Petri automata for Kleene Allegories

LICS 2015

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

July 5, 2015

Motivation : Relation Algebra

Let O be any set, and R, S and T any binary relations over O.

$$\mathrm{Rel} \models 1 \cup R^\star \cdot S \subseteq (R \cup S)^\star$$

$$\mathrm{Rel} \models (R \cap S) \cdot T \subseteq (R \cdot T) \cap (S \cdot T)$$

$$\mathrm{Rel} \models (R \cdot S) \cap T \subseteq R \cdot (S \cap R^\smile \cdot T)$$

$$\mathrm{Rel} \models (1 \cap (R \cdot S))^\star \subseteq R \cdot S \cdot R$$

Relation operators

identity relation	:	1
empty relation	:	0
composition	:	$R \cdot S$
union	:	$R \cup S$
intersection	:	$R \cap S$
converse	:	$R^\smile$
reflexive transitive closure	:	$R^\star$

Motivation : Relation Algebra

Let O be any set, and R, S and T any binary relations over O.

$$\mathrm{Rel} \models 1 \cup R^\star \cdot S \subseteq (R \cup S)^\star$$

$$\mathrm{Rel} \models (R \cap S) \cdot T \subseteq (R \cdot T) \cap (S \cdot T)$$

$$\mathrm{Rel} \models (R \cdot S) \cap T \subseteq R \cdot (S \cap R^\smile \cdot T)$$

$$\mathrm{Rel} \not\models (1 \cap (R \cdot S))^\star \subseteq R \cdot S \cdot R$$

Relation operators

identity relation	:	1
empty relation	:	0
composition	:	$R \cdot S$
union	:	$R \cup S$
intersection	:	$R \cap S$
converse	:	$R^\smile$
reflexive transitive closure	:	$R^\star$

Motivation : Relation Algebra

Let O be any set, and R, S and T any binary relations over O.

$$\mathrm{Rel} \models 1 \,\cup\, R^\star \cdot S \;\subseteq\; (R \cup S)^\star$$

$$\mathrm{Rel} \models (R \cap S) \cdot T \;\subseteq\; (R \cdot T) \cap (S \cdot T)$$

$$\mathrm{Rel} \models (R \cdot S) \cap T \;\subseteq\; R \cdot (S \cap R^\smile \cdot T)$$

$$\mathrm{Rel} \not\models (1 \cap (R \cdot S))^\star \;\subseteq\; R \cdot S \cdot R$$

Relation operators

identity relation	:	1
empty relation	:	0
composition	:	$R \cdot S$
union	:	$R \cup S$
intersection	:	$R \cap S$
converse	:	$R^\smile$
reflexive transitive closure	:	$R^\star$

Simple and boring : could it be done automatically ?

Outline

Regular expressions

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

Regular expressions

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

Theorem

$$\mathrm{Rel} \models e = f \Leftrightarrow \mathsf{L}(e) = \mathsf{L}(f)$$

where $L(e) \subseteq X^\star$ is the rational language represented by e.

Regular expressions

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

Theorem

$$\mathrm{Rel} \models e = f \;\Leftrightarrow\; \mathsf{L}(e) = \mathsf{L}(f)$$

where $L(e) \subseteq X^\star$ is the rational language represented by e.

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

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

$$\mathrm{Rel} \models e = f \overset{?}{\Leftrightarrow} \mathsf{L}(e) = \mathsf{L}(f)$$

Kleene Allegories

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

$$\mathrm{Rel} \models e = f \;\not\Leftarrow\; \mathsf{L}(e) = \mathsf{L}(f)$$

Counterexample

$\mathsf{L}(a \cap b) = \emptyset = \mathsf{L}(0)$	$\mathsf{L}(a) = \{a\} = \mathsf{L}(a^\smile)$
$\mathrm{Rel} \not\models a \cap b = 0$	$\mathrm{Rel} \not\models a = a^\smile$

A different approach is needed.

Outline

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

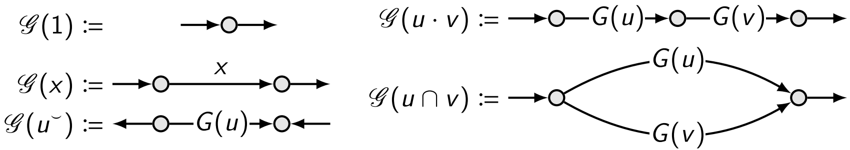
Outline

Graphs/Ground terms

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

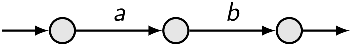
Graphs/Ground terms

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



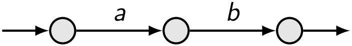
Examples

$\mathscr{G}\,(a \cdot b)$:

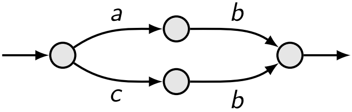


Examples

$\mathscr{G}(a \cdot b)$:

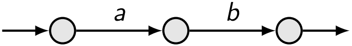


$\mathscr{G}((a \cdot b) \cap (c \cdot b))$:

