Topics in Ω-Automata
A Journey through Lassos, Algebra, Coalgebra and Expressions

MSc Thesis *(Afstudeerscriptie)*

written by

Mike Cruchten
(born April 22nd, 1995 in Ettelbruck, Luxembourg)

under the supervision of **prof. dr. Yde Venema** and **dr. Tobias Kappé**, and submitted to the Examinations Board in partial fulfillment of the requirements for the degree of

MSc in Logic

at the *Universiteit van Amsterdam.*

Date of the public defense:
June 27th, 2022

Members of the Thesis Committee:
dr. Benno van den Berg (Chair)
dr. Nick Bezhanishvili
dr. Tobias Kappé (Supervisor)
dr. Jurriaan Rot
prof. dr. Yde Venema (Supervisor)

Institute for Logic, Language and Computation

Abstract

In recent years, automata theory has been brought into the realm of category theory which allowed the generalisation of results and provided new perspectives. With the introduction of the Ω-automaton, an ω-automaton which arises as a coalgebra, there is a wide range of directions to explore. Unlike other types of ω-automata, Ω-automata are deterministic and their acceptance is local. Like deterministic finite automata, they can be minimised and as they are coalgebraic, we can use categorical techniques to study them.

We investigate Ω-automata from multiple perspectives and introduce several new tools along the way. Lassos are studied in more depth giving rise to the Lasso Representation Lemma which specifies the exact relation between lassos and ultimately periodic words. We also partially establish the connection between Wilke algebras and Ω-automata, highlighting the relationship between language acceptance and language recognition. Using the already existing Myhill-Nerode theorem for ω-regular languages, we carry out a strengthening which gives a lower bound for Ω-automata based on the index of the Nerode congruence we define. Furthermore, we present a Brzozowski construction for Ω-automata using lasso expressions. This gives new insights into more effective ways of constructing Ω-automata and brings us closer to the question of size constraints. Finally, we discuss some minimisation procedures for Ω-automata, in particular the Brzozowski minimisation algorithm and an algorithm exploiting a dual equivalence.

Acknowledgements

First and foremost I would like to thank my supervisors Yde Venema and Tobias Kappé for making this thesis possible. Thank you Tobias for showing me why I fell in love with automata and regular expressions in the first place and also for constantly believing in me, your kind words of support have always been very uplifting. Thank you Yde, first of all for the many fantastic lectures in which you made the complex look so simple, but also for your unique points of view and criticisms (for which I am extremely grateful). In what seemed like perfect harmony, the both of you never failed to inspire and encourage me to try my best. I hope to follow your examples in the future.

I would also like to thank the thesis committee for their time and kind feedback. I had the pleasure to take courses under almost all of you, which made the defense even more enjoyable and special.

Next, I would like to thank the friends that I made while at the ILLC. I have worked with some of you and discussed many fascinating topics with others and I hope you enjoyed it as much as I did. Thank you, Jasmijn, for having worked and grown with me. We shared some great memories over the past years and I am extremely grateful for our friendship. A second thank you goes to Geoffrey, who has supported me more than anyone else over the past two years and shown me the beauty which lies in poppies.

Finally, I would also like to thank my family and friends for their continuous support. Your visits and the many occasions we spent together have always brightened my days.

Contents

Introduction

Since its establishment around the middle of the 20th century, automata theory has grown to become a very wide and important field with its many applications and connections to other areas ([24]). It is primarily used in theoretical computer science with applications in formal verification, complexity theory, model checking ([31]) and many more. There are various types of automata, some of the most widely studied of which are automata operating on finite words such as deterministic finite automata (DFAs) and nondeterministic finite automata (NFAs). Another class of automata are the so-called ω-automata, which are automata that operate on infinite words, also called streams.

Over the last decades, the research community has started to bring categorical tools into the picture to study automata, which by now are considered standard examples of coalgebras ([28, 4, 3, 17, 16]). Many constructions and algorithms on DFAs have been translated into the categorical framework giving new perspectives to well-known results. Some examples include

1. Minimisation: Partition refinement (also known as Hopcroft's algorithm) and Brzozowski minimisation ([1, 7])
2. Soundness and completeness for regular expressions ([16])
3. Myhill-Nerode Theorem ([7, 12])

Some common ω-automata are Büchi, Muller, Rabin and parity automata. They can be classified according to their acceptance condition and whether they are deterministic or not. All of the ones listed share a common characteristic for their acceptance condition, namely that it is non-local in the sense that acceptance is based on the states that are traversed infinitely often. This makes it difficult to describe them as coalgebras.

A solution to this problem arises through the following fact:

Fact 1.1 ([9]) *An ω-regular language is uniquely characterised by its ultimately periodic fragment, i.e. for ω-regular languages $\mathscr{L}, \mathscr{K}$ we have*

$$\mathscr{L} = \mathscr{K} \iff UP(\mathscr{L}) = UP(\mathscr{K})$$

From this it follows that it is sufficient to accept the ultimately periodic fragment of an ω-regular language. This gave rise to a multitude of frameworks such as $L_{\$}$-automata ([9]), families of DFAs (FDFAs) ([2]) and Ω-automata ([11]), the latter of which is presented as a coalgebra. Having ω-automata which are coalgebraic paves a path towards minimisation algorithms, a Myhill-Nerode theorem, and also allows us to make connections to other fields, all within the categorical framework.

In order to portray how well-behaved Ω-automata really are, we argue that in many ways, Ω-automata behave like DFAs. More specifically, a lot of useful constructions for DFAs can be adapted to constructions for Ω-automata. As we go through the different sections, this pattern becomes more and more apparent and provides strong arguments in favour of studying Ω-automata in more detail.

Although Ω-automata resemble DFAs in many ways, the constructions we realise are usually more involved than similar constructions for DFAs. It is for this reason that we have to develop additional tools such as lasso expressions.

In Section 2, we will introduce the reader to the preliminaries, giving definitions and fixing some conventions. From this section onwards we address in each section a particular problem or question about Ω-automata.

Section 3 lays out the categorical foundation for reasoning about Ω-automata through category theory. Unlike DFAs, the base category for Ω-automata is $\mathbf{Set}^2$ whose objects are pairs of sets and morphisms are pairs of functions. We establish some basic results which allow us to lift adjunctions and equivalences from **Set** to $\mathbf{Set}^2$. This is useful especially when tackling minimisation procedures for Ω-automata. The motivation behind this is that we will lift certain equivalences and adjunctions, which are used in constructions for DFAs, to Ω-automata.

In Section 4, we reason more about lassos which play a central role throughout the thesis. On the one hand they provide a great deal of intuition when reasoning about Ω-automata. On the other hand, they form the basis for tools such as the Lasso Representation Lemma and lasso expressions which play an important role in some of the constructions we undertake. The relation between Ω-automata and lassos can be compared to that of DFAs and finite words.

Section 5 is devoted to the algebraic theory of ω-regular languages. The link between the algebraic and coalgebraic theory of ω-regular languages has been studied for multiple types of ω-automata, but not for Ω-automata yet. We provide a construction which turns an Ω-automaton into a Wilke algebra and show that we can go from language acceptance to language recognition. Interestingly, we can introduce the construction by analogy to DFAs.

Section 6 discusses the well-known Myhill-Nerode theorem. This theorem has already been discussed in [12] but we will undertake some slight changes to strengthen the result. As for DFAs, we define a Myhill-Nerode equivalence relation but contrary to the classical equivalence relation, it will be two-sorted. Moreover, for the actual theorem we have to impose an additional constraint on the ω-language. For the rest the theorems are analogous in that we get ω-regularity if and only if the index of the equivalence relation is finite. Moreover, the size of the minimal Ω-automaton accepting an ω-regular language $\mathscr{L}$ is bound from below by the index of the two-sorted equivalence relation.

Section 7 looks at the construction of Ω-automata from ω-regular expressions. The inspiration for our construction comes from Brzozowski's construction which gives a procedure to turn a regular expression into a DFA. We show that for the most part the construction is straight-forward and pinpoint the main challenge. Solving this challenge is done by using lasso expressions. We show how to build lasso expressions that are somewhat equivalent to ω-regular expressions and how this can be used to work towards the construction of a Brzozowski Ω-automaton.

In Section 8 we will look at minimisation procedures for Ω-automata. Some common minimisation procedures for DFAs such as Brzozowski's minimisation algorithm and partition refinement have been framed in categorical terms. We take these results and lift them to Ω-automata, showing that the minimisation procedures work analogously for Ω-automata.

Preliminaries

Throughout the thesis, we fix a set of letters Σ called the *alphabet*. A *finite word* over Σ is a map $u : n \to \Sigma$ for some $n \in \mathbb{N}$ and $|u| = n$ is the length of u. The unique word of length 0 is called the *empty word* and denoted ϵ. An *infinite word* (or *stream*) over Σ is a map $\alpha : \omega \to \Sigma$. The collection of finite, finite non-empty, ultimately periodic and infinite words is denoted respectively by Σ^*, Σ^+, Σ^{up} and Σ^ω. We use a, b, c for letters, u, v, w for finite words and α, β, γ for streams. The *concatenation* of a finite word u with a finite word v (resp. infinite word α) is denoted by juxtaposition uv (resp. $u\alpha$). We denote by u^ω the stream obtained by concatenating u with itself infinitely many times. A stream of the form u^ω is called *periodic*, and one of the form uv^ω is called *ultimately periodic*. For a word u and a positive integer k, we write $u[k]$ for the k^{th} symbol of u (starting at 0). We use $u[i \dots j]$ to denote the substring of u that consists of all the symbols from i to j included. For a formula φ, we write $[\varphi]$ for the function

$$[\varphi] = \begin{cases} 1 & \text{, if } \varphi \text{ is true} \\ 0 & \text{, otherwise.} \end{cases}$$

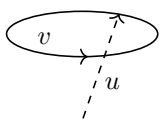
A language (resp. ω-language) is a subset $L \subseteq \Sigma^*$ (resp. $\mathscr{L} \subseteq \Sigma^\omega$). It is called regular (resp. ω-regular) if it is accepted by an automaton (resp. ω-automaton). We denote languages by L, K, M and ω-languages by $\mathscr{L}, \mathscr{K}$. We write $\mathrm{P}(\mathscr{L})$ for the periodic, and $\mathrm{UP}(\mathscr{L})$ for the ultimately periodic fragment of $\mathscr{L}$, i.e.

$$\mathrm{P}(\mathscr{L}) = \{\alpha \in \mathscr{L} \mid \exists u \in \Sigma^+ : \alpha = u^\omega\}$$
$$\mathrm{UP}(\mathscr{L}) = \{\alpha \in \mathscr{L} \mid \exists u \in \Sigma^*, v \in \Sigma^+ : \alpha = uv^\omega\}$$

For a DFA without initial state $\mathbb{A} = (Q, \delta, F)$, we write $L(\mathbb{A}, i)$ for the language accepted at $i \in Q$.

Set is the category whose objects are sets and whose morphisms are functions between sets. We write **2** for the discrete two-element category. For a category $\mathbf{C}$, we write $\widehat{\mathbf{C}}$ for the category of presheaves on $\mathbf{C}$ whose objects are functors $F : \mathbf{C}^{\mathrm{op}} \to \mathbf{Set}$ and whose morphisms are natural transformations. We denote the final object in **Set** by $1 = \{\star\}$ and the initial set by 0 or $\emptyset$. As a convention, we use ! to denote a morphism which is unique due to a universal property.

Definition 2.1 ([12]) A *lasso* is a pair $(u, v) \in \Sigma^* \times \Sigma^+$ representing the stream uv^ω. We call u the *spoke* and v the *loop* of the lasso. The collection of lassos is denoted by Σ^{*+}.



Before we introduce Ω-automata, we start with a more general type of automaton, the *lasso automaton*. Lasso automata operate on lassos as defined above. An Ω-automaton is a particular kind of lasso automaton which will be defined later on.

Definition 2.2 ([12]) A *lasso automaton* is a tuple $\mathbb{A} = (X, Y, i, \rho, \sigma, \xi, F)$ where

$$\rho : X \to X^{\Sigma} \qquad \sigma : X \to Y^{\Sigma} \qquad \xi : Y \to Y^{\Sigma},$$

$i \in X$ and $F \subseteq Y$. We call X the *spoke states* and Y the *loop states*. The maps ρ, σ, ξ are called the *spoke*, *switch* and *loop transition* respectively. The spoke state i is called the *initial state* and the collection F of loop states are called the *accepting states*. Instead of specifying F, we sometimes give a map $\chi_F : Y \to 2$ where $y \in F \iff \chi_F(y) = 1$.

We first give an example of a lasso automaton and show how it operates on a lasso. Afterwards we will introduce some additional definitions which will help us to formally define when a lasso is accepted by a lasso automaton.

Let $\mathbb{A}$ be the following lasso automaton where $X = \{0, 1, 2\}$ and $Y = \{3, 4, 5, 6, 7\}$. The spoke, switch and loop transitions are given by solid, dotted and dashed lines respectively.

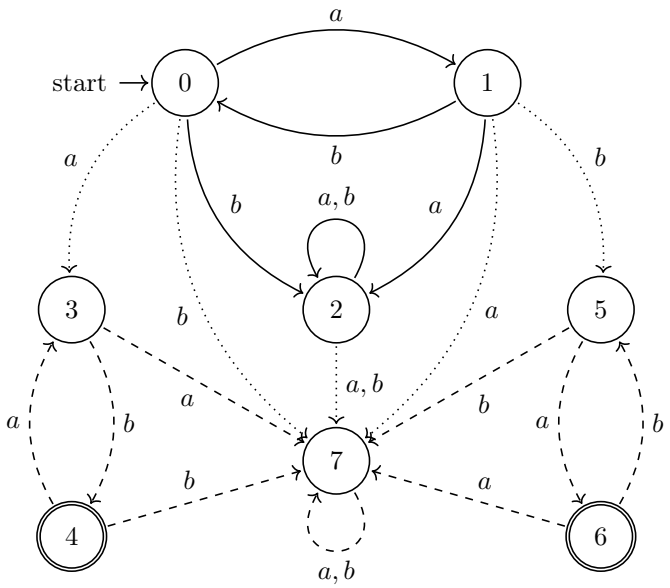


Figure 1: Lasso-automaton $\mathbb{A}$

We consider the following lasso $(ab, abab)$. The idea is that the spoke part of the lasso is read via the spoke transitions of the lasso automaton. After we are done reading the spoke part of the lasso, we move on to read the loop part, taking the switch transition for the first letter and then proceeding via the loop transitions.

So for $(ab, abab)$, we first traverse the spoke part reading ab

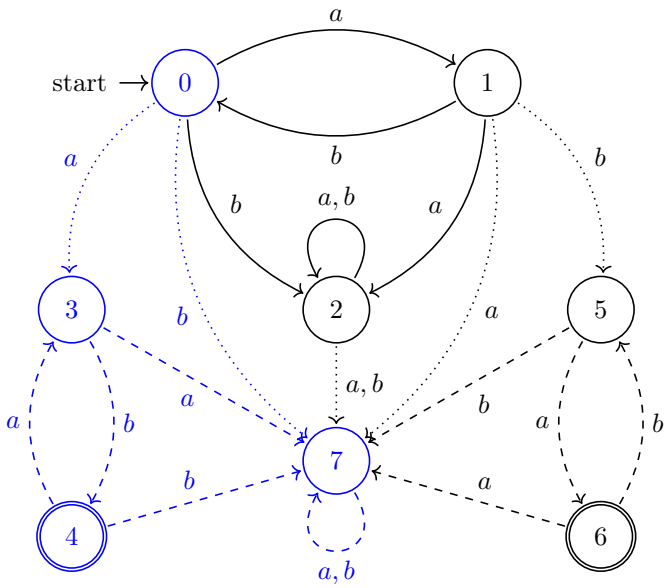
$$0 \xrightarrow{a} 1 \xrightarrow{b} 0,$$

which takes us from 0 back to 0. From here we read the loop part $abab$ this time using the switch and loop transitions

$$0 \overset{a}{\dashrightarrow} 3 \overset{b}{\dashrightarrow} 4 \overset{a}{\dashrightarrow} 3 \overset{b}{\dashrightarrow} 4.$$

As we ended up in an accepting state the lasso $(ab, abab)$ is accepted by $\mathbb{A}$.

Given a spoke state x, we can consider x together with all the $y \in Y$ which are reachable from x just by means of the switch transition and loop transitions. This forms a DFA called the *loop automaton* at x which we denote by $(\mathbb{A}_\ell, x)$. In the following diagram we highlight the loop automaton at 0 for $\mathbb{A}$, $(\mathbb{A}_\ell, 0)$, in blue.



The language accepted by the loop automaton at x is called the *loop language* of x and is denoted by $\mathrm{Loop}(x)$ (or alternatively by $L(\mathbb{A}_\ell, x)$). We make this definition more precise.

Definition 2.3 ([12]) For a lasso automaton $\mathbb{A} = (X, Y, i, \rho, \sigma, \xi, F)$, we define the *loop automaton of* $\mathbb{A}$ to be the DFA $\mathbb{A}_\ell = (X \uplus Y, \sigma : \xi, F)$ where $(\sigma : \xi) : X \uplus Y \to Y^\Sigma$ is the transition map defined as

$$\sigma : \xi(z) = \begin{cases} \sigma(z) & \text{, if } z \in X \\ \xi(z) & \text{, if } z \in Y. \end{cases}$$

For a spoke state $x \in X$, we call the initialised DFA $(\mathbb{A}_\ell, x)$ the *loop automaton at* x. The regular language accepted by $(\mathbb{A}_\ell, x)$ is called the *loop language at* x and denoted $\mathrm{Loop}(x)$.

This definition alone allows us to describe when a lasso of the form (ϵ, u) is accepted at some spoke state x. As the spoke part of the lasso is empty, we proceed by reading u starting at x and using the switch and loop transitions. If we reach an accepting state, then u is accepted by the loop automaton at x, so $u \in \mathrm{Loop}(x)$. In other words we have

$$(\epsilon, u) \text{ is accepted at } x \iff u \in \mathrm{Loop}(x).$$

The other definition we require deals with naturally extending ρ, σ and ξ. Given some $x \in X$ and some $a \in \Sigma$, then $\rho(x)(a)$ is the spoke state we end up at after reading a from x. This map can naturally be extended to a map $\widehat{\rho}$ where $\widehat{\rho}(x)(u)$ is the state we end up at after reading an arbitrary finite word $u \in \Sigma^*$ from x.

Definition 2.4 For $\delta : Z \to Z^{\Sigma}$, we inductively define the map $\widehat{\delta} : Z \to Z^{\Sigma^*}$ by

$$\widehat{\delta}(z)(\epsilon) = z \qquad\qquad \widehat{\delta}(z)(ua) = \delta(\widehat{\delta}(z,u), a),$$

and we define $\widetilde{\delta} : Z \to Z^{\Sigma^+}$ by $\widetilde{\delta}(z)(ua) = \delta(\widetilde{\delta}(z,u), a)$.

Definition 2.5 For a map $\delta : Z_1 \to Z_2^{\Sigma}$, we will sometimes use the equivalent map $\delta^\flat : Z_1 \times \Sigma \to Z_2$ obtained by uncurrying. If no confusion arises, we will use δ and $\delta^\flat$ interchangeably. For a map $\tau : Z_1 \to Z_2^{Z_3}$, we define the map $\tau^\sharp : Z_3 \to Z_2^{Z_1}$ as

$$\tau^\sharp(z_3)(z_1) = \tau(z_1)(z_3).$$

We can now state when a lasso is accepted by a lasso automaton.

Definition 2.6 ([12]) Let $\mathbb{A} = (X, Y, i, \rho, \sigma, \xi, F)$ be a lasso automaton. A lasso $(u, av) \in \Sigma^{*+}$ is accepted by $\mathbb{A}$ if $\widehat{\xi}(\sigma(\widehat{\rho}(i,u), a), v) \in F$. This is equivalent to the following:

1. $av \in \mathrm{Loop}(\widehat{\rho}(i,u))$,
2. $\widetilde{\sigma : \xi}(\widehat{\rho}(i,u), av) \in F$.

The *lasso language* accepted by $\mathbb{A}$ is given by

$$\mathrm{Lassos}(\mathbb{A}, i) = \{(u,v) \in \Sigma^{*+} \mid \widetilde{\sigma : \xi}(\widehat{\rho}(i,u), v) \in F\}.$$

A lasso language is called *regular* if it is accepted by a finite lasso automaton.

Item (1.) states that av is accepted by the loop automaton at $\widehat{\rho}(i,u)$, i.e. at the state we reach after reading u from the initial state. The second item uses $\widetilde{\sigma : \xi}$ and we included it because it sometimes allows us to be more concise with our notation (as seen with the definition of $\mathrm{Lassos}(\mathbb{A}, i)$).

Next, we discuss Ω-automata. The lasso automaton $\mathbb{A}$ in Figure 1 has a special property. Let $u_1 v_1^\omega = u_2 v_2^\omega$ for $u_1, u_2 \in \Sigma^*$ and $v_1, v_2 \in \Sigma^+$, then

$$(u_1, v_1) \in \mathrm{Lassos}(\mathbb{A}, i) \iff (u_2, v_2) \in \mathrm{Lassos}(\mathbb{A}, i).$$

This property is also called the *saturation property* [2] and states that for any two lassos which represent the same ultimately periodic word, they must either be both accepted or rejected. An Ω-automaton is a lasso automaton which has the saturation property.

Another characterisation of Ω-automata is through the notions of circularity and coherence ([11]).

Definition 2.7 ([12]) A regular language L is called *circular* if for all $u \in \Sigma^+$ we have $u \in L \iff u^k \in L$ for all $k \geq 1$. A lasso automaton $\mathbb{A}$ is called *circular* if $\mathrm{Loop}(x)$ is circular for all $x \in X$ and it is called *coherent* if

$$au \in \mathrm{Loop}(x) \iff ua \in \mathrm{Loop}(\rho(x,a))$$

for $a \in \Sigma, u \in \Sigma^+$. An Ω-*automaton* is a circular and coherent lasso automaton.

We state an important fact about Ω-automata.

Proposition 2.8 ([11]) *For a lasso-automaton $\mathbb{A} = (X, Y, i, \rho, \sigma, \xi, \chi_F)$ (where all states are reachable from i), the following are equivalent:*

1. *Lassos$(\mathbb{A}, i) = \{(u, v) \in \Sigma^{*+} \mid uv^\omega \in \mathscr{L}\}$ for some ω-regular language $\mathscr{L}$,*
2. *$\mathbb{A}$ is circular and coherent.*

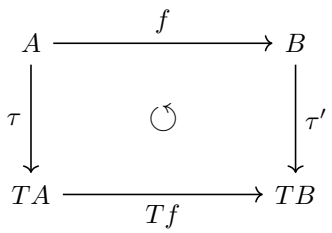
Although we introduced lasso automata as operating primarily on lassos, we can define what it means for an Ω-automaton to accept streams. The interested reader is referred to [12, Definition 20]. We do not give this definition as it is not used in the thesis. We will however state the following proposition

Proposition 2.9 ([12]) *Let $\mathbb{A} = (X, Y, i, \rho, \sigma, \xi, \chi_F)$ be an Ω-automaton and let $L(\mathbb{A}, i)$ be the ω-language accepted by $\mathbb{A}$ according to [12, Definition 20], then $L(\mathbb{A}, i)$ is ω-regular, and*

$$\mathit{Lassos}(\mathbb{A}, i) = \{(u, v) \in \Sigma^{*+} \mid uv^\omega \in L(\mathbb{A}, i)\}.$$

As briefly touched upon in the introduction, Ω-automata (and lasso automaton for that matter) arise naturally as coalgebras. When discussing them as such, we usually omit the initial state and call them Ω-coalgebras. As coalgebras, we have access to the concepts of Ω-coalgebra morphisms and bisimulations. We will introduce lasso automata now through coalgebras.

Definition 2.10 ([17]) Let $T : \mathbf{C} \to \mathbf{C}$ be an endofunctor on some category $\mathbf{C}$. A T-*coalgebra* is a pair (A, τ) where A is an object in $\mathbf{C}$ and $\tau : A \to TA$. A map $f : (A, \tau) \to (B, \tau')$ is a T-*coalgebra morphism* if $f : A \to B$ is a morphism in $\mathbf{C}$ which makes the following diagram commute:



In [11], the authors introduce an endofunctor $\Omega : \hat{\mathbf{2}} \to \hat{\mathbf{2}}$, where they describe objects of $\hat{\mathbf{2}}$ as pairs of sets and morphisms between them as pairs of functions. We will get back to this in Section 3 and show that $\hat{\mathbf{2}}$ is indeed isomorphic to the product category of **Set** with itself. For what follows, we will stick with the original authors and think of $\hat{\mathbf{2}}$ as the product category $\mathbf{Set} \times \mathbf{Set}$ which we also write as $\mathbf{Set}^2$.

Definition 2.11 ([12]) Let $\Omega : \mathbf{Set}^2 \to \mathbf{Set}^2$ be the endofunctor given by

$$\begin{aligned}\Omega(X,Y) &= (X^\Sigma \times Y^\Sigma, Y^\Sigma \times 2),\\ \Omega(f,g) &= (f\circ(-)\times g\circ(-), g\circ(-)\times \mathrm{id}_2).\end{aligned}$$

Definition 2.12 ([12]) Let $\mathbb{A} = ((X,Y),(\langle\rho,\sigma\rangle, \langle\xi,\chi_F\rangle))$ be an Ω-coalgebra. Given some spoke state $i \in X$, $(\mathbb{A}, i)$ is a lasso automaton and we sometimes write $\mathbb{A}$ simply as $\mathbb{A} = (X, Y, \rho, \sigma, \xi, F)$.

It is easy to see how lasso automata are just pointed Ω-coalgebras (i.e. Ω-coalgebras with a distinguished initial state) and vice-versa. Given (X,Y), we have that $\Omega(X,Y) = (X^\Sigma \times Y^\Sigma, Y^\Sigma \times 2)$. Specifying a map $(X,Y) \to (X^\Sigma \times Y^\Sigma, Y^\Sigma \times 2)$ is the same as giving four maps $\rho : X \to X^\Sigma$,$\sigma : X \to Y^\Sigma$, $\xi : Y \to Y^\Sigma$ and $\chi_F : Y \to 2$.

Definition 2.13 ([12]) Let $\mathbb{A} = (X, Y, \rho, \sigma, \xi, \chi_F)$ and $\mathbb{A}' = (X', Y', \rho', \sigma', \xi', \chi'_F)$. An Ω-coalgebra morphism $(f_0, f_1) : \mathbb{A} \to \mathbb{A}'$ is a pair of morphisms $f_0 : X \to X'$, $f_1 : Y \to Y'$ in $\mathbf{Set}^2$ such that the following diagrams commute:

$$\begin{array}{ccc} X & \xrightarrow{f_0} & X' \\ {\scriptstyle\rho}\downarrow & \circlearrowleft & \downarrow{\scriptstyle\rho'} \\ X^\Sigma & \xrightarrow[f_0\circ(-)]{} & X'^\Sigma \end{array} \qquad \begin{array}{ccc} X & \xrightarrow{f_0} & X' \\ {\scriptstyle\sigma}\downarrow & \circlearrowleft & \downarrow{\scriptstyle\sigma'} \\ Y^\Sigma & \xrightarrow[f_1\circ(-)]{} & Y'^\Sigma \end{array} \qquad \begin{array}{ccc} Y & \xrightarrow{f_1} & Y' \\ {\scriptstyle\xi}\downarrow & \circlearrowleft & \downarrow{\scriptstyle\xi'} \\ Y^\Sigma & \xrightarrow[f_1\circ(-)]{} & Y'^\Sigma \end{array} \qquad \begin{array}{ccc} Y & \xrightarrow{f_1} & Y' \\ {\scriptstyle\chi_F}\downarrow & \circlearrowleft & \downarrow{\scriptstyle\chi'_F} \\ 2 & \xrightarrow[\mathrm{id}_2]{} & 2 \end{array}$$

Finally, we also define the notion of bisimulation between two Ω-coalgebras.

Definition 2.14 ([12]) Two Ω-automata $\mathbb{A} = (X, Y, \rho, \sigma, \xi, F)$ and $\mathbb{A}' = (X', Y', \rho', \sigma', \xi', F')$ are *bisimilar* if there exists a pair of relations (Z_0, Z_1) with $Z_0 \subseteq X \times X', Z_1 \subseteq Y \times Y'$ such that for all $(x, x') \in Z_0$, $(y, y') \in Z_1$ and $a \in \Sigma$ we have

1. $(\rho(x,a), \rho(x',a)) \in Z_0$,
2. $(\sigma(x,a), \sigma(x',a)) \in Z_1$,
3. $(\xi(y,a), \xi(y',a)) \in Z_1$,
4. $y \in F \iff y' \in F'$.

Two lasso automata $(\mathbb{A}, i)$ and $(\mathbb{A}', i')$ are *bisimilar*, written $(\mathbb{A}, i) \leftrightarroweq (\mathbb{A}', i')$ if there is a bisimulation between $\mathbb{A}, \mathbb{A}'$ linking i and i'.

We want to point out that bisimulation captures language equivalence for Ω-automata.

Proposition 2.15 ([12]) *Let $(\mathbb{A}, i), (\mathbb{A}', i)$ be two Ω-automata, then*

$$(\mathbb{A}, i) \leftrightarroweq (\mathbb{A}', i') \iff L(\mathbb{A}, i) = L(\mathbb{A}', i').$$

Convention 2.16 Throughout the thesis it will be convenient to refer to some arbitrary Ω-coalgebra. For that reason we introduce the convention that we may reference to an arbitary Ω-coalgebra as given below:

$$\mathbb{A} = (X, Y, \rho, \sigma, \xi, F).$$

If we need an initial state, it will be denoted by i. We will usually use x's for spoke and y's for loop states respectively.

Convention 2.17 For the most part of this thesis (apart from section 7), whenever we talk about arbitrary lassos and Ω-automata, we may without loss of generality assume that they are reachable, i.e. that all states can be reached from the initial state. The justification for this is that removing non-reachable states does not impact the language that is accepted.

The Category $\mathbf{Set}^2$

The base category of Ω-coalgebras is the category of presheaves on the discrete two element category which we denoted $\hat{\mathbf{2}}$. As we will undertake quite a few constructions in this category, it will be advantageous to first explore its properties. That being said, the following results will mostly be used in Section 8 and just form a technical foundation, so the reader may want to skip the proofs. We assume familiarity with several categorical concepts such as presheaves, limits, colimits, exponentials, equivalences and adjunctions.

This section will demonstrate, that the category $\hat{\mathbf{2}}$ is isomorphic to the category $\mathbf{Set}^2$ which has as objects pairs of sets and as morphisms pairs of functions. By virtue of $\hat{\mathbf{2}}$ being a category of presheaves, it follows that we are dealing with a well-behaved category which, amongst others, is complete, cocomplete and cartesian closed. Moreover, essentially all constructions can be done pointwise. In the case of limits and colimits this is again a consequence of us working in a category of presheaves. Moreover, as the discrete two element category $\mathbf{2}$ has no non-trivial morphisms, exponentials can be computed pointwise.

Towards the end of the section we will show that we can lift adjunctions and equivalences from $\mathbf{Set}$ to $\mathbf{Set}^2$ in a natural way.

Proposition 3.1 *The categories $\boldsymbol{Set}^2$ and $\hat{\mathbf{2}}$ are isomorphic.*

Proof. In order to show the claim, we will explicitly construct an isomorphism. Consider the following two functors:

- $I : \mathbf{Set}^2 \to \hat{\mathbf{2}}$
  1. $I(A,B)(0) = A$, $I(A,B)(1) = B$
  2. $I(f,g)_0 = f$, $I(f,g)_1 = g$
- $J : \hat{\mathbf{2}} \to \mathbf{Set}^2$
  1. $JF = (F(0), F(1))$
  2. $J\mu = (\mu_0, \mu_1)$

We claim that $I \circ J = \mathrm{id}_{\hat{\mathbf{2}}}$ and $J \circ I = \mathrm{id}_{\mathbf{Set}^2}$. Starting with the objects, let F be a presheaf on $\mathbf{2}$, then

$$(I \circ J(F))(0) = I(F(0), F(1))(0) = F(0)$$
$$(I \circ J(F))(1) = I(F(1), F(1))(1) = F(1).$$

Moreover, as $\mathbf{2}$ has only identity morphisms, $(I \circ J)(F) = F$. Next let $(A,B) \in \mathbf{Set}^2$, and let $F = I(A,B)$, then

$$(J \circ I)(A,B) = J(F) = (F(0), F(1)) = (A,B).$$

As for the morphisms, let $\sigma : F \implies G$ be a natural transformation, then

$$(I \circ J)(\sigma)_0 = I(\mu_0, \mu_1)_0 = \mu_0$$
$$(I \circ J)(\sigma)_1 = I(\mu_0, \mu_1)_1 = \mu_1.$$

so $(I \circ J)(\mu) = \mu$. Finally, for $(f, g) \in \mathbf{Set}_1^2$, let $I(f, g) = \mu$ (so $\mu_0 = f, \mu_1 = g$), then

$$(J \circ I)(f, g) = J(\mu) = (\mu_0, \mu_1) = (f, g),$$

which completes the proof. □

Corollary 3.2 ***Set***^{2} *is complete, cocomplete and cartesian closed. Limits and colimits are computed pointwise, so are exponentials. A morphism is epi (resp. mono) if it is componentwise epi (resp. mono).*

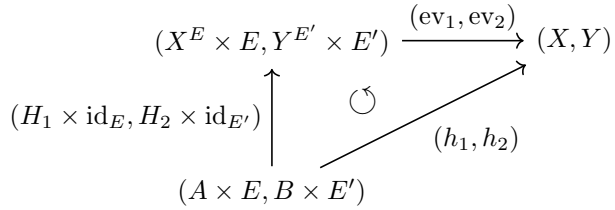
Proof. This follows from the fact that limits and colimits in a category of presheaves are computed pointwise. Again by the properties of presheaves, we also get completeness and cocompleteness ([22]).

As it is not true in general that exponentials can be computed pointwise in a presheaf we will show that exponentials can in fact be computed pointwise in $\mathbf{Set}^2$.

