Information and Computation 244 (2015) 49–75

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

The dual equivalence of equations and coequations for automata

A. Ballester-Bolinches [a,*], E. Cosme-Llópez [a], J. Rutten [b,c]

[a] Departament d'Àlgebra, Universitat de València Dr. Moliner, 50, 46100 Burjassot, València, Spain
[b] CWI Science Park, 123, 1098 XG Amsterdam, The Netherlands
[c] Radboud Universiteit Nijmegen, Heyendaalseweg, 135, 6500 GL Nijmegen, The Netherlands

ARTICLE INFO

Article history:
Received 26 May 2014
Received in revised form 29 May 2015
Available online 18 August 2015

Keywords:
Automata
Languages
Algebra
Coalgebra

ABSTRACT

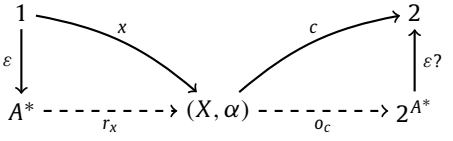
The transition structure $\alpha : X \to X^A$ of a deterministic automaton with state set X and with inputs from an alphabet A can be viewed both as an algebra and as a coalgebra. We use this algebra–coalgebra duality as a common perspective for the study of equations and coequations. For every automaton (X, α), we define two new automata: $\mathsf{free}(X, \alpha)$ and $\mathsf{cofree}(X, \alpha)$ representing, respectively, the greatest set of equations and the smallest set of coequations satisfied by (X, α). Both constructions are shown to be functorial. Our main result is that the restrictions of free and cofree to, respectively, preformations of languages and to quotients A^*/C of A^* with respect to a congruence relation C, form a dual equivalence. As a consequence, we present a variant of Eilenberg's celebrated variety theorem for varieties of monoids (in the sense of Birkhoff) and varieties of languages.

1. Introduction

In this paper, a deterministic automaton is a pair (X, α) consisting of a possibly infinite set X of states and a transition function $\alpha : X \to X^A$, with inputs from an alphabet A. Because of the isomorphism

$$(X \times A) \to X \quad \cong \quad X \to X^A$$

a deterministic automaton can be viewed both as an algebra [1,2] and as a coalgebra [3,4]. This algebra–coalgebra duality in the modelling of automata leads us to the following setting for our investigations:



(1)

In the middle, we have our automaton (X, α). Any function $x : 1 \to X$ represents the choice of a designated *point*, that is, initial state, $x \in X$. Dually, any function $c : X \to 2$ gives us a (binary) *colouring* of the states in X or, equivalently, a set

* Corresponding author.
E-mail addresses: adolfo.ballester@uv.es (A. Ballester-Bolinches), enric.cosme@uv.es (E. Cosme-Llópez), jjmmrutten@gmail.com (J. Rutten).

http://dx.doi.org/10.1016/j.ic.2015.08.001
0890-5401/

$\{x \mid c(x) = 1\}$ of final or accepting states. On the left side of our diagram, A^* is the automaton of all words over A, with transitions

$$v \xrightarrow{a} va$$

and with the empty word ε as initial state. Furthermore, every point $x : 1 \to X$ determines a unique *homomorphism* (that is, transition preserving function)

$$r_x \colon A^* \to X \qquad w \mapsto x_w$$

that sends any word w to the state x_w reached from the initial state x on input w. Dually, on the right side of our diagram, 2^{A^*} is the automaton of all languages over A, with transitions

$$L \xrightarrow{a} L_a = \{v \in A^* \mid av \in L\}$$

and colouring function $\varepsilon?$, asking whether the empty word belongs to a language or not

$$\varepsilon?(L) = \begin{cases} 1 & \text{if } \varepsilon \in L \\ 0 & \text{if } \varepsilon \notin L \end{cases}$$

Every colouring $c : X \to 2$ determines a unique homomorphism

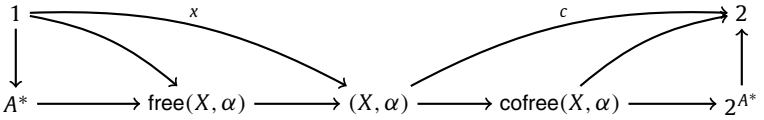
$$o_c \colon X \to 2^{A^*} \qquad x \mapsto \{w \in A^* \mid c(x_w) = 1\}$$

that sends a state x to the language that it accepts.

As it turns out, a pointed automaton (X, x, α) is an algebra (and not a coalgebra); a coloured automaton (X, c, α) is a coalgebra (and not an algebra). And a pointed and coloured automaton (X, x, c, α), which is what in the literature is usually taken as the definition of 'deterministic automaton', is neither an algebra nor a coalgebra.

Now sets of *equations* will live in the left – algebraic – part of our diagram and correspond to the *kernels* of the homomorphisms r_x; that is, sets of pairs of words (v, w) with $x_v = x_w$. Dually, sets of *coequations* live in the right – coalgebraic – part of our diagram and correspond to the *image* of the homomorphisms o_c; that is, sets of languages containing $o_c(x)$, for every $x \in X$. Satisfaction of sets of equations and coequations by the automaton (X, α) will then be defined by quantifying over all points $x : 1 \to X$ and all colourings $c : X \to 2$, respectively.

The main contribution of the present paper will be the observation that equations and coequations of automata are related by a dual equivalence. To this end, we will further refine diagram (1) as follows:



The new diagram includes, for every automaton (X, α) a new automaton $\mathsf{free}(X, \alpha)$, which will be shown to represent the *largest set of equations* satisfied by (X, α). And, dually, we will construct an automaton $\mathsf{cofree}(X, \alpha)$, which will represent the *smallest set of coequations* satisfied by (X, α). The automaton $\mathsf{free}(X, \alpha)$ will turn out to be isomorphic to the so-called *transition monoid* from algebraic language theory [5,6] and as a consequence, $\mathsf{cofree}(X, \alpha)$ can be viewed as its dual.

Next, we will show that the constructions of $\mathsf{free}(X, \alpha)$ and $\mathsf{cofree}(X, \alpha)$ are in fact functorial, that is, they act also on (certain) homomorphisms of automata. If we then restrict the functor cofree to the image of the category of automata under free, we obtain our main result: a dual equivalence. This dual equivalence relates, more precisely, two special classes of automata: on the one hand, the class of quotients A^*/C of the automaton A^* with respect to a congruence relation $C \subseteq A^* \times A^*$; on the other hand, the class of preformations of languages, which in the present paper are defined as subautomata of the automaton 2^{A^*} that are complete atomic Boolean algebras closed under left and right language derivatives. As it turns out, this duality is a lifting of the well-known dual equivalence between sets and complete atomic Boolean algebras: on congruence quotients, cofree acts as the powerset construction, and on preformations, applying free amounts to taking the set of atoms.

We then illustrate the dual equivalence between equations and coequations by applications to both regular languages and non-regular ones, such as context-free languages. Furthermore, we will show how to use the duality to give (co)equational definitions of interesting classes of languages, again not restricted to regular ones. We also present a variant of Eilenberg's celebrated variety theorem [2]. We replace pseudovarieties in the original work of Eilenberg by varieties of monoids (in the sense of Birkhoff [7]). Further, we replace varieties of regular languages by varieties of languages, which are classes of formal languages closed under some properties defined in terms of equations and coequations. Following the spirit of the original result by Eilenberg, we prove that there is a one-to-one correspondence between varieties of monoids and varieties of languages. Finally, we introduce the notion of *equational bisimulation* and a corresponding coinduction proof principle. For a given congruence relation C, we can show that a language satisfies C and hence belongs to the corresponding preformation of languages, by constructing a suitable equational bisimulation.

Related work

The algebra–coalgebra duality of diagram (1) is a modern rendering of the duality between *reachability* and *observability* of automata [8,9], which ultimately goes back to Kalman's duality between controllability and observability in system theory [10,11].

Our work builds on [12] and [13], using the combined algebra–coalgebra perspective on automata that was used there to give a new proof and various generalisations of Brzozowski's [14] minimisation algorithm. Our work is remotely related to [15], where the same perspective plays a role, albeit in a rather different manner. None of these papers, however, – nor for that matter any other paper we know of – discusses the relation between equations and coequations for automata.

We already mentioned that the automaton $\mathsf{free}(X,\alpha)$ is isomorphic to the transition monoid of (X,α), which is usually defined in terms of the function space X^X. We define $\mathsf{free}(X,\alpha)$ here by means of a product, because it allows us to define $\mathsf{cofree}(X,\alpha)$ using coproducts, making it the dual of the transition monoid. In Section 8, we shall discuss the connection between our work and the approach of algebraic language theory [2,6], where the notions of the *syntactic monoid* (first introduced by Rabin and Scott [16]) and congruence play a central role.

The way we have obtained the dual equivalence, namely, as a restriction of the (more generally defined) constructions of free and cofree – or, in other words, the constructions of the syntactic monoid and its dual – seems to be new. For the case of *finite automata*, our duality as such coincides with the use of Stone duality in [17, Theorem 1]. This is explained in some detail in Section 10. This last section moreover discusses how our work connects with the duality results appearing in Almeida [18,19], Pippenger [20], Gehrke [21,17] and Gehrke, Grigorieff and Pin [22]. Based on that discussion, Section 10 presents also some ideas for future research.

2. Preliminaries

Sets and languages

For a set X, we denote its cardinal by $|X|$. For sets X and Z we define $X^Z = \{g \mid g : Z \to X\}$. For sets X, Y, Z and functions $f : X \to Y$ we define $f^Z : X^Z \to Y^Z$ by $f^Z(g) = f \circ g$. We define the *image* and the *kernel* of a function $f : X \to Y$ by

$$\mathsf{im}(f) = \{y \in Y \mid \exists x \in X, \quad f(x) = y\}$$
$$\mathsf{ker}(f) = \{(x_1, x_2) \in X \times X \mid f(x_1) = f(x_2)\}$$

Let A be a (possibly infinite) alphabet, in all our examples fixed to $\{a, b\}$. We write A^* for the set of all finite sequences (words) over A. We denote the empty word by ε and the concatenation of two words v and w by vw. A *language* L over A is a subset $L \subseteq A^*$ and we denote the set of all languages over A by

$$2^{A^*} = \{L \mid L \subseteq A^*\}$$

(ignoring here and sometimes below the difference between subsets and characteristic functions). For a language $L \subseteq A^*$ and $a \in A$ we define the *a-derivative* of L by

$$L_a = \{v \in A^* \mid av \in L\}$$

and we define, more generally,

$$L_w = \{v \in A^* \mid wv \in L\}$$

In fact, L_a and L_w are also called *right derivatives* of L, in contrast to the *left derivative* of L, which we define by

$${}_aL = \{v \in A^* \mid va \in L\} \quad {}_wL = \{v \in A^* \mid vw \in L\}$$

One readily verifies that the operations $(\ )_w$ and ${}_w(\ )$ of right and left derivatives commute with the Boolean operations of (possibly infinite) union, intersection and complement, on languages.

Algebras and coalgebras

For a functor $H : \mathit{Set} \to \mathit{Set}$, an *H-algebra* is a pair (S,α) consisting of a set S and a function $\alpha : H(S) \to S$. An *H-coalgebra* is a pair (S,α) with $\alpha : S \to H(S)$. We will be considering algebras and coalgebras of the following specific functors:

$$F(S) = S^A$$
$$G(S) = S \times A$$
$$(2 \times F)(S) = 2 \times S^A$$
$$(1 + G)(S) = 1 + (S \times A)$$

Automata

An *automaton* is a pair (X, α) consisting of a (possibly infinite) set X of states and a transition function

$$\alpha : X \to X^A$$

In pictures, we use the following notation:

$\Leftrightarrow \quad \alpha(x)(a) = y$

We will also write $x_a = \alpha(x)(a)$ and, more generally,

$$x_\varepsilon = x \qquad x_{wa} = \alpha(x_w)(a)$$

We observe that automata are *F-coalgebras*. Because there is, for any A and X, an isomorphism

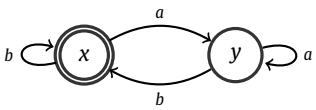
$$(\tilde{}) : (X \to X^A) \to ((X \times A) \to X) \qquad \tilde{\alpha}(x, a) = \alpha(x)(a)$$

automata are also *G-algebras* [23].

An automaton can be decorated by means of a *colouring* function

$$c : X \to 2$$

using a basic set of colours $2 = \{0, 1\}$. We call a state x *accepting* (or final) if $c(x) = 1$, and non-accepting if $c(x) = 0$. We call a triple (X, c, α) a *coloured* automaton. In pictures, we use a double circle to indicate that a state is accepting. For instance, in the following automaton



the state x is accepting and the state y is not. By pairing the functions c and α, we see that coloured automata are $(2 \times F)$-coalgebras:

$$\langle c, \alpha \rangle : X \to 2 \times X^A$$

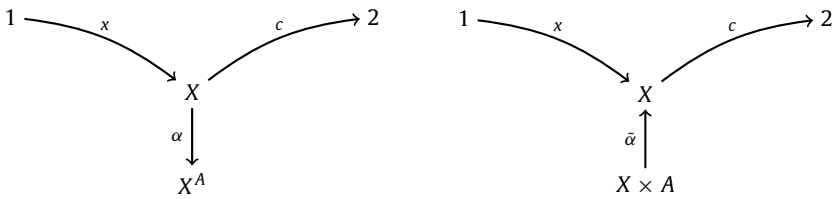
An automaton can also have an *initial state* $x \in X$, here represented by a function

$$x : 1 \to X$$

where $1 = \{0\}$. We call a triple (X, x, α) a *pointed* automaton. By pairing the functions x and $\tilde{\alpha}$, we see that pointed automata are $(1 + G)$-algebras:

$$[x, \tilde{\alpha}] : (1 + (X \times A)) \to X$$

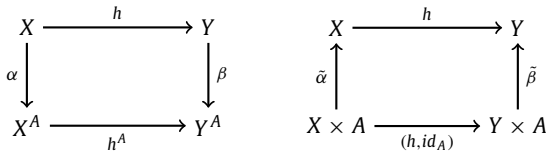
We call a 4-tuple (X, x, c, α) a *pointed and coloured automaton*. We could depict it by either of the following two diagrams



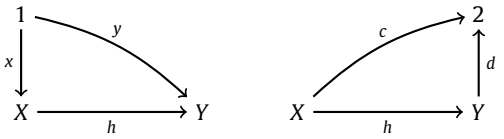
Depending on the context we will be using on diagram or the other, the choice is just a matter of personal preference. We observe further that pointed and coloured automata are simply called *automata* in most of the literature on automata theory. A pointed and coloured automaton (X, x, c, α) is neither an algebra nor a coalgebra – because of c and x, respectively – which can be a cause of fascination and confusion alike.

