Decidability of Identity-free Kleene Lattices

Talk at the LAC meeting

Paul Brunet & Damien Pous
Plume team – LIP, CNRS, ENS de Lyon, Inria, UCBL, Université de Lyon, UMR 5668

November 20^{th}, 2014

Plan

Regular expressions

$$e, f \in \mathcal{R}eg_X ::= \mathbb{0} \mid \mathbb{1} \mid x \in X \mid e \cdot f \mid e \vee f \mid e^\star$$

Interpretations

Regular expressions

$$e, f \in \mathcal{R}eg_X ::= \mathbb{0} \mid \mathbb{1} \mid x \in X \mid e \cdot f \mid e \vee f \mid e^\star$$

Interpretations

- *languages :* Σ a finite set, $\sigma : X \to \mathcal{P}(\Sigma^\star)$,
  $\emptyset$, $\{\epsilon\}$, concatenation, union

Rationnal languages correspond to $[\![\_]\!] : \begin{array}{ccc} X & \to & \mathcal{P}(X^\star) \\ x & \mapsto & \{x\}. \end{array}$

Regular expressions

$$e, f \in \mathcal{R}eg_X ::= \mathbb{0} \mid \mathbb{1} \mid x \in X \mid e \cdot f \mid e \vee f \mid e^\star$$

Interpretations

- *languages :* Σ a finite set, $\sigma : X \to \mathcal{P}(\Sigma^\star)$,
  $\emptyset$, $\{\epsilon\}$, concatenation, union
- *relations :* S an set, $\sigma : X \to \mathcal{P}(S \times S)$,
  $\emptyset$, Id_S, composition, union

Rationnal languages correspond to $[\![\_]\!] : \begin{array}{ccc} X & \to & \mathcal{P}(X^\star) \\ x & \mapsto & \{x\}. \end{array}$

Regular expressions

$$e, f \in \mathcal{R}eg_X ::= \mathbb{0} \mid \mathbb{1} \mid x \in X \mid e \cdot f \mid e \vee f \mid e^\star$$

Interpretations

- *languages :* Σ a finite set, $\sigma : X \to \mathcal{P}(\Sigma^\star)$,
  $\emptyset$, $\{\epsilon\}$, concatenation, union
- ***relations :*** **S an set,** $\sigma : \mathbf{X} \to \mathcal{P}(\mathbf{S} \times \mathbf{S})$**,**
  $\emptyset$**,** $\mathbf{Id}_S$**, composition, union**

Rationnal languages correspond to $[\![\_]\!] : \begin{array}{ccc} X & \to & \mathcal{P}(X^\star) \\ x & \mapsto & \{x\}\,. \end{array}$

Model equivalence

$e, f \in \mathcal{R}eg_X$

$$Rel \models e = f \quad \text{if} \quad \forall S, \forall \sigma : X \to \mathcal{P}\left(S \times S\right), \sigma(e) = \sigma(f)$$

Model equivalence

$e, f \in \mathcal{R}eg_X$

$$Rel \models e = f \quad \text{if} \quad \forall S, \forall \sigma : X \to \mathcal{P}\left(S \times S\right), \sigma(e) = \sigma(f)$$

$$Rel \models e = f \;\Leftrightarrow\; [\![e]\!] = [\![f]\!]$$

Intersection

$$e, f \in \mathcal{R}eg_X^{\wedge} ::= \mathbb{0} \mid \mathbb{1} \mid x \in X \mid e \cdot f \mid {\color{red} e \wedge f} \mid e \vee f \mid e^{\star}$$

Intersection

$$e, f \in \mathcal{R}eg_X^{\wedge} ::= \mathbb{0} \mid \mathbb{1} \mid x \in X \mid e \cdot f \mid e \wedge f \mid e \vee f \mid e^{\star}$$

$$Rel \models e = f \;\not\Leftarrow\; [\![e]\!] = [\![f]\!]$$

Example

$$[\![a \wedge b]\!] = \emptyset = [\![\mathbb{0}]\!]$$

$$\sigma(a) = \{(x, y), (y, z)\}$$
$$\sigma(b) = \{(y, z), (z, t)\}$$
$$\sigma(a \wedge b) = \{(y, z)\} \neq \emptyset = \sigma(\mathbb{0})$$

A different approach is needed.

