On the Upward/Downward Closures of Petri Nets

Mohammed Faouzi Atig\textsuperscript{1}, Roland Meyer\textsuperscript{2}, Sebastian Muskalla\textsuperscript{2} and Prakash Saivasan\textsuperscript{2}

August 25, MFCS 2017, Aalborg

\textsuperscript{1} Uppsala University, Sweden
mohamed_faouzi.atig@it.uu.se
\textsuperscript{2} TU Braunschweig, Germany
{roland.meyer, s.muskalla, p.saivasan}@tu-bs.de
Goal

Study the *upward and downward closures* of *Petri net* coverability languages
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

Study the **upward and downward closures of Petri net coverability languages**

Upward/Downward closures in general

Good:

- Always regular
Goal

Study the upward and downward closures of Petri net coverability languages

Upward/Downward closures in general

Good:

Always simply regular
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

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

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
\[(Labeled) \text{ Petri Nets}\]

\[
N = \left( \begin{array}{c}
\{p_0, p_1, p_2, p_3, p_f\}, \{t_0, t_1, t_2, t_f\}, F \\
\text{places } P \\
\text{transitions } T
\end{array} \right)
\]
(Labeled) Petri Nets

\[ N = \left( \begin{array}{c} \{p_0, p_1, p_2, p_3, p_f\} \\ \text{places } P \\ \{t_0, t_1, t_2, t_f\}, F \\ \text{transitions } T \end{array} \right) \]

\[
\begin{pmatrix}
1 \\
0 \\
0 \\
0 \\
\| \\
M_0
\end{pmatrix}
\]
(Labeled) Petri Nets

\[ N = \left( \begin{array}{c} \{p_0, p_1, p_2, p_3, p_f\}, \{t_0, t_1, t_2, t_f\}, F \end{array} \right) \]

\[ M_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \| \\ M_0 \end{pmatrix} \]

\[ [t_0] \]
(Labeled) Petri Nets

\[ N = \left( \begin{array}{c}
\{p_0, p_1, p_2, p_3, p_f\}, \\
\{t_0, t_1, t_2, t_f\}, F
\end{array} \right) \]

\[ \begin{pmatrix}
1 \\
0 \\
0 \\
0 \\
0
\end{pmatrix} \quad [t_0] \quad \begin{pmatrix}
0 \\
0 \\
0 \\
2^3 \\
M_f
\end{pmatrix} \]
(Labeled) Petri Nets

\[ N = \left( \begin{array}{c}
\{ p_0, p_1, p_2, p_3, p_f \}, \{ t_0, t_1, t_2, t_f \}, F \\
\end{array} \right) \]

\[ M_0 = \begin{pmatrix}
1 \\
0 \\
0 \\
0 \\
0 \\
\end{pmatrix}, \quad M_f = \begin{pmatrix}
0 \\
0 \\
0 \\
2^3 \\
\end{pmatrix} \]
### (Labeled) Petri Nets

A Petri net is a directed graph with two types of nodes: places (circles) and transitions (squares). Edges connect places to transitions. Places can hold tokens, which are represented by dots. Transitions can be enabled (indicated by a black dot) and can fire, changing the number of tokens in the connected places.

The net in the image is labeled with places $p_0$, $t_0$, $p_1$, $t_1$, $p_2$, $t_2$, $p_3$, $t_f$, and a final place $p_f$.

The Petri net is mathematically described by the set $N$:

$$N = \left( \{ p_0, p_1, p_2, p_3, p_f \}, \{ t_0, t_1, t_2, t_f \}, F \right)$$

Where $F$ represents the firing function, which determines when transitions can fire.

