Regular Varieties of Automata and Coequations

J. Salamanca[1], A. Ballester-Bolinches[2], M.M. Bonsangue[1,3],
E. Cosme-Llópez[2], and J.J.M.M. Rutten[1,4]

[1] CWI Amsterdam
[2] Universitat de València
[3]LIACS - Leiden University
[4] Radboud University Nijmegen

Abstract. In this paper we use a duality result between equations and coequations for automata, proved by Ballester-Bolinches, Cosme-Llópez, and Rutten to characterize nonempty classes of deterministic automata that are closed under products, subautomata, homomorphic images, and sums. One characterization is as classes of automata defined by regular equations and the second one is as classes of automata satisfying sets of coequations called varieties of languages. We show how our results are related to Birkhoff's theorem for regular varieties.

1 Introduction

Initial algebras provide minimal canonical models for inductive data types, recursive definition principles of functions and induction as a corresponding proof principle [7]. Over the last decade, coalgebras have emerged as a mathematical structure suitable for capturing infinite data structures and infinite computations [12]. Final coalgebras are the categorical dual of initial algebras. They represent infinite data or behavior defined by observations rather than constructors, and come equipped with corecursive definitions of functions and coinduction as a dual proof principle of induction [1].

More generally, algebraic theories of free algebras are specified by equations. Dually, cofree coalgebras generalize final coalgebras, where coequations are used instead of equations. Intuitively, coequations can be thought of as behaviours, or pattern specifications [12], that a coalgebra is supposed to exhibit or adhere to. By Birkhoff's celebrated theorem [5], a class of algebras is equationally defined if and only if it is a variety, i.e., closed under homomorphic images, subalgebras and products. Dually, on the coalgebraic side [12, 3, 8], generally less is known about the notions of coequations and covarieties.

In the present paper, we study deterministic automata both from an algebraic perspective and a coalgebraic one. From the algebraic perspective, deterministic automata are algebras with unary operations. In this context, an equation is just a pair of words, and it holds in an automaton if for every initial state, the states reached from that state by both words are the same. Coalgebraically, an automaton is a deterministic transition system with final states (the observations). A coequation is then a set of languages, and an automaton satisfies a coequation if for every possible observation (colouring the states as either final or not) the language accepted by the automaton is within the specified coequation.

In [4] a subset of the authors have established a new duality result between equations and coequations. Building on their work, we show that classes of deterministic automata closed under products, subautomata, homomorphic images, and sums are definable both by congruences and by varieties of languages. The first characterization, by congruences, is algebraic. In this case, congruences are equational theories of *regular equations* which give rise to regular varieties. An equation $e_1 = e_2$ is regular if the sets of variables occurring in e_1 and e_2 are the same. It is worth mentioning that another characterization of regular varieties was given by Taylor [13]; we will show how that characterization relates to the one we will present here.

The second characterization is a coalgebraic one. Here coequations are used to define classes of automata. Coequations are given by sets of languages, and a central concept in our characterization is that of variety of languages: sets of languages that are both (complete atomic) Boolean algebras and closed under right and left derivatives (details will follow). The coalgebraic characterization will look less familiar than the algebraic one. It is interesting since it will turn out to be equivalent to definitions by so-called regular equations, thus yielding a novel restriction of Birkhoff's theorem.

As a consequence, classes of automata closed under products, subautomata, homomorphic images, and sums can be defined by both equations and coequations. In fact, the first three closure properties characterize an algebraic variety (cf. Birkhoff theorem [5]), whereas the last three closure properties define a coalgebraic covariety [12, 3, 8]. Our result fits into the recent line of work which uses Stone-like duality as a tool for proving the correspondence between local varieties of regular languages and local pseudovarieties of monoids [9, 2]. The main difference is that we do not impose any restriction on the state space of the automata and on the size of the input alphabet.

2 Preliminaries

In this section we introduce the notation and main concepts we will use in the paper. (See [4] for more details). Given two sets X and Y we define

$$Y^X = \{f \mid f : X \to Y\}$$

For a function $f \in Y^X$, we define the *kernel* and the *image* of f by

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

and

$$\mathrm{Im}(f) = \{f(x) \mid x \in X\}.$$

For $Y_0 \subseteq Y$, the set $f^{-1}(Y_0) \subseteq X$ is defined as

$$f^{-1}(Y_0) = \{x \in X \mid f(x) \in Y_0\}.$$

We define the set $2 = \{0,1\}$ and, for any set X and $B \subseteq X$, we define the function $\chi_B : X \to 2$ by

$$\chi_B(x) = \begin{cases} 1 & \text{if } x \in B, \\ 0 & \text{if } x \notin B. \end{cases}$$

If $B \subseteq X$ then $\chi_B \in 2^X$, and for any $f \in 2^X$ we get the subset $f^{-1}(\{1\})$ of X. The previous correspondence between subsets of X and elements in 2^X is bijective, so elements in 2^X and subsets of X will often be identified.