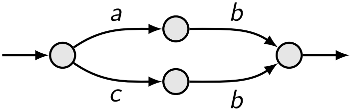


Examples

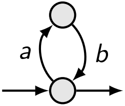
$\mathscr{G}(a \cdot b)$:



$\mathscr{G}((a \cdot b) \cap (c \cdot b))$:

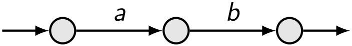


$\mathscr{G}((a \cdot b) \cap 1)$:

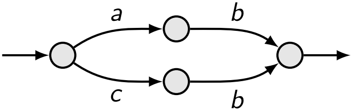


Examples

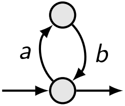
$\mathscr{G}(a \cdot b)$:



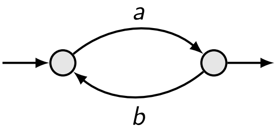
$\mathscr{G}((a \cdot b) \cap (c \cdot b))$:



$\mathscr{G}((a \cdot b) \cap 1)$:



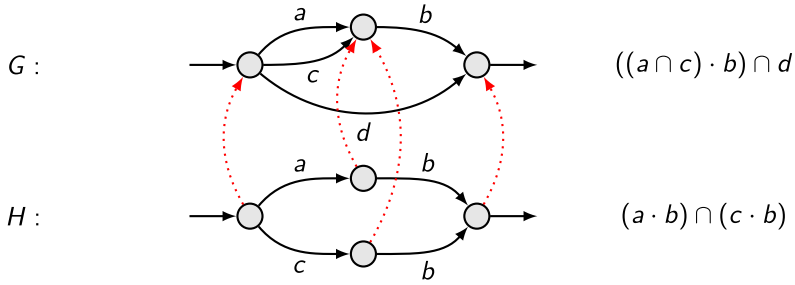
$\mathscr{G}(a \cap b^{\smile})$:



Preorder

Preorder on graphs

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



Characterization theorem

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

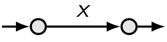
Theorem

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

- 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

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

$\mathscr{G}(1) := \{$  $\}$ $\quad \mathscr{G}(x) := \{$ 

$\}$

$$\mathscr{G}(e^\smile) := \left\{ G^{-1} \mid G \in \mathscr{G}(e) \right\}$$

$$\mathscr{G}(e \cdot f) := \{ G \,;\, G' \mid G \in \mathscr{G}(e) \text{ and } G' \in \mathscr{G}(f) \}$$

$$\mathscr{G}(e \cap f) := \{ G \parallel G' \mid G \in \mathscr{G}(e) \text{ and } G' \in \mathscr{G}(f) \}$$

$$\mathscr{G}(0) := \emptyset \quad \mathscr{G}(e \cup f) := \mathscr{G}(e) \cup \mathscr{G}(f)$$

$$\mathscr{G}(e^\star) := \bigcup_{n \in \mathbb{N}} \{ G_1; \cdots ; G_n \mid \forall i, G_i \in \mathscr{G}(e) \} \; .$$

Characterization theorem

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

Theorem

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

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

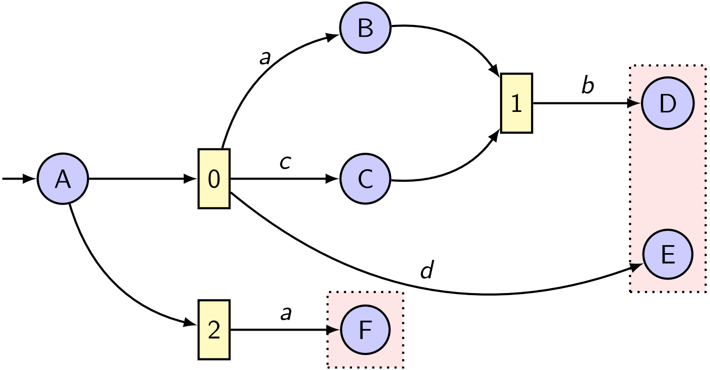
Follows easily from:

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

Outline

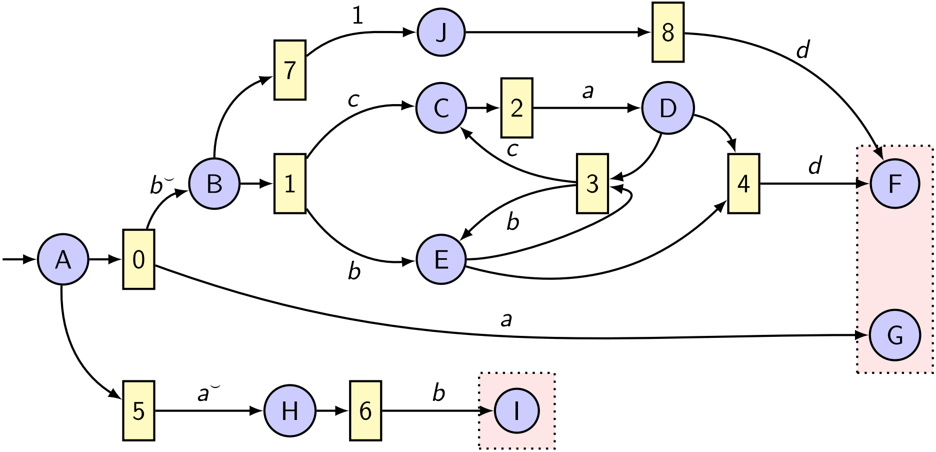
Example

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

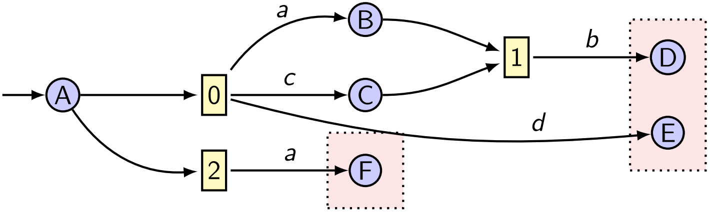


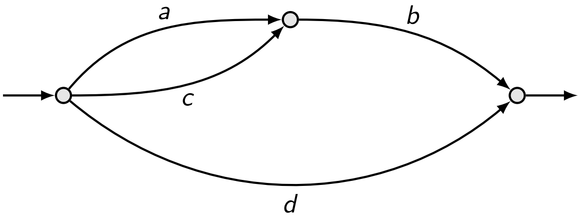
Example

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

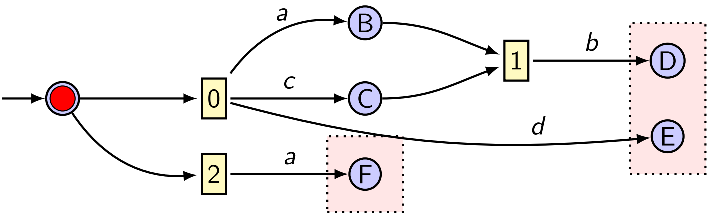


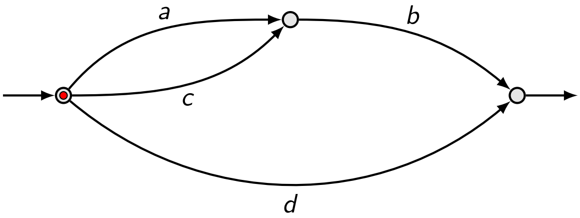
Reading a graph in an automaton



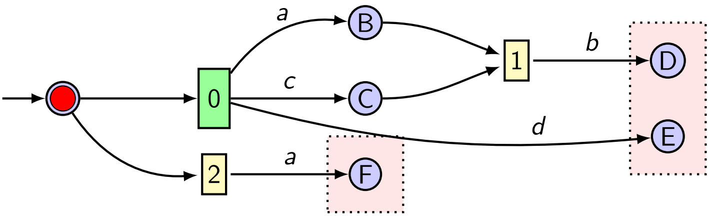


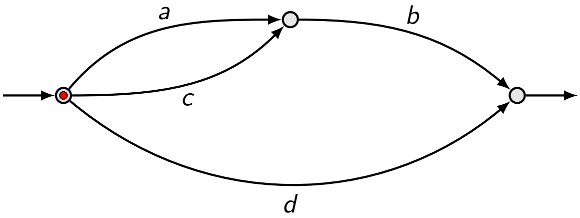