The firing function can be represented by a matrix $M_0$.

$$M_0 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

And the final marking $M_f$ can be represented as:

$$M_f = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 2^3 \end{pmatrix}$$
Petri Nets

\[ N = \left( \begin{array}{c} \{ p_0, p_1, p_2, p_3, p_f \} \setminus \{ t_0, t_1, t_2, t_f \}, F \\ \text{places } P \\ \text{transitions } T \end{array} \right) \]

\[ \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \left[ t_0 t_1 t_2 t_2 t_f t_f t_f t_f t_f t_f t_f t_f \right] \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 2^3 \end{pmatrix} \]

\[ M_0 \]

\[ p_f \]
(Labeled) Petri Nets

\[
N = \left( \begin{array}{c}
\{p_0, p_1, p_2, p_3, p_f\}, \\
\{t_0, t_1, t_2, t_f\}, F
\end{array} \right)
\]

\[
\begin{pmatrix}
1 \\
0 \\
0 \\
0 \\
0
\end{pmatrix} \begin{pmatrix}
t_0 \\
t_1 \\
t_2 \\
t_2 \\
t_2 \\
t_2 \\
t_2 \\
t_2 \\
t_2 \\
t_f \\
t_f \\
t_f \\
t_f \\
t_f
\end{pmatrix} \begin{pmatrix}
0 \\
0 \\
0 \\
0 \\
8
\end{pmatrix} \geq \begin{pmatrix}
0 \\
0 \\
0 \\
0 \\
2^3
\end{pmatrix}
\]

\[
M_0 \|
\end{pmatrix}
\]

\[
M_f
\]

\[
= \begin{pmatrix}
1 \\
0 \\
0 \\
0 \\
0
\end{pmatrix}
\begin{pmatrix}
t_0 \\
t_1 \\
t_2 \\
t_2 \\
t_2 \\
t_2 \\
t_2 \\
t_2 \\
t_2 \\
t_f \\
t_f \\
t_f \\
t_f \\
t_f
\end{pmatrix} \begin{pmatrix}
0 \\
0 \\
0 \\
0 \\
8
\end{pmatrix} \geq \begin{pmatrix}
0 \\
0 \\
0 \\
0 \\
2^3
\end{pmatrix}
\]

\[
M_0 \|
\end{pmatrix}
\]

\[
M_f
\]
(Labeled) Petri Nets

\[ N = \left( \{a\}, \{p_0, p_1, p_2, p_3, p_f\}, \{t_0, t_1, t_2, t_f\}, F, \lambda \right) \]

\[ \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} t_0 t_1 t_2 t_2 t_2 t_f t_f t_f t_f t_f t_f \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \geq \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \]

\[ M_0 \]

\[ M_f \]
(Labeled) Petri Nets

\[ N = (\{a\}, \{p_0, p_1, p_2, p_3, p_f\}, \{t_0, t_1, t_2, t_f\}, F, \lambda) \]

\[
\begin{pmatrix}
1 \\
0 \\
0 \\
0 \\
\end{pmatrix}
\begin{pmatrix}
t_0 & t_1 & t_2 & t_f & t_f & t_f & t_f & t_f & t_f & t_f \\
\end{pmatrix}
\triangleq \lambda
\begin{pmatrix}
0 \\
0 \\
0 \\
8 \\
\end{pmatrix}
\begin{pmatrix}
0 \\
0 \\
0 \\
2^3 \\
\end{pmatrix}
\]

\[ \varepsilon e e e e e a a a a a a a a = a^8 \]

\[ M_0 \]

\[ M_f \]
Coverability Language

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

Coverability language

$$L(N, M_0, M_f) = \{ \lambda(\sigma) \mid M_0[\sigma]M, M \geq M_f \}$$
Coverability Language

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

**Coverability language**

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

In the example:

$$\mathcal{L}(N, M_0, M_f) = \{a^8\}$$
Subword relation

\[ v \leq w \text{ iff } v \text{ obtained from } w \text{ by deleting letters} \]
\[ \text{iff } w \text{ obtained from } v \text{ by inserting letters} \]
Upward and Downward Closure

Subword relation

\[ v \preceq w \quad \text{iff} \quad v \text{ obtained from } w \text{ by deleting letters} \]

\[ \text{iff} \quad w \text{ obtained from } v \text{ by inserting letters} \]

Upward closure

\[ \mathcal{L}^\uparrow = \{ w \mid \exists v \in \mathcal{L} : v \preceq w \} \]
Subword relation

\[ v \preceq w \iff v \text{ obtained from } w \text{ by deleting letters} \]
\[ \iff w \text{ obtained from } v \text{ by inserting letters} \]