For any family of sets $\{X_i\}_{i \in I}$, we define their disjoint union by

$$\sum_{i \in I} X_i = \bigcup_{i \in I} \{i\} \times X_i$$

For any set A we denote by A^* the free monoid with generators A, its identity element will be denoted by ϵ. Elements in 2^{A^*} are called *languages* over A, which can also be seen as subsets of A^*. Given a language $L \in 2^{A^*}$ and $w \in L$, we define the *left derivative* ${}_wL$ of L with respect to w and the *right derivative* L_w of L with respect to w as the elements ${}_wL, L_w \in 2^{A^*}$ such that for every $u \in A^*$

$${}_wL(u) = L(uw) \text{ and } L_w(u) = L(wu).$$

Let A be a (not necessarily finite) alphabet. A *deterministic automaton* on A is a pair (X, α) where $\alpha : X \times A \to X$ is a function. Let (X, α) and (Y, β) be deterministic automata on A. We say that (X, α) is a *subautomaton* of (Y, β) if $X \subseteq Y$ and for every $x \in X$ and $a \in A$, $\alpha(x, a) = \beta(x, a) \in X$. A function $h : X \to Y$ is a *homomorphism* from (X, α) to (Y, β) if for every $x \in X$ and $a \in A$, $h(\alpha(x, a)) = \beta(h(x), a)$. We say that (Y, β) is a *homomorphic image* of (X, α) if there exists a surjective homomorphism $h : X \to Y$.

The *product* of a family $\mathcal{X} = \{(X_i, \alpha_i)\}_{i \in I}$ of deterministic automata is the deterministic automaton $\prod_{i \in I}(X_i, \alpha_i) = (\prod_{i \in I} X_i, \bar{\alpha})$ where

$$\bar{\alpha}(f, a)(i) = \alpha_i(f(i), a),$$

Furthermore, the *sum* of the family $\mathcal{X}$ is defined as the deterministic automaton $\sum_{i \in I}(X_i, \alpha_i) = (\sum_{i \in I} X_i, \hat{\alpha})$ where

$$\hat{\alpha}((i, x), (a)) = (i, \alpha_i(x, a)).$$

Given an automaton (X, α) we can add an initial state $x_0 \in X$ or a colouring $c : X \to 2$ to get a *pointed automaton* (X, x_0, α) or a *coloured automaton* (X, c, α), respectively. For a deterministic automaton (X, α), $x \in X$, and $u \in A^*$, we define $u(x) \in X$ inductively as follows

$$u(x) = \begin{cases} x & \text{if } u = \epsilon, \\ \alpha(w(x), a) & \text{if } u = wa, \end{cases}$$

thus $u(x)$ is the state we reach from x by processing the word u.

By using the correspondence

$$\alpha : X \times A \to X \;\;\Leftrightarrow\;\; \alpha' : X \to X^A$$

given by $\alpha(x, a) = \alpha'(x)(a)$, we have that pointed automata are F-algebras for the endofunctor F on Set given by $F(X) = 1 + (A \times X)$. Dually, coloured automata are G-coalgebras for the endofunctor G on Set given by $G(X) = 2 \times X^A$.

The initial F–algebra is the pointed automaton (A^*, ϵ, τ), where the states are strings over A, the empty string ϵ is the initial state, and the transition function τ is concatenation, that is $\tau(w, a) = wa$, for all $w \in A^*$ and $a \in A$. For any pointed automaton (X, x_0, α), the unique F–algebra morphism

$$r_{x_0} : (A^*, \epsilon, \tau) \to (X, x_0, \alpha)$$

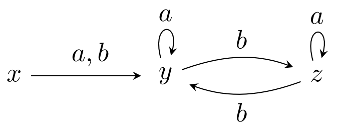
is given by $r_{x_0}(w) = w(x_0)$. The F–algebra morphism r_{x_0} is called the *reachability* map, and it maps every word w to the state $w(x_0)$ which is the state we reach from x_0 by processing the word w.

Dually, the final G–coalgebra is the coloured automaton $(2^{A^*}, \hat{\epsilon}, \hat{\tau})$, where states are languages over A, accepting states are only the languages that contain the empty word ϵ, i.e. $\hat{\epsilon}(L) = L(\epsilon)$, and the transition function $\hat{\tau} : 2^{A^*} \to (2^{A^*})^A$ is the right derivative operation, that is $\hat{\tau}(L)(a) = L_a$, for all $L \in 2^{A^*}$ and $a \in A$. For any coloured automaton (X, c, α), the unique G–coalgebra morphism

$$o_c : (X, c, \alpha) \to (2^{A^*}, \hat{\epsilon}, \hat{\tau})$$

is given by $o_c(x) = \lambda w.c(w(x)) \in 2^{A^*}$. The G–coalgebra morphism o_c is called the *observability* map, and it maps every state x to the language $o_c(x)$ accepted from the state x according to the colouring c.

