On the Upward/Downward Closures of Petri Nets

Mohammed Faouzi Atig[1], Roland Meyer[2], **Sebastian Muskalla**[2] and Prakash Saivasan[2]

August 25, MFCS 2017, Aalborg

1 Uppsala University, Sweden
`mohamed_faouzi.atig@it.uu.se`

2 TU Braunschweig, Germany
`{roland.meyer, s.muskalla, p.saivasan}@tu-bs.de`

Goal

Goal

Study the upward and downward closures of Petri net coverability languages

Goal

Goal

Study the upward and downward closures of Petri net coverability languages

Why?

Petri nets are an important model for concurrent systems

Upward and downward closures are useful approximations for verification purposes

Goal

Goal

Study the upward and downward closures of Petri net coverability languages

Upward/Downward closures in general

Good:

Always regular

Goal

Goal

Study the upward and downward closures of Petri net coverability languages

Upward/Downward closures in general

Good:

Always simply regular

Goal

Goal

Study the upward and downward closures of Petri net coverability languages

Upward/Downward closures in general

Good:

Always simply regular

Bad:

Representations might be not be effectively computable or very large

Goal

Goal

Study the upward and downward closures of Petri net coverability languages

Here:

- Closures effectively regular
- Want to construct finite state automata (FSA) as representations

Goal

Goal

Study the upward and downward closures of Petri net coverability languages

Here:

Closures effectively regular

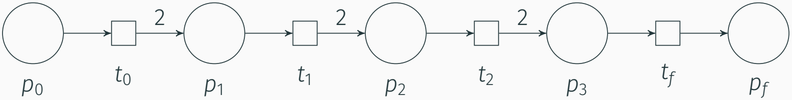
Want to construct finite state automata (FSA) as representations

↳ Time needed for the construction?

↳ Size of the minimal FSAs?

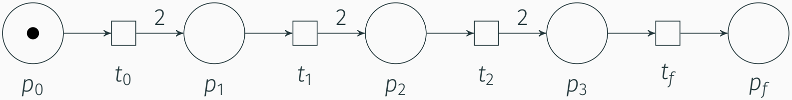
Petri Net Coverability Languages and their Closures

(Labeled) Petri Nets



$$N = \Big(\underbrace{\{p_0, p_1, p_2, p_3, p_f\}}_{\text{places } P}, \underbrace{\{t_0, t_1, t_2, t_f\}}_{\text{transitions } T}, F \Big)$$

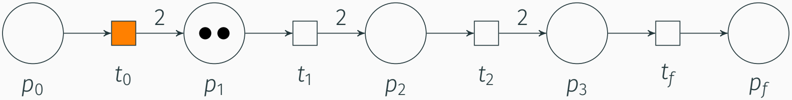
(Labeled) Petri Nets



$$N = \Big(\underbrace{\{p_0, p_1, p_2, p_3, p_f\}}_{\text{places } P}, \underbrace{\{t_0, t_1, t_2, t_f\}}_{\text{transitions } T}, F \Big)$$

$$\underset{M_0}{\underset{\parallel}{\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}}}$$

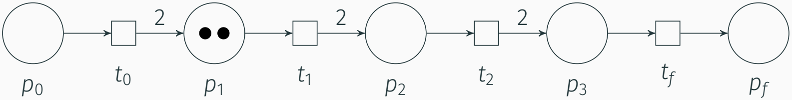
(Labeled) Petri Nets



$$N = \Big(\underbrace{\{p_0, p_1, p_2, p_3, p_f\}}_{\text{places } P}, \underbrace{\{t_0, t_1, t_2, t_f\}}_{\text{transitions } T}, F \Big)$$

$$\underset{\substack{\shortparallel \\ M_0}}{\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}} \quad [t_0\rangle$$

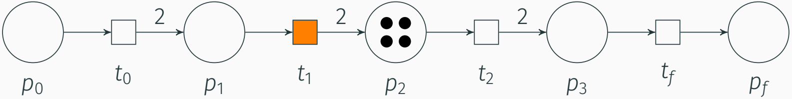
(Labeled) Petri Nets



$$N = \Big(\underbrace{\{p_0, p_1, p_2, p_3, p_f\}}_{\text{places } P}, \underbrace{\{t_0, t_1, t_2, t_f\}}_{\text{transitions } T}, F \Big)$$

$$\underset{M_0}{\overset{\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}}{\parallel}} \quad [t_0\rangle \qquad\qquad \underset{M_f}{\overset{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 2^3 \end{pmatrix}}{\parallel}}$$

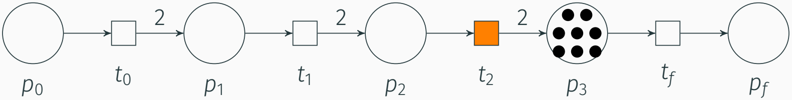
(Labeled) Petri Nets



$$N = \Big(\underbrace{\{p_0, p_1, p_2, p_3, p_f\}}_{\text{places } P}, \underbrace{\{t_0, t_1, t_2, t_f\}}_{\text{transitions } T}, F \Big)$$

$$\underset{M_0}{\underset{\parallel}{\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}}} \quad [t_0 t_1 t_1\rangle \qquad \underset{M_f}{\underset{\parallel}{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 2^3 \end{pmatrix}}}$$

(Labeled) Petri Nets

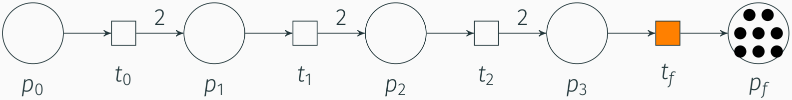


$$N = \left(\underbrace{\{p_0, p_1, p_2, p_3, p_f\}}_{\text{places } P}, \underbrace{\{t_0, t_1, t_2, t_f\}}_{\text{transitions } T}, F \right)$$

$$\underset{M_0}{\overset{}{\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}}} \quad [t_0 t_1 t_1 t_2 t_2 t_2 t_2 t_2\rangle \qquad \underset{M_f}{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 2^3 \end{pmatrix}}$$

