Lecture three:

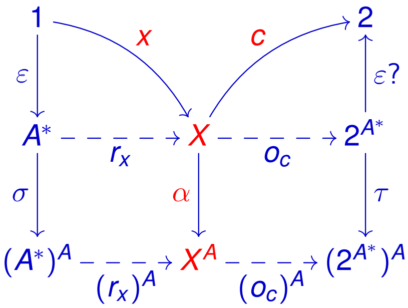
Automata and the algebra-coalgebra duality

Jan Rutten

CWI Amsterdam & Radboud University Nijmegen

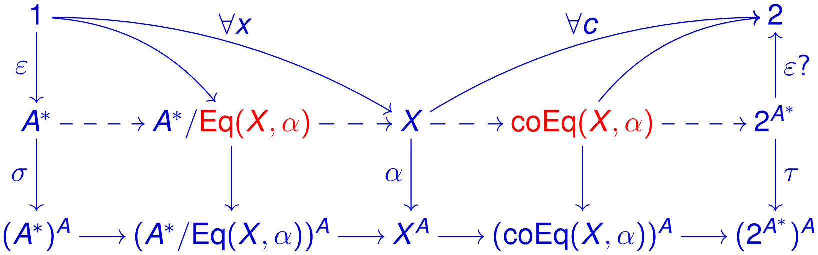
IPM, Tehran - 13 January 2016

This lecture will explain two diagrams:



Algebra-coalgebra duality in Brzozowski's minimization algorithm
Bonchi, Bonsangue, Hansen, Panangaden, Rutten, Silva
ACM Transactions on Computational Logic (TOCL) 2013

This lecture will explain two diagrams:



The dual equivalence of equations and coequations for automata.
A. Ballester-Bolinches, E. Cosme-Llopez, J. Rutten.
Information and Computation Vol. 244, 2015, pp. 49-75.

Motivation

- A modern perspective on a classical subject
- A good illustration of the algebra-coalgebra duality
- Leading to very efficient algorithms (in Lecture four)

Table of contents

1. (Co)algebra - a mini tutorial

Algebras

algebras are pairs (X, α) where:

$$\begin{array}{c} F(X) \\ \Big\downarrow{\scriptstyle \alpha} \\ X \end{array}$$

Coalgebras

coalgebras are pairs (X, α) where:

$$\begin{array}{c} X \\ \alpha \big\downarrow \\ F(X) \end{array}$$

Examples of algebras

$$\begin{array}{c} \mathbb{N} \times \mathbb{N} \\ \Big\downarrow + \\ \mathbb{N} \end{array}$$

$$\begin{array}{c} 1 + \mathbb{N} \\ \Big\downarrow [0, \mathit{successor}] \\ \mathbb{N} \end{array} \quad \equiv \quad \begin{array}{c} 1 \\ \searrow 0 \\ \mathbb{N} \\ \Big\downarrow \mathit{successor} \\ \mathbb{N} \end{array}$$

Examples of algebras

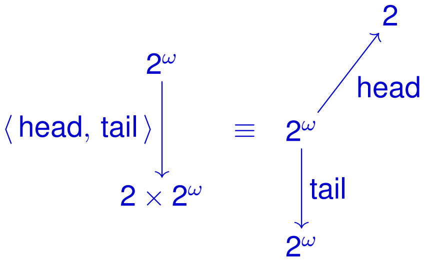
$$\begin{array}{c} \mathbb{N}\times\mathbb{N} \\ {\scriptstyle +}\big\downarrow \\ \mathbb{N} \end{array}$$

$$\begin{array}{c} 1+\mathbb{N} \\ {\scriptstyle [0,\ \mathit{successor}]}\big\downarrow \\ \mathbb{N}\end{array} \quad \equiv \quad \begin{array}{ccc} 1 & & \\ & \searrow^{0} & \\ & & \mathbb{N} \\ & & \big\downarrow{\scriptstyle \mathit{successor}} \\ & & \mathbb{N}\end{array}$$

Examples of coalgebras

$$\begin{array}{c} X \\ \downarrow \alpha \\ \mathcal{P}(A \times X) \end{array} \qquad x \xrightarrow{a} y \quad \Longleftrightarrow \quad \langle a, y \rangle \in \alpha(x)$$

Examples of coalgebras



Thus:

algebras: $$\begin{array}{c} F(X) \\ \alpha \big\downarrow \\ X \end{array}$$

coalgebras: $$\begin{array}{c} X \\ \alpha \big\downarrow \\ F(X) \end{array}$$

Thus:

algebras: $\begin{array}{c} \\ \downarrow \\ X \end{array}$ coalgebras: $\begin{array}{c} X \\ \downarrow \\ \\ \end{array}$

All the rest: by example

- homomorphisms
- bisimulations
- initial algebras, final coalgebras
- induction, coinduction

Table of contents

2. A small exam: algebra or coalgebra?

Initial state

$$1 \xrightarrow{x} X$$

where X is a (possibly infinite) set and

$$1 = \{0\}$$

$$x \in X$$

We will call X **pointed**, with point (or: initial state) x.

Accepting states

$$X \xrightarrow{c} 2$$

where

$$2 = \{0, 1\}$$

We will call c a **colouring**. And:

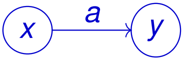
- if $c(x) = 1$ then we call x *accepting*.
- if $c(x) = 0$ then we call x *non-accepting*.