Plan

1. Introduction
   - Kleene Algebra
   - Kleene Lattices
2. Graph languages
   - Ground terms
   - Regular expressions with intersection
3. Petri Automata
   - Definition
   - Recognition by Petri automata
4. Decision Procedure
5. Conclusions

Graphs/Ground terms

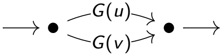
$$u, v \in \mathcal{W}_X ::= \mathbb{0} \mid \mathbb{1} \mid x \in X \mid u \cdot v \mid u \wedge v \mid u \vee v \mid u^\star$$

Graphs/Ground terms

$G(\mathbb{1}) := \quad \longrightarrow \bullet \longrightarrow$

$G(x) := \quad \longrightarrow \bullet \xrightarrow{\;x\;} \bullet \longrightarrow$

$G(u \cdot v) := \quad \longrightarrow \bullet \xrightarrow{G(u)} \bullet \xrightarrow{G(v)} \bullet \longrightarrow$

$G(u \wedge v) :=$ 

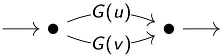
Graphs/Ground terms

$G(\mathbb{1}) := \;\longrightarrow \bullet \longrightarrow$

$G(x) := \;\longrightarrow \bullet \xrightarrow{x} \bullet \longrightarrow$

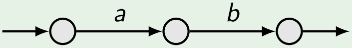
$G(u \cdot v) := \;\longrightarrow \bullet \xrightarrow{G(u)} \bullet \xrightarrow{G(v)} \bullet \longrightarrow$

$G(u \wedge v) :=$



Example

$G(a \cdot b)$:

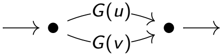


Graphs/Ground terms

$G(\mathbb{1}) := \longrightarrow \bullet \longrightarrow$

$G(x) := \longrightarrow \bullet \xrightarrow{x} \bullet \longrightarrow$

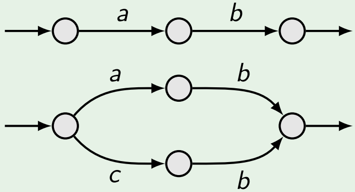
$G(u \cdot v) := \longrightarrow \bullet \xrightarrow{G(u)} \bullet \xrightarrow{G(v)} \bullet \longrightarrow$

$G(u \wedge v) :=$ 

Example

$G(a \cdot b)$:

$G((a \cdot b) \wedge (c \cdot b))$:



Graphs/Ground terms

$G(\mathbb{1}) := \ \longrightarrow \bullet \longrightarrow$

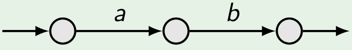
$G(x) := \ \longrightarrow \bullet \xrightarrow{x} \bullet \longrightarrow$

$G(u \cdot v) := \ \longrightarrow \bullet \xrightarrow{G(u)} \bullet \xrightarrow{G(v)} \bullet \longrightarrow$

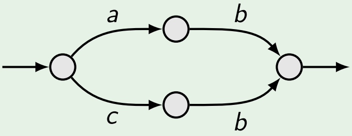
$G(u \wedge v) := \ \longrightarrow \bullet \overset{G(u)}{\underset{G(v)}{\rightrightarrows}} \bullet \longrightarrow$

Example

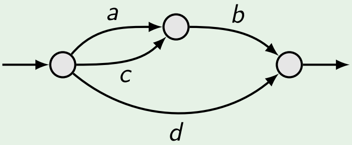
$G(a \cdot b)$:



$G((a \cdot b) \wedge (c \cdot b))$:



$G(((a \wedge c) \cdot b) \wedge d)$:

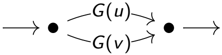


Graphs/Ground terms

$G(\mathbb{1}) := \longrightarrow \bullet \longrightarrow$

$G(x) := \longrightarrow \bullet \xrightarrow{\;x\;} \bullet \longrightarrow$

$G(u \cdot v) := \longrightarrow \bullet \xrightarrow{G(u)} \bullet \xrightarrow{G(v)} \bullet \longrightarrow$

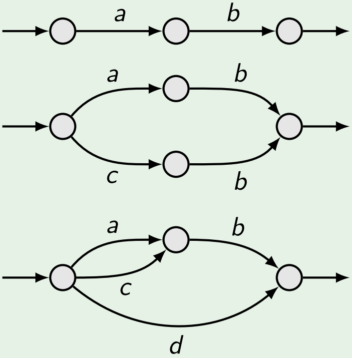
$G(u \wedge v) :=$ 

Example

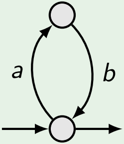
$G(a \cdot b)$:

