VARIETIES AND COVARIETIES OF LANGUAGES

Australian National University

Canberra, 2013

Enric Cosme-Llópez

Departament d'Àlgebra
Universitat de València

CONTENTS

PRELIMINARIES

ALGEBRA-COALGEBRA

Given a category $\mathbf{X}$ and an endofunctor $F : \mathbf{X} \to \mathbf{X}$.

Definition

A F-algebra consists of a pair (X, α), where X is an object of $\mathbf{X}$ and $\alpha : FX \to X$ an arrow in $\mathbf{X}$.

ALGEBRA-COALGEBRA

Given a category $\mathbf{X}$ and an endofunctor $F : \mathbf{X} \to \mathbf{X}$.

Definition

A F-algebra consists of a pair (X, α), where X is an object of $\mathbf{X}$ and $\alpha : FX \to X$ an arrow in $\mathbf{X}$.

Definition

A F-coalgebra consists of a pair (X, α), where X is an object of $\mathbf{X}$ and $\alpha : X \to FX$ an arrow in $\mathbf{X}$.

ALGEBRA-COALGEBRA

Given a category $\mathbf{X}$ and an endofunctor $F : \mathbf{X} \to \mathbf{X}$.

Definition

A F-algebra consists of a pair (X, α), where X is an object of $\mathbf{X}$ and $\alpha : FX \to X$ an arrow in $\mathbf{X}$.

Definition

A F-coalgebra consists of a pair (X, α), where X is an object of $\mathbf{X}$ and $\alpha : X \to FX$ an arrow in $\mathbf{X}$.

We call X the base and α the structure map of the (co)algebra.

AUTOMATA

Definition

Let A be a finite alphabet. An automaton is a pair consisting of a (possibly infinite) set X of states and a transition function

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

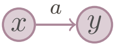
AUTOMATA

Definition

Let A be a finite alphabet. An automaton is a pair 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:



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

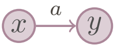
AUTOMATA

Definition

Let A be a finite alphabet. An automaton is a pair 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,

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

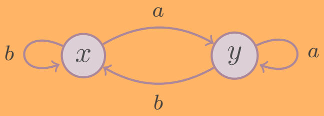
AUTOMATA

On the examples we will use $A = \{a, b\}$.

AUTOMATA

On the examples we will use $A = \{a, b\}$.

Example



AUTOMATA

Because of the isomorphism

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

the transition structure of an automaton X with inputs from an alphabet A can be viewed both as an G-algebra and as a F-coalgebra for the endofunctors on the category $\mathbf{Set}$ given by:

$$\begin{array}{rcl} G(X) & = & X \times A \\ F(X) & = & X^A \end{array}$$

POINTED AUTOMATA

Definition

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

$$x : 1 \to X$$

We call the triple (X, α, x) a pointed automaton

POINTED AUTOMATA

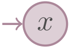
Definition

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

$$x : 1 \to X$$

We call the triple (X, α, x) a pointed automaton

We will depict initial states with an entering arrow:



POINTED AUTOMATA

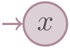
Definition

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

$$x : 1 \to X$$

We call the triple (X, α, x) a pointed automaton

We will depict initial states with an entering arrow:



Note that pointed automata are $(1 + G)$-algebras.

COLOURED AUTOMATA

Definition

An automaton can be decorated by means of a colouring function

$$c : X \to 2$$

We call a state x accepting if $c(x) = 1$ otherwise it is called non-accepting. We call the triple (X, α, c) a coloured automaton

COLOURED AUTOMATA

Definition

An automaton can be decorated by means of a colouring function

$$c : X \to 2$$

We call a state x accepting if $c(x) = 1$ otherwise it is called non-accepting. We call the triple (X, α, c) a coloured automaton

We will depict accepting states with a double mark:

COLOURED AUTOMATA

Definition

An automaton can be decorated by means of a colouring function

$$c : X \to 2$$

We call a state x accepting if $c(x) = 1$ otherwise it is called non-accepting. We call the triple (X, α, c) a coloured automaton

We will depict accepting states with a double mark:

Note that pointed automata are $(2 \times F)$-coalgebras.

