The Downward-Closure of Petri Net Languages

Peter Habermehl[1] Roland Meyer[1] Harro Wimmel[2]

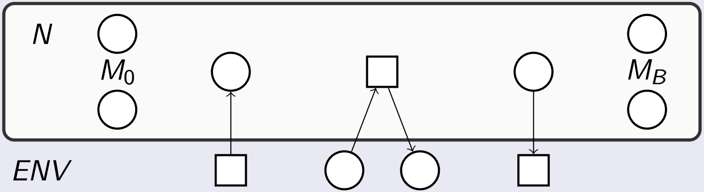
LIAFA, University Paris 7, and CNRS

University of Rostock

June 2010

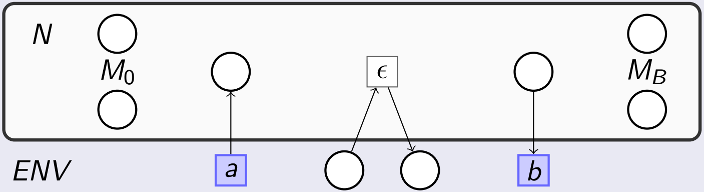
Stability Analysis

Problem



Stability Analysis

Problem

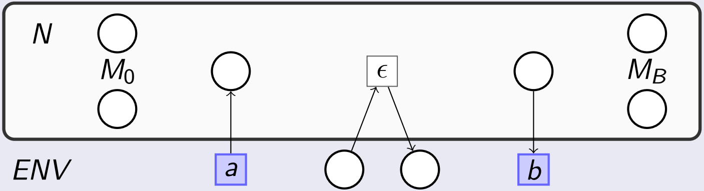


Goal

Determine influences N can tolerate without reaching M_B from M_0

Stability Analysis

Problem



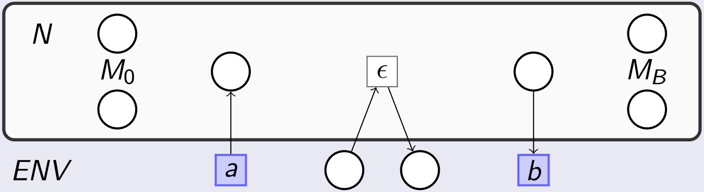
Approximate environmental behaviour $\mathcal{L}_h(N, M_0, M_B)$

Scattered embedding [Hig52] forgets letters

$$\mathcal{L}_h(N, M_0, M_B) \subseteq \mathcal{L}_h(N, M_0, M_B)\downarrow$$

Stability Analysis

Problem

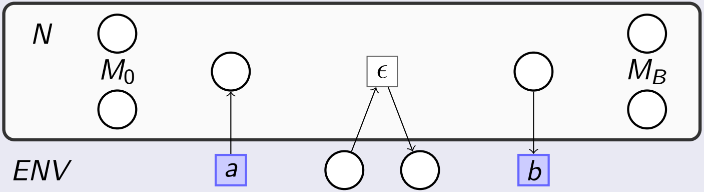


Approximate environmental behaviour $\mathcal{L}_h(N, M_0, M_B)$

$\mathcal{L}\downarrow$ regular:

Stability Analysis

Problem

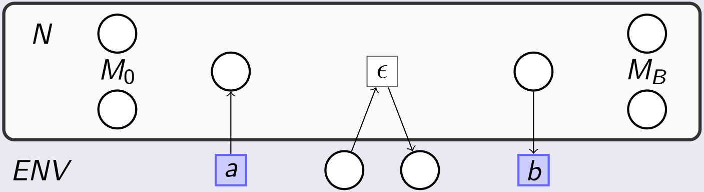


Approximate environmental behaviour $\mathcal{L}_h(N, M_0, M_B)$

$\mathcal{L}\downarrow$ regular: complement upward-closed, finite basis by wqo

Stability Analysis

Problem



Approximate environmental behaviour $\mathcal{L}_h(N, M_0, M_B)$

$\mathcal{L}\downarrow$ regular: complement upward-closed, finite basis by wqo

$$\overline{\mathcal{L}_h(N, M_0, M_B)\downarrow} \subseteq \overline{\mathcal{L}_h(N, M_0, M_B)}$$

Contribution

Result

Downward-closure of Petri net languages computable

Contribution

Result

Downward-closure of Petri net languages computable

Language types

- Ordinary $\mathcal{L}_h(N, M_0, M_f)$ accept by markings (PN)

Contribution

Result

Downward-closure of Petri net languages computable

Language types

- Ordinary $\mathcal{L}_h(N, M_0, M_f)$ accept by markings (PN)
- Terminal $\mathcal{T}_h(N, M_0)$ accept by deadlocks (TPN)

Contribution

Result

Downward-closure of Petri net languages computable

Language types

- Ordinary $\mathcal{L}_h(N, M_0, M_f)$ accept by markings (PN)
- Terminal $\mathcal{T}_h(N, M_0)$ accept by deadlocks (TPN)
- Covering $\mathcal{C}_h(N, M_0, M_f)$ accept by upward-closed sets (CPN)

Contribution

Result

Downward-closure of Petri net languages computable

Language types

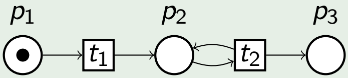
- Ordinary $\mathcal{L}_h(N, M_0, M_f)$ accept by markings (PN)
- Terminal $\mathcal{T}_h(N, M_0)$ accept by deadlocks (TPN)
- Covering $\mathcal{C}_h(N, M_0, M_f)$ accept by upward-closed sets (CPN)

Further applications

- Analysis of asynchronous systems
- Compositional verification
- Regular approximation

Petri Nets and Coverability Graphs

Petri net

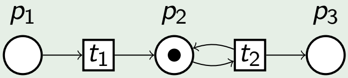


Its coverability graph [KM69]

$$\downarrow$$
$$(1, 0, 0)$$

Petri Nets and Coverability Graphs

Petri net

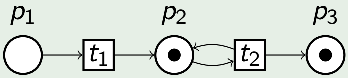


Its coverability graph [KM69]

$$(1, \overset{\downarrow}{0}, 0) \xrightarrow{t_1} (0, 1, 0)$$

Petri Nets and Coverability Graphs

Petri net

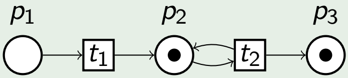


Its coverability graph [KM69]

$$(1,0,0) \xrightarrow{t_1} (0,1,0) \xrightarrow{t_2} (0,1,1)$$

Petri Nets and Coverability Graphs

Petri net

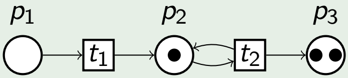


Its coverability graph [KM69]

$$(1,0,0) \xrightarrow{t_1} (0,1,0) \xrightarrow{t_2} (0,1,\omega)$$

Petri Nets and Coverability Graphs

Petri net



Its coverability graph [KM69]

$$(1,0,0) \xrightarrow{t_1} (0,1,0) \xrightarrow{t_2} (0,1,\omega) \circlearrowleft^{t_2}$$

