Co-Algebraic Implementation of HD-automata

Partitioning Algorithm

Roberto Raggi & Emilio Tuosto

{etuosto,raggi}@di.unipi.it.

Dipartimento di Informatica

Univerità di Pisa

Italy

Plan of the talk

- Finite state verification...

Plan of the talk

- Finite state verification...
  - ...in open systems...

Plan of the talk

- Finite state verification...
  - ...in open systems...
  - ... via Semantic minimization

Plan of the talk

- Finite state verification...
  - ...in open systems...
  - ... via Semantic minimization
- Co-Algebraic Presentation of Transition Systems

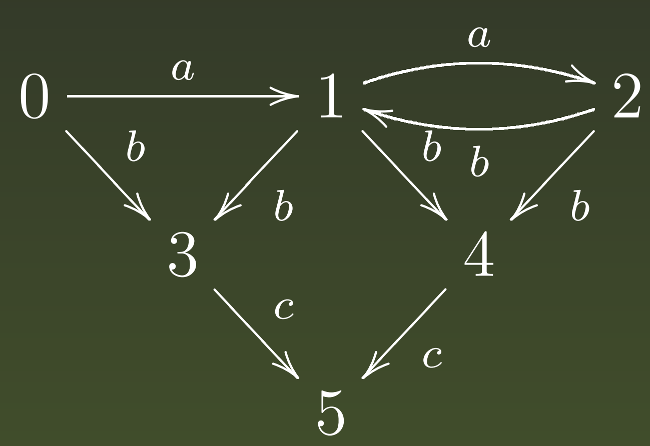
Let's start...

