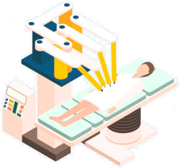
The Equational Theory of Positive Relation Algebra

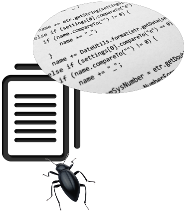
PACE meeting

November 10th, 2016

Paul Brunet
Équipe Plume – Univ Lyon, UCB Lyon 1, CNRS, ENS de Lyon, LIP

Critical software

Imperative programs

List of instructions:

```
cmd1;
cmd2;
cmd3;
```

Relation between memory states:

Imperative programs

List of instructions:

```
cmd1; ←
cmd2;
cmd3;
```

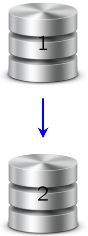
Relation between memory states:

Imperative programs

List of instructions:

```
cmd1;
cmd2;←
cmd3;
```

Relation between memory states:



Imperative programs

List of instructions:

```
cmd1;
cmd2;
cmd3;←
```

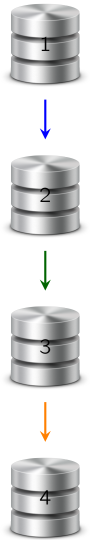
Relation between memory states:

Imperative programs

List of instructions:

```
cmd1;
cmd2;
cmd3;
```

Relation between memory states:

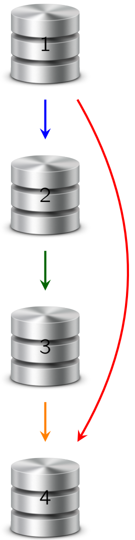


Imperative programs

List of instructions:

```
cmd1;
cmd2;
cmd3;
```

Relation between memory states:



Relational program semantics

$x \leftarrow 1; ((y \leftarrow x) \oplus (y \leftarrow 0))$

$(x \leftarrow 1; y \leftarrow x) \oplus (x \leftarrow 1; y \leftarrow 0)$

Relational program semantics

$$x \leftarrow 1; ((y \leftarrow x) \oplus (y \leftarrow 0)) \longmapsto a \cdot (b \cup c)$$

$$(x \leftarrow 1; y \leftarrow x) \oplus (x \leftarrow 1; y \leftarrow 0) \longmapsto (a \cdot b) \cup (a \cdot c)$$

Relational program semantics

$x \leftarrow 1; ((y \leftarrow x) \oplus (y \leftarrow 0))$ $\quad$ $(x \leftarrow 1; y \leftarrow x) \oplus (x \leftarrow 1; y \leftarrow 0)$

$$\mathrm{Rel} \models a \cdot (b \cup c) = (a \cdot b) \cup (a \cdot c)$$

Relational program semantics

$$x \leftarrow 1; ((y \leftarrow x) \oplus (y \leftarrow 0)) \equiv (x \leftarrow 1; y \leftarrow x) \oplus (x \leftarrow 1; y \leftarrow 0)$$

$$\mathrm{Rel} \models a \cdot (b \cup c) = (a \cdot b) \cup (a \cdot c)$$

Relation Algebra

Relational Operators

identity relation	1
empty relation	0
universal relation	$\top$
composition	$R \cdot S$
union	$R \cup S$
intersection	$R \cap S$
converse	$R^{\smile}$
reflexive transitive closure	$R^{\star}$
complement	R^{c}

Relation Algebra

Relational Operators

identity relation	1	$:= \{\langle x, x\rangle \mid x \in O\}$
empty relation	0	
universal relation	$\top$	
composition	$R \cdot S$	
union	$R \cup S$	
intersection	$R \cap S$	
converse	$R^{\smile}$	
reflexive transitive closure	$R^{\star}$	
complement	R^{c}	

Relation Algebra

Relational Operators

identity relation	1	$:= \{\langle x, x\rangle \mid x \in O\}$
empty relation	0	$:= \emptyset$
universal relation	$\top$	
composition	$R \cdot S$	
union	$R \cup S$	
intersection	$R \cap S$	
converse	$R^{\smile}$	
reflexive transitive closure	$R^{\star}$	
complement	R^{c}	

Relation Algebra

Relational Operators

identity relation	1	$:= \{\langle x, x\rangle \mid x \in O\}$
empty relation	0	$:= \emptyset$
universal relation	$\top$	$:= O \times O$
composition	$R \cdot S$	
union	$R \cup S$	
intersection	$R \cap S$	
converse	$R^{\smile}$	
reflexive transitive closure	$R^{\star}$	
complement	R^{c}	

Relation Algebra

Relational Operators

identity relation	1	$:= \{\langle x, x\rangle \mid x \in O\}$
empty relation	0	$:= \emptyset$
universal relation	$\top$	$:= O \times O$
composition	$R \cdot S$	$:= \{\langle x, z\rangle \mid \exists y : x\, R\, y\, S\, z\}$
union	$R \cup S$	
intersection	$R \cap S$	
converse	$R^{\smile}$	
reflexive transitive closure	$R^{\star}$	
complement	R^{c}	

Relation Algebra

Relational Operators

identity relation	1	$:= \{\langle x, x\rangle \mid x \in O\}$
empty relation	0	$:= \emptyset$
universal relation	$\top$	$:= O \times O$
composition	$R \cdot S$	$:= \{\langle x, z\rangle \mid \exists y : x\, R\, y\, S\, z\}$
union	$R \cup S$	$:= \{\langle x, y\rangle \mid x\, R\, y \text{ or } x\, S\, y\}$
intersection	$R \cap S$	
converse	$R^{\smile}$	
reflexive transitive closure	$R^{\star}$	
complement	R^{c}	

Relation Algebra

Relational Operators

identity relation	1	$:= \{\langle x, x\rangle \mid x \in O\}$
empty relation	0	$:= \emptyset$
universal relation	$\top$	$:= O \times O$
composition	$R \cdot S$	$:= \{\langle x, z\rangle \mid \exists y : x\, R\, y\, S\, z\}$
union	$R \cup S$	$:= \{\langle x, y\rangle \mid x\, R\, y \text{ or } x\, S\, y\}$
intersection	$R \cap S$	$:= \{\langle x, y\rangle \mid x\, R\, y \text{ and } x\, S\, y\}$
converse	$R^{\smile}$	
reflexive transitive closure	$R^{\star}$	
complement	R^{c}	

Relation Algebra

Relational Operators

identity relation	1	$:= \{\langle x, x\rangle \mid x \in O\}$
empty relation	0	$:= \emptyset$
universal relation	$\top$	$:= O \times O$
composition	$R \cdot S$	$:= \{\langle x, z\rangle \mid \exists y : x\, R\, y\, S\, z\}$
union	$R \cup S$	$:= \{\langle x, y\rangle \mid x\, R\, y \text{ or } x\, S\, y\}$
intersection	$R \cap S$	$:= \{\langle x, y\rangle \mid x\, R\, y \text{ and } x\, S\, y\}$
converse	$R^{\smile}$	$:= \{\langle x, y\rangle \mid y\, R\, x\}$
reflexive transitive closure	$R^{\star}$	
complement	R^{c}	

Relation Algebra

Relational Operators

identity relation	1	$:= \{\langle x, x\rangle \mid x \in O\}$
empty relation	0	$:= \emptyset$
universal relation	$\top$	$:= O \times O$
composition	$R \cdot S$	$:= \{\langle x, z\rangle \mid \exists y : x\, R\, y\, S\, z\}$
union	$R \cup S$	$:= \{\langle x, y\rangle \mid x\, R\, y \text{ or } x\, S\, y\}$
intersection	$R \cap S$	$:= \{\langle x, y\rangle \mid x\, R\, y \text{ and } x\, S\, y\}$
converse	$R^{\smile}$	$:= \{\langle x, y\rangle \mid y\, R\, x\}$
reflexive transitive closure	$R^{\star}$	$:= 1 \cup R \cup RR \cup RRR \cup \dots$
complement	R^{c}	

Relation Algebra

Relational Operators

identity relation	1	$:= \{\langle x, x\rangle \mid x \in O\}$
empty relation	0	$:= \emptyset$
universal relation	$\top$	$:= O \times O$
composition	$R \cdot S$	$:= \{\langle x, z\rangle \mid \exists y : x\, R\, y\, S\, z\}$
union	$R \cup S$	$:= \{\langle x, y\rangle \mid x\, R\, y \text{ or } x\, S\, y\}$
intersection	$R \cap S$	$:= \{\langle x, y\rangle \mid x\, R\, y \text{ and } x\, S\, y\}$
converse	$R^{\smile}$	$:= \{\langle x, y\rangle \mid y\, R\, x\}$
reflexive transitive closure	$R^{\star}$	$:= 1 \cup R \cup RR \cup RRR \cup \ldots$
complement	R^{c}	$:= \{\langle x, y\rangle \mid \text{not } x\, R\, y\}$

Universal laws of relation algebra

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

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

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

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

$$\mathrm{Rel} \models \top \cdot R \cdot \top \cdot S \cdot \top \subseteq \top \cdot S \cdot \top \cdot R \cdot \top$$

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

Relational Operators

identity relation	:	1
empty relation	:	0
universal relation	:	$\top$
composition	:	$R \cdot S$
union	:	$R \cup S$
intersection	:	$R \cap S$
converse	:	$R^\smile$
refl. trans. closure	:	$R^\star$
complement	:	R^c

Universal laws of relation algebra

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

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

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

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

$$\mathrm{Rel} \models \top \cdot R \cdot \top \cdot S \cdot \top \subseteq \top \cdot S \cdot \top \cdot R \cdot \top$$

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

Relational Operators

identity relation	:	1
empty relation	:	0
universal relation	:	$\top$
composition	:	$R \cdot S$
union	:	$R \cup S$
intersection	:	$R \cap S$
converse	:	$R^\smile$
refl. trans. closure	:	$R^\star$
complement	:	R^c

Universal laws of relation algebra

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

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

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

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

$$\mathrm{Rel} \models \top \cdot R \cdot \top \cdot S \cdot \top \subseteq \top \cdot S \cdot \top \cdot R \cdot \top$$

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

Simple and boring

Relational Operators

identity relation	: 1
empty relation	: 0
universal relation	: $\top$
composition	: $R \cdot S$
union	: $R \cup S$
intersection	: $R \cap S$
converse	: $R^\smile$
refl. trans. closure	: $R^\star$
complement	: R^c

Universal laws of relation algebra

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

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

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

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

$$\mathrm{Rel} \models \top \cdot R \cdot \top \cdot S \cdot \top \subseteq \top \cdot S \cdot \top \cdot R \cdot \top$$

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

Relational Operators

identity relation	:	1
empty relation	:	0
universal relation	:	$\top$
composition	:	$R \cdot S$
union	:	$R \cup S$
intersection	:	$R \cap S$
converse	:	$R^\smile$
refl. trans. closure	:	$R^\star$
complement	:	R^c

Simple and boring : **could we have a computer do it?**

Universal laws of relation algebra

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

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

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

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

$$\mathrm{Rel} \models \top \cdot R \cdot \top \cdot S \cdot \top \subseteq \top \cdot S \cdot \top \cdot R \cdot \top$$

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

Relational Operators

identity relation	:	1
empty relation	:	0
universal relation	:	$\top$
composition	:	$R \cdot S$
union	:	$R \cup S$
intersection	:	$R \cap S$
converse	:	$R^\smile$
refl. trans. closure	:	$R^\star$
complement	:	R^c

Simple and boring : **could we have a computer do it?**

$\hookrightarrow$ **No!** (Undecidable)

Universal laws of relation algebra

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

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

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

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

$$\mathrm{Rel} \models \top \cdot R \cdot \top \cdot S \cdot \top \subseteq \top \cdot S \cdot \top \cdot R \cdot \top$$

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

Relational Operators

identity relation	:	1
empty relation	:	0
universal relation	:	$\top$
composition	:	$R \cdot S$
union	:	$R \cup S$
intersection	:	$R \cap S$
converse	:	$R^\smile$
refl. trans. closure	:	$R^\star$
~~complement~~	~~:~~	~~R^c~~

Simple and boring : **could we have a computer do it?**

$\hookrightarrow$ Positive Relation Algebra

Outline

I. Introduction
- Motivation & Context
- Kleene Algebra
- Extensions

II. Kleene Allegory
- Terms and graphs
- Petri automata
- Comparing automata

III. Kleene Theorems
- Kleene theorem for simple Petri automata
- Kleene theorem for general Petri automata

IV. Overview and outlook

Outline

Regular expressions & languages

Let $\Sigma = \{a, b, \dots\}$ be a finite alphabet.

