On the Decidability of Normed BPA

Yuxi Fu

Bologna, 22-23 April, 2013
Infinite state systems have been studied in Process Rewriting Systems for some time. The focus has been on the decidability of reachability, equivalence, . . . .

There are very few decidability results in the presence of internal actions. Many problems are open.

One such problem asks if the weak bisimilarity on BPA processes is decidable.
Verification on Infinite State System
From Finite State to Infinite State


It was soon realized that, from the point of view of automatic verification, bisimulation equivalence is the only good equivalence (Groote and Hüttel, 1994).
1. Strong bisimilarity for equivalence between specifications:

\[ Spec_0 \sim Spec_1. \]

2. Branching bisimilarity for correctness of implementation:

\[ Impl \approx Spec \iff Impl \approx Spec. \]

3. Consequently branching bisimilarity for program equivalence:

\[ Pr_0 \approx Pr_1 \iff \exists Spec. Pr_0 \approx Spec \sim Pr_1 \iff Pr_0 \approx Pr_1. \]
A binary relation $\mathcal{R}$ is a **branching bisimulation** if the following is valid whenever $\alpha \mathcal{R} \beta$:

1. If $\beta \mathcal{R}^{-1} \alpha \xrightarrow{\ell} \alpha'$ then one of the following is valid:
   (i) $\ell = \tau$ and $\alpha \mathcal{R} \beta'$.
   (ii) $\beta \Rightarrow \beta'' \mathcal{R}^{-1} \alpha$ for some $\beta''$ such that $\exists \beta'. \beta'' \xrightarrow{\ell} \beta' \mathcal{R}^{-1} \alpha'$.

2. If $\alpha \mathcal{R} \beta \xrightarrow{\ell} \beta'$ then one of the following is valid:
   (i) $\ell = \tau$ and $\alpha' \mathcal{R} \beta$.
   (ii) $\alpha \Rightarrow \alpha'' \mathcal{R} \beta$ for some $\alpha''$ such that $\exists \alpha'. \alpha'' \xrightarrow{\ell} \alpha' \mathcal{R} \beta'$.

3. If $\alpha = \epsilon$ then $\beta \Rightarrow \epsilon$, and if $\beta = \epsilon$ then $\alpha \Rightarrow \epsilon$.

The **branching bisimilarity** $\simeq$ is the largest branching bisimulation.
A process rewriting system $\Gamma$ is a triple $(\mathcal{V}, \mathcal{A}, \Delta)$ where

- $\mathcal{V} = \{X_1, \ldots, X_n\}$ is a finite set of variables,
- $\mathcal{A} = \{a_1, \ldots, a_m\} \cup \{\tau\}$ is a finite set of actions, and
- $\Delta$ is a finite set of transition rules.

A process defined in $\Gamma$ is a member of the set $\mathcal{V}^*$ of finite strings of elements of $\mathcal{V}$. Let $\epsilon$ be the empty string. Let $\alpha, \beta, \gamma, \ldots \in \mathcal{V}^*$.

A transition rule is of the form $\alpha \xrightarrow{\ell} \beta$, where $\ell$ ranges over $\mathcal{A}$. The transitional semantics is closed under composition:

$$\alpha\gamma \xrightarrow{\ell} \beta\gamma$$

for all $\gamma$ whenever $\alpha \xrightarrow{\ell} \beta$. 
Sequential process: \( \alpha \beta \) is understood as \( \alpha . \beta \):
- **BPA**: all rules are of the form \( X \xrightarrow{\ell} \beta \).
- **PDA**: all rules are of the form \( \alpha \xrightarrow{\ell} \beta \).

Parallel process: \( \alpha \beta \) is understood as \( \alpha | \beta \):
- **BPP**: all rules are of the form \( X \xrightarrow{\ell} \beta \).
- **PN**: all rules are of the form \( \alpha \xrightarrow{\ell} \beta \).