Semantic Minimization

1990 Kannellakis, Smolka (IC 86)

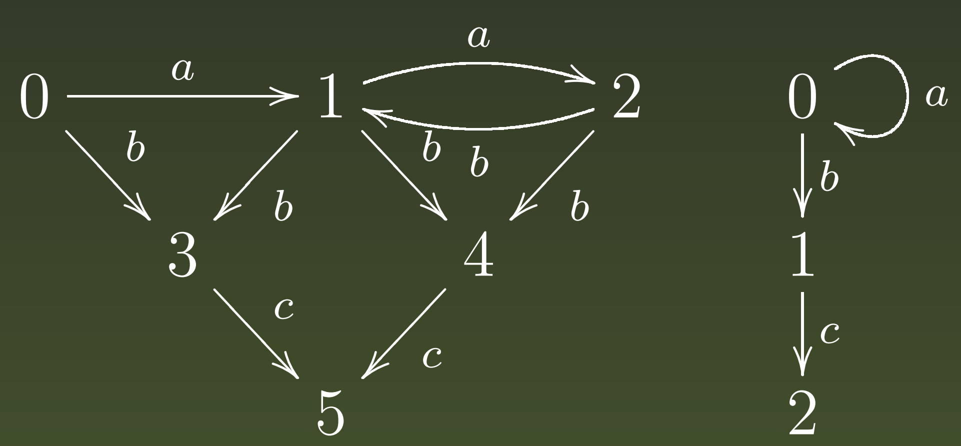
Semantic Minimization

1990 Kannellakis, Smolka (IC 86)



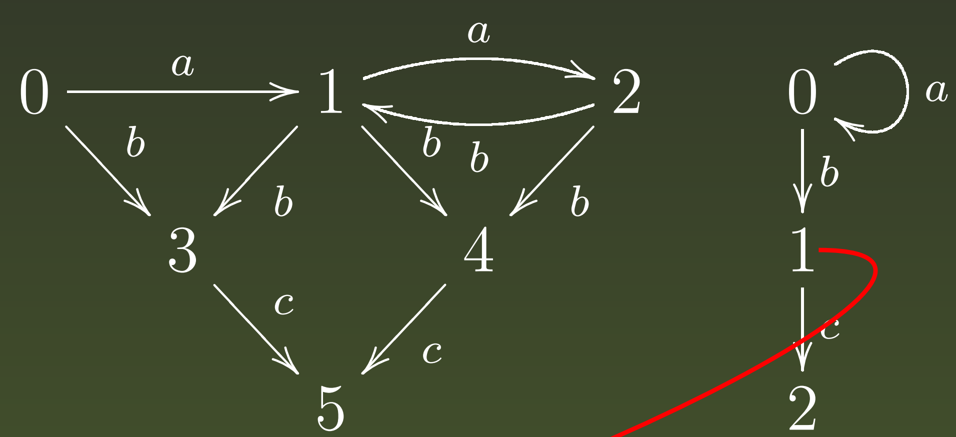
Semantic Minimization

1990 Kannellakis, Smolka (IC 86)



Semantic Minimization

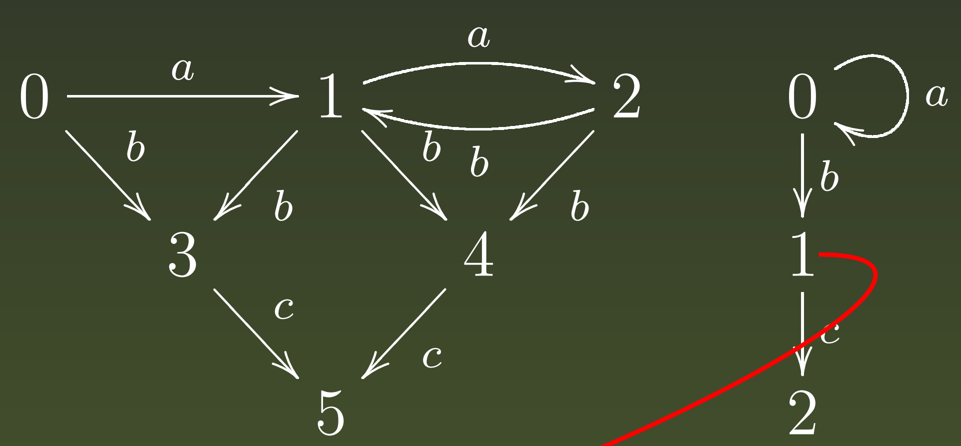
1990 Kannellakis, Smolka (IC 86)



Mimimal automaton more suitable for verifi cation

Semantic Minimization

1990 Kannellakis, Smolka (IC 86)

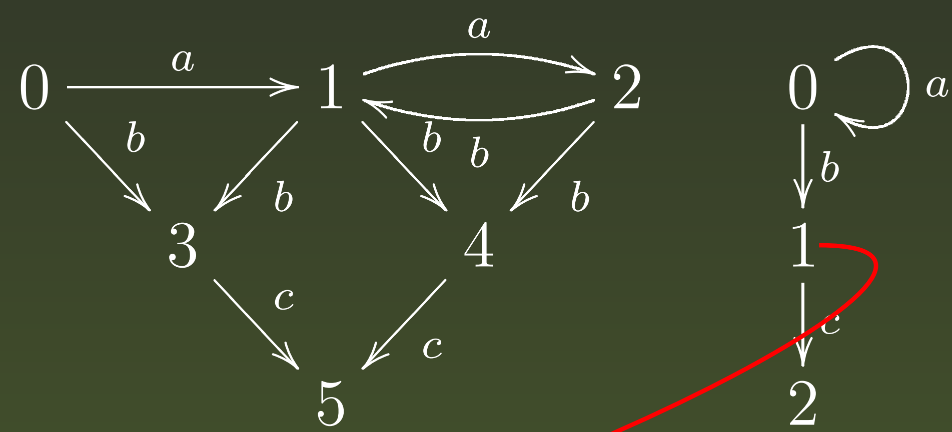


Mimimal automaton more suitable for verifi cation

Side effects:

Semantic Minimization

1990 Kannellakis, Smolka (IC 86)



Mimimal automaton more suitable for verifi cation

Side effects:

- Checking bysimulation for free

$$\mathcal{A} \longrightarrow \mathcal{M}in \longleftarrow \mathcal{B} \qquad \Rightarrow \qquad \mathcal{A} \sim \mathcal{B}$$

- Composition of minimal components

Open Systems

1. www
2. wireless communication
3. Dynamically reconfigurable systems
4. ...

Open Systems

1. www
2. wireless communication
3. Dynamically reconfigurable systems
4. ...

Problem: $A(x) \stackrel{\text{def}}{=} x(y).B(y)$

Open Systems

1. www
2. wireless communication
3. Dynamically reconfigurable systems
4. ...

Problem: $A(x) \stackrel{\text{def}}{=} x(y).B(y)$

$$A(x) \xrightarrow{x\ z} B(z)$$

Open Systems

1. www
2. wireless communication
3. Dynamically reconfigurable systems
4. ...

Problem: $A(x) \stackrel{\mathrm{def}}{=} x(y).B(y)$

$$A(x) \xrightarrow{x\ z} B(z) \quad \boxed{\forall\ z}$$

Open Systems via HD-Automata

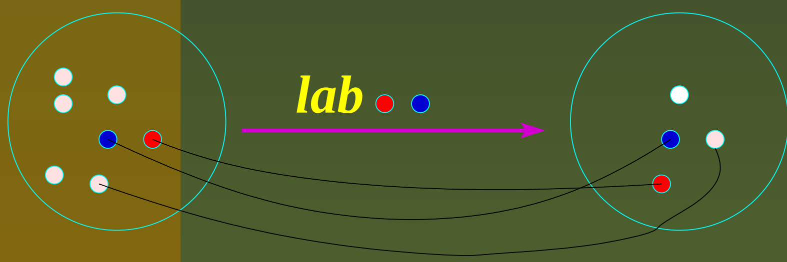
- Open system modeled as HD-Automata

Open Systems via HD-Automata

- Open system modeled as HD-Automata
- Extends classical automata

Open Systems via HD-Automata

- Open system modeled as HD-Automata
- Extends classical automata
- Allocation/Deallocation of resources



