Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 298 (2013) 7–28

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Varieties and Covarieties of Languages (Extended Abstract)

Jan Rutten[1]

CWI and Radboud University
Amsterdam, The Netherlands

Adolfo Ballester-Bolinches[2] Enric Cosme-Llópez[3]

Departament d'Àlgebra
Universitat de València
València, Spain

Abstract

Because of the isomorphism $(X \times A) \to X \cong X \to (A \to X)$, the transition structure 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. This algebra-coalgebra duality goes back to Arbib and Manes, who formulated it as a duality between reachability and observability, and is ultimately based on Kalman's duality in systems theory between controllability and observability. Recently, it was used to give a new proof of Brzozowski's minimization algorithm for deterministic automata. Here we will use the algebra-coalgebra duality of automata as a common perspective for the study of both varieties and covarieties, which are classes of automata and languages defined by equations and coequations, respectively. We make a first connection with Eilenberg's definition of varieties of languages, which is based on the classical, algebraic notion of varieties of (transition) monoids.

Keywords: Automata, variety, covariety, equation, coequation, algebra, coalgebra.

1 Introduction

Because of the isomorphism

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

the transition structure of a deterministic automaton with state set X and with inputs from an alphabet A can be viewed both as an algebra [11] and as a coalgebra

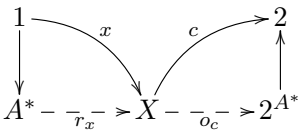
[1] Email: janr@cwi.nl
[2] Email: Adolfo.Ballester@uv.es
[3] Email: enric.cosme@uv.es

http://dx.doi.org/10.1016/j.entcs.2013.09.005

[19,20]. As a consequence, both the algebraic notion of *variety* and the coalgebraic notion of *covariety* apply. In this paper, we present a preliminary version of what is to become a systematic study of varieties and covarieties of automata and of formal languages.

We will define a variety of automata (viewed as algebras) in the usual way, as a class defined by *equations* [12]. A covariety of automata (viewed as coalgebras) will be a class defined by *coequations* [20]. Varieties and covarieties of automata will then be used to define varieties and covarieties of *languages*. Our notion of a variety of languages is different from the classical definition by Eilenberg [12,18], and we will make some initial observations on how the two notions are related.

The setting of our investigations will be the following picture:



(1)

(This diagram will be explained in more detail in Section 3.) In the middle, we have the state set X of a given automaton. On the left, A^* is the set of all words over A, and on the right, 2^{A^*} is the set of all languages over A. For every choice of a *point* (initial state) $x \in X$, the function r_x sends any word w to the state x_w reached from x on input w. And for every choice of a *colouring* (set of final states) $c : X \to 2$, the function o_c sends any state to the language it accepts.

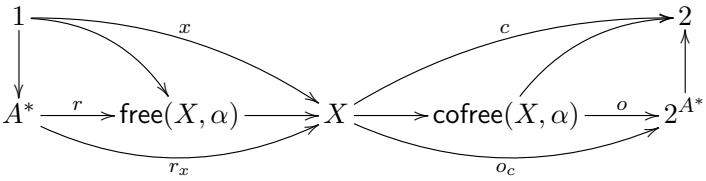
Both the pointed automata A^* (with the empty word as point) and X with point x, are algebras. And both the coloured automata 2^{A^*} (with colouring as explained later) and X with colouring c, are coalgebras. The unique existence of the function (in fact, a homomorphism of algebras) r_x is induced by the *initiality* of A^*. And the unique existence of the function (a homomorphism of coalgebras) o_c is induced by the *finality* of 2^{A^*}.

(Sets of) equations will live in the left – algebraic – part of our diagram; in short, they correspond to *quotients* of A^*. And (sets of) coequations live in the right – coalgebraic – part of our diagram; they will correspond to *subautomata* of 2^{A^*}. As a consequence, diagram (1) allows us to define both varieties and covarieties, and to study their properties from a common perspective.

The *algebra-coalgebra duality* of diagram (1) is a modern rendering of the duality between *reachability* and *observability* of automata [2,1], which ultimately goes back to Kalman's duality between controllability and observability in system theory [14,15]. (See also [7,9] for further categorical generalisations.)

Recently [6,3], this algebra-coalgebra duality of automata was used to give a new proof and various generalisations of Brzozowski's minimization algorithm [8]. The present work goes in a different direction, focusing on (co)equations and

(co)varieties. Notably, we will further refine diagram (1) as follows:



(For details, see Section 5.) The new diagram includes, for every automaton X with transition function $\alpha : X \to X^A$, the (pointed) automaton $\mathsf{free}(X, \alpha)$, which represents the *largest set of equations* satisfied by (X, α). And, dually, we will construct a (coloured) automaton $\mathsf{cofree}(X, \alpha)$, which represents the *smallest set of coequations* satisfied by (X, α).