Upward closure

\[ \mathcal{L}^\uparrow = \{ w \mid \exists v \in \mathcal{L} : v \preceq w \} \]

Downward closure

\[ \mathcal{L}^\downarrow = \{ w \mid \exists v \in \mathcal{L} : w \preceq v \} \]
Upward and Downward Closure

Subword relation

\( v \leq w \) iff \( v \) obtained from \( w \) by deleting letters

iff \( w \) obtained from \( v \) by inserting letters

Upward closure

\[ \mathcal{L}^{\uparrow} = \{ w \mid \exists v \in \mathcal{L} : v \leq w \} \]

Downward closure

\[ \mathcal{L}^{\downarrow} = \{ w \mid \exists v \in \mathcal{L} : w \leq v \} \]

In the example:

\[ \mathcal{L}(N, M_0, M_f)^{\uparrow} = \{ a^k \mid k \geq 8 \} \]

\[ \mathcal{L}(N, M_0, M_f)^{\downarrow} = \{ a^k \mid k \leq 8 \} \]
Computing the Upward Closure
### Computing the Upward Closure

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

**Compute:** FSA \(A\) with \(\mathcal{L}(A) = \mathcal{L}(N, M_0, M_f)^\uparrow\).
Computing the Upward Closure

| Given: | Petri net \((N, M_0, M_f)\). |
| Compute: | FSA \(A\) with \(L(A) = L(N, M_0, M_f)\)↑. |

**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.*
Lemma (Upper Bound)

One can compute an FSA of \textit{doubly exponential size} for the upward closure in \textit{doubly exponential time}.
### 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).*
Theorem (Rackoff 1978)

*Petri net coverability can be solved using exponential space (and doubly exponential time).*
### Theorem (Rackoff 1978)

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

Assume places are ordered, $P = [1..\ell]$. 
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.
**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 $f(\ell) \leq 2^{2^{O(n \cdot \log n)}}$
Theorem (Rackoff 1978)

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

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

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

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

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

Show $f(i + 1) \leq (2^n f(i))^{i+1} + f(i)$.
Computing the Upward Closure - Upper Bound

\[ f(i + 1) \leq (2^n f(i))^{i+1} + f(i) \]

Take an arbitrary \((i + 1)\)-bounded, \((i + 1)\)-covering computation
Computing the Upward Closure - Upper Bound

\[ f(i + 1) \leq (2^n f(i))^{i+1} + f(i) \]

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

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

\[ 2^n f(i) \]
Computing the Upward Closure - Upper Bound

\[ f(i + 1) \leq (2^n f(i))^{i+1} + f(i) \]

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

\[ f(i + 1) \leq (2^n f(i))^{i+1} + f(i) \]

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

\[ f(i + 1) \leq (2^n f(i))^{i+1} + f(i) \]

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

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

- Identify **repetitions**
- Delete **loops**

\[ \downarrow \text{ Obtain new computation of length at most } (2^n f(i))^{i+1} \]
2nd case: Some place, say $i + 1$, exceeds $2^n \cdot f(i)$:
2\textsuperscript{nd} case: Some place, say $i + 1$, exceeds $2^n \cdot f(i)$:

Treat first part as in 1\textsuperscript{st} case
2nd case: Some place, say $i + 1$, exceeds $2^n \cdot f(i)$:

Treat first part as in 1st case

Replace second part by $i$-covering, $i$-bounded computation of length $\leq f(i)$
2\textsuperscript{nd} case: Some place, say $i + 1$, exceeds $2^n \cdot f(i)$:

Treat first part as in 1\textsuperscript{st} case

Replace second part by $i$-covering, $i$-bounded computation of length $\leq f(i)$
What do we need to change?
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
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\textsuperscript{st} case: Deleting loops creates a subword ✓
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

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

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

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

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

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

<table>
<thead>
<tr>
<th>Theorem (Lipton 1976)</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Petri net reachability is EXPSPACE-hard.</em></td>
</tr>
</tbody>
</table>
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

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

We obtain

\[
\mathcal{L}(N(n), M_0, M_f)^\uparrow = \left\{ a^k \mid k \geq 2^{2^n} \right\}.
\]
Computing the Downward Closure
Computing the Downward Closure

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

Given: Petri net \((N, M_0, M_f)\).
Compute: FSA \(A\) with \(L(A) = L(N, M_0, M_f) \downarrow\).

