Acyclic Petri and Workflow Nets with Resets

Dmitry Chistikov
University of Warwick
United Kingdom

Wojciech Czerwiński
University of Warsaw
Poland

Piotr Hofman
University of Warsaw
Poland

Filip Mazowiecki
University of Warsaw
Poland

Henry Sinclair-Banks
University of Warwick
United Kingdom

FSTTCS’23
18th December 2023
International Institute of Information Technology, Hyderabad, India
Petri Nets

Marking:
The number of tokens in each place.

Input to decision problems are: a Petri net $N$, an initial marking $u$, and a target marking $v$.

Reachability:
Does there exist a run from $u$ to $v$ in $N$?

Coverability:
Does there exist a run from $u$ to $w$ in $N$ where $w \not\preceq v$?
Marking:
The number of tokens in each place.

Run:
As sequence of markings.

Input to decision problems are: a Petri net $N$, an initial marking $u$, and a target marking $v$.

Reachability:
Does there exist a run from $u$ to $v$ in $N$?

Coverability:
Does there exist a run from $u$ to $w$ in $N$ where $w \not= v$?

Henry Sinclair-Banks

Acyclic Petri and Workflow Nets with Resets
Petri Nets

Marking: The number of tokens in each place.

Run: As sequence of markings.

Input to decision problems are: a Petri net $N$, an initial marking $u$, and a target marking $v$.

Reachability: Does there exist a run from $u$ to $v$ in $N$?

Coverability: Does there exist a run from $u$ to $w$ in $N$ where $w \not= v$?

Henry Sinclair-Banks

Acyclic Petri and Workflow Nets with Resets
**Petri Nets**

**Marking:** The number of tokens in each place.

\[ p_1, p_2, q, r_1, r_2 \]

\[ (3, 4, 2, 5, 1) \]

**Run:** A sequence of markings.

\[ (4, 5, 0, 0, 0) \xrightarrow{t_1} (3, 4, 3, 0, 0) \xrightarrow{t_2} (3, 4, 2, 5, 1) \]

Input to decision problems are: a Petri net \( N \), an initial marking \( u \), and a target marking \( v \).

**Reachability:** Does there exist a run from \( u \) to \( v \) in \( N \)?

**Coverability:** Does there exist a run from \( u \) to \( w \) in \( N \) where \( w \geq v \)?
**Features:** A designated initial place $i$ that cannot be produced to, a designated final place $f$ that cannot be consumed from, and all places and transitions are on some path from $i$ to $f$.

The complexities of reachability and coverability are the same for workflow nets as for Petri nets.
Petri Nets with Resets

Semantics:
First consume tokens, then reset places, then produce tokens.

Reachability in Petri nets with resets is undecidable. [Araki and Kasami '76]

Coverability in Petri nets with resets is Ackermann-complete. [Schnoebelen '10]
[Figueria, Figueira, Schmitz, and Schnoebelen '11]
**Semantics:** First consume tokens, then reset places, then produce tokens.
Semantics: First consume tokens, then reset places, then produce tokens.
Semantics: First consume tokens, then reset places, then produce tokens.
Semantics: First consume tokens, then reset places, then produce tokens.
**Semantics:** First consume tokens, then reset places, then produce tokens.

Reachability in Petri nets with resets is undecidable. \[\text{[Araki and Kasami '76]}\]

Coverability in Petri nets with resets is Ackermann-complete. \[\text{[Schnoebelen '10]}\]

\[\text{[Figueira, Figueira, Schmitz, and Schnoebelen '11]}\]
Semantics: First consume tokens, then reset places, then produce tokens.

Acyclicity: No cycles in the graph of places with consumption arcs and production arcs. Reset edges do not count!

We study both acyclic Petri nets with resets and acyclic workflow nets with resets.

Without resets, reachability and coverability in acyclic Petri and workflow nets are all NP-complete.
### Results

<table>
<thead>
<tr>
<th></th>
<th>Coverability</th>
<th>Reachability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acyclic workflow nets with resets</td>
<td>PSPACE-hard</td>
<td>in PSPACE</td>
</tr>
<tr>
<td>Acyclic Petri nets with resets</td>
<td>in PSPACE</td>
<td></td>
</tr>
</tbody>
</table>

**Theorem 1:** Reachability in acyclic workflow nets with resets is in PSPACE.

**Theorem 2:** Coverability in acyclic Petri nets with resets is in PSPACE.

**Theorem 3:** Coverability in acyclic workflow nets with resets is PSPACE-hard.
### Results

<table>
<thead>
<tr>
<th></th>
<th>Coverability</th>
<th>Reachability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acyclic workflow nets with resets</td>
<td>PSPACE-hard</td>
<td>PSPACE-complete</td>
</tr>
<tr>
<td>Acyclic Petri nets with resets</td>
<td>PSPACE-complete</td>
<td></td>
</tr>
</tbody>
</table>