We already mentioned above that our definition of a variety of languages is different from Eilenberg's, which is derived from the (classical, algebraic) notion of variety of *monoids*. A first step towards an understanding of the relation between the classical and the present notion of variety consists of the – elementary but to us somewhat surprising – observation that $\mathsf{free}(X, \alpha)$ is isomorphic to the so-called *transition monoid* of X (which is called the *syntactic* monoid in case X is minimal) [18]. This observation furthermore implies that the coloured automaton $\mathsf{cofree}(X, \alpha)$ can be viewed as a dual version of the transition monoid.

Much remains to be further understood. We already mentioned the connection with Eilenberg's variety theorem. Furthermore, we would also like to relate the present algebra-coalgebra perspective to recent developments on varieties of languages, notably [13] and [4,5]. Finally, it should be possible to generalise the present setting, along the lines of [6,3], from deterministic automata to other structures such as Mealy machines, weighted automata etc.

2 Preliminaries

Let A be a finite 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.

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

$$\mathsf{im}(f) = \{y \in Y \mid \exists x \in X,\ f(x) = y\}$$

$$\mathsf{ker}(f) = \{(x_1, x_2) \in X \times X \mid f(x_1) = f(x_2)\}$$

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,

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

We define the *initial value* $L(0)$ of L by

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

For a functor $F : Set \to Set$, an *F-algebra* is a pair (S, α) consisting of a set S and a function $\alpha : F(S) \to S$. An *F-coalgebra* is a pair (S, α) with $\alpha : S \to F(S)$.

We will be using the following functors:

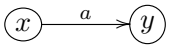
$$\begin{aligned} F(S) &= S^A \\ G(S) &= S \times A \\ (2 \times F)(S) &= 2 \times S^A \\ (1 + G)(S) &= 1 + (S \times A) \end{aligned}$$

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

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

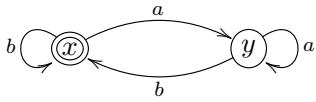
automata are also *G-algebras* [17].

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:

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

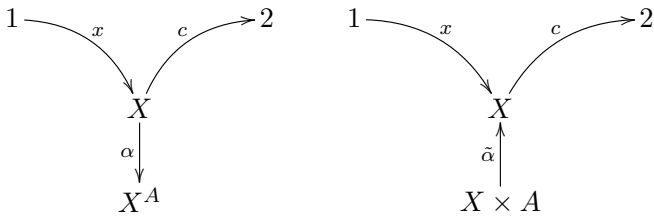
An automaton can also have an *initial state* $x \in X$, here represented by a function

$$x : 1 \to X$$

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 two following diagrams

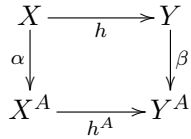


We will be using the diagram on the left, which is just a matter of personal preference.

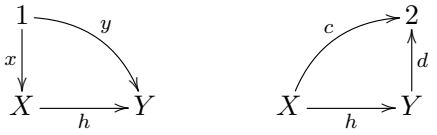
We observe further that pointed and coloured automata are simply called *automata* in most of the literature on automata theory. A pointed and coloured automaton (X, x, c, α) is neither an algebra nor a coalgebra – because of c and x, respectively – which can be a cause of fascination and confusion alike.

Homomorphisms, subautomata, bisimulations

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



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.

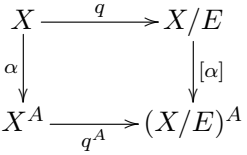
We call a relation $R \subseteq X \times Y$ a *bisimulation of automata* if for all $(x, y) \in X \times Y$,

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

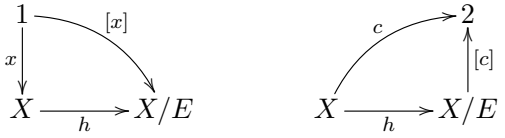
(where $x_a = \sigma(x)(a)$ and $y_a = \tau(y)(a)$). For pointed automata (X, x, α) and (Y, y, β), R is a *pointed bisimulation* if, moreover, $(x, y) \in R$. And for coloured automata (X, x, α) and (Y, y, β), R is a *coloured bisimulation* if, moreover, for all $(x, y) \in X \times Y$,

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

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:

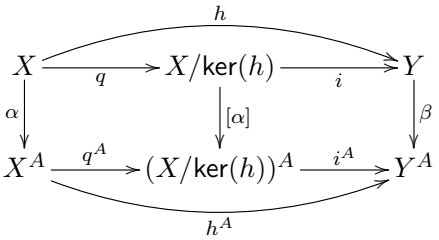


with the obvious definitions of X/E, q and $[\alpha]$. If the equivalence E is a pointed bisimulation on (X, x, α) or a coloured bisimulation on (X, c, α), then we moreover have, respectively,



with, again, the obvious definitions of $[x]$ and $[c]$.

For a homomorphism $h : X \to Y$, $\mathsf{ker}(h)$ is a bisimulation equivalence on X and $\mathsf{im}(h)$ is a subautomaton of Y. Any homomorphism h factors through quotient and inclusion homomorphisms as follows:



Note that $X/\mathsf{ker}(h) \cong \mathsf{im}(h)$. Because q is surjective and i is injective, the pair (q, i) 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 $(v, w) \in A^* \times A^*$,

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

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

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

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

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

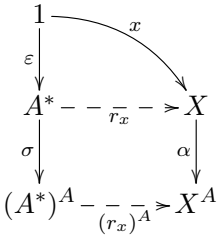
All of the above binary (co)products can be easily generalised to (co)products of arbitrary families of automata.

