Petri automata for Kleene Allegories

Talk at the Rapido meeting

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

June 19, 2015

Motivation : Relation Algebra

$$(R \cap S) \circ T \subseteq (R \circ T) \cap (S \circ T)$$

Motivation : Relation Algebra

$$(R \cap S) \circ T \subseteq (R \circ T) \cap (S \circ T)$$

Let $(i, j) \in (R \cap S) \circ T$,

Motivation : Relation Algebra

$$(R \cap S) \circ T \subseteq (R \circ T) \cap (S \circ T)$$

Let $(i, j) \in (R \cap S) \circ T$,

- there is k such that $\begin{cases} (i, k) \in (R \cap S) \\ (k, j) \in T \end{cases}$

Motivation : Relation Algebra

$$(R \cap S) \circ T \subseteq (R \circ T) \cap (S \circ T)$$

Let $(i,j) \in (R \cap S) \circ T$,

- there is k such that $\begin{cases} (i,k) \in (R \cap S) \\ (k,j) \in T \end{cases}$
- thus $\begin{cases} (i,k) \in R \\ (i,k) \in S \\ (k,j) \in T \end{cases}$

Motivation : Relation Algebra

$$(R \cap S) \circ T \subseteq (R \circ T) \cap (S \circ T)$$

Let $(i,j) \in (R \cap S) \circ T$,

- there is k such that $\begin{cases} (i,k) \in (R \cap S) \\ (k,j) \in T \end{cases}$
- thus $\begin{cases} (i,k) \in R \\ (i,k) \in S \\ (k,j) \in T \end{cases}$
- thus $\begin{cases} (i,j) \in R \circ T \\ (i,j) \in S \circ T \end{cases}$

Motivation : Relation Algebra

$$(R \cap S) \circ T \subseteq (R \circ T) \cap (S \circ T)$$

Let $(i, j) \in (R \cap S) \circ T$,

- there is k such that $\begin{cases} (i, k) \in (R \cap S) \\ (k, j) \in T \end{cases}$
- thus $\begin{cases} (i, k) \in R \\ (i, k) \in S \\ (k, j) \in T \end{cases}$
- thus $\begin{cases} (i, j) \in R \circ T \\ (i, j) \in S \circ T \end{cases}$
- hence $(i, j) \in (R \circ T) \cap (S \circ T)$.

Motivation : Relation Algebra

$$(R \cap S) \circ T \subseteq (R \circ T) \cap (S \circ T)$$

Let $(i,j) \in (R \cap S) \circ T$,

- there is k such that $\begin{cases} (i,k) \in (R \cap S) \\ (k,j) \in T \end{cases}$
- thus $\begin{cases} (i,k) \in R \\ (i,k) \in S \\ (k,j) \in T \end{cases}$
- thus $\begin{cases} (i,j) \in R \circ T \\ (i,j) \in S \circ T \end{cases}$
- hence $(i,j) \in (R \circ T) \cap (S \circ T)$.

Simple and boring : could it be done automatically ?

Outline

Regular expressions

$$e, f \in \mathrm{Reg}_X ::= 0 \mid 1 \mid x \in X \mid e \cdot f \mid e \cup f \mid e^\star$$

Interpretations

Regular expressions

$$e, f \in \mathrm{Reg}_X ::= 0 \mid 1 \mid x \in X \mid e \cdot f \mid e \cup f \mid e^\star$$

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

$$e, f \in \mathrm{Reg}_X ::= 0 \mid 1 \mid x \in X \mid e \cdot f \mid e \cup f \mid e^\star$$

Interpretations

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

$$e, f \in \mathrm{Reg}_X ::= 0 \mid 1 \mid x \in X \mid e \cdot f \mid e \cup f \mid e^\star$$

Interpretations

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

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

Relational equivalence

$e, f \in \mathsf{Reg}_X$

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

Relational equivalence

$e, f \in \mathrm{Reg}_X$

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

Theorem

$$\mathrm{Rel} \models e = f \Leftrightarrow [\![e]\!] = [\![f]\!]$$

Corollary

Relational equivalence is decidable for regular expressions.

Kleene Allegories

$$e, f \in \mathrm{Reg}_X^{\smile\cap} ::= 0 \mid 1 \mid x \in X \mid e \cdot f \mid e \cap f \mid e \cup f \mid e^\star \mid e^\smile$$

Kleene Allegories

$$e, f \in \mathrm{Reg}_X^{\smile\cap} ::= 0 \mid 1 \mid x \in X \mid e \cdot f \mid e \cap f \mid e \cup f \mid e^\star \mid e^\smile$$

$$\mathrm{Rel} \models e = f \overset{?}{\Leftrightarrow} [\![e]\!] = [\![f]\!]$$

Kleene Allegories

$$e, f \in \mathrm{Reg}_X^{\smile\cap} ::= 0 \mid 1 \mid x \in X \mid e \cdot f \mid e \cap f \mid e \cup f \mid e^\star \mid e^\smile$$

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

Example

$$[\![a \cap b]\!] = \emptyset = [\![0]\!]$$

$$\begin{aligned} \sigma(a) &= \{(x,y),(y,z)\} \\ \sigma(b) &= \{(y,z),(z,t)\} \\ \sigma(a \cap b) &= \{(y,z)\} \neq \emptyset = \sigma(0) \end{aligned}$$

$$[\![a]\!] = \{a\} = [\![a^\smile]\!]$$

$$\begin{aligned} \sigma(a) &= \{(x,y)\} \\ \sigma(a^\smile) &= \{(y,x)\} \neq \sigma(a) \end{aligned}$$

A different approach is needed.