$M_0 = (1, 0, 0, 0, 0)^T$, $M_f = (0, 0, 0, 0, 2^3)^T$

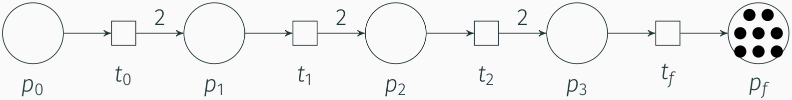
(Labeled) Petri Nets



$$N = \Big(\underbrace{\{p_0, p_1, p_2, p_3, p_f\}}_{\text{places } P}, \underbrace{\{t_0, t_1, t_2, t_f\}}_{\text{transitions } T}, F \Big)$$

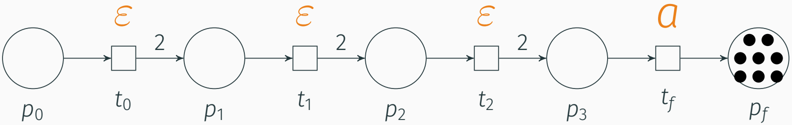
$$\underset{M_0}{\overset{\|}{\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}}} \quad [t_0 t_1 t_1 t_2 t_2 t_2 t_2 t_f t_f t_f t_f t_f t_f t_f t_f\rangle \quad \underset{M_f}{\overset{\|}{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 2^3 \end{pmatrix}}}$$

(Labeled) Petri Nets



$$N = \Big(\underbrace{\{p_0, p_1, p_2, p_3, p_f\}}_{\text{places } P}, \underbrace{\{t_0, t_1, t_2, t_f\}}_{\text{transitions } T}, F \Big)$$

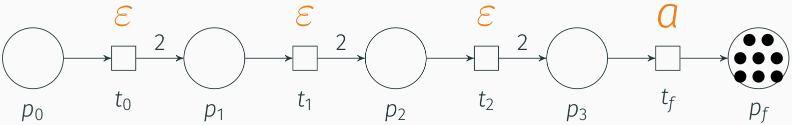
$$\underbrace{\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}}_{M_0} \quad [t_0 t_1 t_1 t_2 t_2 t_2 t_2 t_f t_f t_f t_f t_f t_f t_f t_f\rangle \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 8 \end{pmatrix} \geqslant \underbrace{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 2^3 \end{pmatrix}}_{M_f}$$



$$N = \left(\{a\}, \underbrace{\{p_0, p_1, p_2, p_3, p_f\}}_{\text{places } P}, \underbrace{\{t_0, t_1, t_2, t_f\}}_{\text{transitions } T}, F, \lambda\right)$$

$$\underbrace{\begin{pmatrix}1\\0\\0\\0\\0\end{pmatrix}}_{M_0} \quad [t_0 t_1 t_1 t_2 t_2 t_2 t_2 t_f t_f t_f t_f t_f t_f t_f t_f\rangle \begin{pmatrix}0\\0\\0\\0\\8\end{pmatrix} \geqslant \underbrace{\begin{pmatrix}0\\0\\0\\0\\2^3\end{pmatrix}}_{M_f}$$

(Labeled) Petri Nets



$$N = \Big(\{a\},\ \underbrace{\{p_0, p_1, p_2, p_3, p_f\}}_{\text{places } P},\ \underbrace{\{t_0, t_1, t_2, t_f\}}_{\text{transitions } T},\ F,\ \lambda\Big)$$

$$\underset{M_0}{\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}} \ \underset{\varepsilon\varepsilon\varepsilon\varepsilon\varepsilon\varepsilon\varepsilon aaaaaaaa = a^8}{\underset{\downarrow \lambda}{[t_0 t_1 t_1 t_2 t_2 t_2 t_2 t_f t_f t_f t_f t_f t_f t_f t_f\rangle}} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 8 \end{pmatrix} \geqslant \underset{M_f}{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 2^3 \end{pmatrix}}$$

Coverability Language

(N, M_0, M_f) Petri net with initial and final marking

Coverability language

$$\mathcal{L}\big(N, M_0, M_f\big) = \big\{\lambda(\sigma) \;\big|\; M_0[\sigma\rangle M, M \geqslant M_f\big\}$$

(N, M_0, M_f) Petri net with initial and final marking

Coverability language

$$\mathcal{L}(N, M_0, M_f) = \{\lambda(\sigma) \mid M_0[\sigma\rangle M, M \geqslant M_f\}$$

In the example: $\mathcal{L}(N, M_0, M_f) = \{a^8\}$

Upward and Downward Closure

Subword relation

$$v \preceq w \quad \text{iff} \quad v \text{ obtained from } w \text{ by deleting letters}$$
$$\quad\quad\quad \text{iff} \quad w \text{ obtained from } v \text{ by inserting letters}$$

Upward and Downward Closure

Subword relation

$$v \preceq w \quad \text{iff} \quad v \text{ obtained from } w \text{ by deleting letters}$$
$$\quad \text{iff} \quad w \text{ obtained from } v \text{ by inserting letters}$$

Upward closure

$$\mathcal{L}\!\uparrow = \{w \mid \exists v \in \mathcal{L} \colon v \preceq w\}$$

Upward and Downward Closure

Subword relation

$$v \preceq w \quad \text{iff} \quad v \text{ obtained from } w \text{ by deleting letters}$$
$$\quad \text{iff} \quad w \text{ obtained from } v \text{ by inserting letters}$$

Upward closure

$$\mathcal{L}\uparrow = \{w \mid \exists v \in \mathcal{L} \colon v \preceq w\}$$

Downward closure

$$\mathcal{L}\downarrow = \{w \mid \exists v \in \mathcal{L} \colon w \preceq v\}$$

Upward and Downward Closure

Subword relation

$$v \preceq w \quad \text{iff} \quad v \text{ obtained from } w \text{ by deleting letters}$$
$$\quad\quad\quad \text{iff} \quad w \text{ obtained from } v \text{ by inserting letters}$$

Upward closure

$$\mathcal{L}\uparrow = \{w \mid \exists v \in \mathcal{L} \colon v \preceq w\}$$

Downward closure

$$\mathcal{L}\downarrow = \{w \mid \exists v \in \mathcal{L} \colon w \preceq v\}$$

In the example: $\mathcal{L}(N, M_0, M_f)\uparrow = \{a^k \mid k \geqslant 8\}$

$$\mathcal{L}(N, M_0, M_f)\downarrow = \{a^k \mid k \leqslant 8\}$$

Computing the Upward Closure

Computing the Upward Closure

Computing the Upward Closure

Given: Petri net (N, M_0, M_f).