Regular expressions

$$e, f \in \mathrm{Reg}\langle\Sigma\rangle ::= 0 \mid 1 \mid a \mid e \cdot f \mid e \cup f \mid e^\star$$

Regular expressions & languages

Let $\Sigma = \{a, b, \dots\}$ be a finite alphabet.

Regular expressions

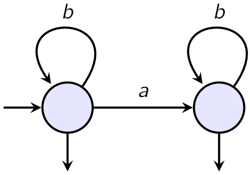
$$e, f \in \mathrm{Reg}\langle\Sigma\rangle ::= 0 \mid 1 \mid a \mid e \cdot f \mid e \cup f \mid e^\star$$

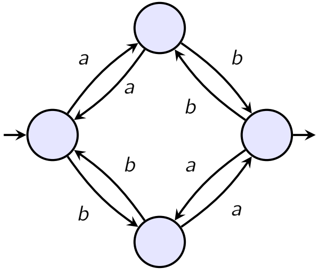
Examples

$$\mathcal{L}\left(a \cdot (b \cup c)\right) = \{ab, ac\} \qquad \mathcal{L}\left(a \cdot 0\right) = \emptyset$$

$$\mathcal{L}\left(a^\star\right) = \{\varepsilon, a, aa, \dots\} = \{a^n \mid n \in \mathbb{N}\}$$

Automata

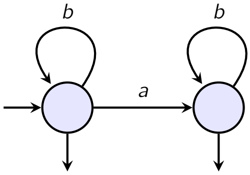


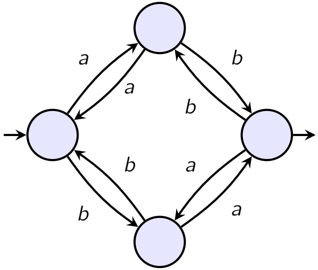


Automata

Theorem

Automata equivalence is decidable.





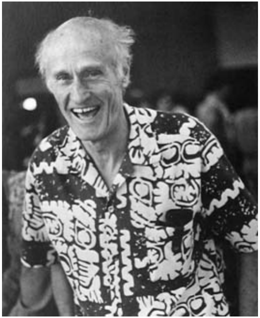
Automata

Theorem

Automata equivalence is decidable.

Kleene Theorem

A language is regular if and only if it is recognised by an automaton.



Stephen Cole Kleene

Automata

Theorem

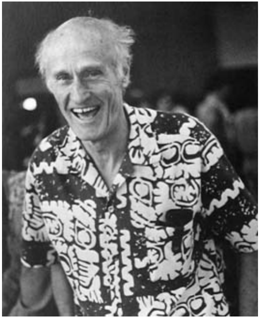
Automata equivalence is decidable.

Kleene Theorem

A language is regular if and only if it is recognised by an automaton.

Corollary

There is an algorithm testing if two regular expressions e and f represent the same language.



Stephen Cole Kleene

Kleene Algebra

A Kleene algebra is an algebraic structure $\langle K, \cup, \cdot, ^\star, 0, 1\rangle$ such that

- $\langle K, \cup, \cdot, 0, 1\rangle$ is an idempotent semiring.
- The star satisfies the following laws:

$$1 \cup a \cdot a^\star \leqslant a^\star$$
$$b \cup a \cdot x \leqslant x \Rightarrow a^\star \cdot b \leqslant x$$

(+ symmetric)

John Horton Conway

Examples:

Algebras of languages: the set of languages over some finite alphabet A.

Algebras of relations: the set of binary relations over some set O.

Equivalence of expressions

Everything is equivalent

$$\forall O, \forall a, b, c, \ldots \in \mathcal{P}(O \times O), e = f?$$

- $\mathrm{Rel} \models e = f$: universal law of relational Kleene algebra.

Equivalence of expressions

Everything is equivalent

$$\forall O, \forall a, b, c, ... \in \mathcal{P}(O \times O), e = f?$$

- $\mathrm{Rel} \models e = f$: universal law of relational Kleene algebra.

$$\mathrm{Rel} \models e = f \iff \mathcal{L}(e) = \mathcal{L}(f)$$

Equivalence of expressions

Everything is equivalent

$$\forall A, \forall a, b, c, ... \in \mathcal{P}\left(A^\star\right), e = f?$$

- $\mathrm{Rel} \models e = f$: universal law of relational Kleene algebra.
- $\mathrm{Lang} \models e = f$: universal law of language Kleene algebra.

$$\mathrm{Rel} \models e = f \Longleftrightarrow \mathcal{L}(e) = \mathcal{L}(f) \Longleftrightarrow \mathrm{Lang} \models e = f$$

Equivalence of expressions

Everything is equivalent

$$\forall K, \forall a, b, c, \ldots \in K, e = f?$$

- $\mathrm{Rel} \models e = f$: universal law of relational Kleene algebra.
- $\mathrm{Lang} \models e = f$: universal law of language Kleene algebra.
- $\mathrm{KA} \vdash e = f$: universal law of Kleene algebra.
  (i.e. logical consequence of the axioms of KA).

$$\mathrm{KA} \vdash e = f$$

$$\Updownarrow$$

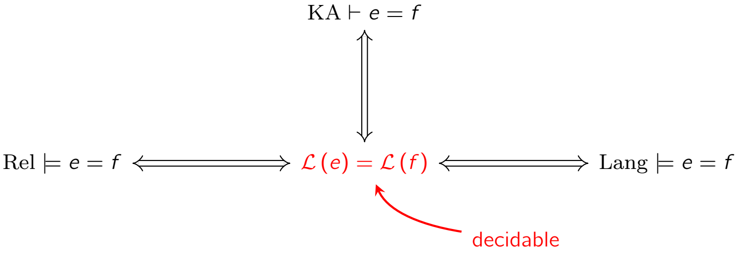
$$\mathrm{Rel} \models e = f \iff \mathcal{L}(e) = \mathcal{L}(f) \iff \mathrm{Lang} \models e = f$$

Equivalence of expressions

Everything is equivalent, and decidable

$$\forall K, \forall a, b, c, \ldots \in K, e = f?$$

- $\mathrm{Rel} \models e = f$: universal law of relational Kleene algebra.
- $\mathrm{Lang} \models e = f$: universal law of language Kleene algebra.
- $\mathrm{KA} \vdash e = f$: universal law of Kleene algebra.
  (i.e. logical consequence of the axioms of KA).



Extensions of Kleene Algebra

Relational Operators

identity relation	:	1	✓
empty relation	:	0	✓
universal relation	:	$\top$	
composition	:	$R \cdot S$	✓
union	:	$R \cup S$	✓
intersection	:	$R \cap S$	
converse	:	$R^{\smile}$	
refl. trans. closure	:	$R^{\star}$	✓
complement	:	R^{c}	

Extensions of Kleene Algebra

Relational Operators

identity relation	:	1	✓
empty relation	:	0	✓
universal relation	:	$\top$	
composition	:	$R \cdot S$	✓
union	:	$R \cup S$	✓
intersection	:	$R \cap S$	
converse	:	$R^{\smile}$	
refl. trans. closure	:	$R^{\star}$	✓
complement	:	R^{c}	✗

Extensions of Kleene Algebra

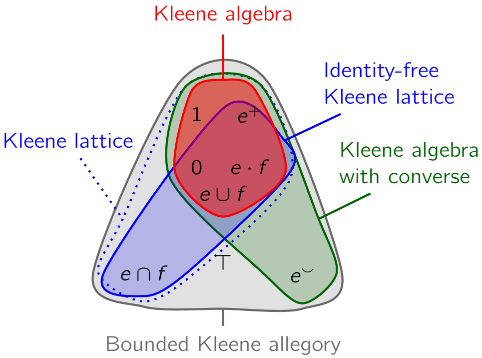
Relational Operators

identity relation	:	1	✓
empty relation	:	0	✓
universal relation	:	$\top$	?
composition	:	$R \cdot S$	✓
union	:	$R \cup S$	✓
intersection	:	$R \cap S$	?
converse	:	$R^{\smile}$	?
refl. trans. closure	:	$R^{\star}$	✓
complement	:	R^{c}	✗

Extensions of Kleene Algebra

Relational Operators

identity relation	:	1	✓
empty relation	:	0	✓
universal relation	:	$\top$	?
composition	:	$R \cdot S$	✓
union	:	$R \cup S$	✓
intersection	:	$R \cap S$	?
converse	:	$R^{\smile}$	?
refl. trans. closure	:	$R^{\star}$	✓
complement	:	R^{c}	✗



Equivalence of expressions

Regular expressions e, f.

$$\mathrm{KA} \vdash e = f$$

$$\Updownarrow$$

$$\mathrm{Rel} \models e = f \iff \mathcal{L}(e) = \mathcal{L}(f) \iff \mathrm{Lang} \models e = f$$

decidable

Equivalence of expressions

Richer expressions e, f.

$$\mathrm{KA} \vdash e = f$$

$$\Updownarrow$$

$$\mathrm{Rel} \models e = f \iff {}^{\blacktriangleleft}\mathcal{G}(e) = {}^{\blacktriangleleft}\mathcal{G}(f) \iff \mathrm{Lang} \models e = f$$

decidable ?

Outline

Kleene Allegories

$$e, f \in \mathrm{AReg}\langle\Sigma\rangle ::= 0 \mid 1 \mid a \mid e \cdot f \mid e \cap f \mid e \cup f \mid e^{\star} \mid e^{\smile} \mid \top$$

Kleene Allegories

$$e, f \in \mathrm{AReg}\langle\Sigma\rangle ::= 0 \mid 1 \mid a \mid e \cdot f \mid e \cap f \mid e \cup f \mid e^{\star} \mid e^{\smile} \mid \top$$

$$\mathrm{Rel} \models e = f \overset{?}{\not\Leftrightarrow} \mathcal{L}(e) = \mathcal{L}(f)$$

Kleene Allegories

$$e, f \in \mathrm{AReg}\langle\Sigma\rangle ::= 0 \mid 1 \mid a \mid e \cdot f \mid e \cap f \mid e \cup f \mid e^\star \mid e^\smile \mid \top$$

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

Counterexample

$\mathcal{L}(a \cap b) = \mathcal{L}(0)_{=\emptyset}$	$\mathcal{L}(a) = \mathcal{L}(a^\smile)_{=\{a\}}$	$\mathcal{L}(\top a \top b \top) \neq \mathcal{L}(\top b \top a \top)$
$\mathrm{Rel} \not\models a \cap b = 0$	$\mathrm{Rel} \not\models a = a^\smile$	$\mathrm{Rel} \models \top a \top b \top = \top b \top a \top$

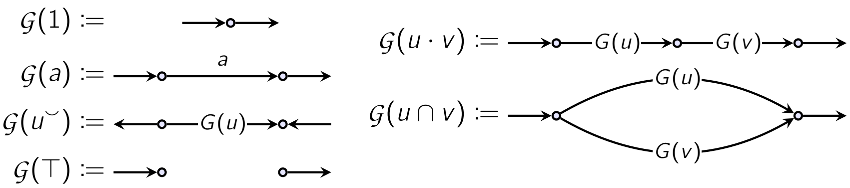
A different approach is needed.