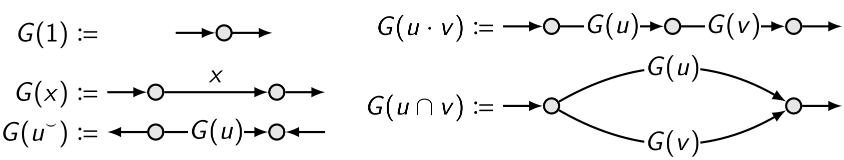
Outline

Outline

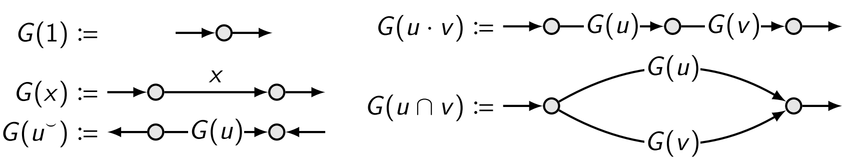
Graphs/Ground terms

$$u, v \in \mathsf{W}_X ::= 0 \mid 1 \mid x \in X \mid u \cdot v \mid u \cap v \mid u \cup v \mid u^\star \mid u^\smile$$

Graphs/Ground terms

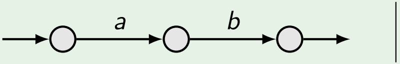


Graphs/Ground terms

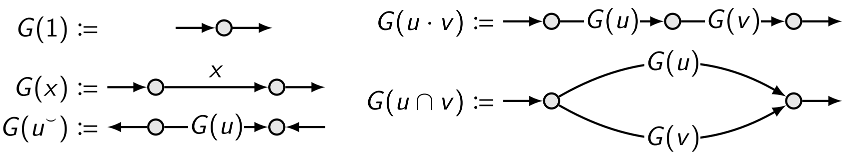


Example

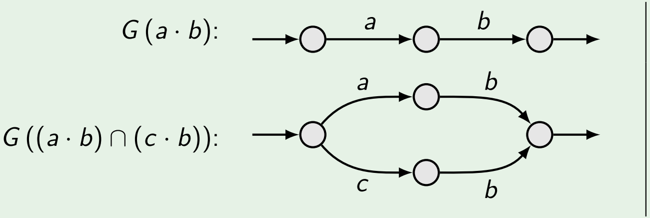
$G\,(a \cdot b)$:



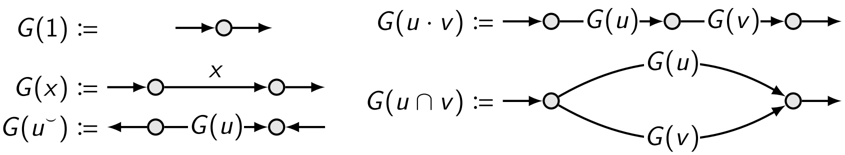
Graphs/Ground terms



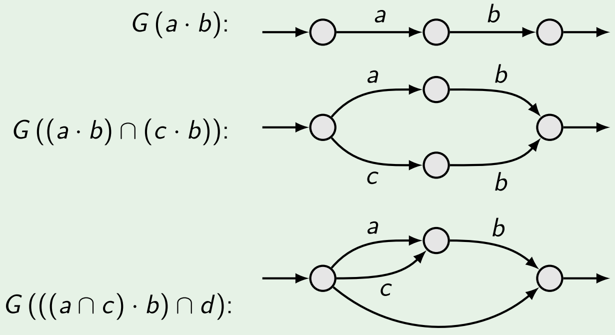
Example



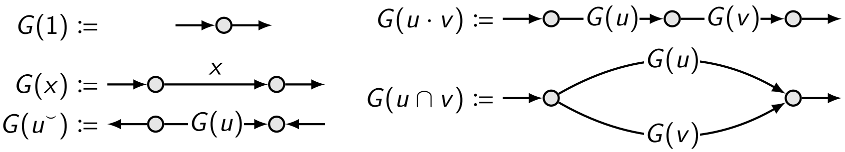
Graphs/Ground terms



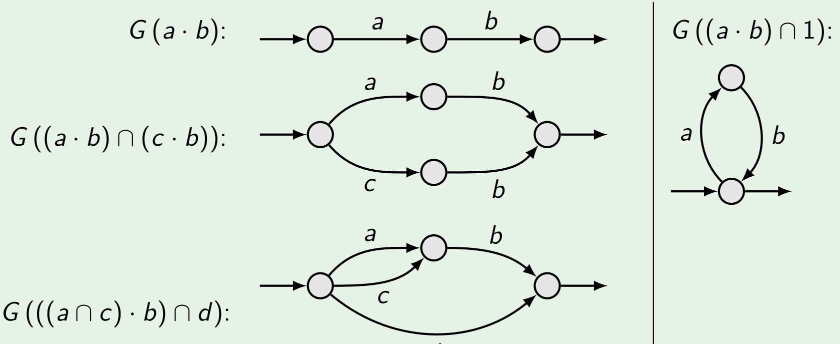
Example



Graphs/Ground terms



Example

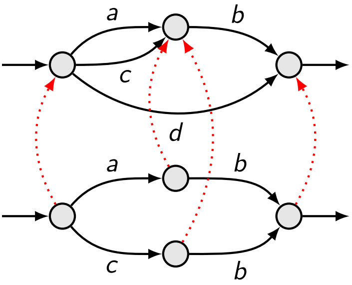


Preorder

Preorder on graphs

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

G :



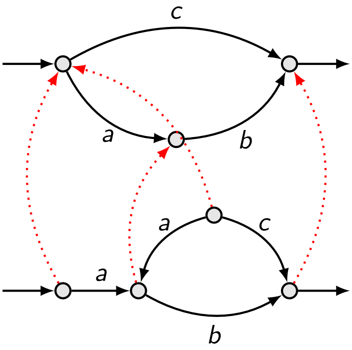
$((a \cap c) \cdot b) \cap d$

H :

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

Example: Modularity law

$$\mathrm{Rel} \models (a \cdot b) \cap c \leq a \cdot (b \cap a^{\smile} \cdot c) \tag{1}$$



Characterization theorem

Theorem

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

$$\mathrm{Rel} \models u \leqslant v \Leftrightarrow G(u) \blacktriangleleft G(v)$$

- 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

$$[\![\_]\!] : \mathrm{Reg}_X^{\smile\cap} \to \mathcal{P}(\mathrm{W}_X)$$

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

Graph language of an expression

$e \in \mathrm{Reg}_X^{\smile\cap}$,

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

Characterization theorem

${}^{\blacktriangleleft}S$ is the downwards closure of S with respect to $\blacktriangleleft$.