Example 1. Consider the deterministic automaton (X, α) on $A = \{a, b\}$ given by:



Then we have:

i) The image of the reachability map r_x, with x as initial state, on the word $aabb$ is $r_x(aabb) = y$. Similar calculations, for possible different initial states, are the following:

$$r_x(ba^5ba) = z, \quad r_y(ba^5ba) = y, \quad r_y(b^{11}) = z, \quad r_z(aba) = y, \quad r_x(\epsilon) = x.$$

ii) The image of the observability map o_c for the colouring $c = \{x, z\}$ on the state x (i.e. the language accepted by the automaton if the initial state is x and the set of accepting states is $\{x, z\}$) is

$$o_c(x) = \epsilon \cup (a \cup b)a^*b(a^*ba^*b)^*a^*.$$

□

3 Equations and coequations

In this section we summarize some concepts and facts from [4] that will be used in the following sections.

Let (X, α) be a deterministic automaton on A. An *equation* is a pair $(u, v) \in A^* \times A^*$, sometimes also denoted by $u = v$. Given $(u, v) \in A^* \times A^*$, we define $(X, \alpha) \models_{\mathsf{e}} (u, v)$ – and say: (X, α) satisfies the equation (u, v) – as follows:

$$(X, \alpha) \models_{\mathsf{e}} (u, v) \;\Leftrightarrow\; \forall x \in X \;\; u(x) = v(x) \;\Leftrightarrow\; \forall x \in X \;\; (u, v) \in \ker(r_x),$$

and for any set of equations $E \subseteq A^* \times A^*$ we write $(X, \alpha) \models_{\mathsf{e}} E$ if $(X, \alpha) \models_{\mathsf{e}} (u, v)$ for every $(u, v) \in E$. Basically, an equation (u, v) is satisfied by an automaton if the states reached by u and v from any initial state $x \in X$, are the same.

An equivalence relation C on A^* is a *congruence* on A^* if for any $t, u, v, w \in A^*$, $(t, v) \in C$ and $(u, w) \in C$ imply $(tu, vw) \in C$. If C is a congruence on A^*, the *congruence quotient* A^*/C has a pointed automaton structure $A^*/C = (A^*/C, [\epsilon], \alpha)$ with transition function given by $\alpha([w], a) = [wa]$, which is well defined since C is a congruence.

A set of *coequations* is a subset $D \subseteq 2^{A^*}$. We define $(X, \alpha) \models_{\mathsf{c}} D$ – and say: (X, α) satisfies the set of coequations D – as follows:

$$(X, \alpha) \models_{\mathsf{c}} D \;\Leftrightarrow\; \forall c \in 2^X, x \in X \;\; o_c(x) \in D \;\Leftrightarrow\; \forall c \in 2^X \;\; \mathrm{Im}(o_c) \subseteq D.$$

In other words, an automaton satisfies a coequation D if for every colouring, the language accepted by the automaton, starting from any state, belongs to D. Note that, categorically, coequations are dual to equations [12].

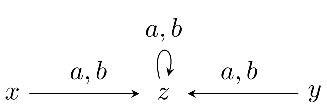
Next we show how to construct the maximum set of equations and the minimum set of coequations satisfied by an automaton (X, α). To get the maximum set of equations of (X, α), we define the pointed deterministic automaton $\mathbf{free}(X, \alpha)$ as follows:

1. Define the pointed deterministic automaton $\prod(X, \alpha) = (\prod_{x \in X} X, \Delta, \hat{\alpha})$ where $\hat{\alpha}$ is the product of α $|X|$ times, that is $\hat{\alpha}(\theta, a)(x) = \alpha(\theta(x), a)$, and $\Delta \in \prod_{x \in X} X$ is given by $\Delta(x) = x$. Then, by initiality of $A^* = (A^*, \epsilon, \tau)$, we get a unique F-algebra homomorphism $r_\Delta : A^* \to \prod(X, \alpha)$.
2. Define $\mathbf{free}(X, \alpha)$ and $\mathbf{Eq}(X, \alpha)$ as

$$\mathbf{free}(X, \alpha) := A^*/\ker(r_\Delta) \text{ and } \mathbf{Eq}(X, \alpha) := \ker(r_\Delta)$$

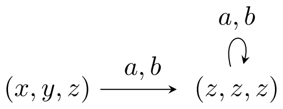
Notice that $\mathbf{free}(X, \alpha)$ has the structure of a pointed automaton.

Example 2. Let $A = \{a, b\}$ and consider the following automaton (X, α) on A:



Then by definition $\mathbf{free}(X, \alpha) = A^*/\ker(r_\Delta) \cong \mathrm{Im}(r_\Delta)$. So in order to construct $\mathbf{free}(X, \alpha)$ we only need to construct the reachable part of $\prod(X, \alpha)$ from the state $\Delta = (x, y, z)$, which gives us the following automaton $\mathrm{Im}(r_\Delta)$:

(†)



In this case $\mathbf{free}(X, \alpha) = A^* / \ker(r_\Delta)$, where $\ker(r_\Delta)$ is the equivalence relation that corresponds to the partition $\{\{\epsilon\}, (a \cup b)^+\}$ of A^*. Hence, $\mathbf{free}(X, \alpha)$ is isomorphic to the automaton (†) in which $(x, y, z) \mapsto [\epsilon]$ and $(z, z, z) \mapsto [a]$. □

By construction, we have the following theorem.

Theorem 3. *(Proposition 6, [4])* $\mathbf{Eq}(X, \alpha)$ *is the maximum set of equations satisfied by* (X, α).