Homomorphisms, subautomata, bisimulations

A function $h : X \to Y$ is a *homomorphism* between automata (X, α) and (Y, β) if it makes the following equivalent diagrams commute:



An *epimorphism* is a homomorphism that is surjective, and a *monomorphism* is a homomorphism that is injective. A homomorphism of pointed automata (X, x, α) and (Y, y, β) and of coloured automata (X, c, α) and (Y, d, β) moreover respects initial values and colours, respectively:



If in the diagrams above $X \subseteq Y$, and (i) h is subset inclusion

$$h : X \subseteq Y$$

(and, moreover (ii) $x = y$ or (iii) $c = d$), then we call X a (i) *subautomaton* of Y (respectively (ii) *pointed* and (iii) *coloured subautomaton*). For an automaton (X, α) and $x \in X$, the *subautomaton generated by x*, denoted by

$$\langle x \rangle \subseteq X$$

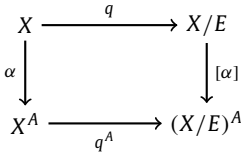
consists of the smallest subset of X that contains x and is closed under transitions. We call a relation $R \subseteq X \times Y$ a *bisimulation of automata* if for all $(x, y) \in X \times Y$,

$$(x, y) \in R \;\Rightarrow\; \forall a \in A, (x_a, y_a) \in R$$

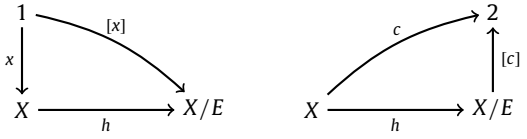
(where $x_a = \alpha(x)(a)$ and $y_a = \beta(y)(a)$). For pointed automata (X, x, α) and (Y, y, β), R is a *pointed bisimulation* if, moreover, $(x, y) \in R$. And for coloured automata (X, c, α) and (Y, d, β), R is a *coloured bisimulation* if, moreover, for all $(x, y) \in X \times Y$,

$$(x, y) \in R \;\Rightarrow\; c(x) = d(y)$$

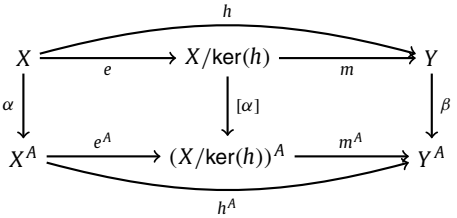
A bisimulation $E \subseteq X \times X$ is called a bisimulation *on* X. If E is an equivalence relation then we call it a *bisimulation equivalence*. The quotient map of a bisimulation equivalence on X is a homomorphism of automata:



with the obvious definitions of X/E, q and $[\alpha]$. If the equivalence E is a pointed bisimulation on (X, x, α) or a coloured bisimulation on (X, c, α), then we moreover require, respectively,



with, again, the obvious definitions of $[x]$ and $[c]$. For a homomorphism $h : X \to Y$, $\ker(h)$ is a bisimulation equivalence on X and $\mathrm{im}(h)$ is a subautomaton of Y. Any homomorphism h is equal to the composition of an epimorphism followed by a monomorphism, as follows:



with $e(x) = [x] = \{z \in X \mid h(z) = h(x)\}$, and $m([x]) = h(x)$. Note that $X/\ker(h) \cong \mathrm{im}(h)$. The pair (e, m) is called an *epi–mono factorisation* of h.

Congruence relations

A *right congruence* is an equivalence relation $E \subseteq A^* \times A^*$ such that, for all $(v, w) \in A^* \times A^*$,

$$(v, w) \in E \;\Rightarrow\; \forall u \in A^*, (vu,\ wu) \in E$$

A *left congruence* is an equivalence relation $E \subseteq A^* \times A^*$ such that, for all $(v, w) \in A^* \times A^*$,

$$(v, w) \in E \;\Rightarrow\; \forall u \in A^*, (uv,\ uw) \in E$$

We call E a *congruence* if it is both a right and a left congruence. Note that E is a right congruence iff it is a bisimulation equivalence on (A^*, σ).

Products and coproducts of automata

Automata (are both G-algebras and F-coalgebras and hence) have both products and coproducts, as follows.

- The *product* of two automata (X, α) and (Y, β) is given by $(X \times Y, \gamma)$ where $X \times Y$ is the Cartesian product and where
$$\gamma : (X \times Y) \to (X \times Y)^A \qquad \gamma((x, y))(a) = (\,\alpha(x)(a),\ \beta(y)(a)\,)$$
- The *coproduct* (or: sum) of two automata (X, α) and (Y, β) is given by $(X + Y, \gamma)$ where $X + Y$ is the disjoint union and where
$$\gamma : (X + Y) \to (X + Y)^A \qquad \gamma(z)(a) = \begin{cases} \alpha(z)(a) & \text{if } z \in X \\ \beta(z)(a) & \text{if } z \in Y \end{cases}$$

Pointed automata (are $(1 + G)$-algebras and hence) have products, as follows. The product of two pointed automata (X, x, α) and (Y, y, β) is given by $(X \times Y, (x, y), \gamma)$ with $(X \times Y, \gamma)$ as above and with initial state

$$(x, y) : 1 \to X \times Y$$

Coloured automata (are $(2 \times F)$-coalgebras and hence) have coproducts, as follows. The coproduct of two coloured automata (X, c, α) and (Y, d, β) is given by $(X + Y, [c, d], \gamma)$ with $(X + Y, \gamma)$ as above and with colouring function

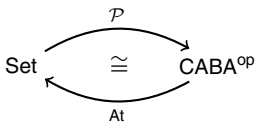
$$[c, d] : (X + Y) \to 2 \qquad [c, d](z) = \begin{cases} c(z) & \text{if } z \in X \\ d(z) & \text{if } z \in Y \end{cases}$$

All of the above binary (co)products can be easily generalised to arbitrary families of automata.

Complete atomic Boolean algebras

A Boolean algebra B is called *complete* if every subset has both a supremum and an infimum, with respect to the ordering defined by $a \leq b \Leftrightarrow a \wedge b = a$. An element $a \in B$ is called *atomic* whenever, for all $b \in A$: if $b \leq a$ then either $b = 0$ or $b = a$. A Boolean algebra B is called *atomic* if every element $b \in B$ can be expressed as the supremum of a (possibly infinite) set of atoms in B.

The class of all complete atomic Boolean algebras together with Boolean algebra homomorphisms forms a category CABA. Every complete atomic Boolean algebra B is isomorphic to $\mathcal{P}(S)$, for some set S. (As a consequence, the cardinality of a *finite* Boolean algebra, which is always complete and atomic, is a power of 2). More precisely, there exists the following dual equivalence between the category Set of sets and functions, and the category CABA:



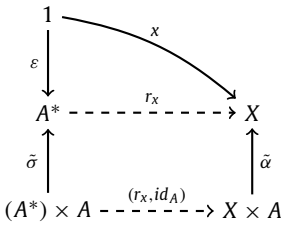
where the functor At maps a complete atomic Boolean algebra to its set of atoms.

3. Setting the scene

The set A^* forms a pointed automaton $(A^*, \varepsilon, \sigma)$ with initial state ε and transition function σ defined by

$$\sigma : A^* \to (A^*)^A \qquad \sigma(w)(a) = wa \tag{2}$$

It is *initial* in the following sense: for any given automaton (X, α), every choice of initial state $x : 1 \to X$ induces a unique function $r_x : A^* \to X$, given by $r_x(w) = x_w$, that makes the following diagram commute:



This property makes $(A^*, \varepsilon, \sigma)$ an *initial* $(1+G)$*-algebra*. Equivalently, the automaton (A^*, σ) is a *G-algebra that is free on the set* 1. The function r_x maps a word w to the state x_w reached from the initial state x on input w and is therefore called the *reachability* map for (X, x, α).

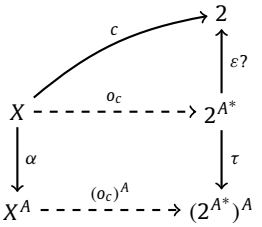
The set 2^{A^*} of languages forms a coloured automaton $(2^{A^*}, \varepsilon?, \tau)$ with colouring function $\varepsilon?$ defined by

$$\varepsilon?\colon 2^{A^*} \to 2 \qquad \varepsilon?(L) = \begin{cases} 1 & \text{if } \varepsilon \in L \\ 0 & \text{if } \varepsilon \notin L \end{cases}$$

and transition function τ defined by

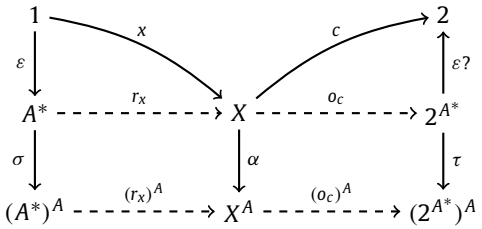
$$\tau\colon 2^{A^*} \to (2^{A^*})^A \qquad \tau(L)(a) = L_a \tag{3}$$

It is *final* in the following sense: for any given automaton (X, α), every choice of colouring function $c : X \to 2$ induces a unique function $o_c : X \to 2^{A^*}$, given by $o_c(x) = \{w \mid c(x_w) = 1\}$, that makes the following diagram commute:



This property makes $(2^{A^*}, \varepsilon?, \tau)$ a *final* $(2 \times F)$*-coalgebra*. Equivalently, the automaton $(2^{A^*}, \tau)$ is an F*-coalgebra that is cofree on the set* 2. The function o_c maps a state x to the language $o_c(x)$ accepted by x. Since the language $o_c(x)$ can be viewed as the observable behaviour of x, the function o_c is called the *observability* map.

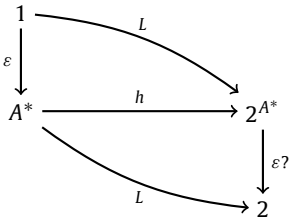
Summarising, we have set the following scene for our investigations:



(4)

If the reachability map r_x is *surjective* then we call (X, x, α) *reachable*. If the observability map o_c is *injective* then we call (X, c, α) *observable*. And if r_x is surjective *and* o_c is injective then we call (X, x, c, α) (reachable and observable, or:) *minimal*.

Fixing the language $L \in 2^{A^*}$, we obtain the following variation of the picture above:



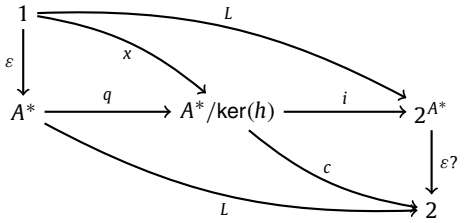
where the lower L is in fact the characteristic function of $L \subseteq A^*$, and where the homomorphism h satisfies $h(w) = L_w$. As a consequence, we have

$$h(v) = h(w) \Leftrightarrow v \equiv_{\mathsf{MN}} w$$

where on the right, we have the celebrated *Myhill–Nerode* equivalence, defined by

$$v \equiv_{\mathsf{MN}} w \Leftrightarrow \forall u \in A^*,\ vu \in L \Leftrightarrow wu \in L$$

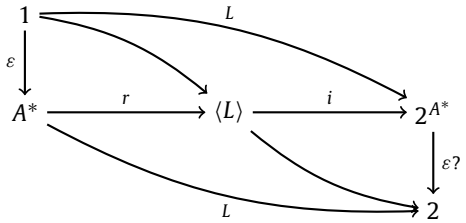
A *minimal automaton accepting* L is now obtained by the epi–mono factorisation of h:



where $x = q \circ \varepsilon$ and $c = \varepsilon? \circ i$. This minimal automaton is unique up-to isomorphism because epi–mono factorisations are. And because $A^*/\mathsf{ker}(h) \cong \mathsf{im}(h)$, it is equal to

$$\langle L \rangle \subseteq 2^{A^*}$$

that is, the subautomaton of $(2^{A^*}, \tau)$ generated by L. All in all we have obtained the following picture:



(5)

with $r(w) = L_w$ and $i(K) = K$, for all $w \in A^*$ and $K \in \langle L \rangle$. In this case, $\mathsf{ker}(r) = \equiv_{\mathsf{MN}}$.

In conclusion of this section, we observe that $\langle L \rangle$ is finite iff the language L is *rational*. This fact is a version [14,24] of Kleene's correspondence between finite automata and rational languages [25].

4. Equations and coequations

We will be referring to the situation of (4).