Compute: FSA A with $\mathcal{L}(A) = \mathcal{L}\big(N, M_0, M_f\big)\uparrow$.

Computing the Upward Closure

Given: Petri net (N, M_0, M_f).

Compute: FSA A with $\mathcal{L}(A) = \mathcal{L}\big(N, M_0, M_f\big)\uparrow$.

Theorem

Upper bound: *One can compute an FSA of doubly exponential size representing the upward closure in doubly exponential time.*

Lower bound: *This is optimal.*

Computing the Upward Closure - Upper Bound

Lemma (Upper Bound)

One can compute an FSA of doubly exponential size for the upward closure in doubly exponential time.

Computing the Upward Closure - Upper Bound

Lemma (Upper Bound)

One can compute an FSA of doubly exponential size for the upward closure in doubly exponential time.

Theorem (Rackoff 1978)

Petri net coverability can be solved using exponential space (and doubly exponential time).

Computing the Upward Closure - Upper Bound

Theorem (Rackoff 1978)

Petri net coverability can be solved using exponential space (and doubly exponential time).

Computing the Upward Closure - Upper Bound

Theorem (Rackoff 1978)

Petri net coverability can be solved using exponential space (and doubly exponential time).

Assume places are ordered, $P = [1..\ell]$.

Computing the Upward Closure - Upper Bound

Theorem (Rackoff 1978)

Petri net coverability can be solved using exponential space (and doubly exponential time).

Assume places are ordered, $P = [1..\ell]$.

Consider i-bounded computations: Allow negative values on the places $[i + 1..\ell]$

Computing the Upward Closure - Upper Bound

Theorem (Rackoff 1978)

Petri net coverability can be solved using exponential space (and doubly exponential time).

Assume places are ordered, $P = [1..\ell]$.

Consider i-bounded computations: Allow negative values on the places $[i+1..\ell]$

Define $f(i)$ upper bound on the length of an i-bounded, i-covering computation from an arbitrary initial marking

Computing the Upward Closure - Upper Bound

Theorem (Rackoff 1978)

Petri net coverability can be solved using exponential space (and doubly exponential time).

Assume places are ordered, $P = [1..\ell]$.

Consider i-bounded computations: Allow negative values on the places $[i+1..\ell]$

Define $f(i)$ upper bound on the length of an i-bounded, i-covering computation from an arbitrary initial marking

Prove $f(\ell) \leqslant 2^{2^{\mathcal{O}(n \cdot \log n)}}$

Computing the Upward Closure - Upper Bound

Theorem (Rackoff 1978)

Petri net coverability can be solved using exponential space (and doubly exponential time).

Assume places are ordered, $P = [1..\ell]$.

Consider i-bounded computations: Allow negative values on the places $[i+1..\ell]$

Define $f(i)$ upper bound on the length of an i-bounded, i-covering computation from an arbitrary initial marking

Prove $f(\ell) \leqslant 2^{2^{\mathcal{O}(n \cdot \log n)}}$

Show $f(i+1) \leqslant (2^n f(i))^{i+1} + f(i)$

Computing the Upward Closure - Upper Bound

$$f(i+1) \leqslant (2^n f(i))^{i+1} + f(i)$$

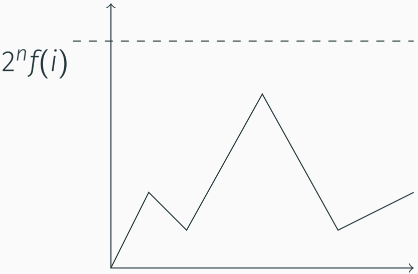
Take an arbitrary $(i+1)$-bounded, $(i+1)$-covering computation

Computing the Upward Closure - Upper Bound

$$f(i+1) \leqslant (2^n f(i))^{i+1} + f(i)$$

Take an arbitrary $(i+1)$-bounded, $(i+1)$-covering computation

1st case: Values on all places $[1..i+1]$ bounded by $2^n \cdot f(i)$:



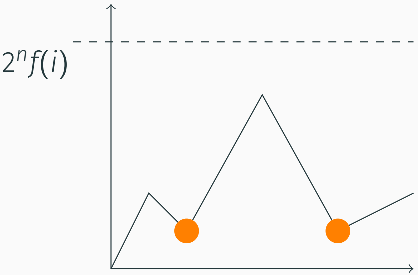
Computing the Upward Closure - Upper Bound

$$f(i+1) \leqslant (2^n f(i))^{i+1} + f(i)$$

Take an arbitrary $(i+1)$-bounded, $(i+1)$-covering computation

1st case: Values on all places $[1..i+1]$ bounded by $2^n \cdot f(i)$:

Identify repetitions



Computing the Upward Closure - Upper Bound

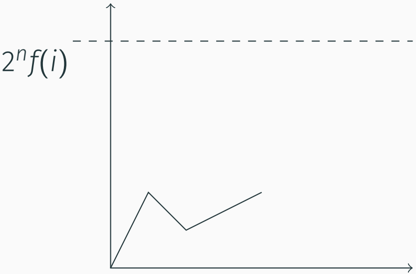
$$f(i+1) \leqslant (2^n f(i))^{i+1} + f(i)$$