3 Setting the scene

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

$$\sigma : A^* \to (A^*)^A \qquad \sigma(w)(a) = wa$$

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

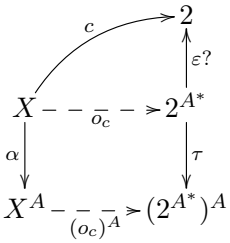
$$\varepsilon? \; : \; 2^{A^*} \to 2 \qquad \varepsilon?(L) = L(0)$$

and transition function τ defined by

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

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

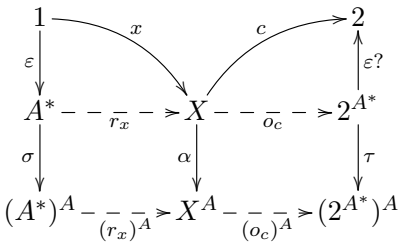
$o_c(x) = \{w \mid c(x_w) = 1\}$, that makes the following diagram commute:



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.

The scene

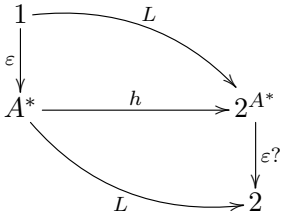
Summarizing, we have set the following scene for our investigations:



(2)

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*.

For a given language $L \in 2^{A^*}$, there is the following variation of the picture above:

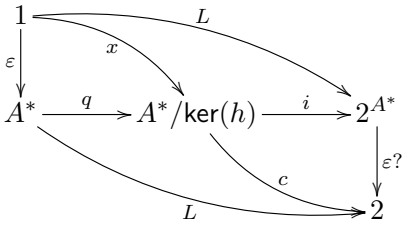


where the lower L is in fact the characteristic function of $L \subseteq A^*$, and where the homomorphism h satisfies $h = r_L = o_L$ and $h(w) = L_w$. As a consequence, we have $h(v) = h(w)$ iff

$$\forall u \in A^*,\ vu \in L \Leftrightarrow wu \in L$$

which we recognise as the celebrated *Myhill-Nerode* equivalence. A *minimal au-*

tomaton 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 $(2^{A^*}, \varepsilon?)$ generated by L.

In conclusion of this section, we observe that $\langle L \rangle$ is finite iff the language L is *rational*. This fact is a version [8,10] of Kleene's correspondence between finite automata and rational languages [16].

4 Equations and coequations

We will be referring to the situation of (2).

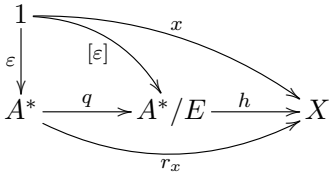
Definition 4.1 [equations] A *set of equations* is a bisimulation equivalence relation $E \subseteq A^* \times A^*$ on the automaton (A^*, σ). We define $(X, x, \alpha) \models E$ – and say: *the pointed automaton* (X, x, α) *satisfies* E – by

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

Because

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

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

$$\begin{aligned}(X, \alpha) \models E &\Leftrightarrow \forall x : 1 \to X,\ \ (X, x, \alpha) \models E \\ &\Leftrightarrow \forall x \in X,\ \forall (v, w) \in E,\ x_v = x_w\end{aligned}$$

□

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 the following shorthand:

$$(X, x, \alpha) \models v = w \;\Leftrightarrow\; (X, x, \alpha) \models E_{v=w}$$

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

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

Definition 4.2 [coequations]

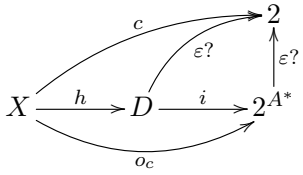
A *set of coequations* is a subautomaton $D \subseteq 2^{A^*}$ of the automaton $(2^{A^*}, \tau)$. We define $(X, c, \alpha) \models D$ – and say: *the coloured automaton* (X, c, α) *satisfies* D – by

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

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

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

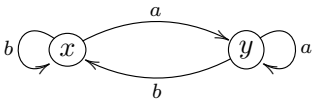
We define $(X, \alpha) \models D$ – and say: *the automaton* (X, α) *satisfies* D – by

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

□

Example 4.3 We consider the automaton (Z, γ) defined by the following diagram:

$(Z, \gamma) \;=$



Here are some examples of equations:

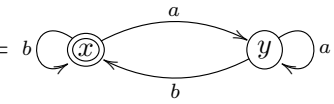
$$(Z, x, \gamma) \models \{b = \varepsilon,\; ab = \varepsilon,\; aa = a\}$$
$$(Z, y, \gamma) \models \{a = \varepsilon,\; ba = \varepsilon,\; bb = b\}$$

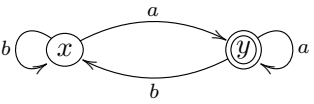
Taking the intersection of the (bisimulation equivalences generated by) these sets, we obtain that

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

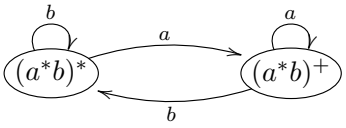
The above set of equations or, again more precisely, the bisimulation equivalence relation on (A^*, σ) generated by it, is the largest set of equations satisfied by (Z, γ).