Main Result

Definition (Ordinary Petri net language in *PN*)

$$\mathcal{L}_h(N, M_0, M_f) := \{h(\sigma) \mid M_0[\sigma\rangle M_f\}$$

Main Result

Definition (Ordinary Petri net language in *PN*)

$$\mathcal{L}_h(N, M_0, M_f) := \{h(\sigma) \mid M_0[\sigma\rangle M_f\}$$

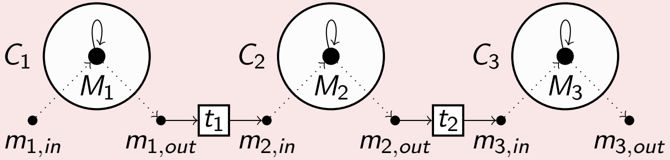
Theorem (Representation)

A regular expression ϕ is computable with

$$\mathcal{L}_h(N, M_0, M_f)\downarrow = \phi$$

Lambert's Marked Graph Transition Sequences [Lam92]

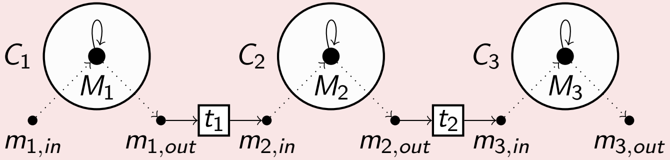
Marked graph transition sequence $G = C_1.t_1.C_2.t_2.C_3$



Properties

Lambert's Marked Graph Transition Sequences [Lam92]

Marked graph transition sequence $G = C_1.t_1.C_2.t_2.C_3$

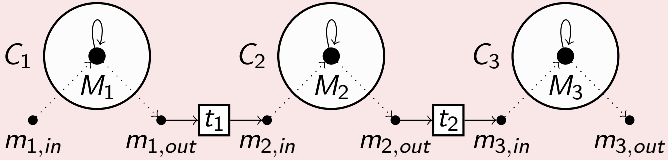


Properties

- C strongly connected coverability graph

Lambert's Marked Graph Transition Sequences [Lam92]

Marked graph transition sequence $G = C_1.t_1.C_2.t_2.C_3$

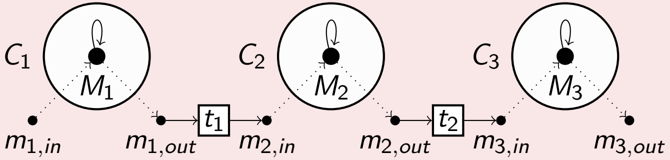


Properties

- C strongly connected coverability graph
- M initial marking

Lambert's Marked Graph Transition Sequences [Lam92]

Marked graph transition sequence $G = C_1.t_1.C_2.t_2.C_3$

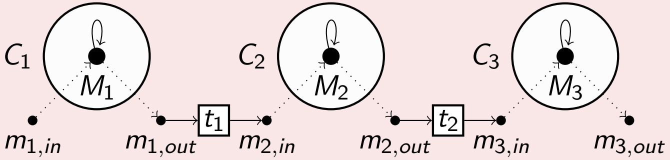


Properties

- C strongly connected coverability graph
- M initial marking, m_{in} input marking

Lambert's Marked Graph Transition Sequences [Lam92]

Marked graph transition sequence $G = C_1.t_1.C_2.t_2.C_3$

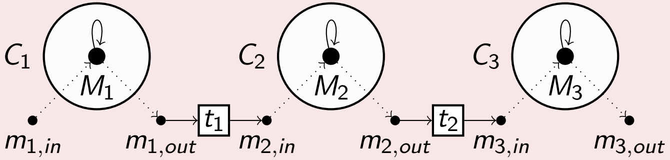


Properties

- C strongly connected coverability graph
- M initial marking, m_{in} input marking , m_{out} output marking

Lambert's Marked Graph Transition Sequences [Lam92]

Marked graph transition sequence $G = C_1.t_1.C_2.t_2.C_3$



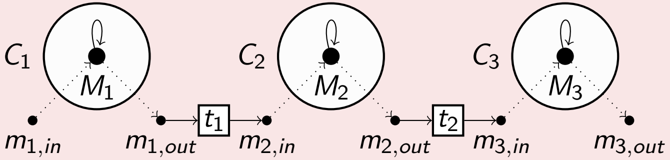
Properties

- C strongly connected coverability graph
- M initial marking, m_{in} input marking , m_{out} output marking
- Input and output less abstract $\preceq_\omega$ than initial marking

$$m_{in} \preceq_\omega M \quad \text{if} \quad m_{in}(p) = M(p) \quad \text{or} \quad M(p) = \omega$$

Lambert's Marked Graph Transition Sequences [Lam92]

Marked graph transition sequence $G = C_1.t_1.C_2.t_2.C_3$

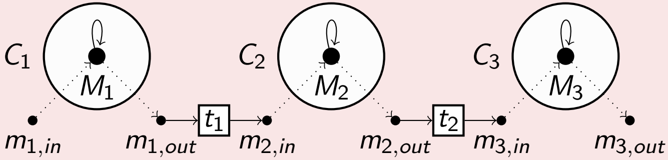


Solutions $\mathcal{L}(G)$

Transition sequence σ through $G = C_1.t_1.C_2.t_2.C_3$

Lambert's Marked Graph Transition Sequences [Lam92]

Marked graph transition sequence $G = C_1.t_1.C_2.t_2.C_3$



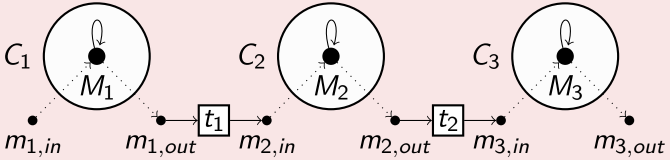
Solutions $\mathcal{L}(G)$

Transition sequence σ through $G = C_1.t_1.C_2.t_2.C_3$

- Start in $m_{1,in}$

Lambert's Marked Graph Transition Sequences [Lam92]

