Pomset languages and concurrent Kleene algebras

Completeness and decidability of CKA

Theory seminar, QMU – February 13, 2018

Paul Brunet[1], Damien Pous[2], Georg Struth[3], Tobias Kappé[1], Alexandra Silva[1], and Fabio Zanasi[1]

University College London[1], ENS de Lyon – CNRS[2], University of Sheffield[3]

UCL ENS DE LYON The University Of Sheffield.

Kleene Algebra

Equivalence of sequential programs

$$(x := 1; y := 2); (x := y \oplus y := x) \qquad \equiv \qquad x := 1; (y := 2; x := y) \oplus (y := 2; y := x)$$

Kleene Algebra

Equivalence of sequential programs

$$(x := 1; y := 2); (x := y \oplus y := x) \qquad \equiv \qquad x := 1; (y := 2; x := y) \oplus (y := 2; y := x)$$

$$(x1 \cdot y2) \cdot (xy + yx)$$

Kleene Algebra

Equivalence of sequential programs

$$(x := 1; y := 2); (x := y \oplus y := x) \qquad \equiv \qquad x := 1; (y := 2; x := y) \oplus (y := 2; y := x)$$

$$(x1 \cdot y2) \cdot (xy + yx) = x1 \cdot (y2 \cdot (xy + yx)) \qquad \text{(associativity of } \cdot\text{)}$$

Kleene Algebra

Equivalence of sequential programs

$$(x := 1; y := 2); (x := y \oplus y := x) \qquad \equiv \qquad x := 1; (y := 2; x := y) \oplus (y := 2; y := x)$$

$$\begin{aligned}(x1 \cdot y2) \cdot (xy + yx) &= x1 \cdot (y2 \cdot (xy + yx)) && \text{(associativity of } \cdot\text{)} \\ &= x1 \cdot ((y2 \cdot xy) + (y2 \cdot yx)) && \text{(distributivity)}\end{aligned}$$

Kleene Algebra

Equivalence of sequential programs

A Kleene algebra is structure $\langle K, 0, 1, +, \cdot, \star \rangle$ such that:

1) $\langle K, 0, 1, +, \cdot \rangle$ is an idempotent semiring;
2) $\forall x \in K, 1 + x \cdot x^\star = x^\star$;
3) $\forall x, y, z \in K, x + y \cdot z \leq z \Rightarrow y^\star \cdot x \leq z$.

Theorem

$$\mathrm{KA} \vdash e = f \Leftrightarrow \mathcal{L}(e) = \mathcal{L}(f) .$$

☞ Krob, "A Complete System of B-Rational Identities", 1990.

☞ Kozen, "A Completeness Theorem for Kleene Algebras and the Algebra of Regular Events", 1991.

☞ Kozen & Silva, "Left-Handed Completeness", 2012.

Kleene Algebra

Equivalence of sequential programs

Kleene Algebra

$\Updownarrow$

regular languages

$\Updownarrow$

finite state automata

Kleene Algebra

Equivalence of sequential programs

Completeness

Kleene Algebra

$\Updownarrow$

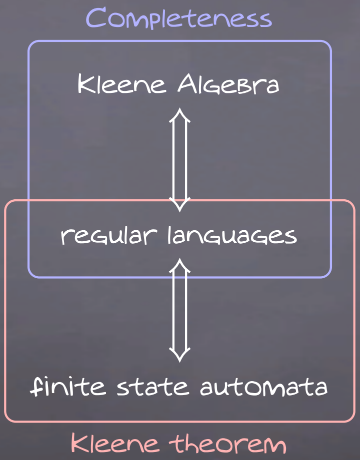
regular languages

$\Updownarrow$

finite state automata

Kleene Algebra

Equivalence of sequential programs

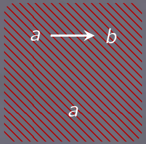


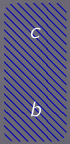
Concurrent Kleene Algebras
Equivalence of parallel programs

$$e, f \in \mathbb{E} ::= 0 \mid 1 \mid a \mid e + f \mid e \cdot f \mid e^\star \mid e \parallel f$$

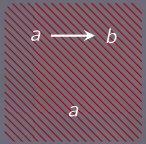
Bi-Kleene Algebra	Concurrent Kleene Algebra
$\Updownarrow$	$\Updownarrow$
series-rational pomset languages	down-closed series-rational languages
$\Updownarrow$	$\Updownarrow$
automata ?	automata ?

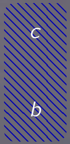
Pomsets

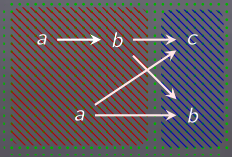
$P_1 =$ 

$P_2 =$ 

Pomsets

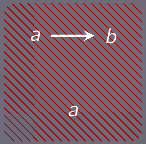
$P_1 =$ 

$P_2 =$ 

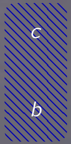
$P_1 \cdot P_2 =$ 

Pomsets

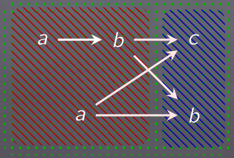
$P_1 =$



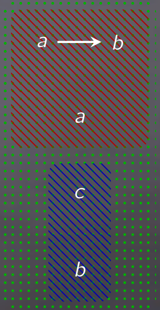
$P_2 =$



$P_1 \cdot P_2 =$



$P_1 \parallel P_2 =$



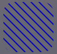
Rational pomset languages

$$e, f \in \mathbb{E} ::= a \mid 0 \mid 1 \mid e \cdot f \mid e \parallel f \mid e + f \mid e^\star .$$

Rational pomset languages

$$e, f \in \mathbb{E} ::= a \mid 0 \mid 1 \mid e \cdot f \mid e \parallel f \mid e + f \mid e^\star.$$

$[\![a]\!] :=$ {  }

$[\![1]\!] :=$ {  }

$$[\![0]\!] := \emptyset$$

$$[\![e + f]\!] := [\![e]\!] \cup [\![f]\!]$$

$$[\![e \cdot f]\!] := [\![e]\!] \cdot [\![f]\!]$$

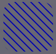
$$[\![e \parallel f]\!] := [\![e]\!] \parallel [\![f]\!]$$

$$[\![e^\star]\!] := \bigcup_{n \in \mathbb{N}} [\![e]\!]^n$$

Rational pomset languages

$$e, f \in \mathbb{E} ::= a \mid 0 \mid 1 \mid e \cdot f \mid e \parallel f \mid e + f \mid e^\star.$$

$[\![a]\!] := \{$  $\}$

$[\![1]\!] := \{$  $\}$

$$[\![0]\!] := \emptyset \qquad [\![e + f]\!] := [\![e]\!] \cup [\![f]\!]$$

$$[\![e \cdot f]\!] := [\![e]\!] \cdot [\![f]\!] \qquad [\![e \parallel f]\!] := [\![e]\!] \parallel [\![f]\!]$$

$$[\![e^\star]\!] := \bigcup_{n \in \mathbb{N}} [\![e]\!]^n$$

Definition

A set of pomsets S is called a rational pomset language if there is an expression $e \in \mathbb{E}$ such that $S = [\![e]\!]$.

Kleene Algebra

Equivalence of sequential programs

A Kleene algebra is structure $\langle K, 0, 1, +, \cdot, \star\rangle$ such that:

1) $\langle K, 0, 1, +, \cdot\rangle$ is an idempotent semiring;
2) $\forall x \in K, 1 + x \cdot x^\star = x^\star$;
3) $\forall x, y, z \in K, x + y \cdot z \leq z \Rightarrow y^\star \cdot x \leq z$.

Theorem

$$\mathrm{KA} \vdash e = f \Leftrightarrow \mathcal{L}(e) = \mathcal{L}(f).$$

☞ Krob, "A Complete System of B-Rational Identities", 1990.

☞ Kozen, "A Completeness Theorem for Kleene Algebras and the Algebra of Regular Events", 1991.

☞ Kozen & Silva, "Left-Handed Completeness", 2012.

Bi-Kleene algebra

A bi-Kleene algebra is structure $\langle K, 0, 1, +, \cdot, \star, \parallel \rangle$ such that:

1) $\langle K, 0, 1, +, \cdot \rangle$ is an idempotent semiring;
2) $\forall x \in K, 1 + x \cdot x^\star = x^\star$;
3) $\forall x, y, z \in K, x + y \cdot z \leq z \Rightarrow y^\star \cdot x \leq z$;

Bi-Kleene algebra

A bi-Kleene algebra is structure $\langle K, 0, 1, +, \cdot, \star, \| \rangle$ such that:

1) $\langle K, 0, 1, +, \cdot \rangle$ is an idempotent semiring;
2) $\forall x \in K, 1 + x \cdot x^\star = x^\star$;
3) $\forall x, y, z \in K, x + y \cdot z \leq z \Rightarrow y^\star \cdot x \leq z$;
4) $\langle K, 0, 1, +, \| \rangle$ is a commutative idempotent semiring.

Bi-Kleene algebra

A bi-Kleene algebra is structure $\langle K, 0, 1, +, \cdot, \star, \| \rangle$ such that:

1) $\langle K, 0, 1, +, \cdot \rangle$ is an idempotent semiring;
2) $\forall x \in K, 1 + x \cdot x^\star = x^\star$;
3) $\forall x, y, z \in K, x + y \cdot z \leq z \Rightarrow y^\star \cdot x \leq z$;
4) $\langle K, 0, 1, +, \| \rangle$ is a commutative idempotent semiring.

Theorem

$$\mathrm{biKA} \vdash e = f \Leftrightarrow [\![e]\!] = [\![f]\!].$$

☞ Laurence & Struth, "Completeness theorems for bi-Kleene algebras and series-parallel rational pomset languages", 2014.

Pomset order

$$\begin{tikzcd} a \arrow[r] \arrow[dr] & b \\ c \arrow[r] \arrow[ur] & d \end{tikzcd} \quad \sqsubseteq \quad \begin{tikzcd} a \arrow[r] & b \\ c \arrow[r] & d \end{tikzcd}$$

Pomset order

$$
\begin{tikzcd}
a \arrow[r] \arrow[dr] & b \\
c \arrow[r] \arrow[ur] & d
\end{tikzcd}
\quad \sqsubseteq \quad
\begin{tikzcd}
a \arrow[r] & b \\
c \arrow[r] & d
\end{tikzcd}
$$

Definition

$P_1 \sqsubseteq P_2$ if there is a function $\varphi : P_2 \to P_1$ such that:

1) φ is a bijection
2) φ preserves labels
3) φ preserves ordered pairs

☞ Gischer, "The equational theory of pomsets", 1988.

☞ Grabowski, "On partial languages", 1981.

Pomset order

Definition

$P_1 \sqsubseteq P_2$ if there is a function $\varphi : P_2 \to P_1$ such that:

1) φ is a bijection
2) φ preserves labels
3) φ preserves ordered pairs

☞ Gischer, "The equational theory of pomsets", 1988.

☞ Grabowski, "On partial languages", 1981.

Notation: ${}^{\sqsubseteq}S := \{P \mid \exists P' \in S : P \sqsubseteq P'\}$.

Concurrent Kleene algebra

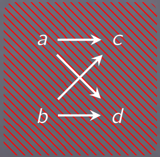
A concurrent Kleene algebra is bi-Kleene algebra $\langle K, 0, 1, +, \cdot, \star, \parallel\rangle$ such that:

$$(a \parallel b) \cdot (c \parallel d) \leq (a \cdot c) \parallel (b \cdot d)$$

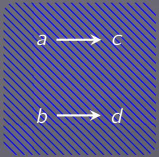
Concurrent Kleene algebra

A concurrent Kleene algebra is bi-Kleene algebra $\langle K, 0, 1, +, \cdot, \star, \parallel \rangle$ such that:

$$(a \parallel b) \cdot (c \parallel d) \leq (a \cdot c) \parallel (b \cdot d)$$



$\sqsubseteq$



Concurrent Kleene algebra

A concurrent Kleene algebra is bi-Kleene algebra $\langle K, 0, 1, +, \cdot, \star, \parallel \rangle$ such that:

$$(a \parallel b) \cdot (c \parallel d) \leq (a \cdot c) \parallel (b \cdot d)$$

$$\begin{tikzcd} a \arrow[r] \arrow[dr] & c \\ b \arrow[r] \arrow[ur] & d \end{tikzcd} \quad \sqsubseteq \quad \begin{tikzcd} a \arrow[r] & c \\ b \arrow[r] & d \end{tikzcd}$$

Theorem

$$\mathrm{CKA} \vdash e = f \Rightarrow {}^{\sqsubseteq}[\![e]\!] = {}^{\sqsubseteq}[\![f]\!].$$

☞ Hoare, Möller, Struth & Wehrman, "Concurrent Kleene Algebra", 2009.