$G((a \cdot b) \wedge (c \cdot b))$:

$G(((a \wedge c) \cdot b) \wedge d)$:



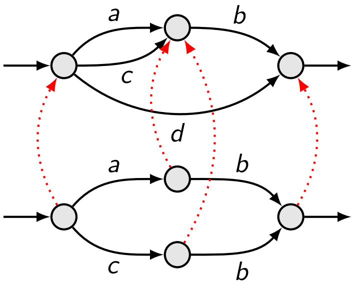
$G((a \cdot b) \wedge \mathbb{1})$:



Preorders

Preorder on graphs

$G \blacktriangleleft G'$ if there exists a graph morphism from G' to G.



$$((a \wedge c) \cdot b) \wedge d$$

$$\vartriangle$$

$$(a \cdot b) \wedge (c \cdot b)$$

Preorder on terms

$u \vartriangleleft v$ if $G(u) \blacktriangleleft G(v)$.

Characterization theorem

Theorem

$u, v \in \mathcal{W}_X$,

$$Rel \models u \leqslant v \Leftrightarrow u \lhd v$$

- P. J. Freyd and A. Scedrov. *Categories, Allegories.* NH, 1990
- H. Andréka and D. Bredikhin.

  The equational theory of union-free algebras of relations.
  Alg. Univ., 33(4) :516–532, 1995

Graphs/Ground terms languages

$$[\![\_]\!] : \mathcal{R}eg_X^{\wedge} \to \mathcal{P}(\mathcal{W}_X)$$

$$\begin{aligned}
[\![\mathbb{0}]\!] &:= \emptyset \\
[\![\mathbb{1}]\!] &:= \{\mathbb{1}\} \\
[\![x]\!] &:= \{x\} \\
[\![e \cdot f]\!] &:= \{w \cdot w' \mid w \in [\![e]\!] \text{ and } w' \in [\![f]\!]\} \\
[\![e \wedge f]\!] &:= \{w \wedge w' \mid w \in [\![e]\!] \text{ and } w' \in [\![f]\!]\} \\
[\![e \vee f]\!] &:= [\![e]\!] \cup [\![f]\!] \\
[\![e^\star]\!] &:= \bigcup_{n \in \mathbb{N}} \{w_1 \cdot \dots \cdot w_n \mid \forall i, w_i \in [\![e]\!]\} \; .
\end{aligned}$$

Graph language of an expression

$e \in \mathcal{R}eg_X^{\wedge}$,

$$G(e) := \{G(w) \mid w \in [\![e]\!]\} .$$

Characterization theorem

$S^\blacktriangleleft := \{G \mid G \blacktriangleleft G', G' \in S\}.$

Theorem

$e, f \in \mathcal{R}eg_X^\wedge$,

$$Rel \models e \leqslant f \Leftrightarrow G(e)^\blacktriangleleft \subseteq G(f)^\blacktriangleleft$$

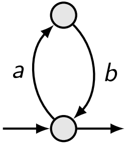
Almost proven in :

H. Andréka, S. Mikulás, and I. Németi. The equational theory of Kleene lattices. *TCS*, 412(52) :7099–7108, 2011

Plan

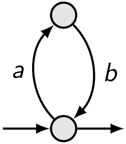
Restriction : identity-free terms

$G\left((a \cdot b) \wedge \mathbb{1}\right)$:



Restriction : identity-free terms

$G((a \cdot b) \wedge \mathbb{1})$:

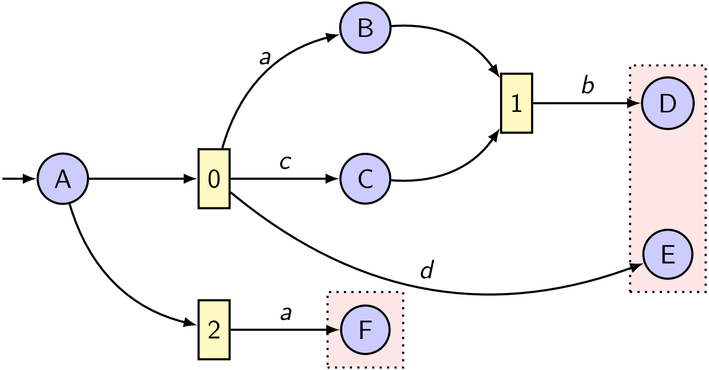


$$u, v \in \mathcal{W}_X^- ::= \mathbb{0} \mid \mathbb{1} \mid x \in X \mid u \cdot v \mid u \wedge v \mid u \vee v \mid u^\star$$

$$e, f \in \mathcal{R}eg_X^{\wedge -} ::= \mathbb{0} \mid \mathbb{1} \mid x \in X \mid e \cdot f \mid e \wedge f \mid e \vee f \mid e^+$$