THE SCENE

We call a 4-tuple (X, α, x, c) a pointed and coloured automaton.

THE SCENE

We call a 4-tuple (X, α, x, c) a pointed and coloured automaton.

It can be depicted by the diagram:

$$\begin{tikzcd}
1 \arrow[dr, bend left, "x"] & & 2 \\
& X \arrow[ur, bend left, "c"] \arrow[d, "\alpha"'] & \\
& X^A &
\end{tikzcd}$$

THE SCENE

We call a 4-tuple (X, α, x, c) a pointed and coloured automaton.

It can be depicted by the diagram:

$$\begin{tikzcd}
1 \arrow[dr, bend left, "x"] & & 2 \\
& X \arrow[ur, bend left, "c"] \arrow[d, "\alpha"'] & \\
& X^A &
\end{tikzcd}$$

THE SCENE

We call a 4-tuple (X, α, x, c) a pointed and coloured automaton.

It can be depicted by the diagram:

$$\begin{tikzcd}
1 \arrow[dr, bend left, "x"] & & 2 \\
& X \arrow[ur, bend left, "c"] \arrow[d, "\alpha"'] & \\
& X^A &
\end{tikzcd}$$

AUTOMATA HOMOMORPHISMS

Definition

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

$$\begin{tikzcd}
X \arrow[r, "h"] \arrow[d, "\alpha"'] & Y \arrow[d, "\beta"] \\
X^A \arrow[r, "h^A"'] & Y^A
\end{tikzcd}$$

AUTOMATA HOMOMORPHISMS

Definition

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

$$\begin{tikzcd}
X \arrow[r, "h"] \arrow[d, "\alpha"'] & Y \arrow[d, "\beta"] \\
X^A \arrow[r, "h^A"'] & Y^A
\end{tikzcd}$$

A homomorphism of pointed automata and of coloured automata must preserve initial values and colours, respectively.

AUTOMATA HOMOMORPHISMS

Definition

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

$$\begin{tikzcd}
X \arrow[r, "h"] \arrow[d, "\alpha"'] & Y \arrow[d, "\beta"] \\
X^A \arrow[r, "h^A"'] & Y^A
\end{tikzcd}$$

A homomorphism of pointed automata and of coloured automata must preserve initial values and colours, respectively.

If $X \subseteq Y$ and $h : X \hookrightarrow Y$ is the inclusion function, we will say that X is a subautomaton of Y. It will be denoted by $X \leq Y$.

BISIMULATION

Definition

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

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

BISIMULATION

Definition

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

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

For pointed automata (X, α, x) and (Y, β, y), R is a pointed bisimulation if, moreover, $(x, y) \in R$.

BISIMULATION

Definition

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

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

For pointed automata (X, α, x) and (Y, β, y), R is a pointed bisimulation if, moreover, $(x, y) \in R$.

For coloured automata (X, α, c) and (Y, β, d), R is a coloured bisimulation if, moreover,

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

BISIMULATION EQUIVALENCE

Definition

A bisimulation $E \subseteq X \times X$ which is also an equivalence relation is called a bisimulation equivalence.

BISIMULATION EQUIVALENCE

Definition

A bisimulation $E \subseteq X \times X$ which is also an equivalence relation is called a bisimulation equivalence.

The quotient map of a bisimulation equivalence on X is a homomorphism automata:

$$\begin{tikzcd} X \arrow[r, "\pi"] \arrow[d, "\alpha"'] & X/E \arrow[d, "{[\alpha]}"] \\ X^A \arrow[r, "\pi^A"'] & (X/E)^A \end{tikzcd}$$

SETTING THE SCENE

INITIAL ALGEBRA

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

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

INITIAL ALGEBRA

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

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

Proposition

$(A\ , \sigma, \varepsilon)$ is an initial $(1 + G)$-algebra.

INITIAL ALGEBRA

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

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

Proposition

$(A\ ,\sigma,\varepsilon)$ is an initial $(1+G)$-algebra.