Outline

$\mathbb{E}$

Outline

$$\begin{tikzcd} \mathbb{E} \arrow[d, "[\![-]\!]"] \\ \mathcal{P}(\mathbb{P}) \end{tikzcd}$$

Outline

$$
\begin{tikzcd}
\mathbb{E} \arrow[d, "\llbracket \_ \rrbracket"] \\
\mathcal{P}(\mathbb{P}) \arrow[d, "{}^{\sqsubseteq}(\_)"] \\
\mathcal{P}(\mathbb{P})
\end{tikzcd}
$$

Outline

$$\begin{tikzcd}
 & \mathbb{E} \arrow[dl, bend right, "(_)\downarrow"'] \arrow[d, "[\![_]\!]"] \\
\mathbb{E} & \mathcal{P}(\mathbb{P}) \arrow[d, "{}^{\sqsubseteq}(_)"] \\
 & \mathcal{P}(\mathbb{P})
\end{tikzcd}$$

Outline

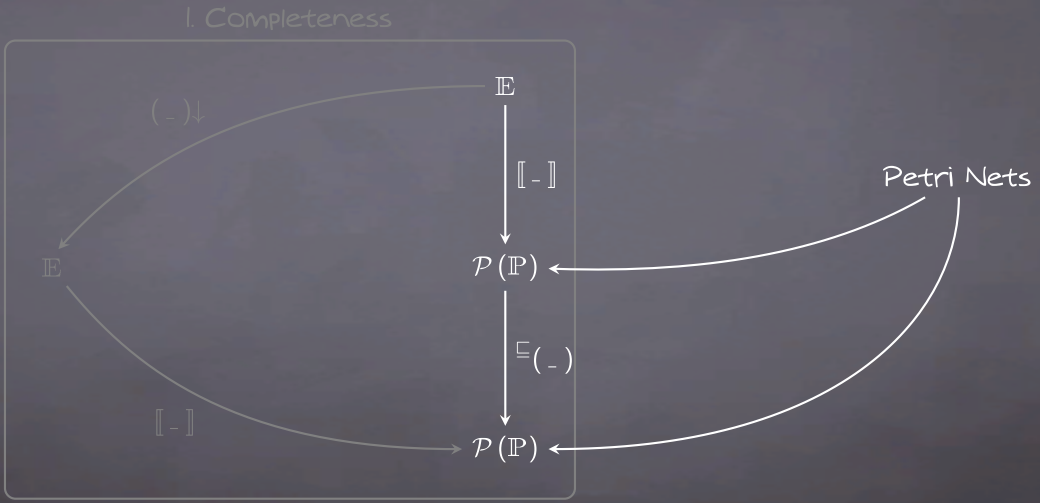
$$
\begin{tikzcd}
 & \mathbb{E} \arrow[dl, bend right, "(\_)\downarrow"'] \arrow[d, "\llbracket \_ \rrbracket"] \\
\mathbb{E} \arrow[dr, bend right, "\llbracket \_ \rrbracket"'] & \mathcal{P}(\mathbb{P}) \arrow[d, "{}^{\sqsubseteq}(\_)"] \\
 & \mathcal{P}(\mathbb{P})
\end{tikzcd}
$$

Outline

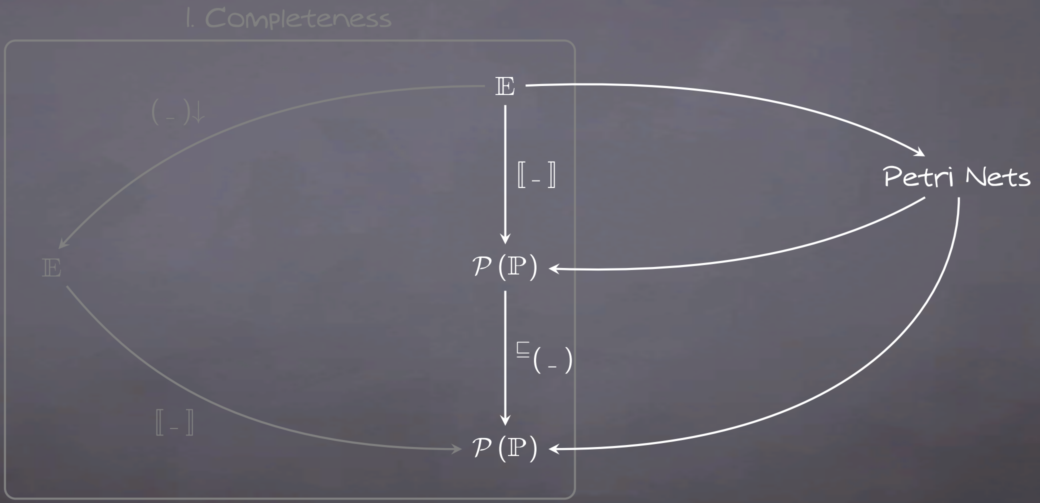
1. Completeness

$$
\begin{tikzcd}
& \mathbb{E} \arrow[dl, bend right, "(\_)\downarrow"'] \arrow[d, "[\![\_]\!]"] \\
\mathbb{E} \arrow[dr, bend right, "[\![\_]\!]"'] & \mathcal{P}(\mathbb{P}) \arrow[d, "{}^{\sqsubseteq}(\_)"] \\
& \mathcal{P}(\mathbb{P})
\end{tikzcd}
$$

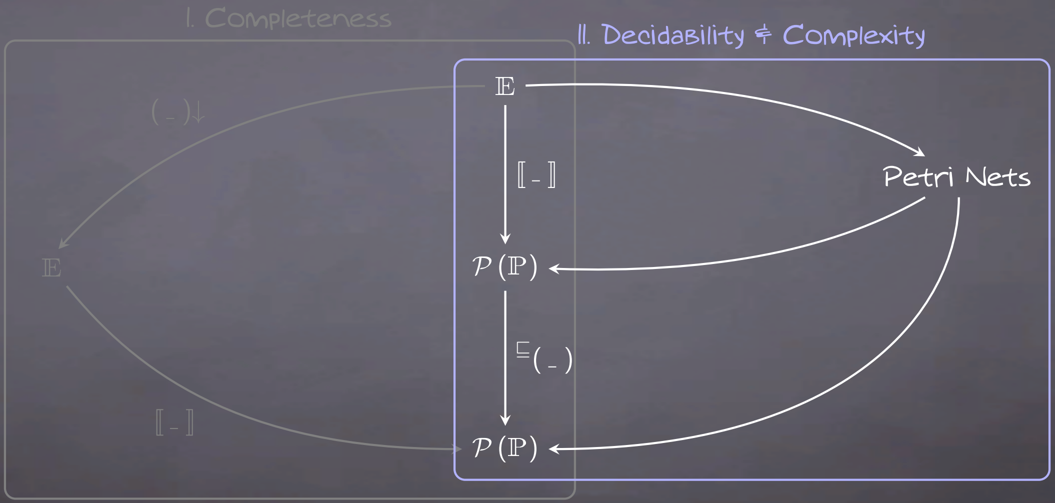
Outline


1. Completeness
$\mathbb{E}$
$(\_)\downarrow$
$[\![\_]\!]$
$\mathbb{E}$
$\mathcal{P}(\mathbb{P})$
${}^{\sqsubseteq}(\_)$
$[\![\_]\!]$
$\mathcal{P}(\mathbb{P})$
Petri Nets

Outline


1. Completeness
$\mathbb{E}$
$(\_)\downarrow$
$[\![\_]\!]$
Petri Nets
$\mathbb{E}$
$\mathcal{P}(\mathbb{P})$
${}^{\sqsubseteq}(\_)$
$[\![\_]\!]$
$\mathcal{P}(\mathbb{P})$

Outline

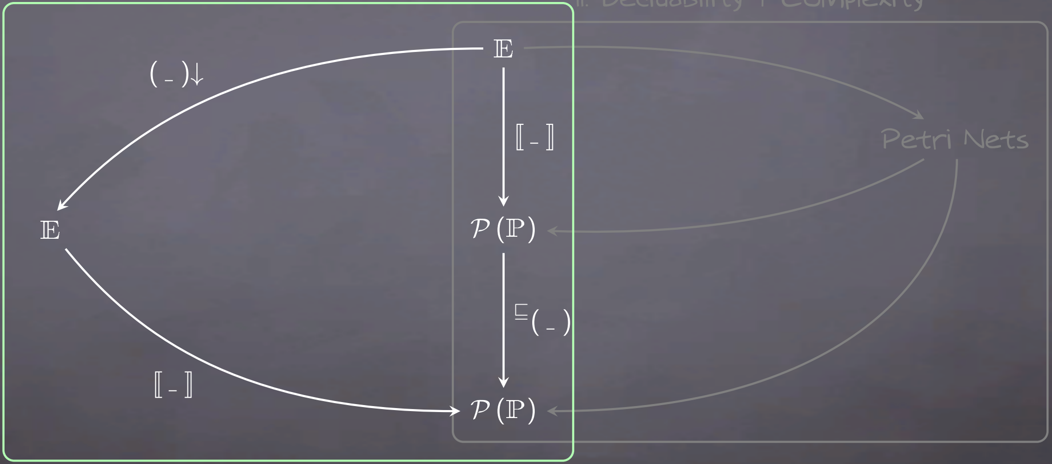

I. Completeness
II. Decidability & Complexity
$\mathbb{E}$
$(\_)\!\downarrow$
$\mathbb{E}$
$[\![\_]\!]$
$[\![\_]\!]$
$\mathcal{P}(\mathbb{P})$
${}^{\sqsubseteq}(\_)$
$\mathcal{P}(\mathbb{P})$
Petri Nets

Outline

Kappé, Brunet, Silva & Zanasi, "Concurrent Kleene Algebra: Free Model and Completeness", 2018

I. Completeness

II. Decidability & Complexity



Syntactic closures are nice...

Definition

An expression $e{\downarrow}$ is a closure of e if $\mathrm{CKA} \vdash e{\downarrow} = e$ and $[\![e{\downarrow}]\!] = {}^{\sqsubseteq}[\![e]\!]$.

Syntactic closures are nice...

Definition

An expression $e{\downarrow}$ is a closure of e if $\overbrace{\mathrm{CKA} \vdash e{\downarrow} = e}^{(1)}$ and $\overbrace{[\![e{\downarrow}]\!] = {}^{\sqsubseteq}[\![e]\!]}^{(2)}$.

Lemma

If every series-rational expression admits a closure, the axioms of CKA are complete with respect to down-closed pomset languages.

☞ Laurence & Struth, "Completeness theorems for pomset languages and concurrent Kleene Algebras", (draft).

Syntactic closures are nice...

Definition

An expression $e\!\downarrow$ is a closure of e if $\overbrace{\mathrm{CKA} \vdash e\!\downarrow = e}^{(1)}$ and $\overbrace{[\![e\!\downarrow]\!] = {}^{\sqsubseteq}[\![e]\!]}^{(2)}$.

Lemma

If every series-rational expression admits a closure, the axioms of CKA are complete with respect to down-closed pomset languages.

☞ Laurence & Struth, "Completeness theorems for pomset languages and concurrent Kleene Algebras", (draft).