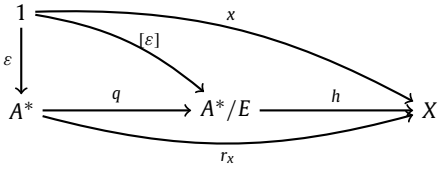
Definition 1 *(Equations).* A *set of equations* is a bisimulation equivalence relation $E \subseteq A^* \times A^*$ on the automaton (A^*, σ). We define $(X, x, \alpha) \models E$ – and say: *the pointed automaton* (X, x, α) *satisfies* E – by

$$(X, x, \alpha) \models E \Leftrightarrow \forall (v, w) \in E,\ x_v = x_w$$

Because

$$\forall (v, w) \in E,\ x_v = x_w \Leftrightarrow E \subseteq \mathsf{ker}(r_x)$$

we have, equivalently, that $(X, x, \alpha) \models E$ iff the reachability map r_x factors through A^*/E:



where the homomorphisms (of pointed automata) q and h are given by

$$q(w) = [w] \qquad h([w]) = r_x(w)$$

We define $(X, \alpha) \models E$ – and say: *the automaton* (X, α) *satisfies* E – by

$$(X, \alpha) \models E \Leftrightarrow \forall x : 1 \to X, \quad (X, x, \alpha) \models E$$
$$\Leftrightarrow \forall x \in X,\ \forall (v, w) \in E,\ x_v = x_w \qquad \square$$

Note that we consider *sets* of equations E and that $(v, w) \in E$ implies $(vu, wu) \in E$, for all $v, w, u \in A^*$, because E is – by definition – a bisimulation relation on (A^*, σ). Still we shall sometimes consider also *single* equations $(v, w) \in A^* \times A^*$ and use shorthand such as

$$(X, \alpha) \models v = w$$

to denote

$$(X, \alpha) \models \mathbf{v = w}$$

where $\mathbf{v = w}$ is defined as the smallest bisimulation equivalence on A^* containing (v, w). Furthermore, we shall use also variations such as

$$(X, \alpha) \models \{v = w,\ t = u\} \Leftrightarrow (X, \alpha) \models v = w \wedge (X, \alpha) \models t = u$$

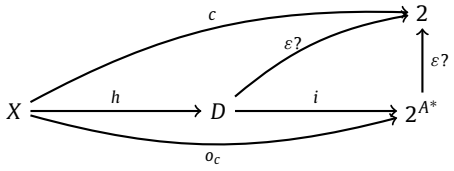
Definition 2 *(Coequations).* A *set of coequations* is a subautomaton $D \subseteq 2^{A^*}$ of the automaton $(2^{A^*}, \tau)$. We define $(X, c, \alpha) \models D$ – and say: *the coloured automaton* (X, c, α) *satisfies* D – by

$$(X, c, \alpha) \models D \Leftrightarrow \forall x \in X,\ o_c(x) \in D$$

Because

$$\forall x \in X,\ o_c(x) \in D \Leftrightarrow \mathsf{im}(o_c) \subseteq D$$

we have, equivalently, that $(X, c, \alpha) \models D$ iff the observability map o_c factors through D:



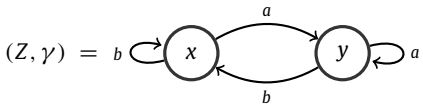
where the homomorphisms (of coloured automata) h and i are given by

$$h(x) = o_c(x) \qquad i(L) = L$$

We define $(X, \alpha) \models D$ – and say: *the automaton* (X, α) *satisfies* D – by

$$\begin{aligned}(X, \alpha) \models D &\Leftrightarrow \forall c : X \to 2, \quad (X, c, \alpha) \models D \\ &\Leftrightarrow \forall c : X \to 2,\ \forall x \in X,\ o_c(x) \in D \qquad \square\end{aligned}$$

Example 3. We consider the automaton (Z, γ) defined by the following diagram:



Here are some examples of equations:

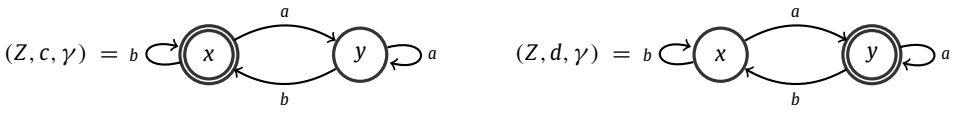
$$(Z, x, \gamma) \models \{b = \varepsilon,\ ab = \varepsilon,\ aa = a\}$$

$$(Z, y, \gamma) \models \{a = \varepsilon,\ ba = \varepsilon,\ bb = b\}$$

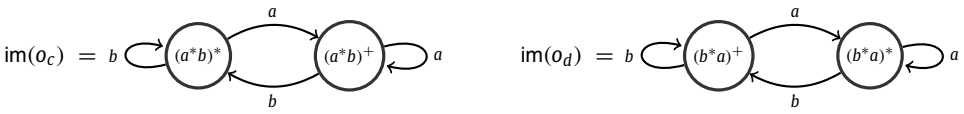
Taking the intersection of the (bisimulation equivalences generated by) these sets, we obtain that

$$(Z, \gamma) \models \{aa = a,\ bb = b,\ ab = b,\ ba = a\}$$

The above set of equations or, again more precisely, the bisimulation equivalence relation on (A^*, σ) generated by it, is the largest set of equations satisfied by (Z, γ). For examples of coequations, we consider the following 2 (out of all 4 possible) coloured versions of (Z, γ):



(Thus $c(x) = 1$, $c(y) = 0$, $d(x) = 0$ and $d(y) = 1$.) The observability mappings o_c and o_d map these automata to



It follows that

$$(Z, c, \gamma) \models \{(a^*b)^*,\ (a^*b)^+\} \qquad (Z, d, \gamma) \models \{(b^*a)^*,\ (b^*a)^+\} \qquad \square$$

5. Free and cofree automata

Let (X, α) be an arbitrary automaton. We show how to construct an automaton that corresponds to the *largest set of equations* satisfied by (X, α). And, dually, we construct an automaton that corresponds to the *smallest set of coequations* satisfied by (X, α).

Definition 4. Let $X = \{x_i \mid i \in I\}$ be the set of states of an automaton (X, α). We define a pointed automaton $\mathsf{free}(X, \alpha)$ in two steps, as follows:

(i) First, we take the product of the pointed automata (X, x_i, α) that we obtain by letting the initial element x_i range over X. This yields a pointed automaton $(\Pi X, \bar{x}, \bar{\alpha})$ with

$$\Pi X = \prod_{x : 1 \to X} X_x \cong X^{|X|}$$

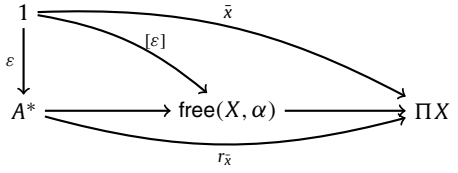
(where $X_x = X$), with $\bar{x} = (x_i)_{i \in I}$, and with $\bar{\alpha} : \Pi X \to (\Pi X)^A$ defined component-wise

$$\bar{\alpha}((y_i)_{i \in I})(a) = ((y_i)_a)_{i \in I}$$

(ii) Next we consider the reachability map $r_{\bar{x}} \colon A^* \to \Pi X$ and define:

$$\mathsf{Eq}(X, \alpha) = \ker(r_{\bar{x}}) \qquad \mathsf{free}(X, \alpha) = A^*/\mathsf{Eq}(X, \alpha)$$

This yields the pointed automaton $(\mathsf{free}(X, \alpha), [\varepsilon], [\sigma])$:



Note that $\mathsf{free}(X, \alpha) \cong \mathsf{im}(r_{\bar{x}})$. $\square$

Definition 5. Let $X = \{x_i \mid i \in I\}$ be the set of states of an automaton (X, α). We define a coloured automaton $\mathsf{cofree}(X, \alpha)$ in two steps, as follows:

(i) First, we take the coproduct of the $2^{|X|}$ coloured automata (X, c, α) that we obtain by letting c range over the set $X \to 2$ of all colouring functions. This yields a coloured automaton $(\Sigma X, \hat{c}, \hat{\alpha})$ with

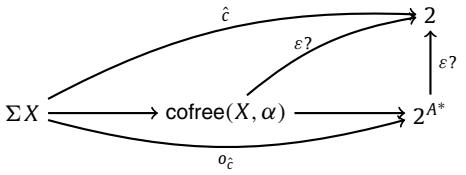
$$\Sigma X = \sum_{c\colon\ X \to 2} X_c$$

(where $X_c = X$), and with $\hat{c}$ and $\hat{\alpha}$ defined component-wise.

(ii) Next we consider the observability map $o_{\hat{c}} \colon \Sigma X \to 2^{A^*}$ and define:

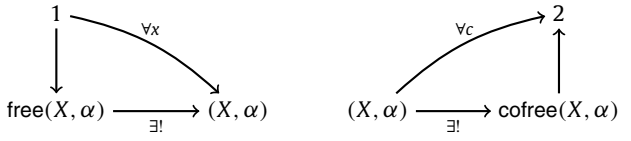
$$\mathsf{coEq}(X, \alpha) = \mathsf{im}(o_{\hat{c}}) \qquad \mathsf{cofree}(X, \alpha) = \mathsf{coEq}(X, \alpha)$$

This yields the coloured automaton $(\mathsf{cofree}(X, \alpha), \varepsilon?, \tau)$:



Note that $\mathsf{cofree}(X,\alpha) \cong \Sigma X/\ker(o_{\hat{c}})$. □

The automata $\mathsf{free}(X,\alpha)$ and $\mathsf{cofree}(X,\alpha)$ are *free* and *cofree* on (X,α), respectively, because of the following universal properties:



For every point $x\colon 1 \to X$ there exists a unique homomorphism from $\mathsf{free}(X,\alpha)$ to (X,x,α), given by the "x-th" projection from the product ΠX to X. Dually, for every colouring $c\colon X \to 2$, there exists a unique homomorphism from (X,c,α) to $\mathsf{cofree}(X,\alpha)$, given by the "$c$-th" embedding of X into the coproduct ΣX.

The main raison d'être for the constructions of free and cofree is that they represent the sets $\mathsf{Eq}(X,\alpha)$ and $\mathsf{coEq}(X,\alpha)$, which are, by construction, the largest set of equations and the smallest set of coequations satisfied by (X,α).

Proposition 6. *The set* $\mathsf{Eq}(X,\alpha)$ *is the* largest *set of equations satisfied by* (X,α)*:*

$$\mathsf{Eq}(X,\alpha) = \bigcup \{E \subseteq A^* \times A^* \mid E \text{ is a set of equations and } (X,\alpha) \models E\}$$

The set $\mathsf{coEq}(X,\alpha)$ *is the* smallest *set of coequations satisfied by* (X,α)*:*

$$\mathsf{coEq}(X,\alpha) = \bigcap \{D \subseteq 2^{A^*} \mid D \text{ is a set of coequations and } (X,\alpha) \models D\} \qquad \square$$

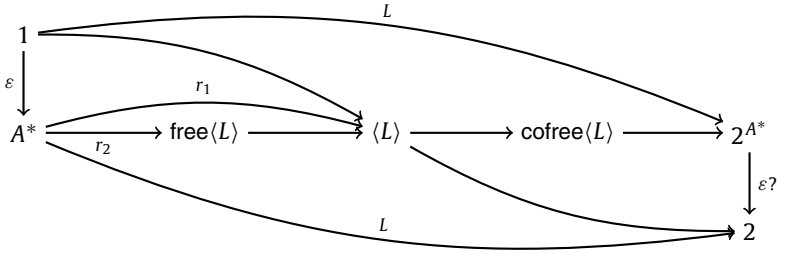
Proposition 7. *The set of equations* $\mathsf{Eq}(X,\alpha)$ *is a congruence on* A^*.

Proof. We already know that $\mathsf{Eq}(X,\alpha)$ is a right-congruence. Let $(v,w) \in \mathsf{Eq}(X,\alpha)$ and $u \in A^*$. For a state $x \in X$, we have

$$x_{uv} = (x_u)_v = (x_u)_w = x_{uw}$$

(since equations $(v,w) \in \mathsf{Eq}(X,\alpha)$ hold in *all* states of X). It follows that $(uv,uw) \in \mathsf{Eq}(X,\alpha)$ and we conclude that $\mathsf{Eq}(X,\alpha)$ is a congruence. □

Applying the picture above to the minimal automaton $\langle L\rangle$ of a given language $L \in 2^{A^*}$ yields the following refinement of (5):



(6)

We already saw in (5) that $\ker(r_1) = \equiv_{\mathsf{MN}}$, the Myhill–Nerode equivalence for L. Furthermore, it follows from Proposition 6 and Proposition 7 that

$$\mathsf{Eq}\langle L\rangle = \ker(r_2) = \equiv_L \tag{7}$$

where $\equiv_L$ is the so-called *syntactic congruence* of L, which is defined, for all $v,w \in A^*$, by

$$v \equiv_L w \quad \text{if and only if} \quad \forall u_1,u_2 \in A^*,\ (u_1 v u_2 \in L \Leftrightarrow u_1 w u_2 \in L)$$

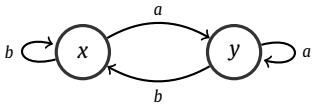
Corollary 8. *For a language $L \in 2^{A^*}$, the congruences $\mathsf{Eq}\langle L\rangle$ and $\equiv_L$ coincide.*

Proof. Let $(v, w) \in \mathsf{Eq}\langle L\rangle$ and let u_1 be an arbitrary word in A^*. The language L_{u_1} is in $\langle L\rangle$ and satisfies the equation $L_{u_1 v} = L_{u_1 w}$, that is, for any word $u_2 \in A^*$,

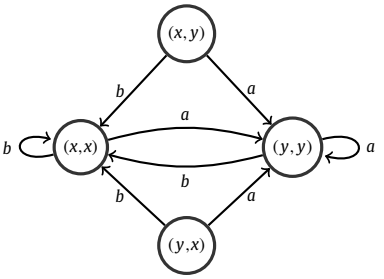
$$(u_2 \in L_{u_1 v} \Leftrightarrow u_2 \in L_{u_1 w}) \qquad \text{equivalently,} \qquad (u_1 v u_2 \in L \Leftrightarrow u_1 w u_2 \in L)$$

that is, $(v, w) \in \equiv_L$. The other inclusion is proved similarly. □

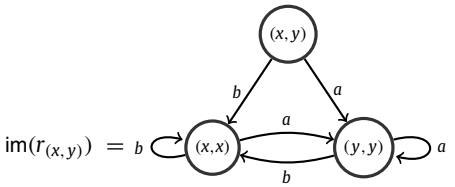
Example 9 *(Example 3 continued).* We consider our previous example

$(Z, \gamma) =$ 

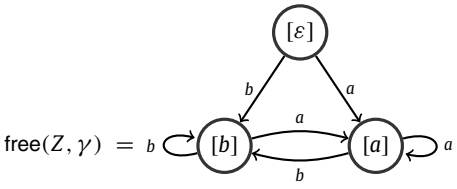
The product of (Z, x, γ) and (Z, y, γ) is:

$(\Pi Z, (x, y), \gamma) =$ 

Taking $\mathsf{im}(r_{(x,y)})$ yields the part that is reachable from (x, y):

$\mathsf{im}(r_{(x,y)}) =$ 

We know that $\mathsf{free}(Z, \gamma) \cong \mathsf{im}(r_{(x,y)})$, which leads to the following isomorphic automaton:

$\mathsf{free}(Z, \gamma) =$ 

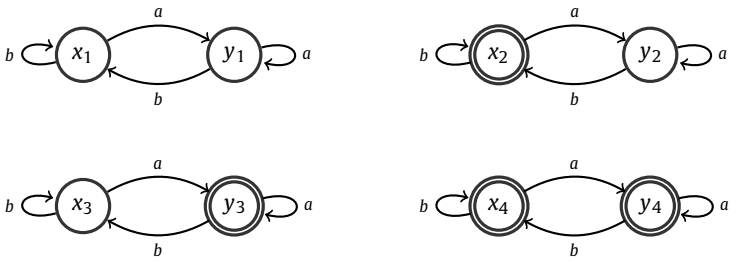
Since $\mathsf{free}(Z, \gamma) = A^*/\mathsf{Eq}(Z, \gamma)$, we can deduce from the above automaton that $\mathsf{Eq}(Z, \gamma)$ consists of

$$\mathsf{Eq}(Z, \gamma) = \{aa = a,\ bb = b,\ ab = b,\ ba = a\}$$

where the set on the right represents the smallest bisimulation equivalence – in fact, a congruence – on (A^*, σ). The set $\mathsf{Eq}(Z, \gamma)$ is the largest set of equations satisfied by (Z, γ).

