Decidability of Weak Simulation on One-Counter Nets

Piotr Hofman\textsuperscript{1} Richard Mayr\textsuperscript{2} Patrick Totzke\textsuperscript{2}

University of Warsaw\textsuperscript{1} University of Edinburgh\textsuperscript{2}

September 21, 2013
Weak Steps

For $a \neq \tau \in \text{Act}$ and define

\[
\begin{align*}
\tau &\Rightarrow := \tau \Rightarrow * \\
\rightarrow a &\Rightarrow := \tau \Rightarrow * \rightarrow a \rightarrow \tau \Rightarrow *
\end{align*}
\]
Weak Simulation Games

...are played in rounds between Spoiler and Duplicator. If a player cannot move the other wins. Infinite plays are won by Duplicator.
Weak Simulation Games

... are played in rounds between Spoiler and Duplicator. If a player cannot move the other wins. Infinite plays are won by Duplicator.

**In each round**

<table>
<thead>
<tr>
<th>α</th>
<th>vs.</th>
<th>β</th>
</tr>
</thead>
</table>

1. Spoiler moves from α
2. Duplicator responds from β
3. game continues from α' vs. β'
Weak Simulation Games

...are played in rounds between Spoiler and Duplicator. If a player cannot move the other wins. Infinite plays are won by Duplicator.

In each round

\[
\begin{array}{c}
\alpha \quad \text{vs.} \quad \beta \\
\downarrow \\
\alpha' \\
\end{array}
\]

1. Spoiler moves from \( \alpha \)
2. Duplicator responds from \( \beta \)
3. Game continues from \( \alpha' \) vs. \( \beta' \)
Weak Simulation Games

... are played in rounds between Spoiler and Duplicator. If a player cannot move the other wins. Infinite plays are won by Duplicator.

In each round

1. Spoiler moves from $\alpha$
2. Duplicator responds from $\beta$
3. game continues from $\alpha'$ vs. $\beta'$
Weak Simulation Games

... are played in rounds between Spoiler and Duplicator. If a player cannot move the other wins. Infinite plays are won by Duplicator.

**In each round**

\[
\begin{array}{c}
\alpha \quad \text{vs.} \quad \beta \\
\downarrow a \quad \downarrow a \\
\alpha' \quad \text{vs.} \quad \beta'
\end{array}
\]

1. Spoiler moves from \( \alpha \)
2. Duplicator responds from \( \beta \)
3. Game continues from \( \alpha' \) vs. \( \beta' \)
Weak Simulation Games

...are played in rounds between Spoiler and Duplicator. If a player cannot move the other wins. Infinite plays are won by Duplicator.

<table>
<thead>
<tr>
<th>In each round</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$ vs. $\beta$</td>
</tr>
<tr>
<td>$a$</td>
</tr>
<tr>
<td>$\alpha'$ vs. $\beta'$</td>
</tr>
</tbody>
</table>

1. Spoiler moves from $\alpha$
2. Duplicator responds from $\beta$
3. game continues from $\alpha'$ vs. $\beta'$

Def: Weak Simulation $\preceq$

$\alpha \preceq \beta$ iff Duplicator has a strategy to win from $\alpha$ vs. $\beta$. 
Example

\[ a, 0 \quad \text{\(\tau\), +1} \quad a, 0 \]

\[ \text{A} \quad \text{B} \quad \text{C} \]

\[ a, 0 \not\preceq B \]

\[ A \overset{a, -1}{\rightarrow} C \]
Example

\[ A_0 \not\preceq B_0 \]
Example

\[ a, 0 \]

\[ \tau, +1 \]

\[ a, 0 \]

\[ a, 0 \]

\[ a, -1 \]

- \( A_0 \not\preceq B_0 \)
- \( A_0 \preceq B_0 \)
Example

- $A_0 \not\leq B_0$
- $A_0 \sqsubseteq B_0$
- $B_0 \xrightarrow{a} C_n$ for every $n \in \mathbb{N}$
Example

- $A_0 \nleq B_0$
- $A_0 \nleq B_0$
- $B_0 \xrightarrow{a} Cn$ for every $n \in \mathbb{N}$
We show decidability of the

**OCN Weak Simulation Problem**

Input: A net $\mathcal{N} = (Q, \text{Act}, \delta)$ and configurations $pm, qn$.