Let $(X, Y), (E, E'), (A, B) \in \mathbf{Set}^2$. As limits are computed pointwise in a category of presheaves, we get

$$(X^E, Y^{E'}) \times (E, E') = (X^E \times E, Y^{E'} \times E')$$
$$(A, B) \times (E, E') = (A \times E, B \times E').$$

Our claim is that the exponential $(X, Y)^{(E,E')}$ is isomorphic to $(X^E, Y^{E'})$. We will show that $(X^E, Y^{E'})$ satisfies the universal property of the exponential. So we have to show that for each (A, B) and $(h_1, h_2) : (A, B) \times (E, E') \to (X, Y)$ there is a unique $(H_1, H_2) : (A, B) \to (X^E, Y^{E'})$ which makes the diagram below commute.



This is equivalent to showing that the function

$$(e_1, e_2) : \mathrm{Hom}_{\mathrm{Set}^2}((A, B), (X^E, Y^{E'})) \longrightarrow \mathrm{Hom}_{\mathrm{Set}^2}((A \times E, B \times E'), (X, Y))$$
$$(H_1, H_2) \longmapsto (\mathrm{ev}_1 \circ (H_1 \times \mathrm{id}_E), \mathrm{ev}_2 \circ (H_2 \times \mathrm{id}_{E'}))$$

is a bijection. If we show that it is a bijection componentwise then we are done as monos and epis are computed pointwise and in this case, a map is a bijection if it is both mono and epi.

Let $h_1 : A \times E \to X$, define

$$H_1 : A \longrightarrow X^E$$
$$a \longmapsto (H_1(a) : E \to X) : e \mapsto h_1(a, e),$$

which shows that e_1 is surjective as

$$\mathrm{ev}_1 \circ (H_1 \times \mathrm{id}_E)(a, e) = \mathrm{ev}_1(H_1(a), e) = H_1(a)(e) = h_1(a, e).$$

As for the injectivity of e_1, let

$$\mathrm{ev}_1 \circ (H_1 \times \mathrm{id}_E)(a, e) = \mathrm{ev}_1 \circ (H_1' \times \mathrm{id}_E)(a, e),$$

then by the definition of ev_1 we have

$$\mathrm{ev}_1 \circ (H_1 \times \mathrm{id}_E)(a, e) = H_1(a, e),$$

hence

$$H_1(a, e) = H_1'(a, e),$$

which shows injectivity. The proof for e_2 is analogous. It follows that exponentials can be computed pointwise. □

Lemma 3.3 *Let $\boldsymbol{C}, \boldsymbol{D}$ be two categories and F_1, F_2, G_1, G_2 be four functors*

such that $F_1 \dashv F_2$ and $G_1 \dashv G_2$, then

$$F_1 \times G_1 \dashv F_2 \times G_2.$$

Proof. Let $F_1 \dashv F_2$ and $G_1 \dashv G_2$, so there are natural bijections

$$m^F_{A,B} : \mathrm{Hom}_{\mathbf{C}}(F_1(A), B) \to \mathrm{Hom}_{\mathbf{Set}}(A, F_2(B))$$
$$m^G_{A,B} : \mathrm{Hom}_{\mathbf{D}}(G_1(A), B) \to \mathrm{Hom}_{\mathbf{Set}}(A, G_2(B)).$$

As is standard notation, if two maps f and g correspond to each other under a natural bijection, we will call them transposes of one another. As we are dealing with two natural bijections at the same time, we write $\overline{f}^F = g$ to mean that f and g are transposes under m^F and similarly write $\overline{f}^G = g$ for transposes under m^G. We define $H_1 = F_1 \times G_1$ and $H_2 = F_2 \times G_2$ and want to show that $H_1 \dashv H_2$. We proceed by defining

$$m_{A,B} : \mathrm{Hom}_{\mathbf{C}\times\mathbf{D}}(H_1(A_0, A_1), (B_0, B_1)) \to \mathrm{Hom}_{\mathbf{Set}^2}((A_0, A_1), H_2(B_0, B_1)),$$

where

$$m_{A,B}(f, g) = (m^F_{A,B}(f), m^G_{A,B}(g)) \qquad \text{and} \qquad m^{-1}_{A,B}(f', g') = (m^{F,-1}_{A,B}(f'), m^{G,-1}_{A,B}(g')),$$

and showing that m is a natural bijection.

Bijectivity follows by how we defined m, in particular, note that

$$\overline{\overline{(f, g)}} = \overline{(\overline{f}^F, \overline{g}^G)} = (\overline{\overline{f}^F}^F, \overline{\overline{g}^G}^G) = (f, g).$$

It remains to show that the naturality condition holds. Let $(f, g) : (A_0', A_1') \to (A_0, A_1)$, $(f', g') : (B_0, B_1) \to (B_0', B_1')$. We show that the following diagram commutes.

$$\begin{array}{ccc}
\mathrm{Hom}_{\mathbf{C}\times\mathbf{D}}(H_1(A_0,A_1),(B_0,B_1)) & \xrightarrow{m_{A,B}} & \mathrm{Hom}_{\mathbf{Set}^2}((A_0,A_1),H_2(B_0,B_1)) \\
\Big\downarrow{\scriptstyle (f',g')\circ -\circ H_1(f,g)} & \circlearrowleft & \Big\downarrow{\scriptstyle H_2(f',g')\circ -\circ (f,g)} \\
\mathrm{Hom}_{\mathbf{C}\times\mathbf{D}}(H_1(A_0',A_1'),(B_0',B_1')) & \xrightarrow[m_{A',B'}]{} & \mathrm{Hom}_{\mathbf{Set}^2}((A_0',A_1'),H_2(B_0',B_1'))
\end{array}$$

Let $(i,j) : H_1(A_0,A_1) \to (B_0,B_1)$, then

$$\begin{aligned}
\overline{(f',g')\circ(i,j)\circ H_1(f,g)} &= \overline{\big(f'\circ i\circ F_1(f), g'\circ j\circ G_1(g)\big)} \\
&= \big(\overline{f'\circ i\circ F_1(f)}^{F}, \overline{g'\circ j\circ G_1(g)}^{G}\big) \\
&= \big(F_2(f')\circ \bar{i}^{F}\circ f, G_2(g')\circ \bar{j}^{G}\circ g\big) \qquad \text{(nat. of } m^F, m^G) \\
&= H_2(f',g')\circ\overline{(i,j)}\circ(f,g),
\end{aligned}$$

as required. □

Lemma 3.4 *Let $F_1 : \boldsymbol{Set} \to \boldsymbol{C}$ and $F_2 : \boldsymbol{Set} \to \boldsymbol{D}$ be two equivalences with weak-inverses G_1 and G_2 respectively, then*

$$F_1 \times F_2 : \boldsymbol{Set}^2 \to \boldsymbol{C}\times\boldsymbol{D}$$

is an equivalence with weak-inverse $G_1 \times G_2$.

Proof. Let $F = F_1\times F_2$ and $G = G_1\times G_2$. Since F_1, F_2 are equivalences with weak-inverses G_1, G_2, we have four natural isomorphisms:

$$\sigma_1 : G_1\circ F_1 \overset{\cong}{\Longrightarrow} \mathrm{id}_{\mathbf{Set}} \qquad \rho_1 : F_1\circ G_1 \overset{\cong}{\Longrightarrow} \mathrm{id}_{\mathbf{C}}$$

$$\sigma_2 : G_2\circ F_2 \overset{\cong}{\Longrightarrow} \mathrm{id}_{\mathbf{Set}} \qquad \rho_2 : F_2\circ G_2 \overset{\cong}{\Longrightarrow} \mathrm{id}_{\mathbf{D}}.$$

We define the natural transformation $\sigma : G\circ F \implies \mathrm{id}_{\mathbf{Set}^2}$ for $(A,B)\in\mathbf{Set}^2$ by

$$\big(\sigma_{A,B} : (G\circ F)(A,B) \longrightarrow (A,B)\big) = (\sigma_{1,A}, \sigma_{2,B}).$$

Similarly, we define a natural transformation $\rho : F\circ G \implies \mathrm{id}_{\mathbf{C}\times\mathbf{D}}$ for $(A,B)\in\mathbf{C}\times\mathbf{D}$ by

$$\big(\rho_{A,B} : (F\circ G)(A,B) \longrightarrow (A,B)\big) = (\rho_{1,A}, \rho_{2,B}).$$

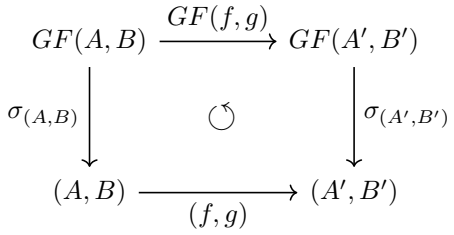
We will show for σ that it is a natural isomorphism, the proof for ρ is analogous. We have to show that σ is an isomorphism at each component and that it satisfies naturality. For the first part, we define

$$\sigma^{-1}_{A,B} = (\sigma^{-1}_{1,A}, \sigma^{-1}_{2,B}),$$

and it follows that, for all $(A,B)\in\mathbf{Set}^2$:

$$\begin{aligned}
\sigma_{A,B}\circ\sigma^{-1}_{A,B} &= (\sigma_{1,A}\circ\sigma^{-1}_{1,A}, \sigma_{2,B}\circ\sigma^{-1}_{2,B}) = (\mathrm{id}_A, \mathrm{id}_B) = \mathrm{id}_{(A,B)} \\
\sigma^{-1}_{A,B}\circ\sigma_{A,B} &= (\sigma^{-1}_{1,A}\circ\sigma_{1,A}, \sigma^{-1}_{2,B}\circ\sigma_{2,B}) = (\mathrm{id}_{G_1F_1(A)}, \mathrm{id}_{G_2F_2(B)}) = \mathrm{id}_{GF(A,B)}.
\end{aligned}$$

Let $(f,g) : (A,B) \to (A',B')$ in $\mathbf{Set}^2$ and consider the following diagram



This diagram commutes since:

$$\begin{aligned}\sigma_{(A',B')} \circ GF(f,g) &= \sigma_{(A',B')}(G_1F_1f, G_2F_2g) \\ &= (\sigma_{1,A'} \circ G_1F_1f, \sigma_{2,B'} \circ G_2F_2g) \\ &= (f \circ \sigma_{1,A}, g \circ \sigma_{2,B}) \\ &= (f,g) \circ \sigma_{(A,B)}.\end{aligned}$$

Hence σ, ρ are natural isomorphisms and F is an equivalence with weak-inverse G. □

Proposition 3.5 *The functor $S : \boldsymbol{Set}^2 \to \boldsymbol{Set}^2$ which acts on objects and morphisms by switching their places (i.e. $S(A,B) = (B,A)$ and $S(f,g) = (g,f)$) is an isomorphism.*

Proof. Clearly S is a functor as

$$S((f_1,g_1) \circ (f_2,g_2)) = ((g_1 \circ g_2),(f_1 \circ f_2)) = (g_1,f_1) \circ (g_2,f_2) = S(f_1,g_1) \circ S(f_2,g_2),$$

$$S(\mathrm{id}_{(A,B)}) = S(\mathrm{id}_A, \mathrm{id}_B) = (\mathrm{id}_B, \mathrm{id}_A) = \mathrm{id}_{(B,A)} = \mathrm{id}_{S(A,B)}.$$

We will show that $S \circ S = \mathrm{id}_{\mathbf{Set}^2}$. Let $(A,B) \in \mathbf{Set}^2$, then

$$S \circ S(A,B) = S(B,A) = (A,B).$$

Similarly, for morphisms (f,g) we get

$$S \circ S(f,g) = S(g,f) = (f,g).$$ □

Rewriting Lassos

Lassos occupy an important position in the study of Ω-automata. They were introduced as representatives of ultimately periodic words and the primary object that Ω-coalgebras operate on. As such, it is important to study their structure and exact relationship with ultimately periodic words.

More specifically, given some ultimately periodic word uv^ω, we see that there are infinitely many lassos that represent it. Intuitively however, we understand that there is a canonical representative for each ultimately periodic word. This section introduces some tools to reason about lassos. When do two lassos represent the same ultimately periodic word? Is there a canonical representative and how do we find it? To answer these questions, we introduce rewrite rules on the collection of lassos with the purpose of reducing and simplifying them.

Before defining the rewrite rules, we briefly explain them through an example to gain some insights into what exactly they do. Notice that

$$\begin{array}{lll} ua(va)(va)(va)\ldots & \text{is equivalent to} & u(av)(av)(av)\ldots \\ u(\underbrace{v\ldots v}_{k \text{ times}})(\underbrace{v\ldots v}_{k \text{ times}})(\underbrace{v\ldots v}_{k \text{ times}})\ldots & \text{is equivalent to} & uvvvvvvvvvvvvv\ldots \end{array}$$

Expressing this in terms of lassos gives:

$$\begin{array}{lll} (ua, va) & \text{is equivalent to} & (u, av) \\ (u, v^k) & \text{is equivalent to} & (u, v) \qquad (k \geq 1). \end{array}$$

Indeed these observations form the basis of our two rewrite rules. We will allow to rewrite the expressions on the left to those on the right.

At this stage, it is useful to see that these two rules relate to circularity and coherence of Ω-coalgebras, i.e.

$$\begin{array}{lll} (a, va) \in \text{Lassos}(x) \iff (\epsilon, av) \in \text{Lassos}(x) & & \text{(Coherence)} \\ (\epsilon, v^k) \in \text{Lassos}(x) \iff (\epsilon, v) \in \text{Lassos}(x) & & \text{(Circularity)} \end{array}$$

Throughout this section, we show that our two rewrite rules are strong enough to obtain, for each ultimately periodic word, a canonical lasso representative. Moreover, we define a type of lasso equivalence such that two lassos are equivalent exactly when the ultimately periodic words they represent are the same.

Definition 4.1 We define the following reduction rules on Σ^{*+}:

$$\frac{a \in \Sigma \quad (ua, va)}{(u, av)}\,(\gamma_1) \qquad\qquad \frac{(u, v^k) \quad (k > 1)}{(u, v)}\,(\gamma_2)$$

We write $(u, v) \to_{\gamma_i} (u', v')$ if (u, v) reduces to (u', v') in one step under γ_i and we let $\to_\gamma = \to_{\gamma_1} \cup \to_{\gamma_2}$.

In order to show that γ always leads to a normal form, we leverage Newman's Lemma. This involves showing that the reduction rules are strongly normalising and locally confluent, from which we can deduce that they are confluent and so each lasso reduces to a unique normal form. As we move through the section we shall remind the reader of common definitions and Newman's Lemma, which can all be found in more detail in [29].

Lemma 4.2 *The reduction rule γ strongly normalises, i.e. it does not allow for infinite reduction chains.*

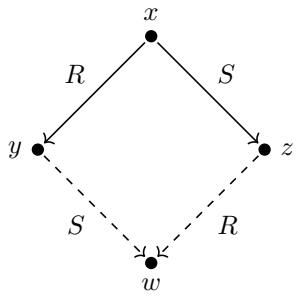
Proof. For the reduction rules γ_1, γ_2, we have that

$$(u, v) \to_{\gamma_1} (u', v') \implies |u'| < |u| \text{ and } |v'| = |v|,$$
$$(u, v) \to_{\gamma_2} (u', v') \implies |u'| = |u| \text{ and } |v'| < |v|.$$

So each reduction rule strictly decreases either the spoke or loop part of the lasso and keeps the other part constant. As the words occurring in a lasso are both finite, we cannot have an infinite chain of γ-reductions. □

Definition 4.3 ([29]) Let R, S be two binary relations on some set X. We say that R *commutes* with S if

$$\forall x, y, z \in X : (xRy \wedge xSz \implies \exists w \in X : ySw \wedge zRw).$$



Lemma 4.4 *The binary relations $\to_{\gamma_1}$ and $\to_{\gamma_2}$ commute.*

Proof. Let $(u, v), (u_1, v_1)$ and (u_2, v_2) be such that

$$(u, v) \to_{\gamma_1} (u_1, v_1) \tag{1}$$
$$(u, v) \to_{\gamma_2} (u_2, v_2) \tag{2}$$

Then by (1) it follows that there are $a \in \Sigma$ and $w \in \Sigma^+$ such that $u = u_1 a$, $v = aw$ and $v_1 = wa$. By (2) we have $u = u_2$ and that there is some $k > 0$ such that $v_2^k = v$. As $v_2^k = v = aw$ there is some w' such that $v_2 = aw'$. This gives us the following:

$$(u, v) = (u_1 a, (aw')^k) \qquad (u_1, v_1) = (u_1, w'(aw')^{k-1}a) = (u_1, (w'a)^k)$$

$$(u_2, v_2) = (u_1 a, aw')$$

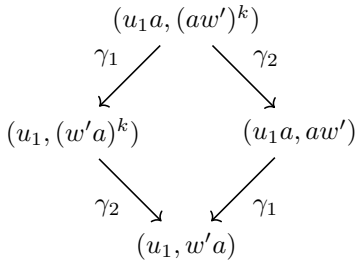
Now consider $(u_3, v_3) = (u_1, w'a)$. We have that

$$(u_1, v_1) = (u_1, (w'a)^k) \to_{\gamma_2} (u_1, w'a)$$

and

$$(u_2, v_2) = (u_1 a, aw') \to_{\gamma_1} (u_1, w'a)$$

as shown in the diagram to the right. From this it follows that $\to_{\gamma_1}$ and $\to_{\gamma_2}$ commute.



□

Definition 4.5 ([29]) Let R be a binary relation on some set X, then

1. R has the *diamond property* if it commutes with itself.
2. R has the *Church-Rosser property*, or is *confluent*, if the transitive-reflexive closure R^* has the diamond property.
3. R is *locally confluent*, or *weakly Church-Rosser*, if it satisfies

$$\forall x, y, z \in X : (xRy \wedge xRz \implies \exists w \in X : yR^*w \wedge zR^*w).$$

Lemma 4.6 *The binary relation $\to_\gamma$ is locally confluent.*

Proof. Let $(u, v) \to_\gamma (u_1, v_1)$ and $(u, v) \to_\gamma (u_2, v_2)$. We distinguish three cases.

1. Assume w.l.o.g. that $(u, v) \to_{\gamma_1} (u_1, v_1)$ and $(u, v) \to_{\gamma_2} (u_2, v_2)$. This case is covered by Lemma 4.4.
2. Assume that $(u, v) \to_{\gamma_1} (u_1, v_1)$ and $(u, v) \to_{\gamma_1} (u_2, v_2)$, then clearly $u = u_1 a = u_2 a$ for some $a \in \Sigma$ and $v_1 = v_2 = aw$ for some w with $v = wa$. Hence $(u_1, v_1) = (u_2, v_2)$.
3. Let $(u, v) \to_{\gamma_2} (u_1, v_1)$ and $(u, v) \to_{\gamma_2} (u_2, v_2)$, then $v = (v_1)^{k_1}$ and $v = (v_2)^{k_2}$ for some $k_1, k_2 > 0$. As $v = v_1^{k_1} = v_2^{k_2}$ we can find v_3, ℓ_1 and ℓ_2 with $v_1 = v_3^{\ell_1}$ and $v_2 = v_3^{\ell_2}$. To see this, let k_3 be the greatest element of the set $\{\ell \mid \exists w \in \Sigma^+ : w^\ell = v\}$, which must exist as v is finite. Then choose $v_3 = v[0 \ldots (k_3 - 1)]$. For v_i, we must then have $\ell_i = \frac{k_3}{k_i}$ and claim that $v_3^{\ell_i} = v_i$. If not then they would differ at some position, but $(v_3^{\ell_i})^{k_i} = v_3^{k_3} = v = v_i^{k_i}$, which is a contradiction. So it follows that

$$(u_1, v_1) \to_{\gamma_2} (u, v_3) \qquad \text{and} \qquad (u_2, v_2) \to_{\gamma_2} (u, v_3).$$

□

Lemma 4.7 (Newman's Lemma, [29]) *If a binary relation R on some set X is strongly normalising and locally confluent, then it is confluent.*

Definition 4.8 ([29]) A lasso $(u, v) \in \Sigma^{*+}$ is a *normal form* if there does not exist $(u', v') \in \Sigma^{*+}$ such that $(u, v) \to_\gamma (u', v')$.

Corollary 4.9 *The binary relation $\to_\gamma$ is confluent and every lasso in Σ^{*+} reduces to a unique normal form.*

Proof. This is a direct consequence of Newman's Lemma. □

Definition 4.10 Let $\sim_\gamma$ be the least equivalence relation including $\to_\gamma$.

Lemma 4.11 (Lasso Representation Lemma) *Let* $(u, v), (u', v') \in \Sigma^{*+}$*, then*

$$(u, v) \sim_\gamma (u', v') \iff uv^\omega = u'v'^\omega.$$

Proof. We show that if $(u, v) \to_\gamma (u', v')$, then uv^ω is the same as $u'v'^\omega$. To make the argument more formal, recall that an ultimately periodic word is just a function $\alpha : \omega \to \Sigma$ which is ultimately periodic. Given a lasso (u, v) with $|u| = i$ and $|v| = j$ we can define

$$\begin{aligned} uv^\omega : \omega &\longrightarrow \Sigma \\ k &\longmapsto uv^\omega[k] = \begin{cases} u[k] & \text{, if } k < i \\ v[(k-i)\%j] & \text{, otherwise} \end{cases} \end{aligned}$$

where $a\%b$ denotes the remainder of the integer division of a by b. For $(ua, va) \to_{\gamma_1} (u, av)$ with $|u| = i, |v| = j$ we get the two functions

$$\begin{aligned} ua(va)^\omega[k] &= \begin{cases} u[k] & \text{, if } k < i \\ a & \text{, if } k = i \\ v[(k-(i+1))\%(j+1)] & \text{, if } k > i \text{ and } (k-(i+1)) \neq j \pmod{j+1} \\ a & \text{, if } k > i \text{ and } (k-(i+1)) = j \pmod{j+1} \end{cases} \\ &= \begin{cases} u[k] & \text{, if } k < i \\ a & \text{, if } k \geq i \text{ and } k = i \pmod{j+1} \\ v[(k-(i+1))\%(j+1)] & \text{, if } k > i \text{ and } k \neq i \pmod{j+1} \end{cases} \end{aligned}$$

$$u(av)^\omega[k] = \begin{cases} u[k] & \text{, if } k < i \\ a & \text{, if } k \geq i \text{ and } k = i \pmod{j+1} \\ v[(k-i)\%(j+1)-1] & \text{, otherwise.} \end{cases}$$

This is the same following this observation:

$$(k-(i+1))\%(j+1) = (k-i)\%(j+1) - 1 \qquad \text{, for } k \neq i \pmod{j+1}.$$

The proof for $\to_{\gamma_2}$ is similar.

From this it follows that if $(u, v) \sim_\gamma (u', v')$, then they both reduce to the same normal form (w_1, w_2) and as the reductions preserve representability, we have $uv^\omega = w_1 w_2^\omega = u'v'^\omega$.

For the other direction, note that given some ultimately periodic word uv^ω seen as a function, we can extract the normal form. Let i be the least element of the set

$$\{\ell \mid \exists j\ \forall k > \ell : uv^\omega[k] = uv^\omega[k+j]\}.$$

Next let j be the smallest element of the set

$$\{\ell \mid \forall k > i : uv^\omega[k] = uv^\omega[k+\ell]\}.$$

From i, j, we can get the normal form. Note that by how we defined i, j, we have that $(uv^\omega[0] \dots uv^\omega[i], uv^\omega[i+1] \dots uv^\omega[i+j])$ is a lasso representing uv^ω. Moreover, it must be the normal form as otherwise we could reduce it which would contradict the minimality of i and j. Now assume that $uv^\omega = u'v'^\omega$ but $(u, v) \not\sim_\gamma (u', v')$, then $(u, v), (u', v')$ would have to reduce to two distinct normal forms, but this cannot be the case as uv^ω and $u'v'^\omega$ yield the same normal form. □

Remark 4.12 In [12], the authors call two lassos $(u, v), (u', v')$ bisimilar if and only if they represent the same ultimately periodic word. In this sense, we may of course think of lassos themselves as coalgebras, where $\sim_\gamma$ captures bisimilarity of lassos.

Definition 4.13 For an ultimately periodic word uv^ω, we define its *normal form* $\mathrm{nf}(uv^\omega)$ to be the normal form of the lasso (u, v).

This is well-defined by the Lasso Representation Lemma 4.11.

Wilke Algebras

Regular languages can be studied, in one part, through the concept of language acceptance by an automaton. We could call this approach coalgebraic, seeing automata as key examples of coalgebras, in particular when talking about deterministic finite automata (DFAs). However, there is also an algebraic approach. We can study regular languages through monoids or semigroups ([27, Chapter 10],[24]). These two approaches form a good starting point and so we start this section by discussing DFAs and monoids.[1]

Given some deterministic finite automaton $\mathbb{A} = (Q, i, \delta, F)$, we may consider its transition map $\delta : Q \to Q^\Sigma$, or equivalently look at the map $\delta^\sharp : \Sigma \to Q^Q$ where

$$\delta^\sharp(a)(q) = \delta(q)(a)$$

and which we can think of as associating to each $a \in \Sigma$ a map $f \in Q^Q$ representing all the paths in the automaton with label a. We can naturally extend $\delta^\sharp$ to a map $\widehat{\delta^\sharp} : \Sigma^* \to Q^Q$ where

$$\widehat{\delta^\sharp}(\epsilon) = \mathrm{id}_Q \qquad \text{and} \qquad \widehat{\delta^\sharp}(ua) = \delta^\sharp(a) \circ \widehat{\delta^\sharp}(u).$$

This reveals that $\mathrm{Im}(\widehat{\delta^\sharp}) \subseteq Q^Q$ can be equipped with a monoid structure (the functions in the image of $\widehat{\delta^\sharp}$ are closed under function composition and id_Q is the identity element) and that $\widehat{\delta^\sharp}$ is in fact a monoid homomorphism whose domain is the free monoid over Σ. What makes this structure, which we refer to as $\mathbb{T}(A)$ (the transition monoid of $\mathbb{A}$), so interesting, is that it allows us to talk about regular languages using monoids and monoid homomorphisms. To make this more explicit, let $f : \Sigma^* \to \mathbb{A}$ be a monoid homomorphism. A subset $L \subseteq \Sigma^*$ is said to be *recognised by* f if there is some $P \subseteq A$ such that $L = f^{-1}[P]$.

In order to see how this relates to language recognition consider $L(\mathbb{A}) \subseteq \Sigma^*$ and let

$$P = \{f \in \mathbb{T}(\mathbb{A}) \mid f(i) \in F\},$$

then

$$L(\mathbb{A}) = \widehat{\delta^\sharp}^{-1}[P].$$

Thus the map $\widehat{\delta^\sharp}$ recognises the regular language accepted by $\mathbb{A}$. This shows that every language which is accepted by some DFA is also recognised by a monoid homomorphism into a finite monoid. Fortunately, the converse is true as well, giving us the following fact:

Fact 5.1 ([27]) *A regular language $L \subseteq \Sigma^*$ is accepted by a deterministic finite automaton if and only if it is recognised by a monoid homomorphism from the free monoid over Σ to a finite monoid.*

Quite naturally, there is an interest in obtaining an algebraic counterpart to the acceptance of ω-regular languages by ω-automata. This counterpart is given by ω-semigroups

[1] We could also have chosen semigroups instead of monoids.

or Wilke algebras. It is standard to use ω-semigroups but they involve an operation of infinite arity. Thomas Wilke introduced a different structure, Wilke algebras, whose operations are of finite arity. Furthermore, he showed that every Wilke algebra can uniquely be extended to an ω-semigroup, and every ω-semigroup can uniquely be equipped with the structure of a Wilke algebra ([33, 25]). For our purposes we decided to use Wilke algebras.

For various types of ω-automata such as non-deterministic Büchi automata and Muller automata, constructions to and from ω-semigroups exist (for a survey see [10, 25], these texts also give a more detailed introduction to the algebraic theories surrounding ω-regular languages).

In this section, our aim is to provide a solution to the following problem.

> Given an Ω-automaton $\mathbb{A}$, construct a finite Wilke algebra $\mathbb{T}(\mathbb{A})$ such that the ω-regular language $L(\mathbb{A})$ accepted by $\mathbb{A}$ is also recognised by a Wilke algebra homomorphism from the Wilke algebra $(\Sigma^+, \Sigma^\omega)$ to $\mathbb{T}(\mathbb{A})$.

We don't give a solution to the converse of this problem but discuss it at the end of this section.

Before we start, we introduce some basic definitions. The concept of a Wilke algebra was originally introduced as a right binoid by Thomas Wilke [33].

Definition 5.2 (Wilke Algebra, [33]) A *Wilke algebra* is a two-sorted algebra $\mathbb{W} = (W_f, W_i, \cdot, \times, {}^\omega)$ with

$$\cdot : W_f \times W_f \to W_f \qquad \times : W_f \times W_i \to W_i$$

$$(-)^\omega : W_f \to W_i,$$

satisfying $\forall u, v, w \in W_f, \beta \in W_i$:

$$(u \cdot v) \cdot w = u \cdot (v \cdot w) \tag{A}$$

$$(u \cdot v) \times \beta = u \times (v \times \beta) \tag{MA}$$

$$u \times (v \cdot u)^\omega = (u \cdot v)^\omega \tag{Co}$$

$$(\underbrace{u \cdot \ldots \cdot u}_{n \text{ times}})^\omega = u^\omega \qquad (n > 0) \tag{Ci}$$

These four equations represent *associativity* (A), *mixed associativity* (MA), *coherence* (Co) and *circularity* (Ci) respectively. A Wilke algebra is called *complete* if every $\beta \in W_i$ is of the form uv^ω for some $u, v \in W_f$.

Definition 5.3 ([33]) A pair of functions $(f_1, f_2) : (A_f, A_i) \to (B_f, B_i)$ between two Wilke algebras $\mathbb{A}$ and $\mathbb{B}$ is a *Wilke algebra homomorphism* if, for all $u, v \in A_f, \alpha \in A_i$:

1. $f_1(u \cdot_{\mathbb{A}} v) = f_1(u) \cdot_{\mathbb{B}} f_1(v)$
2. $f_2(u \times_{\mathbb{A}} \alpha) = f_1(u) \times_{\mathbb{B}} f_2(\alpha)$
3. $f_2(u^{\omega_{\mathbb{A}}}) = f_1(u)^{\omega_{\mathbb{B}}}$

The conditions express, that f_1 respects the product, f_1, f_2 respect the mixed product and f_2 respects the ω-star.

Example 5.4 Some examples of Wilke algebras arise naturally over the alphabet Σ:

1. Take Σ^+ for the finite sort and Σ^{up} for the infinite sort. They form the base set for a Wilke algebra called the *free Wilke algebra* over Σ, where the product and mixed product are given by concatenation and the ω-star is given by concatenating a word infinitely often with itself. The free Wilke algebra over Σ is denoted by $\mathbb{W}_\Sigma^{+,\mathrm{up}}$, or as $\mathbb{W}^{+,\mathrm{up}}$ if Σ is understood.

2. In a similar way, take Σ^ω, then $(\Sigma^+, \Sigma^\omega)$ also forms a Wilke algebra denoted by $\mathbb{W}_\Sigma^{+,\omega}$ (and $\mathbb{W}^{+,\omega}$ if Σ is understood).

It is not hard to see that $\mathbb{W}^{+,\mathrm{up}}$ is in fact a complete Wilke algebra while $\mathbb{W}^{+,\omega}$ is not.

5.1 Transition Wilke Algebra

We would like to remind the reader of our convention 2.16 that we will freely refer to an arbitrary Ω-coalgebra $\mathbb{A} = (X, Y, \rho, \sigma, \xi, F)$.

In the transition monoid construction at the start of this section, we looked at a DFA (Q, i, δ, F) and the map $\widehat{\delta}^\sharp : \Sigma^* \to Q^Q$ which associates to each $u \in \Sigma$ all the paths with label u. In the case of Wilke algebras and Ω-automata, there are two sorts, a finite and an infinite one. For the finite sort, we would like a map whose domain is Σ^+ and for the infinite sort a map whose domain is Σ^{up}. Naturally, we could follow the transition monoid example and try to associate to each $u \in \Sigma^+$ all the paths of shape $x \xrightarrow{u} x'$. In this case the map $\widehat{\rho}^\sharp : \Sigma^+ \to X^X$ would take the role of $\widehat{\delta}^\sharp$. Similarly, for the infinite case we would like to associate ultimately periodic words vw^ω to paths of the shape $x \xrightarrow{vw^\omega} y$. Unfortunately, such paths do not exist in Ω-automata. Instead of associating some $x \in X$ to some $y \in Y$, it would be more natural to associate x to a subset P of Y, namely to all those y such that there is a path $x \xrightarrow{(v',w')} y$ where (v', w') is a lasso representative of vw^ω (we write $x \overset{vw^\omega}{\Longrightarrow} P$). The following diagram illustrates this idea.

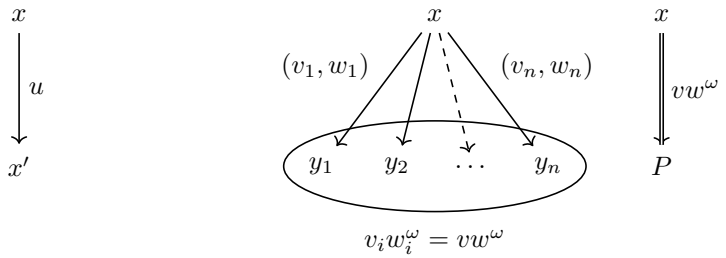


Figure 2: Visualisation of Paths

As a first suggestion, we may hence take as carrier for our transition Wilke algebra the sets

$$T_f = X^X \qquad \text{and} \qquad T_i = (2^Y)^X.$$

Unfortunately, we were not able to equip (T_f, T_i) with the structure of a Wilke algebra. We identify two main reasons for this:

1. the finite sort does not contain enough information to define a map $(-)^\omega : T_f \to T_i$;
2. in T_i we look at functions whose range consists of all possible subsets of Y, but it turns out that we would like to only map spoke states to a special subcollection of subsets of Y, namely those that are 'nice'.

We first address the second problem as, although the problems don't seem very connected, it helps us to solve the first problem. For this, we define a notion of 'niceness' which is that of an admissible set of final states.