Proof. Assume ${}^{\sqsubseteq}[\![e]\!] = {}^{\sqsubseteq}[\![f]\!]$.

Syntactic closures are nice...

Definition

An expression $e{\downarrow}$ is a closure of e if $\overbrace{\mathrm{CKA} \vdash e{\downarrow} = e}^{(1)}$ and $\overbrace{[\![e{\downarrow}]\!] = {}^{\sqsubseteq}[\![e]\!]}^{(2)}$.

Lemma

If every series-rational expression admits a closure, the axioms of CKA are complete with respect to down-closed pomset languages.

☞ Laurence & Struth, "Completeness theorems for pomset languages and concurrent Kleene Algebras", (draft).

Proof. Assume ${}^{\sqsubseteq}[\![e]\!] = {}^{\sqsubseteq}[\![f]\!]$.
By (2), it means that $[\![e{\downarrow}]\!] = [\![f{\downarrow}]\!]$.

Syntactic closures are nice...

Definition

An expression $e{\downarrow}$ is a closure of e if $\overbrace{\mathrm{CKA} \vdash e{\downarrow} = e}^{(1)}$ and $\overbrace{[\![e{\downarrow}]\!] = {}^{\sqsubseteq}[\![e]\!]}^{(2)}$.

Lemma

If every series-rational expression admits a closure, the axioms of CKA are complete with respect to down-closed pomset languages.

☞ Laurence & Struth, "Completeness theorems for pomset languages and concurrent Kleene Algebras", (draft).

Proof. Assume ${}^{\sqsubseteq}[\![e]\!] = {}^{\sqsubseteq}[\![f]\!]$.
By (2), it means that $[\![e{\downarrow}]\!] = [\![f{\downarrow}]\!]$.
By completeness of biKA, it follows that $\mathrm{biKA} \vdash e{\downarrow} = f{\downarrow}$, thus $\mathrm{CKA} \vdash e{\downarrow} = f{\downarrow}$.

Syntactic closures are nice...

Definition

An expression $e{\downarrow}$ is a <u>closure</u> of e if $\overbrace{\mathrm{CKA} \vdash e{\downarrow} = e}^{(1)}$ and $\overbrace{\llbracket e{\downarrow} \rrbracket = {}^{\sqsubseteq}\llbracket e \rrbracket}^{(2)}$.

Lemma

If every series-rational expression admits a closure, the axioms of CKA are complete with respect to down-closed pomset languages.

☞ Laurence & Struth, "Completeness theorems for pomset languages and concurrent Kleene Algebras", (draft).

<u>Proof.</u> Assume ${}^{\sqsubseteq}\llbracket e \rrbracket = {}^{\sqsubseteq}\llbracket f \rrbracket$.
By (2), it means that $\llbracket e{\downarrow} \rrbracket = \llbracket f{\downarrow} \rrbracket$.
By completeness of biKA, it follows that $\mathrm{biKA} \vdash e{\downarrow} = f{\downarrow}$, thus $\mathrm{CKA} \vdash e{\downarrow} = f{\downarrow}$.
By (1) we get that $\mathrm{CKA} \vdash e = e{\downarrow} = f{\downarrow} = f$.

Syntactic closures are nice...

Definition

An expression $e\!\downarrow$ is a closure of e if $\overbrace{\mathrm{CKA} \vdash e\!\downarrow = e}^{(1)}$ and $\overbrace{[\![e\!\downarrow]\!] = {}^{\sqsubseteq}[\![e]\!]}^{(2)}$.

Lemma

If every series-rational expression admits a closure, the axioms of CKA are complete with respect to down-closed pomset languages.

☞ Laurence & Struth, "Completeness theorems for pomset languages and concurrent Kleene Algebras", (draft).

Proof. Assume ${}^{\sqsubseteq}[\![e]\!] = {}^{\sqsubseteq}[\![f]\!]$.
By (2), it means that $[\![e\!\downarrow]\!] = [\![f\!\downarrow]\!]$.
By completeness of biKA, it follows that $\mathrm{biKA} \vdash e\!\downarrow = f\!\downarrow$, thus $\mathrm{CKA} \vdash e\!\downarrow = f\!\downarrow$.
By (1) we get that $\mathrm{CKA} \vdash e = e\!\downarrow = f\!\downarrow = f$. □

… but do they exist?

Let's try and compute the closure by induction:

… but do they exist?

Let's try and compute the closure by induction:

- ☞ $0{\downarrow} = 0$
- ☞ $1{\downarrow} = 1$
- ☞ $a{\downarrow} = a$

… but do they exist?

Let's try and compute the closure by induction:

- ☞ $0{\downarrow} = 0$
- ☞ $1{\downarrow} = 1$
- ☞ $a{\downarrow} = a$
- ☞ $(e + f){\downarrow} = e{\downarrow} + f{\downarrow}$

… BUT DO THEY EXIST?

Let's try and compute the closure by induction:

- $0{\downarrow} = 0$
- $1{\downarrow} = 1$
- $a{\downarrow} = a$
- $(e+f){\downarrow} = e{\downarrow} + f{\downarrow}$
- $(e \cdot f){\downarrow} = e{\downarrow} \cdot f{\downarrow}$

... but do they exist?

Let's try and compute the closure by induction:

- $0\downarrow = 0$
- $1\downarrow = 1$
- $a\downarrow = a$
- $(e + f)\downarrow = e\downarrow + f\downarrow$
- $(e \cdot f)\downarrow = e\downarrow \cdot f\downarrow$
- $(e^\star)\downarrow = e\downarrow^\star$

… but do they exist?

Let's try and compute the closure by induction:

- $0{\downarrow} = 0$
- $1{\downarrow} = 1$
- $a{\downarrow} = a$
- $(e + f){\downarrow} = e{\downarrow} + f{\downarrow}$
- $(e \cdot f){\downarrow} = e{\downarrow} \cdot f{\downarrow}$
- $(e^\star){\downarrow} = e{\downarrow}^\star$
- $(e \parallel f){\downarrow} = ???$

… But do they exist?

Let's try and compute the closure by induction:

- $0{\downarrow} = 0$
- $1{\downarrow} = 1$
- $a{\downarrow} = a$
- $(e + f){\downarrow} = e{\downarrow} + f{\downarrow}$
- $(e \cdot f){\downarrow} = e{\downarrow} \cdot f{\downarrow}$
- $(e^\star){\downarrow} = e{\downarrow}^\star$
- $(e \parallel f){\downarrow} = ???$

We strengthen our induction, by assuming that we have closures for

… But do they exist?

Let's try and compute the closure by induction:

- ☞ $0{\downarrow} = 0$
- ☞ $1{\downarrow} = 1$
- ☞ $a{\downarrow} = a$
- ☞ $(e + f){\downarrow} = e{\downarrow} + f{\downarrow}$
- ☞ $(e \cdot f){\downarrow} = e{\downarrow} \cdot f{\downarrow}$
- ☞ $(e^\star){\downarrow} = e{\downarrow}^\star$
- ☞ $(e \parallel f){\downarrow} = ???$

We strengthen our induction, by assuming that we have closures for

1) every <u>strict subterm</u> of $e \parallel f$,

... but do they exist?

Let's try and compute the closure by induction:

- ☞ $0{\downarrow} = 0$
- ☞ $1{\downarrow} = 1$
- ☞ $a{\downarrow} = a$
- ☞ $(e + f){\downarrow} = e{\downarrow} + f{\downarrow}$
- ☞ $(e \cdot f){\downarrow} = e{\downarrow} \cdot f{\downarrow}$
- ☞ $(e^\star){\downarrow} = e{\downarrow}^\star$
- ☞ $(e \parallel f){\downarrow} = ???$

We strengthen our induction, by assuming that we have closures for

1) every <u>strict subterm</u> of $e \parallel f$,
2) every term with <u>smaller width</u> than $e \parallel f$.

... but do they exist?

Let's try and compute the closure by induction:

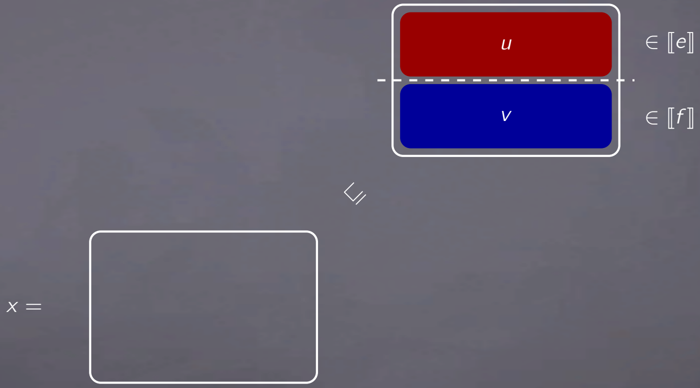
- $0{\downarrow} = 0$
- $1{\downarrow} = 1$
- $a{\downarrow} = a$
- $(e + f){\downarrow} = e{\downarrow} + f{\downarrow}$
- $(e \cdot f){\downarrow} = e{\downarrow} \cdot f{\downarrow}$
- $(e^\star){\downarrow} = e{\downarrow}^\star$
- $(e \parallel f){\downarrow} = ???$

We strengthen our induction, by assuming that we have closures for

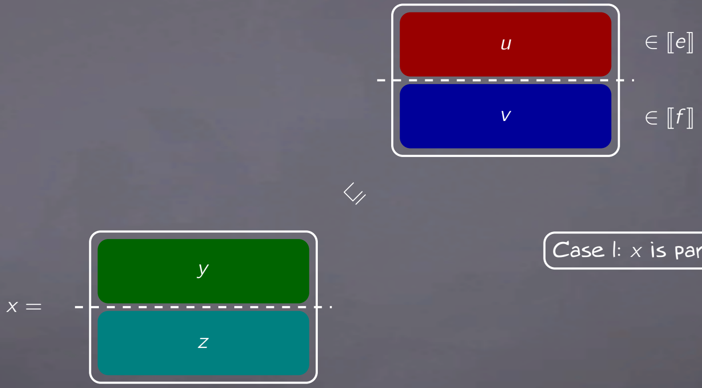
1) every strict subterm of $e \parallel f$,
2) every term with smaller width than $e \parallel f$.

We write the corresponding strict ordering $\prec$.

Who's smaller than a parallel product?

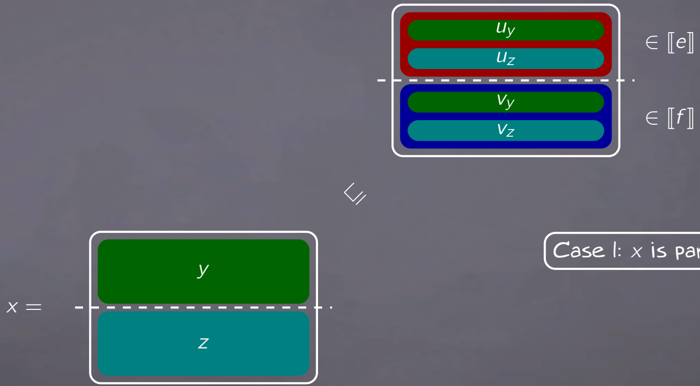


Who's smaller than a parallel product?



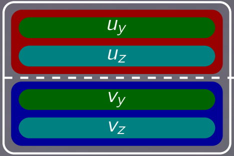
Case 1: x is parallel.

Who's smaller than a parallel product?



Case 1: x is parallel.

Who's smaller than a parallel product?

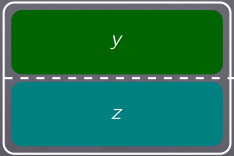


$\in [\![e]\!]$

$\in [\![f]\!]$

$\sqsubseteq_{\parallel}$

$x =$



$\sqsubseteq u_y \parallel v_y$

$\sqsubseteq u_z \parallel v_z$

Case 1: x is parallel.