Take an arbitrary $(i+1)$-bounded, $(i+1)$-covering computation

1st case: Values on all places $[1..i+1]$ bounded by $2^n \cdot f(i)$:

Identify repetitions

Delete loops



Computing the Upward Closure - Upper Bound

$$f(i+1) \leqslant (2^n f(i))^{i+1} + f(i)$$

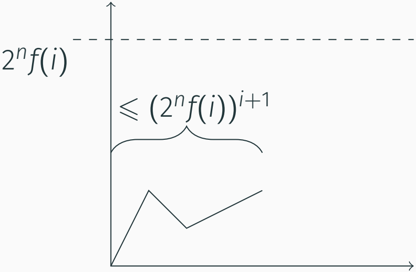
Take an arbitrary $(i+1)$-bounded, $(i+1)$-covering computation

1st case: Values on all places $[1..i+1]$ bounded by $2^n \cdot f(i)$:

Identify repetitions

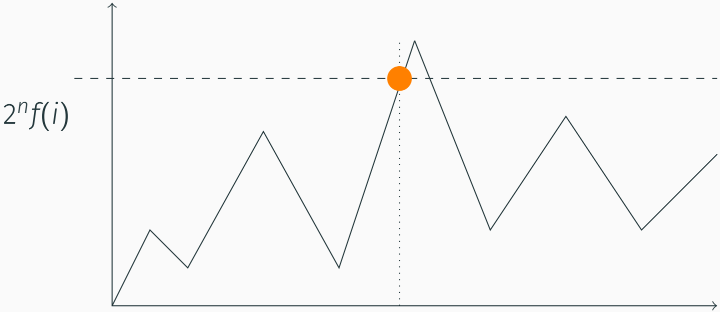
Delete loops

↳ Obtain new computation of length at most $(2^n f(i))^{i+1}$



Computing the Upward Closure - Upper Bound

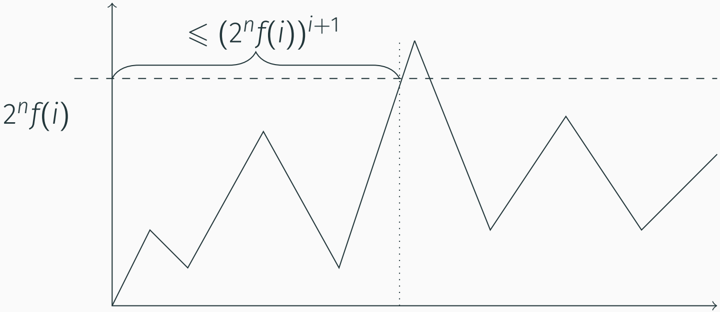
2^{nd} case: Some place, say $i+1$, exceeds $2^n \cdot f(i)$:



Computing the Upward Closure - Upper Bound

2^{nd} **case:** Some place, say $i+1$, exceeds $2^n \cdot f(i)$:

Treat first part as in 1^{st} case

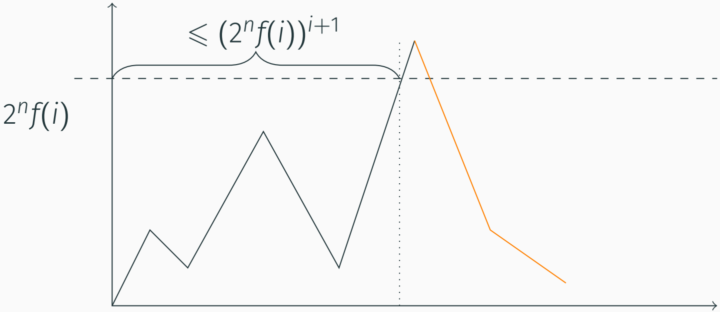


Computing the Upward Closure - Upper Bound

2^{nd} case: Some place, say $i+1$, exceeds $2^n \cdot f(i)$:

Treat first part as in 1^{st} case

Replace second part by i-covering, i-bounded computation of length $\leqslant f(i)$

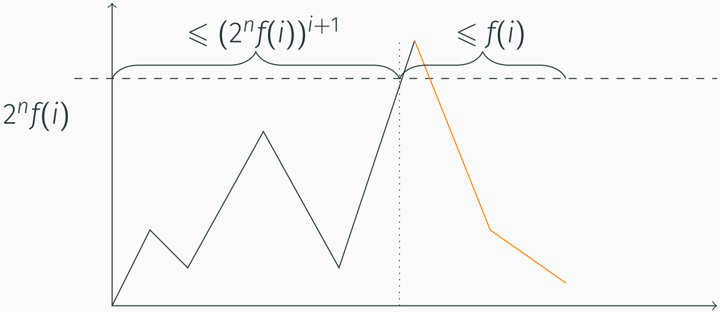


Computing the Upward Closure - Upper Bound

2^{nd} case: Some place, say $i+1$, exceeds $2^n \cdot f(i)$:

Treat first part as in 1^{st} case

Replace second part by i-covering, i-bounded computation of length $\leqslant f(i)$



Computing the Upward Closure - Upper Bound

What do we need to change?

Computing the Upward Closure - Upper Bound

What do we need to change?

Definition of $f(i)$:
upper bound on the length of a i-bounded, i-covering computations from an arbitrary initial marking that generate all minimal words

Computing the Upward Closure - Upper Bound

What do we need to change?

Definition of $f(i)$:
upper bound on the length of a i-bounded, i-covering computations from an arbitrary initial marking that generate all minimal words

1st case: Deleting loops creates a subword ✓

What do we need to change?

Definition of $f(i)$:
upper bound on the length of a i-bounded, i-covering computations from an arbitrary initial marking that generate all minimal words

1^{st} case: Deleting loops creates a subword ✓

2^{nd} case: Replacing second part of the computation ✗

Computing the Upward Closure - Upper Bound

What do we need to change?