Graphs/Ground terms

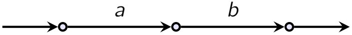
$$u, v \in \mathsf{W}_\Sigma ::= 1 \mid a \mid u \cdot v \mid u \cap v \mid u^{\smile} \mid \top$$

Graphs/Ground terms

$$u, v \in \mathsf{W}_\Sigma ::= 1 \mid a \mid u \cdot v \mid u \cap v \mid u^{\smile} \mid \top$$

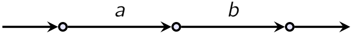


Examples

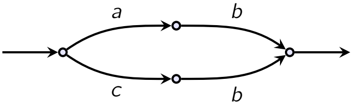
$\mathcal{G}(a \cdot b)$: 

Examples

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

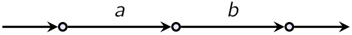


$\mathcal{G}((a \cdot b) \cap (c \cdot b))$:

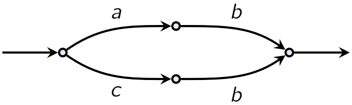


Examples

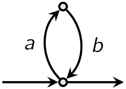
$\mathcal{G}(a \cdot b)$:



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

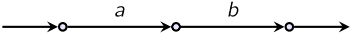


$\mathcal{G}((a \cdot b) \cap 1)$:

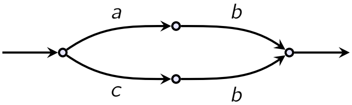


Examples

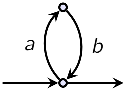
$\mathcal{G}(a \cdot b)$:



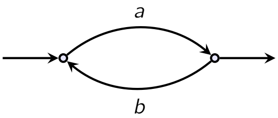
$\mathcal{G}((a \cdot b) \cap (c \cdot b))$:



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

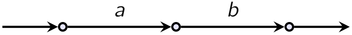


$\mathcal{G}(a \cap b^{\smile})$:

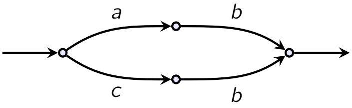


Examples

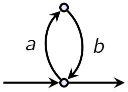
$\mathcal{G}(a \cdot b)$:



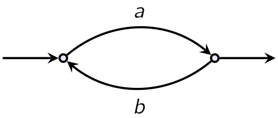
$\mathcal{G}((a \cdot b) \cap (c \cdot b))$:



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



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

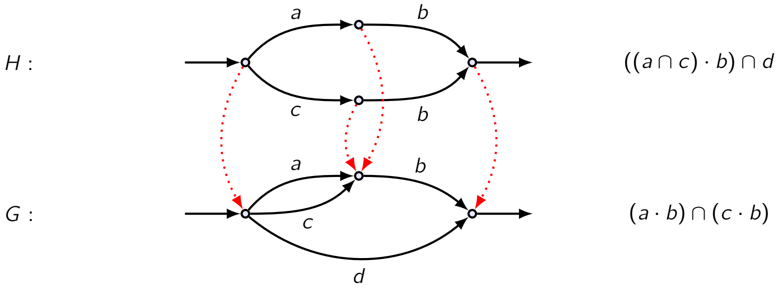


$\mathcal{G}(\top a \top b \top)$:

Preorder

Preorder on graphs

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



Characterization theorem

$$u, v \in \mathsf{W}_\Sigma ::= 1 \mid a \mid u \cdot v \mid u \cap v \mid u^\smile \mid \top$$

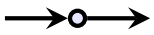
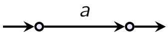
Theorem

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

- Freyd & Scedrov, **Categories, Allegories**, *1990* .
- Andréka & Bredikhin, **The equational theory of union-free algebras of relations**, *1995*

Graph languages

$$e, f \in \mathrm{AReg}\langle\Sigma\rangle ::= 0 \mid 1 \mid a \mid e \cdot f \mid e \cap f \mid e \cup f \mid e^{\star} \mid e^{\smile} \mid \top$$

$\mathcal{G}\left(1\right) := \Big\{$  $\Big\}$ $\mathcal{G}\left(a\right) := \Big\{$ 

$\Big\}$ $\mathcal{G}\left(\top\right) := \Big\{$  $\Big\}$

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

$$\mathcal{G}\left(e \cdot f\right) := \left\{ G \cdot G' \mid G \in \mathcal{G}\left(e\right) \text{ and } G' \in \mathcal{G}\left(f\right) \right\}$$

$$\mathcal{G}\left(e \cap f\right) := \left\{ G \cap G' \mid G \in \mathcal{G}\left(e\right) \text{ and } G' \in \mathcal{G}\left(f\right) \right\}$$

$$\mathcal{G}\left(0\right) := \emptyset \qquad \mathcal{G}\left(e \cup f\right) := \mathcal{G}\left(e\right) \cup \mathcal{G}\left(f\right) \qquad \mathcal{G}\left(e^{\star}\right) := \bigcup_{n \in \mathbb{N}} \mathcal{G}\left(e\right)^{n}.$$

Characterization theorem

${}^{\blacktriangleleft}S$ is the downwards closure of S with respect to $\blacktriangleleft$.
${}^{\blacktriangleleft}S := \{G \mid \exists H \in S : G \blacktriangleleft H\}$.

Theorem

$e, f \in \mathrm{AReg}\langle\Sigma\rangle$,

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

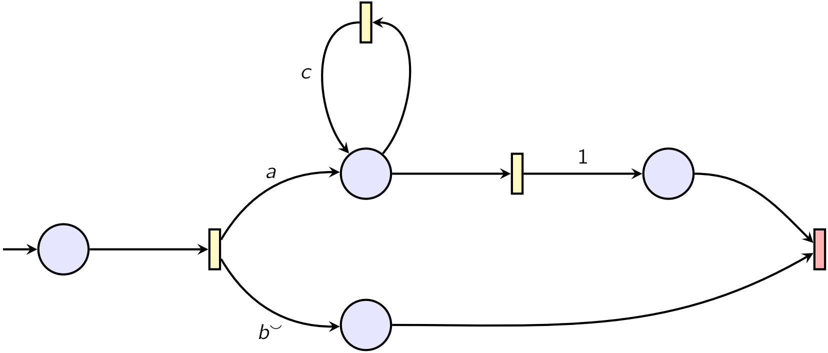
B. & Pous, **Petri automata for Kleene Allegories**, *LICS'15*

Follows easily from:

Andréka, Mikulás & Németi, **The equational theory of Kleene lattices**, *TCS'11*

Outline

Petri automata



Set of labels: $\Sigma' := \Sigma \cup \left\{ a^{\smile} \mid a \in \Sigma \right\} \cup \{1, \top\}$.

Language generated vs. Language recognised

Let $\mathcal{A}$ be a finite state word automaton.

The language of $\mathcal{A}$ is:

Language generated vs. Language recognised

Let $\mathcal{A}$ be a finite state word automaton.

The language of $\mathcal{A}$ is:

- the set of words **accepted** by $\mathcal{A}$; **recognised**

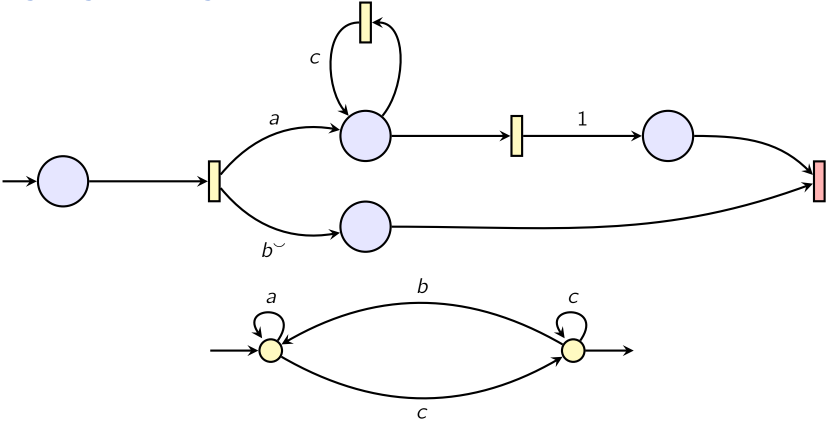
Language generated vs. Language recognised

Let $\mathcal{A}$ be a finite state word automaton.

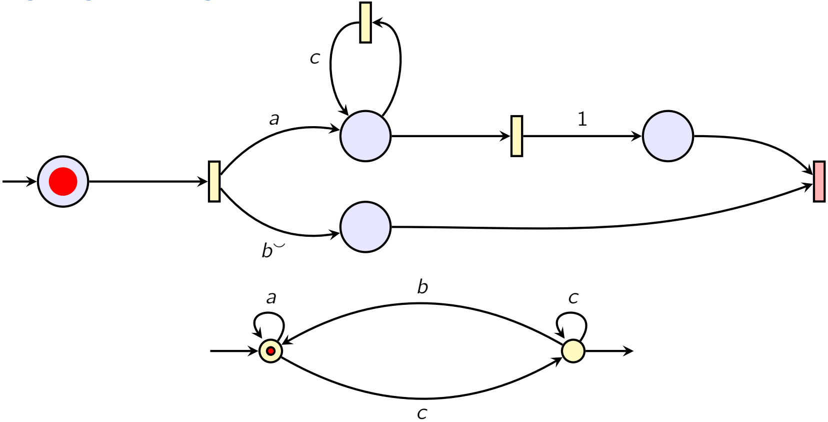
The language of $\mathcal{A}$ is:

- the set of words **accepted** by $\mathcal{A}$; **recognised**
- the set of words labelling **accepting paths** in $\mathcal{A}$. **generated**