For any given automaton (X,α) and every choice of initial state $x : 1 \to X$, it induces a unique function $r_x : A\ \to X$, given by

$$r_x(w) = x_w$$

INITIAL ALGEBRA

This is equivalent to say that the following diagram commutes:

$$\begin{tikzcd}
1 \arrow[d, "\varepsilon"'] \arrow[rd, bend left, "x"] & \\
A \arrow[r, dashed, "r_x"'] \arrow[d, "\sigma"'] & X \arrow[d, "\alpha"'] \\
(A\ )^{A} \arrow[r, dashed, "r_x^A"'] & X^{A}
\end{tikzcd}$$

INITIAL ALGEBRA

This is equivalent to say that the following diagram commutes:

$$\begin{tikzcd}
1 \arrow[d, "\varepsilon"'] \arrow[rd, bend left, "x"] & \\
A \arrow[r, dashed, "r_x"'] \arrow[d, "\sigma"'] & X \arrow[d, "\alpha"'] \\
(A\ \ )^A \arrow[r, dashed, "r_x^A"'] & X^A
\end{tikzcd}$$

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, α, x).

FINAL COALGEBRA

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

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

FINAL COALGEBRA

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

$$\varepsilon? : 2^{A} \to 2 \quad \varepsilon?(L) = \begin{cases} 1 & \text{if } \varepsilon \in L \\ 0 & \text{otherwise} \end{cases}$$

and transition function defined by

$$\tau : 2^{A} \to (2^{A})^{A} \quad \tau(L)(a) = L_a = \{v \in A \mid av \in L\}$$

FINAL COALGEBRA

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

$$\varepsilon? : 2^{A} \to 2 \quad \varepsilon?(L) = \begin{cases} 1 & \text{if } \varepsilon \in L \\ 0 & \text{otherwise} \end{cases}$$

and transition function defined by

$$\tau : 2^{A} \to (2^{A})^{A} \quad \tau(L)(a) = L_a = \{v \in A \mid av \in L\}$$

Proposition

$(2^{A}, \tau, \varepsilon?)$ is a final $(2 \times F)$-coalgebra.

FINAL COALGEBRA

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

$$\varepsilon? : 2^{A} \to 2 \quad \varepsilon?(L) = \begin{cases} 1 & \text{if } \varepsilon \in L \\ 0 & \text{otherwise} \end{cases}$$

and transition function defined by

$$\tau : 2^{A} \to (2^{A})^{A} \quad \tau(L)(a) = L_a = \{v \in A \mid av \in L\}$$

Proposition

$(2^{A}, \tau, \varepsilon?)$ is a final $(2 \times F)$-coalgebra.

For any automaton (X, α) and every choice of colouring function $c : X \to 2$, it induces a unique function $o_c : X \to 2^{A}$, given by

$$o_c(x) = \{w \in A \mid c(x_w) = 1\}$$

FINAL COALGEBRA

This is equivalent to say that the following diagram commutes:

$$\begin{tikzcd}
 & 2 \\
X \arrow[ur, bend left, "c"] \arrow[r, dashed, "o_c"'] \arrow[d, "\alpha"'] & 2^A \arrow[u, "\varepsilon?"'] \arrow[d, "\tau"'] \\
X^A \arrow[r, dashed, "(o_c)^A"'] & (2^A)^A
\end{tikzcd}$$

FINAL COALGEBRA

This is equivalent to say that the following diagram commutes:

$$\begin{tikzcd}
 & 2 \\
X \arrow[ur, bend left, "c"] \arrow[r, dashed, "o_c"'] \arrow[d, "\alpha"'] & 2^A \arrow[u, "\varepsilon?"'] \arrow[d, "\tau"'] \\
X^A \arrow[r, dashed, "(o_c)^A"'] & (2^{A})^A
\end{tikzcd}$$

The function o_c maps a state x to the language $o_c(x)$ accepted by x and is therefore called the observability map for (X, α, c).