Definition 5.5 Let $\mathbb{A} = (X, Y, i, \rho, \sigma, \xi, F)$ be an Ω-coalgebra and $G \subseteq Y$, we write $\mathbb{A}[G/F]$ for the Ω-coalgebra $(X, Y, i, \rho, \sigma, \xi, G)$, i.e. where we replaced the set of final states F by G.

Definition 5.6 Let $\mathbb{A} = (X, Y, \rho, \sigma, \xi, F)$ be an Ω-coalgebra. A subset $G \subseteq Y$ is called an *admissible set of final states* if $\mathbb{A}[G/F]$ is circular and coherent. We denote the set of admissible subsets of Y by $\mathrm{Adm}(Y)$.

We give a reasoning behind why the admissible sets are nice, but before we do that, we have to quickly address another issue. We said that each ultimately periodic word vw^ω should be mapped to a collection of 'paths' of the shape $x \overset{vw^\omega}{\Longrightarrow} P$ as shown in the diagram above, with P being an arbitrary subset of Y. Now we would like to instead map x to an admissible subset of Y. How do we choose which admissible set to map it to?

Ideally, if $x \overset{vw^\omega}{\Longrightarrow} P$, we would like to find some $G \in \mathrm{Adm}(Y)$ such that $P \subseteq G$. A solution to this problem could be, to look at every admissible subset G_i $(i \in I)$ for which $P \subseteq G_i$ and then map x to the collection $\{G_i\}_{i \in I}$. We argue that this collection is indeed a good choice.

Proposition 5.7 *Let $\mathbb{A} = (X, Y, \rho, \sigma, \xi, F)$ be an Ω-coalgebra. The admissible subsets of Y form a Boolean algebra, in particular $Adm(Y)$ is closed under intersection, union and complement.*

Proof. It is easy to check that $\emptyset$ and Y are both admissible. If G is admissible then so is $Y \setminus G$ (the Ω-automaton $\mathbb{A}[G/F]$ is the complement of $\mathbb{A}[(Y \setminus G)/F]$).

We show that $\mathbb{A}[G \cap H/F]$ is circular and coherent.Let $u \in \Sigma^+, v \in \Sigma^*, a \in \Sigma$ and $x \in X$, then

$$\begin{aligned}
\widetilde{\sigma : \xi}(x, u) \in G \cap H &\iff \widetilde{\sigma : \xi}(x, u) \in G \text{ and } \widetilde{\sigma : \xi}(x, u) \in H \\
&\overset{\text{adm}}{\iff} \widetilde{\sigma : \xi}(x, u^k) \in G \text{ and } \widetilde{\sigma : \xi}(x, u^k) \in H && (k > 1) \\
&\iff \widetilde{\sigma : \xi}(x, u^k) \in G \cap H && (k > 1) \\
\widetilde{\sigma : \xi}(x, av) \in G \cap H &\iff \widetilde{\sigma : \xi}(x, av) \in G \text{ and } \widetilde{\sigma : \xi}(x, av) \in H \\
&\overset{\text{adm}}{\iff} \widetilde{\sigma : \xi}(\rho(x, a), ua) \in G \text{ and } \widetilde{\sigma : \xi}(\rho(x, a), ua) \in H \\
&\iff \widetilde{\sigma : \xi}(\rho(x, a), ua) \in G \cap H,
\end{aligned}$$

which shows that it is circular and coherent. As admissible subsets are closed under intersection and complement, they are also closed under union. □

To get back to the matter at hand, we initially considered $x \overset{vw^\omega}{\Longrightarrow} P$ where $P \subseteq Y$ was arbitrary and changed this to $x \overset{vw^\omega}{\Longrightarrow} \{G_i\}_{i\in I}$ where all $G_i \in \mathrm{Adm}(Y)$ with $P \subseteq G_i$. As the admissible subsets form a finite Boolean algebra (for we are mainly interested in finite Ω-automata), it is in particular complete and so the meet of the $\{G_i\}_{i\in I}$ is defined and is the smallest admissible set containing P. Furthermore, the collection $\{G_i\}_{i\in I}$ is upwards-closed meaning that the $\{G_i\}_{i\in I}$ form a principal filter. We can update the T_i to

$$T_i = \left(2^{\mathrm{Adm}(Y)}\right)^X = 2^{X\times \mathrm{Adm}(Y)}.$$

Although the admissible sets of accepting states are well-behaved, there is yet another reason for why we called them 'nice' as the following lemma shows.

Lemma 5.8 *Let $(u,v),(u',v') \in \Sigma^{*+}$ be such that $(u,v) \sim_\gamma (u',v')$. Assume that $x \overset{(u,v)}{\longrightarrow} y$ and $x \overset{(u',v')}{\longrightarrow} y'$, then for any $G \in Adm(Y)$: $y \in G \iff y' \in G$.*

Proof. If $x \overset{(u,v)}{\longrightarrow} y$ and $y \in G$, then $uv^\omega \in L(\mathbb{A}[G/F], x)$. Then as $(u,v) \sim_\gamma (u',v')$, $x \overset{(u',v')}{\longrightarrow} y'$ and as $\mathbb{A}[G/F]$ is circular and coherent, we have that $y' \in G$. The other direction follows by symmetry. □

Another way of expressing the above lemma is that, if $x \overset{vw^\omega}{\Longrightarrow} P$ as in figure 2 and if $G \in \mathrm{Adm}(Y)$, then

$$P \cap G \neq \emptyset \implies P \subseteq G.$$

We slightly adapt our notation and write $x \overset{vw^\omega}{\Longrightarrow} G$ to indicate that, no matter what path with label (v',w') such that $v'w'^\omega = vw^\omega$ we take, we always end up in G. The lemma then states that if we want to find out if for some admissible set G we have $x \overset{vw^\omega}{\Longrightarrow} G$, it is sufficient to take any lasso representative (v_i,w_i) of vw^ω and check if for $x \overset{(v_i,w_i)}{\longrightarrow} y$, we have that $y \in G$. If so, then no matter what lasso representative we choose, we always end up in G and so $x \overset{vw^\omega}{\Longrightarrow} G$.

We promised that this would give a solution to the first problem, i.e. currently our T_f does not contain enough information to define $(-)^\omega : T_f \to T_i$. At the moment, T_f consists of, for each $u \in \Sigma^+$, a collection of paths $x \overset{u}{\to} x' \in \widetilde{\rho}^\sharp(u)$. Given such a path $x \overset{u}{\to} x'$, we essentially want to determine, for a given $G \in \mathrm{Adm}(Y)$, whether $x \overset{u^\omega}{\Longrightarrow} G$. By Lemma 5.8, it would be sufficient to consider any lasso representative (v,w) of u^ω and check if for $x \overset{(v,w)}{\longrightarrow} y$ we have $y \in G$. We know that u^ω has a very nice lasso representative, namely (ϵ,u), so our solution is to enrich our T_f to, for each $u \in \Sigma^+$, not only hold paths of the shape $x \overset{u}{\to} x'$ but also paths of the shape $x \overset{(\epsilon,u)}{\longrightarrow} y$. If we manage to do that, then for each admissible set G, we can easily check if $x \overset{u^\omega}{\Longrightarrow} G$ by checking if $y \in G$.

We update our T_f to

$$T_f = X^X \times Y^X \times Y^Y,$$

where X^X should be thought of as paths of the shape $x \overset{u}{\to} x'$, Y^X as paths of the shape $x \overset{(\epsilon,u)}{\longrightarrow} y$ and Y^Y as paths $y \overset{u}{\to} y'$ which are used to define the product $\circ : T_f \times T_f \to T_i$.

In more detail, the reason why we need Y^Y is that later we use this to compose paths of labels. Having paths in Y then comes in handy and helps us to do just that.

This concludes the presentation of the idea behind the construction. We now turn towards the more technical side, still referring to diagrams along the way.

Definition 5.9 For an Ω-automaton $\mathbb{A} = (X, Y, i, \rho, \sigma, \xi, F)$ we define

$$\mathbb{A}_f = X^X \times Y^X \times Y^Y \qquad \text{and} \qquad \mathbb{A}_i = 2^{X \times \mathrm{Adm}(Y)}.$$

We start by formally defining a map $\mu_{\mathbb{A}_f} : \Sigma \to \mathbb{A}_f$, which to each $a \in \Sigma$ associates all the paths of the shape $z \xrightarrow{a} z'$. By equipping $\mathbb{A}_f$ with the structure of a semigroup, we can automatically extend $\mu_{\mathbb{A}_f}$ to a semigroup homomorphism from the free semigroup over Σ. We remind the reader of the following definition.

Definition 5.10 For a map $\tau : Z_1 \to Z_2^{Z_3}$, we define the map $\tau^\sharp : Z_3 \to Z_2^{Z_1}$ as

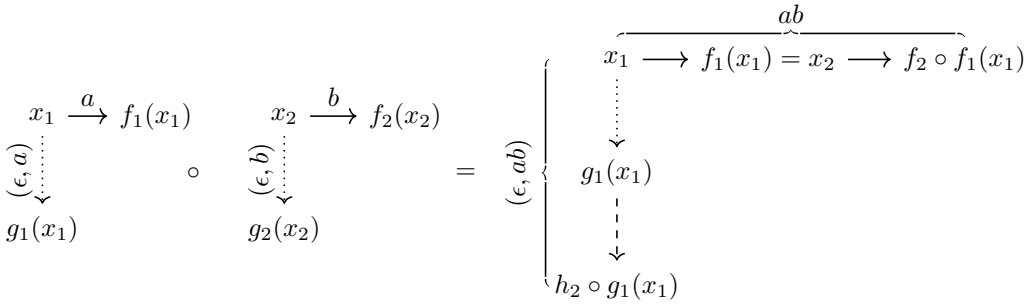
$$\tau^\sharp(z_3)(z_1) = \tau(z_1)(z_3).$$

Definition 5.11 Given an Ω-automaton $\mathbb{A} = (X, Y, i, \rho, \sigma, \xi, F)$, we define the map

$$\begin{aligned} \mu_{\mathbb{A}_f} : \Sigma &\longrightarrow \mathbb{A}_f \\ a &\longmapsto \mu_{\mathbb{A}_f}(a) = (\rho^\sharp(a), \sigma^\sharp(a), \xi^\sharp(a)) \end{aligned}$$

and we write μ_f if $\mathbb{A}$ is understood.

To illustrate how the elements of $\mathbb{A}_f$ can be composed (other than looking at the types), we draw a diagram. Let $\mu_f(a) = (f_1, g_1, h_1)$ and $\mu_f(b) = (f_2, g_2, h_2)$ and consider two states $x_1, x_2 \in X$ such that $f_1(x_1) = x_2$, then we want



Definition 5.12 Let $\circ : \mathbb{A}_f \times \mathbb{A}_f \to \mathbb{A}_f$ be given by

$$(f_1, g_1, h_1) \circ (f_2, g_2, h_2) = (f_2 \circ f_1, h_2 \circ g_1, h_2 \circ h_1).$$

Remark 5.13 One interesting feature of $\circ$, is that in the end result the g_2 disappears. This should not come as a surprise, however, as we stated already in the ideas behind the construction that we need some additional maps in order to make our construction easier.

Proposition 5.14 *The pair $(\mathbb{A}_f, \circ)$ forms a semigroup.*

Proof. Let $(f_1, g_1, h_1), (f_2, g_2, h_2), (f_3, g_3, h_3) \in \mathbb{A}_f$, then

$$\begin{aligned}(f_1, g_1, h_1) \circ ((f_2, g_2, h_2) \circ (f_3, g_3, h_3)) &= (f_1, g_1, h_1) \circ (f_3 \circ f_2, h_3 \circ g_2, h_3 \circ h_2) \\ &= ((f_3 \circ f_2) \circ f_1, (h_3 \circ h_2) \circ g_1, (h_3 \circ h_2) \circ h_1) \\ &= (f_3 \circ (f_2 \circ f_1), h_3 \circ (h_2 \circ g_1), h_3 \circ (h_2 \circ h_1)) \\ &= (f_2 \circ f_1, h_2 \circ g_1, h_2 \circ h_1) \circ (f_3, g_3, h_3) \\ &= ((f_1, g_1, h_1) \circ (f_2, g_2, h_2)) \circ (f_3, g_3, h_3),\end{aligned}$$

as required. □

Remark 5.15 One might wonder if $(\mathbb{A}_f, \circ)$ also forms a monoid. This is in fact not the case, because the switch transition cannot be defined for the empty word. So the second component disallows the existence of an identity element (see Definition 5.11).

Proposition 5.16 *The map $\mu_f : \Sigma \to \mathbb{A}_f$ can uniquely be extended to a semigroup morphism $\mu_f : (\Sigma^+, \cdot) \to (\mathbb{A}_f, \circ)$, which is given by*

$$\mu_f(u) = (\widetilde{\rho}^\sharp(u), \widetilde{\sigma : \xi}^\sharp(u), \widetilde{\xi}^\sharp(u)),$$

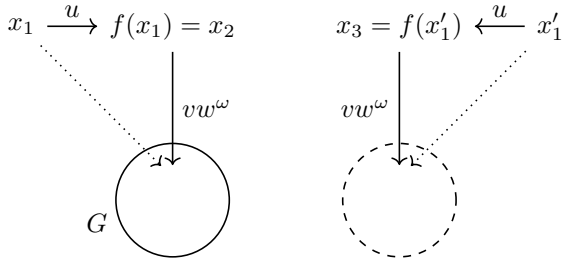
for $u \in \Sigma^+$.

Proof. The unique extension of μ_f is obtained by the freeness of the semigroup over Σ. Showing that μ_f has the given shape can be done by induction on $u \in \Sigma^+$. We already have the base case where $a \in \Sigma$. For the induction case, let $ua \in \Sigma^+$, then

$$\begin{aligned}\mu_f(ua) &= \mu_f(u) \circ \mu_f(a) \\ &= (\widetilde{\rho}^\sharp(u), \widetilde{\sigma : \xi}^\sharp(u), \widetilde{\xi}^\sharp(u)) \circ (\rho^\sharp(a), \sigma^\sharp(a), \xi^\sharp(a)) \\ &= (\rho^\sharp(a) \circ \widetilde{\rho}^\sharp(u), \xi^\sharp(a) \circ \widetilde{\sigma : \xi}^\sharp(u), \xi^\sharp(a) \circ \widetilde{\xi}^\sharp(u)) \\ &= (\widetilde{\rho}^\sharp(ua), \widetilde{\sigma : \xi}^\sharp(ua), \widetilde{\xi}^\sharp(ua)).\end{aligned}$$

□

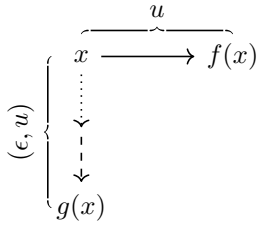
Before we formally define the mixed product and the ω-star, we provide diagrams which serve as a reference point. Let $\mu_f(u) = (f, g, h)$ and some $p \in A_i$ corresponding to an ultimately periodic word vw^ω. Consider $x_1, x_1', x_2, x_3 \in X$ and $G \in \mathrm{Adm}(Y)$, such that $f(x_1) = x_2$, $f(x_1') = x_3$, $p(x_2, G) = 1$ and $p(x_3, G) = 0$. So there is a path from x_2 to G with label (v, w), but there is no such path for x_3. Then in $(f, g, h) \otimes p$, x_1 has a path to G whereas x_1' does not.



This can also be expressed as follows:

$$((f,g,h)\otimes p)\ (x,G)=1 \iff p(f(x),G)=1.$$

For the ω-star operation, we remind ourselves that for $\mu_f(u)=(f,g,h)$ and some $x\in X$, we have:



In order to define $(f,g,h)^\omega$, we have to know for each $x\in X$ and $G\in \mathrm{Adm}(Y)$ whether $x \overset{u^\omega}{\Longrightarrow} G$. By Lemma 5.8, it is hence sufficient to determine whether $g(x)\in G$ (as $x \overset{(\epsilon,u)}{\longrightarrow} g(x)$ and (ϵ,u) is a lasso representative of u^ω).

Definition 5.17 Given an Ω-automaton $\mathbb{A}=(X,Y,i,\rho,\sigma,\xi,F)$, we define the following two operations:

$$\begin{aligned}&\otimes : \mathbb{A}_f\times\mathbb{A}_i \longrightarrow \mathbb{A}_i\\ &((f,g,h),p)\longmapsto (f,g,h)\otimes p=\lambda xG.p(f(x),G),\end{aligned}$$

$$\begin{aligned}&(-)^\omega : \mathbb{A}_f \longrightarrow \mathbb{A}_i\\ &\quad (f,g,h)\longmapsto (f,g,h)^\omega=\lambda xG.[g(x)\in G].\end{aligned}$$

Definition 5.18 For an Ω-automaton $\mathbb{A}=(X,Y,i,\rho,\sigma,\xi,F)$ we define the two sets:

$$\begin{aligned}T_f(\mathbb{A})&=\mu_f[\Sigma^+],\\ T_i(\mathbb{A})&=\{(f,g,h)\otimes(f',g',h')^\omega \mid (f,g,h),(f',g',h')\in T_f(\mathbb{A})\}.\end{aligned}$$

Theorem 5.19 *Given an Ω-automaton $\mathbb{A}=(X,Y,i,\rho,\sigma,\xi,F)$,*

$$\mathbb{T}(\mathbb{A})=(T_f(\mathbb{A}),T_i(\mathbb{A}),\circ,\otimes,^\omega)$$

is a complete Wilke algebra, called the transition Wilke algebra of $\mathbb{A}$.

Proof. If $\mathbb{T}(\mathbb{A})$ is a Wilke algebra, then clearly it is complete by our definition of $T_i(\mathbb{A})$. To show that it is a Wilke algebra, we show that each of the requirements in Definition 5.2 holds.

Associativity (A): We have shown this in 5.14.

Mixed Associativity (MA): We have to show that, for $(f_1,g_1,h_1),(f_2,g_2,h_2)\in T_f(\mathbb{A}), p\in T_i(\mathbb{A})$:

$$((f_1,g_1,h_1)\circ(f_2,g_2,h_2))\otimes p=(f_1,g_1,h_1)\otimes((f_2,g_2,h_2)\otimes p)$$

Computing the left- and right-hand side separately, we find that for all $x \in X$ and $G \in \mathrm{Adm}(Y)$:

$$\begin{aligned}
(((f_1,g_1,h_1)\circ(f_2,g_2,h_2))\otimes p)\ (x,G) &= ((f_2\circ f_1, h_2\circ g_1, h_2\circ h_1)\otimes p)\ (x,G)\\
&= p((f_2\circ f_1)(x),G)\\
((f_1,g_1,h_1)\otimes((f_2,g_2,h_2)\otimes p))\ (x,G) &= ((f_1,g_1,h_1)\otimes(\lambda x'G'.p(f_2(x'),G')))\ (x,G)\\
&= (\lambda x'G'.p(f_2(x'),G'))\ (f_1(x),G)\\
&= p(f_2(f_1(x)),G)\\
&= p((f_2\circ f_1)(x),G).
\end{aligned}$$

Coherence (Co): Given $(f_1,g_1,h_1),(f_2,g_2,h_2)\in T_f(\mathbb{A})$ we have to show that

$$(f_1,g_1,h_1)\otimes((f_2,g_2,h_2)\circ(f_1,g_1,h_1))^\omega = ((f_1,g_1,h_1)\circ(f_2,g_2,h_2))^\omega$$

We again compute both sides separately, for all $x\in X$ and $G\in\mathrm{Adm}(Y)$:

$$\begin{aligned}
((f_1,g_1,h_1)\otimes((f_2,g_2,h_2)\circ(f_1,g_1,h_1))^\omega)\ (x,G)&\\
&= ((f_1,g_1,h_1)\otimes(f_1\circ f_2, h_1\circ g_2, h_1\circ h_2)^\omega)\ (x,G)\\
&= ((f_1,g_1,h_1)\otimes(\lambda x'G'.[(h_1\circ g_2)(x')\in G']))\ (x,G)\\
&= (\lambda x'G'.[(h_1\circ g_2)(x')\in G'])\ (f_1(x),G)\\
&= [(h_1\circ g_2\circ f_1)(x)\in G]\\
((f_1,g_1,h_1)\circ(f_2,g_2,h_2))^\omega\ (x,G) &= (f_2\circ f_1, h_2\circ g_1, h_2\circ h_1)^\omega\ (x,G)\\
&= [(h_2\circ g_1)(x)\in G].
\end{aligned}$$

It remains to show that for all $x\in X$:

$$(h_2\circ g_1)(x)\in G \iff (h_1\circ g_2\circ f_1)(x)\in G.$$

Without loss of generality, let $u,v\in\Sigma^+$ be such that

$$f_1=\widetilde{\rho}^\sharp(u) \qquad g_1=\widetilde{\sigma:\xi}^\sharp(u) \qquad h_1=\widetilde{\xi}^\sharp(u)$$
$$g_2=\widetilde{\sigma:\xi}^\sharp(v) \qquad h_2=\widetilde{\xi}^\sharp(v)$$

Then the above bi-implication rewrites to

$$\widetilde{\xi}(\widetilde{\sigma:\xi}(x,u),v)\in G \iff \widetilde{\xi}(\widetilde{\sigma:\xi}(\widetilde{\rho}(x,u),v),u)\in G,$$

which is equivalent to the coherence condition

$$(\epsilon,uv)\in\mathrm{Lassos}(x) \iff (u,vu)\in\mathrm{Lassos}(x)$$

for the Ω-automaton $\mathbb{A}[G/F]$ and as G is admissible, we are done.

Circularity (Ci): For $(f,g,h)\in T_f(\mathbb{A})$ and $k>1$ we have to show that

$$(f,g,h)^\omega = ((f,g,h)^k)^\omega.$$

First, it is easy to see that

$$(f,g,h)^k = (f^k, h^{k-1} \circ g, h^k).$$

So we get for the two sides that, for each $x \in X$ and $G \in \mathrm{Adm}(Y)$:

$$\begin{aligned}(f,g,h)^\omega\ (x,G) &= [g(x) \in G]\\ ((f,g,h)^k)^\omega &= [(h^{k-1} \circ g)(x) \in G]\end{aligned}$$

and we have to show that

$$h^{k-1}(g(x)) \in G \iff g(x) \in G.$$

Without loss of generality let $u \in \Sigma^+$ be such that $g = \widetilde{\sigma : \xi}^\sharp(u), h = \widetilde{\xi}^\sharp(u)$ which yields

$$\widetilde{\sigma : \xi}(x, uu^{k-1}) \in G \iff \widetilde{\sigma : \xi}(x,u) \in G.$$

This is equivalent to the circularity condition

$$u^k \in \mathrm{Loop}(x) \iff u \in \mathrm{Loop}(x)$$

for the Ω-automaton $\mathbb{A}[G/F]$ and holds as G is admissible. □

5.2 Recognition of Lasso Languages

In this final subsection, we first show that the ultimately periodic fragment of any ω-regular language accepted by some Ω-automaton $\mathbb{A}$, is also recognised by a Wilke algebra homomorphism from the free Wilke algebra onto a (finite) Wilke algebra, and more specifically onto the transition Wilke algebra $\mathbb{T}(\mathbb{A})$. We then use some results from [33] to extend the Wilke algebra morphism and conclude that the ω-regular language is recognised by a Wilke algebra homomorphism from $\mathbb{W}^{+,\omega}$.

Proposition 5.20 *The map $\mu(\mathbb{A}) = (\mu_f, \mu_i) : \mathbb{W}^{+,up} \to \mathbb{T}(\mathbb{A})$ where μ_f is as in Proposition 5.16 and μ_i is given by*

$$\mu_i(uv^\omega) = \lambda x G.[(u,v) \in \mathit{Lassos}(\mathbb{A}[G/F], x)]$$

is a Wilke algebra homomorphism.

Proof. First, we argue that μ_i is well-defined. This follows from the fact that we only consider $G \in \mathrm{Adm}(Y)$, so if $u'v'^\omega = uv^\omega$, then for all x we have

$$(u',v') \in \mathrm{Lassos}(\mathbb{A}[G/F], x) \iff (u,v) \in \mathrm{Lassos}(\mathbb{A}[G/F], x).$$

Next we show that the three conditions from Definition 5.3 hold. The first condition

$$\mu_f(uv) = \mu_f(u) \circ \mu_f(v)$$

holds by 5.16.

To show the second condition let $u \in \Sigma^+$ and $vw^\omega \in \Sigma^{\mathrm{up}}$, then for all $x \in X$ and $G \in \mathrm{Adm}(Y)$ we have

$$\begin{aligned}
&\mu_f(u) \otimes \mu_i(vw^\omega)\ (x, G) \\
&= (\widetilde{\rho}^\sharp(u), \widetilde{\sigma : \xi}^\sharp(u), \widetilde{\xi}^\sharp(u)) \otimes (\lambda x' G'.[(v,w) \in \mathrm{Lassos}(\mathbb{A}[G'/F], x')])\ (x, G) \\
&= (\lambda x' G'.[(v,w) \in \mathrm{Lassos}(\mathbb{A}[G'/F], x')])\ (\widetilde{\rho}^\sharp(u)(x), G) \\
&= [(v,w) \in \mathrm{Lassos}(\mathbb{A}[G/F], \widetilde{\rho}^\sharp(u)(x))] \\
&= [(uv,w) \in \mathrm{Lassos}(\mathbb{A}[G/F], x)] \\
&= \mu_i(uvw^\omega)\ (x, G),
\end{aligned}$$

so $\mu_i(uvw^\omega) = \mu_f(u) \otimes \mu_i(vw^\omega)$. There is one more equation to check. For the third condition let $u \in \Sigma^+$, then for all $x \in X$ and $G \in \mathrm{Adm}(Y)$, we have

$$\begin{aligned}
\mu_i(u^\omega)\ (x, G) &= [(\epsilon, u) \in \mathrm{Lassos}(\mathbb{A}[G/F], x)] \\
&= [\widetilde{\sigma : \xi}^\sharp(u)(x) \in G] \\
&= (\widetilde{\rho}^\sharp(u), \widetilde{\sigma : \xi}^\sharp(u), \widetilde{\xi}^\sharp(u))^\omega\ (x, G) \\
&= \mu_f(u)^\omega\ (x, G).
\end{aligned}$$

Thus $\mu_i(u^\omega) = \mu_f(u)^\omega$ and (μ_f, μ_i) is a Wilke algebra homomorphism. □

Definition 5.21 ([33]) A Wilke algebra homomorphism $(f_1, f_2) : (A_f, A_i) \to (B_f, B_i)$ *recognises* $L \subseteq A_i$ if there exists $P \subseteq B_i$ such that

$$L = f_2^{-1}[P].$$

Theorem 5.22 ([33]) *Let $\mathscr{L} \subseteq \Sigma^\omega$ be an ω-language, then the following are equivalent:*

1. *$\mathscr{L}$ is ω-regular.*
2. *UP($\mathscr{L}$) is recognised by a Wilke algebra homomorphism from the free Wilke algebra $\mathbb{W}^{+,up}$ into a finite Wilke algebra.*
3. *$\mathscr{L}$ is recognised by a Wilke algebra homomorphism $f : \mathbb{W}^{+,\omega} \to \mathbb{S}$ (where $\mathbb{S}$ is finite) having the property that*
$$f(u_0)^\omega = f(u_0 u_1 u_2 \ldots)$$
for every infinite sequence $U = \{u_0, u_1, \ldots\}$ of finite words with $f(u_0) = f(u_1) = \ldots$

Theorem 5.23 *Let $\mathbb{A}$ be an Ω-automaton and $\mathscr{L} = L(\mathbb{A})$, then $\mu(\mathbb{A})$ recognises UP($\mathscr{L}$).*

Proof. Let

$$P = \{p \in \mathbb{T}_i(\mathbb{A}) \mid p(i, F) = 1\}.$$

We claim that

$$\mathrm{UP}(\mathscr{L}) = \mu_i^{-1}[P].$$

Let $uv^\omega \in \mathrm{UP}(\mathscr{L})$, so

$$\mu_i(uv^\omega)\ (i, F) = [(u,v) \in \mathrm{Lassos}(\mathbb{A}, i)] = 1.$$

Hence $\mu_i(uv^\omega) \in P$ and $uv^\omega \in \mu_i^{-1}[P]$. For the other inclusion, let $uv^\omega \in \mu_i^{-1}[P]$, then $\mu_i(uv^\omega) \in P$, so $\mu_i(uv^\omega)\ (i, F) = 1$. From this it follows that $(u, v) \in \text{Lassos}(\mathbb{A}, i)$ thus $uv^\omega \in \text{UP}(\mathscr{L})$. □

Corollary 5.24 *Let $\mathscr{L}$ be accepted by an Ω-automaton $\mathbb{A}$, then it is recognised by a Wilke algebra homomorphism from $\mathbb{W}^{+,\omega}$ to $\mathbb{T}(\mathbb{A})$.*

Proof. This is a direct consequence of 5.22 (for more details see [33, Corollary 2.9, Theorem 1.2]). □

Before concluding this section, we would like to touch upon the converse problem, i.e. given a Wilke algebra homomorphism from $\mathbb{W}^{+,\omega}$ to a finite Wilke algebra, and an ω-regular language $\mathscr{L}$ recognised by said morphism, can we construct an Ω-automaton which also accepts $\mathscr{L}$? This direction is discussed extensively for different types of ω-automata in [25, 10] and seems to be a difficult problem. For instance, constructing a Muller automaton requires some non-trivial results from semigroup theory. Although this topic is not investigated in this thesis, we believe it would be a worthwhile topic for further research.

Myhill-Nerode Theorem

The Myhill-Nerode theorem ([27]) is a very well-known result which characterises the regularity of languages and gives a strict lower bound on the size of deterministic finite automata for a given language.

The paper [12] provides a Myhill-Nerode theorem for Ω-automata which characterises the ω-regular languages as those for which the Myhill-Nerode equivalence relation is of finite index and which are lasso-determined. However, it does not give a tight lower bound on the size of Ω-automata.

This section provides a slightly different equivalence relation under which we obtain a strengthening of the original result which also gives a tight lower bound on the size of Ω-automata accepting a given ω-regular language.

Definition 6.1 Given a lasso language L, we define the two binary relations $\sim_L^0 \subseteq \Sigma^* \times \Sigma^*$ and $\sim_L^1 \subseteq \Sigma^{*+} \times \Sigma^{*+}$ as

$$u_1 \sim_L^0 u_2 \iff \forall (v,w) \in \Sigma^{*+} : \big((u_1 v, w) \in L \iff (u_2 v, w)^\omega \in L\big),$$
$$(u_1, v_1) \sim_L^1 (u_2, v_2) \iff \forall w \in \Sigma^{+} : \big((u_1, v_1 w) \in L \iff (u_2, v_2 w) \in L\big).$$

Proposition 6.2 *Let L be a lasso language, then $\sim_L^0$ and $\sim_L^1$ are both equivalence relations.*

Proof. Reflexivity, symmetry and transitivity follows by the reflexivity, symmetry and transitivity of the biconditional. □

Definition 6.3 For an Ω-automaton $\mathbb{A} = (X, Y, i, \rho, \sigma, \xi, F)$, we define two binary relations $\sim_{\mathbb{A}}^0 \subseteq \Sigma^* \times \Sigma^*$ and $\sim_{\mathbb{A}}^1 \subseteq \Sigma^{*+} \times \Sigma^{*+}$ as

$$u_1 \sim_{\mathbb{A}}^0 u_2 \iff \widehat{\rho}(i, u_1) = \widehat{\rho}(i, u_2),$$
$$(u_1, v_1) \sim_{\mathbb{A}}^1 (u_2, v_2) \iff \widetilde{\sigma : \xi}(\widehat{\rho}(i, u_1), v_1) = \widetilde{\sigma : \xi}(\widehat{\rho}(i, u_2), v_2).$$

Proposition 6.4 *For an Ω-automaton $\mathbb{A}$, the binary relations $\sim_{\mathbb{A}}^0$ and $\sim_{\mathbb{A}}^1$ are both equivalence relations.*

Proof. Reflexivity, symmetry and transitivity follow immediately by the reflexivity, symmetry and transitivity of equality. □

Lemma 6.5 *For any regular lasso language L and any Ω-automaton $\mathbb{A}$ accepting L, the binary relations $\sim_L^i$ are always at least as coarse as the binary relations $\sim_{\mathbb{A}}^i$, i.e.*

$$\sim_{\mathbb{A}}^0 \subseteq \sim_L^0 \qquad \text{and} \qquad \sim_{\mathbb{A}}^1 \subseteq \sim_L^1 .$$

Proof. We start with $\sim_{\mathbb{A}}^0$ and $\sim_L^0$. Let $u_1, u_2 \in \Sigma^*$ be such that $u_1 \sim_{\mathbb{A}}^0 u_2$, so

$$\widehat{\rho}(i, u_1) = \widehat{\rho}(i, u_2).$$

Then for all $(v, w) \in \Sigma^{*+}$ we have that

$$\widetilde{\sigma : \xi}(\widehat{\rho}(i, u_1 v), w) = \widetilde{\sigma : \xi}(\widehat{\rho}(i, u_2 v), w),$$

(as our automaton is deterministic) and so

$$(u_1 v, w) \in L \iff (u_2 v, w) \in L$$

as $\mathbb{A}$ is circular and coherent. This gives us $u_1 \sim_L^0 u_2$ as required.