Definition of $f(i)$:
upper bound on the length of a i-bounded, i-covering computations from an arbitrary initial marking that generate all minimal words

1st case: Deleting loops creates a subword ✓

2nd case: Replacing second part of the computation ✗

↳ handle with care

Computing the Upward Closure - Upper Bound

Finally:

The minimal words of $\mathcal{L}(N, M_0, M_f)\uparrow$ have a computation of length $\leqslant f(\ell) \leqslant 2^{2^{\mathcal{O}(n \cdot \log n)}}$.

Computing the Upward Closure - Upper Bound

Finally:
The minimal words of $\mathcal{L}(N, M_0, M_f)\uparrow$ have a computation of length $\leqslant f(\ell) \leqslant 2^{2^{\mathcal{O}(n \cdot \log n)}}$.

FSA can simulate the net for $f(\ell)$ steps to accept them (and their upward-closure)

Computing the Upward Closure - Lower Bound

Lemma (Lower Bound)

There is a family of Petri nets such that the upward closure cannot be represented by an FSA of less than doubly exponential size.

Computing the Upward Closure - Lower Bound

Lemma (Lower Bound)

There is a family of Petri nets such that the upward closure cannot be represented by an FSA of less than doubly exponential size.

Theorem (Lipton 1976)

Petri net reachability is EXPSPACE-hard.

Computing the Upward Closure - Lower Bound

Theorem (Lipton 1976)

Petri net reachability is EXPSPACE*-hard.*

Computing the Upward Closure - Lower Bound

Theorem (Lipton 1976)

Petri net reachability is EXPSPACE-*hard.*

In the proof, a Petri net of size polynomial in n simulates a counter machine with counter values bounded by 2^{2^n} (including zero tests!)

Computing the Upward Closure - Lower Bound

Theorem (Lipton 1976)

Petri net reachability is EXPSPACE*-hard.*

In the proof, a Petri net of size polynomial in n simulates a counter machine with counter values bounded by 2^{2^n} (including zero tests!)

Using this idea, we construct for each $n \in \mathbb{N}$ a Petri net with

$$\mathcal{L}(N(n), M_0, M_f) = \left\{ a^{2^{2^n}} \right\} .$$

Computing the Upward Closure - Lower Bound

Theorem (Lipton 1976)

Petri net reachability is EXPSPACE-*hard.*

In the proof, a Petri net of size polynomial in n simulates a counter machine with counter values bounded by 2^{2^n} (including zero tests!)

Using this idea, we construct for each $n \in \mathbb{N}$ a Petri net with

$$\mathcal{L}(N(n), M_0, M_f) = \left\{ a^{2^{2^n}} \right\} .$$

We obtain

$$\mathcal{L}(N(n), M_0, M_f)\uparrow = \left\{ a^k \;\middle|\; k \geqslant 2^{2^n} \right\} .$$

Computing the Downward Closure

Computing the Downward Closure

Computing the Downward Closure

Given:	Petri net (N, M_0, M_f).
Compute:	FSA A with $\mathcal{L}(A) = \mathcal{L}\big(N, M_0, M_f\big)\downarrow$.

Computing the Downward Closure

Computing the Downward Closure

Given: Petri net (N, M_0, M_f).

Compute: FSA A with $\mathcal{L}(A) = \mathcal{L}\big(N, M_0, M_f\big)\downarrow$.

Theorem

Upper bound: *One can compute an FSA of non-primitive recursive size representing the downward closure (in non-primitive recursive time).*

Lower bound: *This is optimal.*

Computing the Downward Closure - Upper Bound

Lemma (Upper Bound)

One can compute an FSA of non-primitive recursive size representing the downward closure.

Computing the Downward Closure - Upper Bound

Lemma (Upper Bound)

One can compute an FSA of non-primitive recursive size representing the downward closure.

Proof Sketch.

□

Computing the Downward Closure - Upper Bound

Lemma (Upper Bound)

One can compute an FSA of non-primitive recursive size representing the downward closure.

Proof Sketch.

The Karp-Miller tree (coverability graph) of the Petri net can be seen as finite automaton *KMT*

Computing the Downward Closure - Upper Bound

Lemma (Upper Bound)

One can compute an FSA of non-primitive recursive size representing the downward closure.

Proof Sketch.

The Karp-Miller tree (coverability graph) of the Petri net can be seen as finite automaton KMT

Its language is a subset of the downward closure,
$\mathcal{L}(N, M_0, M_f) \subseteq \mathcal{L}(KMT) \subseteq \mathcal{L}(N, M_0, M_f)\downarrow$

□

Computing the Downward Closure - Upper Bound

Lemma (Upper Bound)

One can compute an FSA of non-primitive recursive size representing the downward closure.

Proof Sketch.

The Karp-Miller tree (coverability graph) of the Petri net can be seen as finite automaton KMT

Its language is a subset of the downward closure,
$\mathcal{L}(N, M_0, M_f) \subseteq \mathcal{L}(KMT) \subseteq \mathcal{L}(N, M_0, M_f)\downarrow$

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

□

Computing the Downward Closure - Upper Bound

Lemma (Upper Bound)

One can compute an FSA of non-primitive recursive size representing the downward closure.

Proof Sketch.

The Karp-Miller tree (coverability graph) of the Petri net can be seen as finite automaton KMT

Its language is a subset of the downward closure,
$\mathcal{L}(N, M_0, M_f) \subseteq \mathcal{L}(KMT) \subseteq \mathcal{L}(N, M_0, M_f)\downarrow$

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

Its size might be non-primitive recursive

□

Computing the Downward Closure - Lower Bound

Lemma (Lower Bound)

There is a family of Petri nets such that the downward closure cannot be represented by an FSA of primitive recursive size.

Computing the Downward Closure - Lower Bound

Lemma (Lower Bound)

There is a family of Petri nets such that the downward closure cannot be represented by an FSA of primitive recursive size.