Parallel splicing and preclosure

Parallel splicing

Δ_e is a finite relation over $\mathbb{E}$ such that:

$$u \parallel v \in [\![e]\!] \Leftrightarrow \exists l \; \Delta_e \; r : u \in [\![l]\!] \wedge v \in [\![r]\!].$$

Parallel splicing and preclosure

Parallel splicing

Δ_e is a finite relation over $\mathbb{E}$ such that:

$$u \parallel v \in [\![e]\!] \Leftrightarrow \exists l \; \Delta_e \; r : u \in [\![l]\!] \wedge v \in [\![r]\!].$$

$$e \odot f = e \parallel f + \sum_{l \Delta_{e \parallel f} r} (l{\downarrow}) \parallel (r{\downarrow}) \,.$$

Parallel splicing and preclosure

Parallel splicing

Δ_e is a finite relation over $\mathbb{E}$ such that:

$$u \parallel v \in [\![e]\!] \Leftrightarrow \exists l \; \Delta_e \; r : u \in [\![l]\!] \wedge v \in [\![r]\!].$$

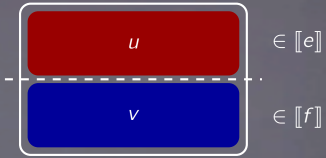
$$e \odot f = e \parallel f + \sum_{l \Delta_{e \parallel f} r} (l{\downarrow}) \parallel (r{\downarrow}).$$

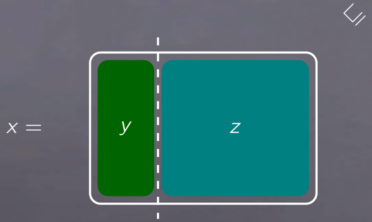
Lemma

$$u \parallel v \in {}^{\sqsubseteq}[\![e \parallel f]\!] \Leftrightarrow u \parallel v \in [\![e \odot f]\!].$$

$$\mathrm{CKA} \vdash e \odot f = e \parallel f.$$

Who's smaller than a parallel product?

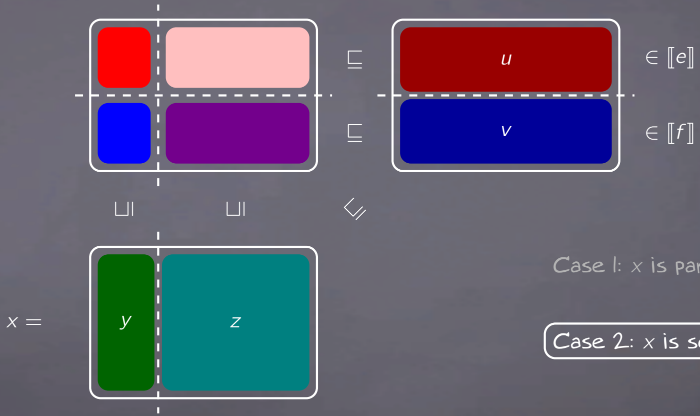




Case 1: x is parallel.

$\boxed{\text{Case 2: } x \text{ is sequential.}}$

Who's smaller than a parallel product?



Case 1: x is parallel.

$\boxed{\text{Case 2: } x \text{ is sequential.}}$

Sequential splicing

Sequential splicing

∇_e is a finite relation over $\mathbb{E}$ such that:

$$u \cdot v \in [\![e]\!] \Leftrightarrow \exists l \; \nabla_e \; r : u \in [\![l]\!] \wedge v \in [\![r]\!].$$

Sequential splicing

Sequential splicing

∇_e is a finite relation over $\mathbb{E}$ such that:

$$u \cdot v \in [\![e]\!] \Leftrightarrow \exists l \ \nabla_e \ r : u \in [\![l]\!] \wedge v \in [\![r]\!].$$

$$u \cdot v \in {}^{\sqsubseteq}[\![e \parallel f]\!] \Leftrightarrow u \cdot v \in [\![e \parallel f + \sum_{\substack{l_e \ \nabla_e \ r_e \\ l_f \ \nabla_f \ r_f}} (l_e \odot l_f) \cdot (r_e \parallel r_f){\downarrow}]\!]$$

Sequential splicing

Sequential splicing

∇_e is a finite relation over $\mathbb{E}$ such that:

$$u \cdot v \in [\![e]\!] \Leftrightarrow \exists l \nabla_e r : u \in [\![l]\!] \wedge v \in [\![r]\!].$$

$$u \cdot v \in {}^{\sqsubseteq}[\![e \parallel f]\!] \Leftrightarrow u \cdot v \in [\![e \parallel f + \sum_{\substack{l_e \nabla_e r_e \\ l_f \nabla_f r_f}} (l_e \odot l_f) \cdot (r_e \parallel r_f)\downarrow]\!]$$

Problem: $r_e \parallel r_f$ is not always smaller than $e \parallel f$...

And then, some magic happens...

And then, some magic happens...

☞ We repeat the construction to get successive equations, involving closures.

And then, some magic happens...

- We repeat the construction to get successive equations, involving closures.
- Only a finite number of unknown closures appear.

And then, some magic happens...

- ☞ We repeat the construction to get successive equations, involving closures.
- ☞ Only a finite number of unknown closures appear.
- ☞ These equations can be structured as a linear system.

And then, some magic happens...

- We repeat the construction to get successive equations, involving closures.
- Only a finite number of unknown closures appear.
- These equations can be structured as a linear system.
- With a fancy fixpoint theorem, we compute the least solution of the system.

And then, some magic happens...

- We repeat the construction to get successive equations, involving closures.
- Only a finite number of unknown closures appear.
- These equations can be structured as a linear system.
- With a fancy fixpoint theorem, we compute the least solution of the system.
- This solution is a closure.

Completeness of CKA

Lemma

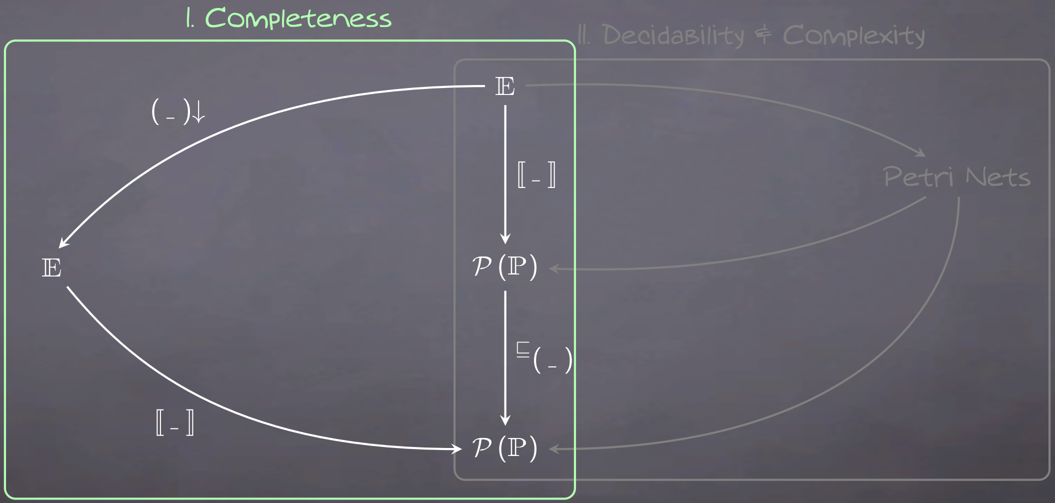
Every series-rational expression admits a closure.

Theorem

$$\mathrm{CKA} \vdash e = f \Leftrightarrow {}^{\sqsubseteq}[\![e]\!] = {}^{\sqsubseteq}[\![f]\!].$$

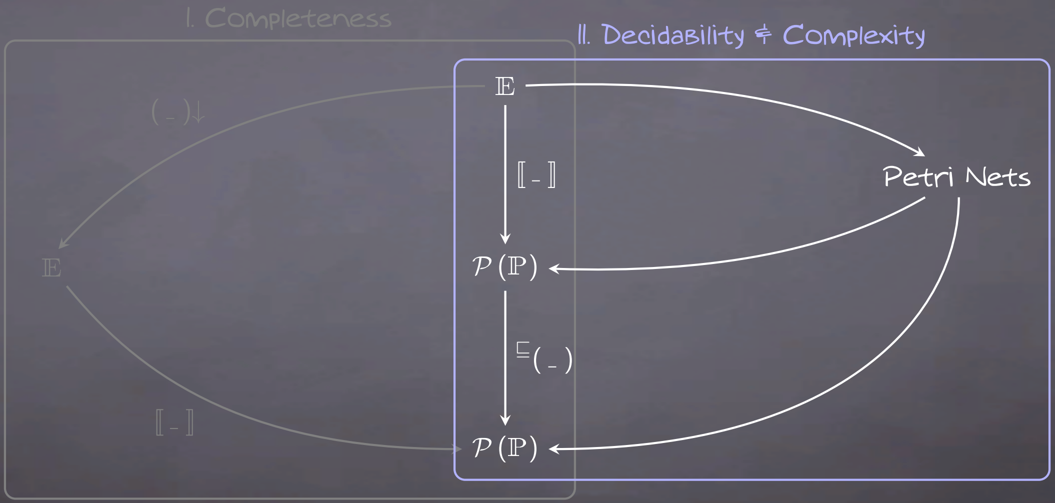
Implementation: `https://doi.org/10.5281/zenodo.926651`.

Outline


I. Completeness
II. Decidability & Complexity
$\mathbb{E}$
$(\_)\downarrow$
$[\![\_]\!]$
$\mathbb{E}$
$\mathcal{P}(\mathbb{P})$
${}^{\sqsubseteq}(\_)$
$[\![\_]\!]$
$\mathcal{P}(\mathbb{P})$
Petri Nets

Outline

Brunet, Pous & Struth, "On decidability of concurrent Kleene algebra", 2017



Two decision problems

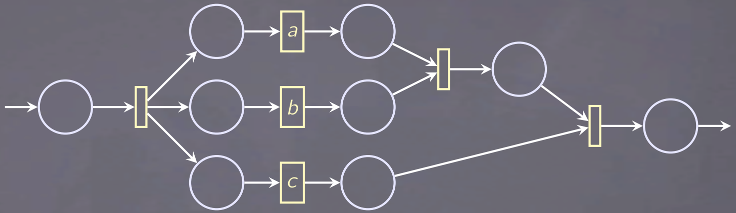
biKA

Given two expressions e, f, are $[\![e]\!]$ and $[\![f]\!]$ equal?

CKA

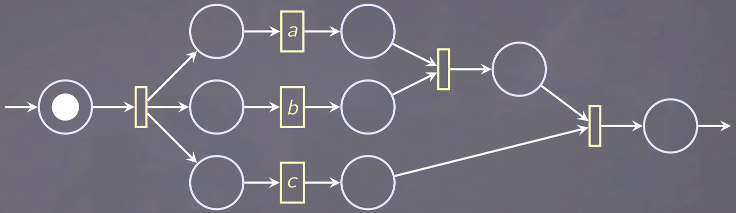
Given two expressions e, f, are ${}^{\sqsubseteq}[\![e]\!]$ and ${}^{\sqsubseteq}[\![f]\!]$ equal?