Now we proceed with $\sim_{\mathbb{A}}^1$ and $\sim_L^1$. Let $(u_1, v_1) \sim_{\mathbb{A}}^1 (u_2, v_2)$, so

$$\widetilde{\sigma : \xi}(\widehat{\rho}(i, u_1), v_1) = \widetilde{\sigma : \xi}(\widehat{\rho}(i, u_2), v_2).$$

Then for all $w \in \Sigma^*$ we have

$$\widetilde{\sigma : \xi}(\widehat{\rho}(i, u_1), v_1 w) = \widetilde{\sigma : \xi}(\widehat{\rho}(i, u_2), v_2 w),$$

and again we have

$$(u_1, v_1 w) \in L \iff (u_2, v_2 w) \in L,$$

as $\mathbb{A}$ is circular and coherent. Hence $(u_1, v_1) \sim_L^1 (u_2, v_2)$. □

Corollary 6.6 *If L is a regular lasso language, then both the equivalence classes of $\sim_L^0$ and $\sim_L^1$ are of finite index.*

Proof. As L is regular, there is some finite Ω-automaton $\mathbb{A}$ which accepts L. We may, w.l.o.g. assume that all states of $\mathbb{A}$ are reachable. By 6.5 it is sufficient to show that the equivalence classes for $\sim_{\mathbb{A}}^i$ are of finite index. This is clearly the case as the number of equivalence classes of $\sim_{\mathbb{A}}^0$ is bounded by the number of spoke states of $\mathbb{A}$. To see this, note that there is a one-to-one correspondence between the equivalence classes of $\sim_{\mathbb{A}}^0$ and the spoke states, given by

$$\begin{aligned} [u]_{\sim_{\mathbb{A}}^0} &\mapsto \widehat{\rho}(i, u) \\ x &\mapsto \{u \mid \widehat{\rho}(i, u) = x\}. \end{aligned}$$

Via an analogous argument, it can be shown that the number of equivalence classes of $\sim_{\mathbb{A}}^1$ is bounded by the number of loop states. □

Definition 6.7 ([12]) An ω-language $\mathscr{L}$ is *lasso-determined* if for every infinite sequence $\{v_i\}_{i \in \omega}$ of nonemtpy words there is an infinite set $Y \subseteq \omega$ such that

$$v_0 v_1 v_2 \ldots \in \mathscr{L} \iff (v_0 \ldots v_j)(v_{j+1} \ldots v_k)^\omega \in \mathscr{L},$$

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

Proposition 6.8 ([12]) *Let $\mathscr{L}, \mathscr{K}$ be two ω-languages which are lasso-determined. Then*

$$\{(u, v) \mid uv^\omega \in \mathscr{L}\} = \{(u, v) \mid uv^\omega \in \mathscr{K}\} \implies \mathscr{L} = \mathscr{K}.$$

Definition 6.9 Given an ω-language $\mathscr{L}$, we define the two binary relations $\sim^0_{\mathscr{L}} \subseteq \Sigma^* \times \Sigma^*$ and $\sim^1_{\mathscr{L}} \subseteq \Sigma^{*+} \times \Sigma^{*+}$ as

$$u_1 \sim^0_{\mathscr{L}} u_2 \iff \forall (v,w) \in \Sigma^{*+} : \big(u_1 v w^\omega \in \mathscr{L} \iff u_2 v w^\omega \in \mathscr{L}\big),$$
$$(u_1, v_1) \sim^1_{\mathscr{L}} (u_2, v_2) \iff \forall w \in \Sigma^{+} : \big(u_1 (v_1 w)^\omega \in \mathscr{L} \iff u_2 (v_2 w)^\omega \in \mathscr{L}\big).$$

Proposition 6.10 *Let $\mathscr{L}$ be an ω-language, then $\sim^0_{\mathscr{L}}$ and $\sim^1_{\mathscr{L}}$ are both equivalence relations.*

Proof. Reflexivity, symmetry and transitivity follows by the reflexivity, symmetry and transitivity of the biconditional. □

Corollary 6.11 *For any ω-regular language $\mathscr{L}$ and any Ω-automaton $\mathbb{A}$ accepting $\mathscr{L}$, the binary relations $\sim^i_{\mathscr{L}}$ are always at least as coarse as the binary relations $\sim^i_{\mathbb{A}}$, i.e.*

$$\sim^0_{\mathbb{A}} \subseteq \sim^0_{\mathscr{L}} \qquad \textit{and} \qquad \sim^1_{\mathbb{A}} \subseteq \sim^1_{\mathscr{L}}.$$

Proof. Given an ω-regular language $\mathscr{L}$ and considering

$$L = \{(u,v) \mid uv^\omega \in \mathscr{L}\},$$

it is not hard to see that, in fact, $\sim^0_L = \sim^0_{\mathscr{L}}$ and $\sim^1_L = \sim^1_{\mathscr{L}}$. Furthermore, any Ω-automaton accepting $\mathscr{L}$ accepts L. The result then follows by Lemma 6.5. □

Corollary 6.12 *If $\mathscr{L}$ is ω-regular, then both the equivalence classes of $\sim^0_{\mathscr{L}}$ and $\sim^1_{\mathscr{L}}$ are of finite index.*

Proof. This follows in a similar fashion as Corollary 6.6 using Corollary 6.11. □

Theorem 6.13 *An ω-language $\mathscr{L}$ is regular if and only if it is lasso determined and both $\sim^0_{\mathscr{L}}, \sim^1_{\mathscr{L}}$ are of finite index. Moreover, the size of the smallest Ω-automaton accepting $\mathscr{L}$ is the number of equivalence classes generated by $\sim^0_{\mathscr{L}}$ and $\sim^1_{\mathscr{L}}$.*

Proof. With regards to ω-regularity, Corollary 6.12 provides the direction from left to right. For the other direction, we use the equivalence classes generated by $\sim^0_{\mathscr{L}}$ and $\sim^1_{\mathscr{L}}$ to construct a finite Ω-automaton accepting $\mathscr{L}$ by which it follows that $\mathscr{L}$ is ω-regular.

Let

$$X = {\Sigma^*}/{\sim^0_{\mathscr{L}}} \qquad \text{and} \qquad Y = {\Sigma^{*+}}/{\sim^1_{\mathscr{L}}}.$$

The initial state i is given by the equivalence class $[\epsilon]_{\sim^0_{\mathscr{L}}}$ and the final states are given by

$$F = \{[(u,v)]_{\sim^1_{\mathscr{L}}} \mid uv^\omega \in \mathscr{L}\}.$$

The transitions are given by

$$\rho([u]_{\sim^0_{\mathscr{L}}}, a) = [ua]_{\sim^0_{\mathscr{L}}},$$
$$\sigma([u]_{\sim^0_{\mathscr{L}}}, a) = [(u,a)]_{\sim^1_{\mathscr{L}}},$$
$$\xi([(u,v)]_{\sim^1_{\mathscr{L}}}, a) = [(u,va)]_{\sim^1_{\mathscr{L}}}.$$

We have to show that this is well-defined. We only show this for the switch transition, the proof for the other transitions is analogous. Let $u_1 \sim^0_{\mathscr{L}} u_2$, then for all $w \in \Sigma^+$ we have in particular that

$$u_1(aw)^\omega \in \mathscr{L} \iff u_2(aw)^\omega \in \mathscr{L},$$

and so $(u_1, a) \sim^1_{\mathscr{L}} (u_2, a)$ as required.

In order to show that $(X, Y, i, \rho, \sigma, \xi, F)$ is indeed an Ω-automaton, we have to show that it is circular and coherent. For circularity, we want that for all $[u]_{\sim^0_{\mathscr{L}}} \in X$, $k > 0$ and $v \in \Sigma^+$:

$$v \in \mathrm{Loop}([u]_{\sim^0_{\mathscr{L}}}) \iff v^k \in \mathrm{Loop}([u]_{\sim^0_{\mathscr{L}}}).$$

As $[u]_{\sim^0_{\mathscr{L}}}$ is reached from the initial state after reading u, we have that

$$\begin{aligned}
v^k \in \mathrm{Loop}([u]_{\sim^0_{\mathscr{L}}}) &\iff (\epsilon, v^k) \in \mathrm{Lassos}([u]_{\sim^0_{\mathscr{L}}}) \\
&\iff (u, v^k) \in \mathrm{Lassos}([\epsilon]_{\sim^0_{\mathscr{L}}}) \\
&\iff \widetilde{\sigma : \xi}(\widehat{\rho}([\epsilon]_{\sim^0_{\mathscr{L}}}, u), v^k) \in F \\
&\iff \widetilde{\sigma : \xi}([u]_{\sim^0_{\mathscr{L}}}, v^k) \in F \\
&\iff [(u, v^k)]_{\sim^0_{\mathscr{L}}} \in F \\
&\iff u(v^k)^\omega \in \mathscr{L},
\end{aligned}$$

and so

$$v \in \mathrm{Loop}([u]_{\sim^0_{\mathscr{L}}}) \iff uv^\omega \in \mathscr{L} \iff u(v^k)^\omega \in \mathscr{L} \iff v^k \in \mathrm{Loop}([u]_{\sim^0_{\mathscr{L}}}).$$

For coherence, the criterion is

$$av \in \mathrm{Loop}([u]_{\sim^0_{\mathscr{L}}}) \iff va \in \mathrm{Loop}(\rho([u]_{\sim^0_{\mathscr{L}}}, a)),$$

which via similar reasoning as above is equivalent to

$$u(av)^\omega \in \mathscr{L} \iff ua(va)^\omega \in \mathscr{L}$$

and again holds trivially.

We have to show that the Ω-automaton we obtained accepts $\mathscr{L}$. For an arbitrary lasso $(u, v) \in \Sigma^{*+}$ we have

$$\begin{aligned}
(u, v) \in \mathrm{Lassos}([\epsilon]_{\sim^0_{\mathscr{L}}}) &\iff \widetilde{\sigma : \xi}(\widehat{\rho}([\epsilon]_{\sim^0_{\mathscr{L}}}, u), v) \in F \\
&\iff [u, v]_{\sim^1_{\mathscr{L}}} \in F \\
&\iff uv^\omega \in \mathscr{L}.
\end{aligned}$$

Finally, the result holds as $\mathscr{L}$ is lasso determined.

As a final remark, we want to point out that this Ω-automaton has to be minimal by Corollary 6.12. $\square$

Construction of Ω-automata from ω-regular Expressions

Ω-automata are very well-behaved in many ways: they are deterministic, their acceptance condition is local and they allow for minimisation. We investigate minimisation in the last section. However, they also have some shortcomings. They are generally of big size (number of states) and up to now we are missing a direct construction method which would allow us to obtain an Ω-automaton from an ω-regular expression. This section deals with the problem of Ω-automaton construction.

In [11, 12], the authors provide a translation from parity and Muller automata to Ω-automata. This means that, given an ω-regular expression, one would first have to construct either a parity or Muller automaton and then proceed to translate it to an Ω-automaton via the constructions provided. This method raises two issues,

1. the construction from parity and Muller automata to Ω-automata is not very efficient, the resulting Ω-automaton is exponential in size of the starting automaton;

2. we first have to come up with a parity or Muller automaton for our ω-regular expression.

For comparison, in the case of deterministic finite automata, the Brzozowski construction ([8]) provides an elegant solution. Given a regular expression e, it allows for the construction of a DFA on the fly which accepts the regular language given by e (denoted by $[\![e]\!]$). Unfortunately, the classical construction does not work for ω-regular expressions. In 2015, Thiemann et al. ([30]) generalised the construction by using a variant of the partial derivative with which they were able to construct a non-deterministic Büchi automaton from an ω-regular expression. Previous papers attempting to construct ω-automata from ω-regular expressions focused on deterministic ω-automata with different acceptance conditions ([26]).

In what follows, we first guide the reader through the classical construction. This includes a journey through regular expressions and the introduction of some technical definitions needed to define the Brzozowski automaton. In short, the first subsection introduces only background material which has been well established.

The second subsection first covers ω-regular expressions as they can be found in the literature. After that, we describe our course of action and develop the necessary tools, some of it closely following the classical definitions and construction. Most of our construction is Brzozowski-esque apart from the switch transition. We outline the problems that arise from the switch transition, and provide a necessary and sufficient criterion under which we can construct a Brzozowski Ω-automaton.

The last section sheds some more light into the switch transition and provides a solution for the problems associated with it. This is done by introducing *lasso expressions*.

7.1 Regular Expressions and the Brzozowski Construction

The purpose of this first subsection is entirely to recapitulate the notions of regular expressions, Kleene algebras and the Brzozowski construction for deterministic finite automata.

More precisely, we cover the syntax and semantics of regular expressions and introduce all the concepts needed for the Brzozowski construction which allows us to construct a DFA from a regular expression. The material introduced throughout this whole subsection can be found in [8, 19, 27, 20] and is well established.

Let us remind ourselves of some operations which we can define on languages and ω-languages. These definitions are standard and can be found in any textbook on regular languages.

Definition 7.1 For $L, K \subseteq \Sigma^*$, $J \subseteq \Sigma^+$ and $\mathscr{L} \subseteq \Sigma^\omega$, we define

$$K \cdot L = \{u \cdot v \mid u \in K, v \in L\} \qquad L^* = \bigcup_{k \in \mathbb{N}} L^k$$
$$K \cdot \mathscr{L} = \{u \cdot \alpha \mid u \in K, \alpha \in \mathscr{L}\} \qquad J^\omega = \{u_1 u_2 \ldots \mid u_i \in J\}$$

with $L^0 = \{\epsilon\}$.

Remark 7.2 If $\epsilon \in J$, then we define $J^\omega = \emptyset$.

Definition 7.3 The set of *regular expressions over* Σ, $\mathcal{R}(\Sigma)$, is given by the following grammar:

$$e, f ::= 0 \mid 1 \mid a \in \Sigma \mid e + f \mid e \cdot f \mid e^*$$

We write $\mathcal{R}$ if Σ is understood.

The semantics for our regular expressions is given by regular languages. In order to define the language semantics, we use Definition 7.1.

Definition 7.4 The *language semantics* $[\![-]\!] : \mathcal{R} \to 2^{\Sigma^*}$ for regular expressions is given recursively:

$$[\![0]\!] = \emptyset \qquad [\![1]\!] = \{\epsilon\} \qquad [\![a]\!] = \{a\}$$
$$[\![e + f]\!] = [\![e]\!] \cup [\![f]\!] \qquad [\![e \cdot f]\!] = [\![e]\!] \cdot [\![f]\!] \qquad [\![e^*]\!] = [\![e]\!]^*$$

The following definition introduces the axiomatisation of *Kleene algebras*. This allows us to reason about regular expressions. Soundness is needed for the Brzozowski construction itself, while it is a known fact that this axiomatisation is also complete w.r.t. the language semantics ([20]).

Definition 7.5 The theory **KA** of Kleene algebras is axiomatised by the following quasi-equations:

$$e + 0 = e \qquad (e + f) + g = e + (f + g) \qquad (e + f) \cdot g = e \cdot g + f \cdot g$$
$$e + e = e \qquad (e \cdot f) \cdot g = e \cdot (f \cdot g) \qquad 1 + e \cdot e^* = e^* = 1 + e^* \cdot e$$
$$e + f = f + e \qquad e \cdot (f + g) = e \cdot f + e \cdot g \qquad e + f \cdot g \leq g \implies f^* \cdot e \leq g$$
$$e \cdot 1 = e = 1 \cdot e \qquad e \cdot 0 = 0 = 0 \cdot e \qquad e + f \cdot g \leq f \implies e \cdot g^* \leq f$$

where $e \leq f$ is an abbreviation for $e + f = f$.

For two regular expressions $e, f \in \mathcal{R}$, we write $e \equiv f$ if they are provably equivalent under **KA** and we write $e \leqq f$ for $e + f \equiv f$. A structure $\mathbb{K} = (K, +, \cdot, ^*, 0, 1)$ satisfying all the above equations is called a *Kleene Algebra*.

Proposition 7.6 (Soundness and Completeness) *Let $e, f \in \mathcal{R}$, then*

$$e \equiv f \iff [\![e]\!] = [\![f]\!].$$

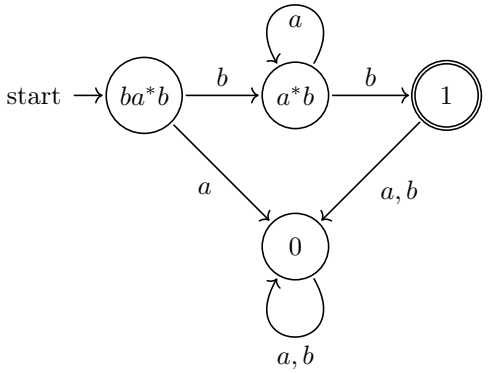
For the Brzozowski automaton, we need three things

1. the base set, which is $\mathcal{R}$ itself
2. a transition map, called the Brzozowski derivative
3. a set of final states

We show the idea behind the Brzozowski construction by means of an example. Let $\Sigma = \{a, b\}$ and consider the regular expression $e = ba^*b$. We take e as initial state. The derivative should be thought of as follows: If we look at the words in $[\![e]\!]$ and we read a b, what are we left with? For instance, $bab \in [\![e]\!]$, so after reading a b, we would have to still read ab. In more general terms, for any word in $[\![e]\!]$, after reading a b, we would still have to read a word of the shape a^*b. This notion of derivative on regular expressions is called the Brzozowski derivative.

Now if we read an a, then we will still remain at the state a^*b. In fact, we can read any finite number of a's and we would remain in a^*b. Finally, reading a b will lead us to the state 1 (after reading a b, the empty word is all that remains). The state 1 will be accepting and in general if an expression contains the empty word we will make it an accepting state.

The automaton we obtain according to this procedure looks as follows:



One can verify for oneself that this DFA accepts the language given by e. For the Brzozowski automaton, we start by defining the set of accepting states N.

Definition 7.7 We define N as the smallest subset of $\mathcal{R}$ which is closed under the following rules:

$$\frac{}{1 \in N} \qquad \frac{e \in N \quad f \in \mathcal{R}}{e + f, f + e \in N} \qquad \frac{e, f \in N}{e \cdot f \in N} \qquad \frac{e \in \mathcal{R}}{e^* \in N}$$

A standard inductive argument shows that N is characterised by the following property:

$$\epsilon \in [\![e]\!] \iff e \in N.$$

Definition 7.8 (Brzozowski Derivative) The *Brzozowski derivative* $d : \mathcal{R} \times \Sigma \to \mathcal{R}$ is defined inductively as:

$$\begin{array}{lll} d(0,a) = 0 & d(e+f,a) = d(e,a) + d(f,a) & d(b,a) = [b = a] \\ d(1,a) = 0 & d(e \cdot f,a) = d(e,a) \cdot f + [e \in N] \cdot d(f,a) & d(e^*,a) = d(e,a) \cdot e^* \end{array}$$

An important result regarding the Brzozowski derivative is the fundamental theorem. It relates a regular expression to its derivatives. The intuition behind the theorem is that a regular expression corresponds to the sum of its derivatives modulo some constant. Although an interesting result in its own right, it is also needed for the Brzozowski construction when showing that the language accepted by the Brzozowski automaton at e is equal to $[\![e]\!]$.

Theorem 7.9 (Fundamental Theorem) *Let $e \in \mathcal{R}$, then*

$$e \equiv [e \in N] + \sum_{a \in \Sigma} a \cdot d(e,a).$$

Having gathered all the necessary components, we are ready to define the Brzozowski automaton.

Definition 7.10 The *Brzozowski automaton* is the deterministic automaton

$$\mathbb{B} = (\mathcal{R}, d, N).$$

The next proposition explains why the Brzozowski automaton is so useful. The language accepted by the Brzozowski automaton initialised at e is $[\![e]\!]$.

Proposition 7.11 *Let $e \in \mathcal{R}$, then*

$$L(\mathbb{B}, e) = [\![e]\!].$$

There is one more technical point left. The Brzozowski automaton initialised at some regular expression e is in general not finite. This is not a problem, however, as we can define an equivalence relation which is compatible with the final states and the Brzozowski derivative. Moreover quotienting the Brzozowski automaton by said equivalence relation gives us a DFA such that there are only finitely many states reachable from the equivalence class of e.

Definition 7.12 Let $\sim \subseteq \mathcal{R} \times \mathcal{R}$ be the least equivalence relation including

$$1 \cdot e \sim e \qquad 0 \cdot e \sim 0 \qquad e \sim e + e \qquad e + f \sim f + e \qquad (e + f) + g \sim e + (f + g)$$

We need a technical lemma which shows that $\sim$ respects final states and the Brzozowski derivative.

Lemma 7.13 *Let $e, f \in \mathcal{R}$ with $e \sim f$, then*

1. $e \in N \iff f \in N$,
2. $\forall a \in \Sigma : d(e,a) \sim d(f,a)$.

Due to Proposition 7.13 the following Definition is well-defined.

Definition 7.14 Let $\underset{\sim}{\mathbb{B}} = (\mathcal{R}/\sim, \underset{\sim}{d}, \underset{\sim}{N})$, where

1. $\underset{\sim}{d}([e]_\sim, a) = [d(e,a)]_\sim$,
2. $[e]_\sim \in \underset{\sim}{N} \iff e \in N$.

The only thing that remains is to show that by taking the quotient, the new automaton we obtain still accepts the same language. A particularly elegant way of showing this is by means of bisimulations.

Proposition 7.15 *For all $e \in \mathcal{R}$:*

$$L(\mathbb{B}, e) = L(\underset{\sim}{\mathbb{B}}, [e]_\sim).$$

Moreover, the collection of states in $\underset{\sim}{\mathbb{B}}$ reachable from $[e]_\sim$ is finite.

7.2 Brzozowski Construction for Ω-automata

We follow the previous section in constructing an Ω-automaton from an ω-regular expression. For this we need to specify a set of spoke states, a set of loop states, a set of accepting states and the three types of transitions (spoke, switch and loop transitions).

For the spoke states we take ω-regular expressions as given in the following definition:

Definition 7.16 ([33]) The set of *ω-regular expressions over* Σ , $\mathcal{R}_\omega(\Sigma)$, is given by the following grammar, where $e, f \in \mathcal{R}$ with $e \notin N$:

$$\kappa, \lambda := 0_\omega \mid \kappa + \lambda \mid f \otimes \kappa \mid e^\omega$$

and we write $\mathcal{R}_\omega$ if Σ is understood.

As with regular expressions, we also give semantics for ω-regular expressions, this time using ω-regular languages.

Definition 7.17 ([33]) The *language map* $[\![-]\!]_\omega : \mathcal{R}_\omega \to 2^{\Sigma^\omega}$ for ω-regular expressions is defined recursively as:

$$[\![0_\omega]\!]_\omega = \emptyset \qquad [\![\kappa + \lambda]\!]_\omega = [\![\kappa]\!]_\omega \cup [\![\lambda]\!]_\omega$$
$$[\![f \otimes \kappa]\!]_\omega = [\![f]\!] \cdot [\![\kappa]\!]_\omega \qquad [\![e^\omega]\!]_\omega = [\![e]\!]^\omega$$

where $e, f \in \mathcal{R}$.

For the loop states of our Ω-automaton, we take regular expressions. For the spoke and loop transitions there are very natural candidates. For the spoke transition we define a Brzozowski derivative for ω-regular expressions, whereas the loop transition is just the Brzozowski derivative for regular expressions. The final states consist exactly of those regular expressions which contain the empty word. This leaves us with one final component to define, the switch transition.

The switch transition has to switch from ω-regular expressions to regular expressions. A clue as to how to define the switch transition is provided by looking at the loop language of a given spoke state. To remind the reader, we defined the loop automaton at a spoke state x to be the DFA whose states are given by x and all the loop states y reachable from x via the switch and loop transitions. The loop language at x is the regular language accepted by the loop automaton at x. The observation is the following: let $\mathbb{B}_\omega$ denote the Brzozowski Ω-automaton we would like to construct and let $\lambda \in \mathcal{R}_\omega$ be a spoke state, then we would like the ω-language accepted at λ to be $[\![\lambda]\!]_\omega$ and so we require

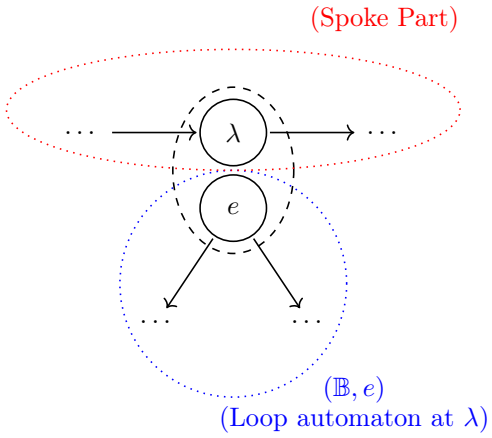
$$u \in \mathrm{Loop}(\lambda) \iff (\epsilon, u) \in \mathrm{Lassos}(\mathbb{B}_\omega, \lambda) \iff u^\omega \in [\![\lambda]\!]_\omega.$$

From this it follows that the loop automaton at λ has to accept the following regular language:

$$\{u \in \Sigma^+ \mid u^\omega \in [\![\lambda]\!]_\omega\}.$$

If we have the regular expression e corresponding to this set, then the loop automaton at λ is just the classical Brzozowski automaton initialised at e.

In this subsection, we show that under the assumption that we can extract e from λ, we can construct a Brzozowski Ω-automaton.



As we build our Brzozowski Ω-automaton, we have to reason not only about regular, but also about ω-regular expressions. In particular, for certain propositions we require a soundness result. Hence we introduce an axiomatisation for ω-regular expressions due to Klaus Wagner ([32]).

Definition 7.18 ([32]) We define $\equiv_\omega$ as the smallest congruence on $\mathcal{R}_\omega$ such that:

$$\begin{array}{lll} \kappa + \lambda \equiv_\omega \lambda + \kappa & e \otimes (\kappa + \lambda) \equiv_\omega e \otimes \kappa + e \otimes \lambda & e^\omega \equiv_\omega (e \cdot e^*)^\omega \\ (\kappa + \lambda) + \mu \equiv_\omega \kappa + (\lambda + \mu) & (e + f) \otimes \kappa \equiv_\omega e \otimes \kappa + f \otimes \kappa & (e \cdot f)^\omega \equiv_\omega e \otimes (f \cdot e)^\omega \\ e \otimes (f \otimes \kappa) \equiv_\omega (e \cdot f) \otimes \kappa & & 0 \otimes \kappa \equiv_\omega 0_\omega \end{array}$$

where $e, f \in \mathcal{R}$.

In [32], Wagner deduces multiple rules from his axiomatisation. We collect them in the following lemma.

Lemma 7.19 ([32]) *Let $e \in \mathcal{R}$, $\kappa \in \mathcal{R}_\omega$ and $k \geq 1$, then*

$$\begin{array}{llll} e \otimes 0_\omega \equiv_\omega 0_\omega & e^\omega \equiv_\omega e \otimes e^\omega & \kappa \equiv_\omega \kappa + \kappa & 1 \otimes \kappa \equiv_\omega \kappa \\ 0_\omega + \kappa \equiv_\omega \kappa & 0^\omega \equiv_\omega 0_\omega & e^\omega \equiv_\omega e^* \otimes e^\omega & e^\omega \equiv_\omega (e^k)^\omega \end{array}$$

Proposition 7.20 (Soundness, [32]) *Let $\kappa, \lambda \in \mathcal{R}_\omega$, then*

$$\kappa \equiv_\omega \lambda \implies [\![\kappa]\!]_\omega = [\![\lambda]\!]_\omega.$$

Next, we present the ω-Brzozowski derivative d_ω and the fundamental theorem for d_ω and ω-regular expressions. We point out that d_ω is defined analogously to the standard Brzozowski derivative.

Definition 7.21 ([30]) Define the function $d_\omega : \mathcal{R}_\omega \times \Sigma \to \mathcal{R}_\omega$ inductively as

$$\begin{array}{ll} d_\omega(0_\omega, a) = 0_\omega & d_\omega(\kappa + \lambda, a) = d_\omega(\kappa, a) + d_\omega(\lambda, a) \\ d_\omega(e \otimes \kappa, a) = d(e, a) \otimes \kappa + [e \in N] \otimes d_\omega(\kappa, a) & d_\omega(e^\omega, a) = d(e, a) \otimes e^\omega \end{array}$$

A fundamental theorem for ω-regular expressions can be found in [30] but it is only given in terms of semantics, i.e. it states that

$$[\![\kappa]\!]_\omega = \bigcup_{a \in \Sigma} \{a\} \cdot [\![d_\omega(\kappa, a)]\!]_\omega.$$

We give here the syntactic version, from which the above follows immediately by soundness (Proposition 7.20).

Theorem 7.22 (Fundamental Theorem for $\mathcal{R}_\omega$) *Let $\kappa \in \mathcal{R}_\omega$, then*

$$\kappa \equiv_\omega \sum_{a \in \Sigma} a \otimes d_\omega(\kappa, a).$$

Proof. We show the fundamental theorem by induction on the construction of ω-regular expressions:

$\underline{\kappa = 0_\omega}$:

$$\sum_{a \in \Sigma} a \otimes d_\omega(0_\omega, a) = \sum_{a \in \Sigma} a \otimes 0_\omega \equiv_\omega 0_\omega$$

$\underline{\kappa = \lambda + \mu}$:

$$
\begin{aligned}
\kappa &= \lambda + \mu \\
&\equiv_\omega \sum_{a\in\Sigma} a \otimes d_\omega(\lambda, a) + \sum_{a\in\Sigma} a \otimes d_\omega(\mu, a) && \text{(I.H.)} \\
&\equiv_\omega \sum_{a\in\Sigma} a \otimes (d_\omega(\lambda, a) + d_\omega(\mu, a)) && \text{(Dist.)} \\
&= \sum_{a\in\Sigma} a \otimes d_\omega(\kappa, a) && \text{(Defn. } d_\omega\text{)}
\end{aligned}
$$

$\underline{\kappa = f \otimes \lambda}$:

$$
\begin{aligned}
\kappa &= f \otimes \lambda \\
&\equiv_\omega \left([f \in N] + \sum_{a\in\Sigma} a \cdot d(f, a) \right) \otimes \lambda && \text{(Thm. 7.9)} \\
&\equiv_\omega [f \in N] \otimes \lambda + \sum_{a\in\Sigma} (a \cdot d(f, a)) \otimes \lambda && \text{(Dist.)} \\
&\equiv_\omega [f \in N] \otimes \sum_{a\in\Sigma} a \otimes d_\omega(\lambda, a) + \sum_{a\in\Sigma} a \otimes (d(f, a) \otimes \lambda) && \text{(I.H. \& Ass.)} \\
&\equiv_\omega \sum_{a\in\Sigma} [f \in N] \otimes (a \otimes d_\omega(\lambda, a)) + \sum_{a\in\Sigma} a \otimes (d(f, a) \otimes \lambda) && \text{(Dis.)} \\
&\equiv_\omega \sum_{a\in\Sigma} ([f \in N] \cdot a) \otimes d_\omega(\lambda, a) + \sum_{a\in\Sigma} a \otimes (d(f, a) \otimes \lambda) && \text{(Ass.)} \\
&\equiv_\omega \sum_{a\in\Sigma} (a \cdot [f \in N]) \otimes d_\omega(\lambda, a) + \sum_{a\in\Sigma} a \otimes (d(f, a) \otimes \lambda) && (a \cdot (0+1) \equiv (0+1) \cdot a) \\
&\equiv_\omega \sum_{a\in\Sigma} a \otimes ([f \in N] \otimes d_\omega(\lambda, a)) + \sum_{a\in\Sigma} a \otimes (d(f, a) \otimes \lambda) && \text{(Ass.)} \\
&\equiv_\omega \sum_{a\in\Sigma} a \otimes ([f \in N] \otimes d_\omega(\lambda, a) + d(f, a) \otimes \lambda) && \text{(Dis.)} \\
&= \sum_{a\in\Sigma} a \otimes d_\omega(\kappa, a) && \text{(Defn. } d_\omega\text{)}
\end{aligned}
$$

$\underline{\kappa = e^\omega}$:

$$\begin{aligned}
\kappa &= e^\omega \\
&\equiv_\omega e \otimes e^\omega && \text{(Lem. 7.19)} \\
&\equiv_\omega \left(\sum_{a \in \Sigma} a \cdot d(e,a) \right) \otimes e^\omega && \text{(Thm. 7.9, } e \notin N\text{)} \\
&\equiv_\omega \sum_{a \in \Sigma} (a \cdot d(e,a)) \otimes e^\omega && \text{(Dist.)} \\
&\equiv_\omega \sum_{a \in \Sigma} a \otimes (d(e,a) \otimes e^\omega) && \text{(Ass.)} \\
&= \sum_{a \in \Sigma} a \otimes d_\omega(\kappa, a) && \text{(Defn. } d_\omega\text{)}
\end{aligned}$$

□

This is the point at which we jump back to the switch transition. At the start, we considered a spoke state λ, and pointed out that the loop automaton at λ has to accept the following regular language

$$\{u \in \Sigma^+ \mid u^\omega \in [\![\lambda]\!]_\omega\}.$$

This collection is referred to as the *pointwise ω-root of* $[\![\lambda]\!]_\omega$.

Definition 7.23 Let $\mathscr{L} \subseteq \Sigma^\omega$, the *pointwise ω-root of* $\mathscr{L}$ is defined as

$$\sqrt[\bullet]{\mathscr{L}} = \{u \in \Sigma^+ \mid u^\omega \in \mathscr{L}\}.$$

This subsection does not cover how to find a regular expression for the pointwise ω-root of an ω-regular language. This is looked at in more detail in the following subsection. Instead, we show that, provided we have a function which computes the regular expression corresponding to the pointwise ω-root, we can construct our desired Brzozowski Ω-automaton.

Definition 7.24 For an ω-regular expression κ, we define

$$d_t(\kappa, a) = d(g, a),$$

for $a \in \Sigma$ and where g is such that

$$[\![g]\!] = \sqrt[\bullet]{[\![\kappa]\!]_\omega}.$$

The switch transition was the last component we needed, allowing us to define the Brzozowski Ω-coalgebra. Once we show that the Brzozowski Ω-coalgebra with initial state κ accepts $[\![\kappa]\!]_\omega$, it is clear that we succeeded and obtained the Brzozowski Ω-automaton.

Definition 7.25 For an ω-regular expression, we define the *Brzozowski Ω-coalgebra* $\mathbb{B}_\omega = (\mathcal{R}_\omega, \mathcal{R}, d_\omega, d_t, d, N)$.

The following technical lemma is helpful in establishing that the pointed Brzozowski Ω-coalgebra $(\mathbb{B}_\omega, \kappa)$ accepts $[\![\kappa]\!]_\omega$.