Question: $pm \preceq qn$?
Our Contribution

We show decidability of the

**OCN Weak Simulation Problem**

**Input:** A net $\mathcal{N} = (Q, \text{Act}, \delta)$ and configurations $pm, qn$.

**Question:** $pm \preceq qn$?

**Theorem**

*For a given net, the relation $\preceq$ is effectively semilinear.*
Why should you care?

In practice, modelling might use both $\infty$-states and branching:
- network protocols/queues keeping track of their workload
- random guesses

Theoretically, surprising:
- rare positive result for behavioral preorder that is not finitely approximable $\preceq \neq \preceq_\omega$.
- goes against the usual ‘finer is easier’ trend
Some Context – Strong Case

- PDA
- Petri Nets
- OCA
- OCN
- NFA

\[ \subseteq \text{undecidable} \]
\[ \preccurlyeq \text{decidable} \]
\[ \sim \text{PSPACE-c} \]
\[ \subseteq \sim \text{PSPACE-c} \]
\[ \subseteq \text{undec.} \]
\[ \preccurlyeq \text{undec.} \]
\[ \sim \text{undec.} \]
\[ \subseteq \text{undecidable} \]
\[ \preccurlyeq \text{undecidable} \]
\[ \sim \text{undecidable} \]
\[ \subseteq \text{undec.} \]
\[ \preccurlyeq \text{undec.} \]
\[ \sim \text{undec.} \]
\[ \subseteq \text{PSPACE-comp.} \]
\[ \preccurlyeq \text{P-comp.} \]
\[ \sim \text{P-comp.} \]
Some Context – Strong Case

- PDA $\subseteq$ undecidable
  $\succ$ undecidable
  $\sim$ decidable [Sen1998], nonel. [BGKM2013]

- OCA $\subseteq$ undecidable
  $\succ$ undecidable [JMS1999]
  $\sim$ PSPACE-c [Srb2009,BGJ2010]

- OCN $\subseteq$ undecidable [HMT2013]
  $\succ$ decidable [AC1998], PSPACE-c [Srb2009,HLMT2013]
  $\sim$ PSPACE-c [Srb2009,BGJ2010]

- Petri Nets $\subseteq$ undecidable [H1994]
  $\succ$ undec. [H1994]
  $\sim$ undec. [Jan1995]

- NFA $\subseteq$ PSPACE-comp.
  $\succ$ P-comp.
  $\sim$ P-comp.
Some Context – Weak Case

- **PDA**: undecidable
  - **Petri Nets**: undecidable
    - **OCN**: undecidable
      - **OCA**: undecidable
        - **NFA**: undecidable
          - **OCN**: undecidable
            - **NFA**: PSPACE-comp.
              - **P-comp.**

Reference:
- HMT2013
- May2003
- Jan1995
- H1994
Proof Overview

Symbolic infinite branching

Reduce $(OCN \preceq OCN) \rightsquigarrow (OCN \preceq \omega\text{-Net})$
**Proof Overview**

1. **Symbolic infinite branching**
   Reduce $(OCN \preceq OCN) \leadsto (OCN \preceq \omega\text{-Net})$

2. **Approximants for the new game**
   \[ \exists \text{ finite sequence } \preceq^0 \supseteq \preceq^1 \supseteq \preceq^2 \supseteq \cdots \supseteq \preceq^k = \preceq \]
## Proof Overview

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Symbolic infinite branching</td>
</tr>
<tr>
<td></td>
<td>Reduce $(OCN \preceq OCN) \leadsto (OCN \preceq \omega\text{-Net})$</td>
</tr>
<tr>
<td>2</td>
<td>Approximants for the new game</td>
</tr>
<tr>
<td></td>
<td>$\exists$ finite sequence $\preceq^0 \supset \preceq^1 \supset \preceq^2 \supset \cdots \supset \preceq^k = \preceq$</td>
</tr>
<tr>
<td>3</td>
<td>Compute approximants for finite $k$</td>
</tr>
<tr>
<td></td>
<td>Recursively compute $\preceq^k$ by reduction to $(OCN \preceq OCN)$</td>
</tr>
<tr>
<td>Proof Overview</td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td><strong>Symbolic infinite branching</strong></td>
<td></td>
</tr>
<tr>
<td>Reduce ((OCN \preceq OCN) \rightsquigarrow (OCN \preceq \omega\text{-Net}))</td>
<td></td>
</tr>
<tr>
<td><strong>Approximants for the new game</strong></td>
<td></td>
</tr>
<tr>
<td>(\exists) finite sequence (\preceq^0 \supseteq \preceq^1 \supseteq \preceq^2 \supseteq \cdots \supseteq \preceq^k = \preceq)</td>
<td></td>
</tr>
<tr>
<td><strong>Compute approximants for finite (k)</strong></td>
<td></td>
</tr>
<tr>
<td>Recursively compute (\preceq^k) by reduction to ((OCN \preceq OCN))</td>
<td></td>
</tr>
</tbody>
</table>
Approximants for strong simulation (OCN vs. $\omega$-Net)