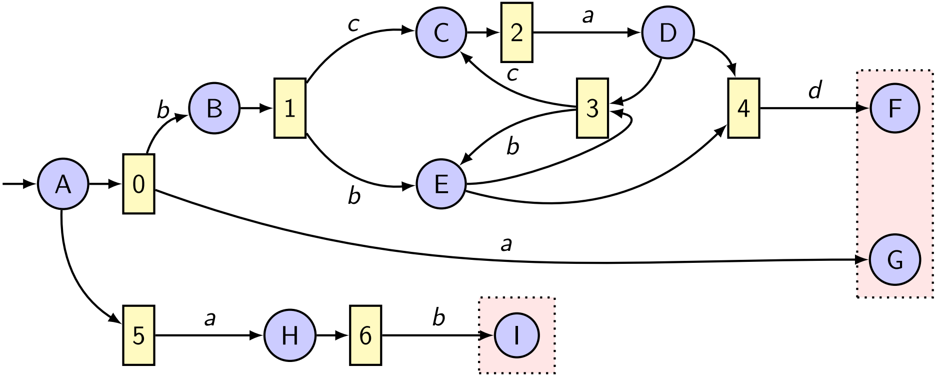
Example

$$(((a \wedge c) \cdot b) \wedge d) \vee a$$

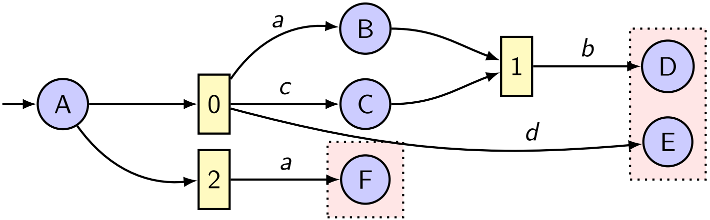


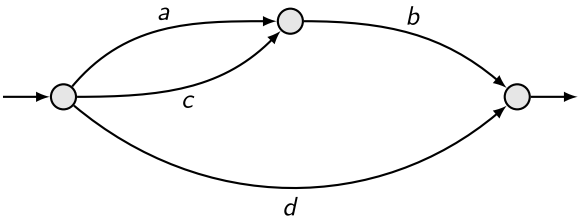
Example

$$\left(b \cdot (a \cdot c \wedge b)^+ \cdot d\right) \wedge a \;\vee\; a \cdot b$$