(Deterministic) automaton

$$\begin{array}{c} X \\ \downarrow {\scriptstyle \alpha} \\ X^A \end{array}$$

with

- X is the set of *states*
- A is the *input alphabet*
- $X^A = \{ g \mid g : A \to X \}$
- notation:



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

(Deterministic) automaton

Because

$$X \times A \longrightarrow X \qquad \cong \qquad X \longrightarrow X^A$$

we have:

$$\begin{array}{c} X \times A \\ \downarrow \tilde{\alpha} \\ X \end{array} \qquad \text{and} \qquad \begin{array}{c} X \\ \downarrow \alpha \\ X^A \end{array}$$

It is *both* an algebra and a coalgebra

(Deterministic) automaton

Because

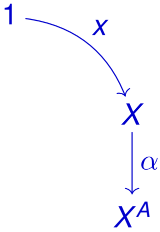
$$X \times A \longrightarrow X \qquad \cong \qquad X \longrightarrow X^A$$

we have:

$$\begin{array}{c} X \times A \\ \downarrow \tilde{\alpha} \\ X \end{array} \qquad \text{and} \qquad \begin{array}{c} X \\ \downarrow \alpha \\ X^A \end{array}$$

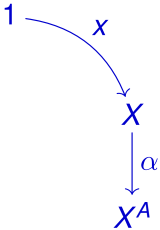
It is *both* an algebra and a coalgebra

A *pointed* automaton (X, x, α)



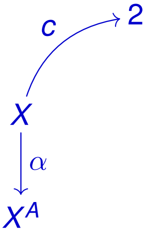
It is an *algebra*, *not* a coalgebra.

A *pointed* automaton (X, x, α)



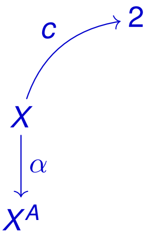
It is an *algebra*, *not* a coalgebra.

A *coloured* automaton (X, c, α)



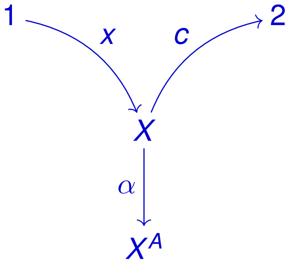
It is a *coalgebra*, *not* an algebra.

A *coloured* automaton (X, c, α)



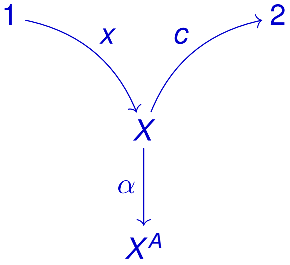
It is a *coalgebra*, *not* an algebra.

A *pointed and coloured* automaton (X, x, c, α)



is *neither* an algebra *nor* a coalgebra.

A *pointed and coloured* automaton (X, x, c, α)



is *neither* an algebra *nor* a coalgebra.

Table of contents

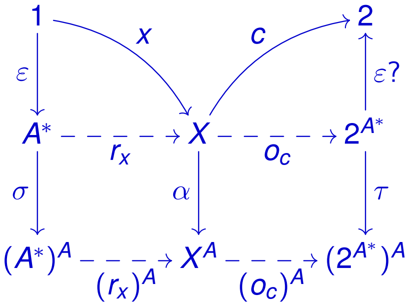
3. The scene:

the algebra-coalgebra duality of automata

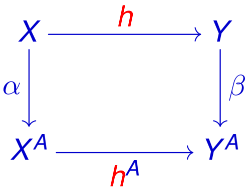
cf. Kalman's duality [1959] controllability - observability

cf. Arbib and Manes categorical approach to automata

The scene: initial algebra *and* final coalgebra

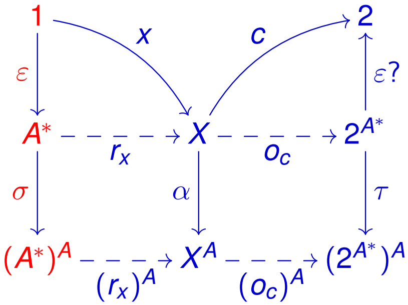


first: homomorphisms of automata



$$\beta(\, h(x)\,)(a) \;=\; h(\,\alpha(x)(a)\,)$$

Initial algebra



The pointed automaton of words

$$1 \xrightarrow{\varepsilon} A^* \xrightarrow{\sigma} (A^*)^A$$

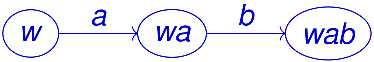
$\varepsilon \quad = \quad$ the empty word as initial state

The pointed automaton of words

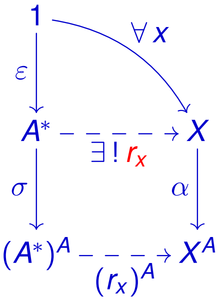
$$\begin{array}{c} 1 \\ \downarrow{\scriptstyle \varepsilon} \\ A^* \\ \downarrow{\scriptstyle \sigma} \\ (A^*)^A \end{array}$$

$\sigma(w)(a) = w \cdot a$

that is, transitions:

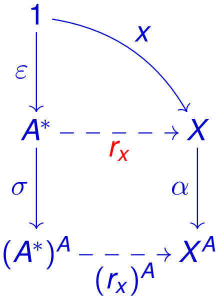


Initial algebra semantics



$r_x(w) = x_w$: the state reached from x on input w

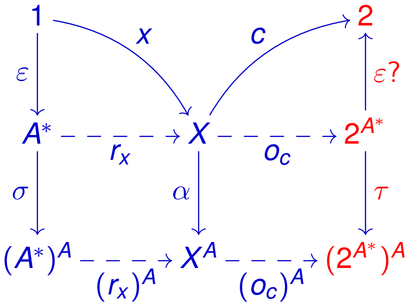
Initial algebra semantics: reachability



r_x $=$ the *reachability* map

if r_x is *surjective* then (X, x, α) is called *reachable*

Final coalgebra



The coloured automaton of languages

$$
\begin{array}{c}
2 \\
\uparrow \varepsilon? \\
2^{A^*} \\
\downarrow \tau \\
(2^{A^*})^A
\end{array}
$$

$$2^{A^*} = \quad \{g \mid g : A^* \to 2\} \quad \cong \quad \{L \mid L \subseteq A^*\}$$