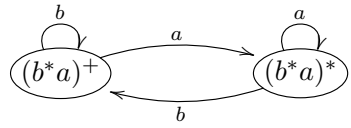
For examples of coequations, we consider the following 2 (out of all 4 possible) coloured versions of (Z, γ):

$(Z, c, \gamma) =$ 

$(Z, d, \gamma) =$ 

(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

$\mathsf{im}(o_c) =$ 

$\mathsf{im}(o_d) =$ 

It follows that

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

□

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). For notational convenience, we assume X to be finite but nothing will depend on that assumption.

Definition 5.1 [free automaton, $\mathsf{Eq}(X, \alpha)$] Let $X = \{x_1, \ldots, x_n\}$ be the set of states of a finite automaton (X, α). We define a pointed automaton $\mathsf{free}(X, \alpha)$ in two steps, as follows:

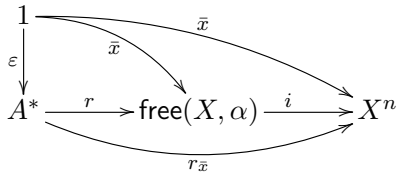
(i) First, we take the product of the n pointed automata (X, x_i, α) that we obtain by letting the initial element x_i range over X. This yields a pointed automaton $(\Pi X, \bar{x}, \bar{\alpha})$ with

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

(where $X_x = X$), with $\bar{x} = (x_1, \ldots, x_n)$, and with $\bar{\alpha} : \Pi X \to (\Pi X)^A$ defined by

$$\bar{\alpha}(y_1, \ldots, y_n)(a) = ((y_1)_a, \ldots, (y_n)_a)$$

(ii) Next we define $(\mathsf{free}(X,\alpha), \bar{x}, \bar{\alpha})$ by $\mathsf{free}(X,\alpha) = \mathsf{im}(r_{\bar{x}})$, where $r_{\bar{x}}$ is the reachability map for $(\Pi X, \bar{x}, \bar{\alpha})$:



Furthermore, we define the following set of equations:

$$\mathsf{Eq}(X,\alpha) = \mathsf{ker}(r)$$

where r is the reachability map for $(\mathsf{free}(X,\alpha), \bar{x}, \bar{\alpha})$. □

Note that

$$\mathsf{free}(X,\alpha) \cong A^*/\mathsf{Eq}(X,\alpha)$$

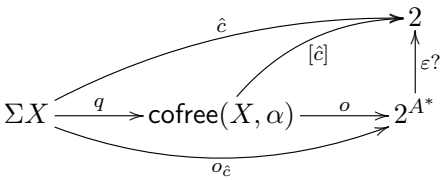
Definition 5.2 [cofree automaton, $\mathsf{coEq}(X,\alpha)$] Let $X = \{x_1, \ldots, x_n\}$ be the set of states of a finite automaton (X,α). We define a coloured automaton $\mathsf{cofree}(X,\alpha)$ in two steps, as follows:

(i) First, we take the coproduct of the 2^n 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

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

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

(ii) Next we define $(\mathsf{cofree}(X,\alpha), [\hat{c}], [\hat{\alpha}])$ by $\mathsf{cofree}(X,\alpha) = \Sigma X/\mathsf{ker}(o_{\hat{c}})$, where $o_{\hat{c}}$ is the observability map for $(\Sigma X, \hat{c}, \hat{\alpha})$:



and where $[\hat{c}]$ and $[\hat{\alpha}]$ are the extensions of $\hat{c}$ and $\hat{\alpha}$ to equivalence classes.

Furthermore, we define

$$\mathsf{coEq}(X,\alpha) = \mathsf{im}(o)$$

where o is the observability map for $(\mathsf{cofree}(X,\alpha), [\hat{c}], [\alpha])$. □

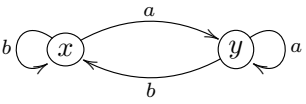
Note that

$$\mathsf{cofree}(X,\alpha) \cong \mathsf{coEq}(X,\alpha)$$

Theorem 5.3 *The set* $\mathsf{Eq}(X,\alpha)$ *is the* largest *set of equations satisfied by* (X,α). *The set* $\mathsf{coEq}(X,\alpha)$ *is the* smallest *set of coequations satisfied by* (X,α). □

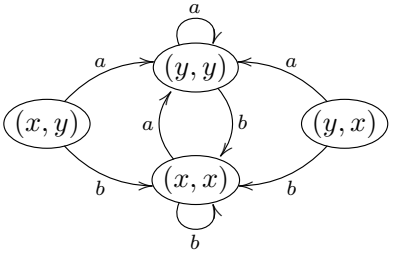
Example 5.4 [Example 4.3 continued] We consider our previous example