Theorem

$e, f \in \mathrm{Reg}_X^{\smile\cap}$,

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

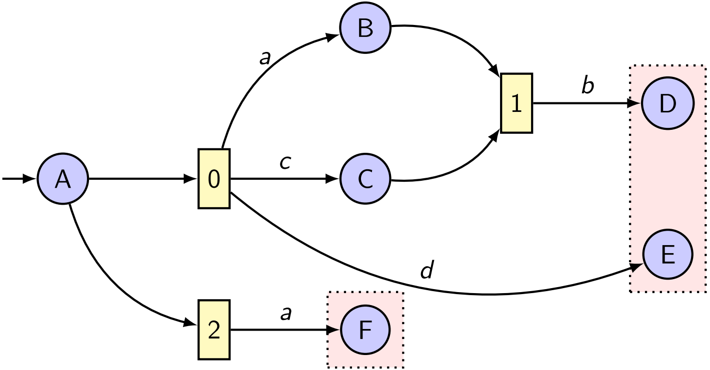
Follows easily from:

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

Outline

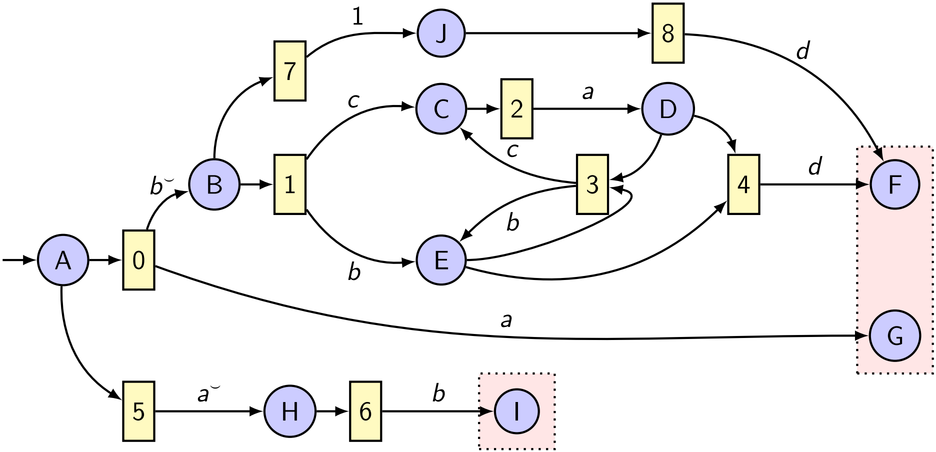
Example

$$(((a \cap c) \cdot b) \cap d) \cup a$$

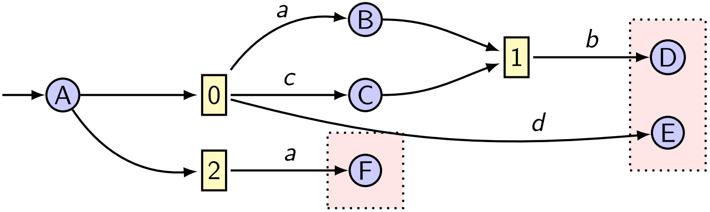


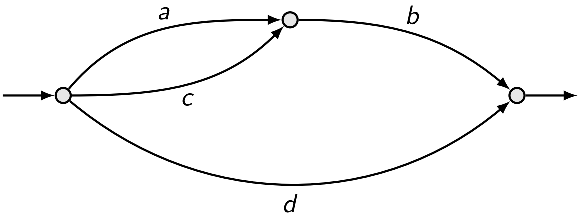
Example

$$\left(b^{\smile} \cdot (a \cdot c \cap b)^{\star} \cdot d\right) \cap a \;\cup\; a^{\smile} \cdot b$$

