w_i}_\rho p$, for each $i \in \omega$.

Define, for $i < j \in \omega$, the word $w_{ij} := w_i \cdots w_{j-1}$ and the state $z_{ij} := \widehat{\sigma:\xi}(p, w_{ij})$. By Ramsey's Theorem there must be an infinite set $N \subseteq \omega$ and a single element $z \in X$ such that $z_{ij} = z$ for all $i, j \in N$. Note that $hz_{ij} = z'$ for all $i, j \in N$, since h is an Ω-morphism. Write $N = \{n_0, n_1, \ldots\}$ with $n_0 < n_1 < \cdots$, and define $u'' := w_0 \cdots w_{n_0-1}$ and $s_i := w_{n_i} \cdots w_{n_{i+1}-1}$, then clearly we have

$$\alpha = uu'u''\vec{s}. \tag{10}$$

Second, we claim that

$$q \xRightarrow{uu'u''}_\rho p. \tag{11}$$

To see this, note that u'' is a finite concatenation of w_i's. Since $p \xRightarrow{w_i}_\rho p$ for each i, it follows that $p \xRightarrow{u''}_\rho p$, and hence we obtain (11) from $q \xRightarrow{uu'}_\rho p \xRightarrow{u''}_\rho p$. In addition, we have

$$p \xRightarrow{s_i}_\rho p \text{ and } p \xRightarrow{s_i}_{\sigma:\xi} z, \text{ for all } i \in \omega. \tag{12}$$

Here the first statement follows from each s_i being a finite concatenation of w_j's, and the second is by assumption on z.

Finally, the fact that $\mathbb{A}, q \rhd \alpha$ is immediate from (10), (11) and (12). ◀

Proof of Corollary 24. It follows from Fact 17 that there is an initialized parity automaton $(\mathbb{P}, p)$ such that $\mathit{Lassos}(\mathbb{A}, q) = \mathit{Lassos}(\mathbb{P}, p)$, and from Fact 14 that $\mathit{Lassos}(\mathbb{P}, p) =$

$Lassos(\mathbb{A}_{\mathbb{P}}, p)$. From this it is immediate that $Lassos(\mathbb{A}, q) = Lassos(\mathbb{A}_{\mathbb{P}}, p)$, and so Fact 10 yields that $\mathbb{A}, q \leftrightarrows \mathbb{A}_{\mathbb{P}}, p$. Then we may derive by Proposition 8 and Theorem 23 that $Streams(\mathbb{A}, q) = Streams(\mathbb{A}_{\mathbb{P}}, p)$, and so by Theorem 22 we find that $Streams(\mathbb{A}, q) = Streams(\mathbb{P}, p)$. This immediately gives that $Streams(\mathbb{A}, q)$ is regular, and gathering our findings we obtain that $Lassos(Streams(\mathbb{A}, q)) = Lassos(Streams(\mathbb{P}, p)) = Lassos(\mathbb{P}, p) = Lassos(\mathbb{A}, q)$. ◀

Proof of Theorem 25. The direction from left to right is immediate by Proposition 8 and Theorem 23. The opposite direction follows from Corollary 24 and Fact 10. ◀

Proof of Theorem 28. The first part of the theorem is immediate by the fact that the quotient map from $\mathbb{A}$ to $\mathbb{A}_{/\leftrightarrows}$ is an Ω-morphism. For part 2, assume that $Streams(\mathbb{A}', p') = L$. Then by Theorem 25 we have that $\mathbb{A}, p \leftrightarrows \mathbb{A}', p'$, so that a routine argument shows that every state in $\mathbb{A}'_{p'}$ is bisimilar to some state in $\mathbb{A}$. This yields a natural map h from $\mathbb{A}'_{p'}$ to $\mathbb{A}_{/\leftrightarrows}$ such that $h(p') = \overline{p}$. We leave it for the reader to verify that this map is an Ω-morphism. ◀