Inductive construction from [Mayr, Meyer 1981], adapted to labeled Petri nets:

$\forall n, x \in \mathbb{N} \;\; \exists \, (N(n), M_0^{(x)}, M_f)$ polynomial in $(n + x)$ such that

$$\mathcal{L}\left(N(n), M_0^{(x)}, M_f\right) = \left\{a^k \;\middle|\; k \leqslant Acker(n, x)\right\} = \mathcal{L}\left(N(n), M_0^{(x)}, M_f\right)\downarrow$$

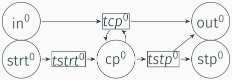
Lemma (Lower Bound)

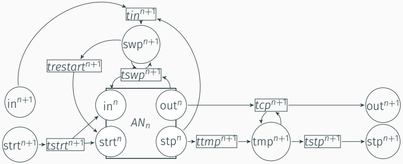
There is a family of Petri nets such that the downward closure cannot be represented by an FSA of primitive recursive size.

Inductive construction from [Mayr, Meyer 1981], adapted to labeled Petri nets:

$\forall n, x \in \mathbb{N} \;\; \exists \left(N(n), M_0^{(x)}, M_f\right)$ polynomial in $(n + x)$ such that

$$\mathcal{L}\left(N(n), M_0^{(x)}, M_f\right) = \left\{ a^k \;\middle|\; k \leqslant Acker(n, x) \right\} = \mathcal{L}\left(N(n), M_0^{(x)}, M_f\right)\downarrow$$





SRE in Downward Closure

Simple Regular Expression

Simple regular expression

$$sre ::= p \mid sre + sre$$
$$p ::= a \mid (a + \varepsilon) \mid \Gamma^* \mid p.p$$

where $\Gamma \subseteq \Sigma$

Known:

Downward and upward closures can be described by SREs

SRE in Downward Closure

SRE in Downward Closure

Given: SRE *sre*, Petri net (N, M_0, M_f).

Decide: $\mathcal{L}(sre) \subseteq \mathcal{L}\big(N, M_0, M_f\big)\downarrow$?

SRE in Downward Closure

<u>SRE in Downward Closure</u>

Given: SRE *sre*, Petri net (N, M_0, M_f).

Decide: $\mathcal{L}(sre) \subseteq \mathcal{L}\big(N, M_0, M_f\big)\downarrow$?

Theorem

SRE in Downward Closure is EXPSPACE-*complete*.

SRE in Downward Closure - Upper Bound

Lemma (Upper Bound)

SRE in Downward Closure can be solved in EXPSPACE.

SRE in Downward Closure - Upper Bound

Lemma (Upper Bound)

SRE in Downward Closure can be solved in EXPSPACE.

SRE is a choice among products

$$sre ::= p \mid sre + sre$$

SRE in Downward Closure - Upper Bound

Lemma (Upper Bound)

SRE in Downward Closure can be solved in EXPSPACE.

SRE is a choice among products

$$sre ::= p \mid sre + sre$$

Show inclusion for each product separately

$$p ::= a \mid (a + \varepsilon) \mid \Gamma^* \mid p.p$$

SRE in Downward Closure - Upper Bound

Lemma (Upper Bound)

SRE in Downward Closure can be solved in EXPSPACE.

SRE is a choice among products

$$sre ::= p \mid sre + sre$$

Show inclusion for each product separately

$$p ::= a \mid (a + \varepsilon) \mid \Gamma^* \mid p.p$$

Problem: Need to enforce that for each word in Γ^*, a covering computation exists

SRE in Downward Closure - Upper Bound

[Zetzsche 2015]: Downward closures computable iff a certain unboundedness problem decidable

SRE in Downward Closure - Upper Bound

[Zetzsche 2015]: Downward closures computable iff a certain unboundedness problem decidable

Theorem (Demri 2013)

The Simultaneous Unboundedness Problem for Petri Nets is EXPSPACE*-complete*.

SRE in Downward Closure - Upper Bound

[Zetzsche 2015]: Downward closures computable iff a certain unboundedness problem decidable

Theorem (Demri 2013)

The Simultaneous Unboundedness Problem for Petri Nets is EXPSPACE-complete.

<u>Simultaneous Unboundedness Problem for Petri Nets</u>

Given: Petri net N, marking M_0, set of places $X \subseteq P$

Decide: $\forall n \in \mathbb{N}\ \exists M_0[\sigma\rangle M$ with $M(p) \geqslant n\ \forall p \in X$?

SRE in Downward Closure - Upper Bound

Lemma (Upper Bound)

SRE in Downward Closure can be solved in EXPSPACE.

Handle each product p separately

SRE in Downward Closure - Upper Bound

Lemma (Upper Bound)

SRE in Downward Closure can be solved in EXPSPACE.

Handle each product p separately

For each expression Γ^* in p, add a place that tracks occurrence of all symbols in Γ
(also track the rest of p)

SRE in Downward Closure - Upper Bound

Lemma (Upper Bound)

SRE in Downward Closure can be solved in EXPSPACE.

Handle each product p separately

For each expression Γ^* in p, add a place that tracks occurrence of all symbols in Γ
(also track the rest of p)

Check whether the places for the Γ^* are simultaneously unbounded

SRE in Downward Closure - Lower Bound

Lemma (Lower Bound)

SRE in Downward Closure is EXPSPACE-hard.

SRE in Downward Closure - Lower Bound

Lemma (Lower Bound)

SRE in Downward Closure is EXPSPACE-*hard*.

Proof.

□

SRE in Downward Closure - Lower Bound

Lemma (Lower Bound)

SRE in Downward Closure is EXPSPACE-hard.

Proof.

Coverability for (unlabeled) Petri nets is EXPSPACE-hard

□

SRE in Downward Closure - Lower Bound

Lemma (Lower Bound)

SRE in Downward Closure is EXPSPACE-hard.

Proof.

Coverability for (unlabeled) Petri nets is EXPSPACE-hard

Label all transitions by ε

□

SRE in Downward Closure - Lower Bound

Lemma (Lower Bound)

SRE in Downward Closure is EXPSPACE-hard.

Proof.