Process Algebra: both \( \alpha . \beta \) and \( \alpha | \beta \):
- **PA**: all rules are of the form \( X \xrightarrow{\ell} \beta \).
A process is normed if it can reach $\epsilon$ after a finite number of steps. Normed BPA for example is abbreviated to nBPA.
A process is normed if it can reach $\epsilon$ after a finite number of steps.

Normed BPA for example is abbreviated to nBPA.
A specification of counter, taken from Milner’s 1989 book:

\[
C_0 = \text{zero}.C_0 + \text{inc}.C_1,
\]

\[
C_{i+1} = \text{dec}.C_i + \text{inc}.C_{i+2}, \text{ where } i \geq 0.
\]
The Counter Example

A specification of counter, taken from Milner’s 1989 book:

\[
C_0 = \text{zero}.C_0 + \text{inc}.C_1, \\
C_{i+1} = \text{dec}.C_i + \text{inc}.C_{i+2}, \text{ where } i \geq 0.
\]

Busi, Gabbrielli and Zavattaro’s implementation:

\[
\text{Counter} = \text{zero}.\text{Counter} + \text{inc}.(d)(O \mid d.\text{Counter}), \\
O = \text{dec.}\overline{d} + \text{inc}.(e)(E \mid e.O), \\
E = \text{dec.}\overline{e} + \text{inc}.(d)(O \mid d.E).
\]
A specification of counter, taken from Milner’s 1989 book:

\[
\begin{align*}
C_0 &= \text{zero}.C_0 + \text{inc}.C_1, \\
C_{i+1} &= \text{dec}.C_i + \text{inc}.C_{i+2}, \text{ where } i \geq 0.
\end{align*}
\]

Busi, Gabbrielli and Zavattaro’s implementation:

\[
\begin{align*}
\text{Counter} &= \text{zero}.\text{Counter} + \text{inc}.(d)(O \mid d.\text{Counter}), \\
O &= \text{dec}.\overline{d} + \text{inc}.(e)(E \mid e.O), \\
E &= \text{dec}.\overline{e} + \text{inc}.(d)(O \mid d.E).
\end{align*}
\]

Implementation in BPA:

\[
\begin{align*}
Z & \xrightarrow{\text{inc}} XZ, \\
Z & \xrightarrow{\text{zero}} Z, \\
X & \xrightarrow{\text{inc}} XX, \\
X & \xrightarrow{\text{dec}} \epsilon.
\end{align*}
\]
1. If a problem is **undecidable**, we try to locate it in the arithmetic hierarchy or analytic hierarchy.

2. If a problem is **decidable**, we look for a completeness result.

3. If a problem is in $\mathbf{P}$, we study its algorithmic aspect.
Technique

Decomposition, bisimulation base, tableau, . . .
Defender’s forcing, **computable bound**, . . .
Dickson Lemma, Presburger Arithmetics, . . .
Write $\gamma \to \lambda$ if $\gamma \xrightarrow{\tau} \lambda \simeq \gamma$.

**Lemma.** Suppose $\alpha, \beta$ are nBPA processes. If $\beta \simeq \alpha \xrightarrow{j} \alpha'$, then there is a bisimulation $\beta \rightarrow^* \beta'' \xrightarrow{j} \beta'$ of $\alpha \xrightarrow{j} \alpha'$ with the length of $\beta \rightarrow^* \beta''$ effectively bounded.
Write $\gamma \rightarrow \lambda$ if $\gamma \xrightarrow{\tau} \lambda \simeq \gamma$.

**Lemma.** Suppose $\alpha, \beta$ are nBPA processes. If $\beta \simeq \alpha \xrightarrow{\tau} \alpha'$, then there is a bisimulation $\beta \rightarrow^* \beta'' \xrightarrow{\tau} \beta'$ of $\alpha \xrightarrow{\tau} \alpha'$ with the length of $\beta \rightarrow^* \beta''$ effectively bounded.