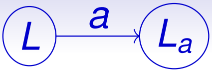
The coloured automaton of languages

$$
\begin{array}{c}
2 \\
\Big\uparrow \varepsilon? \\
2^{A^*} \\
\Big\downarrow \tau \\
(2^{A^*})^A
\end{array}
$$

accepting states: $\varepsilon?(L) = 1 \iff \varepsilon \in L$

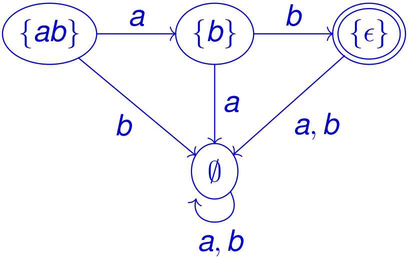
transitions: $\tau(L)(a) \; = \; L_a \; = \; \{w \in A^* \mid a \cdot w \in L\}$

Transitions:



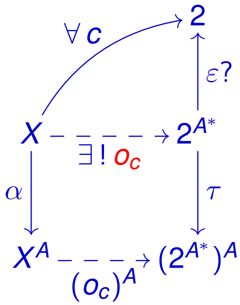
where $L_a = \{w \in A^* \mid a \cdot w \in L\}$.

For instance,



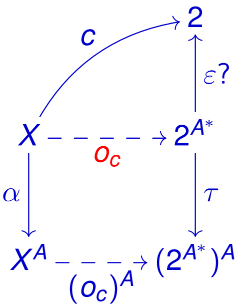
Note that every *state* L accepts . . . the *language* L.

Final coalgebra semantics



$o_c(x) =$ the language accepted by x

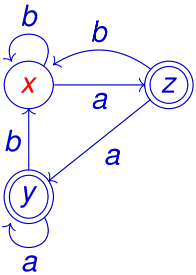
Final coalgebra semantics: observability



o_c = the *observability* map

if o_c is *injective* then (X, c, α) is called *observable*

Example

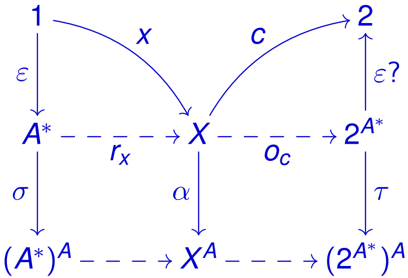


reachable: $y = x_{aa}$ $z = x_a$

not observable: $o_c(y) = o_c(z) = 1 + \{a, b\}^* a$

and so: not minimal

Minimality



minimal $=$ reachable $+$ observable

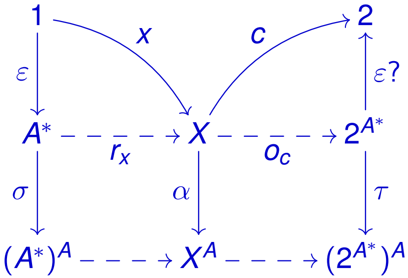
That is, r_x surjective and o_c injective.

Synthesis

Given a language $L \in 2^{A^*}$,   find *minimal*

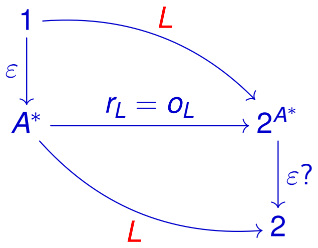
$$(X, x, c, \alpha)$$

accepting L:



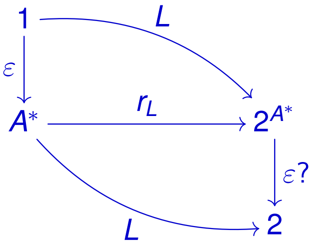
$$o_c(x) = L$$

Synthesis: finding a man in the middle



$$r_L(w) \;=\; o_L(w) \;=\; L_w \;=\; \{v \in A^* \mid w \cdot v \in L\}$$

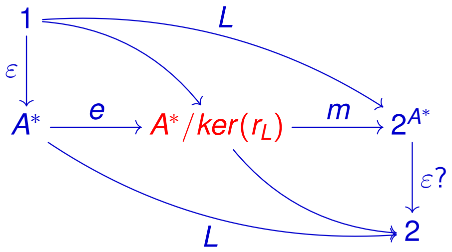
Synthesis: finding a man in the middle



$r_L(v) = r_L(w)$ iff $\forall u \in A^*,\ vu \in L \Leftrightarrow wu \in L$

i.e., $ker(r_L) \;=\;$ *Myhill-Nerode* equivalence

Synthesis by epi-mono factorisation



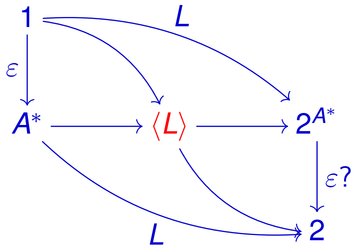
$r_L = m \circ e$

reachable: e is surjective

observable: m is injective

hence: $A^*/ker(r_L) =$ minimal!