**Theorem 1:** Reachability in acyclic workflow nets with resets is in PSPACE.

**Theorem 2:** Coverability in acyclic Petri nets with resets is in PSPACE.

**Theorem 3:** Coverability in acyclic workflow nets with resets is PSPACE-hard.
## Results

<table>
<thead>
<tr>
<th></th>
<th>Coverability</th>
<th>Reachability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acyclic workflow nets with resets</td>
<td>PSPACE-complete</td>
<td>PSPACE-complete</td>
</tr>
<tr>
<td>Acyclic Petri nets with resets</td>
<td>PSPACE-complete</td>
<td></td>
</tr>
</tbody>
</table>

**Theorem 1:** Reachability in acyclic workflow nets with resets is in PSPACE.

**Theorem 2:** Coverability in acyclic Petri nets with resets is in PSPACE.

**Theorem 3:** Coverability in acyclic workflow nets with resets is PSPACE-hard.
## Results

<table>
<thead>
<tr>
<th></th>
<th>Coverability</th>
<th>Reachability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acyclic workflow nets with resets</td>
<td>PSPACE-complete</td>
<td>PSPACE-complete</td>
</tr>
<tr>
<td>Acyclic Petri nets with resets</td>
<td>PSPACE-complete</td>
<td>Undecidable</td>
</tr>
</tbody>
</table>

**Theorem 1:** Reachability in acyclic workflow nets with resets is in PSPACE.

**Theorem 2:** Coverability in acyclic Petri nets with resets is in PSPACE.

**Theorem 3:** Coverability in acyclic workflow nets with resets is PSPACE-hard.

**Theorem 4:** Reachability in acyclic Petri nets with resets is undecidable.

*All results hold for both unary encoding and binary encoding.*
**Ideas: PSPACE Upper Bound**

**Theorem 1:** Reachability in acyclic workflow nets with resets is in PSPACE.

**Proof idea:** Suppose reachability holds, \((1, 0, \ldots, 0) \rightarrow \cdots \rightarrow (0, \ldots, 0, 1)\) in \(\mathcal{N}\).

Ignore resets and consider number of times each transition can be fired.

Fact: The workflow features imply that all transitions consume at least one token.

\[
\text{So, } t_1 \text{ can be only be fired once, then } t_2 \text{ can be fired at most } m \text{ times, } \ldots \quad \ldots \ t_n \text{ can be fired at most } m^{n-1} \text{ times.}
\]

Thus, even without resets, a place cannot contain more than \(m^n\) many tokens.

Finally, a marking can be written using \(\log(m^n)\) many bits.  

Note: \(m, n \leq \text{size}(\mathcal{N})\).
Ideas: PSPACE Lower Bound

**Theorem 3:** Coverability in acyclic workflow nets with resets is PSPACE-hard.

**Proof idea:** Reduce from Quantified SATisfiability (QSAT) directly.

Given $\forall y_1 \exists x_1 \forall y_2 \exists x_2 \cdots \forall y_n \exists x_n \phi(y_1, x_1, y_2, x_2, \ldots, y_n, x_n)$.

Diagram:
- Copy variable assignments $y_1$ to places for literals $y_1$.
- Copy variable assignments $x_1$ to places for literals $x_1$.
- Copy variable assignments $y_2$ to places for literals $y_2$.
- Copy variable assignments $x_n$ to places for literals $x_n$.
- Load into clause places $x_1$, $x_2$, $\ldots$, $x_n$.
- Load into clause places $y_1$, $y_2$, $\ldots$, $y_n$.
- Connect to clause $m$.
Example: \((y_1 \lor \overline{x}_1 \lor \overline{y}_2) \land (\overline{x}_1 \lor \overline{y}_2 \lor \overline{x}_2) \land (y_2 \lor \overline{x}_2 \lor \overline{y}_3) \land (\overline{x}_2 \lor \overline{y}_3 \lor \overline{x}_3)\)
**Ideas: Undecidability**

**Theorem 4:** Reachability in acyclic Petri nets with resets is undecidable.

**Lemma:** The reachability problem for acyclic Petri nets with zero tests is reducible in logarithmic space to the reachability problem for acyclic Petri nets with resets.

**Lemma:** The reachability problem in Petri nets with zero tests is reducible in logarithmic space to the reachability problem in acyclic Petri nets with zero tests.

**Proof idea:** Simulate (not necessarily acyclic) transitions using a “transition controller”.

\[ p \rightarrow s \rightarrow r \]

\[ q \rightarrow t \]

\[ s' \rightarrow 0 \]

\[ t' \rightarrow 0 \]
Acyclic Petri and Workflow Nets with Resets

<table>
<thead>
<tr>
<th></th>
<th>Coverability</th>
<th>Reachability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acyclic workflow nets with resets</td>
<td>PSPACE-complete</td>
<td>PSPACE-complete</td>
</tr>
<tr>
<td>Acyclic Petri nets with resets</td>
<td>PSPACE-complete</td>
<td>Undecidable</td>
</tr>
</tbody>
</table>

**Future work:** Study (the decidability of) soundness in acyclic workflow nets with resets.

**Future work:** Is reachability or coverability in acyclic affine Petri or workflow nets decidable?

**Thank You!**

*Presented by Henry Sinclair-Banks, University of Warwick, UK*

*FSTTCS'23 in International Institute of Information Technology, Hyderabad, India*