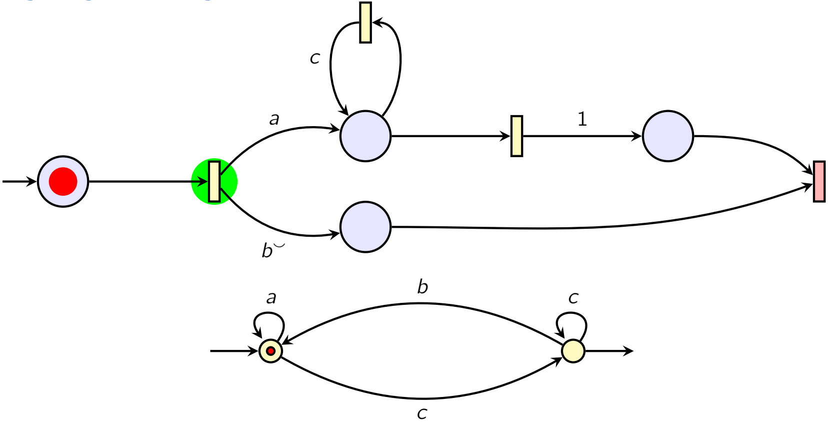
Language recognised by an automaton



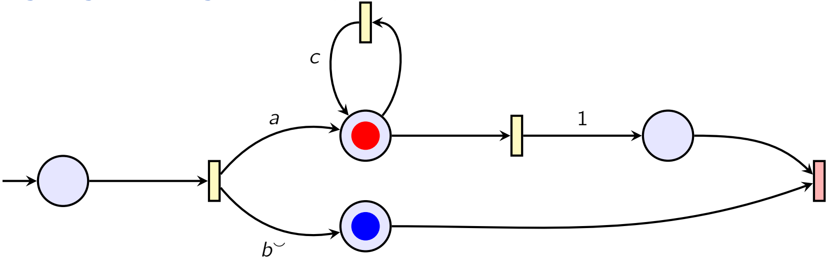
Language recognised by an automaton

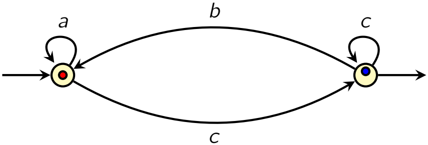


Language recognised by an automaton

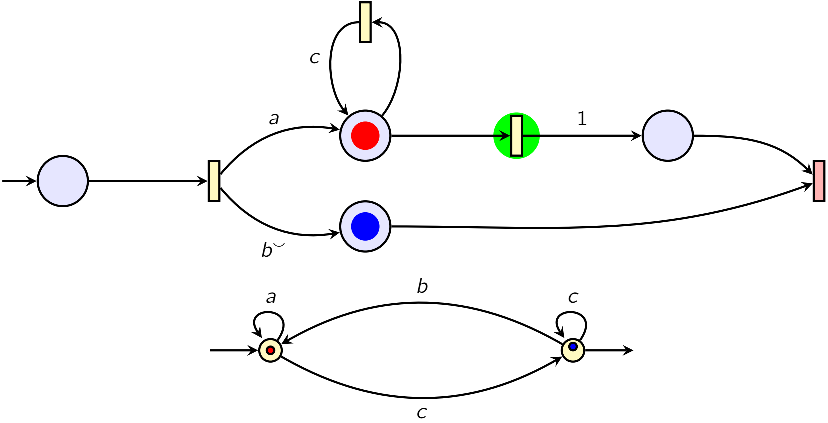


Language recognised by an automaton

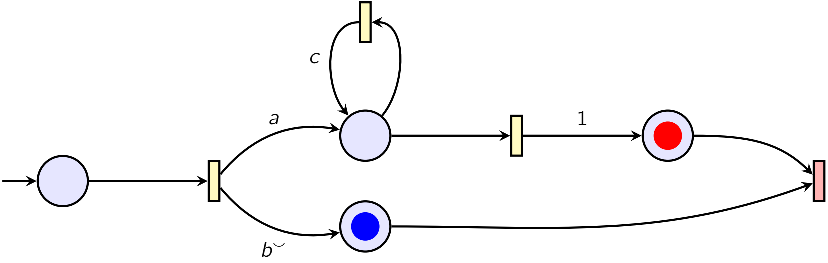


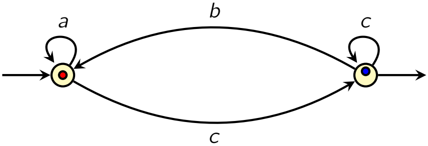


Language recognised by an automaton

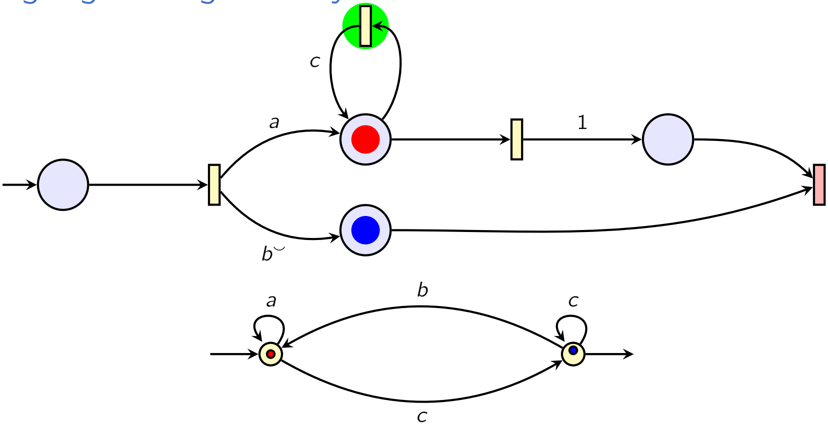


Language recognised by an automaton

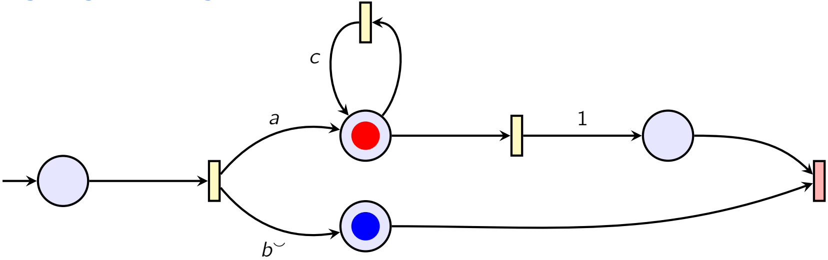


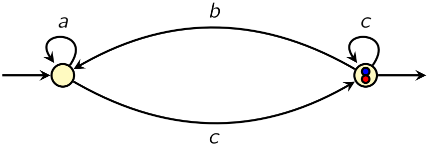


Language recognised by an automaton

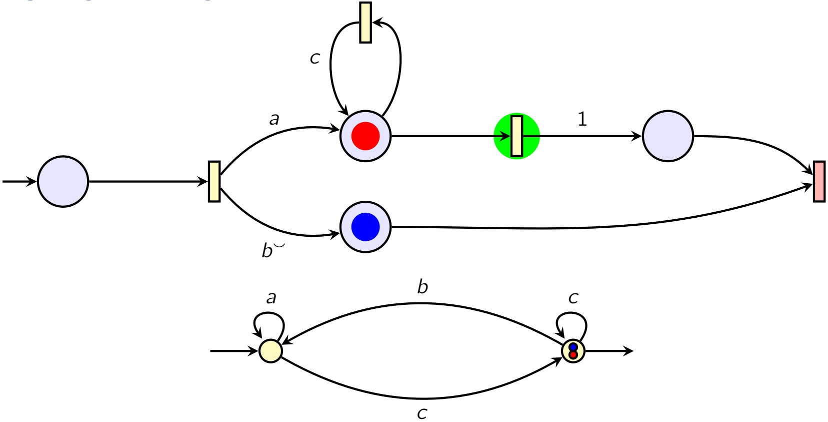


Language recognised by an automaton

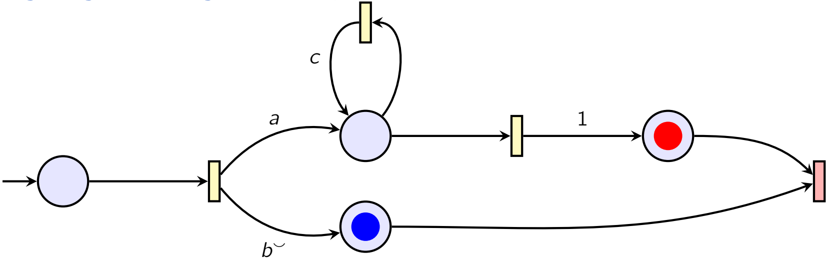


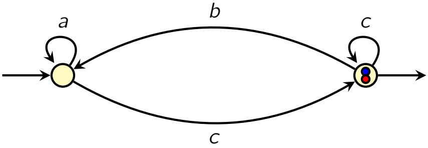


Language recognised by an automaton

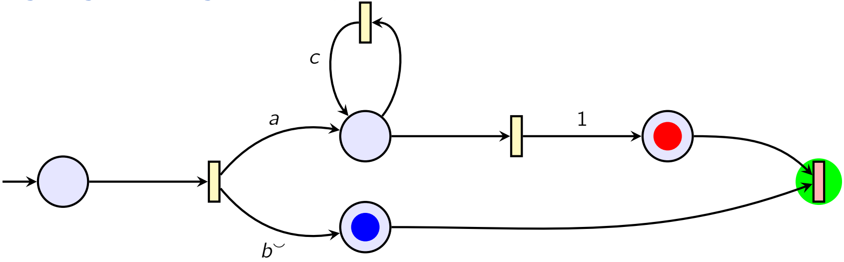


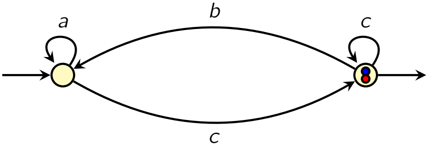
Language recognised by an automaton



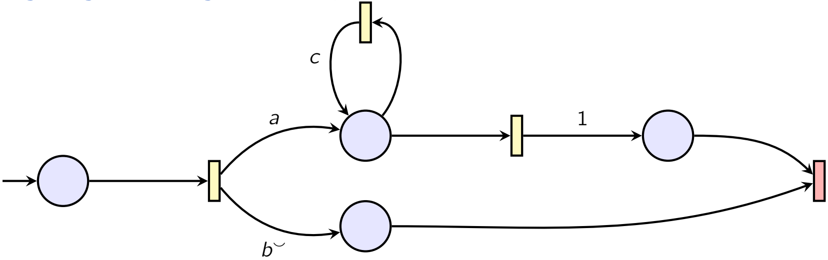


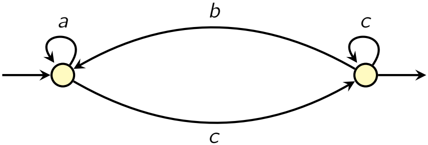
Language recognised by an automaton





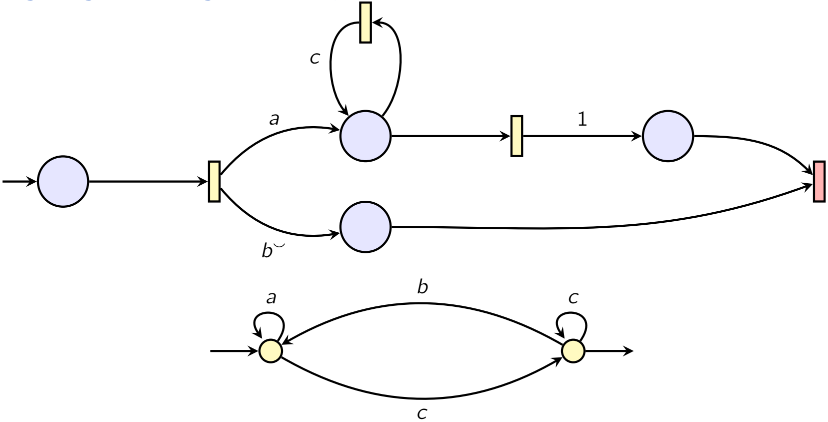
Language recognised by an automaton





Success!

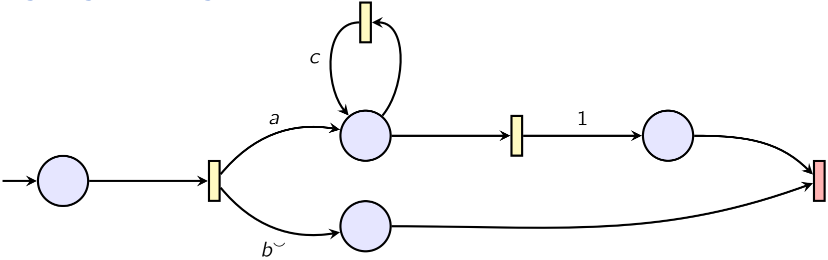
Language recognised by an automaton

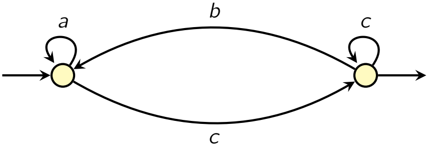


Language of $\mathcal{A}$

$$\mathcal{L}(\mathcal{A}) = \{G \mid \mathcal{A} \text{ accepts } G\}$$

Language recognised by an automaton





Language of $\mathcal{A}$

$$\mathcal{L}(\mathcal{A}) = \{ G \mid \mathcal{A} \text{ accepts } G \}$$

Here: $\mathcal{L}(\mathcal{A}) = {}^{\blacktriangleleft}\mathcal{G}\left((a \cdot c^\star) \cap b^\smile\right)$.

Petri automata for Kleene allegories

Correctness

For any $e \in \mathrm{AReg}\langle\Sigma\rangle$,

$$e$$

B. & Pous, **Petri automata for Kleene Allegories**, *LICS'15*

Petri automata for Kleene allegories

Correctness

For any $e \in \mathrm{AReg}\langle\Sigma\rangle$,

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

B. & Pous, **Petri automata for Kleene Allegories**, *LICS'15*

Petri automata for Kleene allegories

Correctness

For any $e \in \mathrm{AReg}\langle\Sigma\rangle$,

$$\mathcal{L}(\mathcal{A}(e))$$

B. & Pous, **Petri automata for Kleene Allegories**, *LICS'15*

Petri automata for Kleene allegories

Correctness

For any $e \in \mathrm{AReg}\langle\Sigma\rangle$,

$$\mathcal{L}(\mathcal{A}(e)) = {}^{\blacktriangleleft}\mathcal{G}(e).$$

B. & Pous, **Petri automata for Kleene Allegories**, *LICS'15*

Petri automata for Kleene allegories

Correctness

For any $e \in \text{AReg}\langle\Sigma\rangle$,

$$\mathcal{L}(\mathcal{A}(e)) = {}^{\blacktriangleleft}\mathcal{G}(e).$$

B. & Pous, **Petri automata for Kleene Allegories**, *LICS'15*

So far:

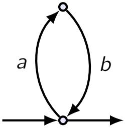
$e, f \in \text{AReg}\langle\Sigma\rangle$

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

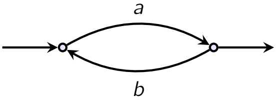
Outline

Restriction: identity-free Kleene lattice terms

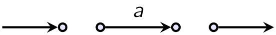
$\mathcal{G}\left((a \cdot b) \cap 1\right)$:



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

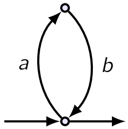


$\mathcal{G}\left(\top a \top\right)$:

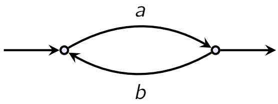


Restriction: identity-free Kleene lattice terms

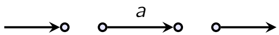
$\mathcal{G}((a \cdot b) \cap 1)$:



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



$\mathcal{G}(\top a \top)$:



Identity-free Kleene Lattice

$$u, v \in \mathrm{W}^{-}_{\Sigma} ::= 1 \mid a \mid u \cdot v \mid u \cap v \mid u^{\smile} \mid \top$$

$$e, f \in \mathrm{GReg}\langle\Sigma\rangle ::= 0 \mid 1 \mid a \mid e \cdot f \mid e \cap f \mid e \cup f \mid e^{+} \mid e^{\smile} \mid \top$$

$\hookrightarrow$ Series-Parallel graphs & simple automata

Decision procedure

$e, f \in \mathrm{GReg}\langle\Sigma\rangle$

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

Problem:

How to compare two simple Petri automata?