**Corollary.** $\not\simeq_{nBPA}$ is semidecidable.
Technique

Decomposition, **bisimulation base**, tableau, . . .
Defender’s forcing, computable bound, . . .
Dickson Lemma, Presburger Arithmetics, . . .
An axiom system $\mathcal{B}$ for nBPA is a finite binary relation on nBPA processes. An axiom $(\alpha, \beta)$ of $\mathcal{B}$ is often written as $\alpha = \beta$.

Write $\mathcal{B} \vdash \alpha = \beta$ if the equality $\alpha = \beta$ can be derived from the axioms of $\mathcal{B}$ by repetitively using equivalence and congruence rules.
A finite axiom system $\mathcal{B}$ for nBPA is a bisimulation base if the following hold for every axiom $(\alpha_0, \beta_0)$ of $\mathcal{B}$: If $\beta_0 \mathcal{B}^{-1} \alpha_0 \rightarrow \alpha_1 \rightarrow \ldots \rightarrow \alpha_n \overset{\ell}{\rightarrow} \alpha'$ then there are $\beta_1, \ldots, \beta_n, \beta'$ such that $\mathcal{B} \vdash \alpha_1 = \beta_1, \ldots, \mathcal{B} \vdash \alpha_n = \beta_n$ and $\mathcal{B} \vdash \alpha' = \beta'$ and the following hold:

(i) For each $i$ with $0 \leq i < n$, either $\beta_i = \beta_{i+1}$, or $\beta_i \rightarrow \beta_{i+1}$, or there are $\beta_{i1}^1, \ldots, \beta_{ik_i}^{k_i}$ st $\beta_i \rightarrow \beta_{i1}^1 \rightarrow \ldots \rightarrow \beta_{ik_i}^{k_i} \rightarrow \beta_{i+1}$ and $\mathcal{B} \vdash \alpha_i = \beta_{i1}^1, \ldots, \mathcal{B} \vdash \alpha_i = \beta_{ik_i}^{k_i}$.

(ii) Either $\ell = \tau$ and $\beta_n = \beta'$, or $\beta_n \overset{\ell}{\rightarrow} \beta'$, or there are $\beta_{n1}^1, \ldots, \beta_{nk_n}^{k_n}$ st $\beta_n \rightarrow \beta_{n1}^1 \rightarrow \ldots \rightarrow \beta_{nk_n}^{k_n} \overset{\ell}{\rightarrow} \beta_{i+1}$ and $\mathcal{B} \vdash \alpha_n = \beta_{n1}^1, \ldots, \mathcal{B} \vdash \alpha_n = \beta_{nk_n}^{k_n}$.

(iii) If $\beta_0 = \epsilon$ then $\alpha_0 \rightarrow \alpha_1 \rightarrow \ldots \rightarrow \alpha_k \rightarrow \epsilon$ for some $\alpha_1, \ldots, \alpha_k$ with $k \geq 0$ such that $\mathcal{A} \vdash \alpha_1 = \epsilon, \ldots, \mathcal{A} \vdash \alpha_k = \epsilon$. 
Lemma. If $\mathcal{B}$ is a bisimulation base, then $\mathcal{B} \subseteq \sim$. 
Technique

Decomposition, bisimulation base, tableau, . . .
Defender’s forcing, computable bound, . . .
Dickson Lemma, Presburger Arithmetics, . . .
A tableau system is a way of constructing a bisimulation base.
A tableau system is way of constructing bisimulation base.

**Lemma.** Given nBPA processes $\alpha, \beta$ there is an effective procedure, by constructing tableau systems, to generate a bisimulation base that contains $(\alpha, \beta)$ whenever $\alpha \simeq \beta$.