Reading a graph in an automaton



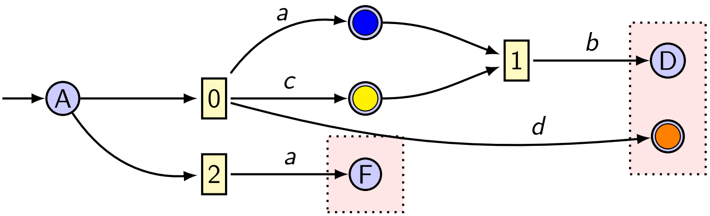


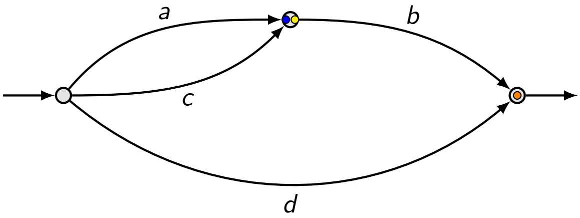
Reading a graph in an automaton



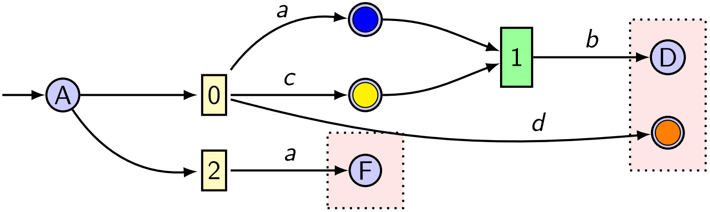


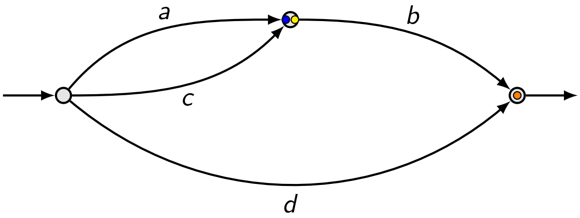
Reading a graph in an automaton



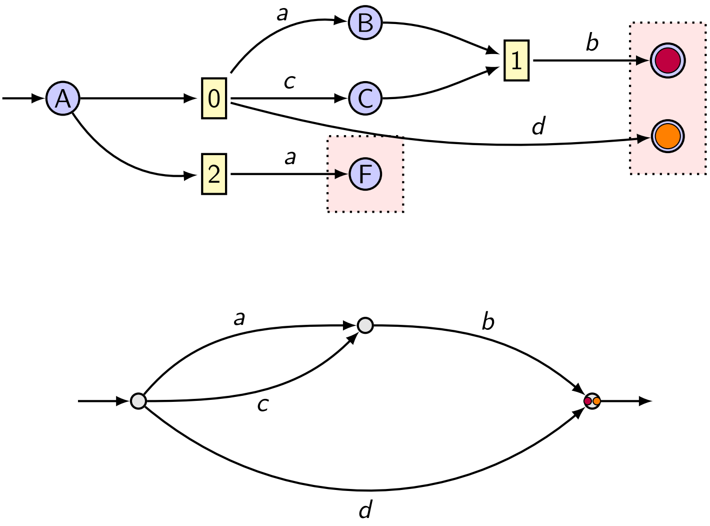


Reading a graph in an automaton



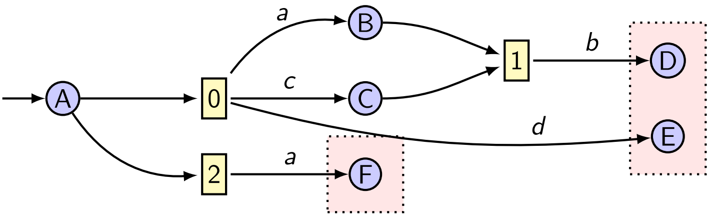


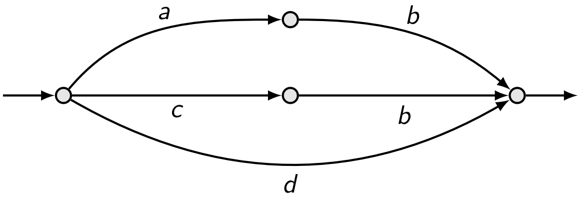
Reading a graph in an automaton



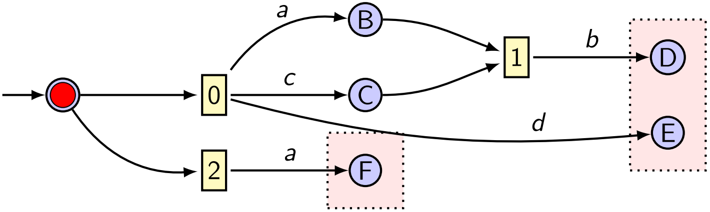
Success!