$(Z,\gamma) \;=$



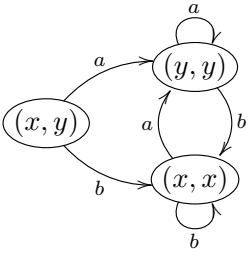
The product of (Z,x,γ) and (Z,y,γ) is:

$(\Pi Z,(x,y),\bar{\gamma}) \;=$



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

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



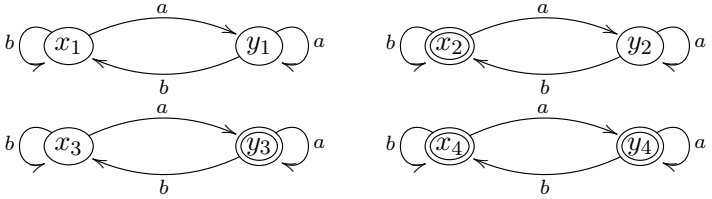
The set $\mathsf{Eq}(Z,\gamma)$ is defined as $\mathsf{ker}(r_{(x,y)})$, and consists of (the smallest bisimulation equivalence on (A^*,σ) generated by)

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

This 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

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

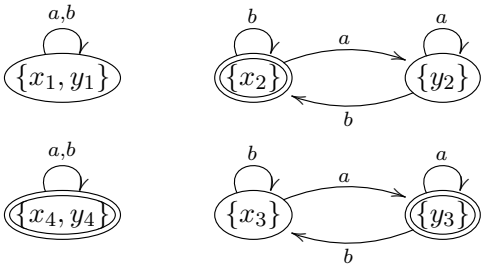


The observability map $o_{\hat{c}} : \Sigma Z \to 2^{A^*}$ is given by

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

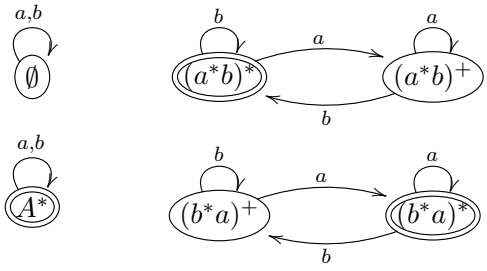
Computing the quotient $\Sigma Z/\mathsf{ker}(o_{\hat{c}})$ yields:

$(\mathsf{cofree}(Z,\gamma), [\hat{c}], [\hat{\gamma}]) =$



The image of this automaton under the reachability map $o : \mathsf{cofree}(Z,\gamma) \to 2^{A^*}$ is

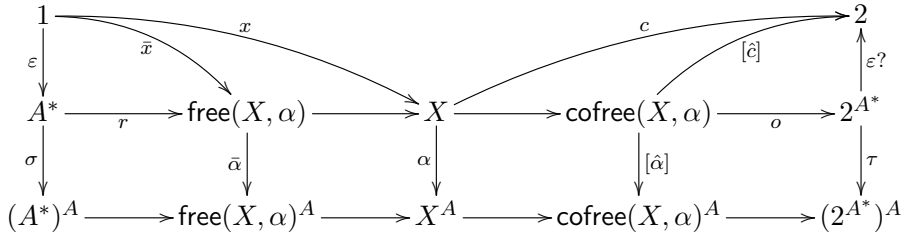
$\mathsf{coEq}(Z,\gamma) =$



(3)

This is the smallest set of coequations satisfied by (Z,γ). □

Summarizing the present section, we have obtained, for every automaton (X,α), the following refinement of (2):



where x ranges over the elements of X and c ranges over all possible colourings of X. The free and cofree automata represent the largest set of equations and the smallest set of coequations satisfied by (X,α):

$$\mathsf{Eq}(X,\alpha) = \mathsf{ker}(r) \qquad \mathsf{coEq}(X,\alpha) = \mathsf{im}(o)$$

Note that the free and cofree automata are constructed for the automaton (X,α),

without point and without colouring. In conclusion, let us mention again that all of the above easily generalises to *infinite* X.

6 Varieties and covarieties

We define varieties and covarieties by means of equations and coequations, first for automata and next for languages.

Definition 6.1 [variety of automata] For every set E of equations we define the *variety* V_E by

$$V_E = \{\, (X,\alpha) \mid (X,\alpha) \models E \,\}$$

□

Definition 6.2 [covariety of automata] For every set D of coequations we define the *covariety* C_D by

$$C_D = \{\, (X,\alpha) \mid (X,\alpha) \models D \,\}$$

□

Every variety of automata defines a set of languages, which we will again call a variety. Dually, every covariety of automata defines a set of languages , which we will again call a covariety.

