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 $O$.

$$
\text{Rel} |\models 1 \cup R^* \cdot S \subseteq (R \cup S)^*
$$

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

$$
\text{Rel} |\models (R \cdot S) \cap T \subseteq R \cdot (S \cap R^\sim \cdot T)
$$

$$
\text{Rel} |\models (1 \cap (R \cdot S))^* \subseteq R \cdot S \cdot R
$$

**Relation operators**

- identity relation : $1$
- empty relation : $0$
- composition : $R \cdot S$
- union : $R \cup S$
- intersection : $R \cap S$
- converse : $R^\sim$
- reflexive transitive closure : $R^*$
Motivation: Relation Algebra

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

\[
\text{Rel} |\models 1 \cup R^* \cdot S \subseteq (R \cup S)^*
\]

\[
\text{Rel} |\models (R \cap S) \cdot T \subseteq (R \cdot T) \cap (S \cdot T)
\]

\[
\text{Rel} |\models (R \cdot S) \cap T \subseteq R \cdot (S \cap R^* \cdot T)
\]

\[
\text{Rel} |\not\models (1 \cap (R \cdot S))^* \subseteq R \cdot S \cdot R
\]
Motivation : Relation Algebra

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

\[
\text{Rel} \models 1 \cup R^* \cdot S \subseteq (R \cup S)^* \\
\text{Rel} \models (R \cap S) \cdot T \subseteq (R \cdot T) \cap (S \cdot T) \\
\text{Rel} \models (R \cdot S) \cap T \subseteq R \cdot (S \cap R^r \cdot T) \\
\text{Rel} \not\models (1 \cap (R \cdot S))^* \subseteq R \cdot S \cdot R
\]

Simple and boring : could it be done automatically ?
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
Regular expressions

\[ e, f \in \text{Reg}_X := 0 \mid 1 \mid x \in X \mid e \cdot f \mid e \cup f \mid e^* \]
Regular expressions

\[ e, f \in \text{Reg}_X \ ::= \ 0 \ | \ 1 \ | \ x \in X \ | \ e \cdot f \ | \ e \cup f \ | \ e^* \]

**Theorem**

\[ \text{Rel} \models e = f \iff L(e) = L(f) \]

where \( L(e) \subseteq X^* \) is the rational language represented by \( e \).
Regular expressions

\[ e, f \in \text{Reg}_X := 0 \mid 1 \mid x \in X \mid e \cdot f \mid e \cup f \mid e^* \]

Theorem

\[ \text{Rel} \models e = f \iff L(e) = L(f) \]

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

Corollary

Relational equivalence is decidable for regular expressions.
Kleene Allegories

\[ e, f \in \text{Reg}_X \implies 0 \mid 1 \mid x \in X \mid e \cdot f \mid e \sqcap f \mid e \sqcup f \mid e^* \mid e^\sim \]
Kleene Allegories

\[ e, f \in \text{Reg}_X :\!:\! rowspan{2} = 0 \mid 1 \mid x \in X \mid e \cdot f \mid e \cap f \mid e \cup f \mid e^* \mid e^\sim \]

\[ \text{Rel} \models e = f \iff L(e) = L(f) \]
Kleene Allegories

\[ e, f \in \text{Reg}_X \ni 0 \ | \ 1 \ | \ x \in X \ | \ e \cdot f \ | \ e \cap f \ | \ e \cup f \ | \ e^* \ | \ e^\sim \]

\[ \text{Rel} \models e = f \iff L(e) = L(f) \]

Counterexample

\[ L(a \cap b) = \emptyset = L(0) \quad | \quad L(a) = \{a\} = L(a^\sim) \]
\[ \text{Rel} \not\models a \cap b = 0 \quad | \quad \text{Rel} \not\models a = a^\sim \]

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

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
Graphs/Ground terms

\[ u, v \in W_X ::= 0 \mid 1 \mid x \in X \mid u \cdot v \mid u \cap v \mid u \cup v \mid u^* \mid u^\sim \]
Graphs/Ground terms

\[ u, v \in W_X \; ::= \; 0 \; | \; 1 \; | \; x \in X \; | \; u \cdot v \; | \; u \cap v \; | \; u \cup v \; | \; u^* \; | \; u^\sim \]

\[ G(1) := \quad G(u \cdot v) := \quad G(u^\sim) := \]

\[ G(x) := \quad x \quad G(u^*) := \]

\[ G(u \cap v) := \]

Paul Brunet  
Kleene Allegories & Petri automata  
July 5, 2015
Examples

$G(a \cdot b)$:
Examples

\[ G(a \cdot b) : \]

\[ G((a \cdot b) \cap (c \cdot b)) : \]
Examples

\[ G(a \cdot b) : \]

\[ G((a \cdot b) \cap (c \cdot b)) : \]

\[ G((a \cdot b) \cap 1) : \]
Examples

\[ \mathcal{G} (a \cdot b): \]