Lemma 7.26 *Let κ be an ω-regular expression and g a regular expression such that $[\![g]\!] = \dot{\sqrt[\omega]{[\![\kappa]\!]_\omega}}$, then*

$$[\![g]\!] = L(\mathbb{B}, g) = L(\mathbb{B}_{\omega,\ell}, \kappa) = \mathit{Loop}(\mathbb{B}_\omega, \kappa).$$

Proof. We show that the two automata $B = (\mathcal{R}, g, d, N)$ and $\mathbb{B}_{\omega,\ell} = (\mathcal{R}_\omega \uplus \mathcal{R}, \kappa, d_t : d, N)$ (where we only consider the reachable part) are bisimilar. In fact, they are identical with the exception for g and κ. To see this we show that for all $u \in \Sigma^+$, we have

$$\widetilde{d}(g, u) = \widetilde{d_t : d}(\kappa, u).$$

The base case is given by the definition of d_t. For the induction step, we have

$$\widetilde{d}(g, ua) = d(\widetilde{d}(g, u), a) \overset{\text{(I.H.)}}{=} d(\widetilde{d_t : d}(\kappa, u), a) = \widetilde{d_t : d}(\kappa, ua).$$

As they both have the same final states, they are bisimilar and so they accept the same regular language. □

Theorem 7.27 *The structure $\mathbb{B}_\omega = (\mathcal{R}_\omega, \mathcal{R}, d_\omega, d_t, d, N)$ is an Ω-automaton with*

$$L(\mathbb{B}_\omega, \kappa) = [\![\kappa]\!]_\omega.$$

Proof. We show that for any ω-regular expression κ, the lasso language $\mathrm{Lassos}(\mathbb{B}_\omega, \kappa)$ consists exactly of the ultimately periodic fragment of $[\![\kappa]\!]_\omega$. This is done by induction.

For the base case let $v \in \Sigma^+$ and consider the lasso (ϵ, v), we want to show that

$$v^\omega \in [\![\kappa]\!]_\omega \iff (\epsilon, v) \in \mathrm{Lassos}(\mathbb{B}_\omega, \kappa).$$

We have that

$$\begin{aligned} v^\omega \in [\![\kappa]\!]_\omega &\iff v \in \dot{\sqrt[\omega]{[\![\kappa]\!]_\omega}} && \text{(Defn. 7.23)} \\ &\iff v \in \mathrm{Loop}(\mathbb{B}_\omega, \kappa) && \text{(Lemma 7.26)} \\ &\iff (\epsilon, v) \in \mathrm{Lassos}(\mathbb{B}_\omega, \kappa). \end{aligned}$$

For the inductive case, consider the lasso $(au, v) \in \Sigma^{*+}$, then

$$\begin{aligned} auv^\omega \in [\![\kappa]\!]_\omega &\iff uv^\omega \in [\![d_\omega(\kappa, a)]\!]_\omega && \text{(Sound. \& Thm 7.22)} \\ &\iff (u, v) \in \mathrm{Lassos}(\mathbb{B}_\omega, d_\omega(\kappa, a)) && \text{(I.H.)} \\ &\iff (au, v) \in \mathrm{Lassos}(\mathbb{B}_\omega, \kappa). \end{aligned}$$

So $L(\mathbb{B}_\omega, \kappa) = [\![\kappa]\!]_\omega$, and the initialised Ω-coalgebra $(\mathbb{B}_\omega, \kappa)$ is an Ω-automaton for all ω-regular expressions κ. □

As in the classical case, our Brzozowski Ω-automaton is in general not finite. This is dealt with in a similar manner as before by defining an equivalence relation and quotienting the Brzozowski Ω-automaton.

Definition 7.28 We overload $\sim$ and also write $\sim$ for the smallest equivalence relation on $\mathcal{R}_\omega$ including

$$1 \otimes \kappa \sim \kappa \quad 0 \otimes \kappa \sim 0_\omega \quad \kappa \sim \kappa + \kappa \quad \kappa + \lambda \sim \lambda + \kappa \quad (\kappa + \lambda) + \mu \sim \kappa + (\lambda + \mu)$$

As we already know that the classical Brzozowski DFA quotiented by $\sim$ has only finitely many states which are reachable by any given regular expression, we use this to our advantage. Many of the concepts we defined were analogous to those defined in the classical setting. Given some ω-regular expression, we could substitute all the ω-stars for Kleene stars. What we would end up with is a regular expression. Moreover, this substitution respects the transition functions and the equivalence relations $\sim$. This later allows us to deduce that our quotient Ω-automaton must also be finite if we only consider reachable states.

Definition 7.29 We define the map $\pi : \mathcal{R}_\omega \to \mathcal{R}$ inductively:

$$\pi(0_\omega) = 0 \qquad \pi(e^\omega) = e^*$$
$$\pi(\kappa + \lambda) = \pi(\kappa) + \pi(\lambda) \qquad \pi(f \otimes \kappa) = f \cdot \pi(\kappa)$$

Lemma 7.30 *The map π is injective.*

Proof. We show this by induction on the construction of expressions. For $\kappa = 0_\omega$, given $\pi(\kappa) = \pi(\lambda)$ it follows that

$$0 = \pi(\kappa) = \pi(\lambda) \implies \lambda = 0_\omega = \kappa.$$

Let $\kappa = e^\omega$ and $\pi(\kappa) = \pi(\lambda)$, then

$$e^* = \pi(\kappa) = \pi(\lambda) \implies \lambda = e^\omega = \kappa.$$

Let $\kappa = \kappa_1 + \kappa_2$ and $\pi(\kappa) = \pi(\lambda)$, then

$$\pi(\kappa_1) + \pi(\kappa_2) = \pi(\kappa) = \pi(\lambda).$$

This implies that $\lambda = \lambda_1 + \lambda_2$ and $\pi(\kappa_i) = \pi(\lambda_i)$. By the induction hypothesis we have $\kappa_i = \lambda_i$ so $\kappa = \lambda$. Lastly, let $\kappa = f \otimes \kappa_1$ and $\pi(\kappa) = \pi(\lambda)$, then

$$f \cdot \pi(\kappa_1) = \pi(\kappa) = \pi(\lambda),$$

so $\lambda = f \otimes \lambda_1$ and $\pi(\kappa_1) = \pi(\lambda_1)$. By induction hypothesis $\kappa_1 = \lambda_1$ and so $\kappa = \lambda$. □

We show a technical lemma which states that π respects derivatives and the equivalence relation $\sim$.

Lemma 7.31 *Let $\kappa, \lambda \in \mathcal{R}_\omega$, then*

1. $\pi(d_\omega(\kappa, a)) = d(\pi(\kappa), a)$,
2. $\kappa \sim \lambda \iff \pi(\kappa) \sim \pi(\lambda)$.

Proof. We show (1.) by induction on κ. There are four cases to consider. For the first case, let $\kappa = 0_\omega$, then

$$\pi(d_\omega(0_\omega, a)) = \pi(0_\omega) = 0 = d(0, a) = d(\pi(0_\omega), a).$$

For the second case, we consider $e \otimes \kappa$, then

$$\begin{aligned}
\pi(d_\omega(e \otimes \kappa, a)) &= \pi(d(e,a) \otimes \kappa + [e \in N] \otimes d_\omega(\kappa, a)) \\
&= d(e,a) \cdot \pi(\kappa) + [e \in N] \cdot \pi(d_\omega(\kappa, a)) \\
&= d(e,a) \cdot \pi(\kappa) + [e \in N] \cdot d(\pi(\kappa), a) && \text{(I.H.)} \\
&= d(e \cdot \pi(\kappa), a) \\
&= d(\pi(e \otimes \kappa), a).
\end{aligned}$$

For the third case, we have

$$\begin{aligned}
\pi(d_\omega(\kappa + \lambda, a)) &= \pi(d_\omega(\kappa, a) + d_\omega(\lambda, a)) \\
&= \pi(d_\omega(\kappa, a)) + \pi(d_\omega(\lambda, a)) \\
&= d(\pi(\kappa), a) + d(\pi(\lambda), a) && \text{(I.H.)} \\
&= d(\pi(\kappa) + \pi(\lambda), a) \\
&= d(\pi(\kappa + \lambda), a).
\end{aligned}$$

And for the remaining case:

$$\pi(d_\omega(e^\omega, a)) = \pi(d(e,a) \otimes e^\omega) = d(e,a) \cdot \pi(e^\omega) = d(e,a) \cdot e^* = d(e^*, a) = d(\pi(e^\omega), a).$$

With regards to (2.), the proof follows through the following observations:

$$\pi(\kappa + \kappa) = \pi(\kappa) + \pi(\kappa) \qquad \pi(1 \otimes \kappa) = 1 \cdot \pi(\kappa) \qquad \pi(0 \otimes \kappa) = 0 \cdot \pi(\kappa) \qquad \square$$

Proposition 7.32 *Let $\kappa \sim \lambda$, then $d_\omega(\kappa, a) \sim d_\omega(\lambda, a)$ for all $a \in \Sigma$.*

Proof. Let $\kappa \sim \lambda$, then by Lemma 7.31, we have that $\pi(\kappa) \sim \pi(\lambda)$. By Lemma 7.13 it follows that $d(\pi(\kappa), a) \sim d(\pi(\lambda), a)$ and again by 7.31 we have $\pi(d_\omega(\kappa, a)) \sim \pi(d_\omega(\lambda, a))$ and finally that $d_\omega(\kappa, a) \sim d_\omega(\lambda, a)$ as required. $\square$

Before we can take the quotient there is one more technical difficulty. We would like to have that $\kappa \sim \lambda$ implies that $d_t(\kappa, a) \sim d_t(\lambda, a)$ for all $a \in \Sigma$. As it stands now, and using completeness (Prop. 7.6), we can show that if $\kappa \sim \lambda$, then $d_t(\kappa, a) \equiv d_t(\lambda, a)$:

$$\begin{aligned}
\kappa \sim \lambda &\implies \kappa \equiv_\omega \lambda \\
&\implies [\![\kappa]\!]_\omega = [\![\lambda]\!]_\omega \\
&\implies \sqrt[\bullet]{[\![\kappa]\!]_\omega} = \sqrt[\bullet]{[\![\lambda]\!]_\omega} \\
&\implies g \equiv g' && \left([\![g]\!] = \sqrt[\bullet]{[\![\kappa]\!]_\omega}, [\![g']\!] = \sqrt[\bullet]{[\![\lambda]\!]_\omega}\right) \\
&\implies d(g,a) \equiv d(g',a) \\
&\implies d_t(\kappa, a) \equiv d_t(\lambda, a)
\end{aligned}$$

This, however, does not imply that $d_t(\kappa, a) \sim d_t(\lambda, a)$. There is two ways around this issue. The first solution would be to add additional requirements to how we obtain a regular expression g from an ω-regular expression κ with $[\![g]\!] = \sqrt[\bullet]{[\![\kappa]\!]_\omega}$. We could for instance formalise a function $r : \mathcal{R}_\omega \to \mathcal{R}$ which maps every ω-regular expression κ

to $r(\kappa) = g$ satisfying $[\![g]\!] = \dot{\sqrt[\omega]{[\![\kappa]\!]_\omega}}$ and on top of that require r to satisfy additional properties such as $r(\kappa + \lambda) = r(\kappa) + r(\lambda)$. The idea is that the additional requirements would allow us to conclude that $d_t(\kappa, a) \sim d_t(\lambda, a)$.

A second and less favourable solution is to argue that we do not need this requirement. The spoke and loop part of the Ω-automaton can namely be quotiented separately. The issue with this is that we would still have to provide the switch transition. In this case, we would have to choose from each $[\kappa]_\sim$ equivalence class a representative κ' and define $\underset{\sim}{d_t}([\kappa]_\sim, a) = [d(\kappa', a)]_\sim$.

For our purposes, we chose the first approach and assume that for $\kappa \sim \lambda$ we also have $d_t(\kappa, a) \sim d_t(\lambda, a)$. By Proposition 7.32 and our assumption, the following definition is well-defined.

Definition 7.33 Let $\underset{\sim}{\mathbb{B}_\omega} = (\mathcal{R}_\omega/\!\sim, \mathcal{R}/\!\sim, \underset{\sim}{d_\omega}, \underset{\sim}{d_t}, \underset{\sim}{d}, \underset{\sim}{N})$ and $\kappa \in \mathcal{R}_\omega$, where

1. $\underset{\sim}{d_\omega}([\kappa]_\sim, a) = [d_\omega(\kappa, a)]_\sim$,
2. $\underset{\sim}{d_t}([\kappa]_\sim, a) = [d_t(\kappa, a)]_\sim$.

We turn to the question of size. The loop part is finite simply by Proposition 7.15, so we only have to show that the spoke part is also finite. The idea is to turn the spoke part of $\underset{\sim}{\mathbb{B}_\omega}$ into a DFA and then use π and Proposition 7.15 to show that its size is bounded. All we need to turn the spoke part into a DFA is a set of final states.

Lemma 7.34 *Let*

$$N_\omega = \{\kappa \mid \pi(\kappa) \in N\}.$$

If $\kappa \sim \lambda$, then $\kappa \in N_\omega \iff \lambda \in N_\omega$.

Proof. Let $\kappa \sim \lambda$, by Lemma 7.31 we have that $\pi(\kappa) \sim \pi(\lambda)$, then by Lemma 7.13 it follows that

$$\kappa \in N_\omega \iff \pi(\kappa) \in N \overset{7.13}{\iff} \pi(\lambda) \in N \iff \lambda \in N_\omega. \qquad \square$$

Proposition 7.35 *Let $\kappa \in \mathcal{R}$, then the size of the DFA $(\mathcal{R}_\omega/\!\sim, [\kappa]_\sim, \underset{\sim}{d_\omega}, \underset{\sim}{N_\omega})$ is less than or equal to that of $(\mathcal{R}/\!\sim, [\pi(\kappa)]_\sim, \underset{\sim}{d}, \underset{\sim}{N})$ (where we only consider the reachable part in both automata).*

Proof. Consider the map $\underset{\sim}{\pi} : \mathcal{R}_\omega/\!\sim \to \mathcal{R}/\!\sim$ given by $\underset{\sim}{\pi}([\kappa]_\sim) = [\pi(\kappa)]_\sim$. This is well-defined because of Lemma 7.31. We show that this map is a coalgebra morphism (where the DFAs are seen as coalgebras for the functor $T = (-)^\Sigma \times 2$). This consists of showing two things:

1. $[\kappa]_\sim \in \underset{\sim}{N_\omega} \iff \underset{\sim}{\pi}([\kappa]_\sim) \in \underset{\sim}{N}$,
2. $\underset{\sim}{\pi}(\underset{\sim}{d_\omega}([\kappa]_\sim, a)) = \underset{\sim}{d}(\underset{\sim}{\pi}([\kappa]_\sim), a)$.

For (1.) we have

$$[\kappa]_\sim \in \underset{\sim}{N_\omega} \iff \kappa \in N_\omega \iff \pi(\kappa) \in N \iff [\pi(\kappa)]_\sim \in \underset{\sim}{N} \iff \underset{\sim}{\pi}([\kappa]_\sim) \in \underset{\sim}{N}.$$

As for (2.) we have

$$\begin{aligned}\underset{\sim}{\pi}(\underset{\sim}{d_\omega}([\kappa]_\sim, a)) &= \underset{\sim}{\pi}([d_\omega(\kappa, a)]_\sim) \\ &= [\pi(d_\omega(\kappa, a))]_\sim \\ &= [d(\pi(\kappa), a)]_\sim \\ &= \underset{\sim}{d}([\pi(\kappa)]_\sim, a) \\ &= \underset{\sim}{d}(\underset{\sim}{\pi}([\kappa]_\sim), a).\end{aligned}$$

Finally, $\underset{\sim}{\pi}$ is injective as

$$\begin{aligned}\underset{\sim}{\pi}([\kappa]_\sim) = \underset{\sim}{\pi}([\lambda]_\sim) &\implies [\pi(\kappa)]_\sim = [\pi(\lambda)]_\sim \\ &\implies \pi(\kappa) \sim \pi(\lambda) \\ &\implies \kappa \sim \lambda \\ &\implies [\kappa]_\sim = [\lambda]_\sim.\end{aligned}$$

Hence the claim follows. □

Proposition 7.36 *For all $\kappa \in \mathcal{R}_\omega$:*

$$L(\underset{\sim}{\mathbb{B}_\omega}, [\kappa]_\sim) = L(\mathbb{B}_\omega, \kappa).$$

Moreover, the collection of states in $\underset{\sim}{\mathbb{B}_\omega}$ reachable from $[\kappa]_\sim$ is finite.

Proof. We can construct a bisimulation between $(\underset{\sim}{\mathbb{B}_\omega}, [\kappa]_\sim)$ and $(\mathbb{B}_\omega, \kappa)$. Let $Z_0 = \{\langle \kappa, [\kappa]_\sim \rangle \mid \kappa \in \mathcal{R}_\omega\}$ and $Z_1 = \{\langle e, [e]_\sim \rangle \mid e \in \mathcal{R}\}$. We claim that (Z_0, Z_1) is a bisimulation. This is rather straight forward to verify. For the final states, we have

$$e \in N \iff [e]_\sim \in \underset{\sim}{N}.$$

For the spoke transitions, consider the pair $\langle \lambda, [\lambda]_\sim \rangle \in Z_0$, then

$$[d_\omega(\lambda, a)]_\sim = \underset{\sim}{d_\omega}([\lambda]_\sim, a),$$

so $\langle d_\omega(\lambda, a), [d_\omega(\lambda, a)]_\sim \rangle \in Z_0$. The proofs for d_t and d are analogous as they all respect $\sim$. Finally, by Proposition 7.15 and Proposition 7.35, it follows that there are only finitely many states in $\underset{\sim}{\mathbb{B}_\omega}$ that are reachable from $[\kappa]_\sim$. □

7.3 Lasso Expressions

This subsection introduces *lasso expressions* in order to provide an answer to finding a regular expression corresponding to the pointwise ω-root of an ω-regular language.

First, we characterise the pointwise ω-root. We show that if a regular language L contains enough information to obtain from it the periodic fragment of an ω-regular language $\mathscr{L}$, and if L is sufficiently large, then the root of L (Definition 7.37) corresponds precisely to the pointwise ω-root of $\mathscr{L}$.

Given some ω-regular expression κ, we use this characterisation to help us find a regular expression corresponding to the pointwise ω-root of $[\![\kappa]\!]_\omega$. This is done by first

constructing a lasso expression which is equivalent to κ in some sense. Next we lift the reduction rule γ_1 from lassos to lasso expressions, to obtain all the lassos of the form (ϵ, u) representing periodic streams. We show that we can get an expression e capturing the loop part of said expressions and that the language $[\![e]\!]$ satisfies all the necessary requirements as laid out by our characterisation.

As we can effectively compute the root of a regular language ([21]), this shows that we can construct the pointwise ω-root of an ω-regular language. The definition of the *root of a regular language* was taken from ([21]). Obtaining the root of a regular language is also discussed in [13, 15].

Definition 7.37 For a regular language L, we define

1. the *pointwise ω-power* of L: $L^{\dot{\omega}} = \{u^\omega \mid u \in L\}$,
2. the *root* of L : $\sqrt{L} = \{u \in \Sigma^+ \mid \exists k \geq 1 : u^k \in L\}$.

Remark 7.38 We want to mention that for an ω-regular language $\mathscr{L}$, it must necessarily be the case that $\sqrt[\dot{\omega}]{\mathscr{L}}$ is regular. To see this, take an ω-regular language and consider a finite Ω-automaton accepting it. If i is the initial state, then $\mathrm{Loop}(i)$ corresponds to the pointwise ω-root of $\mathscr{L}$ and it is regular as it corresponds to the language accepted by the loop automaton at i.

Our first proposition gives a characterisation of the pointwise ω-root. Given a regular language L and an ω-regular language $\mathscr{L}$, it states the constraints under which the root of L is equivalent to the pointwise ω-root of $\mathscr{L}$.

Proposition 7.39 *Let L be a regular language and $\mathscr{L}$ an ω-regular language such that*

1. $L^{\dot{\omega}} \subseteq \mathscr{L}$,
2. $\forall u \in \Sigma^+ : \big(u^\omega \in \mathscr{L} \implies \exists k \geq 1 : u^k \in L\big)$,

then

$$\sqrt{L} = \sqrt[\dot{\omega}]{\mathscr{L}}.$$

Proof. We show the left-to-right inclusion first. Let $u \in \sqrt{L}$, then $u^k \in L$ for some $k \geq 1$ and by the first condition

$$u^\omega = (u^k)^\omega \in L^{\dot{\omega}} \implies u^\omega \in \mathscr{L},$$

and so $u \in \sqrt[\dot{\omega}]{\mathscr{L}}$.

For the other inclusion, let $u \in \sqrt[\dot{\omega}]{\mathscr{L}}$, then $u^\omega \in \mathscr{L}$ and so by the second condition there is some $k \geq 1$ such that $u^k \in L$ by which it follows that $u \in \sqrt{L}$. □

Given an ω-regular expression κ, our aim is to construct a regular expression e and then show that $[\![e]\!]$ satisfies the requirements of Proposition 7.39 by which it follows that e corresponds to the pointwise ω-root of κ.

We may think of the conditions in Proposition 7.39 as stating in some sense that L has to cover $\mathscr{L}$. This is a good first intuition as the pointwise ω-root corresponds to the maximal such regular language as shown by the following proposition:

Proposition 7.40 *Let L be a regular language and $\mathscr{L}$ an ω-regular language, then*

$$L^{\dot{\omega}} = \mathscr{L} \implies L \subseteq \sqrt[\dot{\omega}]{\mathscr{L}}.$$

Proof. Let $L^{\dot{\omega}} = \mathscr{L}$ and let $u \in L$, then $u^\omega \in \mathscr{L}$ and so $u \in \sqrt[\dot{\omega}]{\mathscr{L}}$. □

Remark 7.41 One might wonder whether the other inclusion of Proposition 7.40 holds (i.e. $\sqrt[\dot{\omega}]{\mathscr{L}} \subseteq L$). The following example should make this clearer; Consider the ω-regular language $\mathscr{L} = \{a^\omega\}$ and the regular language $L = \{aa\}$. Clearly $L^{\dot{\omega}} = \mathscr{L}$ as

$$L^{\dot{\omega}} = \{(aa)^\omega\} = \{a^\omega\} = \mathscr{L}.$$

Yet

$$\sqrt[\dot{\omega}]{\mathscr{L}} = \{a^k \mid k \geq 1\} \neq L.$$

Definition 7.42 A *basic lasso expression* is an expression of the form (e, f) where e, f are both regular expressions such that $f \notin N$. A *lasso expression* is a finite sum of basic lasso expressions, i.e. if $(e, f), (e', f')$ are basic lasso expressions, then $(e, f) + (e', f')$ is a lasso expression. Moreover, given a regular expression g and a lasso expression $m = (e_1, f_1) + \ldots + (e_n, f_n)$ we define

$$g \cdot m = g \cdot \sum_{i=1}^{n} (e_i, f_i) = \sum_{i=1}^{n} (g \cdot e_i, f_i) = (g \cdot e_1, f_1) + \ldots + (g \cdot e_n, f_n).$$

We denote the collection of lasso expressions by $\mathscr{L}$.

Definition 7.43 (Semantics) For a lasso expression $m = \sum_i (e_i, f_i)$ we define

$$[\![m]\!]_\ell = \left\{(u, v) \in \Sigma^{*+} \mid \exists i : u \in [\![e_i]\!] \text{ and } v \in [\![f_i]\!]\right\}.$$

Definition 7.44 A lasso expression m *covers* an ω-regular expression κ if

1. $\forall uv^\omega \in [\![\kappa]\!]_\omega : \exists (u', v') \sim_\gamma \mathrm{nf}(uv^\omega) : (u', v') \in [\![m]\!]_\ell$,
2. $\forall (u, v) \in [\![m]\!]_\ell : uv^\omega \in [\![\kappa]\!]_\omega$.

and we call an ω-regular expression *coverable* if it can be covered by some lasso expression.

Given an ω-regular expression κ, a lasso expression m covering κ means that for every ultimately periodic word in κ, there is at least one lasso representative in m. At the same time, for any lasso in m, the ultimately periodic word corresponding to it has to belong to κ. In this way, the lasso expression m can be thought of as being equivalent to κ.

This raises the question of whether any ω-regular expression is coverable. We show that this is indeed the case, which means that lasso expressions are in some sense at least as expressive as ω-regular expressions. Before we can proof this result, we need some definitions.

The next definition is taken from [18, Definition 4.5] (which the authors in turn attribute to [14]).

Definition 7.45 ([18, 14]) We define the splitting relation ∇_e of a regular expression $e \in \mathcal{R}$ recursively as the smallest set generated by these rules:

$$\frac{a \in \Sigma}{1 \nabla_a a} \qquad \frac{\ell \ \nabla_{e_0} \ r}{\ell \ \nabla_{e_0 + e_1} \ r} \qquad \frac{\ell \ \nabla_{e_1} \ r}{\ell \ \nabla_{e_0 + e_1} \ r}$$

$$\frac{\ell \ \nabla_{e_0} \ r}{\ell \ \nabla_{e_0 \cdot e_1} \ r \cdot e_1} \qquad \frac{\ell \ \nabla_{e_1} \ r}{e_0 \cdot \ell \ \nabla_{e_0 \cdot e_1} \ r} \qquad \frac{\ell \ \nabla_e \ r}{e^* \cdot \ell \ \nabla_{e^*} \ r \cdot e^*}$$

We slightly modified the definition by omitting two rules, namely

$$\frac{a \in \Sigma}{a \nabla_a 1} \qquad \frac{\ell_0 \nabla_{e_0} r_0 \quad \ell_1 \nabla_{e_1} r_1}{\ell_0 \| \ell_1 \nabla_{e_0 \| e_1} r_0 \| r_1}$$

The omission of the left rule is because we construct lasso expressions based on these rules and so we want to avoid 1 on the right hand side (which is the side corresponding to the loop expressions). The second rule is omitted because in [18, Definition 4.5] the authors are working with concurrent Kleene algebras which have a parallel operator $\|$ which we do not consider in this context.

To get a better sense of what the sequential splitting accomplishes, we first state some of its properties which can also be found in [18] (again modified to our setting). It is important to point out that our modifications do not have a major impact on this lemma as concurrent Kleene algebras extend Kleene algebras.

Lemma 7.46 ([18]) *Let $e \in \mathcal{R}$, then ∇_e has the following properties:*

1. *$|\nabla_e|$ is finite (i.e. it contains only finitely many pairs),*
2. *if $u \cdot v \in [\![e]\!]$ with $v \neq \epsilon$, then there are $e_0, e_1 \in \mathcal{R}$ such that*
   - *$e_0 \nabla_e e_1$,*
   - *$e_0 \cdot e_1 \leqq e$,*
   - *$u \in [\![e_0]\!]$ and $v \in [\![e_1]\!]$.*

Intuitively, if a word $u \cdot v \in [\![e]\!]$, then we can find a splitting $e_0 \ \nabla_e \ e_1$ of e such that $u \in [\![e_0]\!], v \in [\![e_1]\!]$ and $e_0 \cdot e_1 \leqq e$.

Example 7.47 Consider the alphabet $\Sigma = \{a, b, c\}$ and the regular expression $e = a^* \cdot b + c$, then

$$\nabla_a = \{(1, a)\} \qquad \nabla_b = \{(1, b)\} \qquad \nabla_c = \{(1, c)\}$$

$$\nabla_{a^*} = \{(1 \cdot a^*, a^* \cdot a)\} \qquad \nabla_{a^* \cdot b} = \{(1 \cdot a^*, a^* \cdot a \cdot b), (a^*, b)\}$$

$$\nabla_{a^* \cdot b + c} = \{(1, c), (1 \cdot a^*, a^* \cdot a \cdot b), (a^*, b)\}$$

Now we may take any word $u \cdot v \in [\![e]\!]$ and the lemma says that there is a pair $e_0 \nabla_e e_1$ with $u \in [\![e_0]\!]$ and $v \in [\![e_1]\!]$.

Given an ω-regular expression κ, we show how to obtain a lasso expression m from κ and then move on to show that m covers κ.

Definition 7.48 Let $h : \mathcal{R}_\omega \to \mathcal{L}$ be given by

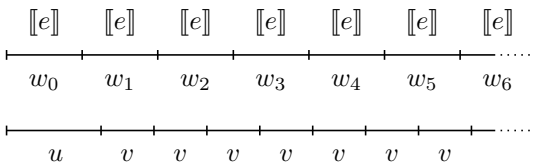
$$h(0_\omega) = (0, 0) \qquad h(e^\omega) = \sum_{e_0 \nabla_e e_1} (e^* \cdot e_0, e_1 \cdot e^* \cdot e_0)$$
$$h(f \otimes \kappa) = f \cdot h(\kappa) \qquad h(\kappa + \lambda) = h(\kappa) + h(\lambda)$$

Proposition 7.49 *Let $\kappa \in \mathcal{R}_\omega$, then $h(\kappa)$ covers κ.*

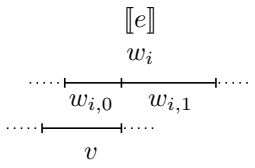
Proof. We show the claim by induction on the construction of ω-regular expressions. For $\kappa = 0_\omega$ it is easy to see that $(0, 0)$ covers κ. Next consider $\kappa = e^\omega$. Let $uv^\omega \in [\![\kappa]\!]_\omega$, we have to show that there is some $(u', v') \sim_\gamma \mathrm{nf}(uv^\omega)$ such that $(u', v') \in [\![h(\kappa)]\!]_\ell$. We have that $[\![e^\omega]\!]_\omega = [\![e]\!]^\omega$, so $uv^\omega \in [\![e]\!]^\omega$. This means that there exist $\{w_i\}_{i<\omega}$ with $w_i \in [\![e]\!]$ such that

$$uv^\omega = w_0 \cdot w_1 \cdot w_2 \cdot w_3 \cdot \ldots.$$

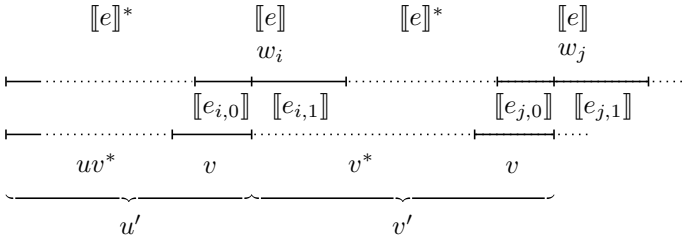
We can visualise this as



We are in particular interested in the positions where a v interval overlaps the start of an $[\![e]\!]$ interval, given by some w_i. Whenever this happens, v gives rise to a sequential splitting of w_i as $w_{i,0} \cdot w_{i,1}$ as seen here:



So every time we have this type of overlap we get a sequential splitting of e, say into $e_{0,i}, e_{1,i}$. As ∇_e contains only finitely many such sequential splittings, there are two indices i, j where $e_{0,i} = e_{0,j}$ and $e_{1,i} = e_{1,j}$. This is visualised below:



Let i be an index as above, for which there is some j where the splitting at i and j are the same. It is easy to see that there are infinitely many such indices giving rise to the same splitting. Moreover, we have that

$$uv^\omega \in [\![e^* \cdot e_{i,0} \cdot (e_{i,1} \cdot e^* \cdot e_{i,0})^\omega]\!]_\omega,$$

which can be seen in the above visualisation, bearing in mind that $e_{i,0} = e_{j,0}$ and $e_{i,1} = e_{j,1}$. From this we can see that there is a lasso $(u', v') \sim_\gamma \mathrm{nf}(uv^\omega)$ such that

$$(u', v') \in [\![(e^* \cdot e_{i,0}, e_{i,1} \cdot e^* \cdot e_{i,0})]\!]_\ell \subseteq [\![h(e^\omega)]\!]_\ell,$$

as $e_{i,0} \nabla_e e_{i,1}$. Showing the second requirement is easier. Let $(u, v) \in [\![h(e^\omega)]\!]_\omega$, we have to show that $uv^\omega \in [\![e^\omega]\!]_\omega$. By our definition of h, there are e_0, e_1 with $e_0 \nabla_e e_1$ such that

$$u \in [\![e^* \cdot e_0]\!] \qquad\qquad v \in [\![e_1 \cdot e^* \cdot e_0]\!]$$

and as

$$v^\omega \in [\![e_1 \cdot e^* \cdot e_0]\!]^\omega = [\![(e_1 \cdot e^* \cdot e_0)^\omega]\!]_\omega \subseteq [\![e_1 \cdot e^\omega]\!]_\omega,$$

we have that

$$uv^\omega \in [\![e^* \cdot e_0 \cdot e_1 \cdot e^\omega]\!]_\omega \subseteq [\![e^\omega]\!]_\omega.$$

The inclusion follows as $e_0 \cdot e_1 \leqq e$.

The case for $\kappa_1 + \kappa_2$ follows as h distributes over addition, so given $uv^\omega \in [\![\kappa_1 + \kappa_2]\!]_\omega$, we have that $uv^\omega \in [\![\kappa_i]\!]_\omega$ for some i, then we can use the induction hypothesis to get that there is some $(u', v') \sim_\gamma \mathrm{nf}(uv^\omega)$ such that $(u', v') \in [\![h(\kappa_i)]\!]_\ell$. Evidently, we then also have $(u', v') \in [\![h(\kappa)]\!]_\ell$. On the other hand, given some $(u, v) \in [\![h(\kappa)]\!]_\ell$. As $h(\kappa_1 + \kappa_2) = h(\kappa_1) + h(\kappa_2)$, we have $(u, v) \in [\![h(\kappa_i)]\!]_\ell$ so $uv^\omega \in [\![\kappa_i]\!]_\omega$ and $uv^\omega \in [\![\kappa]\!]_\omega$.

For the remaining case, we consider an ω-regular expression of the shape $f \otimes \kappa$. Let $uv^\omega \in [\![f \otimes \kappa]\!]_\omega$. Then we can find u'_0, u'_1 and v' such that $u'_0 \in [\![f]\!]$, $u'_1 v'^\omega \in [\![\kappa]\!]_\omega$ and $uv^\omega = u'_0 u'_1 v'^\omega$. So there is a lasso $(u'', v'') \sim_\gamma \mathrm{nf}(u'_1 v'^\omega)$ with $(u'', v'') \in [\![h(\kappa)]\!]_\ell$ by the induction hypothesis. Then

$$(u'_0 u'', v'') \in [\![f \cdot h(\kappa)]\!]_\ell = [\![h(f \otimes \kappa)]\!]_\ell.$$

Finally, let $(u, v) \in [\![h(f \otimes \kappa)]\!]$. Then there is $u_0 \in [\![f]\!]$ and $(u_1, v) \in [\![h(\kappa)]\!]_\ell$ such that $u_0 u_1 = u$. By the induction hypothesis $u_1 v^\omega \in [\![\kappa]\!]_\omega$. So we have $u_0 u_1 v^\omega \in [\![f \otimes \kappa]\!]_\omega$. □

We now have a way of getting a covering for an arbitrary ω-regular expression. We outline the next steps and provide some more visualisations to briefly explain how we proceed. For this, we need the following definition:

Definition 7.50 We define a reverse map on regular expressions, $(-)^r : \mathcal{R} \to \mathcal{R}$ by

$$0^r = 0 \quad 1^r = 1 \quad a^r = a \quad (e + f)^r = e^r + f^r \quad (e \cdot f)^r = f^r \cdot e^r \quad (e^*)^r = (e^r)^*$$

We call e^r the *reverse expression* of e.