Note that the above equations are just identities for algebras with unary operations in which both the left and right terms use the same variable, that is, identities of the form $p(x) \approx q(y)$ where $p, q \in A^*$, and x and y are variables with $x = y$ (see [7, Definition 11.1]). A similar result as the above theorem can be obtained for any identity $p(x) \approx q(y)$. In order to do that one should consider the functor $F'(X) = A \times X$, where A is a fixed alphabet. Clearly, the free F'-algebra on S generators is the initial algebra for the functor $F'_S(X) := S + F'(X) = S + (A \times X)$. Furthermore, as every identity uses at most two variables it is enough to consider the free F'-algebra on 2 generators in order to express the left and the right term of every identity.

We show next how to construct the minimum set of coequations satisfied by (X, α). In this case, we construct the coloured deterministic automaton $\mathbf{cofree}(X, \alpha) \subseteq 2^{A^*}$, by taking the following steps:

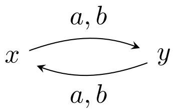
1. Define the coloured automaton $\sum(X, \alpha) = (\sum_{c \in 2^X} X, \Phi, \tilde{\alpha})$ where $\tilde{\alpha}$ and Φ are given by $\tilde{\alpha}(c, x)(a) = (c, a(x))$ and $\Phi(c, x) = c(x)$. Then, by finality of $2^{A^*} = \big(2^{A^*}, \hat{\epsilon}, \hat{\tau}\big)$, we get a unique G-coalgebra homomorphism $o_\Phi : \sum(X, \alpha) \to 2^{A^*}$.
2. Define $\mathbf{cofree}(X, \alpha)$ and $\mathbf{coEq}(X, \alpha)$ as

$$\mathbf{cofree}(X, \alpha) = \mathbf{coEq}(X, \alpha) := \mathrm{Im}(o_\Phi).$$

Similarly as in the case of equations we have the following theorem.

Theorem 4. *(Proposition 6, [4])* $\mathbf{coEq}(X, \alpha)$ *is the minimum set of coequations satisfied by* (X, α).

Example 5. Let $A = \{a, b\}$ and consider the following automaton (X, α) on A:



By definition of $\mathbf{cofree}(X, \alpha)$, we have that

$$\mathbf{cofree}(X, \alpha) = \{o_c(z) \mid c \in 2^X, z \in X = \{x, y\}\} = \{\emptyset, L_{\text{odd}}, L_{\text{even}}, A^*\}$$

where L_{odd} and L_{even} are the sets of words in A^* with an odd and even number of symbols, respectively. □

Next we define varieties of languages, that is, the kind of coequations that we will use in the following section.

Definition 6. *A variety of languages is a set $V \subseteq 2^{A^*}$ such that:*

i) V is a complete atomic Boolean subalgebra of $2^{A^*} = (2^{A^*}, (\ )', \cup, \cap, \emptyset, A^*)$.
ii) V is closed under left and right derivatives: if $L \in V$ then ${}_wL \in V$ and $L_w \in V$, for all $w \in A^*$.

This notion is related to but different from that of *local variety of regular languages*, as defined in [9]: according to our definition above, a variety may contain languages that are non-regular; moreover, a variety has the structure of a *complete atomic* Boolean algebra rather than just a Boolean algebra.

Our main result, in the next section, will be that regular varieties of automata (i.e., defined by regular equations) are characterized by varieties of languages.

There is a correspondence between congruences of A^* and varieties of languages that can be stated as follows.

Theorem 7. *([4]) Let C be a congruence on A^* and let $V \subseteq 2^{A^*}$ be a variety of languages. Then*

i) $\mathbf{cofree}(A^*/C)$ is a variety of languages.
ii) $\mathbf{Eq}(V)$ is a congruence on A^*.
iii) $\mathbf{free} \circ \mathbf{cofree}(A^*/C) = A^*/C$.
iv) $\mathbf{cofree} \circ \mathbf{free}(V) = V$.

Notice that every variety of languages V has a coloured automaton structure $V = (V, \hat{\epsilon}, \hat{\tau})$ because it is closed under the right derivatives. (In ii), we write $\mathbf{Eq}(V)$ rather than $\mathbf{Eq}(V, \hat{\tau})$.) Thus $\mathbf{cofree}(A^*/C)$ has the structure of a coloured automaton, and so expressions iii) and iv) of the theorem above are well defined.

Additionally, we have that for a congruence C on A^*, $\mathbf{cofree}(A^*/C)$ is the complete Boolean subalgebra of 2^{A^*} whose set of atoms is A^*/C. Conversely, given a variety of languages V, $\mathbf{free}(V)$ is the congruence quotient whose associated congruence corresponds to the partition given by the set of atoms of V.

As an application of the previous facts we have the following:

Example 8. Given a family of languages $\mathcal{L} \subseteq 2^{A^*}$, we can construct an automaton (X, α) representing that family in the following sense:

For every $L \in \mathcal{L}$ there exists $x \in X$ and $c \in 2^X$ such that $o_c(x) = L$.