Decision procedure

$e, f \in \mathsf{GReg}\langle\Sigma\rangle$

$$\mathrm{Rel} \models e \subseteq f \iff {}^{\blacktriangleleft}\mathcal{G}(e) \subseteq {}^{\blacktriangleleft}\mathcal{G}(f) \iff \mathcal{L}(\mathcal{A}(e)) \subseteq \mathcal{L}(\mathcal{A}(f)).$$

Problem:

How to compare two simple Petri automata?

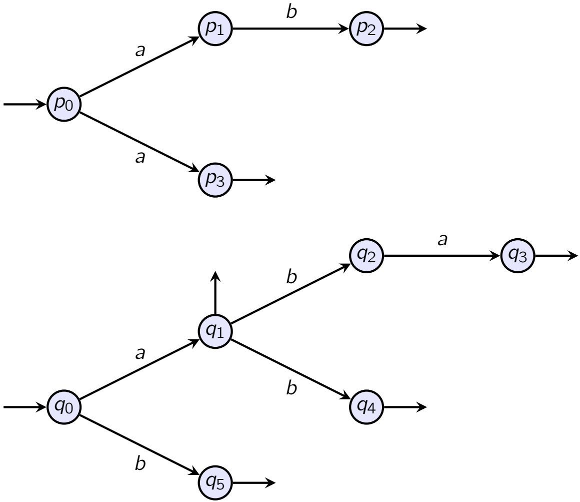
$\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$ if and only if there is a simulation relation

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

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

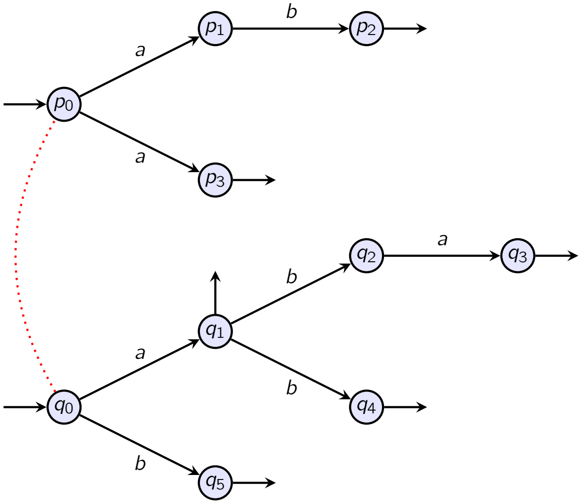
Simulations - non-deterministic finite automata

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



Simulations - non-deterministic finite automata

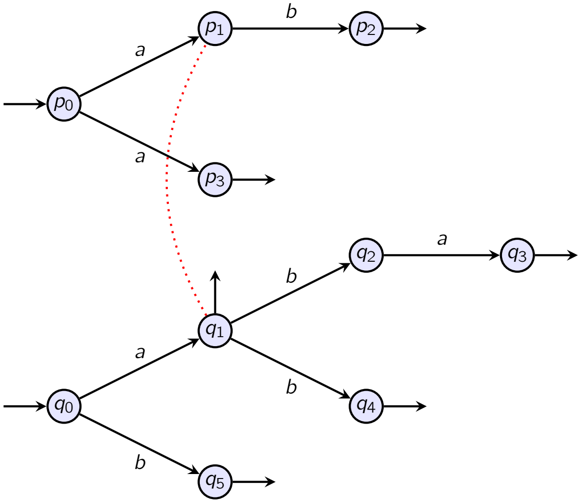
$$\preccurlyeq \subseteq Q_1 \times \mathcal{P}(Q_2)$$



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

Simulations - non-deterministic finite automata

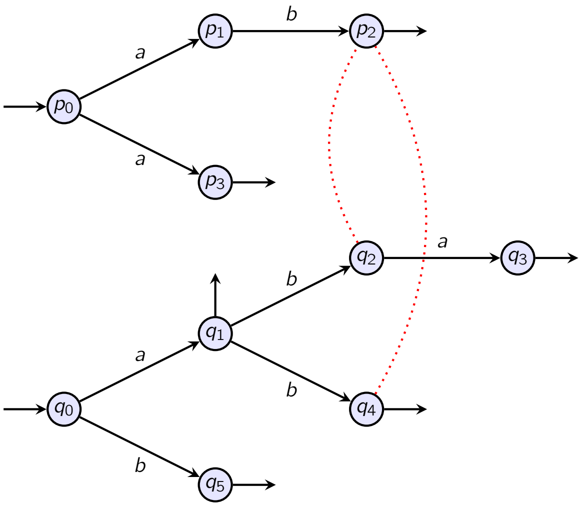
$$\preccurlyeq \subseteq Q_1 \times \mathcal{P}(Q_2)$$



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

Simulations - non-deterministic finite automata

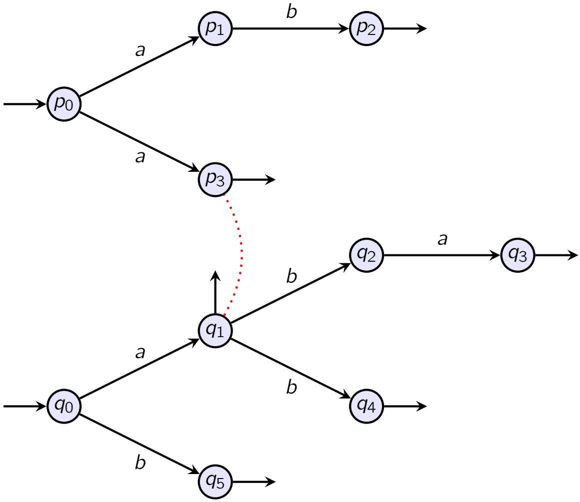
$$\preccurlyeq \subseteq Q_1 \times \mathcal{P}(Q_2)$$



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

Simulations - non-deterministic finite automata

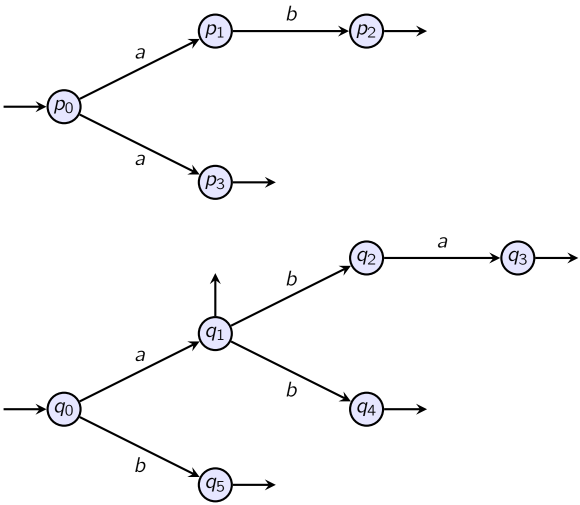
$$\preccurlyeq \subseteq Q_1 \times \mathcal{P}(Q_2)$$



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

Simulations - non-deterministic finite automata

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



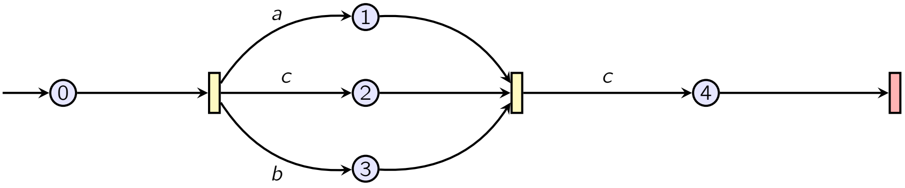
$p_0 \preccurlyeq \{\ q_0\ \}$
$p_1 \preccurlyeq \{\ q_1\ \}$
$p_2 \preccurlyeq \{q_2, q_4\}$
$p_3 \preccurlyeq \{\ q_1\ \}$

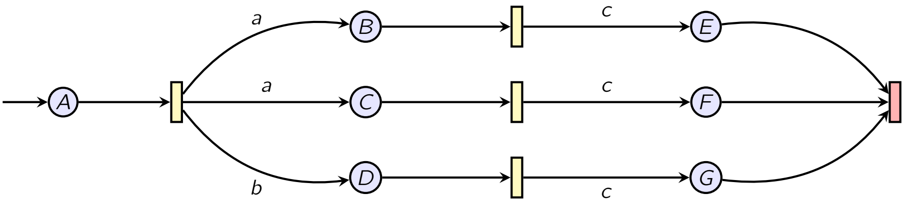
+ condition for accepting states

Simulation - Petri automata

Simulation relation

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

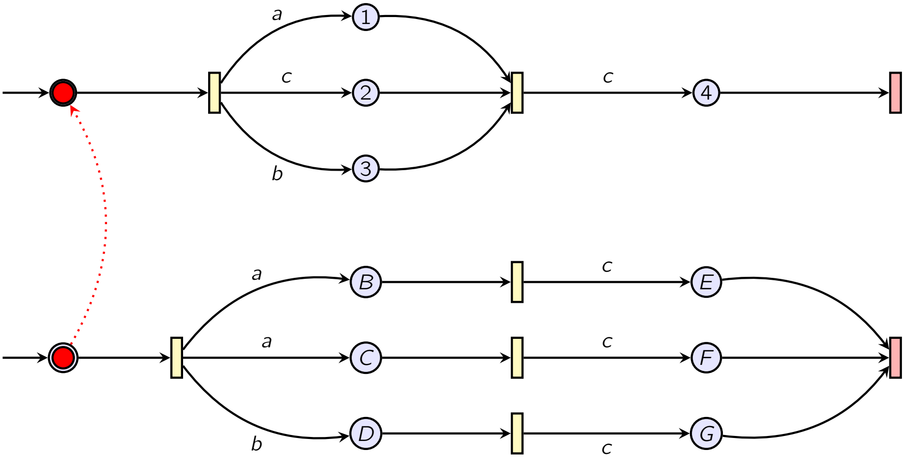




Simulation - Petri automata

Simulation relation

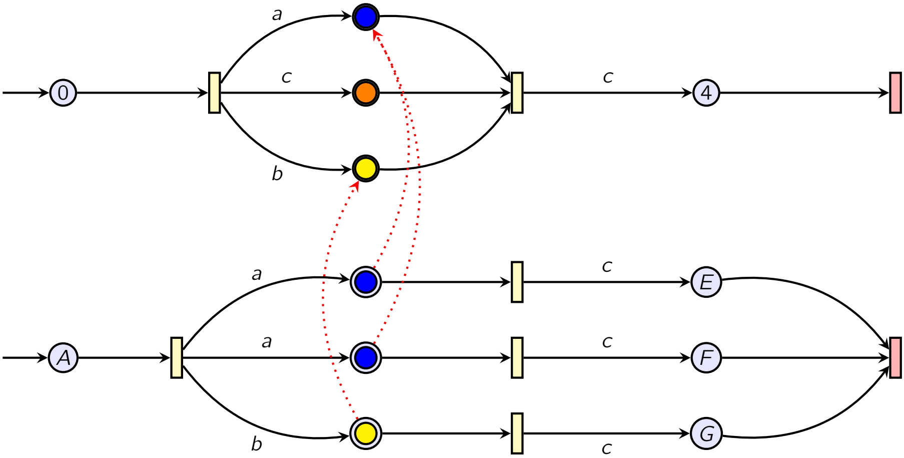
$$\preccurlyeq \subseteq \mathcal{P}(P_1) \times \mathcal{P}(P_2 \rightharpoonup P_1)$$



Simulation - Petri automata

Simulation relation

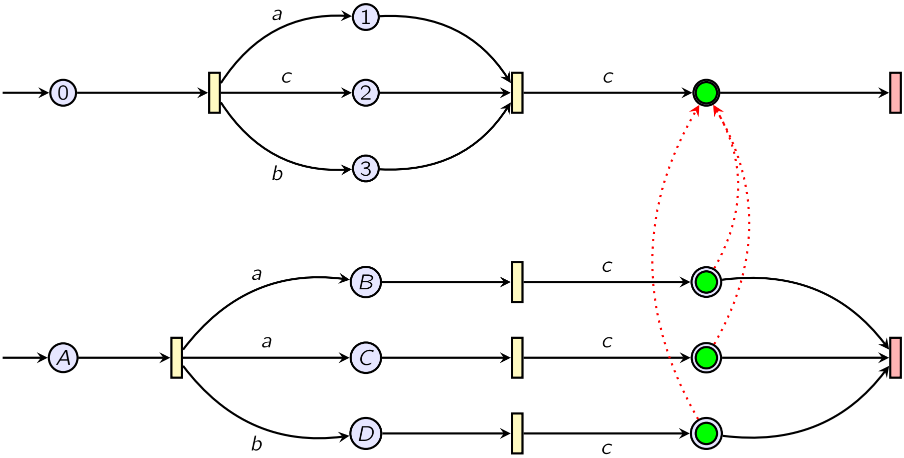
$$\preccurlyeq \subseteq \mathcal{P}(P_1) \times \mathcal{P}(P_2 \rightharpoonup P_1)$$



Simulation - Petri automata

Simulation relation

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



Comparing Petri automata

Theorem

Comparing simple Petri automata is **ExpSpace-complete**.