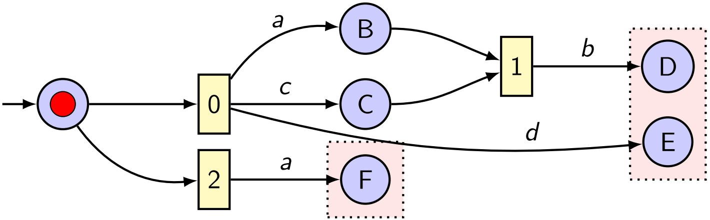


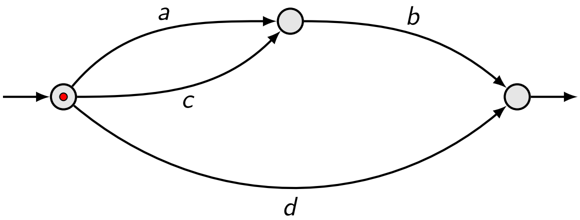
Reading a graph in an automaton



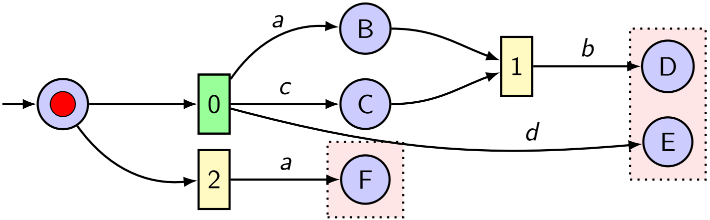


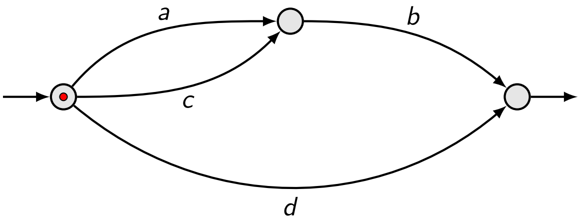
Reading a graph in an automaton



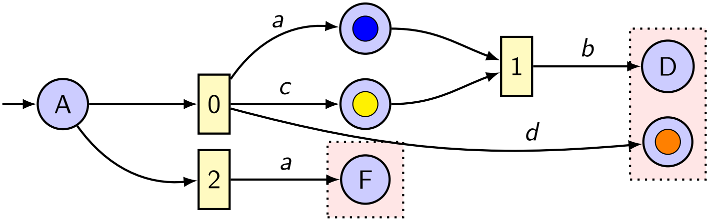


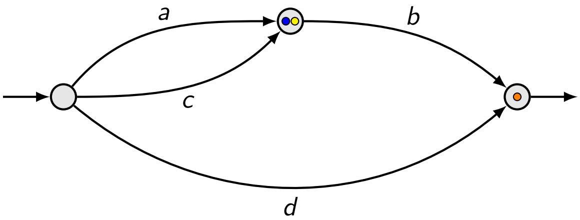
Reading a graph in an automaton