For any given family, we can construct an automaton such that it has the minimum number of states and moreover satisfies the following stronger property:

> There exists $x \in X$ such that for every $L \in \mathcal{L}$ there exists $c \in 2^X$ such that $o_c(x) = L$.

The construction is a follows: let $V(\mathcal{L})$ be the least variety of languages containing $\mathcal{L}$, which always exists. Then the automaton $\mathbf{free}(V(\mathcal{L})) = A^*/\mathbf{Eq}(V(\mathcal{L}))$ has the desired property. In fact, by [4, Lemma 13], there exists, for every $L \in V(\mathcal{L})$, a colouring $c_L : A^*/\mathbf{Eq}(V(\mathcal{L})) \to 2$ such that $o_{c_L}([\epsilon]) = L$. □

The previous example gives us a way to construct a single program (automaton) for a specific set of behaviours (set of languages) in an efficient way (minimum number of states) with the property that the initial configuration (initial state) of the program is the same for every desired behaviour. Here is a small illustration of this fact.

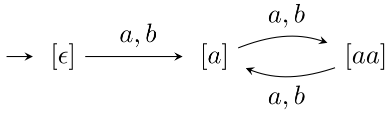
Example 9. Let $A = \{a, b\}$ and consider the following family of languages on A^*

$$\mathcal{L} = \{(a \cup b)^+, L_{\text{odd}}, L_{\text{even}}\}$$

We would like to construct a pointed automaton (X, x_0, α) with the property that for every $L \in \mathcal{L}$ there exists $c_L \in 2^X$ such that $o_{c_L}(x_0) = L$. According to the previous example, we only need to construct the least variety of languages V containing $\mathcal{L}$. In this case, V is the variety of languages (with 8 elements) whose atoms are

$$A_1 = \{\epsilon\}, A_2 = (a \cup b)[(a \cup b)(a \cup b)]^*, \text{ and } A_3 = (a \cup b)(a \cup b)[(a \cup b)(a \cup b)]^*$$

Clearly $\mathcal{L} \subseteq V$ since $(a \cup b)^+ = A_2 \cup A_3$, $L_{\text{odd}} = A_2$, and $L_{\text{even}} = A_1 \cup A_3$. Then the pointed automaton we are looking for is $\mathbf{free}(V)$ which is given by



Clearly, for the colourings $c_1 = \{[a], [aa]\}$, $c_2 = \{[a]\}$, and $c_3 = \{[\epsilon], [aa]\}$ we have that

$$o_{c_1}([\epsilon]) = (a \cup b)^+, o_{c_2}([\epsilon]) = L_{\text{odd}}, \text{ and } o_{c_3}([\epsilon]) = L_{\text{even}}.$$

□

4 Characterization of regular varieties of automata

In this section we show that classes of deterministic automata that are closed under subautomata, products, homomorphic images, and sums are the same as classes of regular varieties of automata (defined below). Furthermore we will give a characterization in terms of coequations.

For an alphabet A let τ_A be the type $\tau_A = \{f_a\}_{a \in A}$ where each f_a is a unary operation symbol. Clearly, an algebra of type τ_A is a deterministic automaton over A since we have the correspondence

$$(X, \alpha : X \times A \to X) \iff (X, \{f_a : X \to X\}_{a \in A})$$

where $f_a(x) = \alpha(x, a)$, for all $a \in A$ and $x \in X$.

Every term $f_{a_n} f_{a_{n-1}} \cdots f_{a_1}(x)$ of type τ_A will be written as $u(x)$ where $u = a_1 \cdots a_{n-1} a_n$. For any $u, v \in A^*$ there are two possible *identities*

$$u(x) \approx v(x) \text{ and } u(x) \approx v(y)$$

which correspond to the formulas

$$\forall x[u(x) = v(x)] \text{ and } \forall x, y[u(x) = v(y)].$$

Identities of the form $u(x) \approx v(x)$ are called *regular identities.* They were first introduced by Płonka [11] and can be identified with pairs $(u, v) \in A^* \times A^*$ as in the previous section.

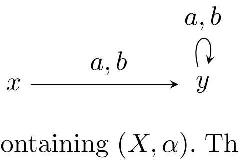
For any set of identities E of type τ_A we define the class $M_{\mathsf{e}}(E)$ of automata satisfying E by

$$M_{\mathsf{e}}(E) = \{(X, \alpha) \mid (X, \alpha) \models_{\mathsf{e}} E\},$$

where E can include identities of the form $u(x) \approx v(y)$. We will write $E \subseteq A^* \times A^*$ if all the identities in E are of the form $u(x) \approx v(x)$, that is if they are regular identities.

Clearly, for any set E of identities, $M_{\mathsf{e}}(E)$ is a *variety*, that is, a class of automata that is closed under products, subautomata, and homomorphic images. (Note that now, we are talking of a variety of *automata*, as opposed to our earlier notion of variety of *languages*.) By Birkhoff's theorem [5], any variety V of automata on A is of the form $V = M_{\mathsf{e}}(E)$ for some set E of identities. Classes of automata of the form $M_{\mathsf{e}}(R)$ where R is a set of regular identities are called *regular varieties of automata.* The next example shows a variety of automata that is not regular.