Proof.

ExpSpace easy: testing for the existence of a simulation can be done in exponential space;

ExpSpace hard: reduction from the universality problem for regular expressions with squaring.

□

B. & Pous, **Petri automata for Kleene Allegories**, *LICS'15*

Meyer & Stockmeyer, **The equivalence problem for regular expressions with squaring requires exponential space**, *SWAT.'72*

Comparing Petri automata

Theorem

Comparing simple Petri automata is **ExpSpace-complete**.

Proof.

ExpSpace easy: testing for the existence of a simulation can be done in exponential space;

ExpSpace hard: reduction from the universality problem for regular expressions with squaring.

□

B. & Pous, **Petri automata for Kleene Allegories**, *LICS'15*

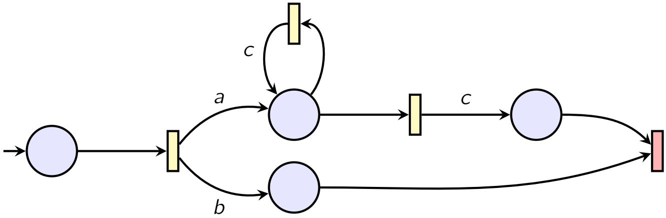
Meyer & Stockmeyer, **The equivalence problem for regular expressions with squaring requires exponential space**, *SWAT.'72*

Corollary

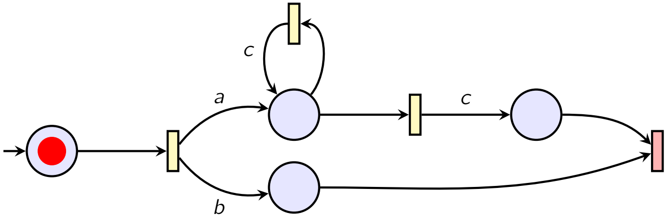
Relational equivalence for Identity-free Kleene lattices is **decidable**.

Outline

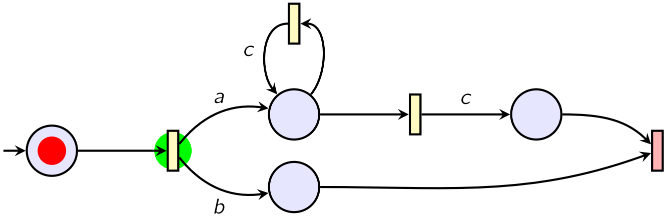
Trace language of an automaton



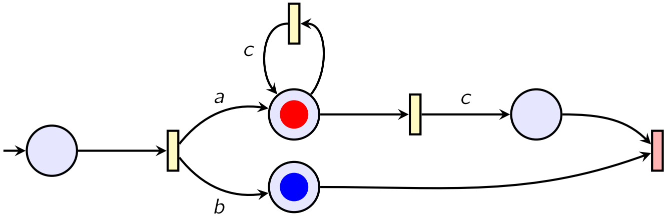
Trace language of an automaton

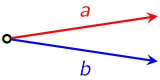


Trace language of an automaton

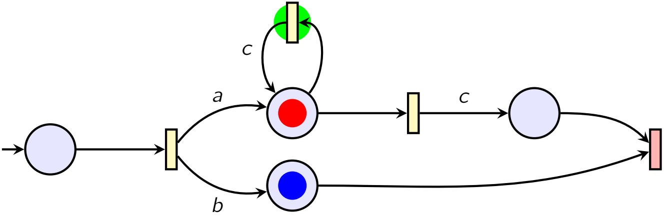


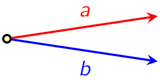
Trace language of an automaton



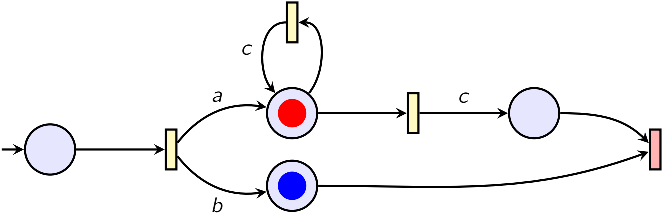


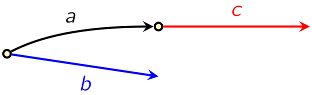
Trace language of an automaton



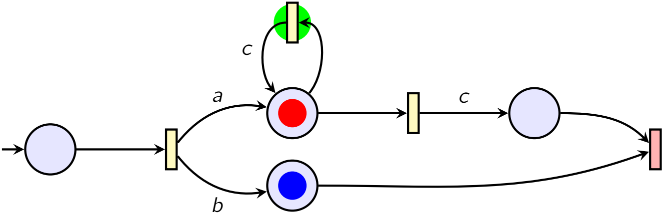


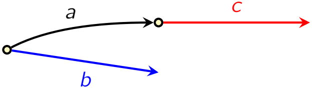
Trace language of an automaton



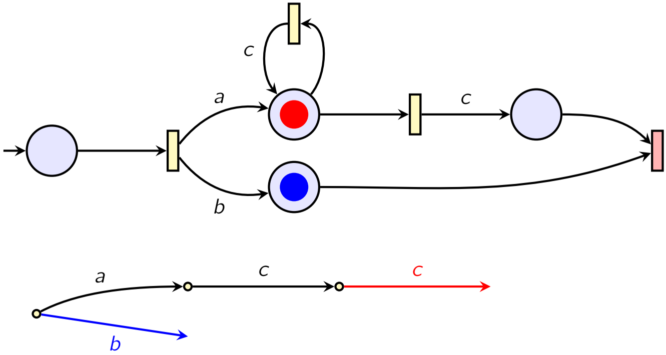


Trace language of an automaton

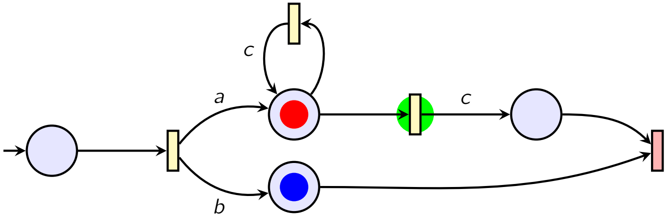


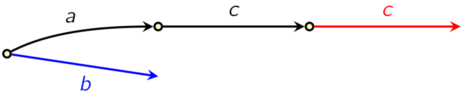


Trace language of an automaton

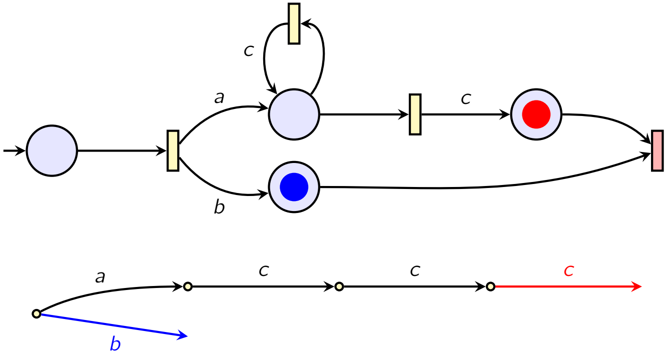


Trace language of an automaton

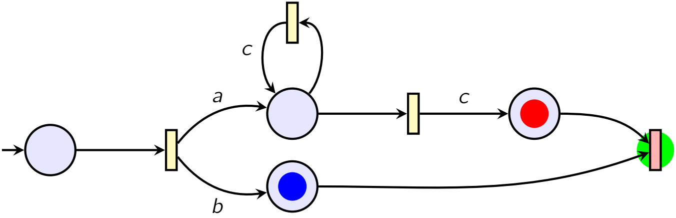


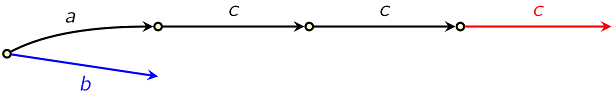


Trace language of an automaton

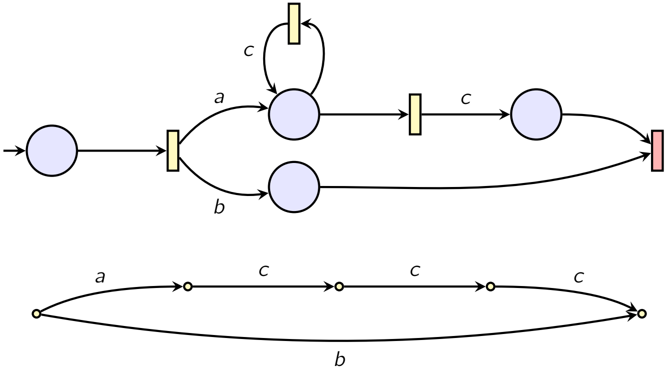


Trace language of an automaton

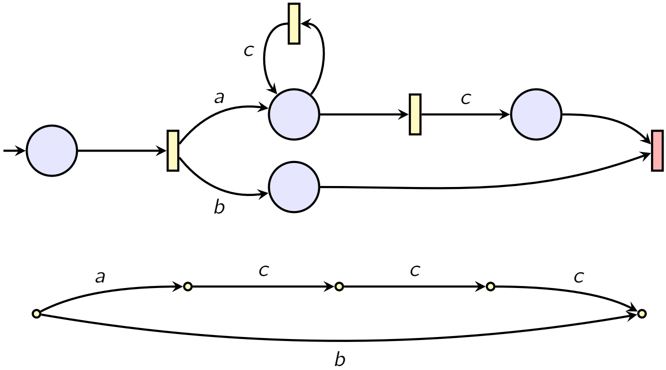




Trace language of an automaton



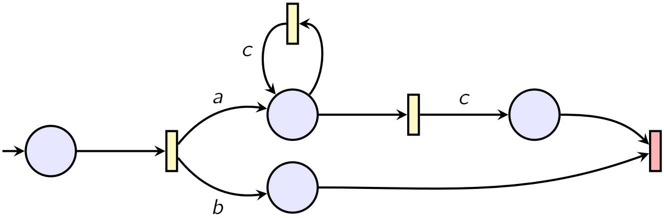
Trace language of an automaton

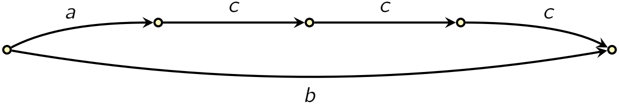


Trace language of $\mathcal{A}$

$$\mathcal{T}r(\mathcal{A}) = \{G \mid G \text{ is the trace of an accepting run in } \mathcal{A}\}$$

Trace language of an automaton





Trace language of $\mathcal{A}$

$$\mathcal{T}r(\mathcal{A}) = \{G \mid G \text{ is the trace of an accepting run in } \mathcal{A}\}$$

Here: $\mathcal{T}r(\mathcal{A}) = \mathcal{G}((a \cdot c^+) \cap b)$.