Proof of Theorem 32. With $\mathbb{A} = (P, X, \rho, \xi, \sigma, F)$, define the maps $h_0 : P \to \mathcal{P}(C^* \times C^+)$ and $h_1 : X \to \mathcal{P}C^*$ by putting

$$\begin{array}{lcl} h_0(p) & := & Lassos(\mathbb{A}, p), \\ h_1(x) & := & L((X, \xi, F), x). \end{array}$$

It is a routine exercise to verify that this is an Ω-morphism. For uniqueness, let $h' : \mathbb{A} \to \mathbb{Z}$ be an Ω-morphism. Then for every spoke state p of $\mathbb{A}$ we find that $hp = Lassos(\mathbb{A}, p) = Lassos(\mathbb{Z}, h'p) = h'p$. ◀

Proof of Theorem 34. We confine ourselves to a sketch. The equivalence of (1) and (2) is a direct consequence of the Theorems 31 and 32, so it suffices to show that (2) $\Leftrightarrow$ (3).

For this purpose, fix a lasso language L. We first show that, for any pair of words u_0, u_1:

$$u_0 \equiv u_1 \text{ iff } \widehat{\zeta_0}(L, u_0) = \widehat{\zeta_0}(L, u_1). \tag{13}$$

Second, for all words u and pairs of nonempty words v_0, v_1, with $L_u := \widehat{\zeta_0}(L, u) (= \{(u'v) \mid (uu', v) \in L\})$ we can prove:

$$v_0 \equiv_{[u]} v_1 \text{ iff } \widehat{\sigma{:}\zeta_1}(L_u, v_0) = \widehat{\sigma{:}\zeta_1}(L_u, v_1) \tag{14}$$

By the previous two steps we may conclude that $(u_0, v_0) \equiv (u_1, v_1)$ if and only if, starting from L in the spoke part of $\mathbb{Z}$, consuming either u_0 or u_1 takes us to the same state L', and from L', consuming either v_0 or v_1, takes us to the same state L'' in the loop part of $\mathbb{Z}$. Now let Y_L be the set of spoke states reachable from L, and for $M \in Y_L$, let Y'_M be the set of loop states reachable from M. It then follows by the above observation that the equivalence classes of $\equiv_L$ are in one-to-one correspondence with the set $\biguplus_{M \in Y_L} M$. This suffices to prove that $\equiv_L$ has finite index iff L generates a finite subcoalgebra in $\mathbb{Z}$. ◀

Proof of Proposition 36. Fix a stream $\alpha = \vec{v}$. By lasso determinacy of L there is an infinite set $Y \subseteq \omega$ as in the definition. Write $Y = \{n_0, n_1, \ldots\}$ with $n_0 < n_1 < \cdots$. Define $w_0 := v_0 \cdots v_{n_0}$ and $w_{i+1} := v_{n_i+1} \cdots v_{n_{i+1}}$. Then $\alpha = \vec{w}$ and for all $j, k \in \omega$ with $j < k$ we have

$$\alpha \in L \iff (w_0 \cdots w_j)(w_{j+1} \cdots w_k)^\omega \in L \tag{15}$$

Now by the LD property of L' there is an infinite set $Y' \in \omega$ as in the definition. Take any two elements $j, k \in Y'$ with $j < k$. Then we have

$$\alpha \in L' \iff (w_0 \cdots w_j)(w_{j+1} \cdots w_k)^\omega \in L' \tag{16}$$

It is then immediate by (15), (16) and the assumption $\mathit{Lassos}(L) = \mathit{Lassos}(L')$ that

$$\alpha \in L \iff \alpha \in L'.$$

Since α was arbitrary, this shows that $L = L'$. ◀

Proof of Theorem 37. The equivalence of (2) and (3) is immediate by Theorem 34, and the implication from (1) to (2/3) follows from the same result, together with the observation that regular ω-languages are lasso-determined.

For the remaining implication (2/3 $\Rightarrow$ 1), assume (2), and let L' be the stream language accepted by the pointed Ω-coalgebra $(\mathbb{Z}, \mathit{Lassos}(L))$. It follows that $\mathit{Lassos}(L) = \mathit{Lassos}(L')$, and since L', being regular, is lasso-determined, Proposition 36 implies $L = L'$, which immediately yields the regularity of L. ◀