Marked graph transition sequence $G = C_1.t_1.C_2.t_2.C_3$



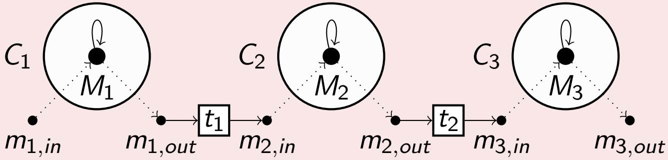
Solutions $\mathcal{L}(G)$

Transition sequence σ through $G = C_1.t_1.C_2.t_2.C_3$

- Start in $m_{1,in}$, fire transitions in C_1

Lambert's Marked Graph Transition Sequences [Lam92]

Marked graph transition sequence $G = C_1.t_1.C_2.t_2.C_3$



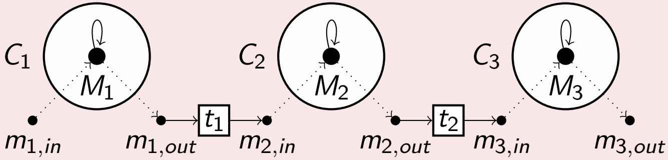
Solutions $\mathcal{L}(G)$

Transition sequence σ through $G = C_1.t_1.C_2.t_2.C_3$

- Start in $m_{1,in}$, fire transitions in C_1, reach $m_{1,out}$

Lambert's Marked Graph Transition Sequences [Lam92]

Marked graph transition sequence $G = C_1.t_1.C_2.t_2.C_3$



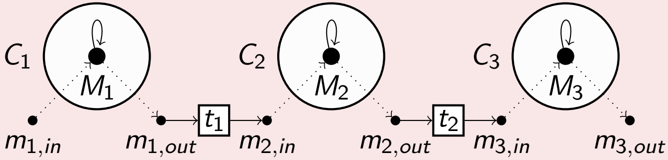
Solutions $\mathcal{L}(G)$

Transition sequence σ through $G = C_1.t_1.C_2.t_2.C_3$

- Start in $m_{1,in}$, fire transitions in C_1, reach $m_{1,out}$
- Fire t_1 to reach $m_{2,in}$, etc.

Lambert's Marked Graph Transition Sequences [Lam92]

Marked graph transition sequence $G = C_1.t_1.C_2.t_2.C_3$



Solutions $\mathcal{L}(G)$

Transition sequence σ through $G = C_1.t_1.C_2.t_2.C_3$

- Start in $m_{1,in}$, fire transitions in C_1, reach $m_{1,out}$
- Fire t_1 to reach $m_{2,in}$, etc.

Concrete values must be reached exactly!

MGTS and Reachability

Reachability problem $RP = (N, M_0, M_f)$

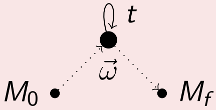
Is M_f reachable from M_0 in N?

MGTS and Reachability

Reachability problem $RP = (N, M_0, M_f)$

Is M_f reachable from M_0 in N?

Corresponding mgts G_{RP}



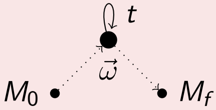
f.a. $t \in T$

MGTS and Reachability

Reachability problem $RP = (N, M_0, M_f)$

Is M_f reachable from M_0 in N?

Corresponding mgts G_{RP}



f.a. $t \in T$

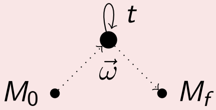
RP holds iff $\mathcal{L}(G_{RP}) \neq \emptyset$

MGTS and Reachability

Reachability problem $RP = (N, M_0, M_f)$

Is M_f reachable from M_0 in N?

Corresponding mgts G_{RP}



f.a. $t \in T$

RP holds iff $\mathcal{L}(G_{RP}) \neq \emptyset$

Even more

$$\mathcal{L}(N, M_0, M_f) = \{\sigma \mid M_0[\sigma\rangle M_f\} = \mathcal{L}(G_{RP})$$

Characteristic Equation

Characteristic equation for mgts

Encode firing in $G = C_1.t_1.C_2.t_2.C_3$ into $Ax = b$

Characteristic Equation

Characteristic equation for mgts

Encode firing in $G = C_1.t_1.C_2.t_2.C_3$ into $Ax = b$

- Variables $x(m_{in}(p))$

Characteristic Equation

Characteristic equation for mgts

Encode firing in $G = C_1.t_1.C_2.t_2.C_3$ into $Ax = b$

- Variables $x(m_{in}(p))$, $x(m_{out}(p))$

Characteristic Equation

Characteristic equation for mgts

Encode firing in $G = C_1.t_1.C_2.t_2.C_3$ into $Ax = b$

- Variables $x(m_{in}(p))$, $x(m_{out}(p))$, $x(t)$

Characteristic Equation

Characteristic equation for mgts

Encode firing in $G = C_1.t_1.C_2.t_2.C_3$ into $Ax = b$

- Variables $x(m_{in}(p))$, $x(m_{out}(p))$, $x(t)$
- Find non-negative solutions

Characteristic Equation

Characteristic equation for mgts

Encode firing in $G = C_1.t_1.C_2.t_2.C_3$ into $Ax = b$

- Variables $x(m_{in}(p))$, $x(m_{out}(p))$, $x(t)$
- Find non-negative solutions

They are computable! [Lam88]

Characteristic Equation

Characteristic equation for mgts

Encode firing in $G = C_1.t_1.C_2.t_2.C_3$ into $Ax = b$

- Variables $x(m_{in}(p))$, $x(m_{out}(p))$, $x(t)$
- Find non-negative solutions

They are computable! [Lam88]

Relationship

$$\mathcal{L}(G) \neq \emptyset \quad \Rightarrow \quad Ax = b \text{ solvable}$$

Characteristic Equation

Characteristic equation for mgts

Encode firing in $G = C_1.t_1.C_2.t_2.C_3$ into $Ax = b$

- Variables $x(m_{in}(p))$, $x(m_{out}(p))$, $x(t)$
- Find non-negative solutions

They are computable! [Lam88]

Relationship

$$\mathcal{L}(G) \neq \emptyset \quad \Rightarrow \quad Ax = b \text{ solvable}$$
$$\Leftarrow \quad \text{does not hold}$$

Characteristic Equation

Characteristic equation for mgts

Encode firing in $G = C_1.t_1.C_2.t_2.C_3$ into $Ax = b$

- Variables $x(m_{in}(p))$, $x(m_{out}(p))$, $x(t)$
- Find non-negative solutions

They are computable! [Lam88]

Relationship

$$\mathcal{L}(G) \neq \emptyset \quad \Rightarrow \quad Ax = b \text{ solvable}$$
$$\Leftarrow \quad \text{does not hold \ldots in general}$$