Example 10. Let $A = \{a, b\}$, and consider the variety V_1 generated by the automaton (X, α) on A given by



that is V_1 is the least variety containing (X, α). Then, an automaton $(Y, \sigma) \in V_1$ if and only if there exists $s \in Y$ such that for every $z \in Y$, $\sigma(z, a) = \sigma(z, b) = s$, that is, an automaton is in V_1 if and only if there is no difference between a and b transitions, and there is a state ('*sink*') that is reachable from any state by inputting the letter a (or, equivalently, that is reachable from any state by inputting the letter b).

Let E be a set of identities such that $V_1 = M_{\mathsf{e}}(E)$. If $E \subseteq A^* \times A^*$ then $(X,\alpha)+(X,\alpha) \in M_{\mathsf{e}}(E)$ but $(X+X,\hat{\alpha}) = (X,\alpha)+(X,\alpha) \notin V_1$ (as $\hat{\alpha}((0,x),a) = (0,y) \neq (1,y) = \hat{\alpha}((1,x),a)$ meaning that there is no sink, a contradiction). Observe that a set of defining identities for V_1 is $E = \{a(x) \approx b(y)\}$. □

It's worth mentioning that a Birkhoff-like theorem for regular varieties was formulated by Taylor in [13, page 4]. Applied to the present situation, it says that a class K of automata on A is a regular variety if and only if K is closed under products, subalgebras, homomorphic images, and $\mathbf{2}_\emptyset \in K$, where $\mathbf{2}_\emptyset$ is the algebra

$$\mathbf{2}_\emptyset = (\{0,1\}, \{f_a : 2 \to 2\}_{a \in A})$$

where f_a is the identity function on the set 2, for every $a \in A$. If $A = \{a, b\}$ then the algebra (automaton) $\mathbf{2}_\emptyset$ is given by

This automaton has, in general, the property that an identity holds in $\mathbf{2}_\emptyset$ if and only if it is regular [10, Lemma 2.1].

Disconnected automata cannot satisfy equations of the form $u(x) \approx v(y)$, which property is the key fact to obtain the following theorem.

Theorem 11. *Let K be a nonempty class of automata on A. The following are equivalent:*

i) K is closed under products, subalgebras, homomorphic images, and sums.
ii) $K = M_{\mathsf{e}}(C)$ for some congruence $C \subseteq A^* \times A^*$ on A^*. That is, K is a regular variety.

Proof. *i)* ⇒ *ii)* By Birkhoff's theorem, $K = M_{\mathsf{e}}(E)$ for some set of equations. Now, E cannot contain identities of the form $u(x) \approx v(y)$ because K is closed under sums. Clearly, $M_{\mathsf{e}}(E) = M_{\mathsf{e}}(\langle E \rangle)$ where $\langle E \rangle$ denotes the least congruence containing E.

ii) ⇒ *i)* Identities of the form $u(x) \approx v(x)$ are preserved under products, subalgebras, homomorphic images, and sums. □

Combining the previous theorem with the characterization of regular varieties given by Taylor, we have that a variety of automata is closed under sums if and only if it contains $\mathbf{2}_\emptyset$. Which can be proved directly by noticing that $\mathbf{2}_\emptyset$ is the sum of the trivial (one element) algebra and, conversely, that the sum of a family $\{(X_i, \alpha_i)\}_{i \in I}$ can be obtained as a homomorphic image of the algebra $\prod_{i \in I}(X_i, \alpha_i) \times \prod_{i \in I} \mathbf{2}_\emptyset$. In fact, let $\phi : I \to \prod_{i \in I} 2$ be an injective function and $i_0 \in I$ a fixed element, then the function $h : \prod_{i\in I} X_i \times \prod_{i\in I} 2 \to \sum_{i\in I} X_i$ defined by

$$h(f,p) = \begin{cases} (i_0, f(i_0)) & \text{if } p \notin \mathrm{Im}(\phi), \\ (i, f(i)) & \text{if } p \in \mathrm{Im}(\phi) \text{ and } \phi(i) = p, \end{cases}$$

is a surjective homomorphism onto $\sum_{i\in I}(X_i, \alpha_i)$. Notice that h is well-defined since ϕ is injective.

Similarly to the equational case, for a set of coequations $D \subseteq 2^{A^*}$, we define the class $M_{\mathsf{c}}(D)$ of automata satisfying the set of coequations D by

$$M_{\mathsf{c}}(D) = \{(X,\alpha) \mid (X,\alpha) \models_{\mathsf{c}} D\}.$$

Lemma 12. *Let C be a congruence on A^*, then*

$$M_{\mathsf{c}}(\mathbf{cofree}(A^*/C)) = M_{\mathsf{e}}(C)$$

Proof. $M_{\mathsf{c}}(\mathbf{cofree}(A^*/C)) \subseteq M_{\mathsf{e}}(C)$: Let (X,α) be an automaton such that $(X,\alpha) \models_{\mathsf{c}} \mathbf{cofree}(A^*/C)$. We have to show that $(X,\alpha) \models_{\mathsf{e}} C$. Fix an equation $(u,v) \in C$ and assume by contradiction that there exists $x \in X$ such that $u(x) \neq v(x)$. Consider the colouring $\delta_{u(x)} : X \to 2$ given by

$$\delta_{u(x)}(z) = \begin{cases} 1 & \text{if } z = u(x), \\ 0 & \text{if } z \neq u(x). \end{cases}$$