THE SCENE

Summarizing:

$$\begin{tikzcd}
1 \arrow[d, "\varepsilon"'] \arrow[rd, bend left, "x"] & & 2 \\
A \arrow[r, dashed, "r_x"'] \arrow[d, "\sigma"'] & X \arrow[r, dashed, "o_c"'] \arrow[ru, bend left, "c"] \arrow[d, "\alpha"'] & 2^A \arrow[u, "\varepsilon?"'] \arrow[d, "\tau"'] \\
(A\ \ )^A \arrow[r, dashed, "r_x^A"'] & X^A \arrow[r, dashed, "(o_c)^A"'] & (2^{A\ \ })^A
\end{tikzcd}$$

THE SCENE

Summarizing:

$$\begin{tikzcd}
1 \arrow[d, "\varepsilon"'] \arrow[dr, bend left, "x"] & & 2 \\
A \arrow[r, dashed, "r_x"'] \arrow[d, "\sigma"'] & X \arrow[r, dashed, "o_c"'] \arrow[d, "\alpha"'] \arrow[ur, bend left, "c"] & 2^A \arrow[u, "\varepsilon?"] \arrow[d, "\tau"'] \\
(A\ \ )^A \arrow[r, dashed, "r_x^A"'] & X^A \arrow[r, dashed, "(o_c)^A"'] & (2^A\ \ )^A
\end{tikzcd}$$

If the reachability map r_x is surjective then we call (X, α, x) reachable.

THE SCENE

Summarizing:

$$\begin{tikzcd}
1 \arrow[d, "\varepsilon"'] \arrow[dr, bend left, "x"] & & 2 \\
A \arrow[r, dashed, "r_x"'] \arrow[d, "\sigma"'] & X \arrow[r, dashed, "o_c"'] \arrow[d, "\alpha"'] \arrow[ur, bend left, "c"] & 2^A \arrow[u, "\varepsilon?"] \arrow[d, "\tau"'] \\
(A\ )^A \arrow[r, dashed, "r_x^A"'] & X^A \arrow[r, dashed, "(o_c)^A"'] & (2^A\ )^A
\end{tikzcd}$$

If the reachability map r_x is surjective then we call (X, α, x) reachable. Note that $(A\ , \sigma, \varepsilon)$ is reachable.

THE SCENE

Summarizing:

$$\begin{tikzcd}
1 \arrow[d, "\varepsilon"'] \arrow[dr, bend left, "x"] & & 2 \\
A \arrow[r, dashed, "r_x"'] \arrow[d, "\sigma"'] & X \arrow[r, dashed, "o_c"'] \arrow[d, "\alpha"'] \arrow[ur, bend left, "c"] & 2^A \arrow[u, "\varepsilon?"] \arrow[d, "\tau"'] \\
(A \ \ )^A \arrow[r, dashed, "r_x^A"'] & X^A \arrow[r, dashed, "(o_c)^A"'] & (2^A \ \ )^A
\end{tikzcd}$$

If the reachability map r_x is surjective then we call (X, α, x) reachable. Note that $(A \ \ , \sigma, \varepsilon)$ is reachable.

If the observability map o_c is injective then we call (X, α, c) observable.

THE SCENE

Summarizing:

$$\begin{tikzcd}
1 \arrow[d, "\varepsilon"'] \arrow[dr, bend left, "x"] & & 2 \\
A \arrow[r, dashed, "r_x"'] \arrow[d, "\sigma"'] & X \arrow[r, dashed, "o_c"'] \arrow[d, "\alpha"'] \arrow[ur, bend left, "c"] & 2^A \arrow[u, "\varepsilon?"] \arrow[d, "\tau"'] \\
(A\ \ )^A \arrow[r, dashed, "r_x^A"'] & X^A \arrow[r, dashed, "(o_c)^A"'] & (2^A\ \ )^A
\end{tikzcd}$$

If the reachability map r_x is surjective then we call (X, α, x) reachable. Note that $(A\ \ , \sigma, \varepsilon)$ is reachable.

If the observability map o_c is injective then we call (X, α, c) observable. Note that $(2^A\ \ , \tau, \varepsilon?)$ is observable.

EQUATIONS AND COEQUATIONS

EQUATIONS

Definition

A set of equations is a bisimulation equivalence $E \subseteq A \times A$ on the initial automaton (A, σ).

EQUATIONS

Definition

A set of equations is a bisimulation equivalence $E \subseteq A \quad \times A \quad$ on the initial automaton $(A \quad , \sigma)$.