Perfect MGTS

Perfect mgts $\mathbb{G} = C_1.t_1.C_2.t_2.C_3$

- Edge variables $x(t)$ unbounded in solution space of $Ax = b$

Perfect MGTS

Perfect mgts $\mathbb{G} = C_1.t_1.C_2.t_2.C_3$

- Edge variables $x(t)$ unbounded in solution space of $Ax = b$
- Marking variables $x(m_{in/out}(p))$ unbounded for ω-entries

Perfect MGTS

Perfect mgts $\mathbb{G} = C_1.t_1.C_2.t_2.C_3$

- Edge variables $x(t)$ unbounded in solution space of $Ax = b$
- Marking variables $x(m_{in/out}(p))$ unbounded for ω-entries

Property

$$\mathcal{L}(\mathbb{G}) \neq \emptyset \qquad \Leftrightarrow \qquad Ax = b \text{ solvable}$$

Perfect MGTS

Perfect mgts $\mathbb{G} = C_1.t_1.C_2.t_2.C_3$

- Edge variables $x(t)$ unbounded in solution space of $Ax = b$
- Marking variables $x(m_{in/out}(p))$ unbounded for ω-entries

Property

$\mathcal{L}(\mathbb{G}) \neq \emptyset \quad \Leftrightarrow \quad Ax = b$ solvable (later)

Perfect MGTS

Perfect mgts $\mathbb{G} = C_1.t_1.C_2.t_2.C_3$

- Edge variables $x(t)$ unbounded in solution space of $Ax = b$
- Marking variables $x(m_{in/out}(p))$ unbounded for ω-entries

Property

$\mathcal{L}(\mathbb{G}) \neq \emptyset \quad \Leftrightarrow \quad Ax = b$ solvable (later)

But not every mgts is perfect!

Perfect MGTS

Perfect mgts $\mathbb{G} = C_1.t_1.C_2.t_2.C_3$

- Edge variables $x(t)$ unbounded in solution space of $Ax = b$
- Marking variables $x(m_{in/out}(p))$ unbounded for ω-entries

Property

$\mathcal{L}(\mathbb{G}) \neq \emptyset \quad \Leftrightarrow \quad Ax = b$ solvable (later)

Idea

Improve perfectness by unrolling

Perfect MGTS

Perfect mgts $\mathbb{G} = C_1.t_1.C_2.t_2.C_3$

- Edge variables $x(t)$ unbounded in solution space of $Ax = b$
- Marking variables $x(m_{in/out}(p))$ unbounded for ω-entries

Property

$\mathcal{L}(\mathbb{G}) \neq \emptyset \quad \Leftrightarrow \quad Ax = b$ solvable (later)

Theorem (Lambert's decomposition theorem [Lam92])

G can be effectively decomposed into a finite set Γ_G of perfect mgts with

$$\mathcal{L}(G) = \bigcup_{\mathbb{G} \in \Gamma_G} \mathcal{L}(\mathbb{G})$$

Solving Reachability

Where are we?

$RP = (N, M_0, M_f)$ holds

Solving Reachability

Where are we?

$$RP = (N, M_0, M_f) \text{ holds} \quad \Leftrightarrow \quad \mathcal{L}(G_{RP}) \neq \emptyset$$

Solving Reachability

Where are we?

$$RP = (N, M_0, M_f) \text{ holds} \quad \Leftrightarrow \quad \bigcup_{\mathbb{G} \in \Gamma_{G_{RP}}} \mathcal{L}(\mathbb{G}) \neq \emptyset$$

Solving Reachability

Where are we?

$$RP = (N, M_0, M_f) \text{ holds} \quad \Leftrightarrow \quad \bigcup_{\mathbb{G} \in \Gamma_{G_{RP}}} \mathcal{L}(\mathbb{G}) \neq \emptyset$$

$$\mathcal{L}(\mathbb{G}) \neq \emptyset \quad \Leftrightarrow \quad Ax = b \text{ solvable}$$

Solving Reachability

Where are we?

$$RP = (N, M_0, M_f) \text{ holds} \quad \Leftrightarrow \quad \bigcup_{\mathbb{G} \in \Gamma_{G_{RP}}} \mathcal{L}(\mathbb{G}) \neq \emptyset$$

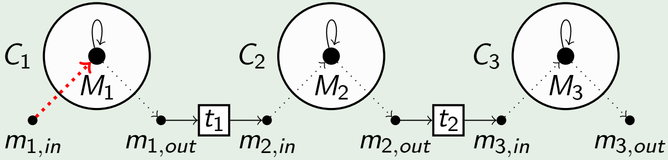
$$\mathcal{L}(\mathbb{G}) \neq \emptyset \quad \Leftrightarrow \quad Ax = b \text{ solvable}$$

Goal

Compute $\sigma \in \mathcal{L}(\mathbb{G})$ from solution to $Ax = b$

Solving Reachability

Perfect mgts

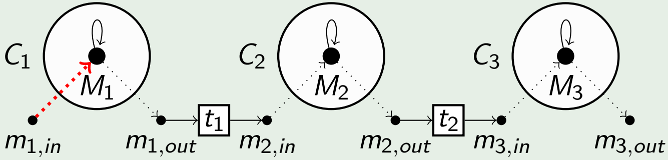


Pumping tokens

Sequence u leads from m_{in} to M

Solving Reachability

Perfect mgts



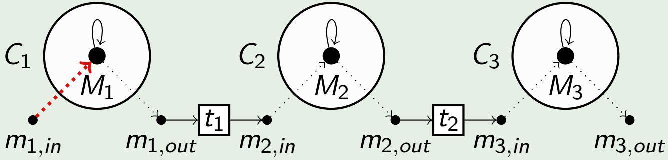
Pumping tokens

Sequence u leads from m_{in} to M

- u adds tokens to p with $M(p) = \omega > m_{in}(p)$

Solving Reachability

Perfect mgts



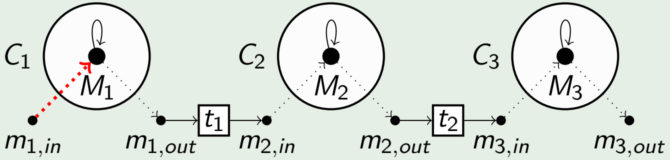
Pumping tokens

Sequence u leads from m_{in} to M

- u adds tokens to p with $M(p) = \omega > m_{in}(p)$
- u does not change tokens on p with $M(p) = m_{in}(p) = k \in \mathbb{N}$

Solving Reachability

Perfect mgts



Pumping tokens

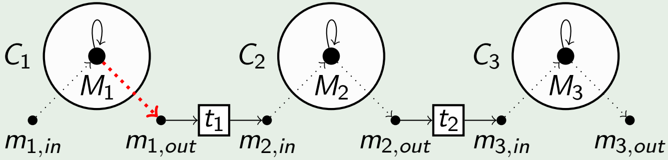
Sequence u leads from m_{in} to M

- u adds tokens to p with $M(p) = \omega > m_{in}(p)$
- u does not change tokens on p with $M(p) = m_{in}(p) = k \in \mathbb{N}$

u is computable! [Lam92]