Then $o_{\delta_{u(x)}}(x) \in \mathbf{cofree}(A^*/C)$ since $(X,\alpha) \models_{\mathsf{c}} \mathbf{cofree}(A^*/C)$. Clearly we have that

$$(\star) \qquad\qquad o_{\delta_{u(x)}}(x)(u) = 1 \neq 0 = o_{\delta_{u(x)}}(x)(v).$$

By applying Lemma 13 from [4] we get a colouring $c : A^*/C \to 2$ such that $o_c([\epsilon]) = o_{\delta_{u(x)}}(x)$, and as $[u] = [v]$ we get that $o_c([\epsilon])(u) = o_c([\epsilon])(v)$, which contradicts $(\star)$.

$M_{\mathsf{e}}(C) \subseteq M_{\mathsf{c}}(\mathbf{cofree}(A^*/C))$: Let (X,α) be an automaton such that $(X,\alpha) \models_{\mathsf{e}} C$. We have to show that $(X,\alpha) \models_{\mathsf{c}} \mathbf{cofree}(A^*/C)$. Fix a colouring $c : X \to 2$ and $x \in X$. Define the colouring $\tilde{c} : A^*/C \to 2$ as $\tilde{c}([w]) := c(w(x))$ which is well-defined since $(X,\alpha) \models_{\mathsf{e}} C$. One easily shows that $o_c(x) = o_{\tilde{c}}([\epsilon])$ which is an element of $\mathbf{cofree}(A^*/C)$. □

By using Theorem 7 and the previous lemma we obtain a dual version of Theorem 11.

Theorem 13. *Let K be a nonempty class of automata on A. The following are equivalent:*

i) K is closed under products, subalgebras, homomorphic images, and sums.
ii) $K = M_{\mathsf{c}}(V)$ for some variety of languages V.

Proof. *i)* $\Rightarrow$ *ii)* By Theorem 11, $K = M_{\mathsf{e}}(C)$ for some congruence $C \subseteq A^* \times A^*$ on A^*. Put $V = \mathbf{cofree}(A^*/C)$, then by the previous lemma $M_{\mathsf{e}}(C) = M_{\mathsf{c}}(V)$ where V is a variety of languages by Theorem 7.

ii) $\Rightarrow$ *i)* Assume that $K = M_{\mathsf{c}}(V)$ for some variety of languages V, then one easily shows that K is closed under subalgebras, homomorphic images and sums. By Theorem 7 $V = \mathbf{cofree}(A^*/C)$ for the congruence $C = \mathbf{Eq}(V)$, then by Lemma 12 $M_{\mathsf{c}}(V) = M_{\mathsf{e}}(C)$ which implies that K is closed under products. □

It is worth mentioning that the property that the class $K = M_{\mathsf{c}}(V)$ is closed under products can be proved directly from the fact that V is a variety of languages as follows: Consider a family $\{(X_i, \alpha_i) \mid i \in I\} \subseteq M_{\mathsf{c}}(V)$ and let $X = (\prod_{i \in I} X_i, \alpha)$ be the product of that family. Fix a colouring $c : \prod_{i \in I} X_i \to 2$ and $x \in \prod_{i \in I} X_i$, we want to show that $o_c(x) \in V$, which follows the fact that V is a complete Boolean algebra and from the equality

$$o_c(x) = \bigvee_{y \in c^{-1}(1)} \left(\bigwedge_{i \in I} o_{\delta_{y(i)}}(x(i)) \right)$$

where $\delta_{y(i)} : X_i \to 2$. In fact,

$$\begin{aligned} w \in o_c(x) &\Leftrightarrow \exists y \in c^{-1}(1) \;\; w(x) = y \\ &\Leftrightarrow \exists y \in c^{-1}(1) \forall i \in I \;\; w(x(i)) = y(i) \\ &\Leftrightarrow \exists y \in c^{-1}(1) \forall i \in I \;\; w \in o_{\delta_{y(i)}}(x(i)) \\ &\Leftrightarrow w \in \bigvee_{y \in c^{-1}(1)} \left(\bigwedge_{i \in I} o_{\delta_{y(i)}}(x(i)) \right). \end{aligned}$$

From Lemma 12 we obtain the following corollary.

Corollary 14. *For any automaton (X, α) and any congruence C we have that*

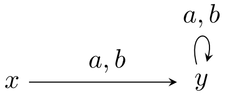
$$(X, \alpha) \models_{\mathsf{e}} C \quad \Leftrightarrow \quad (X, \alpha) \models_{\mathsf{c}} \mathbf{coEq}(A^*/C).$$

Similarly, using the fact that $\mathbf{cofree} \circ \mathbf{free}(V) = V$ for every variety of languages, we get the following corollary.