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

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

\[ \mathcal{G} (a \cap b^-): \]
Preorder

Preorder on graphs

\(G \preceq H\) if there exists a graph morphism from \(H\) to \(G\).

\[
\begin{align*}
G : \quad & (a \cap c) \cdot b \cap d \\
H : \quad & (a \cdot b) \cap (c \cdot b)
\end{align*}
\]
Characterization theorem

\[ u, v \in W_X \implies 1 \mid x \in X \mid u \cdot v \mid u \cap v \mid u \sim \]

Theorem

\[ \text{Rel} \models u \subseteq v \iff \mathcal{G}(u) \blacktriangledown \mathcal{G}(v) \]

Graph languages

$e, f \in \text{Reg}_X \mapsto 0 \mid 1 \mid x \in X \mid e \cdot f \mid e \cap f \mid e \cup f \mid e^* \mid e^*$

\[ \mathcal{G}(1) := \{ \begin{array}{c}
\xrightarrow{e} \\
\end{array} \} \quad \mathcal{G}(x) := \begin{cases} 
\begin{array}{c}
\xrightarrow{e} \\
\xrightarrow{f} \\
\end{array} 
\end{cases} \]

$\mathcal{G}(e^\sim) := \{ G^{-1} \mid G \in \mathcal{G}(e) \}$

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

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

\[ \mathcal{G}(0) := \emptyset \quad \mathcal{G}(e \cup f) := \mathcal{G}(e) \cup \mathcal{G}(f) \]

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

\( \downarrow S \) is the downwards closure of \( S \) with respect to \( \downarrow \).

**Theorem**

\( e, f \in \text{Reg}_\cap \),

\[
\text{Rel} \models e \subseteq f \iff \downarrow G(e) \subseteq \downarrow G(f)
\]

Follows easily from:

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
Example

\[((a \cap c) \cdot b) \cap d) \cup a\]
Example

\[(b^\sim \cdot (a \cdot c \cap b)^* \cdot d) \cap a \cup a^\sim \cdot b\]
Reading a graph in an automaton

Paul Brunet
Kleene Allegories & Petri automata
July 5, 2015
Reading a graph in an automaton
Reading a graph in an automaton
Reading a graph in an automaton

```
A -> 0 -> 2
     |    |
     a    a
     |    |
     1 -> F
     |    |
     b    b
     |    |
     2     1
     |    |
     d    d
     |    |
     D
```

```
A -> 0 -> 2
     |    |
     a    a
     |    |
     1 -> F
     |    |
     b    b
     |    |
     2     1
     |    |
     d    d
     |    |
     D
```
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

\[ \text{Diagram of an automaton} \]
Reading a graph in an automaton

A graph representing an automaton with nodes labeled A, B, C, D, E, F, 0, 1, and 2. The edges are labeled with the letters a, b, c, and d. The graph starts at node A and moves through nodes 0, B, 1, C, D, and F, with transitions labeled as follows:

- From A to 0: a
- From 0 to B: c
- From B to 1: d
- From F to D: b
- From 2 to F: a

The graph illustrates the state transitions in the 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 $e \in \operatorname{Reg}_X^\sim \cap X$, 

$$e$$
Language recognised by an automaton

Correctness

For any $e \in \text{Reg}_X^\sim \cap X$, $\mathcal{A}(e)$
Language recognised by an automaton

Correctness

For any $e \in \text{Reg} X \cap X$, 

$$L(A(e))$$
Language recognised by an automaton

Correctness

For any $e \in \text{Reg}_X \cap \mathcal{X}$,

$$L(\mathcal{A}(e)) = \downarrow \mathcal{L}(e).$$
Language recognised by an automaton

**Correctness**

For any \( e \in \text{Reg}_X \cap \),

\[
L(A(e)) = \text{G}(e).
\]

**So far:**

\( e, f \in \text{Reg}_X \cap \)

\[
\text{Rel} \models e \subseteq f \iff \text{G}(e) \subseteq \text{G}(f) \iff L(A(e)) \subseteq L(A(f)).
\]
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
Restriction: identity-free lattice terms

\[ G ((a \cdot b) \cap 1): \]

\[ G (a \cap b^-): \]

Paul Brunet
Kleene Allegories & Petri automata
July 5, 2015
Restriction: identity-free lattice terms

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

\[ \mathcal{G} (a \cap b^-): \]

Identity-free Kleene Lattice

\[ u, v \in W_X^- := 1 | x \in X | u \cdot v | u \cap v | u^- \]

\[ e, f \in \text{Reg}_X^- := 0 | 1 | x \in X | e \cdot f | e \cap f | e \cup f | e^+ | e^- \]
Decision procedure

\[ e, f \in \text{Reg}_{\chi}^{\cap -} \]

\[ \text{Rel} \models e \subseteq f \iff \mathcal{G}(e) \subseteq \mathcal{G}(f) \iff \mathcal{L}(\mathcal{A}(e)) \subseteq \mathcal{L}(\mathcal{A}(f)). \]

**Problem:**

How to compare two Petri automata?
Decision procedure

\[ e, f \in \text{Reg}_X^\cap - \]

\[ \text{Rel} \models e \subseteq f \iff \mathcal{G}(e) \subseteq \mathcal{G}(f) \iff \mathcal{L}(\mathcal{A}(e)) \subseteq \mathcal{L}(\mathcal{A}(f)). \]

**Problem:**

How to compare two Petri automata?