Solving Reachability

Perfect mgts



Pumping tokens

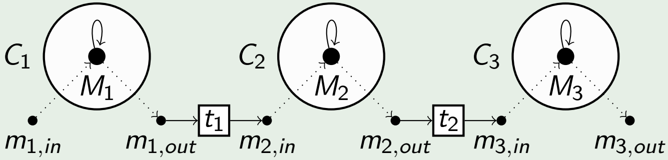
Sequence u leads from m_{in} to M

- u adds tokens to p with $M(p) = \omega > m_{in}(p)$
- u does not change tokens on p with $M(p) = m_{in}(p) = k \in \mathbb{N}$

Sequence v leads from M to m_{out}

Solving Reachability

Perfect mgts



Theorem (Lambert's pumping lemma)

Let $Ax = b$ have solution. Then $\sigma_k \in \mathcal{L}(\mathbb{G})$ for every $k > k_0$ with

$$\sigma_k = (u_1^k.\beta_1.\alpha_1^k.v_1^k).t_1.(u_2^k.\beta_2.\alpha_2^k.v_2^k)\ldots t_{n-1}.(u_n^k.\beta_n.\alpha_n^k.v_n^k)$$

Computing the Downward-Closure

Computing the downward-closure

$\mathcal{L}(N, M_0, M_f)$

Computing the Downward-Closure

Computing the downward-closure

$$\mathcal{L}(N, M_0, M_f) = \mathcal{L}(G_{RP})$$

Computing the Downward-Closure

Computing the downward-closure

$$\mathcal{L}(N, M_0, M_f) = \mathcal{L}(G_{RP}) = \bigcup_{\mathbb{G} \in \Gamma_{G_{RP}}} \mathcal{L}(\mathbb{G})$$

Computing the Downward-Closure

Computing the downward-closure

$$\mathcal{L}(N, M_0, M_f)\downarrow = \mathcal{L}(G_{RP})\downarrow = \bigcup_{\mathbb{G}\in\Gamma_{G_{RP}}} \mathcal{L}(\mathbb{G})\downarrow$$

Computing the Downward-Closure

Computing the downward-closure

$$\mathcal{L}(N, M_0, M_f)\downarrow = \mathcal{L}(G_{RP})\downarrow = \bigcup_{\mathbb{G}\in\Gamma_{G_{RP}}} \mathcal{L}(\mathbb{G})\downarrow$$

Theorem (Downward-closure of mgts)

Let $\mathbb{G} = C_1.t_1.C_2 \ldots t_{n-1}.C_n$ *have solution. Then*

$$\mathcal{L}(\mathbb{G})\downarrow = T_1^*.(t_1+\epsilon).T_2^* \ldots (t_{n-1}+\epsilon).T_n^*$$

Computing the Downward-Closure

Computing the downward-closure

$$\mathcal{L}(N, M_0, M_f)\downarrow = \mathcal{L}(G_{RP})\downarrow = \bigcup_{\mathbb{G}\in\Gamma_{G_{RP}}} \mathcal{L}(\mathbb{G})\downarrow$$

Theorem (Downward-closure of mgts)

Let $\mathbb{G} = C_1.t_1.C_2 \dots t_{n-1}.C_n$ *have solution. Then*

$$\mathcal{L}(\mathbb{G})\downarrow = T_1^*.(t_1+\epsilon).T_2^* \dots (t_{n-1}+\epsilon).T_n^*$$

$\supseteq$: Choose u with all transitions T in C

Computing the Downward-Closure

Computing the downward-closure

$$\mathcal{L}(N, M_0, M_f)\downarrow = \mathcal{L}(G_{RP})\downarrow = \bigcup_{\mathbb{G}\in\Gamma_{G_{RP}}} \mathcal{L}(\mathbb{G})\downarrow$$

Theorem (Downward-closure of mgts)

Let $\mathbb{G} = C_1.t_1.C_2 \ldots t_{n-1}.C_n$ have solution. Then

$$\mathcal{L}(\mathbb{G})\downarrow = T_1^*.(t_1 + \epsilon).T_2^* \ldots (t_{n-1} + \epsilon).T_n^*$$

$\supseteq$: Choose u with all transitions T in C

Apply Lambert's pumping lemma

Computing the Downward-Closure

Computing the downward-closure

$$\mathcal{L}(N, M_0, M_f)\downarrow = \mathcal{L}(G_{RP})\downarrow = \bigcup_{\mathbb{G}\in\Gamma_{G_{RP}}} \mathcal{L}(\mathbb{G})\downarrow$$

Theorem (Downward-closure of mgts)

Let $\mathbb{G} = C_1.t_1.C_2 \dots t_{n-1}.C_n$ *have solution. Then*

$$\mathcal{L}(\mathbb{G})\downarrow = T_1^*.(t_1+\epsilon).T_2^* \dots (t_{n-1}+\epsilon).T_n^*$$

$\supseteq$: Choose u with all transitions T in C

Apply Lambert's pumping lemma

$$T^* \subseteq \bigcup_{k\geq k_0} u^k\downarrow$$

Computing the Downward-Closure

Computing the downward-closure

$$\mathcal{L}(N, M_0, M_f)\downarrow = \mathcal{L}(G_{RP})\downarrow = \bigcup_{\mathbb{G}\in\Gamma_{G_{RP}}} \mathcal{L}(\mathbb{G})\downarrow$$

Theorem (Downward-closure of mgts)

Let $\mathbb{G} = C_1.t_1.C_2 \ldots t_{n-1}.C_n$ *have solution. Then*

$$\mathcal{L}(\mathbb{G})\downarrow = T_1^*.(t_1+\epsilon).T_2^* \ldots (t_{n-1}+\epsilon).T_n^*$$

$\supseteq$: Choose u with all transitions T in C

Apply Lambert's pumping lemma

$$\begin{aligned} & T_1^*.(t_1+\epsilon).T_2^* \ldots (t_{n-1}+\epsilon).T_n^* \\ \subseteq\ & \bigcup_{k\geq k_0} (u_1^k.\beta_1.\alpha_1^k.v_1^k).t_1.(u_2^k.\beta_2.\alpha_2^k.v_2^k) \ldots t_{n-1}.(u_n^k.\beta_n.\alpha_n^k.v_n^k)\downarrow \end{aligned}$$