Labelled Petri nets



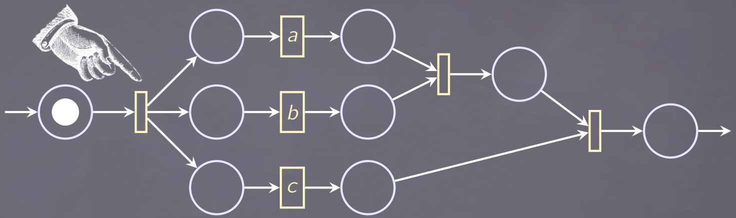
skip

Labelled Petri nets



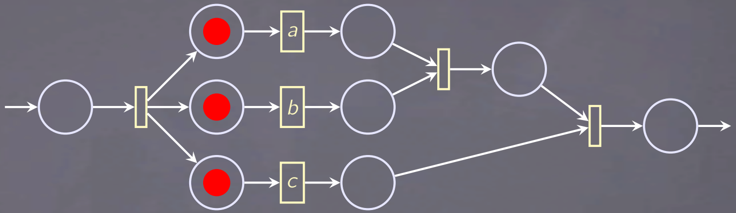
skip

Labelled Petri nets



⏩ skip

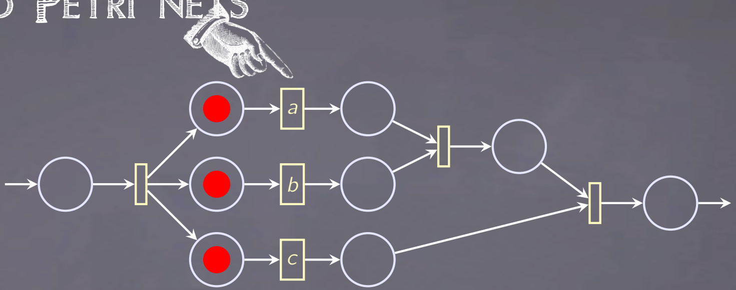
Labelled Petri nets



⏩ skip

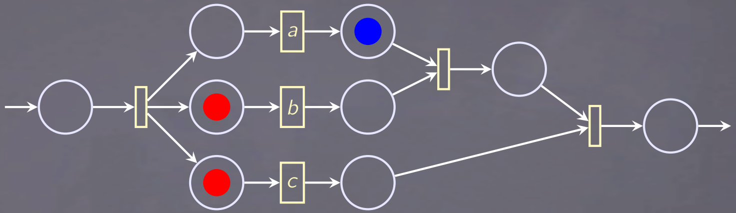
τ

Labelled Petri nets



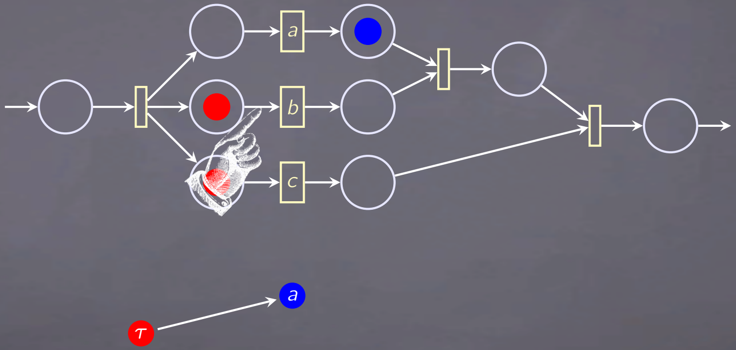
▸▸ skip

Labelled Petri nets



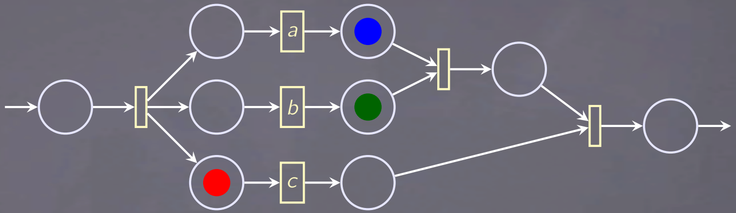
skip

Labelled Petri nets



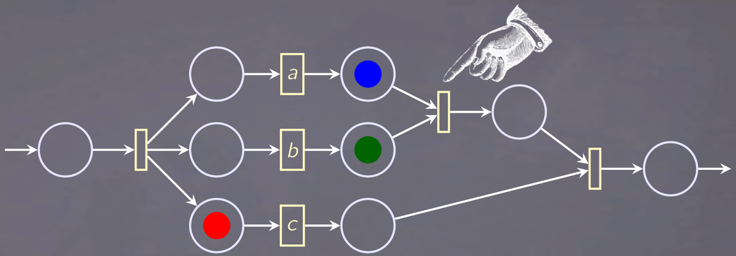
⏩ skip

Labelled Petri nets



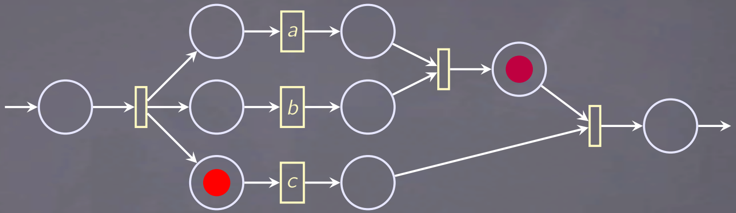
⏩ skip

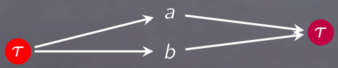
Labelled Petri nets



skip

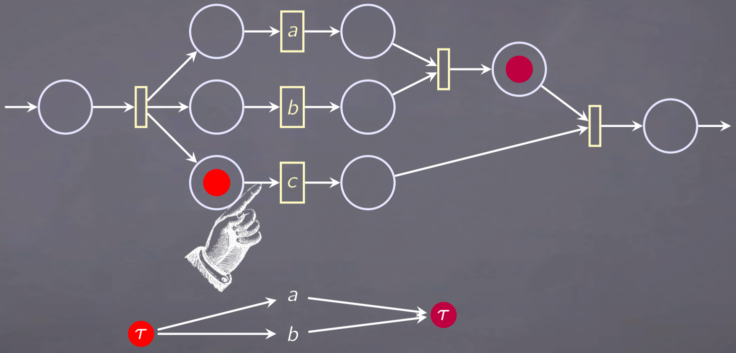
Labelled Petri nets





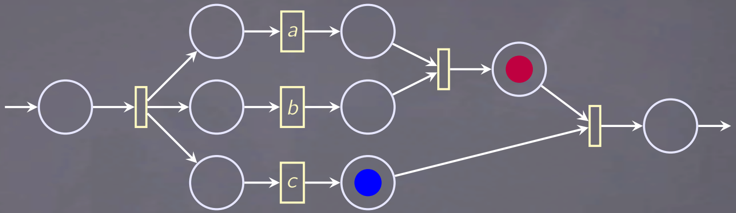
⏩ skip

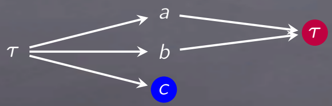
Labelled Petri nets



⏩ skip

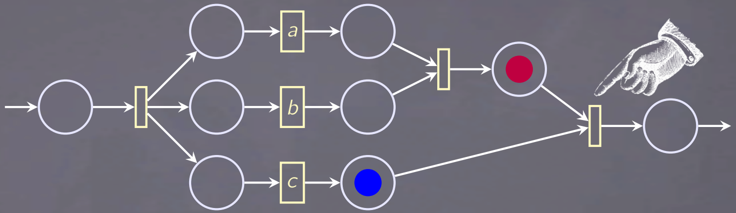
Labelled Petri nets

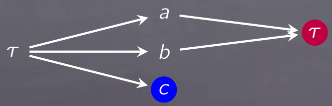




⏩ skip

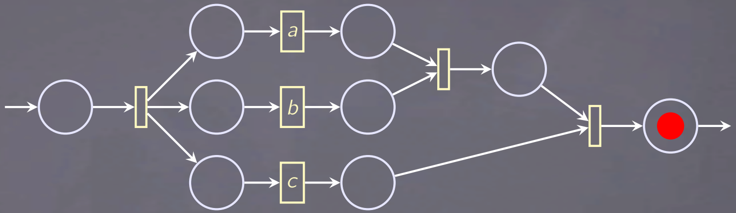
Labelled Petri nets





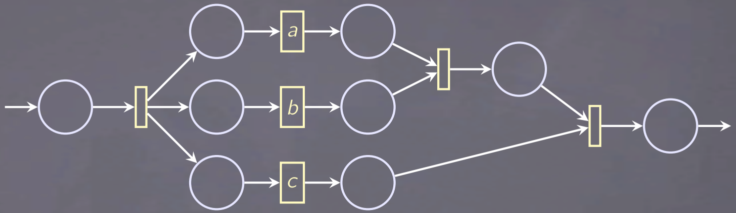
⏩ skip

Labelled Petri nets



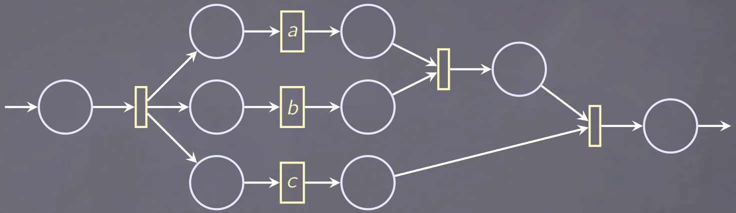
⏩ skip

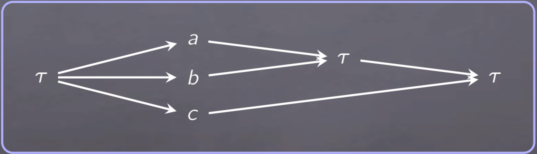
Labelled Petri nets



skip

Labelled Petri nets

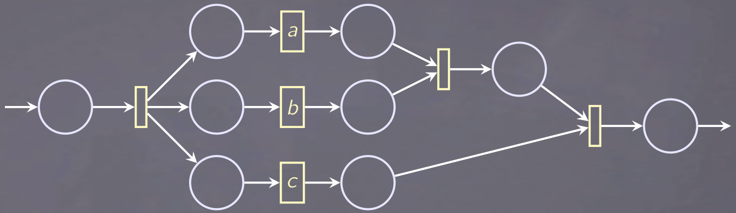


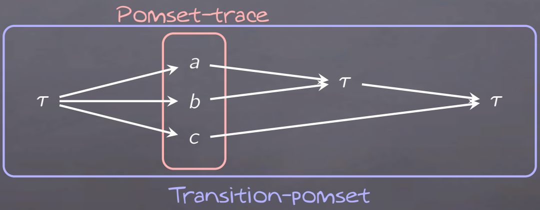


Transition-pomset

⏩ skip

Labelled Petri nets





⏩ skip

Recognisable pomset languages

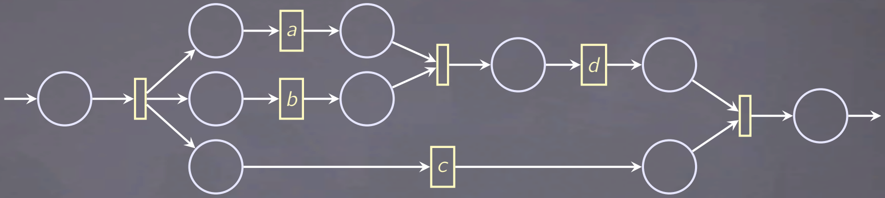
Language generated by a net

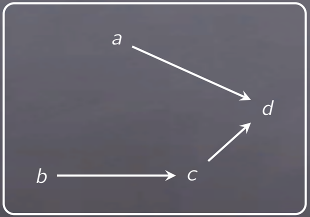
$[\![\mathcal{N}]\!]$ is the set of pomset-traces of accepting runs of $\mathcal{N}$.

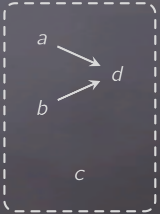
Definition

A set of pomsets S is a recognisable pomset language if there is a net $\mathcal{N}$ such that $S = [\![\mathcal{N}]\!]$.