For an ω-regular expression κ, we want to find a regular expression e such that $[\![e]\!] = \dot{\sqrt[\omega]{[\![\kappa]\!]_\omega}}$. By Proposition 7.39, we know that it is sufficient to find a regular expression e' such that

$$[\![e']\!]^{\dot{\omega}} \subseteq [\![\kappa]\!]_\omega \qquad \text{and} \qquad \forall u \in \Sigma^+ : (u^\omega \in [\![\kappa]\!]_\omega \implies \exists k \geq 1 : u^k \in [\![e']\!]).$$

We lay out how to obtain e' from $h(\kappa)$. We don't focus on the second requirement of Proposition 7.39 for now, but give more details on that later on.

Let $h(\kappa) = \sum_{i=1}^{n}(f_i, g_i)$ and assume that every lasso $(u, v) \in [\![h(\kappa)]\!]_\ell$ is in γ_1 normal form. Let $M = \{(f_i, g_i) \mid \epsilon \in [\![f_i]\!]\}$ and let

$$e' = \sum_{(f,g)\in M} g.$$

The claim is that e' satisfies the first requirement of Proposition 7.39. This is not hard to see: if $u^\omega \in [\![e']\!]^{\dot{\omega}}$, then we can find $v \in [\![e']\!]$ with $v^\omega = u^\omega$, so $(\epsilon, v) \in [\![h(\kappa)]\!]_\ell$ and hence $u^\omega = v^\omega \in [\![\kappa]\!]_\omega$ as $h(\kappa)$ covers κ.

Of course, we assumed all the lassos in $[\![h(\kappa)]\!]_\ell$ to be γ_1-normalised which is not the case in general. However, we show that it is possible to γ_1-normalise all the lassos in $[\![h(\kappa)]\!]$. This is done via automata and derivatives.

To better understand the DFA construction we propose, we first have a closer look at γ_1-normalisation. Consider the lasso (uab, vab). Normalising it w.r.t. γ_1 gives

$$(uab, vab) \to_{\gamma_1} (ua, bva) \to_{\gamma_1} (u, abv).$$

This process can be divided into two independent steps. We first take the derivative on the right of both the spoke and the loop part (in this case first w.r.t. b, then w.r.t. a).

$$(uab, vab) \xrightarrow{b} (ua, va) \xrightarrow{a} (u, v).$$

The second step consists of adding ab to the start of the loop part.

$$(u, v) \longrightarrow (u, abv).$$

This two-step approach can be visualised as

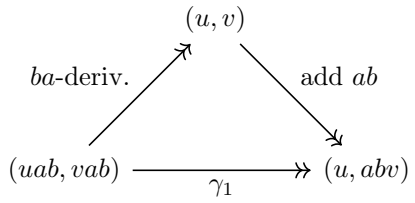


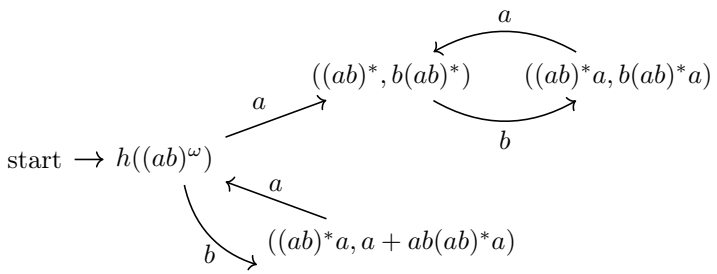
Figure 3: γ_1-normalisation in 2 steps

The states of our DFA consist of lasso expressions. The transition map is a derivative on the right of both the spoke and lasso part of each basic lasso expression. For the second step of the process, we have to add something back to the loop part of each basic lasso expression. This is in general not a word but rather a regular expression. The following example clarifies the idea.

Example 7.51 Let $\Sigma = \{a, b\}$ and consider the ω-regular expression $(ab)^\omega$. The lasso expression $h((ab)^\omega)$ is equivalent to

$$((ab)^*, ab(ab)^*) + ((ab)^*a, b(ab)^*a).$$

If we use lasso expressions as states and take the derivative on the right for both the spoke and loop part[2], we end up with a DFA of the following shape:



The states of particular interest are those which have a basic lasso expression whose spoke part contains the empty word, i.e. $h((ab)^\omega)$ and $((ab)^*, b(ab)^*)$. We are interested particularly in those lasso expressions because they contain all the lassos which are γ_1-normalised. Let's look at $((ab)^*, b(ab)^*)$. In order to get to this state from the initial state, we had to take the derivative w.r.t. a, so according to the second step (Figure 3) we would have to add a to the beginning of the loop expression. However, this is not the only way to end up at $((ab)^*, b(ab)^*)$, we could also first take the derivative w.r.t. aba. What we are hinting at, of course, is that instead of just adding a word, we should add a regular expression. Note that this regular expression can be found by making $((ab)^*, b(ab)^*)$ accepting and finding the regular expression of the DFA we obtain. If we do so, we find $a(ba)^*$. Lastly, we want to point out that we have to add the *reverse expression*. Looking again at Figure 3, we took the ba-derivative but then had to add ab! Adding the reverse expression to the loop part gives $((ab)^*, (ab)^*ab(ab)^*)$.

For $h((ab)^\omega)$, we are only interested in the first term (as the second term does not have ϵ in its spoke expression). Applying the same reasoning as above, we add $(ab)^*$ to obtain $((ab)^*, (ab)^*ab(ab)^*)$.

Finally, taking only the loop expressions of the two basic lasso expressions gives

$$(ab)^*ab(ab)^* + (ab)^*ab(ab)^* \equiv (ab)^+.$$

Remark 7.52 One may at this stage also verify that in fact $[\![(ab)^+]\!]$ satisfies the requirements in Proposition 7.39 with respect to $(ab)^\omega$. In fact it is the case that

$$\overset{\bullet}{\sqrt[\omega]{[\![(ab)^\omega]\!]_\omega}} = [\![(ab)^+]\!].$$

We now turn to the more technical side. We can define the right derivative we need just in terms of the reverse map for regular expressions and the Brzozowski derivative. However, we think it is clearer to spell out the definition in more detail. We define two right derivatives, one for the spoke and one for the loop part of basic lasso expressions. The reason for this is that we want to avoid the empty word in the loop expressions.

[2]for taking derivatives on the loop part, we have to use $d(a, a) = 0$ so we do not obtain the empty word

Definition 7.53 For a regular expression $e \in \mathcal{R}$, we define the *right-spoke derivative* $d_r^s : \mathcal{R} \times \Sigma \to \Sigma$ as

$$\begin{array}{lll} d_r^s(0,a) = 0 & d_r^s(e+f,a) = d_r^s(e,a) + d_r^s(f,a) & d_r^s(b,a) = [b=a] \\ d_r^s(1,a) = 0 & d_r^s(e \cdot f,a) = e \cdot d_r^s(f,a) + d_r^s(e,a) \cdot [f \in N] & d_r^s(e^*,a) = e^* \cdot d_r^s(e,a) \end{array}$$

The *right-loop derivative* d_r^l is defined analogously, with the exception that $d_r^l(b,a) = 0$.

Remark 7.54 We will not prove it, but an equivalent definition for d_r^s is

$$d_r^s(e,a) = d(e^r,a)^r.$$

We combine d_r^s and d_r^l into a derivative for lasso expressions.

Definition 7.55 We define $d_c : \mathcal{L} \times \Sigma \to \mathcal{L}$ as follows. Let $m = \sum_{i=0}^{n}(e_i, f_i)$ be a lasso expression, then

$$d_c(m,a) = d_c\left(\sum_i (e_i,f_i), a\right) = \sum_i \left(d_r^s(e_i,a), d_r^l(f_i,a)\right).$$

Definition 7.56 For a lasso expression m, we call $\mathbb{C}(m) = (\mathcal{L}, m, d_c)$ the *coherence automaton at* m. A lasso expression $n = \sum_{i=0}^{n}(e_i,f_i)$ is *accepting* for $\mathbb{C}(m)$ if it is reachable from m (i.e. there is a path from m to n) and if there is some i such that $\epsilon \in [\![e_i]\!]$ and $[\![f_i]\!] \neq \emptyset$. We denote by $\mathrm{Acc}(\mathbb{C}(m))$ the set of acceptors for $\mathbb{C}(m)$, and let

$$N(n) = \{(e_i,f_i) \mid \epsilon \in [\![e_i]\!], [\![f_i]\!] \neq \emptyset\}.$$

Definition 7.57 Given a lasso expression m and an acceptor n for $\mathbb{C}(m)$, we obtain a DFA

$$\mathbb{C}(m,n) = (\mathcal{L}, m, d_c, \{n\}).$$

Let $\mathcal{R}(\mathbb{C}(m,n))$ be a regular expression corresponding to the regular language accepted by $\mathbb{C}(m,n)$.

Remark 7.58 As with the Brzozowski construction we presented in the previous subsections, the automaton $\mathbb{C}(m,n)$ might not be finite. However, as the derivatives are essentially just Brzozowski derivatives and we are looking at pairs of regular expressions, it should be evident that we can apply the same strategies as before (i.e. define a congruence) to obtain a finite DFA. For clarity and presentation, we will not retrace these steps.

Definition 7.59 Given a lasso expression m, we define the *coherence language* of m as follows:

$$\mathrm{Coh}(m) = \sum_{n \in \mathrm{Acc}(\mathbb{C}(m))} \sum_{(e,f) \in N(n)} \mathcal{R}(\mathbb{C}(m,n))^r \cdot f.$$

At this stage it is a good idea to refer back to Example 7.51 and verify that this corresponds exactly to the approach we outlined. The first sum corresponds to all the

lasso expressions who contain a basic lasso expression whose spoke expression contains the empty word (i.e. all the lasso expressions we are interested in). The second sum considers all the basic lasso expressions whose spoke expression contains the empty word and finally, instead of just taking the loop expression, we also add the reverse of the regular language by which we can go from the initial state to this lasso expression.

We promised earlier to also go into the second condition of Proposition 7.39 which states

$$\forall u \in \Sigma^+ : (u^\omega \in \mathscr{L} \implies \exists k \geq 1 : u^k \in L),$$

for a regular language L and an ω-regular language $\mathscr{L}$. We claim that $[\![\mathrm{Coh}(h(\kappa))]\!]$ satisfies this condition with respect to $[\![\kappa]\!]_\omega$. The reason for this has to do with how we defined h. In short, looking for instance at $h(e^\omega)$, the regular expressions in the loop part of each basic lasso expression are of the shape $e_1 e^* e_0$ for $e_0 \nabla_e e_1$. Intuitively, this expression is able to account for any $u \in \Sigma^+$ as in the requirement. This is made more precise in the following proposition.

Proposition 7.60 *Let κ be an ω-regular language, then*

$$(u, v) \in [\![h(\kappa)]\!]_\ell \implies \forall k \geq 1 : (u, v^k) \in [\![h(\kappa)]\!]_\ell.$$

Proof. We show this by induction on κ. The case $\kappa = 0_\omega$ is trivial. For $\kappa = e^\omega$, we defined

$$h(e^\omega) = \sum_{(e_0, e_1) \in \nabla_e} (e^* \cdot e_0, e_1 \cdot e^* \cdot e_0)\,.$$

So $(u, v) \in [\![h(e^\omega)]\!]_\ell$ means there is some $(e_0, e_1) \in \nabla_e$ such that

$$u \in [\![e^* \cdot e_0]\!] \qquad\qquad v \in [\![e_1 \cdot e^* \cdot e_0]\!]$$

It is clear that $v^k \in [\![e_1 \cdot e^* \cdot e_0]\!]$ for all $k \geq 1$ as

$$\underbrace{(e_1 \cdot e^* \cdot e_0) \cdot \ldots \cdot (e_1 \cdot e^* \cdot e_0)}_{k \text{ times}} \leqq e_1 \cdot e^* \cdot e_0$$

(bearing in mind that $e_0 \cdot e_1 \leqq e$).

The case $\kappa + \lambda$ follows as h distributes over $+$. Finally, for $f \otimes \kappa$ it follows as the multiplication only affects the spoke expressions. □

The final step is the main proposition, which states that for an ω-regular expression κ, the regular expression $\mathrm{Coh}(h(\kappa))$ satisfies the requirements of Proposition 7.39.

To make the proof easier to follow, we introduce a technical lemma.

Lemma 7.61 *Let $u \in \Sigma^*, v \in \Sigma^+$ and $m = h(\kappa)$ for some ω-regular expression κ. The following are equivalent:*

1. $(u, vu) \in [\![m]\!]_\ell$,
2. $(\epsilon, v) \in [\![d_c(m, u^r)]\!]_\ell$,
3. $uv \in [\![Coh(m)]\!]$ *for some* $n \in Acc(\mathbb{C}(m)), (e, f) \in N(n)$ *with* $u \in [\![\mathscr{R}(\mathbb{C}(m, n))^r]\!]$ *and* $v \in [\![f]\!]$.

Proof. (1.) $\iff$ (2.) follows from the definition of d_c. (2.) $\implies$ (3.) Take $n = d_c(m, u^r)$. As $(\epsilon, v) \in [\![n]\!]_\ell$, $n \in \mathrm{Acc}(\mathbb{C}(m))$ and there is some $(e, f) \in N(n)$ with $v \in [\![f]\!]$. Furthermore as n was reached after reading u^r, we have that $u \in [\![\mathcal{R}(\mathbb{C}(m, n))^r]\!]$, so $uv \in [\![\mathrm{Coh}(m)]\!]$. (3.) $\implies$ (2.) If $u \in [\![\mathcal{R}(\mathbb{C}(m, n))^r]\!]$, then $n = d_c(m, u^r)$. As $(e, f) \in N(n)$ and $v \in [\![f]\!]$, we have that $(\epsilon, v) \in [\![n]\!]_\ell$ so

$$(\epsilon, v) \in [\![d_c(m, u^r)]\!]_\ell.$$

□

Proposition 7.62 *Let κ be an ω-regular expression, and $e = Coh(h(\kappa))$, then*

1. $[\![e]\!]^{\dot\omega} \subseteq [\![\kappa]\!]_\omega$,
2. $\forall u \in \Sigma^+ : \big(u^\omega \in [\![\kappa]\!]_\omega \implies \exists k \geq 1 : u^k \in [\![e]\!]\big)$.

Proof. We start with the second condition. Let $u \in \Sigma^+$ and let $v \in \Sigma^+, k \geq 1$ be such that $v^k = u$ and there exists no $v' \in \Sigma^+, k' \geq 1$ such that $v = v'^{k'}$. Then $\mathrm{nf}(u^\omega) = (\epsilon, v)$ and one can show inductively that every lasso which is γ-equivalent to (ϵ, v) must be of the form $(v^{k_1} \cdot v_0, v_1 \cdot v^{k_2} \cdot v_0)$ where $v_0 \cdot v_1 = v^{k_3}$ for some k_1, k_2, k_3. As $h(\kappa)$ covers κ and $u^\omega \in [\![\kappa]\!]_\omega$, we can find a lasso of the shape $(v^{k_1} \cdot v_0, v_1 \cdot v^{k_2} \cdot v_0) \in [\![h(\kappa)]\!]_\ell$. By proposition 7.60,

$$(v^{k_1} \cdot v_0, (v_1 \cdot v^{k_2} \cdot v_0)^{k_4}) \in [\![h(\kappa)]\!]_\ell,$$

and we can choose k_4 to be large enough such that $(v^{k_1} \cdot v_0, (v_1 \cdot v^{k_2} \cdot v_0)^{k_4}) \twoheadrightarrow_{\gamma_1} (\epsilon, u^{k_5})$ for some $k_5 \geq 1$. The rest follows by Lemma 7.61.

For the first condition, let $m = h(\kappa)$. We have to show that

$$[\![\mathrm{Coh}(m)]\!]^{\dot\omega} \subseteq [\![\kappa]\!]_\omega.$$

Let $\alpha \in [\![\mathrm{Coh}(m)]\!]^\omega$, then there exists $w \in [\![\mathrm{Coh}(m)]\!]$ such that $\alpha = w^\omega$. As $w \in [\![\mathrm{Coh}(m)]\!]$, there is some $n \in \mathrm{Acc}(\mathbb{C}(m))$ and some (e, f) occurring in n such that

$$w \in [\![\mathcal{R}(\mathbb{C}(m, n))^r \cdot f]\!].$$

Hence we can find u, v such that $w = uv$, $v \in [\![f]\!]$ and $u \in [\![\mathcal{R}(\mathbb{C}(m, n))^r]\!]$. From which, by Lemma 7.61, it follows that

$$(u, vu) \in [\![m]\!]_\ell.$$

As m covers κ, we have that $u(vu)^\omega = (uv)^\omega = w^\omega = \alpha \in [\![\kappa]\!]_\omega$. □

Corollary 7.63 *Let κ be an ω-regular expression, then*

$$\sqrt{[\![Coh(h(\kappa))]\!]} = \dot{\sqrt[\omega]{[\![\kappa]\!]_\omega}}.$$

Proof. This is a direct consequence of Proposition 7.39 and Proposition 7.62. □

Minimisation of Ω-automata

A distinguishing feature of Ω-automata is that there is a clear path to minimisation (reducing the number of states), this is not a given for many other types of ω-automata. In the previous section, we hinted at the fact that construction methods which convert Muller and parity automata to Ω-automata result in Ω-automata with a lot of states. To some extent this is not a problem, as any Ω-automaton can be minimised to a smallest Ω-automaton accepting the same ω-regular language.

Several papers discuss minimisation of coalgebras in a more abstract setting ([1, 7, 5, 6]). Specifically for Ω-automata, a partition refinement approach is discussed in [12].

This section looks at two other minimisation algorithms for Ω-automata. The first one is the Brzozowski minimisation algorithm which has been studied extensively for DFAs and several other types of automata in a categorical framework ([7]). The other minimisation procedure we look at is minimisation via a dual equivalence, which has also been studied in conjunction with Brzozowski minimisation to account for a wider phenomenon ([6, 5]).

8.1 Minimisation via Adjunction

Brzozowski's minimisation procedure is a fascinating algorithm to minimise arbitrary DFAs. It works by applying the following two steps twice:

1. apply the reverse powerset construction
2. take the reachable part of the obtained DFA

After having done that one obtains a minimal DFA which accepts the same regular language as the DFA one started out with. In [7], the authors give the intuition behind Brzozowski's minimisation procedure using algebra and coalgebra. Throughout this section, we trace the steps outlined in op. cit., adapting it to our environment. To some extent, this subsection can be seen as an instantiation of [7]. However, we want to point out that although the idea remains the same, the actual construction is more involved.

We start by showing that lasso automata have underlying coalgebraic and algebraic structure. This allows us to define a category **LA** of lasso automata. For a given lasso language L we define $\mathbf{LA}(L)$ as the full subcategory of **LA** consisting of those lasso automata accepting L and show that this category has both an initial and a final object.

Using results from the section on the category $\mathbf{Set}^2$, we define a contravariant powerset functor on $\mathbf{Set}^2$ and show that it is self-adjoint. Next we lift this self-adjunction to the level of $\mathbf{LA}(L)$, en route obtaining a reverse powerset construction for lasso automata.

Finally, we show that the Brzozowski minimisation procedure can also be applied to lasso automata, and hence to Ω-automata.

8.1.1 Initial $\mathfrak{A}$-Algebra, Final Ω-Coalgebra and the Category of Lasso Automata

A DFA can be seen as a pointed $(-)^\Sigma \times 2$-coalgebra or equivalently as a copointed $1 + (-) \times \Sigma$-algebra, the initial state being characteristically algebraic and the final states coalgebraic ([7]). Interestingly enough, the transition map can be seen as both through currying and uncurrying:

$$\delta : Q \to Q^\Sigma \text{ (coalgebraic)} \qquad \delta^\flat : Q \times \Sigma \to Q \text{ (algebraic)}$$

We already know that lasso automata are pointed Ω-coalgebras, so we provide an endofunctor on $\mathbf{Set}^2$ through which we can see them also as copointed algebras. With this we define the category of lasso automata $\mathbf{LA}$ and for a lasso language L also the full subcategory $\mathbf{LA}(L)$ of $\mathbf{LA}$, whose objects are all the lasso automata accepting L. The category $\mathbf{LA}(L)$ has an initial and a final object.

Definition 8.1 We define the functor $\mathfrak{A} : \mathbf{Set}^2 \to \mathbf{Set}^2$ given by

1. objects: $\mathfrak{A}(A, B) = (1 + A \times \Sigma, A \times \Sigma + B \times \Sigma)$,
2. morphisms: $\mathfrak{A}(f, g) = (! + f \times \mathrm{id}_\Sigma, f \times \mathrm{id}_\Sigma + g \times \mathrm{id}_\Sigma)$.

Just like the category of Ω-coalgebras has a final coalgebra, the category of $\mathfrak{A}$-algebras has an initial algebra. We first give its definition and then show that it is initial.

Definition 8.2 We define the $\mathfrak{A}$-algebra $\mathbb{S} = \big((\Sigma^*, \Sigma^{*+}), (\mathfrak{i} + \mathfrak{a}, \mathfrak{b} + \mathfrak{c})\big)$ where:

$$\begin{aligned}
&\mathfrak{i} : 1 \to \Sigma^* && \text{with} \quad \mathfrak{i}(\star) = \epsilon \\
&\mathfrak{a} : \Sigma^* \times \Sigma \to \Sigma^* && \text{with} \quad \mathfrak{a}(w, a) = wa \\
&\mathfrak{b} : \Sigma^* \times \Sigma \to \Sigma^{*+} && \text{with} \quad \mathfrak{b}(w, a) = (w, a) \\
&\mathfrak{c} : \Sigma^{*+} \times \Sigma \to \Sigma^{*+} && \text{with} \quad \mathfrak{c}((w, v), a) = (w, va)
\end{aligned}$$

Proposition 8.3 *The $\mathfrak{A}$-algebra $\mathbb{S}$ is the initial $\mathfrak{A}$-algebra. Given some $\mathfrak{A}$-algebra $\mathbb{A} = ((X, Y), (i + \mu, \nu + \pi))$ the unique morphism $(f_0, f_1) : \mathbb{S} \to \mathbb{A}$ is given by*

$$\begin{aligned}
f_0(\epsilon) &:= i(\star) & f_1(w, a) &:= \nu(f_0(w), a) \\
f_0(wa) &:= \mu(f_0(w), a) & f_1(w, va) &:= \pi(f_1(w, v), a)
\end{aligned}$$

Proof. We first show that (f_0, f_1) is an $\mathfrak{A}$-algebra morphism and then we show that it is unique.

For the first part, we have to check the following equations:

$$\begin{aligned}
f_0(\mathfrak{i}(\star)) &= \iota(!(\star)) & f_0(\mathfrak{a}(w, a)) &= \mu(f_0(w), a) \\
f_1(\mathfrak{b}(w, a)) &= \nu(f_0(w), a) & f_1(\mathfrak{c}((w, v), a)) &= \pi(f_1(w, v), a)
\end{aligned}$$

This is a rather straight-forward task:

$$f_0(\mathfrak{i}(\star)) = f_0(\epsilon) = \iota(\star) = \iota(!(\star)),$$

$$f_0(\mathfrak{a}(w,a)) = f_0(wa) = \mu(f_0(w),a),$$
$$f_1(\mathfrak{b}(w,a)) = f_1(w,a) = \nu(f_0(w),a),$$
$$f_1(\mathfrak{c}((w,v),a)) = f_1(w,va) = \pi(f_1(w,v),a).$$

For uniqueness, assume there is an $\mathfrak{A}$-algebra morphism (g_0,g_1) which hence makes the above equations true. We show that $(g_0,g_1) = (f_0,f_1)$ by induction on the construction of words and lassos. We start with f_0 and g_0. For the base case we have

$$g_0(\epsilon) = g_0(\mathfrak{i}(\star)) = \iota(!(\star)) = f_0(\mathfrak{i}(\star)) = f_0(\epsilon).$$

For the inductive case we consider the word $wa \in \Sigma^+$, then

$$g_0(wa) = g_0(\mathfrak{a}(w,a)) = \mu(g_0(w),a) \overset{\text{(I.H.)}}{=} \mu(f_0(w),a) = f_0(\mathfrak{a}(w,a)) = f_0(wa).$$

This shows that $g_0 = f_0$. Next we show that $f_1 = g_1$. For the base case consider a lasso (w,a), where $a \in \Sigma, w \in \Sigma^*$, then

$$g_1(w,a) = g_1(\mathfrak{b}(w,a)) = \nu(g_0(w),a) \overset{f_0=g_0}{=} \nu(f_0(w),a) = f_1(\mathfrak{b}(w,a)) = f_1(w,a).$$

Finally, for the inductive step we consider the lasso $(w,va) \in \Sigma^{*+}$, then

$$\begin{aligned} g_1(w,va) &= g_1(\mathfrak{c}((w,v),a)) \\ &= \pi(g_1(w,v),a) \\ &= \pi(f_1(w,v),a) && \text{(I.H.)} \\ &= f_1(\mathfrak{c}((w,v),a)) \\ &= f_1(w,va), \end{aligned}$$

which concludes the proof. □

The following is the final Ω-coalgebra whose definition and proof of finality can be found in [12].

Definition 8.4 The final Ω-coalgebra is given by the structure $\mathbb{Z} = ((2^{\Sigma^{*+}}, 2^{\Sigma^*}), (\langle \alpha, \beta \rangle, \langle \omega, \gamma \rangle))$ with

$$\begin{aligned} &\alpha : 2^{\Sigma^{*+}} \to (2^{\Sigma^{*+}})^\Sigma && \text{with} \quad \alpha(L)(a) = \{(u,v) \in \Sigma^{*+} \mid (au,v) \in L\} \\ &\beta : 2^{\Sigma^{*+}} \to (2^{\Sigma^*})^\Sigma && \text{with} \quad \beta(L)(a) = \{u \in \Sigma^* \mid (\epsilon, au) \in L\} \\ &\omega : 2^{\Sigma^*} \to 2 && \text{with} \quad \omega(K) = [\epsilon \in K] \\ &\gamma : 2^{\Sigma^*} \to (2^{\Sigma^*})^\Sigma && \text{with} \quad \gamma(K)(a) = \{u \in \Sigma^* \mid au \in K\} \end{aligned}$$

and the final map (h_0,h_1) from an Ω-coalgebra $\mathbb{A} = ((X,Y),(\langle \rho,\sigma \rangle, \langle \xi,\chi \rangle))$ to $\mathbb{Z}$ is given by

$$h_0(x) = \text{Lassos}(\mathbb{A},x) \qquad\qquad h_1(y) = L((Y,\xi,\chi),y)$$

We remarked earlier that the transition map of a DFA can be seen as either algebraic or coalgebraic via currying and uncurrying. This is just an example of an adjunction between the functors $(-) \times \Sigma$ and $(-)^{\Sigma}$. We show that the same holds for the transition maps of Ω-coalgebras. As the initial and final states have to be disregarded, we define a version of $\mathfrak{A}$ and Ω which does not include initial nor final states respectively and then show that the functors we obtain are adjoint.

Definition 8.5 We can restrict the functors $\mathfrak{A}, \Omega$ to exclude the specification of initial and final states and define

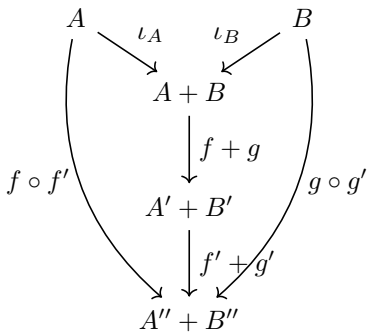
$$\Omega^*(X, Y) = (X^{\Sigma} \times Y^{\Sigma}, Y^{\Sigma}) \qquad \text{and} \qquad \mathfrak{A}^{\dagger}(X, Y) = (X \times \Sigma, X \times \Sigma + Y \times \Sigma).$$

Before we show that $\mathfrak{A}^{\dagger} \dashv \Omega^*$, we define two intermediary functors and show that those are adjoint. This makes our proof easier. In order to define the two functors, we need the following lemma which is used to show that our functors are indeed functors.

Lemma 8.6 *Let $f : A \to A', g : B \to B', f' : A' \to A'', g' : B' \to B''$, then*

1. $(f' + g') \circ (f + g) = (f' \circ f) + (g' \circ g)$
2. $(f' \times g') \circ (f \times g) = (f' \circ f) \times (g' \circ g)$

Proof. Consider the following diagram



We claim that $f \circ f' : A \to A''$ and $g \circ g' : B \to B''$ are the unique morphisms which make this diagram commute. We shall show this for $f \circ f'$, the other follows via symmetric reasoning. Let $a \in A$, then

$$(f' + g') \circ (f + g) \circ \iota_A(a) = (f' + g') \circ \iota_{A'} \circ f(a) = \iota_{A''} \circ f' \circ f(a) = f' \circ f(a).$$

The proof strategy for (2.) is similar. Note that $f \times g$ is defined as follows:

$$(f \times g)(a, b) = (f(a), g(b)).$$

Then

$$\begin{aligned}(f' \times g') \circ (f \times g)(a,b) &= (f' \times g')(f(a), g(b)) \\ &= (f'(f(a)), g'(g(b))) \\ &= ((f' \circ f)(a), (g' \circ g)(b)) \\ &= (f' \circ f) \times (g' \circ g)(a,b).\end{aligned}$$

This gives the required result. □

Definition 8.7 Let C_R and P_L be the following two functors:

– $C_R : \mathrm{Set}^2 \to \mathrm{Set}^2$

1. $C_R(A,B) = (A, A+B)$
2. $C_R(f,g) = (f, f+g)$

– $P_L : \mathrm{Set}^2 \to \mathrm{Set}^2$

1. $P_L(A,B) = (A \times B, B)$
2. $P_L(f,g) = (f \times g, g)$

Proof. We show that C_R and P_L are both functors, starting with C_R. We have to show that C_R preserves composition and the identity morphisms. Let $(f,g) : (A,B) \to (A',B')$ and $(f',g') : (A',B') \to (A'',B'')$, then

$$\begin{aligned}C_R(f',g') \circ C_R(f,g) &= (f', f'+g') \circ (f, f+g) \\ &= (f' \circ f, (f'+g') \circ (f+g)) \\ &= (f' \circ f, f' \circ f + g' \circ g) && \text{(Lem. 8.6)} \\ &= C_R(f' \circ f, g' \circ g) \\ &= C_R((f',g') \circ (f,g)).\end{aligned}$$

As for preservation of the identity, we have that

$$\begin{aligned}C_R(1_{(A,B)}) &= C_R(1_A, 1_B) \\ &= (1_A, 1_A + 1_B) \\ &= 1_{(A,A+B)} \\ &= 1_{C_R(A,B)}.\end{aligned}$$

Hence C_R is a functor. The proof for P_L follows in a similar fashion, again using Lemma 8.6. □

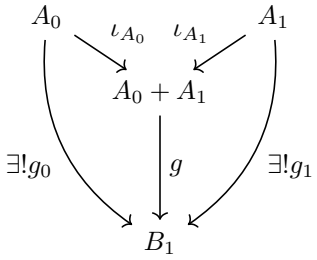
Lemma 8.8 *The functor C_R is left-adjoint to P_L:*

$$C_R \dashv P_L.$$

Proof. We construct a natural isomorphism

$$m_{A,B} : \mathrm{Hom}_{\mathbf{Set}^2}((A_0, A_0+A_1), (B_0, B_1)) \xrightarrow{\cong} \mathrm{Hom}_{\mathbf{Set}^2}((A_0, A_1), (B_0 \times B_1, B_1)).$$

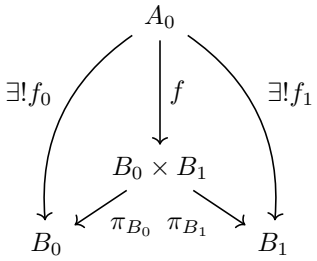
Let $(f,g) : (A_0, A_0+A_1) \to (B_0, B_1)$. By the universal property of the coproduct we can find unique g_0, g_1 such that $g = g_0 + g_1$ and which make the following diagram commute ($g_0 = g_{\restriction A_0}$ and $g_1 = g_{\restriction A_1}$).



Define

$$m_{A,B}(f,g) = m_{A,B}(f, g_0 + g_1) = (f \times g_0, g_1).$$

Now let $(f,g) : (A_0, A_1) \to (B_0 \times B_1, B_1)$. By the universal property of the product, we find unique f_0, f_1 such that $f = f_0 \times f_1$ and the following diagram commutes (take $f_i(a) = \pi_{B_i}(f(a))$).



Now define

$$m^{-1}_{A,B}(f,g) = m^{-1}_{A,B}(f_0 \times f_1, g) = (f_0, f_1 + g).$$

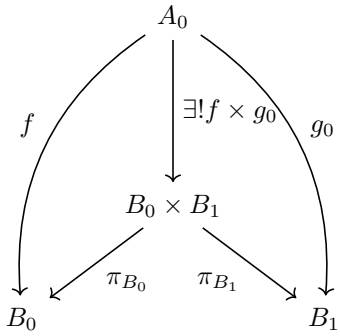
We again use $\overline{(f,g)}$ to denote the transpose of (f,g). Then for $(f,g) : (A_0, A_0 + A_1) \to (B_0, B_1)$ we have

$$\overline{\overline{(f,g)}} = \overline{\overline{(f, g_0 + g_1)}} = \overline{(f \times g_0, g_1)} = (f, g_0 + g_1) = (f,g),$$

and for $(f,g) : (A_0, A_1) \to (B_0 \times B_1, B_1)$ we have

$$\overline{\overline{(f,g)}} = \overline{\overline{(f_0 \times f_1, g)}} = \overline{(f_0, f_1 + g)} = (f_0 \times f_1, g) = (f,g).$$

We give some additional justification for the step $\overline{(f \times g_0, g_1)} = (f, g_0 + g_1)$ (the step $\overline{(f_0, f_1 + g)} = (f_0 \times f_1, g)$ can be justified in a similar manner). It is sufficient to consider the following diagram and notice that $f \times g_0$ is the unique map which makes the diagram commute (universal property of the product $B_0 \times B_1$).