Synthesis by epi-mono factorisation



$$A^*/ker(r_L) \;\cong\; \langle L \rangle \;=\; \{\, L_w \mid w \in A^* \,\}$$

Myhill-Nerode meet Brzozowski

Table of contents

4. The duality of reachability and observability

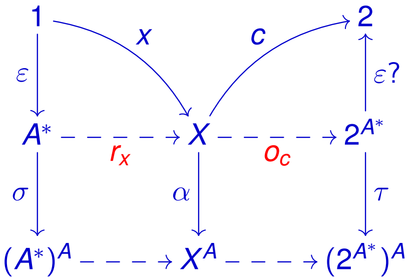
with an application to Brzozowski's minimization algorithm

cf. paper:

Algebra-coalgebra duality in Brzozowski's minimization algorithm
Bonchi, Bonsangue, Hansen, Panangaden, Rutten, Silva
ACM Transactions on Computational Logic (TOCL) 2013

contains various generalisations (Moore, weighted, probabilistic)

Recall: reachability and observability

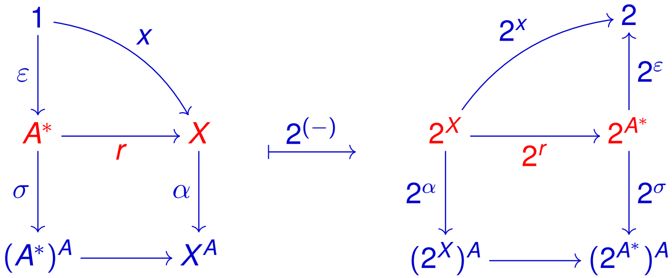


if r_x is *surjective* then (X, x, α) is called *reachable*

if o_c is *injective* then (X, c, α) is called *observable*

minimal $=$ reachable $+$ observable

Reversing automata



$2^{(-)} \;=\;$ contravariant powerset functor

$(2^X, 2^\alpha) \;=\;$ deterministic reverse of (X, α)

Contravariant powerset functor

$$2^{(-)} : \quad \begin{array}{c} V \\ g\big\downarrow \\ W \end{array} \quad \mapsto \quad \begin{array}{c} 2^V \\ \big\uparrow 2^g \\ 2^W \end{array}$$

where

$$2^V = \{S \mid S \subseteq V\} \qquad 2^g(S) = g^{-1}(S)$$

Theorem: g is *surjective* $\Rightarrow$ 2^g is *injective*.

Proof: exercise (use functoriality). □

Contravariant powerset functor

$$2^{(-)}: \quad \begin{array}{c} V \\ g \downarrow \\ W \end{array} \quad \mapsto \quad \begin{array}{c} 2^V \\ \uparrow 2^g \\ 2^W \end{array}$$

where

$$2^V = \{S \mid S \subseteq V\} \qquad 2^g(S) = g^{-1}(S)$$

Theorem: g is *surjective* $\Rightarrow$ 2^g is *injective*.

Proof: exercise (use functoriality). □

Contravariant powerset functor

$$2^{(-)}\ :\quad \begin{array}{c} V \\ g\big\downarrow \\ W \end{array} \quad\mapsto\quad \begin{array}{c} 2^{V} \\ \big\uparrow 2^{g} \\ 2^{W} \end{array}$$

where

$$2^{V} = \{S \mid S \subseteq V\} \qquad 2^{g}(S) = g^{-1}(S)$$

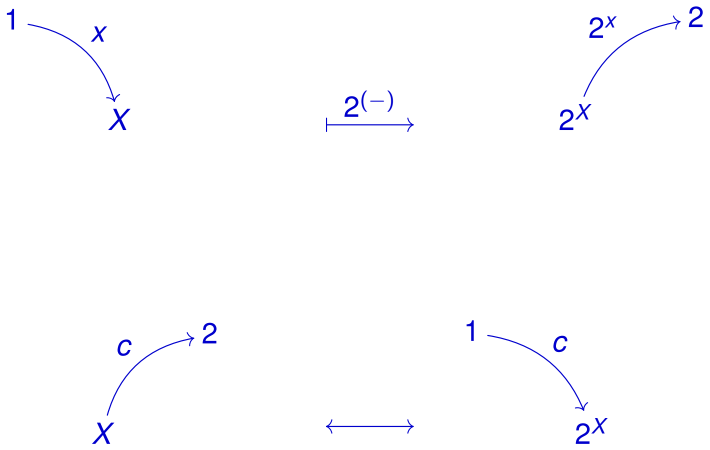
Theorem: g is *surjective* $\Rightarrow$ 2^{g} is *injective*.

Proof: exercise (use functoriality). □

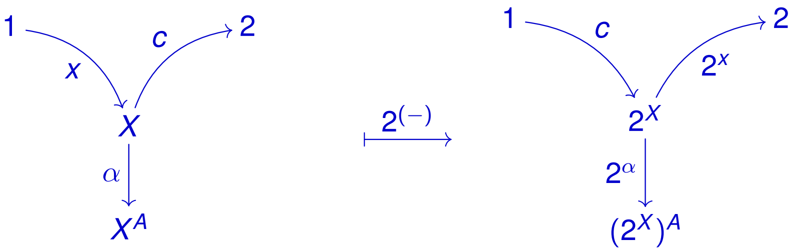
Reversing transitions

$$
\begin{array}{c} X \\ \alpha \downarrow \\ X^A \end{array}
\;\Bigg\|\;
\begin{array}{c} X \times A \\ \downarrow \\ X \end{array}
\quad \xmapsto{\;2^{(-)}\;} \quad
\begin{array}{c} 2^{X \times A} \\ \uparrow \\ 2^X \end{array}
\;\Bigg\|\;
\begin{array}{c} (2^X)^A \\ \uparrow \\ 2^X \end{array}
\;\Bigg\|\;
\begin{array}{c} 2^X \\ \downarrow 2^{\alpha} \\ (2^X)^A \end{array}
$$