Traces, languages and expressions

Lemma

If $\mathcal{A}$ is only labelled with Σ,

$$\mathcal{L}(\mathcal{A}) = {}^{\blacktriangleleft}\mathcal{T}r(\mathcal{A}).$$

Fact

$e \in \mathrm{GReg}\langle\Sigma\rangle$,

$$\mathcal{T}r(\mathcal{A}(e)) = \mathcal{G}(e).$$

Kleene Theorem

Definitions

A set of SP graphs $\mathcal{G}$ is:

- regular: if there is an expression e in $\mathrm{GReg}\langle\Sigma\rangle$ such that $\mathcal{G}(e) = \mathcal{G}$;
- recognisable: if there is a simple Petri automaton $\mathcal{A}$ such that $\mathcal{Tr}(\mathcal{A}) = \mathcal{G}$.

Kleene Theorem

Definitions

A set of SP graphs $\mathcal{G}$ is:

- regular if there is an expression e in $\mathrm{GReg}\langle\Sigma\rangle$ such that $\mathcal{G}(e) = \mathcal{G}$;
- recognisable if there is a simple Petri automaton $\mathcal{A}$ such that $\mathcal{Tr}(\mathcal{A}) = \mathcal{G}$.

Theorem

The class of regular sets of graphs coincides with the recognisable sets of graphs.

Kleene Theorem

Definitions

A set of SP graphs $\mathcal{G}$ is:

- regular: if there is an expression e in $\mathrm{GReg}\langle\Sigma\rangle$ such that $\mathcal{G}(e) = \mathcal{G}$;
- recognisable: if there is a simple Petri automaton $\mathcal{A}$ such that $\mathcal{Tr}(\mathcal{A}) = \mathcal{G}$.

Theorem

The class of regular sets of graphs coincides with the recognisable sets of graphs.

Proof.

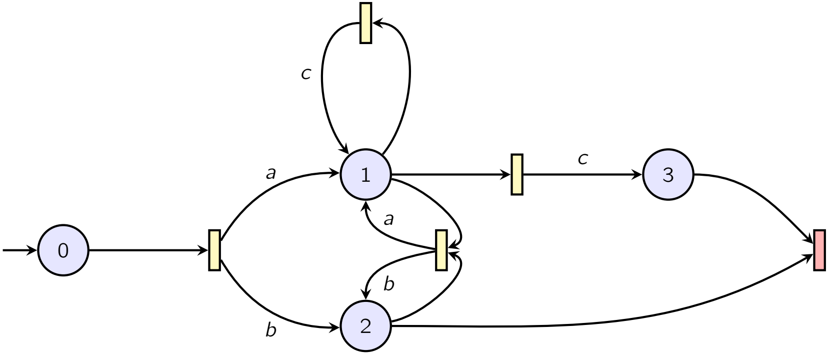
- **Expressions to automata**: inductive construction;

  B. & Pous, **Petri automata for Kleene Allegories**, *LICS'15*

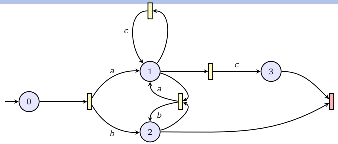
- **Automata to expressions**: next slide.

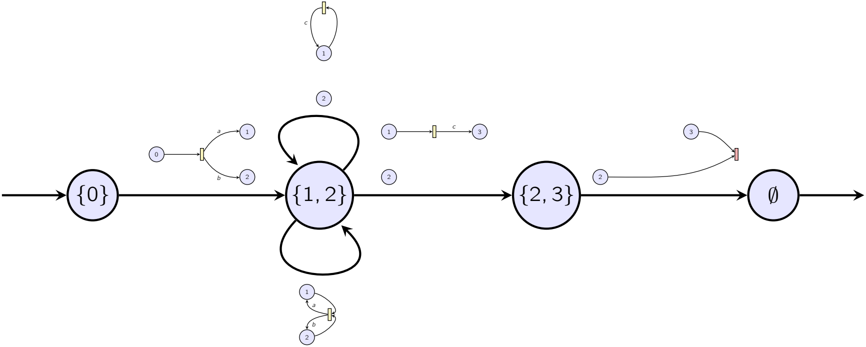
□

From automata to expressions

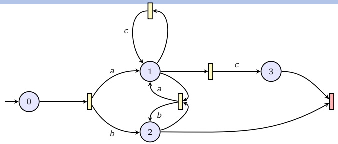


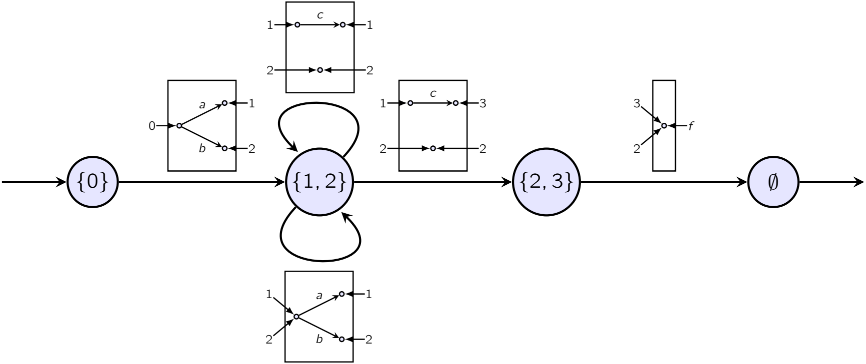
From automata to expressions



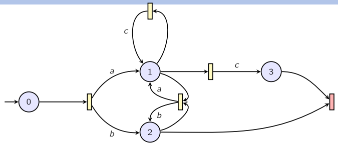


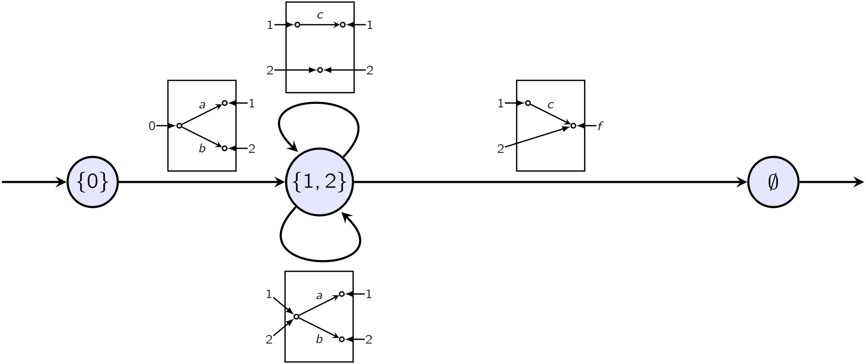
From automata to expressions



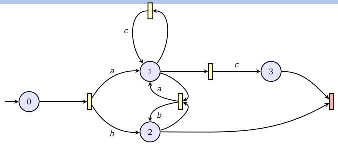


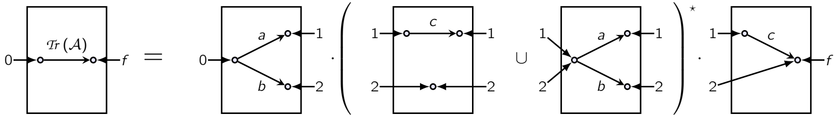
From automata to expressions

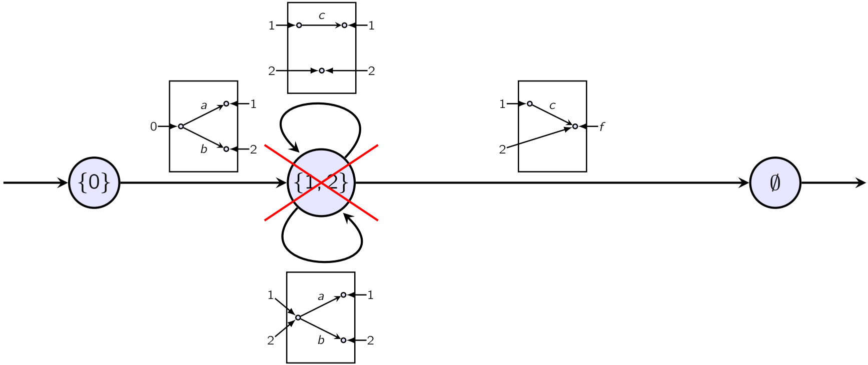




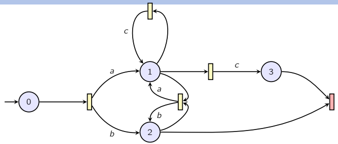
From automata to expressions

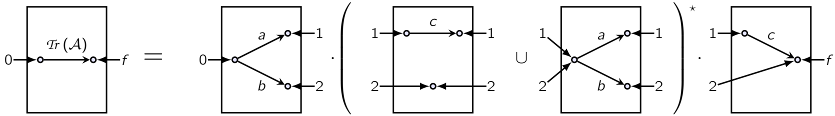




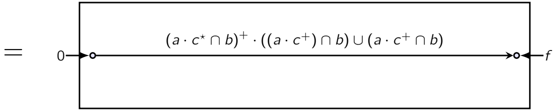


From automata to expressions



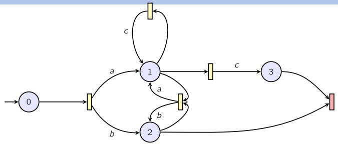


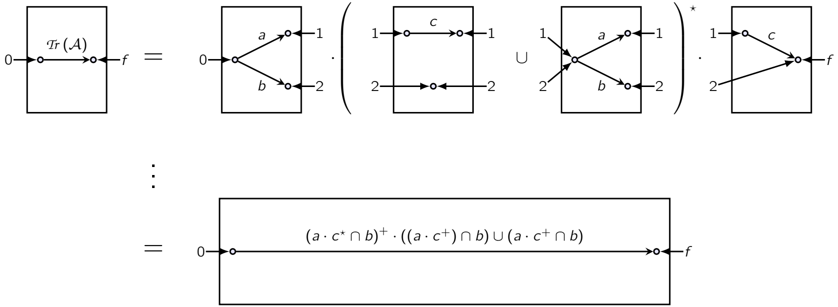
⋮



(With some effort...)

From automata to expressions





Result

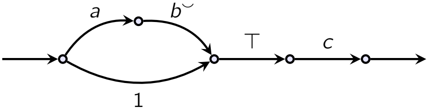
$$\mathcal{E}(\mathcal{A}) = (a \cdot c^{\star} \cap b)^{+} \cdot \left(\left(a \cdot c^{+}\right) \cap b\right) \cup \left(a \cdot c^{+} \cap b\right)$$

$$\mathcal{T}r(\mathcal{A}) = \mathcal{G}(\mathcal{E}(\mathcal{A}))$$

Outline

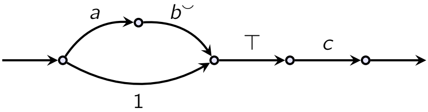
Relationship between $\mathcal{Tr}(\mathcal{A})$ and $\mathcal{L}(\mathcal{A})$

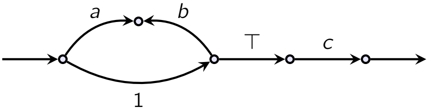
Φ : graphs over $\Sigma' \rightarrow$ graphs over Σ.



Relationship between $\mathcal{Tr}(\mathcal{A})$ and $\mathcal{L}(\mathcal{A})$

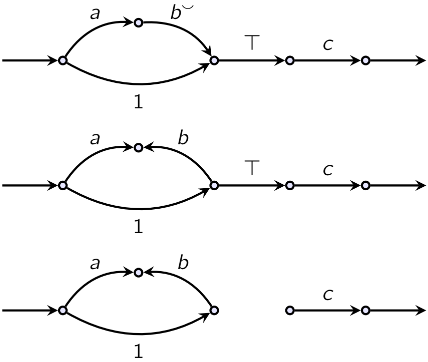
Φ : graphs over $\Sigma' \to$ graphs over Σ.





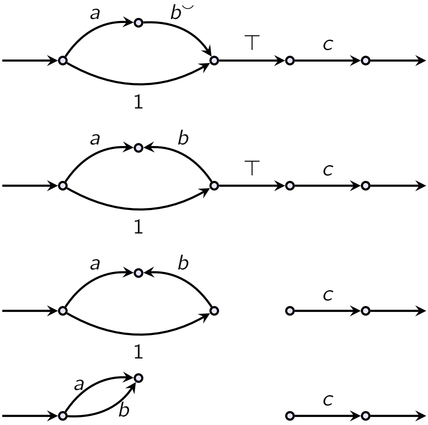
Relationship between $\mathcal{Tr}(\mathcal{A})$ and $\mathcal{L}(\mathcal{A})$

Φ : graphs over $\Sigma' \rightarrow$ graphs over Σ.