Definition 6.3 [variety and covariety of languages] Let V_E be a variety of automata. We define the *variety of languages* $L(V_E)$ by

$$L(V_E) = \{\, L \in 2^{A^*} \mid \langle L \rangle \in V_E \,\}$$

(where $\langle L \rangle$ is the subautomaton of $(2^{A^*}, \tau)$ generated by L). Dually, let C_D be a covariety of automata. We define the *covariety of languages* $L(C_D)$ by

$$L(C_D) = \{\, L \in 2^{A^*} \mid \langle L \rangle \in C_D \,\}$$

□

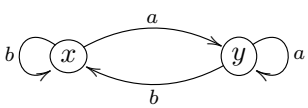
Proposition 6.4 *Every variety V_E is closed under the formation of* subautomata, homomorphic images, *and* products. □

Proposition 6.5 *Every covariety C_D is closed under the formation of* subautomata, homomorphic images, *and* coproducts. □

Proposition 6.6 *A covariety C_D is generally* not *closed under products.*

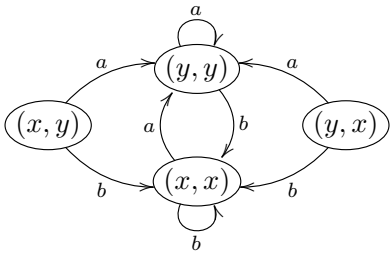
Proof. We give an example of a covariety that is not closed under products. We recall from Example 5.4 the automaton

$$(Z,\gamma) \quad = \quad$$



We saw that $(Z, \gamma) \models D$, with $D = \mathsf{coEq}(Z, \gamma)$ as in (3). The product of (Z, γ) with itself is

$(Z^2, \bar{\gamma}) \;=$



We define a colouring $c : Z^2 \to 2$ by

$c((x,y))$	$c((y,y))$	$c((x,x))$	$c((y,x))$
0	1	1	0

This colouring c induces the observability map $o_c : Z^2 \to 2^{A^*}$, given by

$o_c((x,y))$	$o_c((y,y))$	$o_c((x,x))$	$o_c((y,x))$
A^+	A^*	A^*	A^+

Because $A^+ \notin D$, the automaton $(Z^2, \bar{\gamma}) \not\models D$. Thus C_D is not closed under products. □

Corollary 6.7 Not *every covariety C_D is also a variety.* □

Here are some elementary properties of (co)equations and (covarieties).

Proposition 6.8 *For every set of equations $E \subseteq A^* \times A^*$,*

$$L(V_E) = \{L \in 2^{A^*} \mid \forall (v, w) \in \tilde{E},\ L_v = L_w\}$$

where $\tilde{E}$ is the smallest congruence relation containing E. □

Theorem 6.9 (on equations and varieties) *Let $E \subseteq A^* \times A^*$ be a set of equations. The following statements are equivalent:*

0. *E is a congruence*
1. *$E = \mathsf{Eq}(X, \alpha)$ for some automaton (X, α)*
2. *$(A^*/E, [\sigma]) \models E$*
3. *$\mathsf{Eq}(A^*/E, [\sigma]) = E$*

(with σ as in (2)). Furthermore, any of the above implies:

4. *$L(V_E) = \{L \in 2^{A^*} \mid \forall (v, w) \in E,\ L_v = L_w\}$.*

□

Theorem 6.10 (on coequations and covarieties) *Let $D \subseteq 2^{A^*}$ be a set of coequations. The following statements are equivalent:*

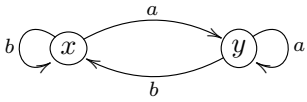
1. $D = \mathsf{coEq}(X, \alpha)$ *for some automaton* (X, α)
2. $(D, \tau) \models D$
3. $\mathsf{coEq}(D, \tau) = D$
4. $L(C_D) = D$

(with τ *as in (2)).* □

Corollary 6.11 *Every variety of languages* $L(V_E)$ *is also a covariety of languages.* □

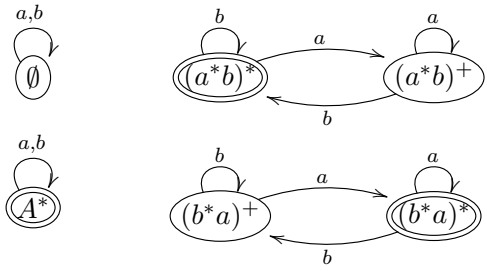
Example 6.12 [Example 5.4 continued] Recall the automaton

$$(Z, \gamma) \quad = \quad$$



and recall

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



The smallest *covariety* containing (Z, γ) is

$$C_{\mathsf{coEq}(Z,\gamma)}$$

It contains the languages

$$L(C_{\mathsf{coEq}(Z,\gamma)}) = \{\, \emptyset,\ (a^*b)^*,\ (a^*b)^+,\ (b^*a)^*,\ (b^*a)^+,\ A^* \,\}$$