Next we turn to coequations. The coproduct of all 4 coloured versions of (Z, γ) is

$(\Sigma Z, \hat{c}, \hat{\gamma}) =$

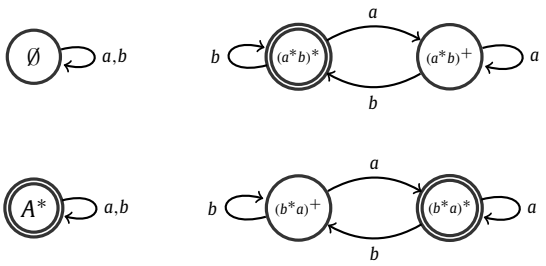


The observability map $o_{\hat{c}} : \Sigma Z \to 2^{A^*}$ is given by

$o_{\hat{c}}(x_1)$	$o_{\hat{c}}(y_1)$	$o_{\hat{c}}(x_2)$	$o_{\hat{c}}(y_2)$	$o_{\hat{c}}(x_3)$	$o_{\hat{c}}(y_3)$	$o_{\hat{c}}(x_4)$	$o_{\hat{c}}(y_4)$
$\emptyset$	$\emptyset$	$(a^*b)^*$	$(a^*b)^+$	$(b^*a)^+$	$(b^*a)^*$	A^*	A^*

Since $\mathsf{cofree}(Z, \gamma) = \mathsf{im}(o_{\hat{c}})$, this yields

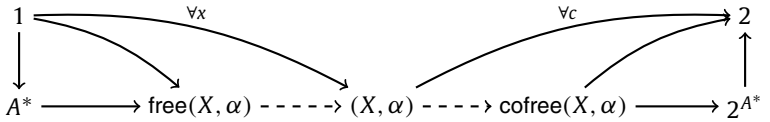
$\mathsf{cofree}(Z, \gamma) =$



(8)

The set of states of this automaton is $\mathsf{cofree}(Z, \gamma)$, which is the smallest set of coequations satisfied by (Z, γ). □

Summarising the present section, we have obtained, for every automaton (X, α), the following refinement of our previous scene (4):



(9)

The automata $\mathsf{free}(X, \alpha)$ and $\mathsf{cofree}(X, \alpha)$ represent the largest set of equations and the smallest set of coequations satisfied by (X, α). As we mentioned earlier, all of this applies to *infinite* X as well.

6. A dual equivalence

In this section, we shall first show that – when suitably restricted – the constructions of free and cofree are in fact functorial, that is, they act not only on automata but also on homomorphisms. Next we shall see that by restricting the functors free and cofree further still, they turn out to form a dual equivalence.

We will be using the following categories:

$\mathcal{A}$: the category of automata (X, α) and automata homomorphisms
$\mathcal{A}_m$: the category of automata (X, α) and automata monomorphisms
$\mathcal{A}_e$: the category of automata (X, α) and automata epimorphisms

As it turns out, we can extend the definitions of free and cofree to monomorphisms and epimorphisms, respectively, such that we obtain functors of the following type:

$$\mathsf{free} \colon \mathcal{A}_m \to (\mathcal{A}_e)^{\mathsf{op}} \qquad \mathsf{cofree} \colon \mathcal{A}_e \to (\mathcal{A}_m)^{\mathsf{op}}$$

Here the superscript op indicates a reversal of arrows: for monomorphisms,

$$(X, \alpha) \xrightarrow{\ m\ } (Y, \beta) \qquad \xmapsto{\ \mathsf{free}\ } \qquad \mathsf{free}(Y, \beta) \xrightarrow{\ \mathsf{free}(m)\ } \mathsf{free}(X, \alpha)$$

where $\mathsf{free}(m)$ is defined by

$$\mathsf{free}(m)([w]_{\mathsf{Eq}(Y,\beta)}) = [w]_{\mathsf{Eq}(X,\alpha)}$$

Because m is a monomorphism, we have $\mathsf{Eq}(Y,\beta) \subseteq \mathsf{Eq}(X,\alpha)$, which implies that $\mathsf{free}(m)$ is a well-defined epimorphism. Similarly, for epimorphisms,

$$(X,\alpha) \xrightarrow{\ e\ } (Y,\beta) \qquad \overset{\mathsf{cofree}}{\longmapsto} \qquad \mathsf{cofree}(Y,\beta) \xrightarrow{\ \mathsf{cofree}(e)\ } \mathsf{cofree}(X,\alpha)$$

where $\mathsf{cofree}(e)$ is just set inclusion. Because e is an epimorphism, we have $\mathsf{coEq}(Y,\beta) \subseteq \mathsf{coEq}(X,\alpha)$, which implies that $\mathsf{cofree}(e)$ is a well-defined monomorphism.

6.1. The first theorem: congruence quotients

Next we introduce the category $\mathcal{C}$ of *congruence quotients*, which is defined as follows:

$$\mathsf{objects}(\mathcal{C}) = \{(A^*/C, [\sigma]) \mid C \subseteq A^* \times A^* \text{ is a congruence relation}\}$$

$$\mathsf{arrows}(\mathcal{C}) = \{e\colon A^*/C \to A^*/D \mid e \text{ is an epimorphism of automata}\}$$

where $[\sigma]$ corresponds to the quotient transition derived from the transition introduced in (2). We observe that $\mathcal{C}$ is a subcategory of $\mathcal{A}_e$ and that it is in fact a set: $\mathcal{C}$ is isomorphic to the set of all congruence relations on A^*, together with set inclusion. That is, there exists a (unique) epimorphism $e\colon A^*/C \to A^*/D$ if and only if $C \subseteq D$.

Since congruence quotients come equipped with a canonical choice of transition function, that is, $[\sigma]$, we shall often simply write A^*/C for $(A^*/C, [\sigma])$.

Theorem 10. $\mathsf{free}(\mathcal{A}_m) = \mathcal{C}^{op}$

Proof. For every automaton (X,α), $\mathsf{free}(X,\alpha) = A^*/\mathsf{Eq}(X,\alpha)$ is a congruence, by Proposition 7. For the reverse inclusion, consider a congruence $C \subseteq A^* \times A^*$. One readily shows that

$$\mathsf{Eq}(A^*/C) = C$$

which implies $\mathsf{free}(A^*/C) = A^*/\mathsf{Eq}(A^*/C) = A^*/C$. This proves the theorem for objects. For arrows, we already saw that free maps a monomorphism to an epimorphism of congruence quotients. Conversely, let $e : A^*/C \to A^*/D$ be an epimorphism. We define

$$m\colon A^*/D \to (A^*/C + A^*/D)$$

where $+$ denotes the disjoint union of automata. Because

$$\mathsf{Eq}(A^*/C + A^*/D) = C \cap D$$

and because $C \subseteq D$, it follows that $\mathsf{free}(A^*/C + A^*/D) = A^*/(C \cap D) = A^*/C$, which implies that $\mathsf{free}(m) = e$. □

6.2. The second theorem: preformations of languages

We will be using the following notion of a *preformation of languages*.

Definition 11. A preformation of languages is a set $V \subseteq 2^{A^*}$ such that:

(i) V is a complete atomic Boolean subalgebra of 2^{A^*}.
(ii) for all $L \in 2^{A^*}$: if $L \in V$ then for all $a \in A$, both $L_a \in V$ and ${}_aL \in V$.

We note that, being a subalgebra of 2^{A^*}, a preformation V always contains both $\emptyset$ and A^*. □

Next we define the *category* $\mathcal{PL}$ of preformations of languages, as follows:

$$\mathsf{objects}(\mathcal{PL}) = \{(V,\tau) \mid V \subseteq 2^{A^*} \text{ is a preformation of languages}\},$$

$$\mathsf{arrows}(\mathcal{PL}) = \{m\colon V \to W \mid m \text{ is an monomorphism of automata}\},$$

where τ corresponds to the transition introduced in (3). The category $\mathcal{PL}$ is a subcategory of $\mathcal{A}_m$; furthermore, $\mathcal{PL}$ is in fact a set and the arrows in $\mathcal{PL}$ are just set inclusion. Since preformations of languages come equipped with a canonical choice of transition function, that is, τ (right-derivatives of languages), we shall often simply write V for (V,τ).

The main result of this subsection will be that

$$\mathsf{cofree}(\mathcal{C}) = (\mathcal{PL})^{\mathsf{op}}$$

which we shall prove in several steps.

We begin with an elementary but useful property of colourings, which uses the following definition. For an automaton (X, α) and state $x \in X$, we define the following ("one-point") colouring:

$$\delta_x : X \to 2, \qquad \delta_x(y) = 1 \Leftrightarrow x = y$$

Lemma 12. *For every automaton (X, α), state $y \in X$ and colouring $c : X \to 2$,*

$$o_c(y) = \bigcup \{ o_{\delta_x}(y) \mid x \in X \text{ and } c(x) = 1 \}$$

The states of congruence quotients are equivalence classes of words $w \in A^*$, that is, languages $[w] \subseteq A^*$. The following lemma shows that each of them occurs as the observable behaviour of the initial state $[\varepsilon]$, under the corresponding one-point colouring.

Lemma 13. *For every congruence quotient $A^*/C \in \mathcal{C}$ and every $[w] \in A^*/C$,*

$$o_{\delta_{[w]}}([\varepsilon]) = [w]$$

Proof. For all $v \in A^*$,

$$v \in o_{\delta_{[w]}}([\varepsilon]) \Leftrightarrow \delta_{[w]}([\varepsilon]_v) = 1 \Leftrightarrow [\varepsilon]_v = [w] \Leftrightarrow [v] = [w] \Leftrightarrow v \in [w] \qquad \square$$

The following lemma shows that all the observable behaviour of a congruence quotient stems from its initial state.

Lemma 14. *For every congruence quotient $A^*/C \in \mathcal{C}$ and every $L \in \mathsf{coEq}(A^*/C)$, there exists a colouring $c\colon A^*/C \to 2$ with*

$$o_c([\varepsilon]) = L$$

Proof. If $L \in \mathsf{coEq}(A^*/C)$ then there exist a state $[w] \in A^*/C$ and a colouring $d\colon A^*/C \to 2$ with $o_d([w]) = L$. We define a new colouring $c\colon A^*/C \to 2$, for all $[v] \in A^*/C$, by

$$c([v]) = d([w]_v)$$

Note that c is well-defined because C is a (left) congruence on A^*. It now follows that

$$v \in o_c([\varepsilon]) \Leftrightarrow c([\varepsilon]_v) = 1 \Leftrightarrow c([v]) = 1 \Leftrightarrow d([w]_v) = 1 \Leftrightarrow v \in o_d([w]) \Leftrightarrow v \in L$$

which concludes the proof. $\square$

Combining the above, we obtain the following characterisation.

Proposition 15. *For every congruence quotient $A^*/C \in \mathcal{C}$,*

$$\mathsf{coEq}(A^*/C) = \{ L \in 2^{A^*} \mid L = \bigcup V \text{ for some } V \subseteq A^*/C \}$$

As a consequence,

$$\mathsf{coEq}(A^*/C) \cong \mathcal{P}(A^*/C)$$

Proof. There is a trivial one-to-one correspondence between colourings $c : A^*/C \to 2$ and subsets $V \subseteq A^*/C$ given by $V_c = c^{-1}(1)$. Using Lemma 12 and Lemma 13, we obtain, as a consequence, that

$$\begin{aligned} o_c([\varepsilon]) &= \bigcup \{ o_{\delta_K}([\varepsilon]) \mid K \in A^*/C \text{ and } c(K) = 1 \} \\ &= \bigcup \{ K \mid K \in A^*/C \text{ and } c(K) = 1 \} \\ &= \bigcup V_c \end{aligned}$$

The first equality of the proposition now follows from Lemma 14. Since the languages $L \in A^*/C$ form a partitioning of A^*, the second identity (isomorphism) follows. $\square$

We are ready to prove the following.

Proposition 16. *For every congruence quotient $A^*/C \in \mathcal{C}$,*

$\mathsf{coEq}(A^*/C)$ *is a preformation of languages*

with A^*/C as the set of atoms.

Proof. It follows from Proposition 15 that $\mathsf{coEq}(A^*/C)$ is a complete atomic Boolean algebra, with A^*/C as the set of atoms, and containing A^* and $\emptyset$.

Because $\mathsf{coEq}(A^*/C)$ is a subautomaton of $(2^{A^*}, \tau)$, it is closed under right derivatives.

In order to prove that it is also closed under left derivatives, consider $L \in \mathsf{coEq}(A^*/C)$ and $w \in A^*$. By Lemma 14, there exists a colouring $c : A^*/C \to 2$ with $L = o_c([\varepsilon])$. We define a new colouring $c_w : A^*/C \to 2$, for $[v] \in A^*/C$, by

$$c_w([v]) = c([vw])$$

(Note that c_w is well-defined because C is a (left) congruence on A^*.) Because

$$v \in o_{c_w}([\varepsilon]) \Leftrightarrow c_w([v]) = 1 \Leftrightarrow c([vw]) = 1 \Leftrightarrow vw \in L \Leftrightarrow v \in {}_wL$$

it follows that $o_{c_w}([\varepsilon]) = {}_wL$. And because $o_{c_w}([\varepsilon])$ is in $\mathsf{coEq}(A^*/C)$, so is ${}_wL$. □

Still on our way towards a proof of $\mathsf{cofree}(\mathcal{C}) = (\mathcal{PL})^{\mathsf{op}}$, let us next fix a preformation of languages $V \in \mathcal{PL}$ and show that it is the image under cofree of a congruence quotient on A^*. To this end, we define the following mapping:

$$\eta : A^* \to \mathsf{At}(V) \qquad \eta(w) = \text{ the unique atom } L \in V \text{ with } w \in L$$

Because V is a complete atomic Boolean algebra containing A^*, η is well-defined and surjective. We shall show next that it is a congruence quotient of A^*.