point $\iff$ colouring



Reversing the entire automaton



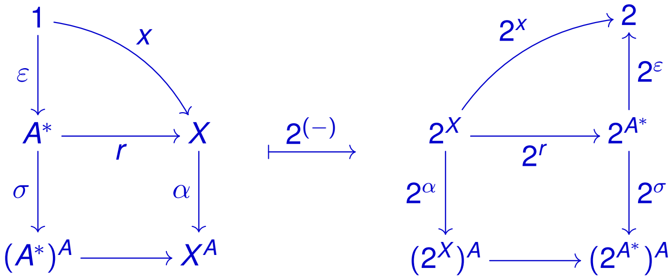
point and colouring are exchanged . . .

transitions are reversed . . .

the result is again deterministic . . .

(X, x, c, α) accepts $L \Rightarrow (2^X, c, 2^x, 2^\alpha)$ accepts L^{rev} !!

Duality between reachability and observablity

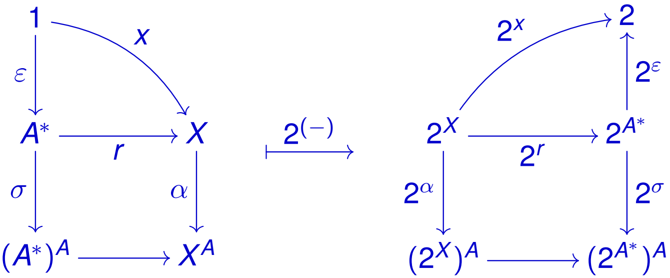


Theorem: r is *surjective* $\Rightarrow$ 2^r is *injective*. □

$\Rightarrow$

Theorem: (X, x, α) is *reachable* $\Rightarrow$ $(2^X, 2^x, 2^\alpha)$ is *observable*. □

Duality between reachability and observablity

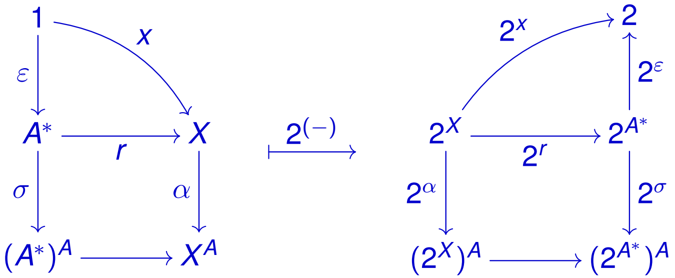


Theorem: r is *surjective* $\Rightarrow$ 2^r is *injective*. □

$\Rightarrow$

Theorem: (X, x, α) is *reachable* $\Rightarrow$ $(2^X, 2^x, 2^\alpha)$ is *observable*. □

Duality between reachability and observablity



Theorem: r is *surjective* $\Rightarrow$ 2^r is *injective*. $\square$

$\Rightarrow$

Theorem: (X, x, α) is *reachable* $\Rightarrow$ $(2^X, 2^x, 2^\alpha)$ is *observable*. $\square$

Corollary: Brzozowski's minimization algorithm

(i) X accepts L

(ii) 2^X accepts L^{rev}

(iii) take reachable part: $Y = reach(2^X)$

(iv) 2^Y accepts $(L^{rev})^{rev} = L$

(v) Y is *reachable* $\Rightarrow$ 2^Y is *observable*

(vi) take reachable part

(vii) result: reachable + observable = minimal automaton for L

Table of contents

5. The coinduction proof method

- is here illustrated: equality of languages

- is used in various theorem provers (COQ, Isabelle, CIRC)

Coinductive proof techniques for language equivalence
J. Rot, M. Bonsangue, J. Rutten
Proceedings LATA 2013, LNCS 7810

Bisimulation relations on automata

$$\begin{array}{c} X \\ \downarrow \alpha \\ X^A \end{array}$$

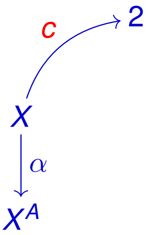
$R \subseteq X \times X$ is a *bisimulation*:

$$\forall (x, y) \in R, \ \forall a \in A : \quad (x_a, \, y_a) \in R$$

where

$$x_a = \alpha(x)(a) \qquad y_a = \alpha(y)(a)$$

. . . on coloured automata



$R \subseteq X \times X$ is a **bisimulation** if, for all $(x, y) \in R$,

$$\forall a \in A : \quad (x_a, y_a) \in R$$

and

$$c(x) = c(y)$$

Bisimulations on languages

$$\begin{array}{c} 2 \\ \uparrow \varepsilon? \\ 2^{A^*} \\ \downarrow \tau \\ (2^{A^*})^A \end{array}$$

$R \subseteq 2^{A^*} \times 2^{A^*}$ is a bisimulation if, for all $(K, L) \in R$,

$$\forall a \in A : \quad (K_a, L_a) \in R$$

and

$$\varepsilon \in K \;\Leftrightarrow\; \varepsilon \in L$$

Bisimulations on languages

$R \subseteq 2^{A^*} \times 2^{A^*}$ is a bisimulation if, for all $(K, L) \in R$,

$$\forall a \in A : \quad (K_a, L_a) \in R$$

and

$$\varepsilon \in K \quad \Leftrightarrow \quad \varepsilon \in L$$

where we recall that

$$K_a = \{ w \mid a \cdot w \in K \}$$

$$L_a = \{ w \mid a \cdot w \in L \}$$

Coinduction proof principle

By the finality of 2^{A^*}, we have:

$$(K, L) \in R, \text{ bisimulation} \quad \Rightarrow \quad K = L$$

Example: Arden's Rule

We will prove *Arden's Rule*:

$$L = KL + M \quad \wedge \quad \varepsilon \notin K \quad \Rightarrow \quad L = K^*M$$

by coinduction.