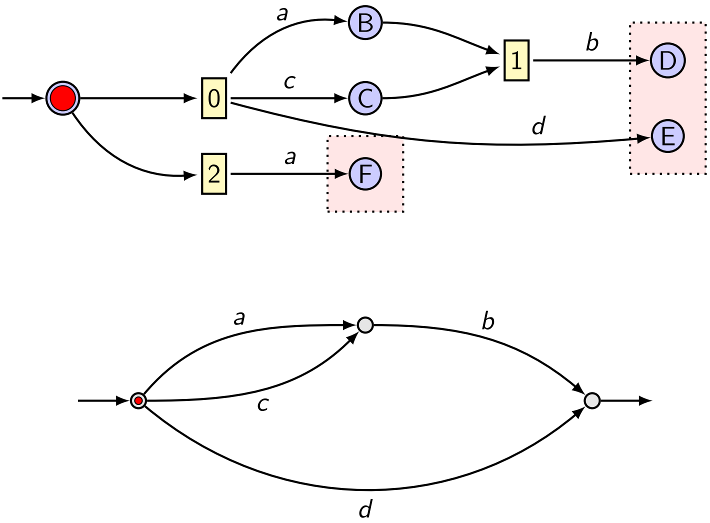


Reading a graph in an automaton

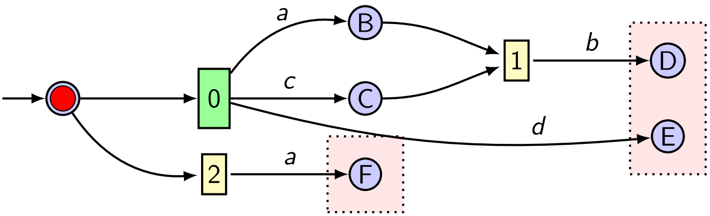


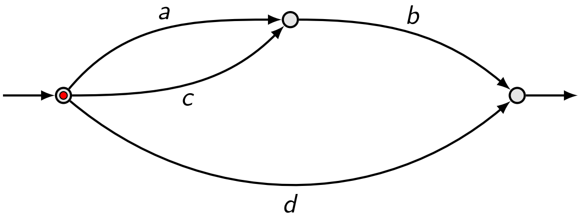


Reading a graph in an automaton

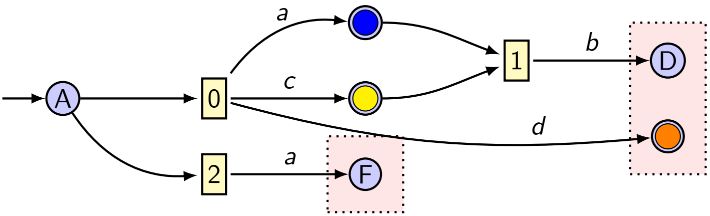


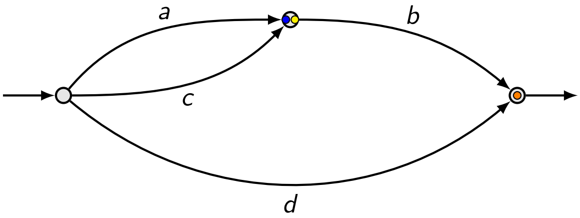
Reading a graph in an automaton



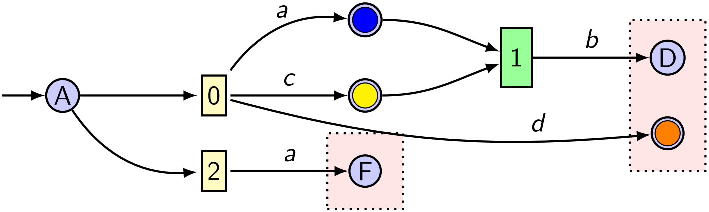


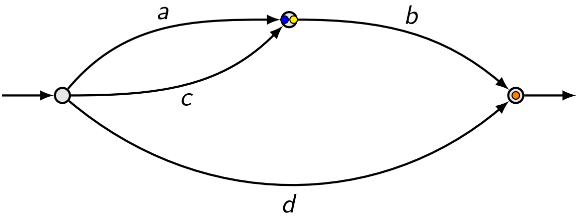
Reading a graph in an automaton



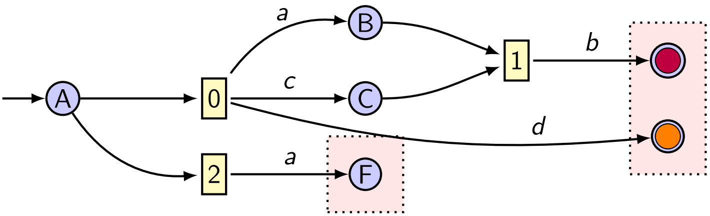


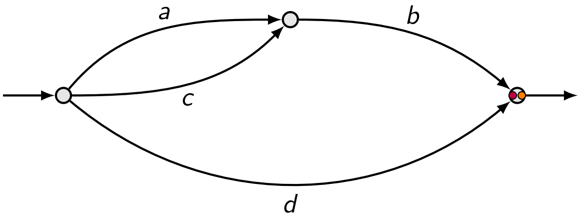
Reading a graph in an automaton





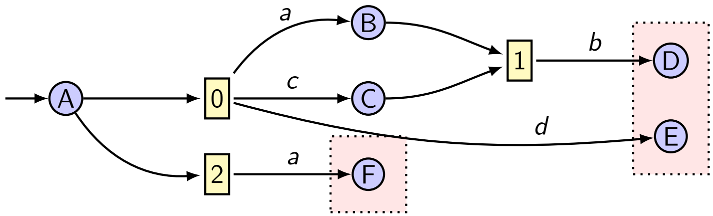
Reading a graph in an automaton

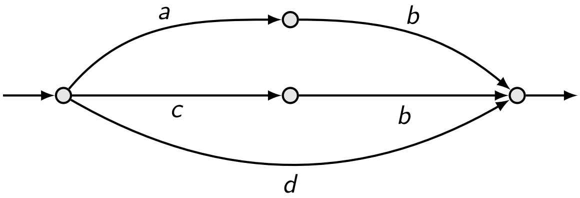




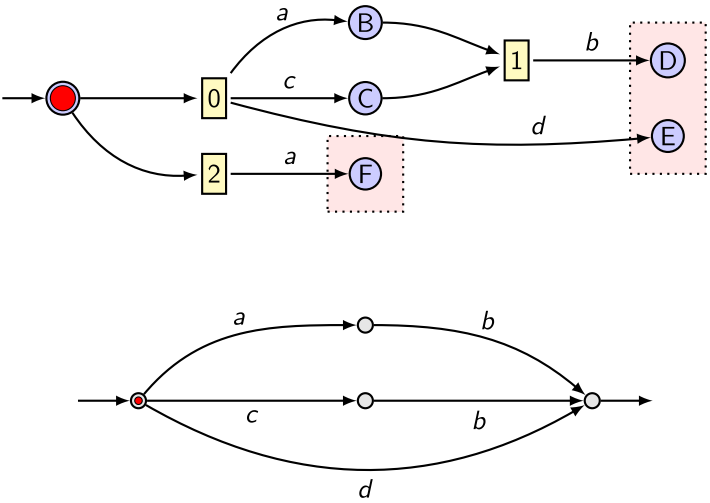
Success!