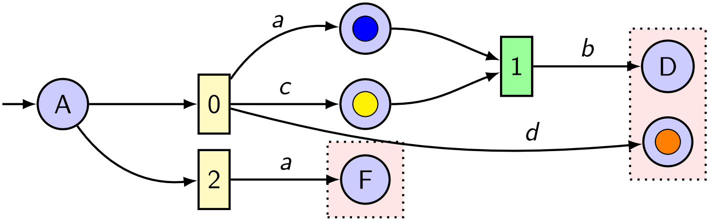


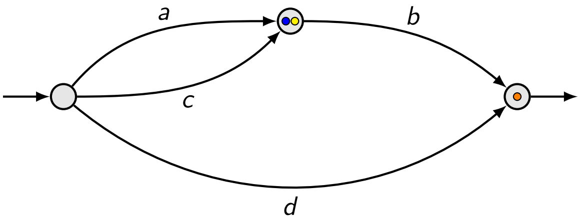
Reading a graph in an automaton



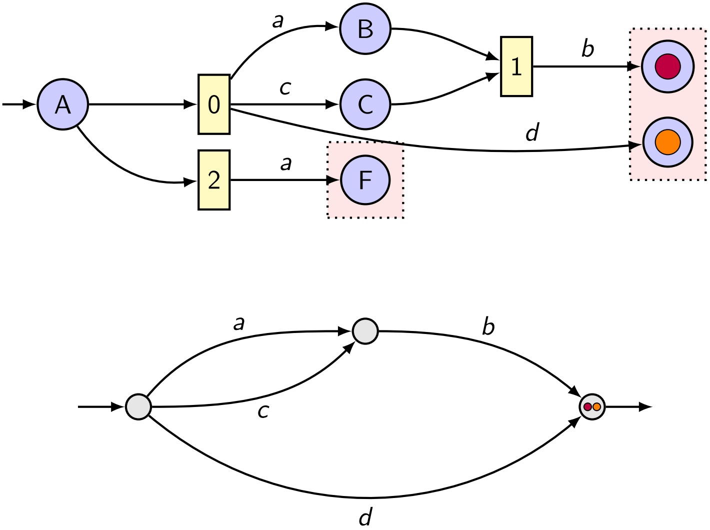


Reading a graph in an automaton



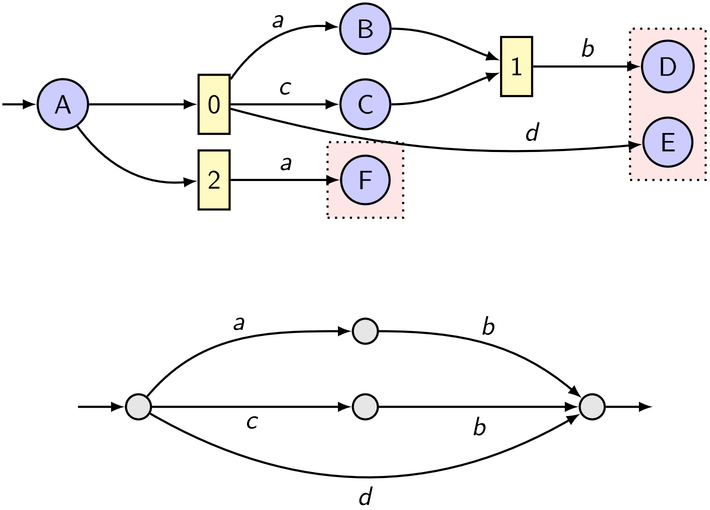


Reading a graph in an automaton

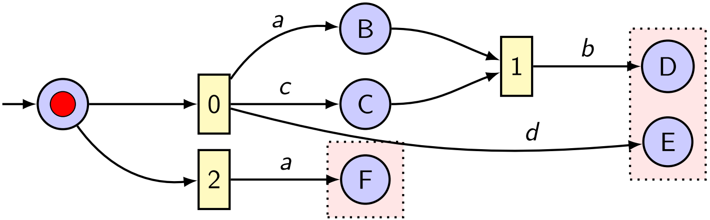


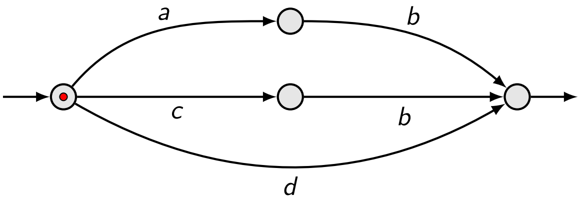
Success !