The smallest *variety* containing (Z, γ) is

$$V_{\mathsf{Eq}(Z,\gamma)}$$

were we recall that $\mathsf{Eq}(Z, \gamma)$ is the smallest bisimulation equivalence (in fact, a congruence) generated by the set

$$\{aa = a,\ bb = b,\ ab = b,\ ba = a\}$$

We have

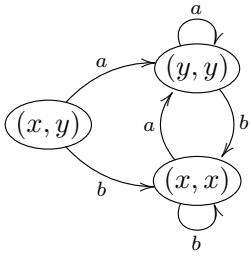
$$\begin{aligned} L(V_{\mathsf{Eq}(Z,\gamma)}) &= \{L \in 2^{A^*} \mid (L_{aa} = L_a) \ \wedge\ (L_{bb} = L_b) \ \wedge\ (L_{ab} = L_b) \ \wedge\ (L_{ba} = L_a)\,\} \\ &= \{\, \emptyset, 1,\ (a^*b)^*,\ (a^*b)^+,\ (b^*a)^*,\ (b^*a)^+,\ A^+,\ A^* \,\} \end{aligned}$$

The latter set of languages can be, equivalently, determined using the fact that

$$\begin{aligned} V_{\mathsf{Eq}(Z,\gamma)} &= C_{\mathsf{coEq}(\,(A^*,\sigma)/\mathsf{Eq}(Z,\gamma)\,)} \\ &= C_{\mathsf{coEq}(\,\mathsf{free}(Z,\gamma)\,)} \end{aligned}$$

To this end, we recall that

$$(\mathsf{free}(Z,\gamma),(x,y),\bar{\gamma}) \quad = \quad$$



and compute $\mathsf{coEq}(\,\mathsf{free}(Z,\gamma)\,)$ by means of the following table, which contains all possible colourings c of $\mathsf{free}(Z,\gamma)$, together with the corresponding value of o_c:

c	$c((x,y))$	$c((y,y))$	$c((x,x))$	$o_c((x,y))$	$o_c((y,y))$	$o_c((x,x))$
c_1	0	0	0	$\emptyset$	$\emptyset$	$\emptyset$
c_2	0	0	1	$(a^*b)^+$	$(a^*b)^+$	$(a^*b)^*$
c_3	0	1	0	$(b^*a)^+$	$(b^*a)^*$	$(b^*a)^+$
c_4	0	1	1	A^+	A^*	A^*
c_5	1	0	0	1	$\emptyset$	$\emptyset$
c_6	1	0	1	$(a^*b)^*$	$(a^*b)^+$	$(a^*b)^*$
c_7	1	1	0	$(b^*a)^*$	$(b^*a)^*$	$(b^*a)^+$
c_8	1	1	1	A^*	A^*	A^*

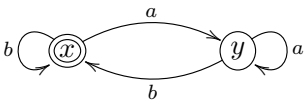
In the end, this leads to the same set of languages. We conclude this example by observing that

$$L(C_{\mathsf{coEq}(Z,\gamma)}) \subseteq L(V_{\mathsf{Eq}(Z,\gamma)})$$

as expected. □

Example 6.13 Here we focus on a single given language, say: $L = (a^*b)^*$. A minimal automaton for L is

$$(Z,x,c,\gamma) \quad = \quad$$



It follows from Example 6.12 that the smallest covariety of languages containing L is

$$L(C_{\mathsf{coEq}(Z,\gamma)}) = \{\,\emptyset,\ (a^*b)^*,\ (a^*b)^+,\ (b^*a)^*,\ (b^*a)^+,\ A^*\,\}$$

and that the smallest variety containing L is

$$L(V_{\mathsf{Eq}(Z,\gamma)}) = \{\, \emptyset, 1,\ (a^*b)^*,\ (a^*b)^+,\ (b^*a)^*,\ (b^*a)^+,\ A^+,\ A^* \,\}$$

□

Example 6.14 Here are some further examples of varieties and covarieties.

(i) The smallest congruence generated by $\{\, a = \varepsilon, b = \varepsilon \,\}$ is $E = A^* \times A^*$. As a consequence,

$$L(V_E) = \{\, \emptyset,\ A^* \,\}$$

The same for $E = \{\, b = \varepsilon, ab = \varepsilon, aa = a \,\}$.

(ii) If E is the smallest congruence generated by $\{aa = \varepsilon,\ b = \varepsilon \,\}$, then

$$L(V_E) = \{\, \emptyset,\ ((ab^*a) + b)^*,\ ((ab^*a) + b)^*ab^*,\ \{a, b\}^* \,\}$$

(iii) If E is the smallest congruence generated by $\{aa = \varepsilon,\ bb = \varepsilon \,\}$, then the variety $L(V_E)$ is infinite and contains both rational and non-rational languages.

(iv) For $D = 2^{A^*}$, the covariety C_D contains *all* automata (X, α).

(v) For $D = \mathsf{rat}(2^{A^*})$,

$$C_D = \{(X, \alpha) \mid (X, \alpha) \text{ is finitely generated}\,\}$$

that is, all (X, α) such that $\langle x \rangle \subseteq X$ is finite, for all $x \in X$.

(vi) If $D = \{\, \{a\},\ 1,\ \emptyset \,\}$ then $C_D = \emptyset$.