Reading a graph in an automaton

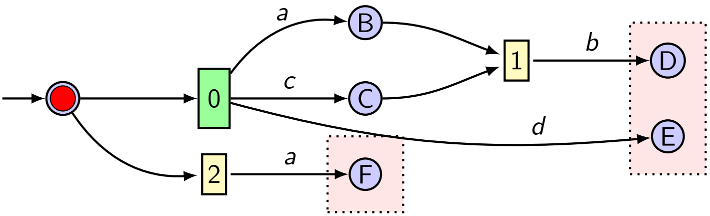


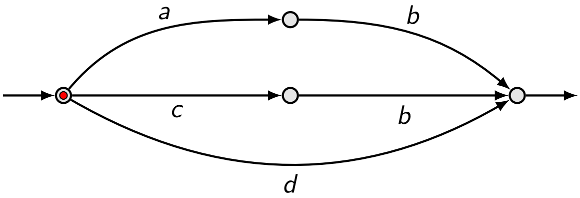


Reading a graph in an automaton

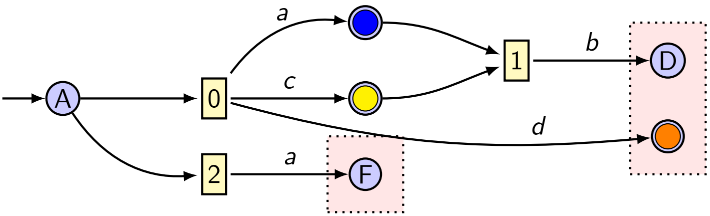


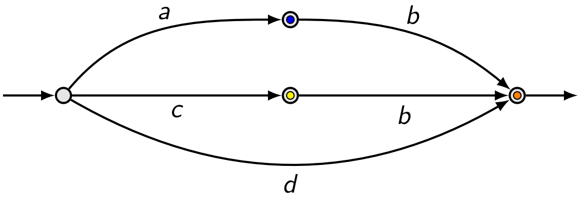
Reading a graph in an automaton



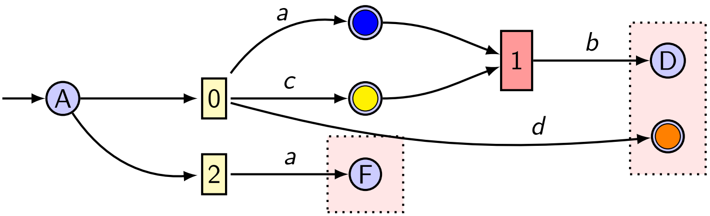


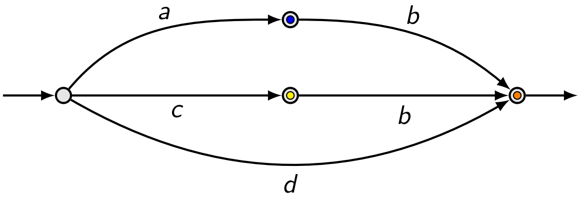
Reading a graph in an automaton





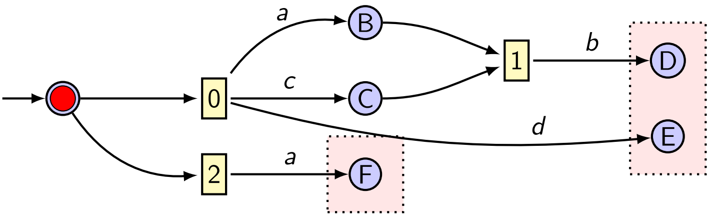
Reading a graph in an automaton

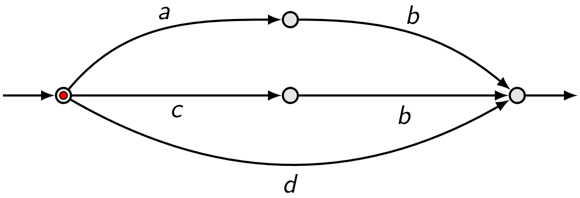




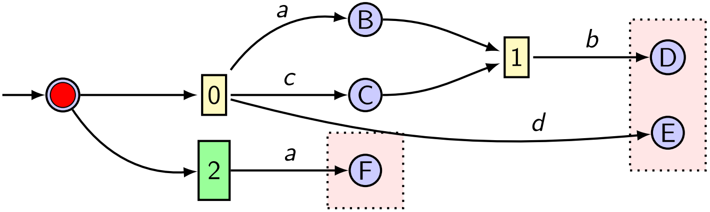
Failure!