Reading a graph in an automaton

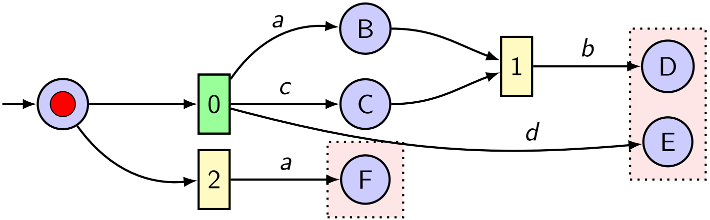


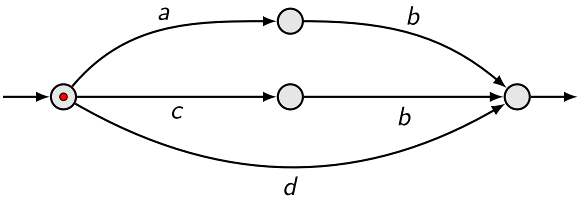
Reading a graph in an automaton



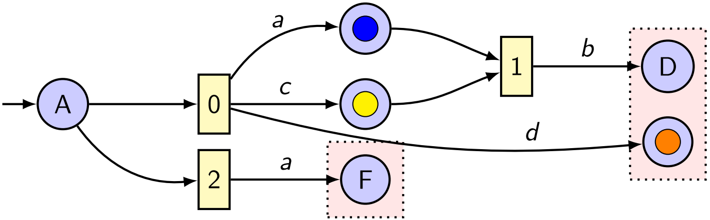


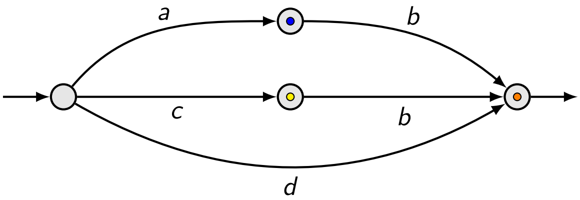
Reading a graph in an automaton



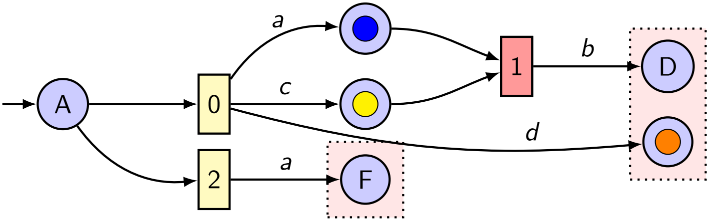


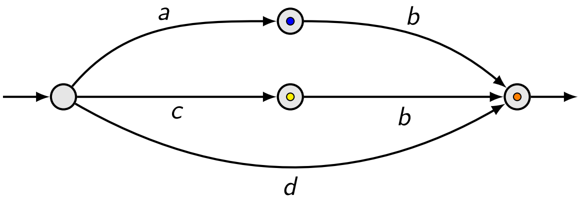
Reading a graph in an automaton





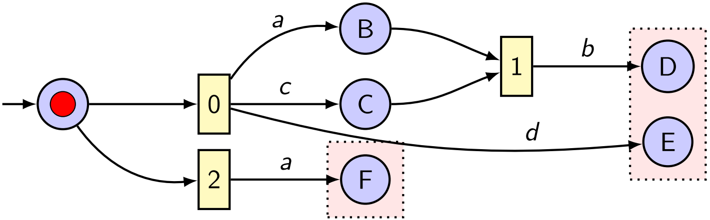
Reading a graph in an automaton

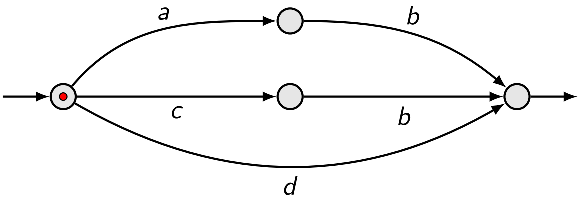




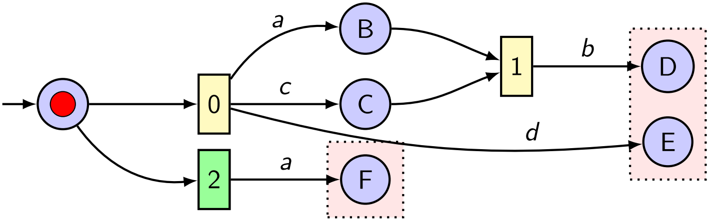
Failure !

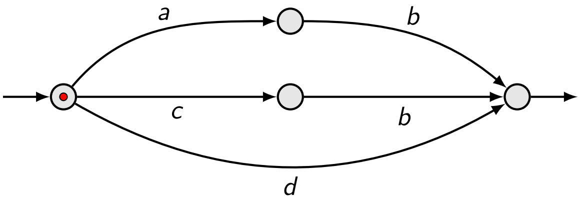
Reading a graph in an automaton



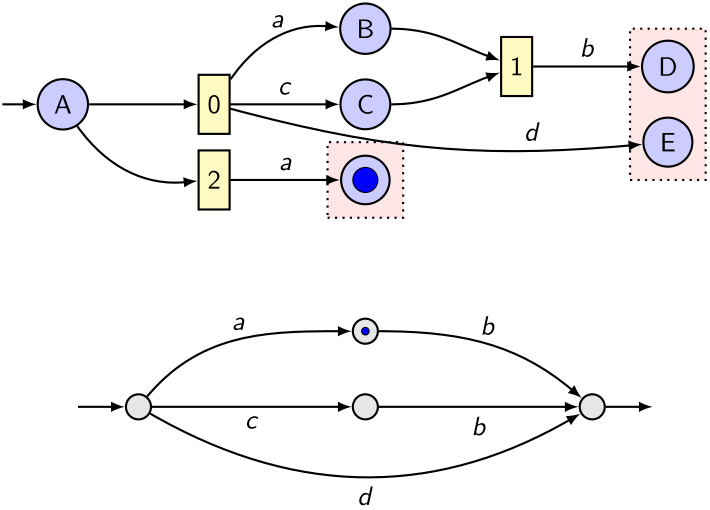


Reading a graph in an automaton

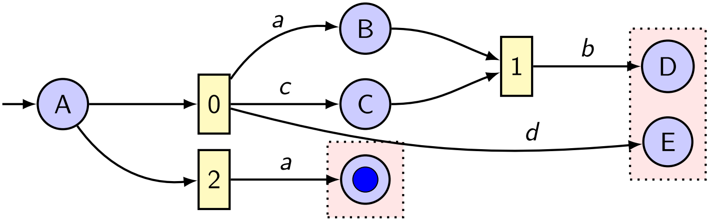


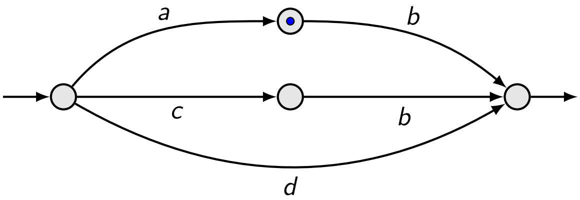


Reading a graph in an automaton



Reading a graph in an automaton





Failure !