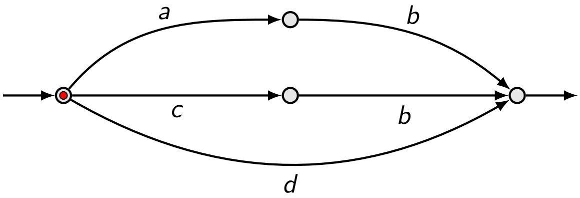
Reading a graph in an automaton



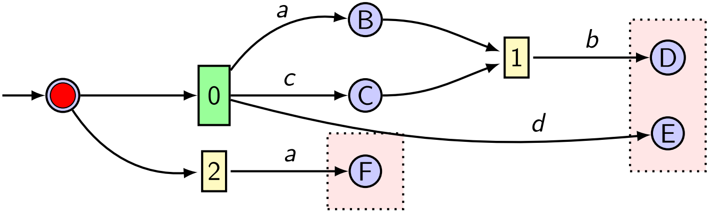


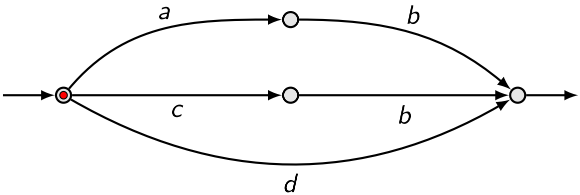
Reading a graph in an automaton



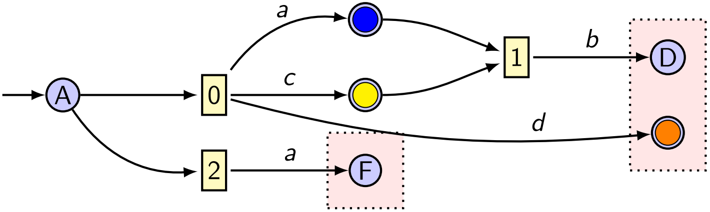


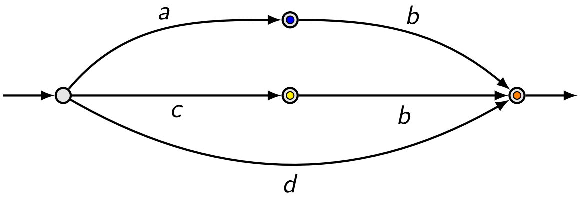
Reading a graph in an automaton



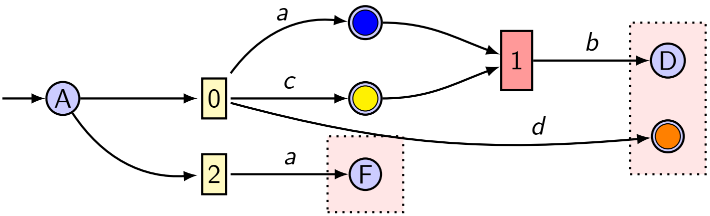


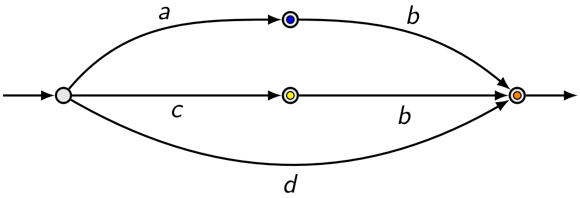
Reading a graph in an automaton





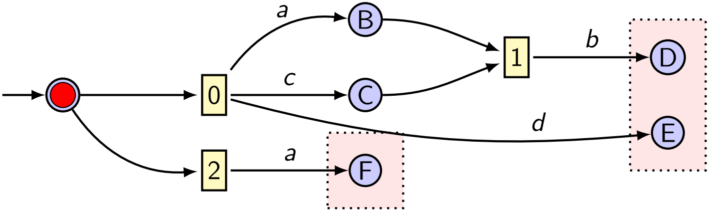
Reading a graph in an automaton

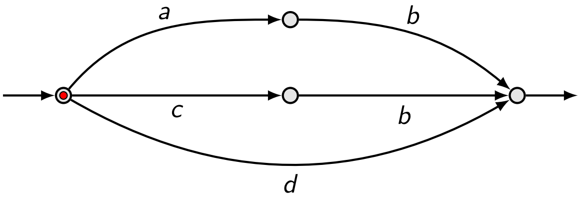




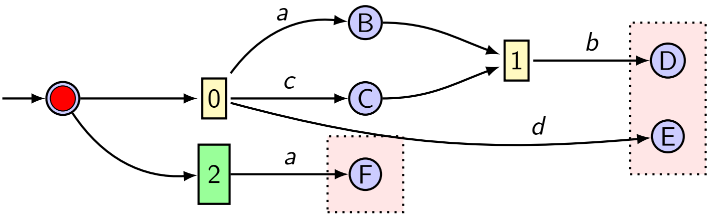
Failure!

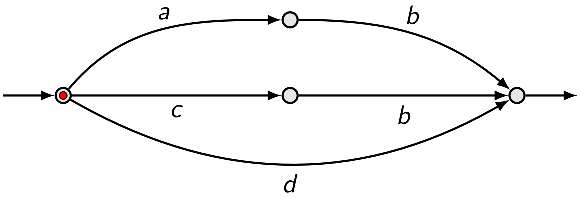
Reading a graph in an automaton



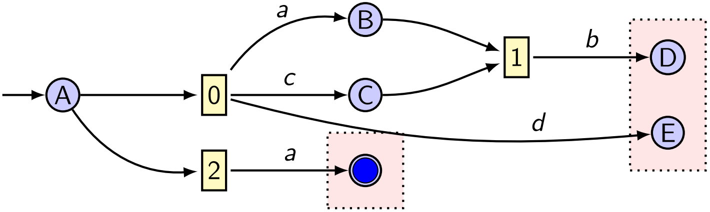


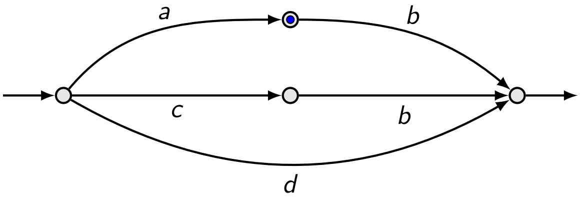
Reading a graph in an automaton



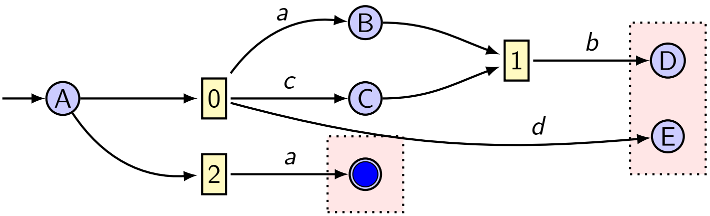


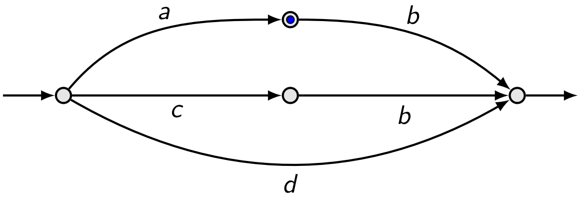
Reading a graph in an automaton





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

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

Language recognised by an automaton

Correctness

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

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

Language recognised by an automaton

Correctness

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

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

So far:

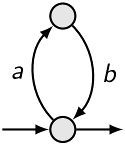
$e, f \in \mathrm{Reg}_X^{\smile\cap}$

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

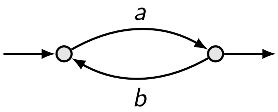
Outline

Restriction: identity-free lattice terms

$\mathscr{G}\left((a \cdot b) \cap 1\right)$:

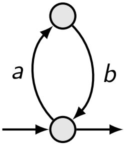


$\mathscr{G}\left(a \cap b^{\smile}\right)$:

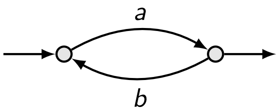


Restriction: identity-free lattice terms

$\mathscr{G}((a \cdot b) \cap 1)$:



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



Identity-free Kleene Lattice

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

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

Decision procedure

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

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

Problem:

How to compare two Petri automata?

Decision procedure

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

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

Problem:

How to compare two Petri automata?

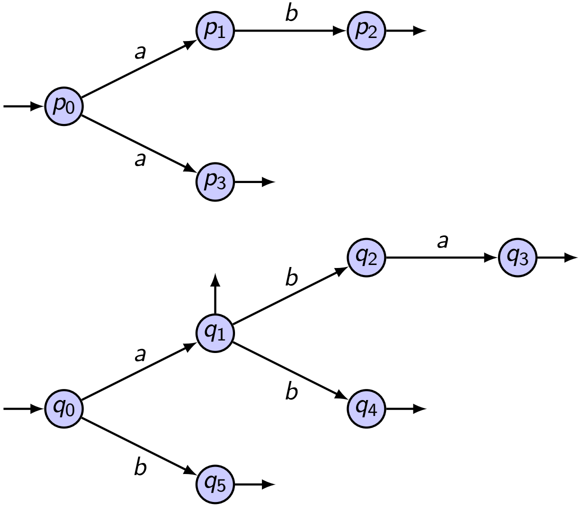
$\mathscr{L}(\mathscr{A}_1) \subseteq \mathscr{L}(\mathscr{A}_2)$ if and only if there is a simulation relation

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

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

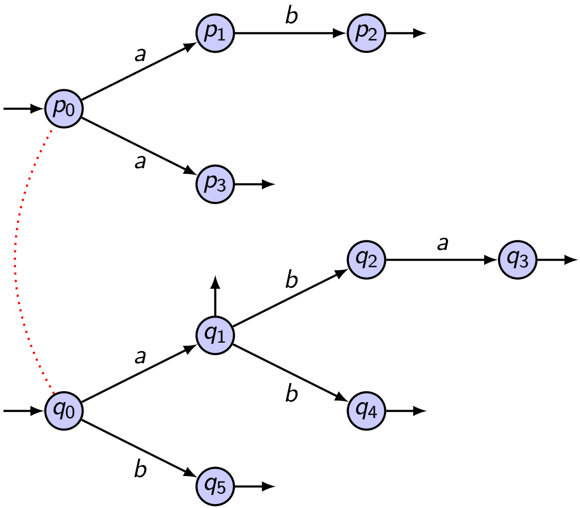
Simulations - Non-deterministic Finite Automata

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



Simulations - Non-deterministic Finite Automata

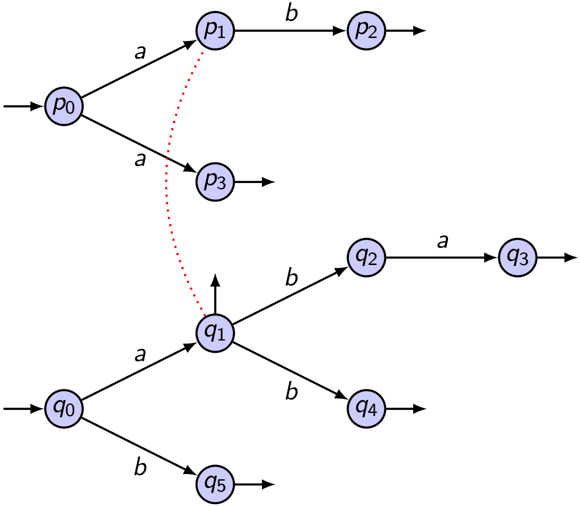
$$\preccurlyeq \subseteq Q_1 \times \mathcal{P}(Q_2)$$