Computing the Downward-Closure

Computing the downward-closure

$$\mathcal{L}(N, M_0, M_f)\downarrow = \mathcal{L}(G_{RP})\downarrow = \bigcup_{\mathbb{G}\in\Gamma_{G_{RP}}} \mathcal{L}(\mathbb{G})\downarrow$$

Theorem (Downward-closure of mgts)

Let $\mathbb{G} = C_1.t_1.C_2 \dots t_{n-1}.C_n$ *have solution. Then*

$$\mathcal{L}(\mathbb{G})\downarrow = T_1^*.(t_1+\epsilon).T_2^* \dots (t_{n-1}+\epsilon).T_n^*$$

Construction of u

Computing the Downward-Closure

Computing the downward-closure

$$\mathcal{L}(N, M_0, M_f)\downarrow = \mathcal{L}(G_{RP})\downarrow = \bigcup_{\mathbb{G}\in\Gamma_{G_{RP}}} \mathcal{L}(\mathbb{G})\downarrow$$

Theorem (Downward-closure of mgts)

Let $\mathbb{G} = C_1.t_1.C_2 \dots t_{n-1}.C_n$ *have solution. Then*

$$\mathcal{L}(\mathbb{G})\downarrow = T_1^*.(t_1+\epsilon).T_2^* \dots (t_{n-1}+\epsilon).T_n^*$$

Construction of u

- Take u_a by Lambert

Computing the Downward-Closure

Computing the downward-closure

$$\mathcal{L}(N, M_0, M_f)\downarrow = \mathcal{L}(G_{RP})\downarrow = \bigcup_{\mathbb{G}\in\Gamma_{G_{RP}}} \mathcal{L}(\mathbb{G})\downarrow$$

Theorem (Downward-closure of mgts)

Let $\mathbb{G} = C_1.t_1.C_2 \ldots t_{n-1}.C_n$ *have solution. Then*

$$\mathcal{L}(\mathbb{G})\downarrow = T_1^*.(t_1+\epsilon).T_2^* \ldots (t_{n-1}+\epsilon).T_n^*$$

Construction of u

- Take u_a by Lambert
- Take u_b with all transitions T in C (strongly connected)

Computing the Downward-Closure

Computing the downward-closure

$$\mathcal{L}(N, M_0, M_f)\downarrow = \mathcal{L}(G_{RP})\downarrow = \bigcup_{\mathbb{G}\in\Gamma_{G_{RP}}} \mathcal{L}(\mathbb{G})\downarrow$$

Theorem (Downward-closure of mgts)

Let $\mathbb{G} = C_1.t_1.C_2 \ldots t_{n-1}.C_n$ *have solution. Then*

$$\mathcal{L}(\mathbb{G})\downarrow = T_1^*.(t_1+\epsilon).T_2^* \ldots (t_{n-1}+\epsilon).T_n^*$$

Construction of u

- Take u_a by Lambert
- Take u_b with all transitions T in C (strongly connected)

$$u := u_a^{m+1}.u_b$$

Computing the Downward-Closure

Computing the downward-closure

$$\mathcal{L}(N, M_0, M_f)\downarrow = \mathcal{L}(G_{RP})\downarrow = \bigcup_{\mathbb{G}\in\Gamma_{G_{RP}}} \mathcal{L}(\mathbb{G})\downarrow$$

Theorem (Downward-closure of mgts)

Let $\mathbb{G} = C_1.t_1.C_2 \ldots t_{n-1}.C_n$ *have solution. Then*

$$\mathcal{L}(\mathbb{G})\downarrow = T_1^*.(t_1 + \epsilon).T_2^* \ldots (t_{n-1} + \epsilon).T_n^*$$

Construction of u

- Take u_a by Lambert
- Take u_b with all transitions T in C (strongly connected)

$$u := u_a^{m+1}.u_b \qquad m = \text{maximal negative effect of } u_b$$

Terminal Languages

Definition (Terminal language in *TPN*)

$$\mathcal{T}_h(N, M_0) = \{h(\sigma) \mid M_0[\sigma\rangle M \text{ and } \neg M[t\rangle \text{ f.a. } t \in T\}$$

Terminal Languages

Idea

Deadlocks given by finite set $\mathcal{P}$ of partial markings M_P, $P \subseteq P'$

$$\mathcal{T}_h(N, M_0) = \bigcup_{M_P \in \mathcal{P}} \mathcal{L}_h(N, M_0, M_P)$$

Terminal Languages

Idea

Deadlocks given by finite set $\mathcal{P}$ of partial markings M_P, $P \subseteq P'$

$$\mathcal{T}_h(N, M_0) = \bigcup_{M_P \in \mathcal{P}} \mathcal{L}_h(N, M_0, M_P)$$

Compute downward-closure of partial languages

$$\mathcal{L}_h(N, M_0, M_P)\downarrow = \phi_{M_P}$$

Terminal Languages

Idea

Deadlocks given by finite set $\mathcal{P}$ of partial markings M_P, $P \subseteq P'$

$$\mathcal{T}_h(N, M_0) = \bigcup_{M_P \in \mathcal{P}} \mathcal{L}_h(N, M_0, M_P)$$

Compute downward-closure of partial languages

$$\mathcal{L}_h(N, M_0, M_P)\downarrow = \phi_{M_P}$$

Theorem (Representation)

$$\mathcal{T}_h(N, M_0)\downarrow = \Sigma_{M_P \in \mathcal{P}} \quad \phi_{M_P}$$

Partial Languages and their Representation

Partial languages: acceptance by corresponding markings

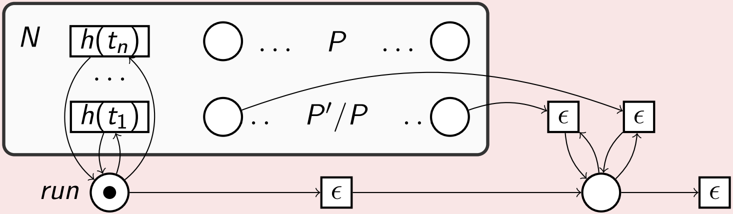
$$\mathcal{L}_h(N, M_0, M_P) = \bigcup_{M|_P = M_P} \mathcal{L}_h(N, M_0, M)$$

Partial Languages and their Representation

Partial languages: acceptance by corresponding markings

$$\mathcal{L}_h(N, M_0, M_P) = \bigcup_{M|_P = M_P} \mathcal{L}_h(N, M_0, M)$$

From infinite union to ordinary language [Hac76]