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

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

\[ \leq \subseteq \mathcal{P}(P_1) \times \mathcal{P}(P_2 \rightarrow P_1) \]

between the configurations of \( \mathcal{A}_1 \) and the partial maps from the places of \( \mathcal{A}_2 \) to the places of \( \mathcal{A}_1 \).
Comparing automata

Simulations - Non-deterministic Finite Automata

\[ \preceq \subseteq Q_1 \times \mathcal{P}(Q_2) \]

Diagram:

- States: \( p_0, p_1, p_2, p_3 \) and \( q_0, q_1, q_2, q_3, q_4, q_5 \)
- Transitions:
  - \( p_0 \rightarrow p_1 \) with label \( a \)
  - \( p_1 \rightarrow p_2 \) with label \( b \)
  - \( p_0 \rightarrow p_3 \) with label \( a \)
  - \( p_3 \rightarrow q_1 \) with label \( a \)
  - \( q_1 \rightarrow q_2 \) with label \( b \)
  - \( q_2 \rightarrow q_3 \) with label \( a \)
  - \( q_1 \rightarrow q_4 \) with label \( b \)
  - \( q_4 \rightarrow q_5 \) with label \( b \)
  - \( q_1 \rightarrow q_5 \) with label \( b \)

Paul Brunet
Kleene Allegories & Petri automata
July 5, 2015
Comparing automata

Simulations - Non-deterministic Finite Automata

\[ \preceq \subseteq Q_1 \times \mathcal{P}(Q_2) \]

\[
\begin{array}{c}
\text{p}_0 \\
\text{p}_1 \\
\text{p}_2 \\
\end{array}
\]

\[
\begin{array}{c}
\text{q}_0 \\
\text{q}_1 \\
\text{q}_2 \\
\text{q}_3 \\
\text{q}_4 \\
\text{q}_5 \\
\end{array}
\]

\[
p_0 \preceq \{ q_0 \}
\]
Simulations - Non-deterministic Finite Automata

\[ \preceq \subseteq Q_1 \times \mathcal{P}(Q_2) \]

\[
\begin{align*}
p_0 &\preceq \{q_0\} \\
p_1 &\preceq \{q_1\}
\end{align*}
\]
Simulations - Non-deterministic Finite Automata

\[
\leq \subseteq Q_1 \times \mathcal{P}(Q_2)
\]

\[
p_0 \leq \{ q_0 \}
\]

\[
p_1 \leq \{ q_1 \}
\]

\[
p_2 \leq \{ q_2, q_4 \}
\]
Comparing automata

Simulations - Non-deterministic Finite Automata

\[ \preceq \subseteq Q_1 \times \mathcal{P}(Q_2) \]

\[ p_0 \preceq \{ q_0 \} \]
\[ p_1 \preceq \{ q_1 \} \]
\[ p_2 \preceq \{ q_2, q_4 \} \]
\[ p_3 \preceq \{ q_1 \} \]

Paul Brunet
Kleene Allegories & Petri automata
July 5, 2015
Comparing automata

Simulations - Non-deterministic Finite Automata

\[ \subseteq Q_1 \times \mathcal{P}(Q_2) \]

\[ p_0 \subseteq \{ q_0 \} \]
\[ p_1 \subseteq \{ q_1 \} \]
\[ p_2 \subseteq \{ q_2, q_4 \} \]
\[ p_3 \subseteq \{ q_1 \} \]

+ condition for accepting states
Comparing automata

Simulation - Petri Automata

Simulation relation

\[ \leq \subseteq \mathcal{P}(P_1) \times \mathcal{P}(P_2 \rightarrow P_1) \]
Simulation - Petri Automata

Simulation relation

$$\preceq \subseteq \mathcal{P}(P_1) \times \mathcal{P}(P_2 \rightarrow P_1)$$
Comparing automata

Simulation - Petri Automata

Simulation relation

\[ \leq \subseteq \mathcal{P}(P_1) \times \mathcal{P}(P_2 \rightarrow P_1) \]
Simulation - Petri Automata

Simulation relation

\[ \leq \subseteq \mathcal{P}(P_1) \times \mathcal{P}(P_2 \rightarrow P_1) \]
Conclusion and future work

Results

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

Ongoing/Future work

Paul Brunet
Kleene Allegories & Petri automata
July 5, 2015
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
Conclusions

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 _\_\~_.

Conclusions

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 _˘_.
- Complete axiomatization.
That’s it!

Thank you!

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