$\leq^k$
Approximants for strong simulation (OCN vs. $\omega$-Net)

\[ k \]

... holds if Duplicator can guarantee to either

- enforce an infinite game or
- explicitly make use of $\infty$-branching $k$ times.
Example

![Diagram](image_url)

- **Weak Simulation**: Our result
- **Proof Technique**: Summary
Example

\[ A \xrightarrow{a, 0} B \xrightarrow{a, \omega} C \xleftarrow{a, -1} \]

\[ A_0 \not\preceq B_0 \preceq C_0 \]

\[ A_0 \preceq B_0 \preceq C_0 \]

\[ A_0 \not\preceq B_0 \preceq C_0 \]
Example

\[ A_0 \not\trianglerighteq B_0 \]
Example

\[ a, 0 \]

\[ A \]

\[ a, -1 \]

\[ B \]

\[ C \]

- \( A_0 \nless B_0 \)
- \( A_0 \preceq^1 B_0 \)
Example

\[ A_0 \not\leq B_0 \]
\[ A_0 \leq^1 B_0 \]
\[ A_0 \not\leq^2 B_0 \]
Example

\[ a, 0 \quad A \quad a, \omega \quad B \quad a, -1 \quad C \]

- \( A_0 \not\preceq B_0 \)
- \( A_0 \preceq^1 B_0 \)
- \( A_0 \not\preceq^2 B_0 \)
- \( \preceq = \preceq^2 \)
Proof Overview

Symbolic infinite branching
Reduce \((\text{OCN} \preceq \text{OCN}) \rightsquigarrow (\text{OCN} \preceq \omega\text{-Net})\)

Approximants for the new game
\(\exists\) finite sequence \(\preceq^0 \supseteq \preceq^1 \supseteq \preceq^2 \supseteq \cdots \supseteq \preceq^k = \preceq\)

Compute approximants for finite \(k\)
Recursively compute \(\preceq^k\) by reduction to \((\text{OCN} \preceq \text{OCN})\)
Proof Overview

Symbolic infinite branching
Reduce \((\text{OCN} \preceq \text{OCN}) \leadsto (\text{OCN} \preceq \omega\text{-Net})\)

Approximants for the new game
\exists \text{ finite sequence } \preceq^0 \supseteq \preceq^1 \supseteq \preceq^2 \supseteq \cdots \supseteq \preceq^k = \preceq

Compute approximants for finite \(k\)
Recursively compute \(\preceq^k\) by reduction to \((\text{OCN} \preceq \text{OCN})\)
Our Main Contribution

OCN Weak Simulation Problem

Input: A net $\mathcal{N} = (Q, \text{Act}, \delta)$ and configurations $pm, qn$.

Question: $pm \preceq qn$?

Theorem

For a given net, the relation $\preceq$ is effectively semilinear.
Our Main Contribution

OCN Weak Simulation Problem

Input: A net $\mathcal{N} = (Q, \text{Act}, \delta)$ and configurations $pm, qn$.

Question: $pm \preceq qn$?

Theorem

For a given net,

- $\preceq$ is semilinear and can be represented in EXPSPACE.
- The Weak Sim. Problem is PSPACE-complete.
Our Main Contribution

OCN Weak Simulation Problem

Input: A net $\mathcal{N} = (Q, \text{Act}, \delta)$ and configurations $pm, qn$.

Question: $pm \preceq qn$?

Theorem

For a given net,

- $\preceq$ is semilinear and can be represented in EXPSPACE.
- The Weak Sim. Problem is PSPACE-complete.

Questions?