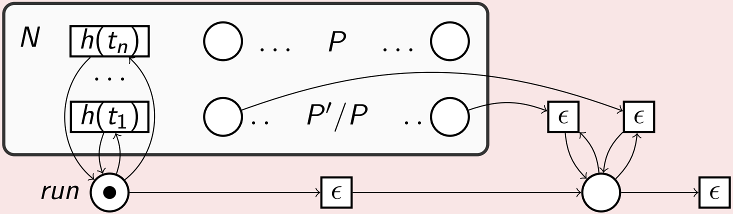


Partial Languages and their Representation

Partial languages: acceptance by corresponding markings

$$\mathcal{L}_h(N, M_0, M_P) = \bigcup_{M|_P = M_P} \mathcal{L}_h(N, M_0, M)$$

From infinite union to ordinary language [Hac76]



Lemma

$$\mathcal{L}_h(N, M_0, M_P) = \mathcal{L}_{h \cup h_g}(N, M_0^{run}, M_P^{empty})$$

Covering Languages

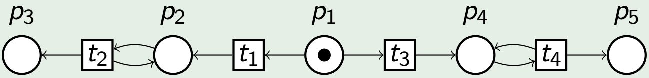
Definition (Covering language in *CPN*)

$$\mathcal{C}_h(N, M_0, M_f) = \{h(\sigma) \mid M_0[\sigma\rangle M \text{ and } M \geq M_f\}$$

Covering Languages

Turn coverability tree into finite automaton

Example



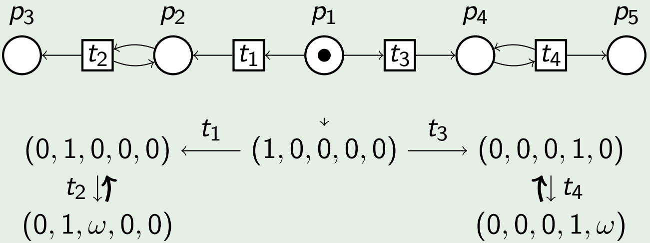
$$\begin{array}{ccccc} (0,1,0,0,0) & \xleftarrow{t_1} & (1,0,0,0,0) & \xrightarrow{t_3} & (0,0,0,1,0) \\ t_2 \downarrow & & & & \downarrow t_4 \\ (0,1,\omega,0,0) & & & & (0,0,0,1,\omega) \end{array}$$

Covering Languages

Turn coverability tree into finite automaton

- Silent transitions to smaller nodes

Example

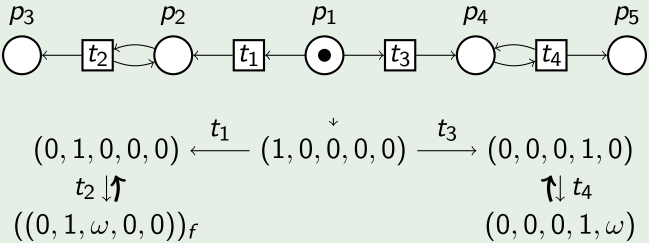


Covering Languages

Turn coverability tree into finite automaton

- Silent transitions to smaller nodes
- Accept when final marking dominated

Example



Covering Languages

Turn coverability tree into finite automaton

- Silent transitions to smaller nodes
- Accept when final marking dominated

Lemma

$$\mathcal{C}_h(N, M_0, M_f)\downarrow = \mathcal{L}(FA)\downarrow$$

Downward-closure of the Automaton

Tree of strongly connected components

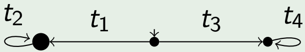
Example

$$
\begin{array}{ccccc}
(0,1,0,0,0) & \xleftarrow{t_1} & (1,0,\overset{\downarrow}{0},0,0) & \xrightarrow{t_3} & (0,0,0,1,0) \\
t_2 \downarrow\!\uparrow & & & & \uparrow\!\downarrow t_4 \\
((0,1,\omega,0,0))_f & & & & (0,0,0,1,\omega)
\end{array}
$$

Downward-closure of the Automaton

Tree of strongly connected components

Example



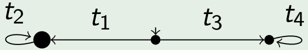
Downward-closure of the Automaton

Tree of strongly connected components

Compute expression recursively

$$\phi_C = T_C^*.(\qquad\qquad\qquad\qquad)$$

Example



Downward-closure of the Automaton

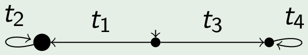
Tree of strongly connected components

Compute expression recursively

$$\phi_C = T_C^*.(\tau_C \qquad\qquad\qquad\qquad)$$

- $\tau_C = \epsilon$ (final) or $\tau_C = \emptyset$ (not final)

Example



Downward-closure of the Automaton

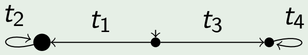
Tree of strongly connected components

Compute expression recursively

$$\phi_C = T_C^*.(\tau_C \quad + \quad \Sigma_{C'} \quad \gamma_{C,C'}.\phi_{C'})$$

- $\tau_C = \epsilon$ (final) or $\tau_C = \emptyset$ (not final)
- $\gamma_{C,C'} = t + \epsilon$ (edge) or $\gamma_{C,C'} = \emptyset$ (no edge)

Example



Downward-closure of the Automaton

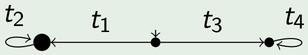
Tree of strongly connected components

Compute expression recursively

$$\phi_C = T_C^*.(\tau_C \quad + \quad \Sigma_{C'} \quad \gamma_{C,C'}.\phi_{C'})$$

- $\tau_C = \epsilon$ (final) or $\tau_C = \emptyset$ (not final)
- $\gamma_{C,C'} = t + \epsilon$ (edge) or $\gamma_{C,C'} = \emptyset$ (no edge)

Example



$$\phi_{C_0} = (t_1 + \epsilon).t_2^* \quad + \quad (t_3 + \epsilon).t_4^*.\emptyset$$

Downward-closure of the Automaton

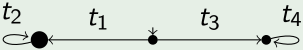
Tree of strongly connected components

Compute expression recursively

$$\phi_C = T_C^*.(\tau_C \quad + \quad \Sigma_{C'} \quad \gamma_{C,C'}.\phi_{C'})$$

- $\tau_C = \epsilon$ (final) or $\tau_C = \emptyset$ (not final)
- $\gamma_{C,C'} = t + \epsilon$ (edge) or $\gamma_{C,C'} = \emptyset$ (no edge)

Example



$$\phi_{C_0} = (t_1 + \epsilon).t_2^*$$

Van Leeuwen's Algebraic Languages

Definition (Algebraic languages over K)

- *K-grammar* is (V, Σ, P, S) with productions P of form

$$A \rightarrow \mathcal{L} \in K \qquad \mathcal{L} \subseteq V^*$$

Van Leeuwen's Algebraic Languages

Definition (Algebraic languages over K)