Corollary 15. *For any automaton (X, α) and any variety of languages V we have that*

$$(X, \alpha) \models_{\mathsf{c}} V \quad \Leftrightarrow \quad (X, \alpha) \models_{\mathsf{e}} \mathbf{Eq}(V).$$

Example 16. Let $A = \{a, b\}$, and consider the regular variety V_2 generated by the automaton (X, α) on A given by



which by Theorem 11 and Theorem 13 can be described in three diferent ways, namely:

i) As the closure under products, subautomata, homomorphic images, and sums of the set $\{(X, \alpha)\}$, which in this case implies that an automaton $(Y, \beta) \in V_2$ if and only if (Y, β) is the sum of elements in V_1 (see Example 10).
ii) $V_2 = M_{\mathsf{e}}(C)$ where C is the congruence generated by $\{a = b, aa = a\}$.
iii) $V_2 = M_{\mathsf{c}}(V)$ where V is the variety of languages $V = \{\emptyset, \{\epsilon\}, A^+, A^*\}$ where $A^+ = A^* \smallsetminus \{\epsilon\}$.

□

We can summarize the results of this section in one theorem as follows.

Theorem 17. *Let K be a nonempty class of automata on A. The following are equivalent:*

i) K is a regular variety, that is $K = M_{\mathsf{e}}(R)$ where R is a set of regular identities, which can be taken to be a congruence on A^*.
ii) K is closed under products, subalgebras, homomorphic images, and sums.
iii) $K = M_{\mathsf{c}}(V)$ for some variety of languages $V \subseteq 2^{A^*}$.

5 Conclusion

Algebras and coalgebras are in general different structures. Deterministic automata have the advantage that they can be defined both as algebras and coalgebras. This not only gives us the advantage of using all the machinery available in those areas but also gives us the possibility to understand and connect unrelated areas and, in some cases, create new results. The results of the present paper are an example of that. (Other examples can be found, for instance, in [4] and [6]).

Homomorphic images and substructures are characterizing properties common to both varieties and covarieties, but varieties are closed under products and tipically not under sums [5], while, dually, covarieties are closed under sums and, in general, not under products [8]. The fact that deterministic automata can be seen as both algebras and coalgebras allowed us to define classes of automata closed under *all* those four constructions. In the present paper, such classes were characterized both equationally and coequationally, and surprisingly, they turned out to be the same as regular varieties of automata, which were studied and characterized by Taylor [13].

As future work we intend to investigate similar results for other structures that can be viewed both as algebras and coalgebras at the same time, such as weighted automata and tree automata.

Acknowledgements

The research of Julian Salamanca is funded by NWO project 612.001.210. The research of Adolfo Ballester-Bolinches has been supported by the grant 11271085 from the *National Natural Science Foundation of China*. The research of Enric Cosme-Llópez has been supported by the predoctoral grant AP2010-2764 from the *Ministerio de Educación* (Spanish Government) and by an internship from *CWI*.

References

1. A. Abel, B. Pientka, D. Thibodeau, A. Setzer. Copatterns: programming infinite structures by observations. In R. Giacobazzi and R. Cousot (eds.), *Proc. of the 40th Annual ACM Symposium on Principles of Programming Languages (POPL '13)*, pp. 27–38, ACM, 2013.
2. J. Adámek, S. Milius, R.S.R. Myers, H. Urbat. Generalized Eilenberg Theorem I: Local Varieties of Languages. To appear in *proceedings of FOSSACS 2015*, Springer 2015.
3. S. Awodey and J. Hughes. Modal operators and the formal dual of Birkhoff's completeness theorem. *Mathematical Structures in Computer Science*, 13(2):233–258, 2003.
4. A. Ballester-Bolinches, E. Cosme-Llópez and J.J.M.M. Rutten. The dual equivalence of equations and coequations for automata. CWI Technical Report report FM-1403, pp. 1–30, 2014.
5. G. Birkhoff. On the structure of abstract algebras. *Proceedings of the Cambridge Philosophical Society* 31: 433454, 1935.
6. 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), 2014.
7. S. N. Burris, and H. P. Sankappanavar. A course in universal algebra. *Graduate Texts in Mathematics* 78, Springer, 1981.
8. R. Clouston and R. Goldblatt. Covarieties of Coalgebras: Comonads and Coequations. In D. Van Hung and M. Wirsing (eds.), *Proc. of 2nd Int. Colloquium on Theoretical Aspects of Computing (ICTAC 2005)* volume 3722 of *Lecture Notes in Computer Science*, pp. 288–302. Springer, 2005.
9. M. Gehrke, S. Grigorieff, J.É. Pin. Duality and equational theory of regular languages. In *Proc. ICALP 2008*, volume 5126 of *Lecture Notes Computer Science*, pp. 246-257. Springer, 2008.
10. E. Graczyńska. Birkhoff's theorems for regular varieties. *Bulletin of the Section of Logic, University of Lódź*, 26(4):210–219, 1997.
11. J. Płonka. On a method of construction of abstract algebras, Fund. Math. LXI, pp. 183–189, 1967.
12. J.J.M.M. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science*, 249(1):3–80, 2000.
13. W. Taylor. Equational logic. *Houston Journal of Mathematics* 5, pp. 1–83, 1979.