Finally, we also have to show naturality, so let $(f, g) : (A_0, A_1) \to (B_0, B_1)$ and $(f', g') : (A'_0, A'_1) \to (B'_0, B'_1)$. We want to show that the following diagram commutes:

$$\begin{array}{ccc} \mathrm{Hom}_{\mathbf{Set}^2}((A_0, A_0 + A_1), (B_0, B_1)) & \xrightarrow{m_{A,B}} & \mathrm{Hom}_{\mathbf{Set}^2}((A_0, A_1), (B_0 \times B_1, B_1)) \\ \Big\downarrow{\scriptstyle (f', g') \circ (-) \circ C_R(f, g)} & \circlearrowleft & \Big\downarrow{\scriptstyle P_L(f', g') \circ (-) \circ (f, g)} \\ \mathrm{Hom}_{\mathbf{Set}^2}((A'_0, A'_0 + A'_1), (B'_0, B'_1)) & \xrightarrow[m_{A',B'}]{} & \mathrm{Hom}_{\mathbf{Set}^2}((A'_0, A'_1), (B'_0 \times B'_1, B'_1)) \end{array}$$

Let $(i, j) \in \mathrm{Hom}_{\mathrm{Set}^2}((A_0, A_0 + A_1), (B_0, B_1))$ with $j = j_1 + j_2$, then

$$\begin{aligned} P_L(f', g') \circ \overline{(i, j)} \circ (f, g) &= (f' \times g', g') \circ (i \times j_1, j_2) \circ (f, g) \\ &= ((f' \times g') \circ (i \times j_1) \circ f, g' \circ j_2 \circ g) \\ &= (f' \circ i \circ f \times g' \circ j_1 \circ f, g' \circ j_2 \circ g) && \text{(Lem. 8.6)} \\ &= \overline{(f' \circ i \circ f, g' \circ j_1 \circ f + g' \circ j_2 \circ g)} \\ &= \overline{(f' \circ i \circ f, g' \circ (j_1 + j_2) \circ (f + g))} && \text{(Lem. 8.6)} \\ &= \overline{(f' \circ i \circ f, g' \circ j \circ (f + g))} \\ &= \overline{(f', g') \circ (i, j) \circ (f, f + g)} \\ &= \overline{(f', g') \circ (i, j) \circ C_R(f, g)}. \end{aligned}$$

□

Having defined our intermediary functors and showed that they form an adjunction, we move on to show that $\mathfrak{A}^\dagger$ is left-adjoint to Ω^*.

Proposition 8.9 *The functor $\mathfrak{A}^\dagger$ is left-adjoint to Ω^*:*

$$\mathfrak{A}^\dagger \dashv \Omega^*.$$

Proof. The following two maps defined on objects can be extended to functors:

$$\begin{aligned} E &: \mathbf{Set} \to \mathbf{Set} \text{ with } E(A) := A^\Sigma, \\ T &: \mathbf{Set} \to \mathbf{Set} \text{ with } T(A) := A \times \Sigma. \end{aligned}$$

We have that $T \dashv E$ (which is just currying and uncurrying) and by Lemma 3.3 we can lift the adjunction to $\mathbf{Set}^2$, i.e. $T^2 \dashv E^2$ where $T^2(A,B) = (A \times \Sigma, B \times \Sigma)$ and $E^2(A,B) = (A^\Sigma, B^\Sigma)$. It is well-known that we can compose the adjunctions $T^2 \dashv E^2$ and $C_R \dashv P_L$ to obtain

$$C_R \circ T^2 \dashv E^2 \circ P_L$$

(note that this is also well-typed). Moreover, we observe that $E^2 \circ P_L \cong P_L \circ E^2$, as $((A \times B)^\Sigma, B^\Sigma) \cong (A^\Sigma \times B^\Sigma, B^\Sigma)$. Finally, we have that $\Omega^* = P_L \circ E^2$ and $\mathfrak{A}^\dagger = C_R \circ T^2$, hence

$$\mathfrak{A}^\dagger \dashv \Omega^*. \qquad \square$$

This shows that a lasso automaton $\mathbb{A} = (X, Y, i, \rho, \sigma, \xi, F)$ can be seen as both a pointed Ω-coalgebra $((X,Y), (\langle \rho, \sigma \rangle, \langle \xi, \chi_F \rangle))$ with initial state $i(\star) \in X$ and as a copointed $\mathfrak{A}$-algebra $((X,Y), (i + \rho^\flat, \sigma^\flat + \xi^\flat))$ with final states F.

Definition 8.10 A lasso automaton is simultaneously a *pointed Ω-coalgebra* and a *copointed $\mathfrak{A}$-algebra*. We define the category $\mathbf{LA}$ whose objects are lasso automata and whose morphisms are maps which respect both the coalgebra and algebra structure. More formally, given two lasso automata $\mathbb{A} = (X, Y, i, \rho, \sigma, \xi, F)$ and $\mathbb{A}' = (X', Y', i', \rho', \sigma', \xi', F')$, a map $(f, g) : \mathbb{A} \to \mathbb{A}'$ is a lasso automaton morphism if it makes the following diagrams commute:

$$\begin{array}{ccc} \mathfrak{A}(X,Y) & \xrightarrow{\mathfrak{A}(f,g)} & \mathfrak{A}(X',Y') \\ {\scriptstyle ((i+\rho^\flat),\ (\sigma^\flat + \xi^\flat))}\downarrow & \circlearrowleft & \downarrow{\scriptstyle ((i'+\rho'^\flat),\ (\sigma'^\flat + \xi'^\flat))} \\ (X,Y) & \xrightarrow[(f,g)]{} & (X',Y') \end{array} \qquad \begin{array}{ccc} (X,Y) & \xrightarrow{(f,g)} & (X',Y') \\ {\scriptstyle (\langle \rho, \sigma \rangle,\ \langle \xi, \chi_F \rangle)}\downarrow & \circlearrowleft & \downarrow{\scriptstyle (\langle \rho', \sigma' \rangle,\ \langle \xi', \chi_{F'} \rangle)} \\ \Omega(X,Y) & \xrightarrow[\Omega(f,g)]{} & \Omega(X',Y') \end{array}$$

Remark 8.11 We point out that any morphism makes the transition part of the left diagram commute if and only if it also makes the transition part of the right diagram commute.

The category $\mathbf{LA}$ does not have initial nor final objects. However, restricting our attention to the full subcategory of $\mathbf{LA}$ whose objects are given by all the lasso automata which accept a given lasso language L gives us a category which has an initial and final object.

Definition 8.12 Given a lasso language L, we define $\mathbf{LA}(L)$ to be the full-subcategory of $\mathbf{LA}$ consisting of those lasso automata accepting L.

Proposition 8.13 *Given a lasso language L, the category $\mathbf{LA}(L)$ has an initial and a final object.*

Proof. We can turn the initial $\mathfrak{A}$-algebra and the final Ω-coalgebra into the initial and final object in $\mathbf{LA}(L)$ by adding final and initial states respectively.

Let $\chi_L : \Sigma^{*+} \to 2$ be given by $\chi_L(u) = [u \in L]$ and define $i_L : \{\star\} \to 2^{\Sigma^{*+}}$ as $i_L(\star) = L$. We claim that $(\mathbb{S}, \chi_L)$ and $(\mathbb{Z}, i_L)$ are both in $\mathbf{LA}(L)$. To see this, let $(u, v) \in \Sigma^{*+}$, then by Definitions 8.2 and 8.4:

$$\begin{aligned}(u,v) \in \mathrm{Lassos}(\mathbb{S}, \epsilon) &\iff \widetilde{\mathfrak{b} : \mathfrak{c}}(\widehat{\mathfrak{a}}(\epsilon, u), v) \in L \\ &\iff (u,v) \in L, \\ (u,v) \in \mathrm{Lassos}(\mathbb{Z}, L) &\iff \epsilon \in \widetilde{\beta : \gamma}(\widehat{\alpha}(L, u), v) \\ &\iff (u,v) \in L.\end{aligned}$$

To show that they are both the initial and final object respectively, we have to show that the unique morphisms we described earlier are unique lasso automata morphisms. Given some lasso automaton $\mathbb{A} = (X, Y, i, \rho, \sigma, \xi, F)$ accepting L and letting $(f_0, f_1) : \mathbb{S} \to \mathbb{A}$, $(h_0, h_1) : \mathbb{A} \to \mathbb{Z}$, as defined in Proposition 8.3 and Definition 8.4, it suffices to check the following two equations:

$$\begin{aligned}h_0(i(\star)) &= !(i_L(\star)) \\ \mathrm{id}_2(\chi_L(u,v)) &= \chi_F(f_1(u,v))\end{aligned}$$

For the first equation we have that

$$h_0(i(\star)) = h_0(i) = \mathrm{Lassos}(\mathbb{A}, i) = L = i_L(\star) = !(i_L(\star)),$$

and for the second equation, we get

$$id_2(\chi_L(u,v)) = [(u,v) \in L] = \chi_F(\widetilde{\sigma : \xi}(\widehat{\rho}(i,u), v)) = \chi_F(f_1(u,v)).$$

Uniqueness is also clearly given. □

8.1.2 The Two-sorted Contravariant Powerset Functor

In the classical setting, Brzozowski's minimisation procedure exploits a dual adjunction between **Set** and $\mathbf{Set}^{\mathrm{op}}$ given by the contravariant powerset functor which can be lifted to the level of coalgebras ([7]). In our setup we can proceed in the same way, though there are some minor differences.

We first define a two-sorted contravariant powerset functor and show that it is self-adjoint. This is followed up by lifting it to the level of lasso automata. We show that our lifting establishes an adjoint situation between the categories $\mathbf{LA}(L)$ and $\mathbf{LA}^{\mathrm{op}}(L^r)$ (where L^r is the reverse lasso language of L).

With this setup, we can state and prove a Brzozowski type minimisation algorithm for Ω-automata.

Definition 8.14 The *two-sorted contravariant powerset functor* $\breve{\mathscr{P}}^2 \colon \mathbf{Set}^2 \to \mathbf{Set}^{2,\mathrm{op}}$ is given by

1. objects: $\breve{\mathscr{P}}^2\,(A, B) = (2^B, 2^A)$,
2. morphisms: $\breve{\mathscr{P}}^2\,(f, g) = (g^{-1}[-], f^{-1}[-])$.

Just like the ordinary contravariant powerset functor, $\breve{\mathscr{P}}^2$ turns epis into monos and vice-versa which is crucial for the Brzozowski minimisation.

Proposition 8.15 *The morphism (f, g) is mono (resp. epi) if and only if $\breve{\mathscr{P}}^2$ (f, g) is epi (resp. mono).*

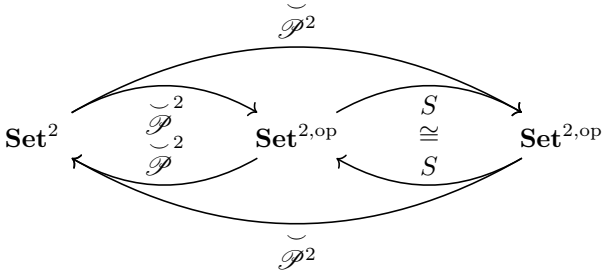
Proof. As a morphism in $\mathbf{Set}^2$ is mono (resp. epi) if and only if it is pointwise mono (resp. epi), it is sufficient to show that f is mono (resp. epi) iff $f^{-1}[-]$ is epi (resp. mono). This is a standard fact so we omit the proof. □

Corollary 8.16 *We can see $\breve{\mathscr{P}}^2$ also as a functor from $\mathbf{Set}^{2,op}$ to $\mathbf{Set}^2$. As such, it is self-adjoint, i.e.*

$$\breve{\mathscr{P}}^2 \dashv \breve{\mathscr{P}}^2.$$

Proof. This follows from two observations

1. $\breve{\mathscr{P}}\colon \mathbf{Set} \to \mathbf{Set}^{\mathrm{op}}$ is self-adjoint and so we can lift the adjunction to $\mathbf{Set}^2$ by Lemma 3.3.
2. $\breve{\mathscr{P}}^2 = S \circ \breve{\mathscr{P}}^2$ where S is the swap functor defined in Definition 3.5.



□

Our next goal is to lift this adjunction to the category $\mathbf{LA}(L)$ for some lasso language L. We first show how to lift $\breve{\mathscr{P}}^2$ to $\mathbf{LA}(L)$. This is done by means of an example. Given some lasso automaton $\mathbb{A}$, we first look at its $\mathfrak{A}$-algebra reduct. So for now, we think of $\mathbb{A}$ as the structure $\mathbb{A} = ((X, Y), (i + \xi^\flat, \sigma^\flat + \rho^\flat))$. Applying $\breve{\mathscr{P}}^2$ to $(X, Y), \mathfrak{A}(X, Y)$ and to the maps $(i + \xi^\flat, \sigma^\flat + \rho^\flat)$ gives

$$\begin{array}{ccc} (X,Y) & & (2^Y, 2^X) \\ \uparrow{\scriptstyle (i+\rho^\flat, \sigma^\flat + \xi^\flat)} & \overset{\breve{\mathscr{P}}^2}{\Longrightarrow} & \downarrow{\scriptstyle ((\xi^\flat + \sigma^\flat)^{-1}, (\rho^\flat + i)^{-1})} \\ (1 + X \times \Sigma, X \times \Sigma + Y \times \Sigma) & & (2^{Y \times \Sigma + X \times \Sigma}, 2^{X \times \Sigma + 1}) \end{array}$$

As $2^{Y\times\Sigma + X\times\Sigma} \cong (2^Y)^\Sigma \times (2^X)^\Sigma$ and $2^{X\times\Sigma+1} \cong (2^X)^\Sigma \times 2$ and by the universal properties of the product, we can simplify the right-hand side to

$$\begin{array}{ccc} & (2^Y, 2^X) & \\ & \Big\downarrow & \\ ((\xi^\flat + \sigma^\flat)^{-1}, (\rho^\flat + i)^{-1}) & & \\ & \Big\downarrow & \\ & (2^{Y\times\Sigma + X\times\Sigma}, 2^{X\times\Sigma+1}) & \end{array} \quad \Longrightarrow \quad \begin{array}{c} (2^Y, 2^X) \\ \Big\downarrow (\langle\xi^{-1}, \sigma^{-1}\rangle, \langle\rho^{-1}, i^{-1}\rangle) \\ ((2^Y)^\Sigma \times (2^X)^\Sigma, (2^X)^\Sigma \times 2) \end{array}$$

The only remaining piece of the lasso automaton is the final states. We do not apply the contravariant powerset functor but instead just take the transpose:

$$\begin{array}{ccc} & (X, Y) & \\ & \Big\downarrow & \\ (!, \chi_F) & & \\ & \Big\downarrow & \\ & (1, 2) & \end{array} \quad \Longrightarrow \quad \begin{array}{c} (2^Y, 2^X) \\ \Big\uparrow (i_F, !) \\ (1, \emptyset) \end{array}$$

The map $\chi_F : Y \to 2^1$ was turned into $i_F : 1 \to 2^Y$ with $i_F(\star) = F$ and $! : X \to 1^\emptyset$ sent to $! : \emptyset \to 1^X$ (as $2^1 \cong 2$, $1^\emptyset = 1 = 1^X$).

From this we can define the reverse lasso automaton of a lasso automaton.

Definition 8.17 Let $\mathbb{A} = (X, Y, i, \rho, \sigma, \xi, F)$ be a lasso automaton, we define its *reverse lasso automaton* as

$$\mathbb{A}^r = (2^Y, 2^X, F, \xi^{-1}, \sigma^{-1}, \rho^{-1}, i^{-1}),$$

where for $\delta \in \{\xi, \sigma, \rho\}$ we have

$$\delta^{-1}[P](a) = \{z \in \mathrm{dom}(\delta) \mid \delta(z, a) \in P\},$$

and $i^{-1}[K] = [i \in K]$.

Our lifted two-sorted contravariant powerset functor sends a lasso automaton $\mathbb{A}$ to its reverse lasso automaton $\mathbb{A}^r$. If we take $\mathbb{A} \in \mathbf{LA}(L)$, then where does $\mathbb{A}^r$ live? Clearly $\mathbb{A}^r \in \mathbf{LA}^{\mathrm{op}}$, so we wonder: For which L' do we have $\mathbb{A}^r \in \mathbf{LA}^{\mathrm{op}}(L')$? The answer turns out to be $L' = L^r$, i.e. the *reverse language* of L. In order to formally show this, we need some additional definitions which allow us to talk, amongst others, of the reverse of a word.

The next definition is standard, see for instance [27].

Definition 8.18 ([27]) For a word $w \in \Sigma^+$, we write w^r for the *reverse word*, i.e.

$$(w_1 \dots w_n)^r = w_n \dots w_1.$$

Lemma 8.19 *For $\delta \in \{\sigma, \xi, \rho\}, Q \subseteq Im(\delta)$ and $v \in \Sigma^+$:*

$$\widetilde{\delta^{-1}}[Q](v) = \widetilde{\delta}^{-1}[Q](v^r).$$

Proof. We prove the claim by induction. For the base case we let $v \in \Sigma$. In this case

$$\begin{aligned}\widetilde{\delta^{-1}}[Q](a) &= \{z \in \mathrm{dom}(\delta) \mid \delta(z,a) \in Q\} \\ &= \{z \in \mathrm{dom}(\delta) \mid \widetilde{\delta}(z,a) \in Q\} \\ &= \widetilde{\delta}^{-1}[Q](a) \\ &= \widetilde{\delta}^{-1}[Q](a^r).\end{aligned}$$

For the inductive step let $v = au$, then

$$\begin{aligned}\widetilde{\delta^{-1}}[Q](v) &= \widetilde{\delta^{-1}}[Q](au) \\ &= \widetilde{\delta^{-1}}[\delta^{-1}[Q](a)](u) \\ &= \widetilde{\delta}^{-1}[\delta^{-1}[Q](a)](u^r) && \text{(I.H.)} \\ &= \{z \in \mathrm{dom}(\delta) \mid \widetilde{\delta}(z,u^r) \in \delta^{-1}[Q](a)\} \\ &= \{z \in \mathrm{dom}(\delta) \mid \delta(\widetilde{\delta}(z,u^r),a) \in Q\} \\ &= \{z \in \mathrm{dom}(\delta) \mid \widetilde{\delta}(z,u^ra) \in Q\} \\ &= \widetilde{\delta}^{-1}[Q](u^ra) \\ &= \widetilde{\delta}^{-1}[Q](v^r).\end{aligned}$$

□

We introduce the notion of the reverse of a lasso language.

Definition 8.20 For a lasso language L, we define its *reverse lasso language* as

$$L^r = \{(u,av) \in \Sigma^{*+} \mid (v^r, au^r) \in L\}.$$

Just as for regular languages, taking the reverse of a lasso language twice gives back the original lasso language.

Lemma 8.21 *For a lasso language L, we have $L = (L^r)^r$.*

Proof. Let $(u,av) \in \Sigma^{*+}$, then

$$\begin{aligned}(u,av) \in L &\iff (v^r, au^r) \in L^r \\ &\iff ((u^r)^r, a(v^r)^r) \in (L^r)^r \\ &\iff (u,av) \in (L^r)^r.\end{aligned}$$

□

In order to show that $\mathbb{A}^r \in \mathbf{LA}^{\mathrm{op}}(L^r)$, it is sufficient to show that $\mathbb{A}^r$ accepts L^r.

Proposition 8.22 *Let $\mathbb{A} = (X, Y, i, \rho, \sigma, \xi, F)$, then*

$$\mathit{Lassos}(\mathbb{A}, i)^r = \mathit{Lassos}(\mathbb{A}^r, F).$$

Proof. Using the definitions and Lemma 8.19, we get

$$\begin{aligned}
\mathrm{Lassos}(\mathbb{A}^r, F) &= \left\{(u, av) \in \Sigma^{*+} \mid i \in \widetilde{\sigma^{-1} : \rho^{-1}}\left(\widehat{\xi^{-1}}\left(F, u\right), av\right)\right\} \\
&= \left\{(u, av) \in \Sigma^{*+} \mid i \in \widetilde{\rho^{-1}}\left(\sigma^{-1}\left(\widehat{\xi^{-1}}\left(F, u\right), a\right), v\right)\right\} \\
&= \left\{(u, av) \in \Sigma^{*+} \mid i \in \widetilde{\rho}^{-1}\left(\sigma^{-1}\left(\widehat{\xi^{-1}}\left(F, u\right), a\right), v^r\right)\right\} \\
&= \left\{(u, av) \in \Sigma^{*+} \mid \widehat{\rho}(i, v^r) \in \sigma^{-1}\left(\widehat{\xi}^{-1}(F, u^r), a\right)\right\} \\
&= \left\{(u, av) \in \Sigma^{*+} \mid \sigma(\widehat{\rho}(i, v^r), a) \in \widehat{\xi}^{-1}(F, u^r)\right\} \\
&= \left\{(u, av) \in \Sigma^{*+} \mid \widehat{\xi}(\sigma(\widehat{\rho}(i, v^r), a), u^r) \in F\right\} \\
&= \left\{(u, av) \in \Sigma^{*+} \mid \widetilde{\sigma : \xi}(\widehat{\rho}(i, v^r), au^r) \in F\right\} \\
&= \left\{(u, av) \in \Sigma^{*+} \mid (v^r, au^r) \in \mathrm{Lassos}(\mathbb{A}, i)\right\} \\
&= \mathrm{Lassos}(\mathbb{A}, i)^r.
\end{aligned}$$

□

Through this observation, we can conclude that $\mathbb{A}^r \in \mathbf{LA}^{\mathrm{op}}(L^r)$. So $\mathbb{A}^r$ is a lasso automaton over $\mathbf{Set}^{2,\mathrm{op}}$ accepting the lasso language L^r. We now formally define the lifting of $\widetilde{\mathscr{P}^2}$ to $\mathbf{LA}$, which will also be denoted by $\widetilde{\mathscr{P}^2}$.

Definition 8.23 We define the functor $\widetilde{\mathscr{P}^2}$: $\mathbf{LA} \to \mathbf{LA}^{\mathrm{op}}$ on objects as $\widetilde{\mathscr{P}^2}\ (\mathbb{A}) = \mathbb{A}^r$ and on morphisms as $\widetilde{\mathscr{P}^2}\ (f, g) = (g^{-1}[-], f^{-1}[-])$.

Proof. We show that the morphism part of the functor is well-defined. We have to prove that if $(f, g) : \mathbb{A} \to \mathbb{B}$ is a lasso automaton morphism, then so is $(g^{-1}[-], f^{-1}[-]) : \mathbb{B}^r \to \mathbb{A}^r$. Let $\mathbb{A} = (X_A, Y_A, i_A, \rho_A, \sigma_A, \xi_A, F_A)$ and $\mathbb{B} = (X_B, Y_B, i_B, \rho_B, \sigma_B, \xi_B, F_B)$. The morphism (f, g) being a lasso automaton morphism means it satisfies the following equations (for $x \in X_A, y \in Y_A$):

$$i_B = f(i_A) \qquad\qquad y \in F_A \iff g(y) \in F_B$$

$$f(\rho_A(x, a)) = \rho_B(f(x), a) \qquad g(\sigma_A(x, a)) = \sigma_B(f(x), a) \qquad g(\xi_A(y, a)) = \xi_B(g(y), a)$$

In order to show that (g^{-1}, f^{-1}) is a lasso automaton morphism, we have to show that the following equalities hold (for $Q \in 2^{Y_B}, P \in 2^{X_B}$):

$$g^{-1}[F_B] = F_A \qquad\qquad i_B \in Q \iff i_A \in f^{-1}[Q]$$

$$g^{-1}[\xi_B^{-1}[Q](a)] = \xi_A^{-1}[g^{-1}[Q]](a) \qquad\qquad f^{-1}[\sigma_B^{-1}[Q](a)] = \sigma_A^{-1}[g^{-1}[Q]](a)$$

$$f^{-1}[\rho_B^{-1}[P](a)] = \rho_A^{-1}[f^{-1}[P]](a)$$

We show them row by row starting at the top-left:

$$y \in F_A \iff g(y) \in F_B \iff y \in g^{-1}[F_B]$$

$$i_B \in Q \iff f(i_A) \in Q \iff i_A \in f^{-1}[Q]$$

$$y \in g^{-1}[\xi_B^{-1}[Q](a)] \iff \xi_B(g(y),a) \in Q \iff g(\xi_A(y,a)) \in Q \iff y \in \xi_A^{-1}[g^{-1}[Q]](a)$$
$$x \in f^{-1}[\sigma_B^{-1}[Q](a)] \iff \sigma_B(f(x),a) \in Q \iff g(\sigma_A(x,a)) \in Q \iff \sigma_A^{-1}[g^{-1}[Q]](a)$$
$$x \in f^{-1}[\rho_B^{-1}[P](a)] \iff \rho_B(f(x),a) \in P \iff f(\rho_A(x,a)) \in P \iff \rho_A^{-1}[f^{-1}[P]](a)$$

This concludes the proof. □

Corollary 8.24 *The functor $\breve{\mathscr{P}}^2$ seen as a functor on* **LA** *is self-adjoint:*

$$\breve{\mathscr{P}}^2 \dashv \breve{\mathscr{P}}^2.$$

Proof. This follows from Corollary 8.16 and Definition 8.23. □

Of particular interest is what happens if we apply $\breve{\mathscr{P}}^2$ to the initial lasso automaton in $\mathbf{LA}(L)$ for some lasso language L. This gives the lasso automaton

$$\mathbb{S}^r = (2^{\Sigma^{*+}}, 2^{\Sigma^*}, L, \mathfrak{c}^{-1}, \mathfrak{b}^{-1}, \mathfrak{a}^{-1}, \mathfrak{i}^{-1}).$$

whose carrier is the same as that of the final lasso automaton in $\mathbf{LA}^{\mathrm{op}}(L^r)$. By finality we get a map, which is in fact the reverse map, for $L \subseteq \Sigma^{*+}, M \subseteq \Sigma^*$:

$$L \mapsto L^r \qquad \text{and} \qquad M \mapsto \{u^r \mid u \in M\}.$$

As this map is an involution, $\mathbb{S}^r$ is isomorphic to the final object (c.f. Lemma 8.21).

Corollary 8.25 *The reverse of the reverse of an Ω-automaton is again an Ω-automaton.*

Proof. Let $\mathbb{A}$ be an Ω-automaton with initial state i, then $(\mathbb{A}^r)^r$ accepts the language $(\mathrm{Lassos}((\mathbb{A}^r)^r, i)^r)^r = \mathrm{Lassos}(\mathbb{A}, i)$ by Proposition 8.22 and so it is again an Ω-automaton [12]. □

8.1.3 Brzozowski Minimisation

In this subsection, we introduce a notion of minimality in terms of *observability* and *reachability* ([4]). We then show that an Ω-automaton can be minimised according to that definition by applying the reverse construction (Definition 8.17) followed by taking the reachable part twice.

Definition 8.26 ([7, 4]) We call an Ω-automaton $\mathbb{A}$ accepting a lasso language L *reachable* if the unique morphism from the initial lasso automaton is epi, and *observable* if the unique morphism to the final lasso automaton is mono. If an Ω-automaton is both reachable and observable, then it is *minimal*.

Remark 8.27 Given an Ω-automaton $\mathbb{A}$ with initial state i, we can get a new Ω-automaton $\mathbb{A}'$ consisting of all the states in $\mathbb{A}$ that are reachable from i. The Ω-automaton $\mathbb{A}'$ accepts the same ω-regular language as $\mathbb{A}$.

Theorem 8.28 *Given an Ω-automaton accepting an ω-regular language $\mathscr{L}$:*

1. *get the reverse lasso-automaton,*
2. *take the reachable part (i.e. the Ω-automaton consisting of all the states reachable from the initial state),*
3. *apply the reverse construction again,*
4. *take the reachable part.*

The resulting lasso-automaton is a minimal Ω-automaton accepting $\mathscr{L}$.

Proof. Let $\mathbb{A}$ be an Ω-automaton accepting a lasso language L, then taking the reachable reverse lasso-automaton gives a lasso automaton $\mathbb{A}^{r\prime}$ accepting L^r. Moreover, the map from the initial lasso automaton in $\mathbf{LA}^{\mathrm{op}}(L^r)$ is surjective. Applying the reverse construction again gives an Ω-automaton which accepts $(L^r)^r = L$ and is observable as $\overset{\smile}{\mathscr{P}}^2$ turns epis into monos. Finally, taking the reachable part again makes the Ω-automaton also reachable. □

8.2 Minimisation via Duality

Another minimisation procedure is given by [6] which uses a dual equivalence instead of a self-adjunction. Following the same ideas, we construct a dual equivalence between Ω-coalgebras over $\mathbf{FinSet}^2$ and L-algebras over $\mathbf{FinBA}^2$ (for a suitable functor L).

This allows us to translate the problem of finding a greatest quotient to that of finding a minimal subalgebra. This problem in turn can be solved through the use of a coalgebraic modal logic which naturally arises via relation lifting over Ω-coalgebras.

As with the previous subsection, this subsection can be seen as an instantiation of the construction from [6]. Similarly, our construction is slightly more involved than the one found in the original paper.

8.2.1 The Category $\mathbf{FinBA}^2$

The category **CABA** of complete atomic Boolean algebras is dually equivalent to **Set**. This can then be lifted to $\mathbf{CABA}^2$ and $\mathbf{Set}^2$. As we are only interested in finite Ω-automata, it is sufficient to only consider $\mathbf{FinBA}^2$ and $\mathbf{FinSet}^2$.

This subsection introduces the category $\mathbf{FinBA}^2$ and establishes a dual equivalence between $\mathbf{FinBA}^2$ and $\mathbf{FinSet}^2$. Moreover, we define an endofunctor $L : \mathbf{FinBA}^2 \to \mathbf{FinBA}^2$ giving rise to L-algebras. The category of L-algebras is dually equivalent to that of Ω-coalgebras. We conclude that finding the greatest quotient of an Ω-coalgebra is equivalent to finding the minimal subalgebra of its dual equivalent.

Definition 8.29 The category $\mathbf{FinBA}^2$ has as objects pairs of boolean algebras and as morphisms pairs of BA-homomorphisms.

Definition 8.30 ([6]) We define the functor $\mathrm{At} : \mathbf{FinBA} \to \mathbf{FinSet}^{\mathrm{op}}$ as follows:

1. objects: $B \mapsto \mathrm{At}(B)$, i.e. the atoms of B,

2. morphisms: $h : A \to B$, then $\mathrm{At}(h) : \mathrm{At}(B) \to \mathrm{At}(A)$ with

$$\mathrm{At}(h)(b) = \bigwedge \{a \in A \mid b \leq h(a)\}.$$

Proposition 8.31 *The categories* $\mathbf{FinSet}^2$ *and* $\mathbf{FinBA}^2$ *are dually equivalent.*

Proof. We have that **FinSet** and **FinBA** are dually equivalent via $\breve{\mathscr{P}}$ and At (see [6]). Then by Lemma 3.4 we can lift the dual equivalence. The functors involved are $\breve{\mathscr{P}} \times \breve{\mathscr{P}} = \breve{\mathscr{P}}^2$ and At^2. □

The next step consists of defining an algebra over $\mathbf{FinBA}^2$ which is dually equivalent to Ω-coalgebras. This requires a functor $L : \mathbf{FinBA}^2 \to \mathbf{FinBA}^2$. From [6], we know how to define L for the second sort, as they show that partially obervable DFAs (PODFAs) are dually equivalent to finite Boolean algebras with operators (FBAOs).

Definition 8.32 Let $L : \mathbf{FinBA}^2 \to \mathbf{FinBA}^2$ be given by

$$L(B_1, B_2) = \left(\biguplus_{a \in \Sigma} B_1 + \biguplus_{a \in \Sigma} B_2, \biguplus_{a \in \Sigma} B_2 + \mathrm{FinBA}(1) \right).$$

where $\mathrm{FinBA}(1)$ denotes the free Boolean algebra generated by the singleton set $\{\star\}$. The second sort of this functor is defined precisely as in [6].

We can think of an L-algebra as consisting of two Boolean algebras with operators

$$(B_1, \{[a) : B_1 \to B_1 \mid a \in \Sigma\}, \{(a) : B_2 \to B_1 \mid a \in \Sigma\}, \top_1, \wedge_1, \neg_1),$$
$$(B_2, \{(a] : B_2 \to B_2 \mid a \in \Sigma\}, \checkmark, \top_2, \wedge_2, \neg_2),$$

where all morphisms $[a), (a), (a]$ are Boolean algebra homomorphisms and $\checkmark$ is a constant. This follows from several observations. First of all, by the universal property of the coproduct, giving a Boolean algebra morphism $f : \biguplus_{a \in \Sigma} B_1 \to B_1$ is the same as giving a family of Boolean algebra morphisms $\{[a) : B_1 \to B_1\}_{a \in \Sigma}$. Analogous observations hold for the other coproducts, giving $\{(a)\}_{a \in \Sigma}$ and $\{(a]\}_{a \in \Sigma}$. Finally, specifying a map from the free Boolean algebra $\mathrm{FinBA}(1)$ to B_2 is the same as picking an element from B_2, i.e. we have a constant $\checkmark \in B_2$.

We write an L-algebra succinctly as a tuple

$$(B_1, B_2, \{[a)\}_{a \in \Sigma}, \{(a)\}_{a \in \Sigma}, \{(a]\}_{a \in \Sigma}, \checkmark, \top_i, \wedge_i, \neg_i),$$

where $i \in \{1, 2\}$. Often we simply write $\top, \wedge, \neg$ if no confusion arises.

As already mentioned earlier, [6] provides a proof that the second sort is dually equivalent to the second sort of the Ω-functor. Nevertheless, we replicate this part of the proof for the sake of uniformity and completeness.