$p_0 \preccurlyeq \{ \; q_0 \; \}$

Simulations - Non-deterministic Finite Automata

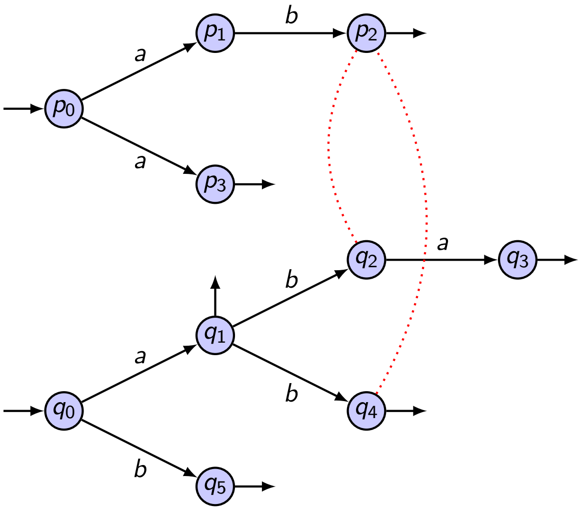
$$\preccurlyeq \subseteq Q_1 \times \mathcal{P}(Q_2)$$



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

Simulations - Non-deterministic Finite Automata

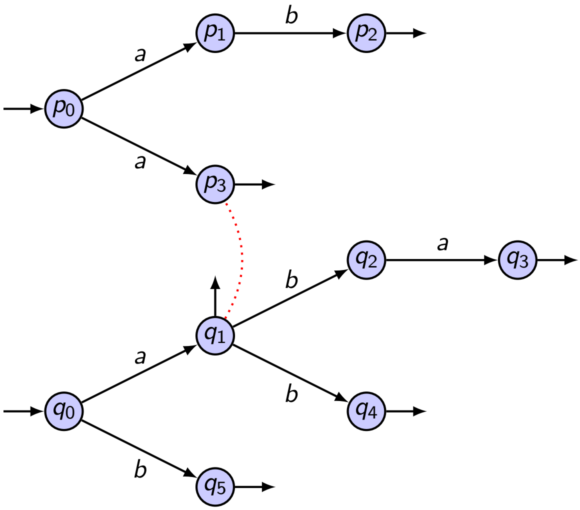
$$\preccurlyeq \subseteq Q_1 \times \mathcal{P}(Q_2)$$



$$\begin{aligned} p_0 &\preccurlyeq \{q_0\} \\ p_1 &\preccurlyeq \{q_1\} \\ p_2 &\preccurlyeq \{q_2, q_4\} \end{aligned}$$

Simulations - Non-deterministic Finite Automata

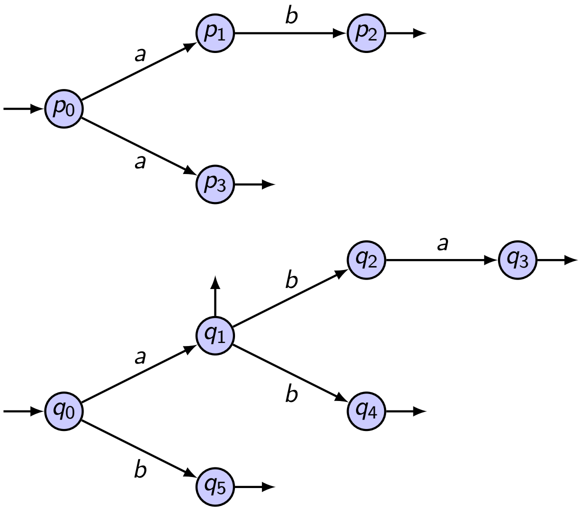
$$\preccurlyeq \subseteq Q_1 \times \mathcal{P}(Q_2)$$



$$\begin{array}{lcc} p_0 & \preccurlyeq & \{\ q_0\ \} \\ p_1 & \preccurlyeq & \{\ q_1\ \} \\ p_2 & \preccurlyeq & \{q_2, q_4\} \\ p_3 & \preccurlyeq & \{\ q_1\ \} \end{array}$$

Simulations - Non-deterministic Finite Automata

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



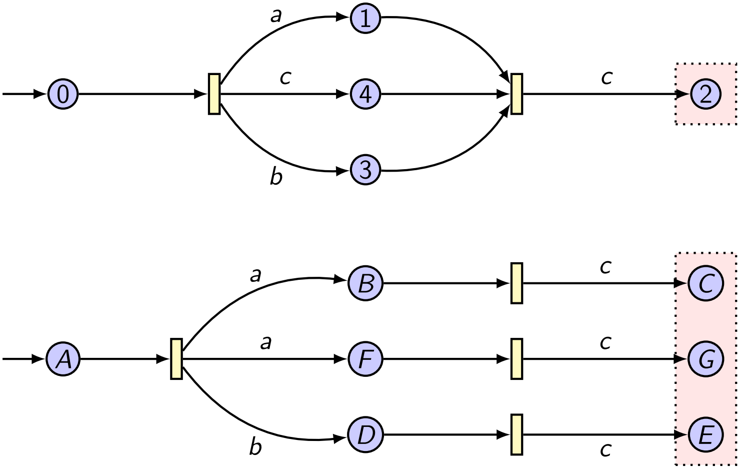
$$\begin{aligned} p_0 &\preccurlyeq \{\ q_0\ \} \\ p_1 &\preccurlyeq \{\ q_1\ \} \\ p_2 &\preccurlyeq \{q_2, q_4\} \\ p_3 &\preccurlyeq \{\ q_1\ \} \end{aligned}$$