**Corollary.** $\simeq_{nBPA}$ is semidecidable.
Theorem. $\simeq_{n\text{BPA}}$ is decidable.
A Bird’s View of Existing Results
<table>
<thead>
<tr>
<th></th>
<th>PN</th>
<th>nPN</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sim )</td>
<td>( \Pi^0_1 )-complete \cite{JS08}</td>
<td>Undecidable \cite{Jan95}</td>
</tr>
<tr>
<td>( \sim )</td>
<td>Undecidable \cite{Jan95}</td>
<td>Undecidable \cite{Jan95}</td>
</tr>
<tr>
<td>( \sim )</td>
<td>Undecidable \cite{Jan95}</td>
<td>Undecidable \cite{Jan95}</td>
</tr>
<tr>
<td>( \sim )</td>
<td>( \Sigma^1_1 )-complete \cite{JS08}</td>
<td>Undecidable \cite{Jan95}</td>
</tr>
</tbody>
</table>

Where is \( \sim_{PN} \)?
<table>
<thead>
<tr>
<th></th>
<th>BPP</th>
<th>nBPP</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\equiv$</td>
<td>Decidable [CHM93]</td>
<td>Decidable [CHM93]</td>
</tr>
<tr>
<td></td>
<td>PSPACE [Jan03]</td>
<td>P [HJM96b]</td>
</tr>
<tr>
<td></td>
<td>PSPACE-hard [Srb02a]</td>
<td>P-hard [BGS92]</td>
</tr>
<tr>
<td>$\mapsto$</td>
<td>$\equiv$</td>
<td>$\equiv$</td>
</tr>
<tr>
<td></td>
<td>PSPACE-hard [Srb02a]</td>
<td>Decidable [CHL11]</td>
</tr>
<tr>
<td>$\mapsto$</td>
<td>$\equiv$</td>
<td>$\equiv$</td>
</tr>
<tr>
<td></td>
<td>PSPACE-hard [Srb03]</td>
<td>PSPACE-hard [Srb03]</td>
</tr>
</tbody>
</table>

Is $\sim_{BPP}$ decidable?
### PDA: between the Decidable and the Undecidable

<table>
<thead>
<tr>
<th></th>
<th>PDA</th>
<th>nPDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>?</td>
<td>$\Sigma_1$-complete [JS08] Undecidable [Srb02c]</td>
<td>$\Sigma_1$-complete [JS08] Undecidable [Srb02c]</td>
</tr>
</tbody>
</table>

Is $\simeq_{nPDA}$ decidable?
### BPA: Exploiting Transition Tree

<table>
<thead>
<tr>
<th></th>
<th>BPA</th>
<th>nBPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\not\approx$?</td>
<td>? EXPTIME-hard [May03]</td>
<td>Decidable</td>
</tr>
<tr>
<td>$\not\approx$?</td>
<td>? EXPTIME-hard [May03]</td>
<td>?</td>
</tr>
</tbody>
</table>

**Is $\approx_{\text{BPA}}$ decidable?**
Remark

For parallel processes (PN, BPP) with silent actions, the only decidability result is due to Czerwiński, Hofman and Lasota (2011).

For sequential processes (PDA, BPA) with silent actions, a decidability result is given in this talk.
Regularity Problem
Regularity problem asks if a given process (seen as an implementation) is equivalent to a finite state (seen as a specification).
<table>
<thead>
<tr>
<th></th>
<th>PN</th>
<th>nPN</th>
</tr>
</thead>
<tbody>
<tr>
<td>⊆</td>
<td>Decidable [JE96]</td>
<td>EXPSPACE-hard [Rac78]</td>
</tr>
<tr>
<td></td>
<td>PSPACE-hard [Srb02a]</td>
<td>EXPSPACE-hard [Lip76]</td>
</tr>
<tr>
<td>⊇</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>⊇</td>
<td>Undecidable [JE96]</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>EXPSPACE-hard [Lip76]</td>
<td>EXPSPACE-hard [Lip76]</td>
</tr>
</tbody>
</table>
### BPP