Coverability for (unlabeled) Petri nets is EXPSPACE-hard

Label all transitions by ε

Note: $\mathcal{L}(N, M_0, M_f) = \{\varepsilon\}$ iff M_f coverable, $\emptyset$ else

□

SRE in Downward Closure - Lower Bound

Lemma (Lower Bound)

SRE in Downward Closure is EXPSPACE-hard.

Proof.

Coverability for (unlabeled) Petri nets is EXPSPACE-hard

Label all transitions by ε

Note: $\mathcal{L}(N, M_0, M_f) = \{\varepsilon\}$ iff M_f coverable, $\emptyset$ else

$\mathcal{L}(\emptyset^*) = \{\varepsilon\} \subseteq \mathcal{L}(N, M_0, M_f) \downarrow$ iff M_f coverable

□

SRE in Upward Closure

SRE in Upward Closure

SRE in Upward Closure

Given: SRE sre, Petri net (N, M_0, M_f).

Decide: $\mathcal{L}(sre) \subseteq \mathcal{L}\big(N, M_0, M_f\big)\!\uparrow$?

SRE in Upward Closure

SRE in Upward Closure

Given: SRE *sre*, Petri net (N, M_0, M_f).

Decide: $\mathcal{L}(sre) \subseteq \mathcal{L}\big(N, M_0, M_f\big)\uparrow$?

Theorem

SRE in Upward Closure is EXPSPACE-complete.

SRE in Upward Closure

SRE in Upward Closure

Given: SRE sre, Petri net (N, M_0, M_f).

Decide: $\mathcal{L}(sre) \subseteq \mathcal{L}\big(N, M_0, M_f\big)\uparrow$?

Theorem

SRE in Upward Closure is EXPSPACE-complete.

Note:

Lower bound (EXPSPACE-hardness) as for SRE in Downward Closure

SRE in Upward Closure - Upper Bound

Lemma (Upper Bound)

SRE in Upward Closure can be solved in EXPSPACE.

SRE in Upward Closure - Upper Bound

Lemma (Upper Bound)

SRE in Upward Closure can be solved in EXPSPACE.

SRE *sre* is a choice among products *p*

SRE in Upward Closure - Upper Bound

Lemma (Upper Bound)

SRE in Upward Closure can be solved in EXPSPACE.

SRE *sre* is a choice among products p

Check inclusion $\mathcal{L}(p) \subseteq \mathcal{L}(N, M_0, M_f)\uparrow$ for each product

SRE in Upward Closure - Upper Bound

Lemma (Upper Bound)

SRE in Upward Closure can be solved in EXPSPACE.

SRE *sre* is a choice among products p

Check inclusion $\mathcal{L}(p) \subseteq \mathcal{L}(N, M_0, M_f)\uparrow$ for each product

For each product, compute its minimal word:

$$\min(a) = a \qquad \min(p.p') = \min(p).\min(p')$$
$$\min(a + \varepsilon) = \varepsilon \qquad \min(\Gamma^*) = \varepsilon$$

SRE in Upward Closure - Upper Bound

Lemma (Upper Bound)

SRE in Upward Closure can be solved in EXPSPACE.

SRE *sre* is a choice among products p

Check inclusion $\mathcal{L}(p) \subseteq \mathcal{L}(N, M_0, M_f)\uparrow$ for each product

For each product, compute its minimal word:

$$\min(a) = a \qquad \min(p.p') = \min(p).\min(p')$$

$$\min(a + \varepsilon) = \varepsilon \qquad \min(\Gamma^*) = \varepsilon$$

We have $\mathcal{L}(p) \subseteq \mathcal{L}(N, M_0, M_f)\uparrow$ iff $\min(p) \in \mathcal{L}(N, M_0, M_f)\uparrow$

SRE in Upward Closure - Upper Bound

Lemma (Upper Bound)

SRE in Upward Closure can be solved in EXPSPACE.

SRE *sre* is a choice among products p

Check inclusion $\mathcal{L}(p) \subseteq \mathcal{L}(N, M_0, M_f)\uparrow$ for each product

For each product, compute its minimal word:

$$\min(a) = a \qquad \min(p.p') = \min(p).\min(p')$$
$$\min(a + \varepsilon) = \varepsilon \qquad \min(\Gamma^*) = \varepsilon$$

We have $\mathcal{L}(p) \subseteq \mathcal{L}(N, M_0, M_f)\uparrow$ iff $\min(p) \in \mathcal{L}(N, M_0, M_f)\uparrow$

Check this using a coverability query in a modified net

Being Downward/Upward closed

Being DC/UC

Being Downward/Upward Closed

Given: Petri net (N, M_0, M_f).

Decide: $\mathcal{L}\big(N, M_0, M_f\big) = \mathcal{L}\big(N, M_0, M_f\big)\downarrow / \uparrow$?

Being DC/UC

Being Downward/Upward Closed

Given: Petri net (N, M_0, M_f).

Decide: $\mathcal{L}(N, M_0, M_f) = \mathcal{L}(N, M_0, M_f)\downarrow / \uparrow$?

Theorem

Being DC and Being UC are decidable.

Being DC/UC

Being Downward/Upward Closed

Given: Petri net (N, M_0, M_f).

Decide: $\mathcal{L}\big(N, M_0, M_f\big) = \mathcal{L}\big(N, M_0, M_f\big)\downarrow / \uparrow$?

Theorem

Being DC and Being UC are decidable.

Note:

$\mathcal{L} \subseteq \mathcal{L}\uparrow$ and $\mathcal{L} \subseteq \mathcal{L}\downarrow$ always hold

Being DC/UC

Being Downward/Upward Closed

Given: Petri net (N, M_0, M_f).

Decide: $\mathcal{L}(N, M_0, M_f) = \mathcal{L}(N, M_0, M_f)\downarrow / \uparrow$?

Theorem

Being DC and Being UC are decidable.

Note:

$\mathcal{L} \subseteq \mathcal{L}\uparrow$ and $\mathcal{L} \subseteq \mathcal{L}\downarrow$ always hold

$\mathcal{L}\uparrow$ and $\mathcal{L}\downarrow$ are effectively regular

Regular lang. included in PN coverability lang.