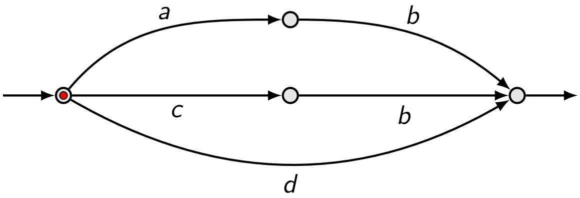
Reading a graph in an automaton



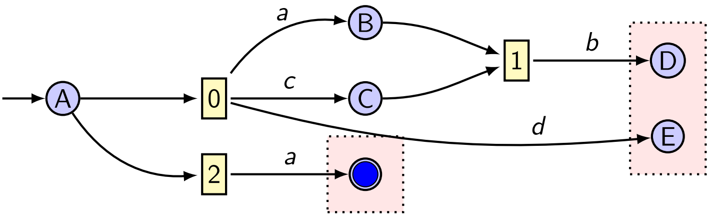


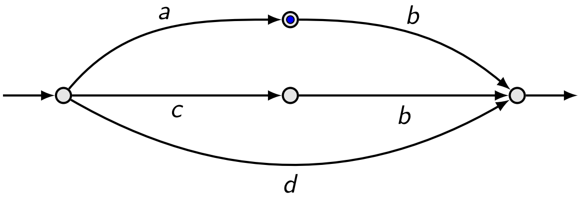
Reading a graph in an automaton



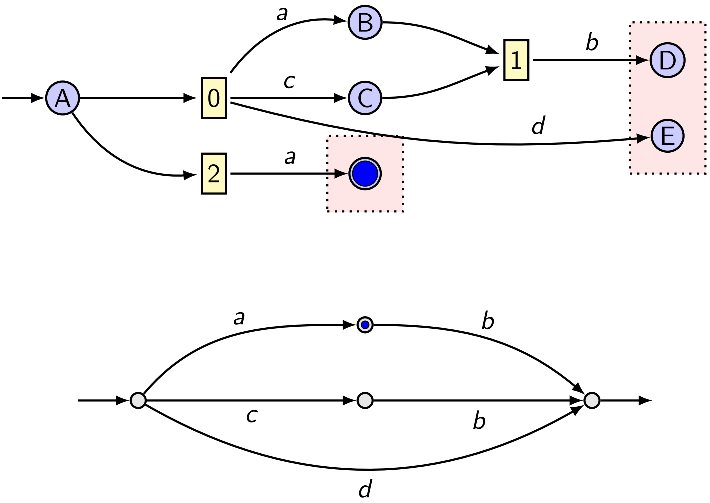


Reading a graph in an automaton





Reading a graph in an automaton



Failure!

Language recognised by an automaton

Correctness

For any $e \in \mathrm{Reg}_X^{\smile\cap}$,

$$e$$

Language recognised by an automaton

Correctness

For any $e \in \mathrm{Reg}_X^{\smile\cap}$,

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

Language recognised by an automaton

Correctness

For any $e \in \text{Reg}_X^{\smile\cap}$,

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

Language recognised by an automaton

Correctness

For any $e \in \mathrm{Reg}_X^{\smile\cap}$,

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

Language recognised by an automaton

Correctness

For any $e \in \mathrm{Reg}_X^{\smile\cap}$,

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

This far:

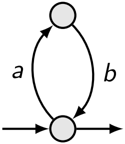
$e, f \in \mathrm{Reg}_X^{\smile\cap}$

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

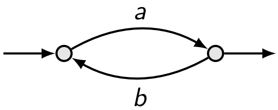
Outline

Restriction: identity-free lattice terms

$G((a \cdot b) \cap 1)$:

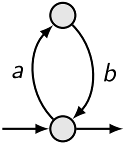


$G(a \cap b^{\smile})$:

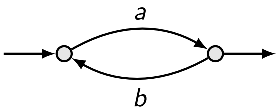


Restriction: identity-free lattice terms

$G((a \cdot b) \cap 1)$:



$G(a \cap b^{\smile})$:



$$u, v \in \mathrm{W}_X^{-} ::= 0 \mid 1 \mid x \in X \mid u \cdot v \mid u \cap v \mid u \cup v \mid u^{\star} \mid u^{\smile}$$

Identity-free Kleene Lattice

$$e, f \in \mathrm{Reg}_X^{\cap -} ::= 0 \mid 1 \mid x \in X \mid e \cdot f \mid e \cap f \mid e \cup f \mid e^{+} \mid e^{\smile}$$

Decision procedure

$e, f \in \mathrm{Reg}_X^{\cap -}$

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

Problem:

How to compare two Petri automata?

Decision procedure

$e, f \in \mathsf{Reg}_X^{\cap -}$

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

Problem:

How to compare two Petri automata?

. . . not that easily!

Decision procedure

$e, f \in \mathrm{Reg}_X^{\cap -}$

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

Problem:

How to compare two Petri automata?

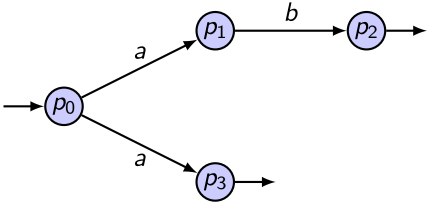
$\mathscr{L}(\mathscr{A}_1) \subseteq \mathscr{L}(\mathscr{A}_2)$ if and only if there is a simulation relation

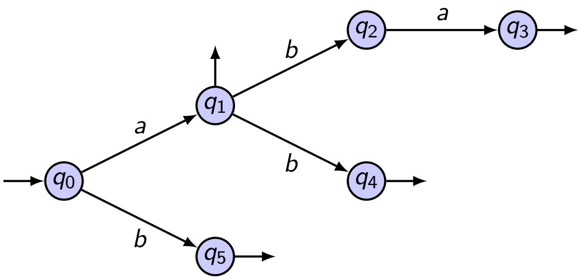
$$\preccurlyeq \subseteq \mathcal{P}(P_1) \times \mathcal{P}(P_2 \dashrightarrow P_1)$$

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

Simulations - Non-deterministic Finite Automata

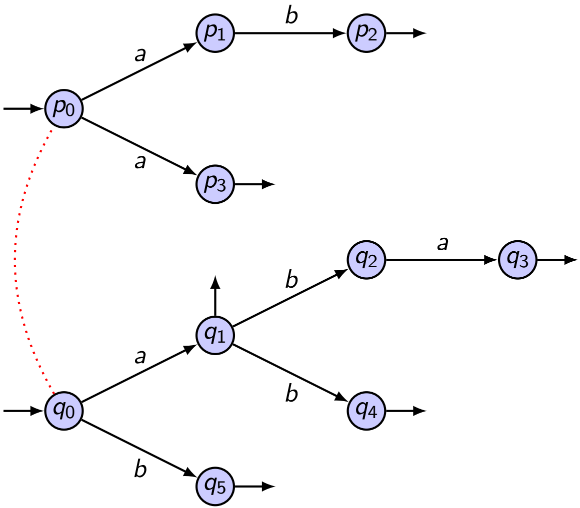
$$\preccurlyeq \subseteq Q_1 \times \mathcal{P}(Q_2)$$