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.
Lemma (Upper Bound)

One can compute an FSA of *non-primitive recursive size* representing the downward closure.
Lemma (Upper Bound)

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

Proof Sketch.
**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 $KMT$. 
Lemma (Upper Bound)

One can compute an FSA of \textit{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, 
\[ \mathcal{L}(N, M_0, M_f) \subseteq \mathcal{L}(KMT) \subseteq \mathcal{L}(N, M_0, M_f) \downarrow \]
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,

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

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

\[
\forall n, x \in \mathbb{N} \ \exists \ (N(n), M_0^{(x)}, M_f) \text{ polynomial in } (n + x) \text{ such that } \\
\mathcal{L}(N(n), M_0^{(x)}, M_f) = \left\{ a^k \mid k \leq Acker(n, x) \right\} = \mathcal{L}(N(n), M_0^{(x)}, M_f) \downarrow
\]
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:

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

\[ \mathcal{L}(N(n), M_0^{(x)}, M_f) = \{ a^k \mid k \leq \text{Acker}(n, x) \} = \mathcal{L}(N(n), M_0^{(x)}, M_f) \downarrow \]
SRE in Downward Closure
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

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

Decide: \( \mathcal{L}(sre) \subseteq \mathcal{L}(N, M_0, M_f) \downarrow \) ?
SRE in Downward Closure

Given: SRE \( sre \), Petri net \((N, M_0, M_f)\).
Decide: \( \mathcal{L}(sre) \subseteq \mathcal{L}(N, M_0, M_f) \downarrow \) ?

Theorem

\( SRE \) in Downward Closure is \textsc{EXPSPACE-complete}. \( \)
<table>
<thead>
<tr>
<th>Lemma (Upper Bound)</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>SRE in Downward Closure can be solved in EXPSPACE.</em></td>
</tr>
</tbody>
</table>
**Lemma (Upper Bound)**

*SRE in Downward Closure can be *solved in* EXPSPACE.*

SRE is a choice among **products**

\[
sre ::= p \mid sre + sre
\]
Lemma (Upper Bound)

SRE in Downward Closure can be solved in \textit{EXPSPACE}.

SRE is a choice among products

\[
sre ::= p \mid sre + sre
\]

Show inclusion for each product separately

\[
p ::= a \mid (a + \varepsilon) \mid \Gamma^* \mid p.p
\]
Lemma (Upper Bound)

SRE in Downward Closure can be solved in \textit{EXPSPACE}.

SRE is a choice among \textit{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 \textit{each word in }\Gamma^*, a covering computation exists
[Zetzsche 2015]: Downward closures computable iff a certain unboundedness problem decidable
[Zetzsche 2015]: Downward closures computable iff a certain unboundedness problem decidable

**Theorem (Demri 2013)**

*The Simultaneous Unboundedness Problem for Petri Nets is EXPSPACE-complete.*
[Zetzsche 2015]: Downward closures computable iff a certain unboundedness problem decidable

**Theorem (Demri 2013)**

*The Simultaneous Unboundedness Problem for Petri Nets is \(\text{EXPSPACE-complete} \).*

**Simultaneous Unboundedness Problem for Petri Nets**

**Given:** Petri net \( N \), marking \( M_0 \), set of places \( X \subseteq P \)

**Decide:** \( \forall n \in \mathbb{N} \) \( \exists M_0[\sigma]M \) with \( M(p) \geq n \forall p \in X \)?
Lemma (Upper Bound)

SRE in Downward Closure can be solved in EXPSPACE.

Handle each product $p$ separately
Lemma (Upper Bound)

\textit{SRE in Downward Closure can be solved in EXPSPACE.}

Handle each product $p$ separately

For each expression $\Gamma^*$ in $p$, add a place that \textit{tracks occurrence of all symbols in } $\Gamma$
(also track the rest of $p$)
**Lemma (Upper Bound)**

*SRE in Downward Closure can be solved in EXPSPACE.*

Handle each product $p$ separately

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

Check whether the places for the $\Gamma^*$ are **simultaneously unbounded**
Lemma (Lower Bound)

SRE in Downward Closure is EXPSPACE-hard.
Lemma (Lower Bound)

SRE in Downward Closure is \textsc{EXPSPACE-hard}.