Definition

We say that the pointed automaton (X, α, x) satisfies E

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

EQUATIONS

Definition

A set of equations is a bisimulation equivalence $E \subseteq A \quad \times A$ on the initial automaton $(A \quad, \sigma)$.

Definition

We say that the pointed automaton (X, α, x) satisfies E

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

We define:

$$(X, \alpha) \models E \quad \Leftrightarrow \quad \forall x : 1 \to X, \;\; (X, \alpha, x) \models E$$

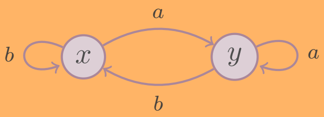
EQUATIONS

Let $v, w \in A$, we consider the shorthand $v = w$ to denote the smallest bisimulation equivalence on A containing (v, w).

EQUATIONS

Let $v, w \in A$, we consider the shorthand $v = w$ to denote the smallest bisimulation equivalence on A containing (v, w).

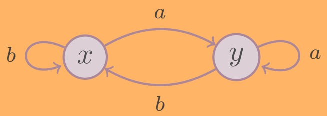
Example



EQUATIONS

Let $v, w \in A$, we consider the shorthand $v = w$ to denote the smallest bisimulation equivalence on A containing (v, w).

Example

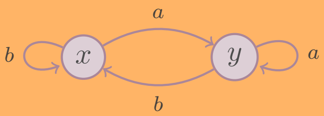


$$(X, \alpha, x) \models \{b = \varepsilon,\ ab = \varepsilon,\ aa = a\}$$

EQUATIONS

Let $v, w \in A$, we consider the shorthand $v = w$ to denote the smallest bisimulation equivalence on A containing (v, w).

Example



$$(X, \alpha, x) \models \{b = \varepsilon,\ ab = \varepsilon,\ aa = a\}$$

$$(X, \alpha, y) \models \{a = \varepsilon,\ ba = \varepsilon,\ bb = b\}$$

EQUATIONS

Proposition

$(X, \alpha, x) \models E \quad \Leftrightarrow \quad E \subseteq \mathsf{ker}(r_x)$

We have, equivalently, that $(X, \alpha, x) \models E$ iff the reachability map r_x factors through $A\ /E$.

EQUATIONS

Proposition

$(X, \alpha, x) \models E \quad \Leftrightarrow \quad E \subseteq \mathsf{ker}(r_x)$

We have, equivalently, that $(X, \alpha, x) \models E$ iff the reachability map r_x factors through $A \ \ /E$.

Definition

We define $\mathsf{Eq}(X, \alpha)$ to be the largest set of equations satisfied by the automaton (X, α).

COEQUATIONS

Definition

A set of coequations is a subautomaton $D \leq 2^{A}$ of the final automaton $(2^{A}, \tau)$.

COEQUATIONS

Definition

A set of coequations is a subautomaton $D \leq 2^{A^*}$ of the final automaton $(2^{A^*}, \tau)$.

Definition

We say that the coloured automaton (X, α, c) satisfies D

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

COEQUATIONS

Definition

A set of coequations is a subautomaton $D \leq 2^{A^*}$ of the final automaton $(2^{A^*}, \tau)$.

Definition

We say that the coloured automaton (X, α, c) satisfies D

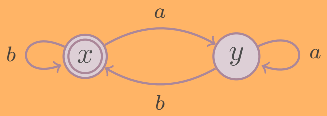
$$(X, \alpha, c) \models D \quad \Leftrightarrow \quad \forall x \in X, \;\; o_c(x) \in D$$

We define:

$$(X, \alpha) \models D \quad \Leftrightarrow \quad \forall c : X \to 2, \;\; (X, \alpha, c) \models D$$