Lemma 17. *The set* $\ker(\eta)$ *is a congruence on* A^* *and hence* η *is a congruence quotient*

$$\eta : (A^*, \sigma) \to (\mathsf{At}(V), [\sigma])$$

Proof. It suffices to show that, for all $v, w \in A^*$, if $\eta(v) = \eta(w)$ then, for all $u \in A^*$,

$$\eta(uv) = \eta(uw) \quad \text{and} \quad \eta(vu) = \eta(wu)$$

In order to prove the first equality, we assume $\eta(v) = \eta(w)$ and consider $\eta(uv)$. Because $uv \in \eta(uv)$ we have $v \in \eta(uv)_u$. Because V is closed under right derivatives, $\eta(uv)_u \in V$ and because V is atomic, we have $\eta(v) \subseteq \eta(uv)_u$. We have the following sequence of implications:

$$\eta(v) \subseteq \eta(uv)_u \Rightarrow \eta(w) \subseteq \eta(uv)_u \Rightarrow w \in \eta(uv)_u \Rightarrow uw \in \eta(uv) \Rightarrow \eta(uw) \subseteq \eta(uv)$$

The same argument will prove $\eta(uv) \subseteq \eta(uw)$, which proves the first equality. The second equality follows by the same argument, using left instead of right derivatives. □

There is also the following.

Lemma 18. $\mathsf{Eq}(V) = \ker(\eta)$.

Proof. We have to show, for all $v, w \in A^*$, that

$$(\text{ for all } L \in V : L_v = L_w) \Leftrightarrow \eta(v) = \eta(w)$$

From $\varepsilon \in \eta(v)_v = \eta(v)_w$ it follows that $w \in \eta(v)$ and hence $\eta(v) = \eta(w)$, which proves the above implication from left to right.

For the implication from right to left, assume $\eta(v) = \eta(w)$. Since V is a complete atomic Boolean algebra, it suffices to prove that $L_v = L_w$ for $L \in \mathsf{At}(V)$, since (right) derivatives commute with unions. So consider $u \in A^*$ and $\eta(u) \in \mathsf{At}(V)$. For all $x \in A^*$,

$$x \in \eta(u)_v \Rightarrow vx \in \eta(u) \Rightarrow \eta(u) = \eta(vx) \Rightarrow \eta(u) = \eta(wx) \Rightarrow x \in \eta(u)_w$$

where the last but one implication follows from Lemma 17. This proves $\eta(u)_v \subseteq \eta(u)_w$. The same argument proves the reverse inclusion, which concludes the proof. □

Combining the two lemmas above now gives the following.

Proposition 19. $\mathsf{free}(V) = (\mathsf{At}(V), [\sigma])$

Proof.

$$\mathsf{free}(V) = (A^*/\mathsf{Eq}(V), [\sigma]) = (A^*/\ker(\eta), [\sigma]) = (\mathsf{At}(V), [\sigma]) \qquad \square$$

Corollary 20. $\mathsf{cofree} \circ \mathsf{free}(V) = V$

Proof. By Proposition 19, $\mathsf{cofree} \circ \mathsf{free}(V) = \mathsf{cofree}(\mathsf{At}(V), [\sigma])$. And by Proposition 15, $\mathsf{cofree}(\mathsf{At}(V), [\sigma]) = V$. □

Finally, we obtain the main result of this subsection.

Theorem 21. $\mathsf{cofree}(\mathcal{C}) = (\mathcal{PL})^{op}$

Proof. The identity holds for objects, by Proposition 16 and Corollary 20. Furthermore, every epimorphism of congruence quotients is mapped by cofree to the reversed inclusion of the corresponding preformations, and conversely, every inclusion of preformations is easily seen to stem from an epimorphism of congruence quotients. □

6.3. The main theorem: free and cofree form a dual equivalence

We have obtained the following dual equivalence.

Theorem 22. *The category $\mathcal{C}$ of congruence quotients is dually equivalent to the category $\mathcal{PL}$ of preformations of languages via the functors* free *and* cofree. *That is,*

$$\mathsf{cofree} : \mathcal{C} \cong (\mathcal{PL})^{op} : \mathsf{free}$$

Proof. For a preformation of languages V,

$$\mathsf{cofree} \circ \mathsf{free}(V) = V$$

by Corollary 20. For a congruence quotient A^*/C, we have

$$\mathsf{free} \circ \mathsf{cofree}(A^*/C) = \mathsf{At}(\mathsf{cofree}(A^*/C)) = A^*/C$$

by Proposition 19 and Proposition 16, respectively. This proves the theorem for objects. One readily shows that this correspondence extends to arrows as well. □

As a consequence of our Theorem 22 we deduce the following corollary

Corollary 23. *For every congruence C in A^* and every language L in 2^{A^*},*

$$L \in \mathsf{coEq}(A^*/C) \quad \Leftrightarrow \quad C \subseteq \mathsf{Eq}\langle L\rangle.$$

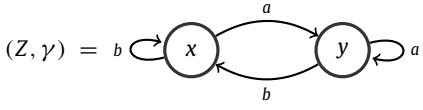
Proof. If $L \in \mathsf{coEq}(A^*/C)$, then $\langle L\rangle$ is completely included in $\mathsf{coEq}(A^*/C)$. By Theorems 10 and 22 there exists an epimorphism from A^*/C to $\mathsf{free}\langle L\rangle$, that is, $C \subseteq \mathsf{Eq}\langle L\rangle$. On the contrary, if $C \subseteq \mathsf{Eq}\langle L\rangle$, there exists an epimorphism from A^*/C to $\mathsf{free}\langle L\rangle$. By Theorem 22, $\mathsf{coEq}(A^*/\mathsf{Eq}\langle L\rangle)$ is completely included in $\mathsf{coEq}(A^*/C)$. Recall that the colouring $\delta_L : A^*/\mathsf{Eq}\langle L\rangle \to 2$, given by $\delta_L([w]) = 1$ iff $w \in L$, is a well-defined function and the equation $o_{\delta_L}([\varepsilon]) = L$ holds, therefore $L \in \mathsf{coEq}(A^*/C)$. □

Corollary 24. *Let L be a language in 2^{A^*}, then $L \in \mathsf{coEq}(A^*/\mathsf{Eq}\langle L\rangle)$.*

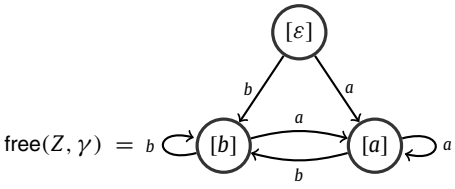
7. Illustrating the duality

We illustrate the duality Theorem 22 with some examples.

Example 25 *(Example 9, continued).* We consider our previous example



for which we had computed

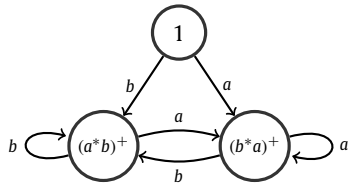


We recall that the transition structure of the automaton $\mathsf{free}(Z,\gamma)$ is inherited from the automaton (A^*,σ) and hence satisfies

$$[w] \xrightarrow{a} [wa]$$

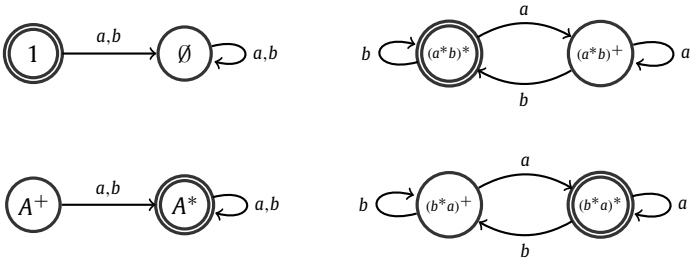
(In particular, transitions between these states are *not* given, as in $(2^{A^*},\tau)$, by right derivatives.) By Lemma 13, each of the languages $[\varepsilon]$, $[a]$ and $[b]$ can be explicitly computed as the behaviour of the initial state $[\varepsilon]$, under the corresponding one-point colouring. This gives:

$\mathsf{free}(Z,\gamma) \;=$



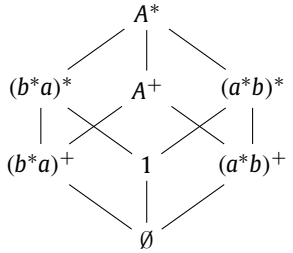
By a computation similar to the one in Example 9, we obtain

$$\mathsf{cofree} \circ \mathsf{free}(Z,\gamma) = \tag{10}$$

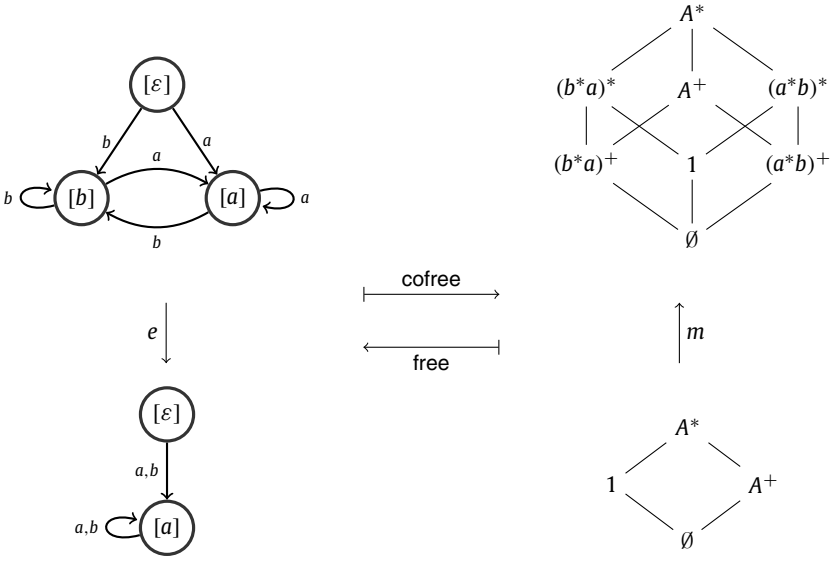


By Proposition 16, the automaton $\mathsf{cofree} \circ \mathsf{free}(X,\alpha)$ is a preformation of languages. In particular, it is a Boolean subalgebra of 2^{A^*}, which we can represent as follows (indicating language inclusion by edges):

$$\mathsf{cofree} \circ \mathsf{free}(Z,\gamma) = \tag{11}$$



(Note that $\mathsf{cofree} \circ \mathsf{free}(Z,\gamma) \cong \mathcal{P}(\mathsf{free}(Z,\gamma))$. Since $\mathsf{free} \circ \mathsf{cofree} \circ \mathsf{free} = \mathsf{free}$, we obtain the following picture, in which we have included an example of an epimorphism e and its image, to illustrate the action of free and cofree on arrows:



(Although it is made superfluous by the duality theorem, it is an interesting little exercise to apply free to the automaton cofree $\circ$ free(Z, γ) 'by hand', that is, by using the definition of free.) $\square$

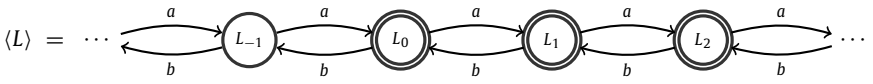
Example 26. Here is an example of an application of the duality Theorem 22 to a language that is not regular. Let $A = \{a, b\}$ and let, for $w \in A^*$,

$$\begin{aligned} |w|_a &= \text{number of } a\text{'s occurring in } w \\ |w|_b &= \text{number of } b\text{'s occurring in } w \end{aligned}$$

We consider the context-free language L defined by

$$L = \{\, w \in A^* \mid |w|_a \geq |w|_b \,\}$$

Its minimal automaton $\langle L \rangle$, which is the smallest subset of 2^{A^*} that contains L and is closed under right derivatives, looks as follows:



where $L_n = \{\, w \in A^* \mid |w|_a + n \geq |w|_b \,\}$, for all $n \in \mathbb{Z}$. If we define a transition function $\alpha : \mathbb{Z} \to \mathbb{Z}^A$ by $n_a = n + 1$ and $n_b = n - 1$, then we obtain an isomorphism $\langle L \rangle \cong (\mathbb{Z}, \alpha)$. It is easy to see that free$\langle L \rangle \cong \langle L \rangle$. If we next define a transition function $\beta : \mathcal{P}(\mathbb{Z}) \to \mathcal{P}(\mathbb{Z})^A$, for all $K \subseteq \mathbb{Z}$, by

$$K_a = K + 1 = \{n + 1 \mid n \in K\} \qquad K_b = K - 1 = \{n - 1 \mid n \in K\}$$

then it follows that cofree$\langle L \rangle \cong (\mathcal{P}(\mathbb{Z}), \beta)$. $\square$

Example 27. In this example, which is taken from [26], we shall illustrate how the duality Theorem 22 can be used for the equational definition of interesting classes of languages. Let $A = \{a, b\}$ and let $\mathbf{ab = ba}$ denote the smallest congruence on A^* containing the equation (ab, ba). It is easy to prove that, for all $v, w \in A^*$,

$$(v, w) \in \mathbf{ab = ba} \;\Leftrightarrow\; |v|_a = |w|_a \text{ and } |v|_b = |w|_b$$

As a consequence, languages $[w]$ in the congruence quotient $A^*/\mathbf{ab = ba}$ satisfy

$$[w] = \{\, v \in A^* \mid v \text{ is a permutation of } w \,\}$$

(with the usual definition of permutation of words). By the duality Theorem 22, we have that $V = \text{cofree}(A^*/\mathbf{ab = ba})$ is a preformation of languages. We now call a language L *commutative* whenever $L \in V$. This terminology is justified by the following equivalences:

$$\begin{aligned} L \in V &\Leftrightarrow L \text{ is the union of permutation equivalence classes } [w] \\ &\Leftrightarrow \langle L \rangle \models ab = ba \end{aligned}$$

The first equivalence follows from the fact that V is a preformation with atoms $[w]$; the second from the fact that free$(V) = A^*/\mathbf{ab = ba}$, whence Eq$(V) = \mathbf{ab = ba}$. $\square$

8. Eilenberg's variety theorem revisited

Eilenberg's variety theorem [2] is a celebrated result in computer science. It underscores the importance of *varieties of finite monoids* or *pseudovarieties* in the study of regular languages. Eilenberg's theorem states that varieties of regular languages are in one-to-one correspondence with *pseudovarieties* of monoids, that is, classes of finite monoids closed under taking submonoids, quotients and finite direct products.