<table>
<thead>
<tr>
<th></th>
<th>BPP</th>
<th>nBPP</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>Decidable [JE96]</td>
<td>NL [Kuč96]</td>
</tr>
<tr>
<td></td>
<td>PSPACE-hard [Srb02a]</td>
<td>NL-hard [Srb02a]</td>
</tr>
<tr>
<td>?</td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>?</td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>?</td>
<td></td>
<td>?</td>
</tr>
</tbody>
</table>

PSPACE-hard [Srb03]
<table>
<thead>
<tr>
<th>PDA</th>
<th>nPDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXPTIME-hard</td>
<td>P [EHRS00]</td>
</tr>
<tr>
<td>[<em>,</em>]</td>
<td>NL-hard [Srb02b]</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>EXPTIME-hard</td>
<td></td>
</tr>
<tr>
<td>[<em>,</em>]</td>
<td></td>
</tr>
</tbody>
</table>

[*,*] = [KM02, Srb02b]
<table>
<thead>
<tr>
<th>BPA</th>
<th>nBPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decidable [BCS95, BCS96] PSPACE-hard [Srb02a]</td>
<td>NL-complete [Srb02a] [Kuč96]</td>
</tr>
<tr>
<td>EXPTIME-hard [May03]</td>
<td>Decidable</td>
</tr>
<tr>
<td>EXPTIME-hard [May03]</td>
<td>NP-hard [Srb03, Stř98]</td>
</tr>
</tbody>
</table>
Except in the case of PDA, all the regularity problems of strong bisimilarity in the setting of PRS is known to be decidable.

In the setting of PRS, the only decidable regularity problem is about the branching bisimilarity for normed BPA. All the other regularity problems are unknown.
Theorem. The regularity of $\sim_{n\text{BPA}}$ is decidable.
Thanks


S. Christensen, Y. Hirshfeld, and F. Moller.
Bisimulation equivalence is decidable for basic parallel processes.


S. Christensen, H. Hüttel, and C. Stirling.
Bisimulation equivalence is decidable for all context-free processes.

Efficient algorithms for model checking pushdown systems.

Y. Hirshfeld, M. Jerrum, and F. Moller.
A polynomial algorithm for deciding bisimilarity of normed context free processes.
Y. Hirshfeld, M. Jerrum, and F. Moller.

H. Hüttel and C. Stirling.

P. Jančar.

P. Jančar.
Strong bisimilarity on basic parallel processes in pspace-complete.
P. Jančar and J. Esparza. 
Deciding finiteness of petri nets up to bisimulation. 

P. Jančar and J. Srba. 
Undecidability of bisimilarity by defender’s forcing. 

S. Kiefer. 
Bpa bisimilarity is exptime-hard. 
2012.

A. Kučera and R. Mayr. 
On the complexity of semantic equivalences for pushdown automata and bpa. 
A. Kučera.
Regularity is decidable for normed bpa and normed bpp processes in polynomial time.

R.J. Lipton.
The reachability problem requires exponential space.
Technical report, Department of Computer Science, Yale University, 1976.

R. Mayr.
Weak bisimilarity and regularity of bpa is exptime-hard.
In *EXPRESS’03*, 2003.

C. Rackoff.
The covering and boundedness problems for vector addition systems.

G. Sénizergues.
Decidability of bisimulation equivalence for equational graphs of finite out-degree.

J. Srba.
Strong bisimilarity and regularity of basic parallel processes is pspace-hard.

J. Srba.
Strong bisimilarity and regularity of basic process algebra is PSPACE-hard.
In ICALP’02, LNCS 2380, pages 716–727, 2002.

J. Srba.
Undecidability of weak bisimilarity for pushdown processes.

J. Srba.
Complexity of weak bisimilarity and regularity for BPA and BPP.

C. Stirling.
Decidability of bisimulation equivalence for normed pushdown processes.

J. Stříbrná.
Hardness results for weak bisimilarity of simple process algebras.