- *K-grammar* is (V, Σ, P, S) with productions P of form

$$A \rightarrow \mathcal{L} \in K \qquad \mathcal{L} \subseteq V^*$$

- *$\mathcal{L}$ algebraic over K* if generated by a K-grammar

Van Leeuwen's Algebraic Languages

Definition (Algebraic languages over K)

- *K-grammar* is (V, Σ, P, S) with productions P of form

$$A \to \mathcal{L} \in K \qquad \mathcal{L} \subseteq V^*$$

- *$\mathcal{L}$ algebraic over K* if generated by a K-grammar
- *K^{∇} class of all algebraic languages over K*

Van Leeuwen's Algebraic Languages

Definition (Algebraic languages over K)

- *K-grammar* is (V, Σ, P, S) with productions P of form

$$A \to \mathcal{L} \in K \qquad \mathcal{L} \subseteq V^*$$

- $\mathcal{L}$ *algebraic over K* if generated by a K-grammar
- K^∇ *class of all algebraic languages over K*

Example (Context-free languages)

$REG^\nabla = CF$

Van Leeuwen's Algebraic Languages

Definition (Algebraic languages over K)

- *K-grammar* is (V, Σ, P, S) with productions P of form

$$A \to \mathcal{L} \in K \qquad \mathcal{L} \subseteq V^*$$

- $\mathcal{L}$ *algebraic over* K if generated by a K-grammar
- K^∇ *class of all algebraic languages over* K

Example (Context-free languages)

$REG^\nabla = CF$

Theorem (Van Leeuwen's Theorem 4.5 [vL78])

$\mathcal{L}\downarrow$ effectively computable for all $\mathcal{L} \in K^\nabla$ iff so for all $\mathcal{L} \in K$

Consequences

Consequence of van Leeuwen's work and our results

$\mathcal{L}\downarrow$ effectively computable for $\mathcal{L}$ in PN^{∇}, TPN^{∇}, and CPN^{∇}

Consequences

Consequence of van Leeuwen's work and our results

$\mathcal{L}\downarrow$ effectively computable for $\mathcal{L}$ in PN^∇, TPN^∇, and CPN^∇

Corollary

Emptiness decidable for languages in PN^∇, TPN^∇, and CPN^∇

Consequences

Consequence of van Leeuwen's work and our results

$\mathcal{L}\downarrow$ effectively computable for $\mathcal{L}$ in PN^∇, TPN^∇, and CPN^∇

Corollary

Emptiness decidable for languages in PN^∇, TPN^∇, and CPN^∇

Example (Language in PN^∇)

$$\underbrace{u_1.u_1^R \ldots u_n.u_n^R}_{\sim A^n} . \quad \underbrace{v_1.v_1^R \ldots v_n.v_n^R}_{\sim B^n} . \quad \underbrace{w_1.w_1^R \ldots w_n.w_n^R}_{\sim C^n}$$

Conclusion

Downward-closure of all Petri net languages computable

Conclusion

Downward-closure of all Petri net languages computable

- Ordinary languages

Conclusion

Downward-closure of all Petri net languages computable

- Ordinary languages via mgts and Lambert's pumping lemma for reachability

Conclusion

Downward-closure of all Petri net languages computable

- Ordinary languages via mgts and Lambert's pumping lemma for reachability
- Terminal languages

Conclusion

Downward-closure of all Petri net languages computable

- Ordinary languages via mgts and Lambert's pumping lemma for reachability
- Terminal languages by reduction from partial languages to ordinary languages

Conclusion

Downward-closure of all Petri net languages computable

- Ordinary languages via mgts and Lambert's pumping lemma for reachability
- Terminal languages by reduction from partial languages to ordinary languages and previous result

Conclusion

Downward-closure of all Petri net languages computable

- Ordinary languages via mgts and Lambert's pumping lemma for reachability
- Terminal languages by reduction from partial languages to ordinary languages and previous result
- Covering languages

Conclusion

Downward-closure of all Petri net languages computable

- Ordinary languages via mgts and Lambert's pumping lemma for reachability
- Terminal languages by reduction from partial languages to ordinary languages and previous result
- Covering languages from coverability tree

Conclusion

Downward-closure of all Petri net languages computable

- Ordinary languages via mgts and Lambert's pumping lemma for reachability
- Terminal languages by reduction from partial languages to ordinary languages and previous result
- Covering languages from coverability tree
- Algebraic languages over Petri nets

Conclusion

Downward-closure of all Petri net languages computable

- Ordinary languages via mgts and Lambert's pumping lemma for reachability
- Terminal languages by reduction from partial languages to ordinary languages and previous result
- Covering languages from coverability tree
- Algebraic languages over Petri nets

Applications

- Stability analysis computes tolerable intrusions

Conclusion

Downward-closure of all Petri net languages computable

- Ordinary languages via mgts and Lambert's pumping lemma for reachability
- Terminal languages by reduction from partial languages to ordinary languages and previous result
- Covering languages from coverability tree
- Algebraic languages over Petri nets

Applications

- Stability analysis computes tolerable intrusions
- Compositional verification

Related Work

- Upward/Downward-closure of context-free languages [vL78]

Related Work

- Upward/Downward-closure of context-free languages [vL78]

  $\mathcal{L}\uparrow$ effectively computable for $\mathcal{L}$ in PN^∇, TPN^∇, CPN^∇

Related Work

- Upward/Downward-closure of context-free languages [vL78]

  $\mathcal{L}\uparrow$ effectively computable for $\mathcal{L}$ in PN^{∇}, TPN^{∇}, CPN^{∇}

- Size of automata representation [GHK07, GHK09]

References I

H. Gruber, M. Holzer, and M. Kutrib.
The size of higman-haines sets.
Theor. Comp. Sci., 387(2):167–176, 2007.

H. Gruber, M. Holzer, and M. Kutrib.
More on the size of higman-haines sets: Effective constructions.
Fundam. Inf., 91(1):105–121, 2009.

M. Hack.
Decidability questions for Petri nets.
Technical report, Cambridge, MA, USA, 1976.

G. Higman.
Ordering by divisibility in abstract algebras.
Proc. London Math. Soc. (3), 2(7):326–336, 1952.

R. M. Karp and R. E. Miller.
Parallel program schemata.
J. Comput. Syst. Sci., 3(2):147–195, 1969.

J. L. Lambert.
Finding a partial solution to a linear system of equations in positive integers.
Comput. Math. Applic., 15(3):209–212, 1988.

References II

J. L. Lambert.
A structure to decide reachability in Petri nets.
Theor. Comp. Sci., 99(1):79–104, 1992.

J. van Leeuwen.
Effective constructions in well-partially-ordered free monoids.
Discrete Mathematics, 21(3):237–252, 1978.