Simulations - Non-deterministic Finite Automata

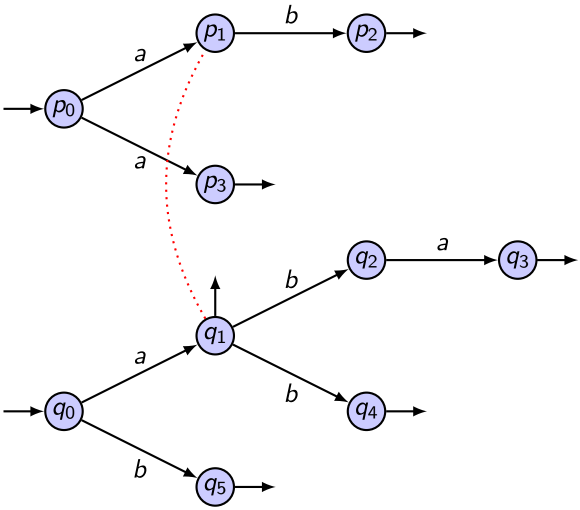
$$\preccurlyeq \subseteq Q_1 \times \mathcal{P}(Q_2)$$



$p_0 \preccurlyeq \{ \quad q_0 \quad \}$

Simulations - Non-deterministic Finite Automata

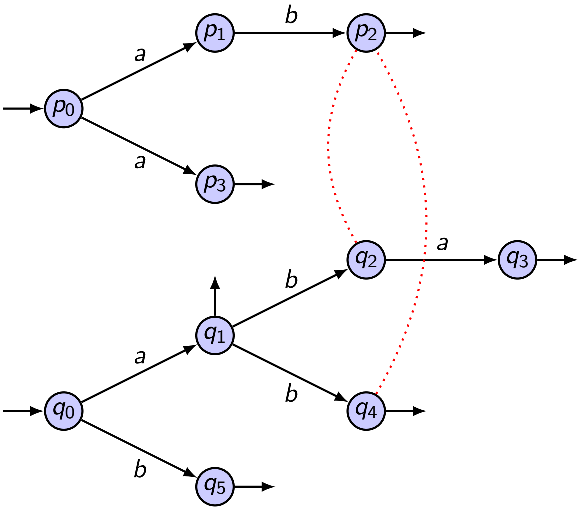
$$\preccurlyeq \subseteq Q_1 \times \mathcal{P}(Q_2)$$



$p_0 \preccurlyeq \{ \ q_0 \ \}$
$p_1 \preccurlyeq \{ \ q_1 \ \}$

Simulations - Non-deterministic Finite Automata

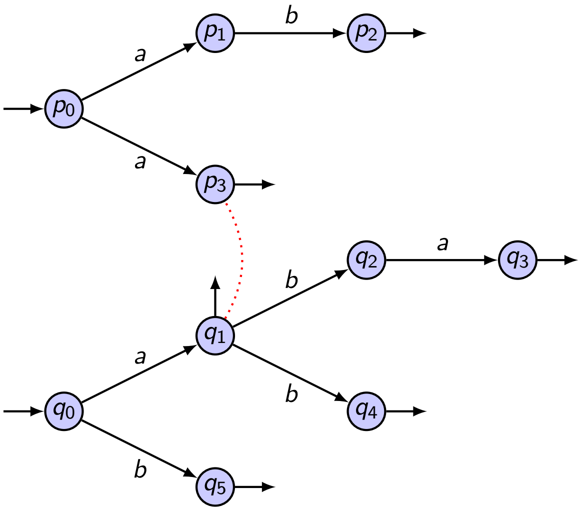
$$\preccurlyeq \subseteq Q_1 \times \mathcal{P}(Q_2)$$



$$p_0 \preccurlyeq \{ q_0 \}$$
$$p_1 \preccurlyeq \{ q_1 \}$$
$$p_2 \preccurlyeq \{q_2, q_4\}$$

Simulations - Non-deterministic Finite Automata

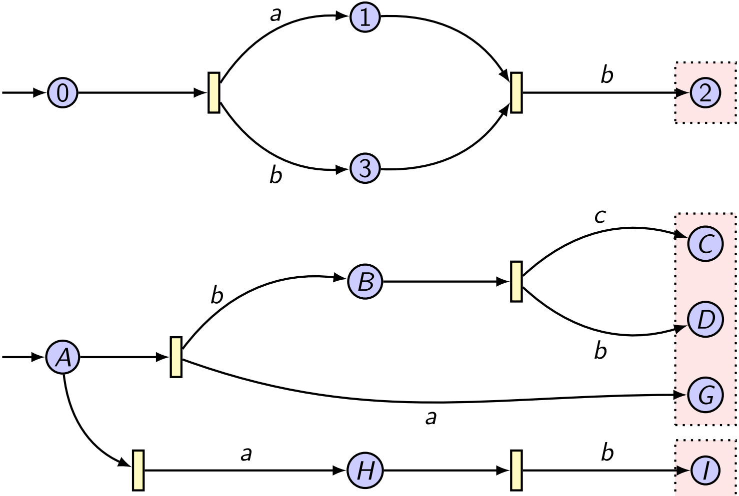
$$\preccurlyeq \subseteq Q_1 \times \mathcal{P}(Q_2)$$



$p_0 \preccurlyeq \{\ q_0\ \}$
$p_1 \preccurlyeq \{\ q_1\ \}$
$p_2 \preccurlyeq \{q_2, q_4\}$
$p_3 \preccurlyeq \{\ q_1\ \}$