- Names are local
- agents identified up-to name permutation
- name creation is well represented

Co-Algebraic Presentation of Automata

Some notations

Set collection of sets

$$q : Q \overset{not}{\equiv} q \in Q \quad (Q : \mathbf{Set})$$

Fun collection of arrows

$$H = \langle S : \mathbf{Set}, D : \mathbf{Set}, h : S \to D \rangle$$

$$S_H \overset{not}{=} S \quad D_H \overset{not}{=} D \quad h_H \overset{not}{=} h$$

$H; K$ denotes composition of $H, K : \mathbf{Fun}$ $(D_H = S_K)$

$$S_{H;K} = S_H$$

$$D_{H;K} = D_K$$

$$h_{H;K} = h_K \circ h_H$$

BasicDefinitions

Transition System

$T = (Q, L, \rightarrow)$

$\rightarrow \subseteq Q \times L \times Q$

Notation

$q \xrightarrow{l} q' \iff (q, l, q') \in \rightarrow$

Basic Definitions

Transition System

$T = (Q, L, \rightarrow)$

$\rightarrow \subseteq Q \times L \times Q$

Notation

$q \xrightarrow{l} q' \iff (q, l, q') \in \rightarrow$

T may be represented as function:

$$\begin{aligned} T &: S \rightarrow \wp(L \times S) \\ T(q) &= \{(l, q') | q \xrightarrow{l} q'\} \end{aligned}$$

Basic Definitions

Transition System

$T = (Q, L, \rightarrow)$

$\rightarrow \subseteq Q \times L \times Q$

Notation

$q \xrightarrow{l} q' \iff (q, l, q') \in \rightarrow$

T may be represented as function:

$$
\begin{aligned}
T &: S \rightarrow \wp(L \times S) \\
T(q) &= \{(l, q') | q \xrightarrow{l} q'\}
\end{aligned}
$$

(step of a) bundle

BasicDefinitions

Transition System

$T = (Q, L, \rightarrow)$

$\rightarrow \subseteq Q \times L \times Q$

Notation

$q \xrightarrow{l} q' \iff (q, l, q') \in \rightarrow$

T may be represented as function:

$$\begin{aligned} T &: S \rightarrow \wp(L \times S) \\ T(q) &= \{(l, q') | q \xrightarrow{l} q'\} \end{aligned}$$

(step of a) bundle

$\beta = \langle d : \textbf{Set}, Step : \wp(L \times D)\rangle$

B collection of bundles

Lifting to functions of bundles...

$T(Q) = \{\beta : B | D_\beta = Q\}$

Let H : **Fun** we define $T(H)$: **Fun** s.t.

- $S_{T(H)} = T(S_H)$
- $D_{T(H)} = T(D_H)$
- $h_{T(H)} : \beta \mapsto \langle D_H, \{\langle l, h_H(q)\rangle | \langle l, q\rangle \in Step_\beta\rangle$

LTS are co-algebras

A LTS on states Q and labels L

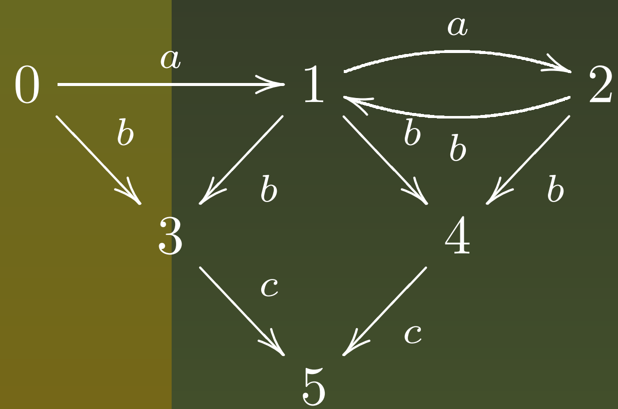
may be expressed as a

co-algebra K : **Fun**

such that

$$S_K = Q \quad D_K = T(Q)$$

Example



$$S_K = \{0, 1, 2, 3, 4, 5\}$$

$$K = \langle S_K, T(S_K), h_K \rangle$$

where

$$\begin{aligned}
h_K(0) &= \langle S_K, \{(a, 1), (b, 3)\} \rangle \\
h_K(1) &= \langle S_K, \{(a, 2), (b, 3), (b, 4)\} \rangle \\
h_K(2) &= \langle S_K, \{(a, 1), (b, 4)\} \rangle \\
h_K(3) &= \langle S_K, \{(c, 5)\} \rangle \\
h_K(4) &= \langle S_K, \{(c, 5)\} \rangle \\
h_K(5) &= \langle S_K, \emptyset \rangle
\end{aligned}$$