Proof.
Lemma (Lower Bound)

SRE in Downward Closure is **EXPSPACE-hard**.

Proof.

*Coverability* for (unlabeled) Petri nets is EXPSPACE-hard.
Lemma (Lower Bound)

*SRE in Downward Closure is EXPSPACE-hard.*

Proof.

- **Coverability** for (unlabeled) Petri nets is EXPSPACE-hard.
- Label all transitions by $\varepsilon$. 

Lemma (Lower Bound)

SRE in Downward Closure is \textbf{EXPSPACE-hard}.

Proof.

- \textbf{Coverability} for (unlabeled) Petri nets is \textbf{EXPSPACE-hard}
- Label all transitions by $\varepsilon$
- Note: $\mathcal{L}(N, M_0, M_f) = \{\varepsilon\}$ iff $M_f$ coverable, $\emptyset$ else
Lemma (Lower Bound)

SRE in Downward Closure is \textit{EXPSPACE-hard}.

Proof.

\textbf{Coverability} for (unlabeled) Petri nets is \textit{EXPSPACE-hard}

Label all transitions by $\varepsilon$

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

Given: SRE $sre$, Petri net $(N, M_0, M_f)$.
Decide: $\mathcal{L}(sre) \subseteq \mathcal{L}(N, M_0, M_f) \uparrow$ ?
SRE in Upward Closure

Given: SRE sre, Petri net \((N, M_0, M_f)\).
Decide: \(L(sre) \subseteq L(N, M_0, M_f)\uparrow\) ?

Theorem

SRE in Upward Closure is \textit{EXPSPACE-complete}.
SRE in Upward Closure

Given: SRE \( sre \), Petri net \((N, M_0, M_f)\).
Decide: \( \mathcal{L}(sre) \subseteq \mathcal{L}(N, M_0, M_f) \uparrow \) ?

**Theorem**

*SRE in Upward Closure is EXPSPACE-complete.*

**Note:**

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

SRE in Upward Closure can be solved in EXPSPACE.
Lemma (Upper Bound)

SRE in Upward Closure can be solved in EXPSPACE.

SRE $sre$ is a choice among products $p$
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
Lemma (Upper Bound)

SRE in Upward Closure can be solved in EXPSPACE.

SRE sre is a choice among products $p$

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

For each product, compute its minimal word:

\[
\begin{align*}
\min(a) &= a \\
\min(p \cdot p') &= \min(p) \cdot \min(p') \\
\min(a + \varepsilon) &= \varepsilon \\
\min(\Gamma^*) &= \varepsilon
\end{align*}
\]
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:

\[
\begin{align*}
\min(a) &= a \\
\min(p.p') &= \min(p).\min(p') \\
\min(a + \varepsilon) &= \varepsilon \\
\min(\Gamma^*) &= \varepsilon
\end{align*}
\]

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 \textit{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 \textit{minimal word}:

\[
\begin{align*}
\min(a) &= a \\
\min(p.p') &= \min(p).\min(p') \\
\min(a + \varepsilon) &= \varepsilon \\
\min(\Gamma^*) &= \varepsilon
\end{align*}
\]

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

Check this using a \textit{coverability query} in a modified net
Being Downward/Upward closed
Being Downward/Upward Closed

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

Decide: \(\mathcal{L}(N, M_0, M_f) = \mathcal{L}(N, M_0, M_f) \downarrow / \uparrow \ ?\)
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 \textit{decidable}. 
Being DC/UC

Being Downward/Upward Closed

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

Decide: \(L(N, M_0, M_f) = L(N, M_0, M_f) \downarrow / \uparrow \) ?

Theorem

Being DC and Being UC are **decidable**.

Note:

\[ L \subseteq L \uparrow \textrm{ and } L \subseteq L \downarrow \] 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:**

\[ \mathcal{L} \subseteq \mathcal{L} \uparrow \text{ and } \mathcal{L} \subseteq \mathcal{L} \downarrow \text{ always hold} \]

\[ \mathcal{L} \uparrow \text{ and } \mathcal{L} \downarrow \text{ are effectively regular} \]
Regular lang. included in PN coverability lang.