Reading a pomset in a net

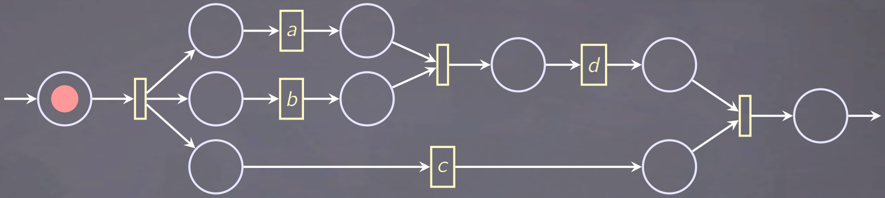


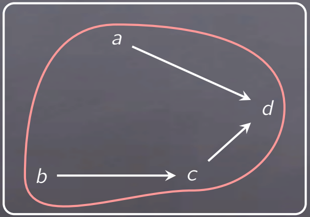


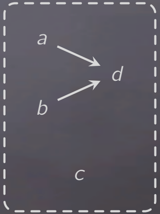


skip

Reading a pomset in a net

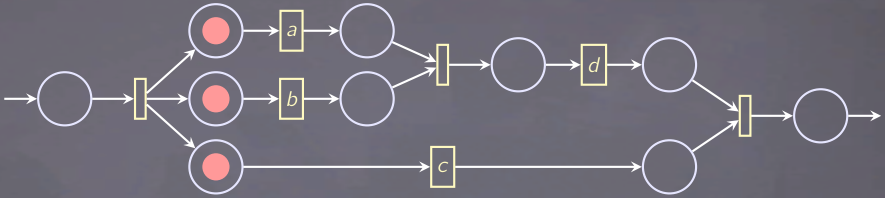


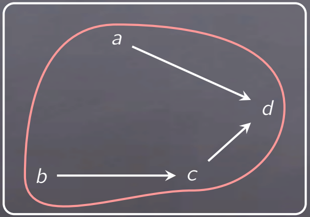


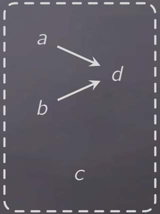


skip

Reading a pomset in a net

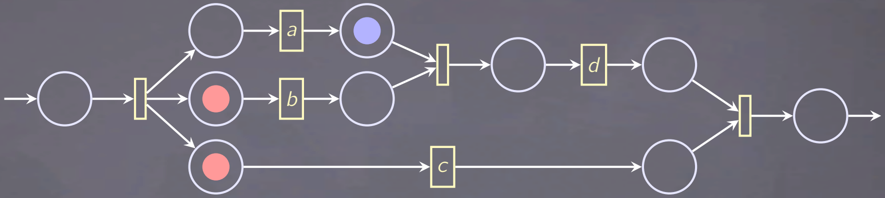


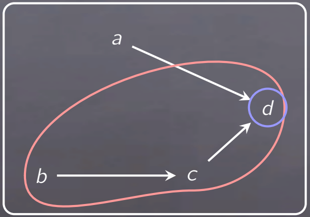


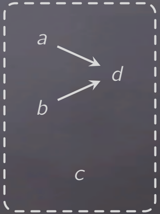


skip

Reading a pomset in a net

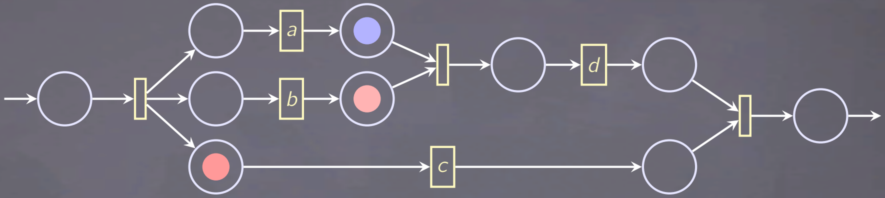




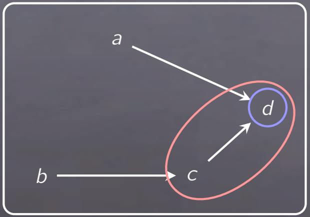


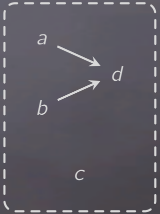
skip

Reading a pomset in a net

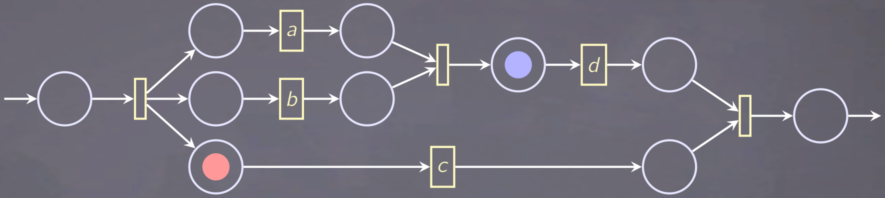


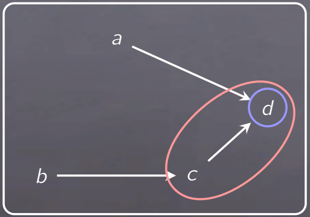
▸▸ skip

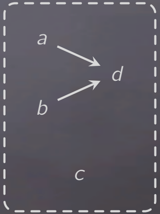




Reading a pomset in a net

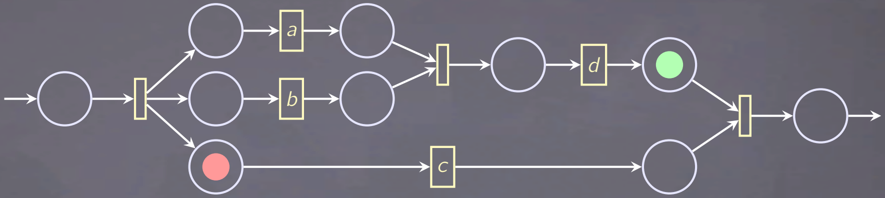


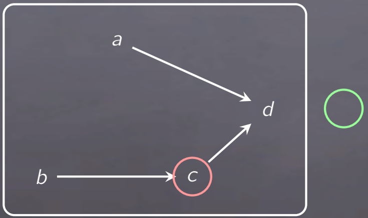


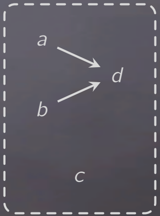


skip

Reading a pomset in a net

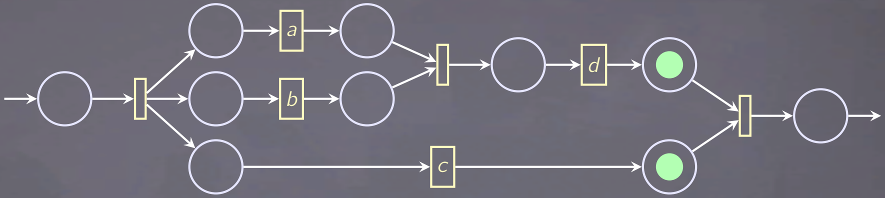


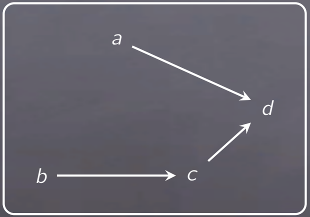


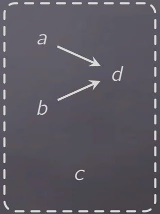


skip

Reading a pomset in a net

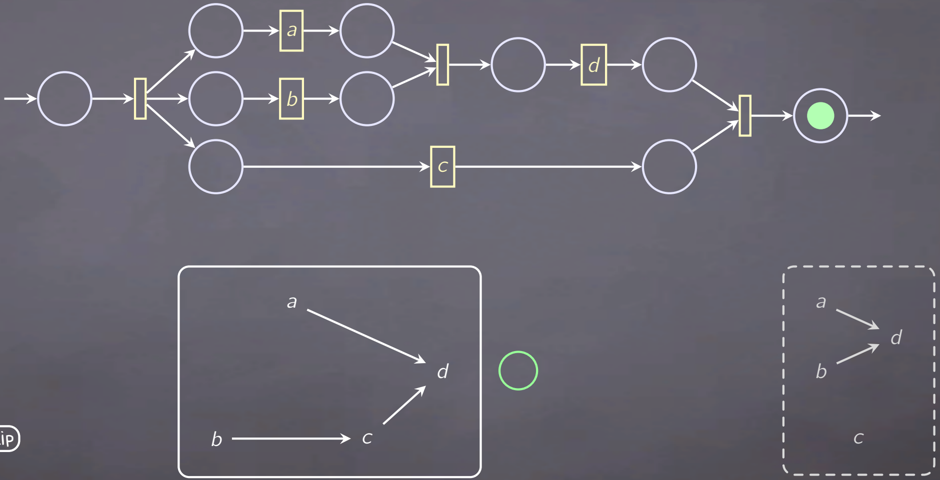






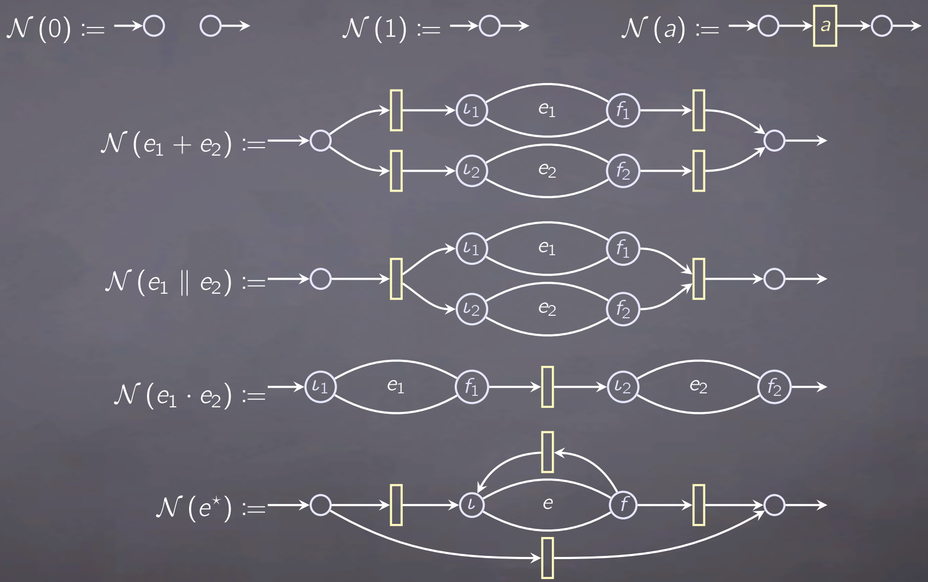
skip

Reading a pomset in a net



skip

From expressions to automata



Solving biKA

Lemma

$$[\![e]\!] = [\![\mathcal{N}(e)]\!].$$

Corollary: Rational pomset languages are recognisable.

Solving biKA

Lemma

$$[\![e]\!] = [\![\mathcal{N}(e)]\!].$$

Corollary: Rational pomset languages are recognisable.

Theorem

Testing containment of pomset-trace languages of two Petri nets is an ExpSpace-complete problem.

☞ Jategaonkar & Meyer, "Deciding true concurrency equivalences on safe, finite nets", 1996.

Solving biKA

Lemma

$$[\![e]\!] = [\![\mathcal{N}(e)]\!].$$

Corollary: Rational pomset languages are recognisable.

Theorem

Testing containment of pomset-trace languages of two Petri nets is an ExpSpace-complete problem.

☞ Jategaonkar & Meyer, "Deciding true concurrency equivalences on safe, finite nets", 1996.

Corollary: The problem biKA lies in the class ExpSpace.