Arden's Rule: $L = K^* M$?

Assume

$$L = KL + M \quad \wedge \quad \varepsilon \notin K$$

Is $\{ (L, K^* M) \}$ a bisimulation? Well . . .

$$\begin{aligned} L_a &= (KL + M)_a \\ &= K_a L + M_a \\ (K^* M)_a &= K_a K^* M + M_a \end{aligned}$$

. . . almost: it is a *bisimulation-up-to-congruence*.

$\Rightarrow$ $\{(UL + V, UK^* M + V) \mid U, V \in 2^{A^*} \}$ is a bisimulation

$\Rightarrow$ $L = K^* M$, by coinduction!

Exercise: check details in the paper.

Arden's Rule: $L = K^* M$?

Assume

$$L = KL + M \quad \wedge \quad \varepsilon \notin K$$

Is $\{ (L, K^* M) \}$ a bisimulation? Well . . .

$$\begin{aligned} L_a &= (KL + M)_a \\ &= K_a L + M_a \\ (K^* M)_a &= K_a K^* M + M_a \end{aligned}$$

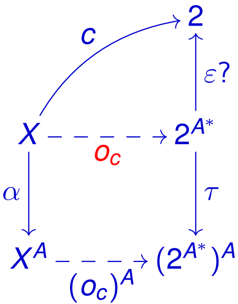
. . . almost: it is a *bisimulation-up-to-congruence*.

$\Rightarrow \quad \{(UL + V, \; UK^* M + V) \mid \; U, V \in 2^{A^*} \}$ is a bisimulation

$\Rightarrow \quad L = K^* M \,,$ by coinduction!

Exercise: check details in the paper.

Behavioural differential equations



An aside: the above diagram can be viewed as a system of behavioural differential equations where the solution is given by finality.

Cf. streams and SDEs.

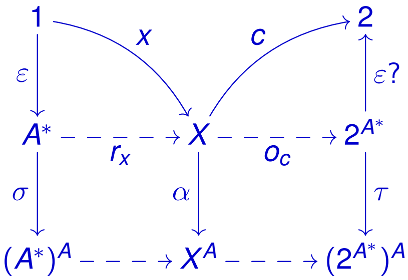
Table of contents

6. Duality between equations and coequations

- defining *classes* of (non-pointed, non-coloured) automata

- words and languages become here tools

Our scene again



Sets of equations: quotients of $(A^*, \varepsilon, \sigma)$

Sets of coequations: subautomata of $(2^{A^*}, \varepsilon?, \tau)$

Equations and satisfaction

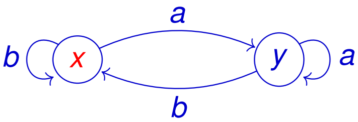
a set of equations $=$ bisimulation equivalence $E \subseteq A^* \times A^*$

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

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

Equations: example

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



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

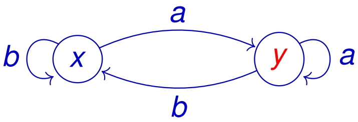
Notation: we use

(i) $v = w$ instead of (v, w)

(ii) shorthand for the induced bisimulation equivalence

Equations: example

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



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

Coequations and satisfaction

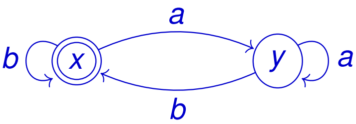
a set of coequations $=$ a subautomaton $D \subseteq 2^{A^*}$

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

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

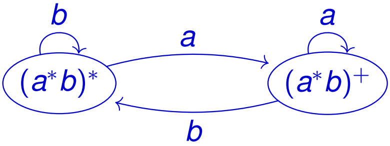
Coequations: example

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



where $c(x) = 1$, $c(y) = 0$.

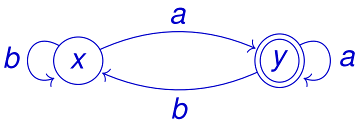
$image(o_c) \quad = $



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

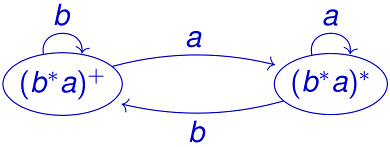
Coequations: example

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



where $d(x) = 0$, $d(y) = 1$.