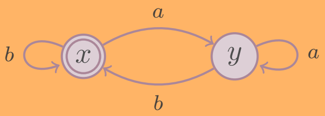
COEQUATIONS

Example



COEQUATIONS

Example

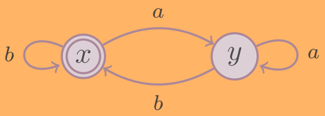


Under the observability map we obtain:

$$o_c(x) = (a\ \ b) \qquad o_c(y) = (a\ \ b)^+$$

COEQUATIONS

Example



Under the observability map we obtain:

$$o_c(x) = (a \ \ b) \qquad\qquad o_c(y) = (a \ \ b)^+$$

therefore,

$$(X, \alpha, c) \models \{(a \ \ b)\ \ , (a \ \ b)^+\}$$

COEQUATIONS

Proposition

$$(X, \alpha, c) \models D \quad \Leftrightarrow \quad \mathrm{im}(o_c) \leq D$$

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

COEQUATIONS

Proposition

$(X, \alpha, c) \models D \quad \Leftrightarrow \quad \text{im}(o_c) \leq D$

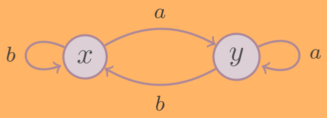
We have, equivalently, that $(X, \alpha, c) \models D$ iff the observability map o_c factors through D.

Definition

We define coEq(X, α) to be the smallest set of coequations satisfied by the automaton (X, α).

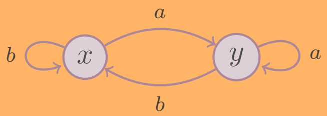
ALL TOGETHER NOW

Example



ALL TOGETHER NOW

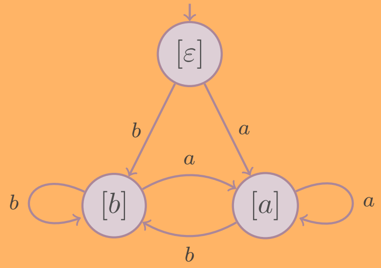
Example



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

ALL TOGETHER NOW

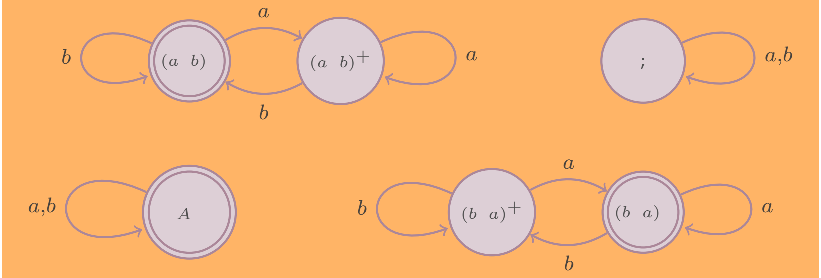
Example



$$A\ \ /\mathsf{Eq}(X,\alpha)$$

ALL TOGETHER NOW

Example



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

VARIETIES AND COVARIETIES

VARIETIES

Definition

For every set E of equations we define the variety V_E by

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

VARIETIES

Definition

For every set E of equations we define the variety V_E by

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

Proposition

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

COVARIETIES

Definition

For every set D of coequations we define the covariety C_D by

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

COVARIETIES

Definition

For every set D of coequations we define the covariety C_D by

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

Proposition

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

LANGUAGES

Definition

Let V_E be a variety. We define the variety of languages $L(V_E)$ by

$$L(V_E) = \{L \in 2^A \ \mid < L > \in V_E\}$$

LANGUAGES

Definition

Let V_E be a variety. We define the variety of languages $L(V_E)$ by

$$L(V_E) = \{L \in 2^A \ \mid < L > \in V_E\}$$

Definition

Let C_D be a covariety. We define the covariety of languages $L(C_D)$ by

$$L(C_D) = \{L \in 2^A \ \mid < L > \in C_D\}$$

ON EQUATIONS AND VARIETIES

Theorem

Let E be a set of equations.

The following statements are equivalent:

ON EQUATIONS AND VARIETIES

Theorem

Let E be a set of equations.

The following statements are equivalent:

i. E is a congruence.

ON EQUATIONS AND VARIETIES

Theorem

Let E be a set of equations.

The following statements are equivalent:

i. E is a congruence.
ii. $E = \mathrm{Eq}(X, \alpha)$ for some automaton (X, α).