Given: Petri net \((N, M_0, M_f)\), FSA \(A\).
Decide: \(\mathcal{L}(A) \subseteq \mathcal{L}(N, M_0, M_f)\) ?
Regular lang. included in PN coverability lang.

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

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

Theorem

Regular lang. included in PN coverability lang. is \textit{decidable}.
Regular lang. included in PN coverability lang.

Given: Petri net \((N, M_0, M_f)\), FSA A.
Decide: \(\mathcal{L}(A) \subseteq \mathcal{L}(N, M_0, M_f)\) ?

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.
**Theorem (Jancar, Esparza, Moller 1999)**

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

\[ T(B) \subseteq T(N', M_0) \text{ is decidable} \]
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 \(\mathcal{T}(B) = \{w \mid q_0 \xrightarrow{w} q \text{ for some state } q\}\),

\(\mathcal{T}(N', M_0) = \{w \mid M_0[\sigma]M \text{ for some } M \text{ and } \sigma, \lambda(\sigma) = w\}\).
Reducing to Trace Inclusion

**Theorem (Jancar, Esparza, Moller 1999)**

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

$T(B) \subseteq T(N', M_0)$ is decidable

where $T(B) = \{ w \mid q_0 \xrightarrow{w} q \text{ for some state } q \}$,

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

**Lemma**

$\mathcal{L}(A) \subseteq \mathcal{L}(N, M_0, M_f)$ iff $T(A.a) \subseteq T(N.a, M_0)$. 
Reducing to Trace Inclusion

Theorem (Jancar, Esparza, Moller 1999)

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

\(T(B) \subseteq T(N', M_0)\) is decidable

where \(T(B) = \{ w \mid q_0 \xrightarrow{w} q \text{ for some state } q \}\),

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

Lemma

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

where \(a\) 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

<table>
<thead>
<tr>
<th></th>
<th>Petri nets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute UC</td>
<td>Doubly exponential*</td>
</tr>
<tr>
<td>Compute DC</td>
<td>Non-prim. rec.*</td>
</tr>
<tr>
<td>SRE in DC</td>
<td>EXPSPACE-compl.</td>
</tr>
<tr>
<td>SRE in UC</td>
<td>EXPSPACE-compl.</td>
</tr>
<tr>
<td>Being DC/UC</td>
<td>Decidable</td>
</tr>
</tbody>
</table>

*: Time for construction & size of minimal FSA
In a **BPP net**, each transition consumes at most one token.
In a BPP net, each transition consumes at most one token.
<table>
<thead>
<tr>
<th>Petri nets</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute UC</td>
<td>Doubly exponential*</td>
</tr>
<tr>
<td>Compute DC</td>
<td>Non-prim. rec.*</td>
</tr>
<tr>
<td>SRE in DC</td>
<td>EXPSPACE-compl.</td>
</tr>
<tr>
<td>SRE in UC</td>
<td>EXPSPACE-compl.</td>
</tr>
<tr>
<td>Being DC/UC</td>
<td>Decidable</td>
</tr>
</tbody>
</table>

*: Time for construction & size of minimal FSA
## Results

<table>
<thead>
<tr>
<th></th>
<th>Petri nets</th>
<th>BPP nets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute UC</td>
<td>Doubly exponential*</td>
<td>Exponential*</td>
</tr>
<tr>
<td>Compute DC</td>
<td>Non-prim. rec.*</td>
<td>Exponential*</td>
</tr>
<tr>
<td>SRE in DC</td>
<td>EXPSPACE-compl.</td>
<td>NP-compl.</td>
</tr>
<tr>
<td>SRE in UC</td>
<td>EXPSPACE-compl.</td>
<td>NP-compl.</td>
</tr>
<tr>
<td>Being DC/UC</td>
<td>Decidable</td>
<td></td>
</tr>
</tbody>
</table>

*: Time for construction & size of minimal FSA
## Results

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

*: Time for construction & size of minimal FSA
Thank you!
Questions?