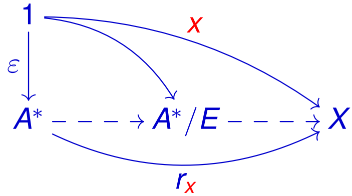
$image(o_d) \quad = $

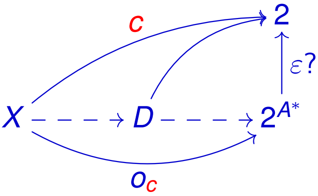


$$(Z, d, \gamma) \models \{(b^*a)^*, (b^*a)^+\}$$

Duality of (co)equations, diagrammatically

$(X, x, \alpha) \models E \quad \Leftrightarrow$

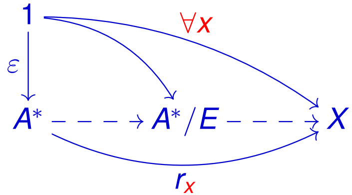


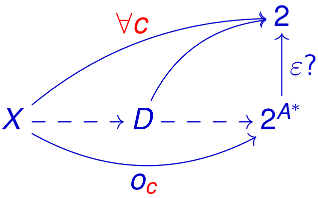


$\Leftrightarrow \quad (X, c, \alpha) \models D$

Duality of (co)equations, diagrammatically

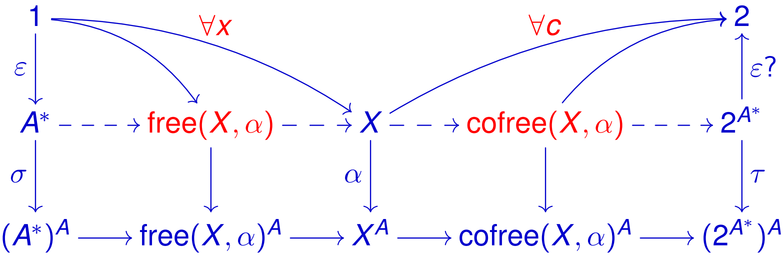
$(X, \alpha) \models E \quad \Leftrightarrow$





$\Leftrightarrow \quad (X, \alpha) \models D$

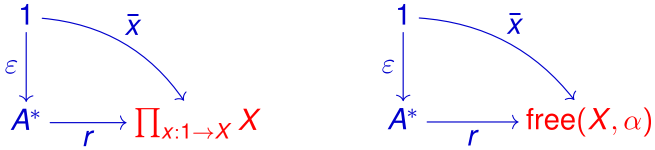
A free and a cofree construction



free(X, α) represents largest set of equations

cofree(X, α) represents smallest set of coequations

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



where we define

$$\mathrm{free}(X,\alpha) \;\equiv\; \mathrm{im}(r) \;\cong\; A^*/\mathrm{Eq}(X,\alpha)$$

with

$$\mathrm{Eq}(X,\alpha) \;\equiv\; \ker(r)$$

$\mathrm{Eq}(X,\alpha)$ = largest set of equations satisfied by (X,α)

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

where we define

$$\mathrm{cofree}(X, \alpha) \;\equiv\; \Sigma X/\ker(o)$$

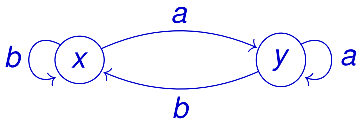
and

$$\mathrm{coEq}(X, \alpha) \;\equiv\; \mathit{image}(o) \;\cong\; \mathrm{cofree}(X, \alpha)$$

$\mathrm{coEq}(X, \alpha) \;=\;$ smallest set of coequations satisfied by (X, α)

Equations: example

$(Z, \gamma) \quad = $



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

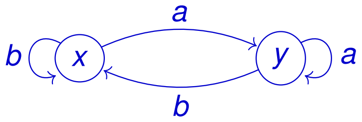
Taking the intersection gives

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

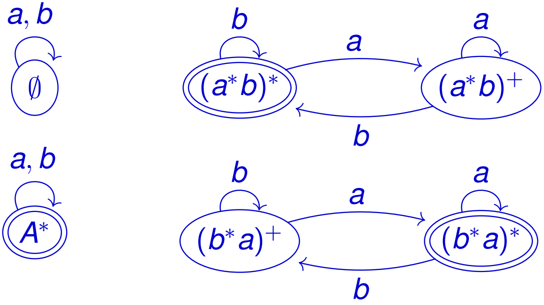
the largest set of equations satisfied by (Z, γ).

Coequations: example

$(Z, \gamma) \quad =$

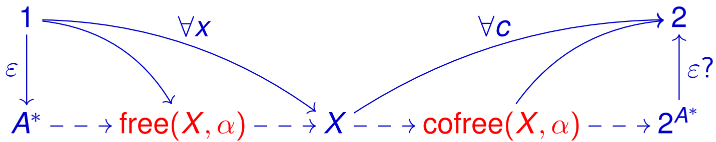


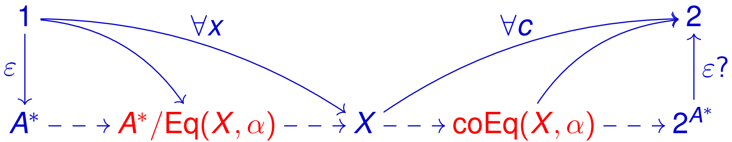
$\text{coEq}(Z, \gamma) \quad =$



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

Summarizing

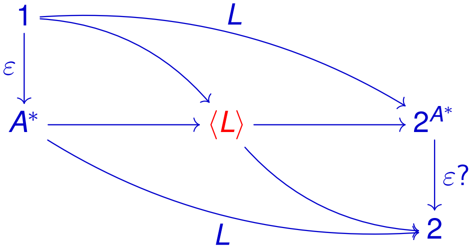




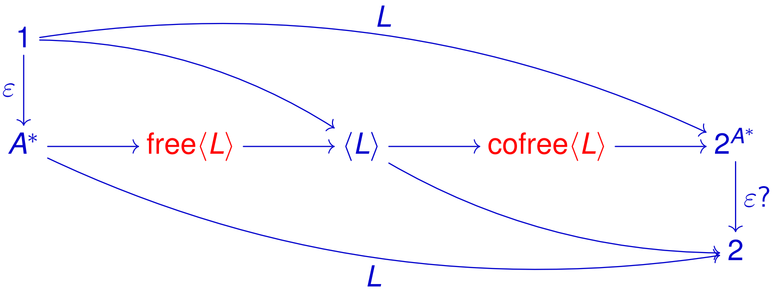
$\mathrm{Eq}(X, \alpha)$ = largest set of equations