Proposition 8.33 *The category of* Ω*-coalgebras is dually equivalent to that of* L*-algebras.*

Proof. We first extend $\breve{\mathscr{P}}^2$ and At^2 to two functors $\widehat{L}, \widehat{O}$ that turn Ω-coalgebras into L-algebras and vice-versa. For an Ω-coalgebra $(X, Y, \rho, \sigma, \xi, \chi_F)$, $\widehat{L}$ turns it into the finite Boolean algebra

$$(2^X, 2^Y, \{(\rho^\sharp(a))^{-1}\}_{a\in\Sigma}, \{(\sigma^\sharp(a))^{-1}\}_{a\in\Sigma}, \{(\xi^\sharp(a))^{-1}\}_{a\in\Sigma}, F, \{X, Y\}, \cap, \{X\backslash(-), Y\backslash(-)\}).$$

And given an L-algebra $(B_1, B_2, \{[a)\}_{a\in\Sigma}, \{(a)\}_{a\in\Sigma}, \{(a]\}_{a\in\Sigma}, \checkmark, \top_i, \wedge_i, \neg_i)$, $\widehat{O}$ turns it into the Ω-coalgebra

$$(\mathrm{At}(B_1), \mathrm{At}(B_2), \mu, \nu, \pi, \chi_\checkmark),$$

where

$$\mu(b_1)(a) = \bigwedge\{b_1' \in B_1 \mid b_1 \leq [a)(b_1')\}, \qquad \nu(b_1)(a) = \bigwedge\{b_2 \in B_2 \mid b_1 \leq (a)(b_2)\},$$
$$\pi(b_2)(a) = \bigwedge\{b_2' \in B_2 \mid b_2 \leq (a](b_2')\}, \quad \chi_\checkmark(b_2) = 1 \iff b_2 \leq \checkmark.$$

For the morphisms, $\widehat{L}$ does the same as $\breve{\mathscr{P}}^2$ and $\widehat{O}$ the same as At^2.

We explicitly define two natural isomorphisms

$$\delta : \mathrm{Id}_{\mathrm{CoAlg}(\Omega)} \implies \widehat{O} \circ \widehat{L} \qquad \text{and} \qquad \epsilon : \mathrm{Id}_{\mathrm{Alg}(L)} \implies \widehat{L} \circ \widehat{O}.$$

Let $\mathbb{A} = (X, Y, \rho, \sigma, \xi, \chi_F)$ be an Ω-coalgebra, define $\delta_\mathbb{A} : \mathbb{A} \to \widehat{O} \circ \widehat{L}(\mathbb{A})$ as

$$\delta_{\mathbb{A},0}(x) = \{x\} \qquad \text{and} \qquad \delta_{\mathbb{A},1}(y) = \{y\}.$$

The inverse is given by $\delta_{\mathbb{A},i}^{-1}(\{z\}) = z$. We have to check that $\delta_\mathbb{A}$ is an Ω-coalgebra morphism. We only show the interesting case of the switch transition (the other transitions follow analogously) and the acceptance condition. For the acceptance, after unravelling the definitions we have to show that

$$y \in F \iff \{y\} \subseteq F,$$

which holds trivially. For the switch transition, we have to verify that

$$\nu(\{x\}, a) = \{\sigma(x, a)\},$$

where $\nu(b_1, a) = \bigwedge\{b_2 \in \mathrm{At}(2^Y) \mid b_1 \subseteq (\sigma^\sharp(a))^{-1}[b_2]\}$. This is again not hard to show:

$$\begin{aligned}
\nu(\{x\}, a) &= \bigwedge\{\{y\} \in \mathrm{At}(2^Y) \mid \{x\} \subseteq (\sigma^\sharp(a))^{-1}[\{y\}]\} \\
&= \bigwedge\{\{y\} \in \mathrm{At}(2^Y) \mid x \in (\sigma^\sharp(a))^{-1}[\{y\}]\} \\
&= \bigwedge\{\{y\} \in \mathrm{At}(2^Y) \mid \sigma(x, a) \in \{y\}\} \\
&= \bigwedge\{\{y\} \in \mathrm{At}(2^Y) \mid \sigma(x, a) = y\} \\
&= \{\sigma(x, a)\}.
\end{aligned}$$

For naturality of δ, let $(f_0, f_1) : \mathbb{A} \to \mathbb{A}'$ be an Ω-coalgebra morphism, we have to check that

$$\{f_0(x)\} = \bigwedge\{\{x'\} \in \mathrm{At}(2^{X'}) \mid \{x\} \subseteq f_0^{-1}[\{x'\}]\},$$
$$\{f_1(y)\} = \bigwedge\{\{y'\} \in \mathrm{At}(2^{Y'}) \mid \{y\} \subseteq f_1^{-1}[\{y'\}]\}.$$

This is indeed the case as

$$\bigwedge\{\{z'\} \in \mathrm{At}(2^{Z'}) \mid \{z\} \subseteq f_i^{-1}[\{z'\}]\} = \bigwedge\{\{z'\} \in \mathrm{At}(2^{Z'}) \mid f_i(z) = z'\} = \{f_i(z)\}.$$

Let $\mathbb{B} = (B_1, B_2, \{[a)\}_{a\in\Sigma}, \{(a)\}_{a\in\Sigma}, \{(a]\}_{a\in\Sigma}, \checkmark, \top_i, \wedge_i, \neg_i)$ be an L-algebra. We define $\epsilon_{\mathbb{B}} : \mathbb{B} \to \widehat{L} \circ \widehat{O}(\mathbb{B})$ as

$$\epsilon_{\mathbb{B},0}(b_1) = \{b_1' \in \mathrm{At}(\mathbb{B}_1) \mid b_1' \leq b_1\} \qquad \text{and} \qquad \epsilon_{\mathbb{B},1}(b_2) = \{b_2' \in \mathrm{At}(\mathbb{B}_2) \mid b_2' \leq b_2\}$$

This time the inverse is given by $\epsilon_{\mathbb{B},i}^{-1}(Q) = \bigvee Q$. We have to show that $\epsilon_{\mathbb{B}}$ is an L-algebra morphism and we only show this for the switch maps $\{(a)\}_{a\in\Sigma}$. The check for the constant $\checkmark$ follows immediately from the definitions. For the switch map we have to check that

$$(\nu^\sharp(a))^{-1}[\epsilon_{\mathbb{B},1}(b_2)] = \epsilon_{\mathbb{B},0}((a)b_2).$$

Unravelling the definitions we get:

$$\begin{aligned}
(\nu^\sharp(a))^{-1}[\epsilon_{\mathbb{B},1}(b_2)] &= (\nu^\sharp(a))^{-1}[\{b_2' \in \mathrm{At}(B_2) \mid b_2' \leq b_2\}] \\
&= \{b_1 \in \mathrm{At}(B_1) \mid \nu(b_1, a) \in \{b_2' \in \mathrm{At}(B_2) \mid b_2' \leq b_2\}\} \\
&= \{b_1 \in \mathrm{At}(B_1) \mid \nu(b_1, a) \leq b_2\} \\
&= \{b_1 \in \mathrm{At}(B_1) \mid \textstyle\bigwedge\{b_2' \in B_2 \mid b_1 \leq (a)b_2'\} \leq b_2\} \\
&= \{b_1 \in \mathrm{At}(B_1) \mid b_1 \leq (a)b_2\} \\
&= \epsilon_{\mathbb{B},0}((a)b_2).
\end{aligned}$$

The penultimate step follows as

$$\textstyle\bigwedge\{b_2' \in B_2 \mid b_1 \leq (a)b_2'\} \leq b_2 \implies b_1 \leq (a)b_2,$$

$$b_1 \leq (a)b_2 \implies b_2 \in \{b_2' \in B_2 \mid b_1 \leq (a)b_2'\} \implies \textstyle\bigwedge\{b_2' \in B_2 \mid b_1 \leq (a)b_2'\} \leq b_2.$$

Lastly we need to prove the naturality of ϵ. Let $(f_0, f_1) : \mathbb{B} \to \mathbb{B}'$. We have to show that

$$\epsilon_{\mathbb{B}',0}(f_0(b_1)) = (\mathrm{At}(f_0))^{-1}[\epsilon_{\mathbb{B},0}(b_1)] \qquad \text{and} \qquad \epsilon_{\mathbb{B}',1}(f_1(b_2)) = (\mathrm{At}(f_1))^{-1}[\epsilon_{\mathbb{B},1}(b_2)].$$

Filling in the definitions, this is equivalent to showing

$$\begin{aligned}
\{b_1' \in \mathrm{At}(B_1') \mid b_1' \leq f_0(b_1)\} &= \{b_1' \in \mathrm{At}(B_1') \mid \mathrm{At}(f_0)(b_1') \in \{c \in \mathrm{At}(B_1) \mid c \leq b_1\}\}, \\
\{b_2' \in \mathrm{At}(B_2') \mid b_2' \leq f_1(b_2)\} &= \{b_2' \in \mathrm{At}(B_2') \mid \mathrm{At}(f_1)(b_2') \in \{d \in \mathrm{At}(B_2) \mid d \leq b_2\}\}.
\end{aligned}$$

We show the first of the two equalities, the other follows analogously.

$$\begin{aligned}
\mathrm{At}(f_0)(b_1') \in \{c \in \mathrm{At}(B_1) \mid c \leq b_1\} &\iff \mathrm{At}(f_0)(b_1') \leq b_1 \\
&\iff \textstyle\bigwedge\{c \in \mathrm{At}(B_1) \mid b_1' \leq f_0(c)\} \leq b_1 \\
&\iff \textstyle\bigwedge\{c \in \mathrm{At}(B_1) \mid b_1' \leq f_0(c)\} \leq b_1 \\
&\iff b_1' \leq f_0(b_1). \qquad \square
\end{aligned}$$

Corollary 8.34 *Given an Ω-coalgebra $\mathbb{A}$, and its dual equivalent $\mathbb{A}'$, if $\mathbb{B}$ is a minimal subalgebra of $\mathbb{A}'$, then the dual of $\mathbb{B}$ is the greatest quotient of $\mathbb{A}$.*

Proof. This is a direct consequence of the dual equivalence we have established. $\square$

8.2.2 Two-sorted Coalgebraic Modal Logic and Minimal Subalgebras

In the previous subsection we established that finding the greatest quotient of an Ω-coalgebra is equivalent to finding the minimal subalgebra of its dual L-algebra. As in [6], we show in the next subsection that this subalgebra can be obtained by looking at definable subsets. The idea is to bring a separation logic into the picture which reflects the structure of the Ω-coalgebra. By considering only the definable subsets we obtain a minimal subalgebra.

This subsection introduces the logic we use to find the minimal subalgebra. We start with a coalgebraic modal logic which arises naturally via relation lifting ([23]) and then define a separation logic which is equi-expressive and more convenient to work with.

Definition 8.35 ([23]) We define the first and second-sort formulas, Fml_1 and Fml_2 respectively, of the language ML_Ω as follows:

$$\varphi_1, \psi_1 ::= \bot \mid \neg\varphi_1 \mid \varphi_1 \vee \psi_1 \mid \nabla_1\alpha$$
$$\varphi_2, \psi_2 ::= \bot \mid \neg\varphi_2 \mid \varphi_2 \vee \psi_2 \mid \nabla_2\beta$$

with $\alpha \in \mathrm{Fml}_1^\Sigma \times \mathrm{Fml}_2^\Sigma$ and $\beta \in \mathrm{Fml}_2^\Sigma \times 2$ (so $\alpha = (\alpha_1, \alpha_2)$ and $\beta = (\beta_1, i)$ with $\alpha_1, \alpha_2, \beta_1$ functions and $i \in \{0, 1\}$).

Definition 8.36 ([23]) Given an Ω-coalgebra $\mathbb{A} = (X, Y, \rho, \xi, \sigma, \chi)$, $x \in X$ and $y \in Y$, we define

$$\begin{aligned}
\mathbb{A}, x \Vdash \nabla_1\alpha \iff \forall c \in \Sigma : \mathbb{A}, \rho(x)(c) \Vdash \alpha_1(c) \text{ and} \\
\mathbb{A}, \sigma(x)(c) \Vdash \alpha_2(c), \\
\mathbb{A}, y \Vdash \nabla_2\beta \iff \forall c \in \Sigma : \mathbb{A}, \xi(y)(c) \Vdash \beta_1(c) \text{ and } \chi(y) = i.
\end{aligned}$$

The semantics of $\neg\varphi, \varphi \vee \psi$ are standard.

Proposition 8.37 *The language ML_Ω is expressive and invariant (for behavioural equivalence and bisimilarity).*

Proof. The endofunctor Ω preserves both weak pullbacks and inclusions. Moreover, we are only considering objects and morphisms in $\mathbf{FinSet}^2$ so Ω is image-finite. It follows that the language ML_Ω is both expressive and invariant. □

The coalgebraic modal logic ML_Ω is not very convenient to work with. So we introduce a different modal logic and show that it is equi-expressive.

Definition 8.38 We define the following two sorted-modal logic ML_Ω^S, where the formulas for the first- and second sort are given as

$$\varphi_1, \psi_1 ::= \bot \mid \neg\varphi_1 \mid \varphi_1 \vee \psi_1 \mid [a]\varphi_1 \mid (a)\varphi_2$$
$$\varphi_2, \psi_2 ::= \checkmark \mid \bot \mid \neg\varphi_2 \mid \varphi_2 \vee \psi_2 \mid (a]\varphi_2$$

where we get, for each $a \in \Sigma$, three modalities ($[a)$, (a) and $(a]$) and we write $\varphi_1, \psi_1 \in \mathrm{Fml}_1$ and $\varphi_2, \psi_2 \in \mathrm{Fml}_2$.

We next give the semantics for ML_Ω^S.

Definition 8.39 Given an Ω-coalgebra $\mathbb{A} = (X, Y, \rho, \sigma, \xi, \chi)$, $x \in X$, $y \in Y$ and $c \in \Sigma$, we define

$$\begin{aligned}
\mathbb{A}, x \Vdash [c]\varphi_1 &\iff \mathbb{A}, \rho(x,c) \Vdash \varphi_1,\\
\mathbb{A}, x \Vdash (c)\varphi_2 &\iff \mathbb{A}, \sigma(x,c) \Vdash \varphi_2,\\
\mathbb{A}, y \Vdash (c]\varphi_2 &\iff \mathbb{A}, \xi(y,c) \Vdash \varphi_2,\\
\mathbb{A}, y \Vdash \checkmark &\iff \chi(y) = 1.
\end{aligned}$$

The semantics of $\neg\varphi, \varphi \vee \psi$ are standard.

Proposition 8.40 *For a finite alphabet Σ, the modal logics ML_Ω and ML_Ω^S are equi-expressive.*

Proof. Consider the function h which translates from ML_Ω to ML_Ω^S. We only give it for $\alpha \in \mathrm{Fml}_1^\Sigma \times \mathrm{Fml}_2^\Sigma$ and $\beta \in \mathrm{Fml}_2^\Sigma \times 2$.

$$\begin{aligned}
h(\nabla_1\alpha) &:= \bigwedge_{c\in\Sigma} \big([c]h(\alpha_1(c)) \wedge (c)h(\alpha_2(c))\big),\\
h(\nabla_2\beta) &:= [i] \wedge \bigwedge_{c\in\Sigma} (c]h(\beta_1(c)).
\end{aligned}$$

where we use $[i]$ to mean $\checkmark$ if $i = 1$ and $\neg\checkmark$ otherwise. We first show that h respects the semantics. This can be done by induction and we only look at the case for the cover modality.

$$\begin{aligned}
\mathbb{A}, x \Vdash \nabla_1\alpha &\iff \forall c \in \Sigma : \mathbb{A}, \rho(x)(c) \Vdash \alpha_1(c) \text{ and } \mathbb{A}, \sigma(x)(c) \Vdash \alpha_2(c)\\
&\iff \forall c \in \Sigma : \mathbb{A}, \rho(x)(c) \Vdash h(\alpha_1(c)) \text{ and } \mathbb{A}, \sigma(x)(c) \Vdash h(\alpha_2(c)) && \text{(I.H.)}\\
&\iff \forall c \in \Sigma : \mathbb{A}, x \Vdash [c]h(\alpha_1(c)) \text{ and } \mathbb{A}, x \Vdash (c)h(\alpha_2(c))\\
&\iff \forall c \in \Sigma : \mathbb{A}, x \Vdash [c]h(\alpha_1(c)) \wedge (c)h(\alpha_2(c))\\
&\iff \mathbb{A}, x \Vdash \bigwedge_{c\in\Sigma} ([c]h(\alpha_1(c)) \wedge (c)h(\alpha_2(c))) && \text{(as } \Sigma \text{ finite)}\\
&\iff \mathbb{A}, x \Vdash h(\nabla_1\alpha)\\
\mathbb{A}, y \Vdash \nabla_2\beta &\iff \forall c \in \Sigma : \mathbb{A}, \xi(y)(c) \Vdash \beta_1(c) \text{ and } \chi(y) = i\\
&\iff \forall c \in \Sigma : \mathbb{A}, \xi(y)(c) \Vdash h(\beta_1(c)) \text{ and } \chi(y) = i && \text{(I.H.)}\\
&\iff \forall c \in \Sigma : \mathbb{A}, y \Vdash (c]h(\beta_1(c)) \text{ and } \chi(y) = i\\
&\iff \mathbb{A}, y \Vdash [i] \wedge \bigwedge_{c\in\Sigma} (c]h(\beta_1(c)) && \text{(as } \Sigma \text{ finite)}\\
&\iff \mathbb{A}, y \Vdash h(\nabla_2\beta)
\end{aligned}$$

This shows that ML_Ω^S is at least as expressive as ML_Ω. To show that they are equi-expressive, we also provide a translation h' the other way. Our focus is again on the

modalities. For $[c]\varphi$ we define $h'([c]\varphi) = \nabla_1\alpha$ where

$$\alpha_1 = \lambda a. \begin{cases} h'(\varphi) & \text{, if } a = c \\ \neg\bot & \text{, otherwise} \end{cases} \qquad \alpha_2 = \lambda a.\neg\bot$$

We can employ the same trick for $(c), (c]$. Checking that h' respects the semantics is straight-forward and can be done by induction. The cover modality case goes as follows:

$$\begin{aligned} \mathbb{A}, x \Vdash h'([a]\varphi) &\iff \mathbb{A}, \rho(x,a) \Vdash h'(\varphi) \text{ and} \\ &\quad \forall c \in \Sigma, c \neq a : (\mathbb{A}, \rho(x,c) \Vdash \top \text{ and } \mathbb{A}, \sigma(x,c) \Vdash \top) \\ &\iff \mathbb{A}, \sigma(x,a) \Vdash h'(\varphi) \\ &\iff \mathbb{A}, \sigma(x,a) \Vdash \varphi \qquad \text{(I.H.)} \\ &\iff \mathbb{A}, x \Vdash [a]\varphi. \qquad \square \end{aligned}$$

8.2.3 Minimisation via Dual Equivalence

In this final part, we introduce the notion of definable sets. Given an Ω-coalgebra $\mathbb{A}$, we show that we can get a minimal subalgebra of its dual by considering its definable subsets. Once we have a minimial subalgebra, translating it back to an Ω-coalgebra gives us the greatest quotient of $\mathbb{A}$ as required.

The following definition is standard.

Definition 8.41 Let $\mathbb{A} = (X, Y, \rho, \sigma, \xi, \chi)$ be an Ω-coalgebra. A subset $P \subseteq X$ is *definable* if there exists $\varphi \in \mathrm{Fml}_1$ such that for all $x \in X$:

$$\mathbb{A}, x \Vdash \varphi \iff x \in P.$$

Similarly, we have that a subset $Q \subseteq Y$ is definable if there exists $\varphi \in \mathrm{Fml}_2$, such that for all $y \in Y$:

$$\mathbb{A}, y \Vdash \varphi \iff y \in Q.$$

Proposition 8.42 *Let $\mathbb{A}$ be an Ω-coalgebra and $\mathbb{B}$ its dual L-algebra. Restricting the carrier of $\mathbb{B}$ to the definable subsets gives a minimial L-subalgebra of $\mathbb{B}$.*

Proof. We start by showing that restricting the carrier to the definable subsets gives us an L-algebra. We then show that it is minimal. Let $\mathbb{A} = (X, Y, \rho, \sigma, \xi, \chi_F)$ and

$$\mathbb{B} = (2^X, 2^Y, \{[a) \mid a \in \Sigma\}, \{(a) \mid a \in \Sigma\}, \{(a] \mid a \in \Sigma\}, \{\checkmark\}, \top_i, \wedge_i, \neg_i).$$

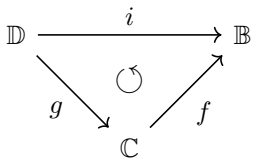
Let D_1, D_2 be the definable subsets of X and Y respectively. We show that they both form a Boolean subalgebra. First of all, we note that $\emptyset, X, Y$ are definable (consider the formulas $\bot, \neg\bot$). Next, we show that they are both closed under negation and union. This is done in a similar way for both sorts, so we only show it for D_1. Let $Q, Q' \in D_1$, we have to show that $Q \cup Q'$ and $\neg Q$ are also definable. Let φ define Q and φ' define Q', it is easy to see that $\varphi \vee \varphi'$ and $\neg\varphi$ define $Q \cup Q'$ and $\neg Q$ respectively.

From this, we get an L-algebra by restricting the operations to the definable subsets. We briefly justify that doing so is well-defined. For the constant $\checkmark$, take the subset defined

by the formula $\checkmark$. The proofs for $[a], (a)$ and $(a]$ being well-defined are analogous, so we only treat one case here. Let $Q \in D_2$, we want to show that $(a)[Q] \in D_1$ which is done by showing that $(a)[Q]$ is definable. As a reminder, we defined $(a) : 2^Y \to 2^X$ as $(a)[Q] = \sigma^{-1}(a)[Q] = \{x \in X \mid \sigma(x, a) \in Q\}$. Let φ define Q. We claim that $(a)\varphi$ defines $(a)[Q]$. Let $x \in (a)[Q]$, then $\sigma(x, a) \in Q$ and so

$$x \in (a)[Q] \iff \sigma(x, a) \in Q \iff \mathbb{A}, \sigma(x, a) \Vdash \varphi \iff \mathbb{A}, x \Vdash (a)\varphi.$$

Finally, we want to show that the L-algebra $\mathbb{D}$ we obtained by restricting the carrier to D_1, D_2 is minimal. This is done by showing that, for any Boolean subalgebra $\mathbb{C} = (C_1, C_2)$ and L-morphism $f : \mathbb{C} \to \mathbb{B}$, there exists a unique $g : \mathbb{D} \to \mathbb{C}$ making the following diagram commute:



We show how to obtain the unique map g. For this, we have to map each $Q \in D_i$ to some $c \in C_i$. We do so by induction on ML^S_Ω-formulas. The top and bottom elements of D_i are mapped to those of the C_i respectively. This covers the sets defined by $\bot$ and $\neg\bot$. For the formula $\checkmark$, let $Q \in D_2$ be defined by $\checkmark$, i.e. $Q = \checkmark$ (seen as a constant). As $\mathbb{C}$ is an L-algebra, it has some constant $\checkmark' \in C_2$, so we let $g(Q) = \checkmark'$. For the first inductive case, let $Q \in D_i$ be defined by $\neg\varphi$. By our induction hypothesis we have a set $Q' \in D_i$ defined by φ such that we already have $g_i(Q')$. Now we map Q to the complement of $g_i(Q')$ in C_i. Next consider the formula $(a)\varphi$ which defines some set $Q \in D_1$. By our induction hypothesis we have a set $P \in D_2$ defined by φ which is mapped to $g_2(P)$. Finally, we set $g(Q) = (a)g_2(P)$. The case for $[a)\varphi, (a]\varphi$ is analogous. This gives us a unique L-morphism g. $\square$

Conclusion

In this thesis, we established results surrounding Ω-automata. We introduced Ω-automata as coalgebras and studied their base category $\mathbf{Set}^2$, showing how to lift adjunctions and equivalences from **Set** to $\mathbf{Set}^2$. This was particularly useful in Section 8.

In Section 4, we equipped lassos with a rewrite system and showed that every lasso has a unique normal form. This led us to prove the Lasso Representation Lemma 4.11 and became a significant tool in later sections.

We showed that from an Ω-automaton, we can construct its transition Wilke algebra. With the transition Wilke algebra we proved that the ω-regular language accepted by our Ω-automaton is also recognised by the Wilke algebra homomorphism from the Wilke algebra $\mathbb{W}^{+,\omega}$ to the transition Wilke algebra. The converse direction of this result is still open and of interest. In particular, we think it would be useful to frame the result in categorical terms.

In Section 6 we strengthened the Myhill-Nerode theorem taken from [12], obtaining a tight lower bound on the size of Ω-automata accepting a certain ω-regular language based on the index of our Myhill-Nerode equivalence relation.

Section 7 provided a Brzozowski type construction for Ω-automata from ω-regular languages. This construction is not entirely constructive as it stands. This section also provides some fascinating topics for further research. We showed that every ω-regular language is coverable by a lasso expression (Prop. 7.49). In fact, the lasso expression we obtain is slightly stronger than that: Let κ be an ω-regular expression, then for any $uv^\omega \in [\![\kappa]\!]_\omega$, a majority of the lassos representing uv^ω will belong to $[\![h(\kappa)]\!]_\ell$. The question is, can we construct a lasso expression which is equivalent, so for all $uv^\omega \in [\![\kappa]\!]_\omega$, we have that all lassos representing uv^ω are in $[\![h(\kappa)]\!]_\ell$. If so, then this would allow a more efficient construction for Ω-automata. Moreover, we would be able to get size constraints on the number of states of an Ω-automaton accepting κ.

Finally, we looked at minimisation procedures. This showed that common minimisation procedures for DFAs can also be applied to Ω-automata with slight modifications.

References

[1] Jirí Adámek, Filippo Bonchi, Mathias Hülsbusch, Barbara König, Stefan Milius, and Alexandra Silva. A coalgebraic perspective on minimization and determinization. In Lars Birkedal, editor, *Foundations of Software Science and Computational Structures - 15th International Conference, FOSSACS 2012, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2012, Tallinn, Estonia, March 24 - April 1, 2012. Proceedings*, volume 7213 of *Lecture Notes in Computer Science*, pages 58–73. Springer, 2012.

[2] Dana Angluin, Udi Boker, and Dana Fisman. Families of DFAs as acceptors of omega-regular languages. In Piotr Faliszewski, Anca Muscholl, and Rolf Niedermeier, editors, *41st International Symposium on Mathematical Foundations of Computer Science, MFCS 2016, August 22-26, 2016 - Kraków, Poland*, volume 58 of *LIPIcs*, pages 11:1–11:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

[3] Michael A. Arbib and Ernest G. Manes. Adjoint machines, state-behavior machines, and duality. *Journal of Pure and Applied Algebra*, 6(3):313–344, 1975.

[4] Michael A. Arbib and H. Paul Zeiger. On the relevance of abstract algebra to control theory. *Autom.*, 5(5):589–606, 1969.

[5] Nick Bezhanishvili, Marcello M. Bonsangue, Helle Hvid Hansen, Dexter Kozen, Clemens Kupke, Prakash Panangaden, and Alexandra Silva. Minimisation in logical form. *CoRR*, abs/2005.11551, 2020.

[6] Nick Bezhanishvili, Clemens Kupke, and Prakash Panangaden. Minimization via duality. In C.-H. Luke Ong and Ruy J. G. B. de Queiroz, editors, *Logic, Language, Information and Computation - 19th International Workshop, WoLLIC 2012, Buenos Aires, Argentina, September 3-6, 2012. Proceedings*, volume 7456 of *Lecture Notes in Computer Science*, pages 191–205. Springer, 2012.

[7] Filippo Bonchi, Marcello M. Bonsangue, Helle Hvid Hansen, Prakash Panangaden, Jan J. M. M. Rutten, and Alexandra Silva. Algebra-coalgebra duality in Brzozowski's minimization algorithm. *ACM Trans. Comput. Log.*, 15(1):3:1–3:29, 2014.

[8] Janusz A. Brzozowski. Derivatives of regular expressions. *J. ACM*, 11(4):481–494, 1964.

[9] Hugues Calbrix, Maurice Nivat, and Andreas Podelski. Ultimately periodic words of rational w-languages. In Stephen D. Brookes, Michael G. Main, Austin Melton, Michael W. Mislove, and David A. Schmidt, editors, *Mathematical Foundations of Programming Semantics, 9th International Conference, New Orleans, LA, USA, April 7-10, 1993, Proceedings*, volume 802 of *Lecture Notes in Computer Science*, pages 554–566. Springer, 1993.

[10] Olivier Carton, Dominique Perrin, and Jean-Eric Pin. Automata and semigroups recognizing infinite words. In Jörg Flum, Erich Grädel, and Thomas Wilke, editors,

Logic and Automata: History and Perspectives [in Honor of Wolfgang Thomas], volume 2 of *Texts in Logic and Games*, pages 133–168. Amsterdam University Press, 2008.

[11] Vincenzo Ciancia and Yde Venema. Stream automata are coalgebras. In Dirk Pattinson and Lutz Schröder, editors, *Coalgebraic Methods in Computer Science - 11th International Workshop, CMCS 2012, Colocated with ETAPS 2012, Tallinn, Estonia, March 31 - April 1, 2012, Revised Selected Papers*, volume 7399 of *Lecture Notes in Computer Science*, pages 90–108. Springer, 2012.

[12] Vincenzo Ciancia and Yde Venema. Omega-automata: A coalgebraic perspective on regular omega-languages. In Markus Roggenbach and Ana Sokolova, editors, *8th Conference on Algebra and Coalgebra in Computer Science, CALCO 2019, June 3-6, 2019, London, United Kingdom*, volume 139 of *LIPIcs*, pages 5:1–5:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[13] Szilárd Zsolt Fazekas. Powers of regular languages. In Volker Diekert and Dirk Nowotka, editors, *Developments in Language Theory, 13th International Conference, DLT 2009, Stuttgart, Germany, June 30 - July 3, 2009. Proceedings*, volume 5583 of *Lecture Notes in Computer Science*, pages 221–227. Springer, 2009.

[14] Nate Foster, Dexter Kozen, Matthew Milano, Alexandra Silva, and Laure Thompson. A coalgebraic decision procedure for NetKAT. In Sriram K. Rajamani and David Walker, editors, *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2015, Mumbai, India, January 15-17, 2015*, pages 343–355. ACM, 2015.

[15] Fabian Frei, Juraj Hromkovic, and Juhani Karhumäki. Roots and powers in regular languages: Recognizing nonregular properties by finite automata. *Fundam. Informaticae*, 175(1-4):173–185, 2020.

[16] Bart Jacobs. A bialgebraic review of deterministic automata, regular expressions and languages. In Kokichi Futatsugi, Jean-Pierre Jouannaud, and José Meseguer, editors, *Algebra, Meaning, and Computation, Essays Dedicated to Joseph A. Goguen on the Occasion of His 65th Birthday*, volume 4060 of *Lecture Notes in Computer Science*, pages 375–404. Springer, 2006.

[17] Bart Jacobs. *Introduction to Coalgebra: Towards Mathematics of States and Observation*, volume 59 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2016.

[18] Tobias Kappé, Paul Brunet, Alexandra Silva, and Fabio Zanasi. Concurrent Kleene algebra: Free model and completeness. In Amal Ahmed, editor, *Programming Languages and Systems - 27th European Symposium on Programming, ESOP 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14-20, 2018, Proceedings*, volume 10801 of *Lecture Notes in Computer Science*, pages 856–882. Springer, 2018.

[19] Stephen Cole Kleene. Representation of events in nerve nets and finite automata. *Annals of Mathematics Studies*, 34:3–41, 1956.

[20] Dexter Kozen. A completeness theorem for kleene algebras and the algebra of regular events. *Inf. Comput.*, 110(2):366–390, 1994.

[21] Bryan Krawetz, John Lawrence, and Jeffrey O. Shallit. State complexity and the monoid of transformations of a finite set. *Int. J. Found. Comput. Sci.*, 16(3):547–563, 2005.

[22] Tom Leinster. Basic category theory. 2016.

[23] Lawrence S. Moss. Coalgebraic logic. *Ann. Pure Appl. Log.*, 96(1-3):277–317, 1999.

[24] Dominique Perrin. Les débuts de la théorie des automates. *Revue des Sciences et Technologies de l'Information - Série TSI : Technique et Science Informatiques*, 14(4):409–433, 1995.

[25] Dominique Perrin and Jean-Eric Pin. *Infinite Words - Automata, Semigroups, Logic and Games*, volume 141 of *Pure and applied mathematics series*. Elsevier Morgan Kaufmann, 2004.

[26] Roman R. Redziejowski. An improved construction of deterministic omega-automaton using derivatives. *Fundam. Informaticae*, 119(3-4):393–406, 2012.

[27] Grzegorz Rozenberg and Arto Salomaa, editors. *Handbook of Formal Languages, Volume 1: Word, Language, Grammar*. Springer, 1997.

[28] Jan J. M. M. Rutten. Universal coalgebra: a theory of systems. *Theor. Comput. Sci.*, 249(1):3–80, 2000.

[29] Terese. *Term Rewriting Systems*, volume 55 of *Cambridge tracts in theoretical computer science*. Cambridge University Press, 2003.

[30] Peter Thiemann and Martin Sulzmann. From ω-regular expressions to Büchi automata via partial derivatives. In Adrian-Horia Dediu, Enrico Formenti, Carlos Martín-Vide, and Bianca Truthe, editors, *Language and Automata Theory and Applications - 9th International Conference, LATA 2015, Nice, France, March 2-6, 2015, Proceedings*, volume 8977 of *Lecture Notes in Computer Science*, pages 287–298. Springer, 2015.

[31] Yih-Kuen Tsay and Moshe Y. Vardi. From linear temporal logics to Büchi automata: The early and simple principle. In Ernst-Rüdiger Olderog, Bernhard Steffen, and Wang Yi, editors, *Model Checking, Synthesis, and Learning - Essays Dedicated to Bengt Jonsson on The Occasion of His 60th Birthday*, volume 13030 of *Lecture Notes in Computer Science*, pages 8–40. Springer, 2021.

[32] Klaus W. Wagner. Eine Axiomatisierung der Theorie der regulären Folgenmengen. *J. Inf. Process. Cybern.*, 12(7):337–354, 1976.

[33] Thomas Wilke. An algebraic theory for regular languages of finite and infinite words. *Int. J. Algebra Comput.*, 3(4):447–490, 1993.