Scattered results in this direction appeared in the mid-sixties. Schützenberger [27], for example, proved that star-free languages are in one-to-one correspondence with aperiodic monoids. The success of Eilenberg's theorem relies on the generality of the result; he understood that finite aperiodic monoids are just an example of a pseudovariety. We can find further instantiations of this result; the rational languages, for example, are associated with the variety of all finite monoids [2,25] and the piecewise testable languages with the variety of finite $\mathcal{J}$-trivial monoids [28].

Several attempts to generalise this result appear in the literature; see for instance [29–31]. These papers aim at extending Eilenberg's result by relaxing some conditions on the class of monoids or on the class of languages. A strong attempt to embrace these results in a common categorical background was made in [15], where the authors introduced varieties of languages in a category $\mathscr{C}$, and proved their correspondence with pseudovarieties of monoids in a closed monoidal category $\mathscr{D}$, provided that $\mathscr{C}$ and $\mathscr{D}$ are dual on finite objects. In any case, all the results involve classes of finite monoids.

In this section, with the Duality Theorem 22 we presented above, we will prove a variation of Eilenberg's original variety theorem: below, we will relate varieties of monoids, instead of pseudovarieties, and varieties of languages, which are now defined in terms of properties of equations and coequations (see Definition 35 below). Although similar, the notion of pseudovariety differs from the notion of *variety* introduced by Birkhoff [7]. A class of monoids is a *variety* of monoids if it is closed under taking substructures, quotients and (not necessarily finite) products. Thus, infinite objects are allowed in a variety. This section provides an interesting example of the expressiveness of the functors free and cofree. Moreover, the results we present for classes of non-necessarily finite monoids subsume Eilenberg's original variety theorem.

8.1. On transition monoids

In algebraic language theory (cf. [1,2,6,32]), regular languages are typically studied in terms of so-called *syntactic* monoids and congruences. We recall that a *monoid* $(M, \cdot, 1)$ consists of a set M, a multiplication operation that is associative, and an element $1 \in M$ with $m \cdot 1 = 1 \cdot m = m$. For every set, there is the monoid $(X^X, \cdot, 1_X)$ defined by

$$X^X = \{\phi \mid \phi : X \to X\} \qquad 1_X(x) = x \qquad \phi \cdot \psi = \psi \circ \phi$$

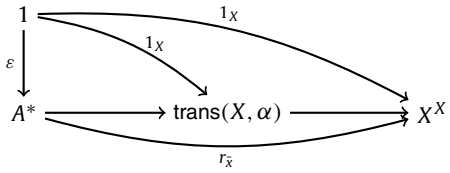
It can be used to define for every automaton (X, α) a pointed automaton

$$(X^X, 1_X, \tilde{\alpha}) \qquad \tilde{\alpha}(\phi)(a)(x) = \phi(x)_a$$

where $\phi(x)_a = \alpha(\phi(x))(a)$, as usual. Now the *transition monoid* [32] for (X, α):

$$(\mathsf{trans}(X, \alpha), 1_X, \tilde{\alpha})$$

is defined by $\mathsf{trans}(X, \alpha) = \mathsf{im}(r_{1_X})$, where r_{1_X} is the reachability map of $(X^X, 1_X, \tilde{\alpha})$:



Theorem 28. *For an automaton* (X, α),

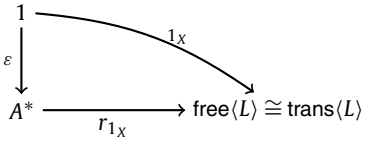
$$(\mathsf{free}(X, \alpha), \bar{x}, \bar{\alpha}) \cong (\mathsf{trans}(X, \alpha), 1_X, \tilde{\alpha})$$

Proof. Let $X = \{x_1, \ldots, x_n\}$. For every $\bar{y} \in \mathsf{free}(X, \alpha)$ we define

$$\phi_{\bar{y}} : X \to X \qquad \phi_{\bar{y}}(x_i) = y_i$$

Then $\phi(\bar{y}) = \phi_{\bar{y}}$ defines an isomorphism of pointed automata. □

We have defined $\mathsf{free}(X, \alpha)$ using the product space ΠX rather than the function space X^X, because it allows us to define the automaton $\mathsf{cofree}(X, \alpha)$ using the coproduct ΣX. As a consequence, $\mathsf{cofree}(X, \alpha)$ can be seen as the dual of $\mathsf{free}(X, \alpha)$ or, equivalently, of the transition monoid. If $(X, \alpha) = \langle L \rangle$, the minimal automaton for a fixed language $L \in 2^{A^*}$:



then the kernel of the reachability map r_{1_X} is the syntactic congruence $\equiv_L$ of L, as we already observed in (6) and (7). Interestingly, the fact that $\mathsf{free}(X, \alpha)$ carries a monoid structure (which it inherits from the concatenation of words in A^*) does not play any role in our proof of the duality between free and cofree.

8.2. Eilenberg's variety theorem

We recall in this subsection the definitions used in the classical Eilenberg's variety theorem, in order to better understand the results and the notions we will present below.

Definition 29. A *variety of finite monoids*, or *pseudovariety*, is a class of finite monoids **H** satisfying:

(i) every homomorphic image of a monoid of **H** belongs to **H**,
(ii) every submonoid of a monoid of **H** belongs to **H**,
(iii) the direct product of a finite family of monoids of **H** also belongs to **H**.

Definition 30. A *variety of regular languages* $\mathcal{H}$ is an assignment to every alphabet A of a family of regular languages satisfying

(i) For each alphabet A, $\mathcal{H}(A)$ is closed under Boolean operations and derivatives.
(ii) If L is a language of $\mathcal{H}(B)$, then for each monoid homomorphism $\varphi\colon A^* \to B^*$ the language $\varphi^{-1}(L)$ belongs to $\mathcal{H}(A)$.

Recall that the syntactic morphism is just the quotient homomorphism $\eta\colon B^* \to \mathsf{free}\langle L\rangle$ (see Corollary 8). At first sight, no other relation with monoids seems to appear in the definition of variety of regular languages. Nevertheless, Eilenberg [2] proved the following striking theorem.

Theorem 31. *(See [2].) There is a one-to-one correspondence between varieties of finite monoids and varieties of regular languages.*

In order to prove that result, Eilenberg associates with each variety of finite monoids **H**, the set $\mathcal{H}(A)$ of all recognisable languages of A^* whose syntactic monoid belongs to **H**. Conversely, to each variety of regular languages $\mathcal{H}$, he associates the variety of finite monoids **H** generated by the syntactic monoids of every regular language L in $\mathcal{H}(A)$, for a certain alphabet A. These constructions define mutually inverse bijective correspondences between varieties of finite monoids and varieties of regular languages.

8.3. A variant of Eilenberg's theorem

We present in this subsection some algebraic definitions in order to prove a variant of Eilenberg's variety theorem [2]. Here, varieties of finite monoids are replaced by varieties of monoids (as stated by Birkhoff [7]) and varieties of regular languages are replaced by varieties of languages. The definition of variety of languages is given in terms of equations and coequations.

Varieties of monoids

Definition 32. A *variety of monoids* is a class of monoids **V** satisfying:

(i) every homomorphic image of a monoid of **V** belongs to **V**,
(ii) every submonoid of a monoid of **V** belongs to **V**,
(iii) the direct product of every family of monoids of **V** also belongs to **V**.

There are two points in which this definition differs from that of pseudovariety (Definition 29). One is that all monoids in **V** are not assumed to be finite. The second one is that **V** is closed under arbitrary direct products. Birkhoff proved two main results; the characterisation of varieties by sets of identities and the closure conditions a class of algebras must satisfy in order to be a variety. To simplify some proofs below, we will work with *subdirect products*.

Following [33, p. 78], we say that a monoid M is a *subdirect product* of the product of a family of monoids $\{M_i \mid i \in I\}$ if M is a submonoid of the direct product $\prod_{i\in I} M_i$ and each induced projection π_i from M onto M_i is surjective. A monoid M which is isomorphic to such a submonoid P is also called a subdirect product of the monoids $\{M_i \mid i \in I\}$. The following theorem of Kogalovskiĭ [34] (see also [35,36]) characterises varieties of monoids in terms of quotients and subdirect products.

Theorem 33. *(See [34].) A class of monoids* **V** *is a variety if and only if it is closed under taking arbitrary subdirect products and quotients.*

Varieties of monoids are equationally defined classes of monoids [35,7]. For a monoid M, its *residual* with respect to a formation of monoids **V**, written $M_{\mathbf{V}}$, is defined as

$$M_{\mathbf{V}} = \bigcap \{C \subseteq M \times M \mid C \text{ is a congruence and } M/C \in \mathbf{V}\}.$$

The above family is not empty as the *total relation* $\nabla_M = M \times M$ is always included.

Proposition 34. *If* **V** *is a variety of monoids, for every monoid M, the quotient $M/M_{\mathbf{V}}$ is a monoid in* **V**.

Proof. Note that $M/M_{\mathbf{V}}$ is the subdirect product of the family of all quotients of M in **V**. Kogalovskiĭ's Theorem 33 guarantees us that this subdirect product is in **V**. □

Varieties of languages

Definition 35. A *variety of languages* $\mathcal{V}$ is an assignment to every alphabet A of a family of formal languages satisfying:

(i) for each alphabet A, if L is a language in $\mathcal{V}(A)$, then $\mathsf{coEq}(A^*/\mathsf{Eq}\langle L\rangle)$ is included in $\mathcal{V}(A)$;
(ii) for each alphabet A, if the family $\{\mathsf{coEq}(A^*/C_i) \mid i \in I\}$ is included in $\mathcal{V}(A)$, then so is $\mathsf{coEq}(A^*/\bigcap_{i\in I} C_i)$;
(iii) for every two alphabets A and B, if L is a language in $\mathcal{V}(B)$ and $\eta\colon B^* \to \mathsf{free}\langle L\rangle$ denotes the quotient homomorphism, then for each monoid homomorphism $\varphi\colon A^* \to B^*$, the set $\mathsf{coEq}(A^*/\ker(\eta\circ\varphi))$ is included in $\mathcal{V}(A)$.

We will see that varieties of languages are in one-to-one correspondence with varieties of monoids. Consequently, we adopted the name "variety of languages" to emphasise this property. In the definition above, if $\mathcal{V}$ assigns to each alphabet a family of regular languages and we replace the arbitrary family of congruences $\{C_i \mid i \in I\}$ in item (ii) we can recover the notion of variety of regular language (Definition 30) originally introduced by Eilenberg. Here, we require closure under arbitrary intersection of congruences to mirror the respective closure under arbitrary products in the definition of variety of monoid.

A variant of Eilenberg's variety theorem

Proposition 36. *Every variety of monoids* $\mathbf{V}$ *induces a variety of languages* $\mathcal{V}$.

Proof. Consider the assignment:

$$\mathcal{V}\colon A \longmapsto \mathsf{coEq}(A^*/A^*_{\mathbf{V}})$$

(i) Let L be a language in $\mathcal{V}(A)$ then by Corollary 23, $A^*_{\mathbf{V}} \subseteq \mathsf{Eq}\langle L\rangle$. It follows that $\mathsf{coEq}(A^*/\mathsf{Eq}\langle L\rangle)$ is included in $\mathcal{V}(A)$.
(ii) Assume that the family $\{\mathsf{coEq}(A^*/C_i \mid i \in I\}$ is included in $\mathcal{V}(A)$, then $A^*_{\mathbf{V}} \subseteq C_i$ for all $i \in I$. It follows that $A^*_{\mathbf{V}} \subseteq \bigcap_{i\in I} C_i$. By Theorem 22, $\mathsf{coEq}(A^*/\bigcap_{i\in I} C_i)$ is also included in $\mathcal{V}(A)$.
(iii) Now, let A and B be two alphabets, let L be a language in $\mathcal{V}(B)$ and let $\eta\colon B^* \to \mathsf{free}\langle L\rangle$ denote the quotient homomorphism. Finally, let $\varphi\colon A^* \to B^*$ be a monoid homomorphism. Since L is a language in $\mathcal{V}(B)$, we conclude that $\mathsf{free}\langle L\rangle$ is a monoid in $\mathbf{V}$. Recall that $A^*/\ker(\eta\circ\varphi)$ is isomorphic to $\mathsf{im}(\eta\circ\varphi)$ which is a submonoid of $\mathsf{free}\langle L\rangle$. Since $\mathbf{V}$ is closed under taking submonoids, we conclude that $A^*/\ker(\eta\circ\varphi)$ is a monoid in $\mathbf{V}$. It follows that the residual $A^*_{\mathbf{V}}$ is included in $\ker(\eta\circ\varphi)$, thus $\mathsf{coEq}(A^*/\ker(\eta\circ\varphi))$ belongs to $\mathcal{V}(A)$. □

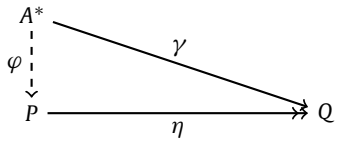
Proposition 37. *Every variety of languages* $\mathcal{V}$ *induces a variety of monoids* $\mathbf{V}$.

Proof. We take $\mathbf{V}$ to be the class of all monoids M that are isomorphic to A^*/C for some alphabet A and some congruence C on A^* satisfying that $\mathsf{coEq}(A^*/C) \subseteq \mathcal{V}(A)$. We will use the characterisation made by Kogalovskiĭ (Theorem 33).