$+$ condition for accepting states

Simulation - Petri Automata

Simulation relation

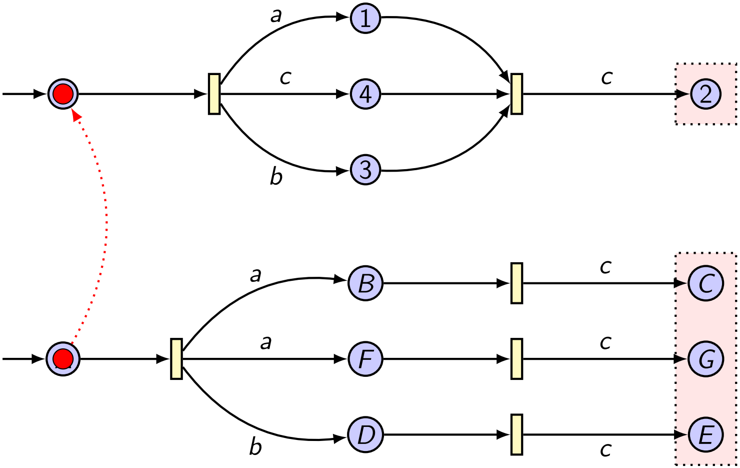
$$\preccurlyeq \subseteq \mathcal{P}(P_1) \times \mathcal{P}(P_2 \dashrightarrow P_1)$$



Simulation - Petri Automata

Simulation relation

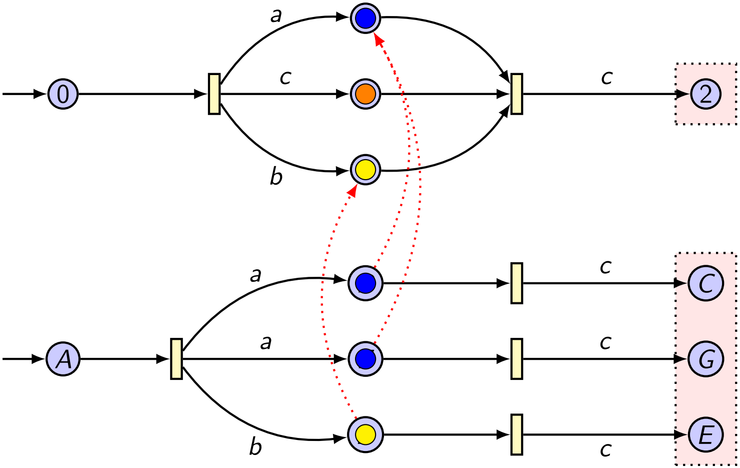
$$\preccurlyeq \subseteq \mathcal{P}(P_1) \times \mathcal{P}(P_2 \dashrightarrow P_1)$$



Simulation - Petri Automata

Simulation relation

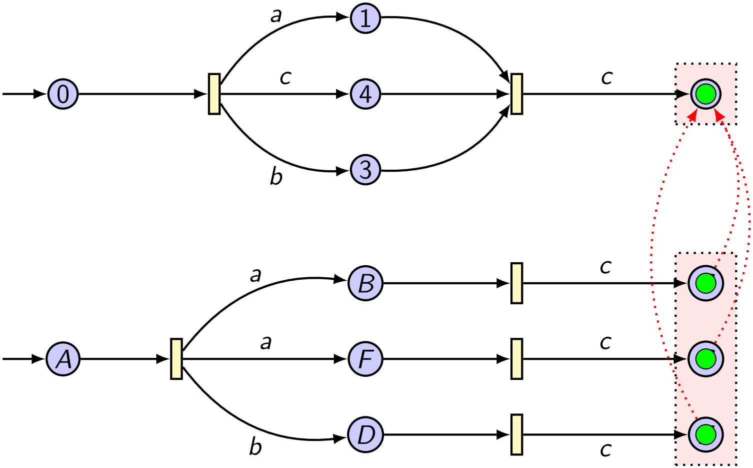
$$\preccurlyeq \subseteq \mathcal{P}(P_1) \times \mathcal{P}(P_2 \dashrightarrow P_1)$$



Simulation - Petri Automata

Simulation relation

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



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.
- (Un)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.
- (Un)decidability with 1 and/or $\_^{\smile}$.
- Complete axiomatization.

That's it!

Thank you !

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

Plan

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