$\mathrm{coEq}(X, \alpha)$ = smallest set of coequations

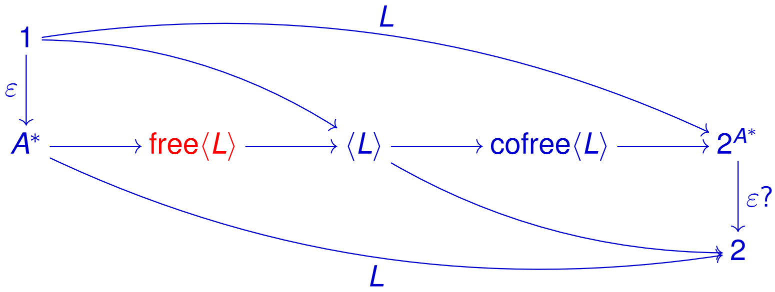
Recall: minimal automaton for a fixed L



Free and cofree of ⟨L⟩



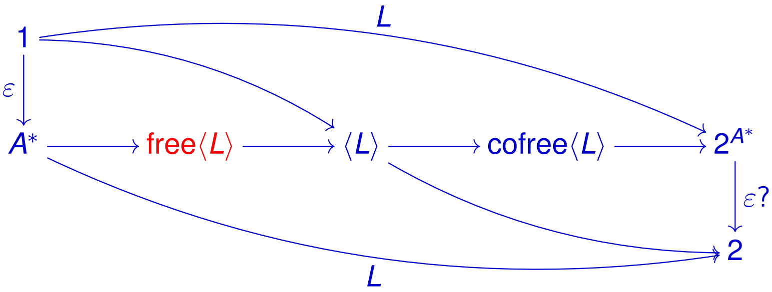
The syntactic monoid of L



$$\text{free}\langle L\rangle \;=\; \text{syn}(L)$$

Cf. algebraic language theory.

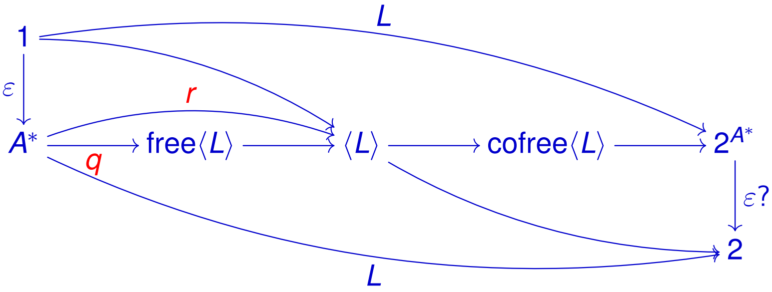
The syntactic monoid of L



$$\text{free}\langle L\rangle \quad = \quad \text{syn}(L)$$

Cf. algebraic language theory.

Theorem



$$
\begin{aligned}
r(v) = r(w) &\Leftrightarrow (v, w) \in \text{Myhill-Nerode congruence} \\
&\Leftrightarrow \forall u \in A^*,\ vu \in L \Leftrightarrow wu \in L \\
q(v) = q(w) &\Leftrightarrow (v, w) \in \text{syntactic congruence} \\
&\Leftrightarrow \forall u_1, u_2 \in A^*,\ u_1 v u_2 \in L \Leftrightarrow u_1 w u_2 \in L
\end{aligned}
$$

Table of contents

7. A dual equivalence

- Between certain classes of equations and coequations.
- It is an initial result about expressiveness.

A dual equivalence

Theorem:

$$\text{cofree}: \quad \mathcal{C} \;\cong\; PL^{op} \quad : \text{free}$$

where $\mathcal{C}$ is the category of all **congruence quotients**

$$A^*/C$$

and PL is the category of all **preformations of languages**:

sets $V \subseteq 2^{A^*}$ such that

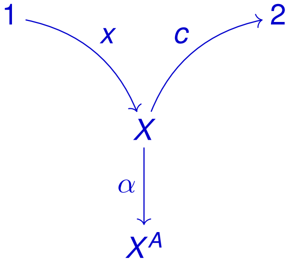
(i) V is a complete atomic Boolean subalgebra of 2^{A^*}

(ii) $\forall L \in 2^{A^*} \quad L \in V \;\Rightarrow\; L_a \in V$ and ${}_aL \in V$

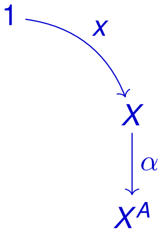
Table of contents

8. In conclusion

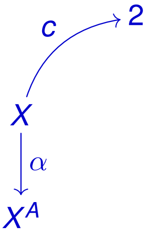
Pointed and coloured automata (X, x, c, α) . . .



. . . are *neither* algebra *nor* coalgebra, but . . .



in part algebra (X, x, α) and . . .



in part coalgebra (X, c, α).

8. In conclusion

The algebra-coalgebra duality of automata leads to

- initial algebra - final coalgebra semantics
- inductive and coinductive proofs
- duality of reachability - observability
- duality of equations - coequations
- duality of varieties - covarieties