(i) Let $f\colon M \to N$ be a surjective monoid homomorphism defined on a monoid M in $\mathbf{V}$. Then there exists a set A and a congruence C satisfying that $M \cong A^*/C$ and $\mathsf{coEq}(A^*/C) \subseteq \mathcal{V}(A)$. Let γ denote the isomorphism between A^*/C and M. Then $f\circ\gamma\colon A^*/C \to N$ is a surjective monoid homomorphism. Moreover, $C \subseteq \ker(f\circ\gamma)$ which implies that $\mathsf{coEq}(A^*/\ker(f\circ\gamma))$ is included in $\mathcal{V}(A)$. Finally, $A^*/\ker(f\circ\gamma)$ is isomorphic to N, which implies that N belongs to $\mathbf{V}$.
(ii) Now, let M be a monoid that can be expressed as the subdirect product of an arbitrary family $\{M_i \mid i \in I\}$ of monoids in $\mathbf{V}$. Therefore, for each index $i \in I$, there exists an alphabet A_i and a congruence C_i on A_i^* satisfying $M_i \cong A_i^*/C_i$ and $\mathsf{coEq}(A_i^*/C_i) \subseteq \mathcal{V}(A_i)$. Let us denote the corresponding quotient homomorphisms as $\eta_i\colon A_i^* \to A_i^*/C_i$. Consider the alphabet $B = \bigcup_{i\in I} A_i$. By the universal property of the free monoid, we can construct a monoid homomorphism $\varphi_i\colon B^* \to A_i^*$, for all $i \in I$. Thus, $\eta_i\circ\varphi_i\colon B^* \to A_i^*/C_i$ is a surjective monoid homomorphism for all $i \in I$. Denote the congruence $\ker(\eta_i\circ\varphi_i)$ by D_i. As $\mathcal{V}$ is a variety of language, the set $\mathsf{coEq}(B^*/D_i)$ belongs to $\mathcal{V}(B)$. Note that M can be expressed as the subdirect product of the family $\{B^*/D_i \mid i \in I\}$. Since B generates each monoid in the family, M is generated by B. It follows that $M \cong B^*/F$ for some congruence F on B^*. Since M is a subdirect product of the monoids B^*/D_i, we have that $\bigcap_{i\in I} D_i \subseteq F$. Note that $\mathsf{coEq}(B^*/\bigcap_{i\in I} D_i)$ is included in $\mathcal{V}(B)$. By Theorem 22, $\mathsf{coEq}(B^*/F)$ is included in $\mathcal{V}(B)$ and, finally, M is a monoid in $\mathbf{V}$. □

In order to prove our variant of Eilenberg's variety theorem, we shall use the following universal property of the free monoid (see [37, p. 10]).

Proposition 38. *Let* $\gamma\colon A^* \to Q$ *be a monoid homomorphism and* $\eta\colon P \to Q$ *be a surjective monoid homomorphism, then there exists a monoid homomorphism* $\varphi\colon A^* \to P$ *with* $\eta\circ\varphi = \gamma$.



Theorem 39. *The assignments* $\mathbf{V} \mapsto \mathcal{V}$ *and* $\mathcal{V} \mapsto \mathbf{V}$ *define mutually inverse correspondences between varieties of monoids and varieties of languages.*

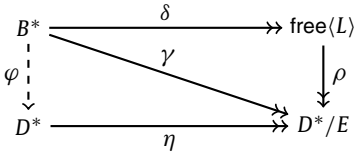
Proof. Consider a variety of monoids $\mathbf{V}$. The first correspondence gives us the variety of languages $\mathcal{V}$ that assigns to each alphabet A the set $\mathsf{coEq}(A^*/A^*_{\mathbf{V}})$. Let $\mathbf{W}$ be the class of all monoids M that are isomorphic to A^*/C for some alphabet A and some congruence C on A^* satisfying that $\mathsf{coEq}(A^*/C) \subseteq \mathcal{V}(A)$.

Let M be a monoid in $\mathbf{V}$ and let B be the set of generators of M. Then $M \cong B^*/F$ for some congruence F on B^*. Recall that $B^*_{\mathbf{V}}$ is included in F, therefore $\mathsf{coEq}(B^*/F)$ is included in $\mathcal{V}(B)$. We conclude that M is a monoid in $\mathbf{W}$. On the contrary, let N be a monoid in $\mathbf{W}$, then N is isomorphic to D^*/E for some alphabet D and some congruence E on D^* satisfying that $\mathsf{coEq}(D^*/E) \subseteq \mathcal{V}(D)$. By Theorem 22, $D^*_{\mathbf{V}} \subseteq E$ and N is a quotient of a monoid in $\mathbf{V}$. We conclude that N is a monoid in $\mathbf{V}$.

Now, let $\mathcal{V}$ be a variety of languages. The first correspondence give us the variety of monoids $\mathbf{V}$ which is defined as the class of all monoids M that are isomorphic to A^*/C for some alphabet A and some congruence C on A^* satisfying that $\mathsf{coEq}(A^*/C) \subseteq \mathcal{V}(A)$. Let $\mathcal{W}$ be the variety of languages that assigns to each alphabet A the set $\mathsf{coEq}(A^*/A^*_{\mathbf{V}})$.

Let L be a language in $\mathcal{V}(B)$, then $\mathsf{coEq}(B^*/\mathsf{Eq}\langle L\rangle)$ is included in $\mathcal{V}(B)$. It follows that $\mathsf{free}\langle L\rangle$ is a monoid in $\mathbf{V}$. Hence, $B^*_{\mathbf{V}} \subseteq \mathsf{Eq}\langle L\rangle$. By Theorem 22, $\mathsf{coEq}(B^*/\mathsf{Eq}\langle L\rangle)$ is included in $\mathcal{W}(B)$. Note that L is always a language in $\mathsf{coEq}(B^*/\mathsf{Eq}\langle L\rangle)$.

Now, let L be a language in $\mathcal{W}(B)$, then $\mathsf{coEq}(B^*/\mathsf{Eq}\langle L\rangle)$ is included in $\mathcal{W}(B)$. By Theorem 22, $B^*_{\mathbf{V}} \subseteq \mathsf{Eq}\langle L\rangle$, therefore $\mathsf{free}\langle L\rangle$ is a monoid in $\mathbf{V}$. By definition of $\mathbf{V}$, the monoid $\mathsf{free}\langle L\rangle$ is isomorphic to D^*/E for some alphabet D and some congruence E on D^* satisfying that $\mathsf{coEq}(D^*/E) \subseteq \mathcal{V}(D)$. Let $\eta\colon D^* \to D^*/E$ and $\delta\colon B^* \to \mathsf{free}\langle L\rangle$ be the corresponding quotient homomorphisms. Let $\rho\colon \mathsf{free}\langle L\rangle \to D^*/E$ be the corresponding monoid isomorphism. It follows that $\gamma = \rho \circ \delta$ is a monoid epimorphism from B^* onto D^*/E. By Proposition 38, there exists a monoid homomorphism $\varphi\colon B^* \to D^*$ with $\eta \circ \varphi = \gamma$. Summarising,



As $\mathcal{V}$ is a variety of languages, the set $\mathsf{coEq}(B^*/\ker(\eta \circ \varphi))$ belongs to $\mathcal{V}(B)$. Note that L is a language in $\mathsf{coEq}(B^*/\ker(\eta \circ \varphi))$. □

The steps on the original proof of Eilenberg's variety theorem can be traced in the proof we just presented. However, in our case, the existence of a relatively free monoid over a given alphabet simplifies most of the proofs. For a variety $\mathbf{V}$ of monoids, since the residual $A^*_{\mathbf{V}}$ is the smallest congruence on any free monoid A^* whose quotient is a monoid in $\mathbf{V}$, the preformation of languages $\mathsf{coEq}(A^*/A^*_{\mathbf{V}})$ is the biggest preformation of languages over A whose syntactic monoid is in $\mathbf{V}$.

Example 40. A monoid M is *commutative* if for all $m, n \in M$, the equation $mn = nm$ holds. The class of all commutative monoids, denoted by $\mathbf{C}$, is a variety of monoids that can be characterised using the identity $xy = yx$. For an alphabet A, the residual of A^* with respect to the variety of commutative monoids is given by

$$A^*_{\mathbf{C}} = \bigvee \{\mathbf{ab} = \mathbf{ba} \mid a, b \in A\}$$

where the right-hand side of the above equation denotes the smallest congruence on A^* containing all congruences $\mathbf{ab} = \mathbf{ba}$, for a, b in A. Every class $[w]$ in $A^*/A^*_{\mathbf{C}}$ is completely determined by the amount of different letters appearing in w. Thus, it is easy to show that, for a finite alphabet A, the monoid $A^*/A^*_{\mathbf{C}}$ is isomorphic to the monoid $\mathbb{N}^A$, of all functions from A to $\mathbb{N}$. In the case of infinite alphabet, we should consider only functions with finite support. Following Example 27, we see that a language L over A is commutative if $L \in \mathsf{cofree}(A^*/A^*_{\mathbf{C}})$.

9. Equational bisimulations

This section introduces the notion of equational bisimulation and we show how it can be used to prove that a language satisfies a given set of equations. First of all, recall the following property on coloured bisimulations (see Section 2 and [3]), which follows from the fact that 2^{A^*} is a final $(2 \times F)$-coalgebra.

Proposition 41. *(See [3].) Let $R \subseteq 2^{A^*} \times 2^{A^*}$ be a coloured bisimulation on $(2^{A^*}, \varepsilon?, \tau)$. If $(K, L) \in R$ then $K = L$.*

It follows that no non-trivial coloured bisimulation can be defined on 2^{A^*}. The above property is often called the coinduction proof method: in order to show that $K = L$, it suffices to define a coloured bisimulation R with $(K, L) \in R$. We refer to [3] for examples that illustrate the usefulness of this proof method. In [38], it is shown how variations on the above proof method lead to surprisingly efficient algorithms for proving the equivalence of non-deterministic finite automata.

Here we generalise the notion of bisimulation for languages as follows. Let $C \subseteq A^* \times A^*$ be a congruence. We call a relation $R \subseteq 2^{A^*} \times 2^{A^*}$ an *equational bisimulation* with respect to C, or *C-bisimulation* for short, if, for all $(K, L) \in R$,

(i) $\varepsilon \in K \Leftrightarrow \varepsilon \in L$
(ii) $\forall (v, w) \in C,\ (K_v, L_w) \in R$

We have the following corresponding proof principle.

Proposition 42. *Let $C \subseteq A^* \times A^*$ be a congruence and let $R \subseteq 2^{A^*} \times 2^{A^*}$ be a C-bisimulation. For all $(K, L) \in R$,*

(i) $K = L$
(ii) $\langle K \rangle \models C$

Proof. Since $(a, a) \in C$, for all $a \in A$, any C-bisimulation is trivially also an ordinary bisimulation. Thus (1) follows from Proposition 41. For (2), let $(K, L) \in R$ and consider any state $K_u \in \langle K \rangle$ and any pair $(v, w) \in C$. Since $(K, K) = (K, L) \in R$ and R is a C-bisimulation, and since $(uv, uw) \in C$, it follows that $(K_{uv}, K_{uw}) \in R$. By (1), we have $K_{uv} = K_{uw}$ and thus $(K_u)_v = (K_u)_w$, which proves (2). □

Example 43. Let $K = aA^* + b(a^*b)^* + b(b^*a)^+$. We shall use Proposition 42 to show that K is commutative. Referring to Example 27, we need to prove that $\langle K \rangle \models ab = ba$. Let

$$M = A^*, \qquad N = (a^*b)^* + (b^*a)^+, \qquad O = (a^*b)^+ + (b^*a)^*$$

and let

$$R = \{\langle K, K \rangle\} \cup \{M, N, O\}^2$$

Then R is an (**ab = ba**)-bisimulation. Thus $\langle K \rangle \models ab = ba$, by Proposition 42. □

Example 44. For a next example, we return to the context-free language of Example 26:

$$L = \{\, w \in A^* \mid |w|_a \geq |w|_b \,\}$$

and show that also L is commutative. Let $L_n = \{\, w \in A^* \mid |w|_a + n \geq |w|_b \,\}$ and let

$$S = \{\langle L_n, L_n \rangle \mid n \in \mathbb{Z}\}$$

Then S is an (**ab = ba**)-bisimulation and thus $\langle L \rangle \models ab = ba$, by Proposition 42. □

10. Discussion

The work by Gehrke

We begin by relating our duality result to the work of Gehrke [17]. There all automata $\mathcal{A} = (Q, A, \delta, I, F)$ are finite. Consequently, the *language recognised by* $\mathcal{A}$, denoted $L(\mathcal{A})$, is regular. For a finite alphabet A, the concatenation operation on A^* gives rise to a *residuated family* of operations on the set of all languages of A^* as follows. *Complex concatenation* on $\mathcal{P}(A^*)$ is given by

$$KL = \{uv \mid u \in K \text{ and } v \in L\}$$

The *residuals* of this operation are uniquely determined by the *residuation laws*:

$$\forall K, L, M \in \mathcal{P}(A^*) \qquad KM \subseteq L \quad \Leftrightarrow \quad M \subseteq K \backslash L \quad \Leftrightarrow \quad K \subseteq L / M$$

In particular, for any word $w \in A^*$, the following operations coincide

$$\{w\} \backslash L = L_w \qquad \text{and} \qquad L / \{w\} = {}_wL$$

Since L is regular, the set $\{{}_yL_x \mid y, x \in A^*\}$ is finite (see [17, Proposition 1]).

Definition 45. (See [17, Definition 4].) Let A be a finite alphabet and $L \subseteq A^*$ a language over A. Let $\mathcal{B}(L)$ be the Boolean subalgebra of $\mathcal{P}(A^*)$ generated by the set $\{{}_yL_x \mid x, y \in A^*\}$. We will call $\mathcal{B}(L)$ the *quotienting ideal* generated by L. More generally a quotienting ideal of $\mathcal{P}(A^*)$ is a Boolean subalgebra which is closed under the quotienting operations $(\ )_x$ and ${}_y(\ )$ for all $x, y \in A^*$.

The following theorem is one of the most important results of [17].

Theorem 46. *(See [17, Theorem 1].) Let L be a language recognised by an automaton. The extended dual of the Boolean algebra with additional operations $(\mathcal{B}(L), \backslash, /)$ is the syntactic monoid of L. In particular, it follows that the syntactic monoid of L is finite and is effectively computable.*

The next proposition states that the dual object to the syntactic monoid of a regular language L coincides with the preformation of languages $\mathsf{cofree} \circ \mathsf{free}\langle L\rangle$ we described in the present paper. To prove it, we will use a previous lemma stating that every class $[w]$ in $A^*/\mathsf{Eq}\langle L\rangle$ belongs to the Boolean algebra generated by the set $\{{}_yL_x \mid x, y \in A^*\}$.