7 Transition monoids

For every (rational) language, one can construct its so-called *syntactic monoid* (that is, the transition monoid of its minimal automaton). Next every (classical, algebraic) variety V of monoids determines a class of languages L by the requirement that its syntactic monoid belongs to V. This is, in short, Eilenberg's definition of a variety of languages. In this section, we take a first step towards an understanding of the relation between Eilenberg's definition and the present one, by the observation that $\mathsf{free}(X, \alpha)$, for every automaton (X, α), is isomorphic to its transition monoid.

A *monoid* $(M, \cdot, 1)$ consists of a set M, a binary operation of multiplication that is associative, and a unit 1 with $m \cdot 1 = 1 \cdot m = m$. For every set, there is the monoid

$$(X^X, \cdot, 1_X)$$

defined by

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

Because of the isomorphism

$$X \to X^A \ \cong\ A \to X^X$$

we have for every automaton (X, α) and $a \in A$ a function

$$\tilde{a} : X \to X \qquad \tilde{a}(x) = \alpha(x)(a) = x_a$$

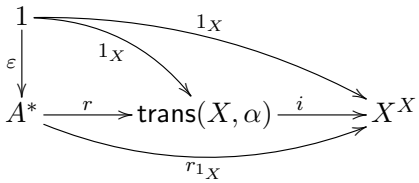
We use it to define for every automaton (X, α) a pointed automaton

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

Next we define the *transition monoid* (cf. [18])

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

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



(where $r(a_1 \cdots a_n) = \tilde{a_1} \cdots \tilde{a_n}$, for $a_1 \cdots a_n \in A^*$).

Theorem 7.1 *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. □

This elementary observation should form the basis for a detailed comparison of the present definition of variety of languages and Eilenberg's definition.

References

[1] M.A. Arbib and E.G. Manes. Adjoint machines, state-behaviour machines, and duality. *Journal of Pure and Applied Algebra*, 6:313–344, 1975.

[2] M.A. Arbib and H.P. Zeiger. On the relevance of abstract algebra to control theory. *Automatica*, 5:589–606, 1969.

[3] F. Bonchi, M. Bonsangue, H. Hansen, P. Panangaden, J. Rutten, and A. Silva. Algebra-coalgebra duality in Brzozowski's minimization algorithm. 2013. Submitted.

[4] A. Ballester-Bolinches, J.-E. Pin, and X. Soler-Escriva. Formations of finite monoids and formal languages: Eilenberg's variety theorem revisited. *Forum Mathematicum*, 2012.

[5] A. Ballester-Bolinches, J.-E. Pin, and X. Soler-Escriva. Languages associated with saturated formations of groups. *Forum Mathematicum*, 2013.

[6] F. Bonchi, M. Bonsangue, J. Rutten, and A. Silva. Brzozowski's algorithm (co)algebraically. In R. Constable and A. Silva, editors, *Logic and Program Semantics.*, volume 7230 of *LNCS*, pages 12–23, 2012.

[7] M. Bidoit, R. Hennicker, and A. Kurz. On the duality between observability and reachability. In Furio Honsell and Marino Miculan, editors, *FoSSaCS*, volume 2030 of *Lect. Notes in Comp. Sci.*, pages 72–87. Springer, 2001.

[8] J.A. Brzozowski. Derivatives of regular expressions. *Journal of the ACM*, 11(4):481–494, 1964.

[9] C. Cirstea. On specification logics for algebra-coalgebra structures: Reconciling reachability and observability. In *Proceedings FoSSaCS*, pages 82–97, 2002.

[10] J.H. Conway. *Regular algebra and finite machines.* Chapman and Hall, 1971.

[11] S. Eilenberg. *Automata, languages and machines (Vol. A).* Pure and applied mathematics. Academic Press, 1974.

[12] S. Eilenberg. *Automata, languages and machines (Vol. B).* Pure and applied mathematics. Academic Press, 1976.

[13] M. Gehrke, S. Grigorieff, and J.-E. Pin. Duality and equational theory of regular languages. In *Proceedings ICALP*, volume 5126 of *LNCS*, pages 246–257, 2008.

[14] R. Kalman. On the general theory of control systems. *IRE Transactions on Automatic Control*, 4(3):110–110, 1959.

[15] R. E. Kalman, P. L. Falb, and M. A. Arbib. *Topics in Mathematical Systems Theory.* McGraw Hill, 1969.

[16] S.C. Kleene. Representation of events in nerve nets and finite automata. In Shannon and McCarthy, editors, *Automata Studies*, pages 3–41. Princeton Univ. Press, 1956.

[17] E.G. Manes and M.A. Arbib. *Algebraic approaches to program semantics.* Texts and monographs in computer science. Springer-Verlag, 1986.

[18] J.-E. Pin. Syntactic semigroups. *Handbook of language theory, Vol. I*, pages 679–746, 1997.

[19] J.J.M.M. Rutten. Automata and coinduction (an exercise in coalgebra). In D. Sangiorgi and R. de Simone, editors, *Proceedings of CONCUR'98*, volume 1466 of *LNCS*, pages 194–218, 1998.

[20] J.J.M.M. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science*, 249(1):3–80, 2000. Fundamental Study.