Language regognised by an automaton

Correctness

For any $e \in \mathcal{R}eg_X^{\wedge -}$,

$$e$$

Language regognised by an automaton

Correctness

For any $e \in \mathcal{R}eg_X^{\wedge -}$,

$$\mathscr{A}(e)$$

Language regognised by an automaton

Correctness

For any $e \in \mathcal{R}eg_X^{\wedge -}$,

$$\mathscr{L}\left(\mathscr{A}(e)\right)$$

Language regognised by an automaton

Correctness

For any $e \in \mathcal{R}eg_X^{\wedge -}$,

$$\mathscr{L}\left(\mathscr{A}\left(e\right)\right) = G\left(e\right)^{\blacktriangleleft}.$$

Plan

Comparing automata

$$Rel \models e \leqslant f \;\Leftrightarrow\; G(e)^{\blacktriangleleft} \subseteq G(f)^{\blacktriangleleft} \;\Leftrightarrow\; \mathscr{L}(\mathscr{A}(e)) \subseteq \mathscr{L}(\mathscr{A}(f)).$$

Problem :

How to compare two Petri automata ?

Comparing automata

$$Rel \models e \leqslant f \;\Leftrightarrow\; G(e)^{\blacktriangleleft} \subseteq G(f)^{\blacktriangleleft} \;\Leftrightarrow\; \mathscr{L}(\mathscr{A}(e)) \subseteq \mathscr{L}(\mathscr{A}(f)).$$

Problem :

How to compare two Petri automata ?

... not that easily !

Comparing automata

$$Rel \models e \leqslant f \;\Leftrightarrow\; G(e)^{\blacktriangleleft} \subseteq G(f)^{\blacktriangleleft} \;\Leftrightarrow\; \mathscr{L}(\mathscr{A}(e)) \subseteq \mathscr{L}(\mathscr{A}(f)).$$

Problem :

How to compare two Petri automata ?

$\mathscr{L}(\mathscr{A}_1) \subseteq \mathscr{L}(\mathscr{A}_2)$ if and only if there is a simulation relation

$$\leqslant \subseteq \mathcal{P}(P_1) \times \mathcal{P}(P_2 \dashrightarrow P_1)$$

between the configurations of $\mathscr{A}_1$ and the partial maps from the places of $\mathscr{A}_2$ to the places of $\mathscr{A}_1$.

Conclusion and future work

Results

- Reduction of relational equivalence to equality of closed graph languages.

Future work

Conclusion and future work

Results

- Reduction of relational equivalence to equality of closed graph languages.
- Representation of closed graph languages through Petri automata.

Future work

Conclusion and future work

Results

- Reduction of relational equivalence to equality of closed graph languages.
- Representation of closed graph languages through Petri automata.
- Decidability of automata equivalence, thus of relational equivalence.

Future work

Conclusion and future work

Results

- Reduction of relational equivalence to equality of closed graph languages.
- Representation of closed graph languages through Petri automata.
- Decidability of automata equivalence, thus of relational equivalence.

This decision procedure was implemented in OCAML, and is available as an online application.

Future work

Conclusion and future work

Results

- Reduction of relational equivalence to equality of closed graph languages.
- Representation of closed graph languages through Petri automata.
- Decidability of automata equivalence, thus of relational equivalence.

This decision procedure was implemented in OCaml, and is available as an online application.

Future work

- Decidability with $\mathbb{1}$.

Conclusion and future work

Results

- Reduction of relational equivalence to equality of closed graph languages.
- Representation of closed graph languages through Petri automata.
- Decidability of automata equivalence, thus of relational equivalence.

This decision procedure was implemented in OCaml, and is available as an online application.

Future work

- Decidability with $\mathbb{1}$.
- Completeness.

Conclusion and future work

Results

- Reduction of relational equivalence to equality of closed graph languages.
- Representation of closed graph languages through Petri automata.
- Decidability of automata equivalence, thus of relational equivalence.

This decision procedure was implemented in OCaml, and is available as an online application.

Future work

- Decidability with $\mathbb{1}$.
- Completeness.
- Extension with converse.

That's it !

Thank you !

The slides of this talk will be available online shortly on my webpage :

`http://perso.ens-lyon.fr/paul.brunet/rklm`.

The content presented here has been accepted for publication in JFLA 2015.

Plan