ON EQUATIONS AND VARIETIES

Theorem

Let E be a set of equations.

The following statements are equivalent:

i. E is a congruence.

ii. $E = \mathrm{Eq}(X, \alpha)$ for some automaton (X, α).

iii. $(A\ /E, [\sigma]) \models E$.

ON EQUATIONS AND VARIETIES

Theorem

Let E be a set of equations.

The following statements are equivalent:

i. E is a congruence.
ii. $E = \mathsf{Eq}(X, \alpha)$ for some automaton (X, α).
iii. $(A \ /E, [\sigma]) \models E$.
iv. $\mathsf{Eq}(A \ /E, [\sigma]) = E$.

ON EQUATIONS AND VARIETIES

Under any of the statements above, we have:

Corollary

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

ON EQUATIONS AND VARIETIES

Under any of the statements above, we have:

Corollary

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

Note that if E is a bisimulation on A , $A\ /E$ has structure of automaton.

ON EQUATIONS AND VARIETIES

Under any of the statements above, we have:

Corollary

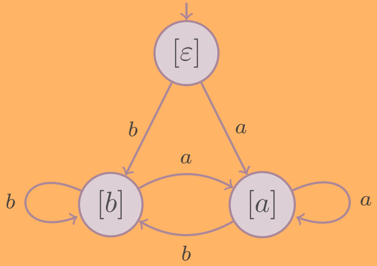
$$L(V_E) = \{L \in 2^{A} \;\; \mid \forall (v,w) \in E,\;\; L_v = L_w\}$$

Note that if E is a bisimulation on A , $A\ /E$ has structure of automaton.

When E is a congruence on A , $A\ /E$ can be both seen as an automaton and as a monoid.

ON EQUATIONS AND VARIETIES

Example



Multiplication law is given by: $[w][v] = [w]_v = [wv]$

ON EQUATIONS AND VARIETIES

Corollary

For every automaton (X, α), A /Eq(X, α) is a monoid.

ON EQUATIONS AND VARIETIES

Corollary

For every automaton (X, α), A /Eq(X, α) is a monoid.

Theorem

$$A\ /\mathrm{Eq}(X, \alpha) \cong \mathrm{trans}(X, \alpha)$$

ON COEQUATIONS AND COVARIETIES

Theorem

Let D be a set of coequations.

The following statements are equivalent:

ON COEQUATIONS AND COVARIETIES

Theorem

Let D be a set of coequations.

The following statements are equivalent:

i. $D = \mathsf{coEq}(X, \alpha)$ for some automaton (X, α).

ON COEQUATIONS AND COVARIETIES

Theorem

Let D be a set of coequations.

The following statements are equivalent:

i. $D = \text{coEq}(X, \alpha)$ for some automaton (X, α).

ii. $(D, \tau) \models D$.

ON COEQUATIONS AND COVARIETIES

Theorem

Let D be a set of coequations.

The following statements are equivalent:

i. $D = \mathrm{coEq}(X, \alpha)$ for some automaton (X, α).

ii. $(D, \tau) \models D$.

iii. $\mathrm{coEq}(D, \tau) = D$.

ON COEQUATIONS AND COVARIETIES

Theorem

Let D be a set of coequations.

The following statements are equivalent:

i. $D = \text{coEq}(X, \alpha)$ for some automaton (X, α).

ii. $(D, \tau) \models D$.

iii. $\text{coEq}(D, \tau) = D$.

iv. $L(C_D) = D$.

BIBLIOGRAPHY

- S. Eilenberg,
  Automata, languages and machines (Vol. A and B),
  Pure and applied mathematics. Academic Press, 1974.

- J. Rutten,
  Universal coalgebra: a theory of systems,
  Elsevier, Theoretical Computer Science, Amsterdam, 2000.

- J. Rutten, A. Ballester-Bolinches, E. Cosme-Llópez
  Varieties and covarieties of languages (preliminary version),
  MFPS XXIX Proceedings, 2013.

- D. Sangiorgi, J. Rutten,
  Advanced Topics in Bisimulation and Coinduction,
  Cambridge Tracts in Theoretical Computer Science, 2012.