Proposition 47. *For a regular language L over an alphabet A^*, the Boolean algebra with additional operations $(\mathcal{B}(L), \backslash, /)$ and $\mathsf{cofree} \circ \mathsf{free}\langle L\rangle$ coincide.*

Proof. We will use the following abbreviation $V = \mathsf{cofree} \circ \mathsf{free}\langle L\rangle$. Let $(v, w) \in \mathsf{Eq}\langle L\rangle$, then for all $x \in A^*$, we have that $L_{xv} = L_{xw}$. Therefore, for $y \in A^*$ we deduce the equations ${}_yL_{xv} = {}_yL_{xw}$. It follows that $\mathsf{Eq}\langle L\rangle \subseteq \mathsf{Eq}\langle {}_yL_x\rangle$. By Corollary 23 ${}_yL_x$ is included in V. Since V is a variety of languages, we conclude that $\mathcal{B}(L)$ is included in V. Recall that for any pair of languages K, M in 2^{A^*}, the equations $K\backslash M = \bigcap_{w\in K} M_w$ and $M/K = \bigcap_{w\in K} {}_wM$ hold. Hence, V is closed under residuals. Now, let $[u]$ be an element in $A^*/\mathsf{Eq}\langle L\rangle$. Since L is regular, every atom in V can be defined according to [39, Proposition 2.14] using finitely many Boolean operations. Thus, it belongs to the Boolean algebra generated by the derivatives of L. It follows that V is included in $\mathcal{B}(L)$. □

Recall that, for regular languages, the set $\mathcal{B}(L)$ is a finite lattice and it is, therefore, complete and atomic. We can say that, for finite automata, our duality coincides with that obtained by Stone duality in Theorem 46. It is interesting to note that our result emerges from a structural study of automata and, so far, no direct appeal to Stone duality is required.

Profinite techniques

Stone duality is used to obtain further stronger results for regular languages and finite monoids. The connection between profinite words and Stone spaces was already discovered by Almeida [18,19]. However, it was Pippenger in [20] the first to formulate it in terms of Stone duality. They both observed that the Boolean algebra of regular languages over A^* is dual to the Stone space $\widehat{A^*}$ of profinite words. This duality extends to a one-to-one correspondence between Boolean algebras of regular languages and quotients of $\widehat{A^*}$.

On the other hand, Reiterman [40] showed that pseudovarieties of monoids are characterised by "implicit" identities that they satisfy. "Implicit" identities are limits of sequences of the ordinary terms that appear in ordinary identities. Eilenberg stated that pseudovarieties could be characterised by infinite sequences of identities, with each monoid satisfying all but finitely many identities in each sequence. (Thus, aperiodic finite semigroups, for example, satisfy the identity $x^k = x^{k+1}$ for sufficiently large k). The modern rendering of such intuitions appear in the work done by Gehrke [21,17] and Pin [22], who take a further step in this direction showing that *lattices of languages* are precisely those classes of regular languages being defined by profinite identities [22]. It follows from these works the strong connection between classes of regular languages, finite monoids and sets of profinite identities. This approach is also very useful to establish effective decision procedures.

This correspondence strongly depends on profinite techniques. Recall that the profinite monoid $\widehat{A^*}$ can be constructed both as the completion of an ultrametric defined on A^* or as the projective limit of all finite monoids whose generators are in A (see [22] and [41], respectively). Indeed, our results in the monoid side refer to objects $(A^*/C$, for some congruence C on $A^*)$ and the results on [20,22,21] and [17] refer to limit constructions (profinite monoid). Indeed, the monoid $\widehat{A^*}$ cannot be written as A^*/C for some congruence C on A^* and, therefore, our results per se do not apply.

However, the functorial approach we present here could be used to retrieve a similar situation. Projective limits (the profinite monoid $\widehat{A^*}$) and inductive limits (the set of all regular languages $Reg(A^*)$) are categorical limits in which all arrows involved are epi or mono, respectively. So far, we know that the category $\mathcal{A}_m$ has inductive limits, whereas $\mathcal{A}_e$ require an additional argument to guarantee that the mediating map from the limit to the monoids is epi. At this point, it seems necessary to appeal to topological arguments (see for instance [42, Lemma 3.1.27]). If such limits in both $\mathcal{A}_m$ and $\mathcal{A}_e$ exist, our equivalence will preserve both limits and colimits and we will retrieve a similar result on limit constructions. All in all, this line of future work deserves further study.

The present paper already contains some contributions that encourage us to continue working along these lines. The first relevant insight is that we are able to deal with *infinite* automata and non-regular languages. It lies in the fact that the duality we find is the (conceptually simpler) discrete duality between sets and complete atomic Boolean algebras. The latter duality is also used in [43], where it was lifted to a dual equivalence between deterministic automata and so-called

Boolean automata. We hope to retrieve some of the results presented in the papers [18–20,22,21,17], specially Reiterman's characterisation in terms of profinite equations. Further limit constructions of non-necessarily finite monoids need to be investigated.

A second useful approach we presented here is the categorical description of the duality presented in Theorem 22 and its more manageable Corollary 23. Of special interest on its own is the variant on Eilenberg's theorem we presented here as an almost immediate consequence. The expressiveness of the functors free and cofree has been decisive in this proof. Recall that the connection between varieties of languages (Definition 35) and congruences is explicitly presented from the very beginning with the help of equations coequations. We want to understand further variants of Eilenberg's result. We are specially interested in the result achieved in [39], where varieties of finite monoids were replaced by the less restrictive concept of formations of finite monoids. We hope that the version for varieties presented here could help us to achieve a result on formations of non-necessarily finite monoids.

Because we are working within the algebra–coalgebra duality, we can use both algebraic notions, such as congruence, and coalgebraic notions, such as bisimulations. Our notion of equational bisimulation, which is a generalisation of the standard notion, seems to be new and so does the corresponding coinduction proof principle. Within this context of the algebra–coalgebra duality, we also want to study the notions of varieties and covarieties of automata. In [44], some initial results are mentioned but with the present duality in place, we expect that more can be said. The notion of equational bisimulation and its corresponding coinduction proof principle deserve further study, both the present instance for automata and its coalgebraic generalisations.

Finally, we want to investigate to what extent our duality can be further generalised to other dynamical systems, such as Moore automata and probabilistic automata. The algebra–coalgebra duality as such has already been extended to such automata in [12,13], leading to generalisations of Brzozowski's algorithm. In addition, we plan to study the connections with [45] and [46], where dualities for generalised rational structures have been studied.

Acknowledgments

We are much obliged to the two anonymous referees: their comments and suggestions have greatly improved our paper. The first author has been supported by the grant MTM2014-54707-C3-1-P from the *Ministerio de Economía y Competitividad* (Spanish Government) and the grant 11271085 from the *National Natural Science Foundation of China*. The second author has been supported by the predoctoral grant AP2010-2764 from the *Ministerio de Educación* (Spanish Government) and the grant MTM2014-54707-C3-1-P from the *Ministerio de Economía y Competitividad* (Spanish Government) and by an internship from *CWI*.

References

[1] S. Eilenberg, Automata, Languages and Machines, vol. A, Pure and Applied Mathematics, Academic Press, 1974.
[2] S. Eilenberg, Automata, Languages and Machines, vol. B, Pure and Applied Mathematics, Academic Press, 1976.
[3] J. Rutten, Automata and coinduction (an exercise in coalgebra), in: D. Sangiorgi, R. de Simone (Eds.), Proceedings of CONCUR'98, in: LNCS, vol. 1466, 1998, pp. 194–218.
[4] J. Rutten, Universal coalgebra: a theory of systems, Theor. Comput. Sci. 249 (1) (2000) 3–80, fundamental study.
[5] J. Sakarovitch, Elements of Automata Theory, Cambridge University Press, 2009.
[6] J.-E. Pin, Mathematical Foundations of Automata Theory, http://www.liafa.jussieu.fr/~jep/PDF/MPRI/MPRI.pdf, 2014.
[7] G. Birkhoff, On the structure of abstract algebras, Math. Proc. Camb. Philos. Soc. 31 (1935) 433–454.
[8] M. Arbib, H. Zeiger, On the relevance of abstract algebra to control theory, Automatica 5 (1969) 589–606.
[9] M. Arbib, E. Manes, Adjoint machines, state-behaviour machines, and duality, J. Pure Appl. Algebra 6 (1975) 313–344.
[10] R. Kalman, On the general theory of control systems, IRE Trans. Autom. Control 4 (3) (1959) 110–111.
[11] R.E. Kalman, P.L. Falb, M.A. Arbib, Topics in Mathematical Systems Theory, McGraw Hill, 1969.
[12] F. Bonchi, M. Bonsangue, J. Rutten, A. Silva, Brzozowski's algorithm (co)algebraically, in: R. Constable, A. Silva (Eds.), Logic and Program Semantics, in: LNCS, vol. 7230, 2012, pp. 12–23.
[13] F. Bonchi, M. Bonsangue, H. Hansen, P. Panangaden, J. Rutten, A. Silva, Algebra–coalgebra duality in Brzozowski's minimization algorithm, ACM Transactions on Computational Logic 15 (1), Article 3.
[14] J. Brzozowski, Derivatives of regular expressions, J. ACM 11 (4) (1964) 481–494.
[15] J. Adamek, S. Milius, R. Myers, H. Urbat, Varieties of languages in a category, ArXiv e-prints.
[16] M.O. Rabin, D. Scott, Finite automata and their decision problems, IBM J. Res. Dev. 3 (2) (1959) 114–125.
[17] M. Gehrke, Duality and recognition, in: P. Sankowski, F. Murlak (Eds.), Mathematical Foundations of Computer Science, in: LNCS, vol. 6907, 2011, pp. 3–18.
[18] J. Almeida, Residually finite congruences and quasi-regular subsets in uniform algebras, Port. Math. 46 (3) (1989) 313–328.
[19] J. Almeida, Finite Semigroups and Universal Algebra, Series in Algebra, World Scientific, 1994.
[20] N. Pippenger, Regular languages and stone duality, Theory Comput. Syst. 30 (2) (1997) 121–134.
[21] M. Gehrke, Stone duality and the recognisable languages over an algebra, in: A. Kurz, et al. (Eds.), Algebra and Coalgebra in Computer Science, CALCO 2009, in: LNCS, vol. 5728, 2009, pp. 236–250.
[22] M. Gehrke, S. Grigorieff, J.-E. Pin, Duality and equational theory of regular languages, in: Proceedings ICALP, in: LNCS, vol. 5126, 2008, pp. 246–257.
[23] E. Manes, M. Arbib, Algebraic Approaches to Program Semantics, The AKM Series in Theoretical Computer Science, Springer, New York, 1986.
[24] J. Conway, Regular Algebra and Finite Machines, Dover Books on Mathematics, Dover Publications, 2012.
[25] S. Kleene, Representation of events in nerve nets and finite automata, in: J. McCarthy, C.E. Shannon, W.R. Ashby (Eds.), Automata Studies, Princeton Univ. Press, 1956, pp. 3–41.
[26] M. Dekkers, Stone duality. An application in the theory of formal languages, master's thesis, Radboud Universiteit Nijmegen, The Netherlands, December 2008.

[27] M. Schützenberger, On finite monoids having only trivial subgroups, Inf. Control 8 (2) (1965) 190–194.
[28] I. Simon, Piecewise testable events, in: Proceedings of the 2nd GI Conference on Automata Theory and Formal Languages, Springer-Verlag, London, UK, 1975, pp. 214–222.
[29] J.-E. Pin, A variety theorem without complementation, Russ. Math. (Izv. VUZ) 39 (1995) 80–90.
[30] H. Straubing, On logical descriptions of regular languages, in: S. Rajsbaum (Ed.), LATIN 2002: Theoretical Informatics, in: Lecture Notes in Computer Science, vol. 2286, Springer, Berlin/Heidelberg, 2002, pp. 528–538.
[31] Z. Ésik, M. Ito, Temporal logic with cyclic counting and the degree of aperiodicity of finite automata, Acta Cybern. 16 (1) (2003) 1–28.
[32] C.M. Reis, H.J. Shyr, Some properties of disjunctive languages on a free monoid, Inf. Control 37 (3) (1978) 334–344.
[33] P. Grillet, Semigroups: an Introduction to the Structure Theory, Marcel Dekker, Inc., New York, 1995.
[34] S.P. Kogalovskiĭ, On Birkhoff's theorem, Usp. Mat. Nauk 20 (5) (1965) 206–207.
[35] H. Neumann, Varieties of Groups, Ergeb. Math. Ihrer Grenzgeb., Springer-Verlag, 1967.
[36] G. Grätzer, Universal Algebra, Mathematics and Statistics, Springer, 2008.
[37] J.-É. Pin, Varieties of Formal Languages, North Oxford/Plenum, London/New York, 1986, translation: Variétés de langages formels, Masson, 1984.
[38] F. Bonchi, D. Pous, Checking NFA equivalence with bisimulations up to congruence, in: Proc. POPL, 2013, pp. 457–468, http://doi.acm.org/10.1145/2429069.2429124.
[39] A. Ballester-Bolinches, J.-É. Pin, X. Soler-Escrivà, Formations of finite monoids and formal languages: Eilenberg's theorem revisited, Forum Math. 26 (6) (2012) 1731–1761, http://dx.doi.org/10.1515/forum-2012-0055.
[40] J. Reiterman, The Birkhoff theorem for finite algebras, Algebra Univers. 14 (1) (1982) 1–10.
[41] J. Almeida, Profinite semigroups and applications, in: Structural Theory of Automata, Semigroups, and Universal Algebra, 2003, pp. 7–18.
[42] J. Rhodes, B. Steinberg, The q-Theory of Finite Semigroups, Springer Monographs in Mathematics, Springer, New York, 2009.
[43] F. Roumen, Canonical automata via duality, Master's thesis, Radboud Universiteit Nijmegen, the Netherlands, 2011.
[44] J. Rutten, A. Ballester-Bolinches, E. Cosme-Llópez, Varieties and covarieties of languages (preliminary version), in: D. Kozen, M. Mislove (Eds.), Proceedings of MFPS XXIX, in: Electron. Notes Theor. Comput. Sci., vol. 298, 2013, pp. 7–28.
[45] N. Bezhanishvili, C. Kupke, P. Panangaden, Minimization via duality, in: C.-H.L. Ong, R.J.G.B. de Queiroz (Eds.), WoLLIC, in: LNCS, vol. 7456, 2012, pp. 191–205.
[46] J. Adámek, S. Milius, R. Myers, H. Urbat, Generalized Eilenberg theorem, I: local varieties of languages, in: A. Muscholl (Ed.), Foundations of Software Science and Computation Structures, in: LNCS, vol. 8412, 2014, pp. 366–380.