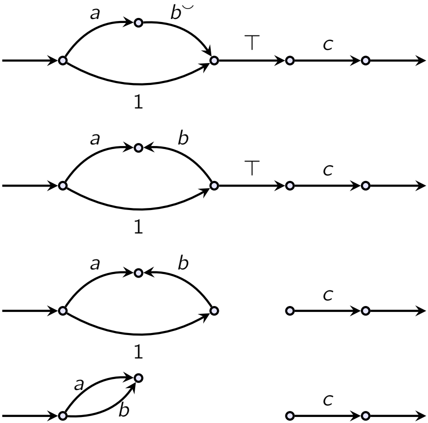
Relationship between $\mathcal{Tr}(\mathcal{A})$ and $\mathcal{L}(\mathcal{A})$

Φ : graphs over $\Sigma' \to$ graphs over Σ.



Relationship between $\mathcal{Tr}(\mathcal{A})$ and $\mathcal{L}(\mathcal{A})$

Φ : graphs over $\Sigma' \rightarrow$ graphs over Σ.



Lemma

$$\mathcal{L}(\mathcal{A}) = {}^{\blacktriangleleft}\Phi\left(\mathcal{Tr}(\mathcal{A})\right).$$

From AReg$\langle\Sigma\rangle$ to GReg$\langle\Sigma'\rangle$

$$e, f \in \mathrm{AReg}\langle\Sigma\rangle ::= 0 \mid 1 \mid a \mid e \cdot f \mid e \cap f \mid e \cup f \mid e^\star \mid e^\smile \mid \top$$
$$e, f \in \mathrm{GReg}\langle\Sigma\rangle ::= 0 \mid a \mid e \cdot f \mid e \cap f \mid e \cup f \mid e^+$$

$$\Sigma' := \Sigma \cup \left\{ a^\smile \mid a \in \Sigma \right\} \cup \{1, \top\}.$$

$$e = (a \cdot b^\star)^\smile \cup \top$$

From AReg⟨Σ⟩ to GReg⟨Σ′⟩

$$e, f \in \mathrm{AReg}\langle\Sigma\rangle ::= 0 \mid 1 \mid a \mid e \cdot f \mid e \cap f \mid e \cup f \mid e^\star \mid e^\smile \mid \top$$
$$e, f \in \mathrm{GReg}\langle\Sigma\rangle ::= 0 \mid a \mid e \cdot f \mid e \cap f \mid e \cup f \mid e^+$$

$$\Sigma' := \Sigma \cup \left\{ a^\smile \mid a \in \Sigma \right\} \cup \{1, \top\}.$$

$$\begin{aligned} e &= (a \cdot b^\star)^\smile \cup \top \\ &\equiv \left(a \cdot \left(1 \cup b^+\right)\right)^\smile \cup \top \end{aligned}$$

From $\mathrm{AReg}\langle\Sigma\rangle$ to $\mathrm{GReg}\langle\Sigma'\rangle$

$$e, f \in \mathrm{AReg}\langle\Sigma\rangle ::= 0 \mid 1 \mid a \mid e \cdot f \mid e \cap f \mid e \cup f \mid e^\star \mid e^\smile \mid \top$$
$$e, f \in \mathrm{GReg}\langle\Sigma\rangle ::= 0 \mid a \mid e \cdot f \mid e \cap f \mid e \cup f \mid e^+$$

$$\Sigma' := \Sigma \cup \left\{a^\smile \mid a \in \Sigma\right\} \cup \{1, \top\}.$$

$$\begin{aligned} e &= (a \cdot b^\star)^\smile \cup \top \\ &\equiv \left(a \cdot \left(1 \cup b^+\right)\right)^\smile \cup \top \\ &\equiv \left(1 \cup \left(b^\smile\right)^+\right) \cdot a^\smile \cup \top \in \mathrm{GReg}\langle\Sigma'\rangle \end{aligned}$$

From $\mathrm{AReg}\langle\Sigma\rangle$ to $\mathrm{GReg}\langle\Sigma'\rangle$

$$e, f \in \mathrm{AReg}\langle\Sigma\rangle ::= 0 \mid 1 \mid a \mid e\cdot f \mid e\cap f \mid e\cup f \mid e^\star \mid e^\smile \mid \top$$
$$e, f \in \mathrm{GReg}\langle\Sigma\rangle ::= 0 \mid a \mid e\cdot f \mid e\cap f \mid e\cup f \mid e^+$$

$$\Sigma' := \Sigma \cup \left\{a^\smile \mid a \in \Sigma\right\} \cup \{1, \top\}.$$

$$\begin{aligned} e &= (a\cdot b^\star)^\smile \cup \top \\ &\equiv \left(a\cdot\left(1\cup b^+\right)\right)^\smile \cup \top \\ &\equiv \left(1\cup\left(b^\smile\right)^+\right)\cdot a^\smile \cup \top \in \mathrm{GReg}\langle\Sigma'\rangle \\ &=: \varphi(e) \quad . \end{aligned}$$

From $\mathrm{AReg}\langle\Sigma\rangle$ to $\mathrm{GReg}\langle\Sigma'\rangle$

$$e,f \in \mathrm{AReg}\langle\Sigma\rangle ::= 0 \mid 1 \mid a \mid e\cdot f \mid e\cap f \mid e\cup f \mid e^\star \mid e^\smile \mid \top$$
$$e,f \in \mathrm{GReg}\langle\Sigma\rangle ::= 0 \mid a \mid e\cdot f \mid e\cap f \mid e\cup f \mid e^+$$

$$\Sigma' := \Sigma \cup \left\{a^\smile \mid a\in\Sigma\right\} \cup \{1,\top\}.$$

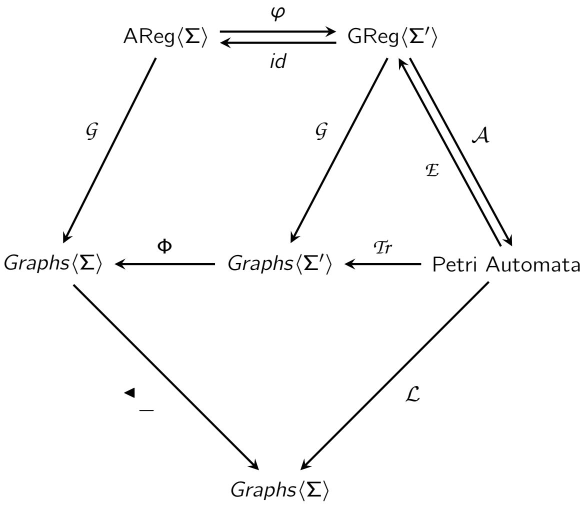
$$\begin{aligned} e &= (a\cdot b^\star)^\smile \cup \top \\ &\equiv \left(a\cdot(1\cup b^+)\right)^\smile \cup \top \\ &\equiv \left(1\cup\left(b^\smile\right)^+\right)\cdot a^\smile \cup \top \in \mathrm{GReg}\langle\Sigma'\rangle \\ &=: \varphi(e) \quad . \end{aligned}$$

$\varphi : \mathrm{AReg}\langle\Sigma\rangle \to \mathrm{GReg}\langle\Sigma'\rangle$

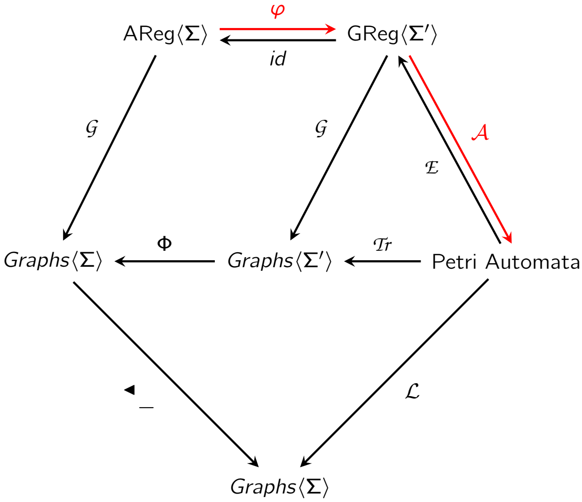
Lemma

$$\mathcal{G}_\Sigma(e) = \Phi\left(\mathcal{G}_{\Sigma'}(\varphi(e))\right).$$

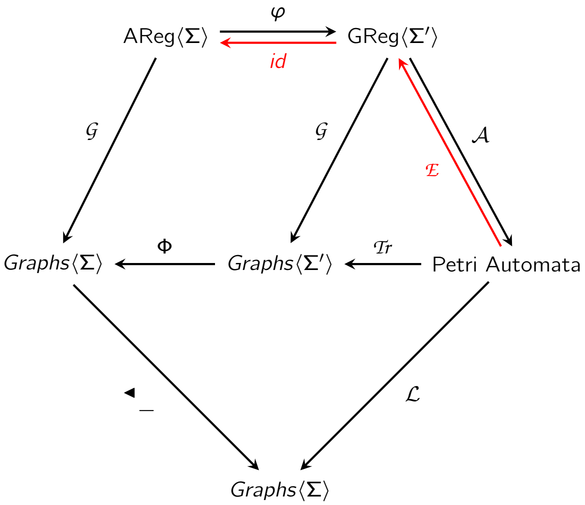
Correspondences



Correspondences



Correspondences



Kleene Theorem

Let $\mathcal{G}$ be a set of graphs labelled with Σ.

Theorem

$\mathcal{G}$ is the language recognised by a Petri automaton if and only if it is the closed graph language of an expression.

Corollary

The equational theory of bounded Kleene allegories is decidable if and only if the comparison of general Petri automata is decidable.

Overview

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

Overview

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

Overview

- **Reduction** of relational equivalence to equality of closed graph languages.
- Representation of closed graph languages through **Petri automata**.
- **Decidability** of simple automata equivalence using a notion of **simulation**, thus of relational equivalence for Identity-free Kleene lattices.

Overview

- **Reduction** of relational equivalence to equality of closed graph languages.
- Representation of closed graph languages through **Petri automata**.
- **Decidability** of simple automata equivalence using a notion of **simulation**, thus of relational equivalence for Identity-free Kleene lattices.
- A proof-of-concept **implementation in OCaml** is available online: `http://perso.ens-lyon.fr/paul.brunet/rklm`.

Overview

- **Reduction** of relational equivalence to equality of closed graph languages.
- Representation of closed graph languages through **Petri automata**.
- **Decidability** of simple automata equivalence using a notion of **simulation**, thus of relational equivalence for Identity-free Kleene lattices.
- A proof-of-concept **implementation in OCaml** is available online: `http://perso.ens-lyon.fr/paul.brunet/rklm`.
- Simple Petri automata equivalence is **ExpSpace-complete**.

Overview

- **Reduction** of relational equivalence to equality of closed graph languages.
- Representation of closed graph languages through **Petri automata**.
- **Decidability** of simple automata equivalence using a notion of **simulation**, thus of relational equivalence for Identity-free Kleene lattices.
- A proof-of-concept **implementation in OCaml** is available online: `http://perso.ens-lyon.fr/paul.brunet/rklm`.
- Simple Petri automata equivalence is **ExpSpace-complete**.
- **Kleene theorems** for:
  - traces of general Petri automata and languages defined by $\mathrm{AReg}\langle\Sigma\rangle$;
  - languages of general PA and closures of languages defined by $\mathrm{AReg}\langle\Sigma\rangle$;
  - traces of simple PA and languages defined by $\mathrm{GReg}\langle\Sigma\rangle$.
  - languages of simple PA and closures of languages defined by $\mathrm{GReg}\langle\Sigma\rangle$;

Outlook

- **Decidability and complexity** of the full relational equivalence:
  - links with ICPDL;
  - comparison of general Petri automata.

Outlook

- **Decidability and complexity** of the full relational equivalence:
  - links with ICPDL;
  - comparison of general Petri automata.
- **Complexity lower bounds** for identity-free Kleene lattices and positive relation algebra.

Outlook

- **Decidability and complexity** of the full relational equivalence:
  - links with ICPDL;
  - comparison of general Petri automata.
- **Complexity lower bounds** for identity-free Kleene lattices and positive relation algebra.
- Study of **Concurrent Kleene Algebra**:
  - graphs $\rightarrow$ pomsets;
  - Petri automata $\rightarrow$ labelled Petri nets.

That's all folks!

Thank you!

See more at:
`http://perso.ens-lyon.fr/paul.brunet`

Outline