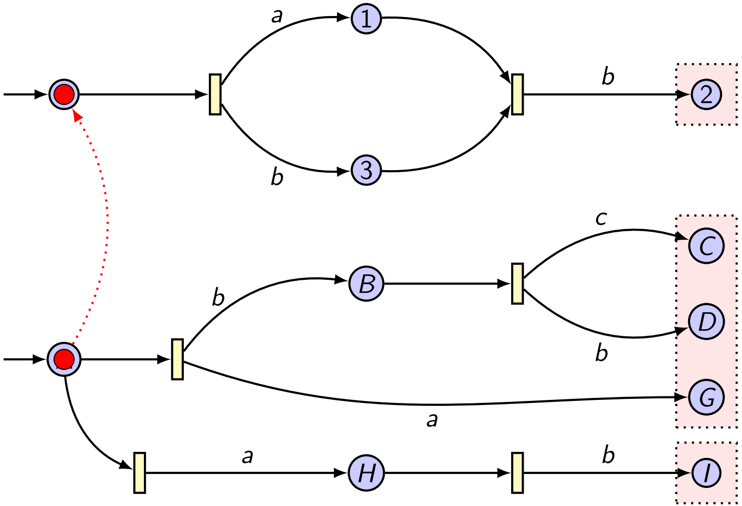
Simulation - Petri Automata

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



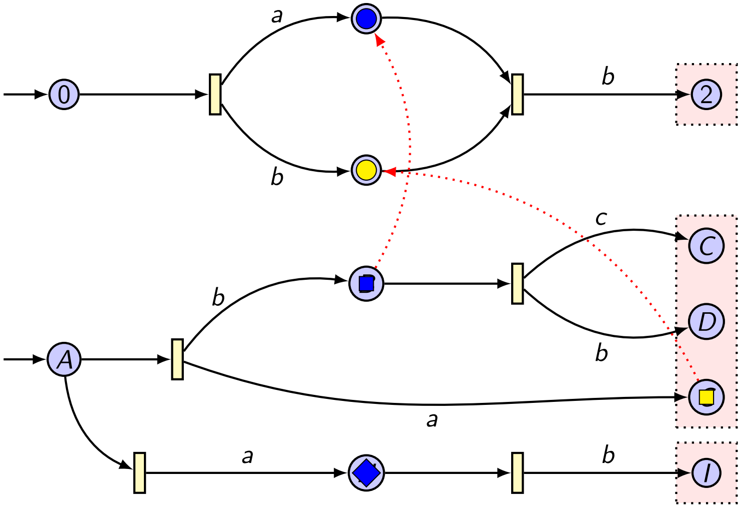
Simulation - Petri Automata

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



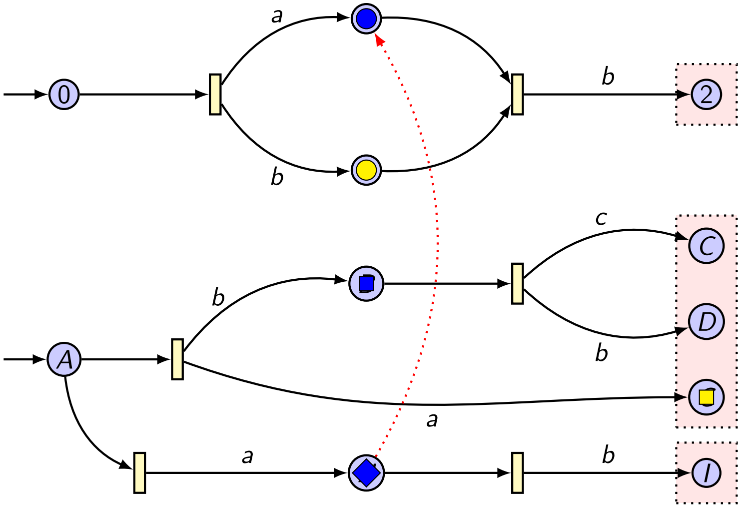
Simulation - Petri Automata

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



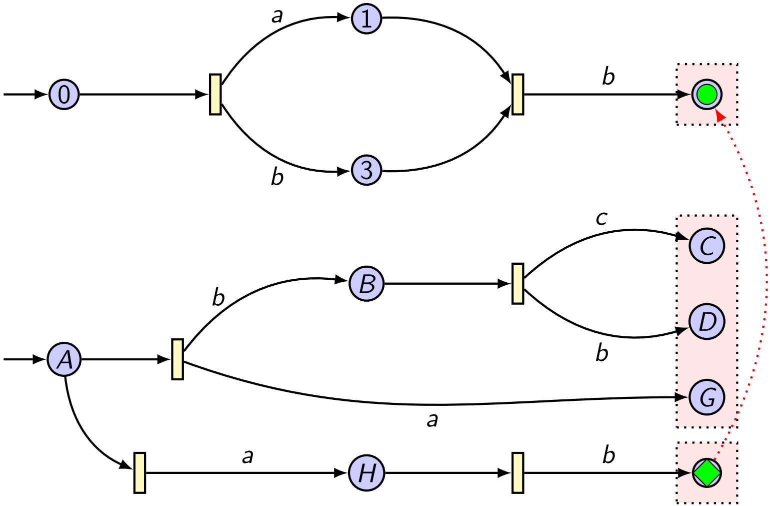
Simulation - Petri Automata

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



Simulation - Petri Automata

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



Conclusion and future work

Results

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

Ongoing/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.

Ongoing/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 simple automata equivalence, thus of relational equivalence for Identity-free Kleene Lattices.

Ongoing/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 simple automata equivalence, thus of relational equivalence for Identity-free Kleene Lattices.
- Simple Petri automata equivalence is EXPSPACE-complete.

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

Ongoing/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 simple automata equivalence, thus of relational equivalence for Identity-free Kleene Lattices.
- Simple Petri automata equivalence is EXPSPACE-complete.

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

Ongoing/Future work

- Converting back Petri automata into expressions.

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 simple automata equivalence, thus of relational equivalence for Identity-free Kleene Lattices.
- Simple Petri automata equivalence is EXPSPACE-complete.

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

Ongoing/Future work

- Converting back Petri automata into expressions.
- Decidability with 1 and/or $\_^{\smile}$.

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 simple automata equivalence, thus of relational equivalence for Identity-free Kleene Lattices.
- Simple Petri automata equivalence is EXPSPACE-complete.

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

Ongoing/Future work

- Converting back Petri automata into expressions.
- Decidability with 1 and/or $\_^{\smile}$.
- Complete axiomatization.

That's it!

Thank you !

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

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

Plan

1. Expressions
   - Kleene Algebra
   - Kleene Allegories
2. Graph languages
   - Ground terms
   - Regular expressions with intersection and converse
3. Petri Automata
   - Examples
   - Recognition by Petri automata
4. Comparing automata
5. Conclusions