Given: Petri net (N, M_0, M_f), FSA A.

Decide: $\mathcal{L}(A) \subseteq \mathcal{L}\big(N, M_0, M_f\big)$?

REG-IN-PNCOV

Regular lang. included in PN coverability lang.

Given: Petri net (N, M_0, M_f), FSA A.

Decide: $\mathcal{L}(A) \subseteq \mathcal{L}\big(N, M_0, M_f\big)$?

Theorem

Regular lang. included in PN coverability lang. is decidable.

REG-IN-PNCOV

Regular lang. included in PN coverability lang.

Given: Petri net (N, M_0, M_f), FSA A.

Decide: $\mathcal{L}(A) \subseteq \mathcal{L}\big(N, M_0, M_f\big)$?

Theorem

Regular lang. included in PN coverability lang. is decidable.

Theorem (Jancar, Esparza, Moller 1999)

Given Petri net (N, M_0) and FSA A.

$\mathcal{T}(A) \subseteq \mathcal{T}(N, M_0)$ *is decidable.*

Reducing to Trace Inclusion

Theorem (Jancar, Esparza, Moller 1999)

Given Petri net (N', M_0) *and FSA B.*

$\mathcal{T}(B) \subseteq \mathcal{T}(N', M_0)$ *is decidable*

Reducing to Trace Inclusion

Theorem (Jancar, Esparza, Moller 1999)

Given Petri net (N', M_0) *and FSA* B.

$\mathcal{T}(B) \subseteq \mathcal{T}(N', M_0)$ *is decidable*

where $$\mathcal{T}(B) = \left\{ w \;\middle|\; q_0 \xrightarrow{w} q \text{ for some state } q \right\},$$

$$\mathcal{T}(N', M_0) = \{ w \mid M_0[\sigma\rangle M \text{ for some } M \text{ and } \sigma, \lambda(\sigma) = w \}.$$

Reducing to Trace Inclusion

Theorem (Jancar, Esparza, Moller 1999)

Given Petri net (N', M_0) *and FSA* B.

$\mathcal{T}(B) \subseteq \mathcal{T}(N', M_0)$ *is decidable*

where $$\mathcal{T}(B) = \left\{ w \;\middle|\; q_0 \xrightarrow{w} q \text{ for some state } q \right\},$$

$$\mathcal{T}(N', M_0) = \{ w \mid M_0[\sigma\rangle M \text{ for some } M \text{ and } \sigma, \lambda(\sigma) = w \}.$$

Lemma

$\mathcal{L}(A) \subseteq \mathcal{L}\big(N, M_0, M_f\big)$ *iff* $\mathcal{T}(A.a) \subseteq \mathcal{T}(N.a, M_0)$.

Reducing to Trace Inclusion

Theorem (Jancar, Esparza, Moller 1999)

Given Petri net (N', M_0) *and FSA* B*.*

$\mathcal{T}(B) \subseteq \mathcal{T}(N', M_0)$ *is decidable*

where $\mathcal{T}(B) = \left\{ w \,\middle|\, q_0 \xrightarrow{w} q \text{ for some state } q \right\}$,

$\mathcal{T}(N', M_0) = \{ w \mid M_0[\sigma\rangle M \text{ for some } M \text{ and } \sigma, \lambda(\sigma) = w \}$.

Lemma

$\mathcal{L}(A) \subseteq \mathcal{L}(N, M_0, M_f)$ *iff* $\mathcal{T}(A.a) \subseteq \mathcal{T}(N.a, M_0)$*.*

where a fresh letter

$A.a$ reduced FSA for $\mathcal{L}(A).a$

$N.a$ = N plus a-labeled transition t_f consuming M_f

BPP Nets

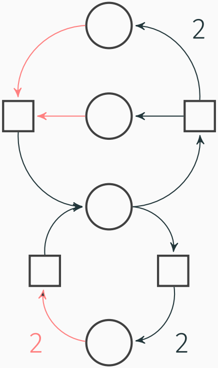
Results

	Petri nets
Compute UC	Doubly exponential*
Compute DC	Non-prim. rec.*
SRE in DC	EXPSPACE-compl.
SRE in UC	EXPSPACE-compl.
Being DC/UC	Decidable

* : Time for construction & size of minimal FSA

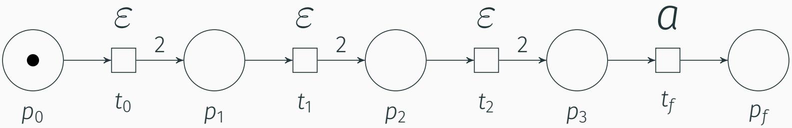
BPP Nets - Negative Example

In a BPP net, each transition consumes at most one token.



BPP Nets - Positive Example

In a BPP net, each transition consumes at most one token.



Results

	Petri nets
Compute UC	Doubly exponential*
Compute DC	Non-prim. rec.*
SRE in DC	EXPSPACE-compl.
SRE in UC	EXPSPACE-compl.
Being DC/UC	Decidable

* : Time for construction & size of minimal FSA

Results

	Petri nets	BPP nets
Compute UC	Doubly exponential*	Exponential*
Compute DC	Non-prim. rec.*	Exponential*
SRE in DC	EXPSPACE-compl.	NP-compl.
SRE in UC	EXPSPACE-compl.	NP-compl.
Being DC/UC	Decidable	

* : Time for construction & size of minimal FSA

Results

	Petri nets	BPP nets	Techniques for upper bound	Techniques for lower bound
Compute UC	Doubly exponential*	Exponential*	Unfoldings	Initial ex.
Compute DC	Non-prim. rec.*	Exponential*	Unfoldings	Initial ex.
SRE in DC	EXPSPACE-compl.	NP-compl.	Presburger	Coverability
SRE in UC	EXPSPACE-compl.	NP-compl.	Coverability	Coverability
Being DC/UC	Decidable			

* : Time for construction & size of minimal FSA

Thank you!

Questions?