What about CKA?

$$^{\sqsubseteq}[\![e]\!] = {}^{\sqsubseteq}[\![f]\!]$$

What about CKA?

$$^{\sqsubseteq}[\![e]\!] = {^{\sqsubseteq}[\![f]\!]} \Leftrightarrow \quad {^{\sqsubseteq}[\![e]\!]} \subseteq {^{\sqsubseteq}[\![f]\!]} \quad \wedge \quad {^{\sqsubseteq}[\![e]\!]} \supseteq {^{\sqsubseteq}[\![f]\!]}$$

What about CKA?

$$
\begin{aligned}
{}^{\sqsubseteq}[\![e]\!] = {}^{\sqsubseteq}[\![f]\!] \Leftrightarrow \quad & {}^{\sqsubseteq}[\![e]\!] \subseteq {}^{\sqsubseteq}[\![f]\!] & \wedge \quad & {}^{\sqsubseteq}[\![e]\!] \supseteq {}^{\sqsubseteq}[\![f]\!] \\
\Leftrightarrow \quad & [\![e]\!] \subseteq {}^{\sqsubseteq}[\![f]\!] & \wedge \quad & {}^{\sqsubseteq}[\![e]\!] \supseteq [\![f]\!]
\end{aligned}
$$

What about CKA?

$$
\begin{aligned}
{}^{\sqsubseteq}[\![e]\!] = {}^{\sqsubseteq}[\![f]\!] &\Leftrightarrow \quad {}^{\sqsubseteq}[\![e]\!] \subseteq {}^{\sqsubseteq}[\![f]\!] \quad \wedge \quad {}^{\sqsubseteq}[\![e]\!] \supseteq {}^{\sqsubseteq}[\![f]\!] \\
&\Leftrightarrow \quad [\![e]\!] \subseteq {}^{\sqsubseteq}[\![f]\!] \quad \wedge \quad {}^{\sqsubseteq}[\![e]\!] \supseteq [\![f]\!] \\
&\Leftrightarrow [\![\mathcal{N}(e)]\!] \subseteq {}^{\sqsubseteq}[\![\mathcal{N}(f)]\!] \wedge {}^{\sqsubseteq}[\![\mathcal{N}(e)]\!] \supseteq [\![\mathcal{N}(f)]\!]
\end{aligned}
$$

What about CKA?

$$\begin{aligned}
{}^{\sqsubseteq}[\![e]\!] = {}^{\sqsubseteq}[\![f]\!] &\Leftrightarrow \quad {}^{\sqsubseteq}[\![e]\!] \subseteq {}^{\sqsubseteq}[\![f]\!] \quad \wedge \quad {}^{\sqsubseteq}[\![e]\!] \supseteq {}^{\sqsubseteq}[\![f]\!] \\
&\Leftrightarrow \quad [\![e]\!] \subseteq {}^{\sqsubseteq}[\![f]\!] \quad \wedge \quad {}^{\sqsubseteq}[\![e]\!] \supseteq [\![f]\!] \\
&\Leftrightarrow [\![\mathcal{N}(e)]\!] \subseteq {}^{\sqsubseteq}[\![\mathcal{N}(f)]\!] \wedge {}^{\sqsubseteq}[\![\mathcal{N}(e)]\!] \supseteq [\![\mathcal{N}(f)]\!]
\end{aligned}$$

Problem

Let $\mathcal{N}_1, \mathcal{N}_2$ be well behaved nets. Is it true that for every run R_1 of $\mathcal{N}_1$ there is a run R_2 in $\mathcal{N}_2$ such that

$$\mathcal{P}om(R_1) \sqsubseteq \mathcal{P}om(R_2)?$$

Idea of the algorithm

Problem

Let $\mathcal{N}_1, \mathcal{N}_2$ be well behaved nets. Is it true that for every run R_1 of $\mathcal{N}_1$ there is a run R_2 in $\mathcal{N}_2$ such that

$$\mathcal{P}om(R_1) \sqsubseteq \mathcal{P}om(R_2)?$$

Idea of the algorithm

Problem

Let $\mathcal{N}_1, \mathcal{N}_2$ be well behaved nets. Is it true that for every run R_1 of $\mathcal{N}_1$ there is a run R_2 in $\mathcal{N}_2$ such that

$$\mathcal{P}om(R_1) \sqsubseteq \mathcal{P}om(R_2)?$$

☞ Build an automaton $\mathscr{A}_1$ for $[\![\mathcal{N}_1]\!]$

Idea of the algorithm

Problem

Let $\mathcal{N}_1, \mathcal{N}_2$ be well behaved nets. Is it true that for every run R_1 of $\mathcal{N}_1$ there is a run R_2 in $\mathcal{N}_2$ such that

$$\mathcal{P}om(R_1) \sqsubseteq \mathcal{P}om(R_2)?$$

- ☞ Build an automaton $\mathscr{A}_1$ for $[\![\mathcal{N}_1]\!]$
- ☞ Build an automaton $\mathscr{A}_2$ for $[\![\mathcal{N}_1]\!] \cap^{\sqsubseteq} [\![\mathcal{N}_2]\!]$

Idea of the algorithm

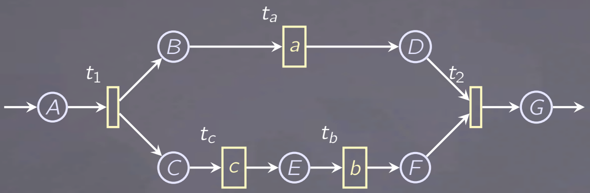
Problem

Let $\mathcal{N}_1, \mathcal{N}_2$ be well behaved nets. Is it true that for every run R_1 of $\mathcal{N}_1$ there is a run R_2 in $\mathcal{N}_2$ such that

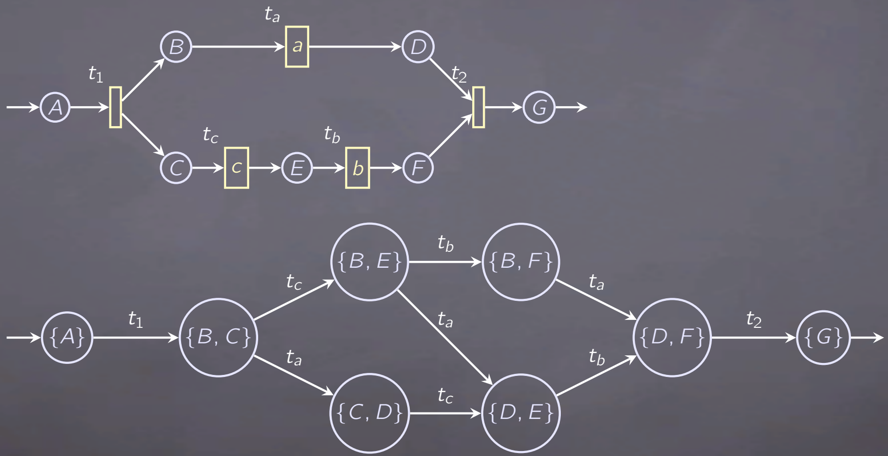
$$\mathcal{P}om(R_1) \sqsubseteq \mathcal{P}om(R_2)?$$

- ☞ Build an automaton $\mathscr{A}_1$ for $[\![\mathcal{N}_1]\!]$
- ☞ Build an automaton $\mathscr{A}_2$ for $[\![\mathcal{N}_1]\!] \cap^{\sqsubseteq} [\![\mathcal{N}_2]\!]$
- ☞ $[\![\mathcal{N}_1]\!] \subseteq^{\sqsubseteq} [\![\mathcal{N}_2]\!]$ if and only if $\mathcal{L}(\mathscr{A}_1) = \mathcal{L}(\mathscr{A}_2)$.

Transition automaton

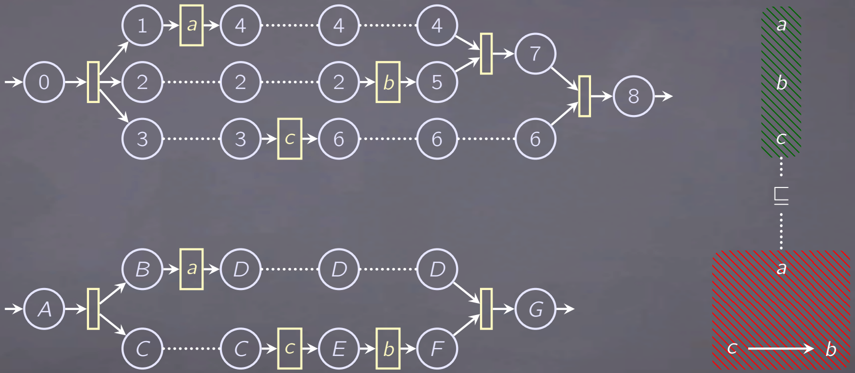


Transition automaton



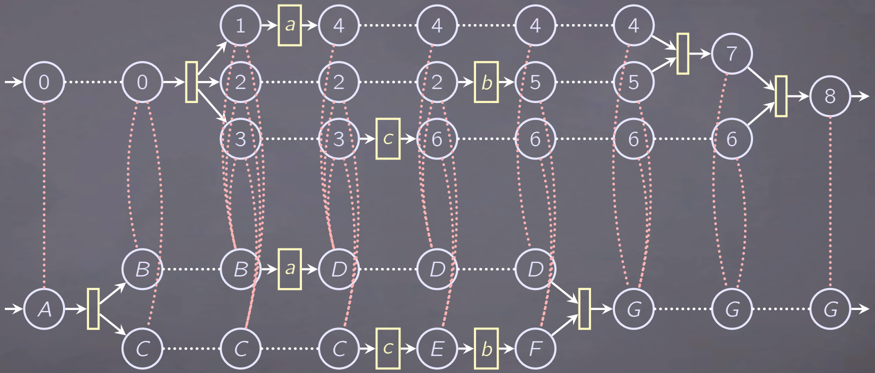
Massaging runs

Massaging runs

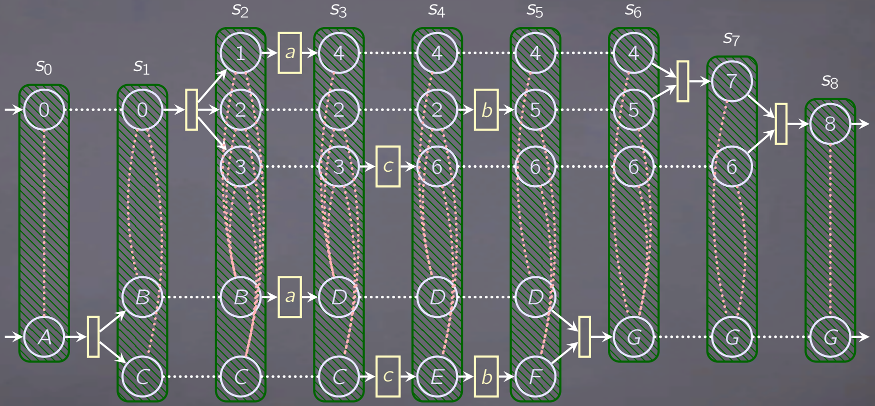


Massaging runs

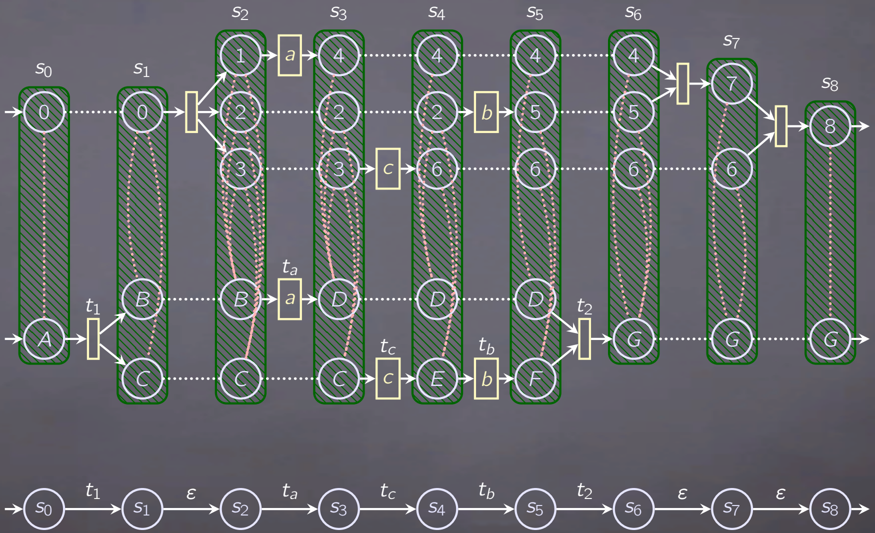
Massaging runs



Massaging runs



Massaging runs



Reduction to automata

Let $\mathcal{N}_1$ and $\mathcal{N}_2$ be some polite nets, of size n, m.

Lemma

There is an automaton $\mathscr{A}(\mathcal{N}_1)$ with $\mathcal{O}(2^n)$ states that recognises the set of accepting runs in $\mathcal{N}_1$.

Reduction to automata

Let $\mathcal{N}_1$ and $\mathcal{N}_2$ be some polite nets, of size n, m.

Lemma

There is an automaton $\mathscr{A}(\mathcal{N}_1)$ with $\mathcal{O}(2^n)$ states that recognises the set of accepting runs in $\mathcal{N}_1$.

Lemma

There is an automaton $\mathcal{N}_1 \prec \mathcal{N}_2$ with $\mathcal{O}(2^{n+m+nm})$ states that recognises the set of accepting runs in $\mathcal{N}_1$ whose pomset belongs to ${}^{\sqsubseteq}[\![\mathcal{N}_2]\!]$.

Decidability + Complexity

Theorem

Given two expressions $e, f \in \mathbb{E}$, we can test if $[\![e]\!] \subseteq^{\sqsubseteq} [\![f]\!]$ in ExpSpace.

Proof.

1) Build $\mathcal{N}(e)$ and $\mathcal{N}(f)$;
2) Build $\mathscr{A}(\mathcal{N}(e))$ and $\mathcal{N}(e) \prec \mathcal{N}(f)$;
3) compare them.

$\square$

Decidability + Complexity

Theorem

Given two expressions $e, f \in \mathbb{E}$, we can test if $[\![e]\!] \subseteq^{\sqsubseteq} [\![f]\!]$ in ExpSpace.

<u>Proof.</u>

1) Build $\mathcal{N}(e)$ and $\mathcal{N}(f)$;
2) Build $\mathscr{A}(\mathcal{N}(e))$ and $\mathcal{N}(e) \prec \mathcal{N}(f)$;
3) compare them.

$\square$

Theorem

The problem CKA is ExpSpace-hard.

(Universality problem for regular expressions with interleaving)

☞ Mayer & Stockmeyer, "The complexity of word problems – this time with interleaving", 1994.

Decidability + Complexity

Theorem

Given two expressions $e, f \in \mathbb{E}$, we can test if $[\![e]\!] \subseteq^{\sqsubseteq} [\![f]\!]$ in ExpSpace.

Proof.

1) Build $\mathcal{N}(e)$ and $\mathcal{N}(f)$;
2) Build $\mathscr{A}(\mathcal{N}(e))$ and $\mathcal{N}(e) \prec \mathcal{N}(f)$;
3) compare them.

$\square$

Theorem

The problem CKA is ExpSpace-hard.

(Universality problem for regular expressions with interleaving)

☞ Mayer & Stockmeyer, "The complexity of word problems – this time with interleaving", 1994.

Corollary: The problem CKA is ExpSpace-complete.

Outline

$\mathbb{E}$

Outline

$$
\begin{tikzcd}
\mathbb{E} \arrow[d, "[\![\_]\!]"] \\
\mathcal{P}(\mathbb{P})
\end{tikzcd}
$$

Outline

$$\begin{tikzcd} \mathbb{E} \arrow[d, "[\![\_]\!]"] \\ \mathcal{P}(\mathbb{P}) \arrow[d, "{}^{\sqsubseteq}(\_)"] \\ \mathcal{P}(\mathbb{P}) \end{tikzcd}$$

Outline

$$
\begin{tikzcd}
 & \mathbb{E} \arrow[dl, bend right, "(\_)\!\downarrow"'] \arrow[d, "\llbracket \_ \rrbracket"] \\
\mathbb{E} & \mathcal{P}(\mathbb{P}) \arrow[d, "{}^{\sqsubseteq}(\_)"] \\
 & \mathcal{P}(\mathbb{P})
\end{tikzcd}
$$

Outline

$$
\begin{tikzcd}
 & \mathbb{E} \arrow[dl, bend right, "(\_)\downarrow"'] \arrow[d, "\llbracket \_ \rrbracket"] \\
\mathbb{E} \arrow[dr, bend right, "\llbracket \_ \rrbracket"'] & \mathcal{P}(\mathbb{P}) \arrow[d, "{}^{\sqsubseteq}(\_)"] \\
 & \mathcal{P}(\mathbb{P})
\end{tikzcd}
$$

Outline

1. Completeness

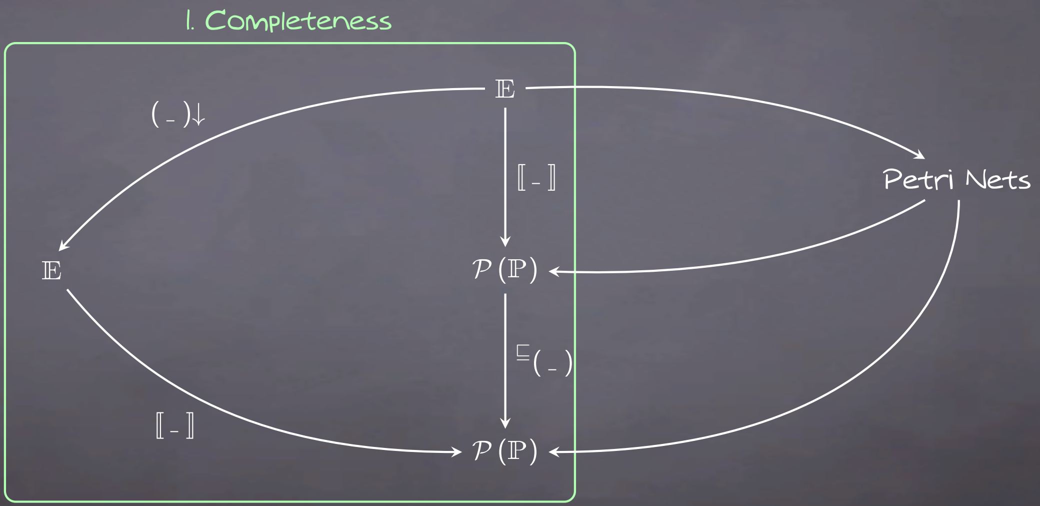
$$
\begin{tikzcd}
 & \mathbb{E} \arrow[dl, bend right, "(\_)\downarrow"'] \arrow[d, "[\![\_]\!]"] \\
\mathbb{E} \arrow[dr, bend right, "[\![\_]\!]"'] & \mathcal{P}(\mathbb{P}) \arrow[d, "{}^{\sqsubseteq}(\_)"] \\
 & \mathcal{P}(\mathbb{P})
\end{tikzcd}
$$

Outline

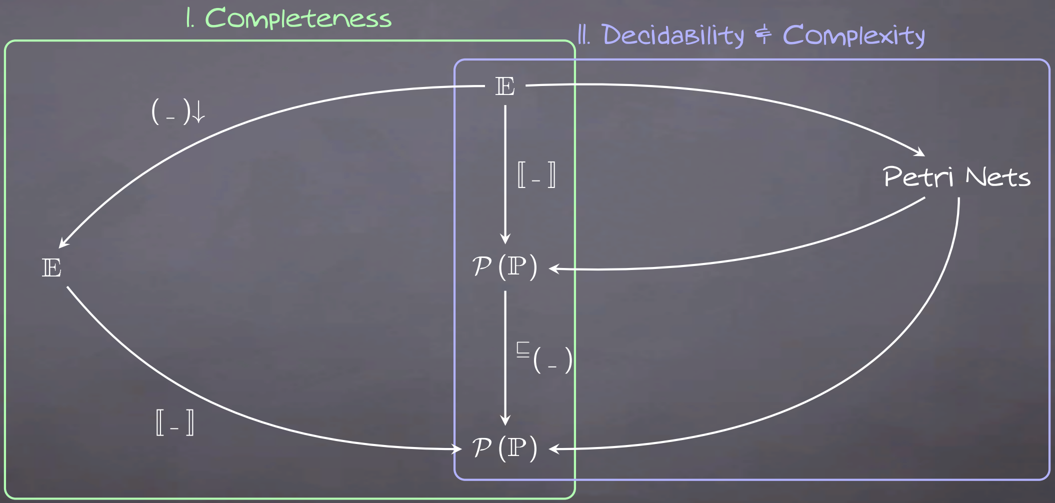
1. Completeness

$$\begin{tikzcd}
 & \mathbb{E} \arrow[dl, bend right, "(_)\downarrow"'] \arrow[d, "{[\![_]\!]}"] & \\
\mathbb{E} \arrow[dr, bend right, "{[\![_]\!]}"'] & \mathcal{P}(\mathbb{P}) \arrow[d, "{}^{\sqsubseteq}(_)"] & \text{Petri Nets} \arrow[l, bend left] \arrow[dl, bend left] \\
 & \mathcal{P}(\mathbb{P}) &
\end{tikzcd}$$

Outline


1. Completeness
$\mathbb{E}$
$(\_)\!\downarrow$
$[\![\_]\!]$
Petri Nets
$\mathbb{E}$
$\mathcal{P}(\mathbb{P})$
${}^{\sqsubseteq}(\_)$
$[\![\_]\!]$
$\mathcal{P}(\mathbb{P})$

Outline


I. Completeness
II. Decidability & Complexity
$\mathbb{E}$
$(\_)\downarrow$
$[\![\_]\!]$
Petri Nets
$\mathbb{E}$
$\mathcal{P}(\mathbb{P})$
${}^{\sqsubseteq}(\_)$
$[\![\_]\!]$
$\mathcal{P}(\mathbb{P})$

Further questions

Further questions

☞ Can we extend the algorithm to a larger class of Petri nets?

Further questions

- Can we extend the algorithm to a larger class of Petri nets?
- What about the parallel star?

Further questions

- ☞ Can we extend the algorithm to a larger class of Petri nets?
- ☞ What about the parallel star?
- ☞ Can I have tests?

Further questions

- ☞ Can we extend the algorithm to a larger class of Petri nets?
- ☞ What about the parallel star?
- ☞ Can I have <u>tests</u>?
- ☞ Might I dream of adding <u>names</u>?

Further questions

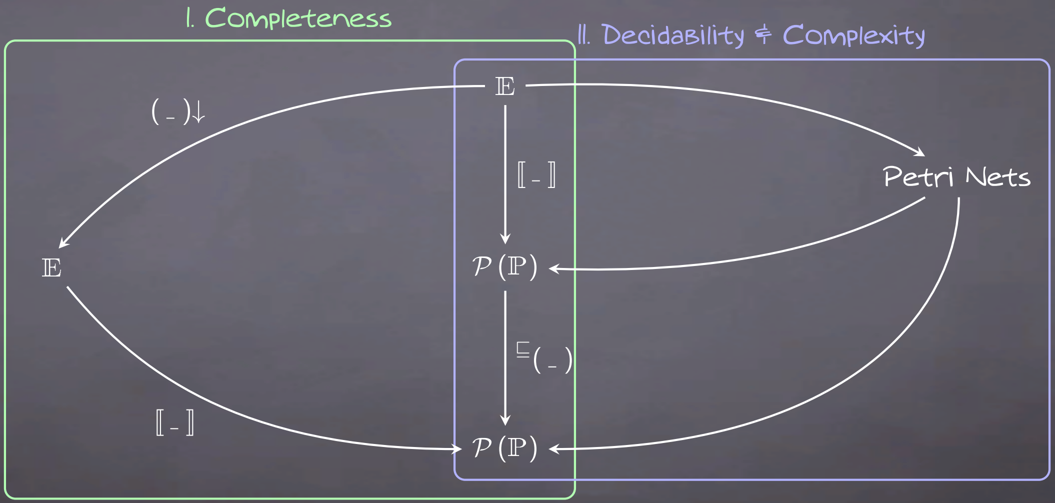
- ☞ Can we extend the algorithm to a larger class of Petri nets?
- ☞ What about the parallel star?
- ☞ Can I have tests?
- ☞ Might I dream of adding names?
- ☞ Insert you favourite operator here...

That's all folks!

Thank you!

See more at:
http://paul.brunet-zamansky.fr

Outline


I. Completeness
II. Decidability & Complexity
$\mathbb{E}$
$(\_)\downarrow$
$[\![\_]\!]$
Petri Nets
$\mathbb{E}$
$\mathcal{P}(\mathbb{P})$
${}^{\sqsubseteq}(\_)$
$[\![\_]\!]$
$\mathcal{P}(\mathbb{P})$

Outline

I. Introduction

II. Completeness

III. Decidability & Complexity

IV. Summary and Outlook