# 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

Milner's work (1984,1989).

Baeten, Bergstra and Klop's work (1987, 1993).

It was soon realized that, from the point of view of automatic verification, bisimulation equivalence is the only good equivalence (Groote and Hüttel, 1994).

# Verification as Equivalence Checking

1. Strong bisimilarity for equivalence between specifications:

$$Spec_0 \sim Spec_1.$$

2. Branching bisimilarity for correctness of implementation:

$$Impl \approx Spec \text{ iff } Impl \simeq Spec.$$

3. Consequently branching bisimilarity for program equivalence:

$$Pr_0 \approx Pr_1 \text{ iff } \exists Spec.Pr_0 \simeq Spec \simeq Pr_1 \text{ iff } Pr_0 \simeq Pr_1.$$

# Branching Bisimilarity

A binary relation $\mathcal{R}$ is a branching bisimulation if the following is valid whenever $\alpha\mathcal{R}\beta$:

1. If $\beta\mathcal{R}^{-1}\alpha \stackrel{\ell}{\longrightarrow} \alpha'$ then one of the following is valid:
   (i) $\ell = \tau$ and $\alpha\mathcal{R}\beta'$.
   (ii) $\beta \Longrightarrow \beta''\mathcal{R}^{-1}\alpha$ for some $\beta''$ such that $\exists\beta'.\beta'' \stackrel{\ell}{\longrightarrow} \beta'\mathcal{R}^{-1}\alpha'$.
2. If $\alpha\mathcal{R}\beta \stackrel{\ell}{\longrightarrow} \beta'$ then one of the following is valid:
   (i) $\ell = \tau$ and $\alpha'\mathcal{R}\beta$.
   (ii) $\alpha \Longrightarrow \alpha''\mathcal{R}\beta$ for some $\alpha''$ such that $\exists\alpha'.\alpha'' \stackrel{\ell}{\longrightarrow} \alpha'\mathcal{R}\beta'$.
3. If $\alpha = \epsilon$ then $\beta \Longrightarrow \epsilon$, and if $\beta = \epsilon$ then $\alpha \Longrightarrow \epsilon$.

The branching bisimilarity $\simeq$ is the largest branching bisimulation.

# Process Rewriting System, Mayr 2000

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 element 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:

$$\alpha\gamma \xrightarrow{\ell} \beta\gamma \text{ for all } \gamma \text{ whenever } \alpha \xrightarrow{\ell} \beta.$$

# Process Rewriting System

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

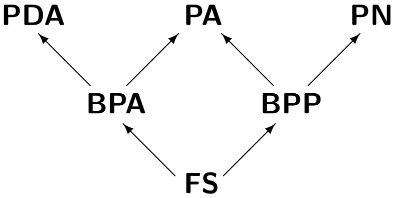
Parallel process: $\alpha\beta$ is understood as $\alpha \mid \beta$:

- **BPP**: all rules are of the form $X \xrightarrow{\ell} \beta$.
- **PN**: all rules are of the form $\alpha \xrightarrow{\ell} \beta$.

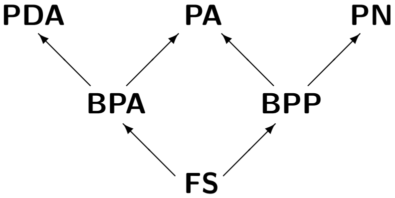
Process Algebra: both $\alpha.\beta$ and $\alpha \mid \beta$:

- **PA**: all rules are of the form $X \xrightarrow{\ell} \beta$.

# Process Rewriting System



# Process Rewriting System




A process is normed if it can reach $\epsilon$ after a finite number of steps.

Normed **BPA** for example is abbreviated to **nBPA**.

# The Counter Example

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

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

# The Counter Example

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

$$\begin{aligned} C_0 &= \mathit{zero}.C_0 + \mathit{inc}.C_1, \\ C_{i+1} &= \mathit{dec}.C_i + \mathit{inc}.C_{i+2}, \text{ where } i \geq 0. \end{aligned}$$

Busi, Gabbrielli and Zavattaro's implementation:

$$\begin{aligned} \mathit{Counter} &= \mathit{zero}.\mathit{Counter} + \mathit{inc}.(d)(O \mid d.\mathit{Counter}), \\ O &= \mathit{dec}.\overline{d} + \mathit{inc}.(e)(E \mid e.O), \\ E &= \mathit{dec}.\overline{e} + \mathit{inc}.(d)(O \mid d.E). \end{aligned}$$

# The Counter Example

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

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

Busi, Gabbrielli and Zavattaro's implementation:

$$\begin{aligned} Counter &= zero.Counter + inc.(d)(O \mid d.Counter), \\ O &= dec.\overline{d} + inc.(e)(E \mid e.O), \\ E &= dec.\overline{e} + inc.(d)(O \mid d.E). \end{aligned}$$

Implementation in BPA:

$$\begin{aligned} Z &\xrightarrow{inc} XZ, \\ Z &\xrightarrow{zero} Z, \\ X &\xrightarrow{inc} XX, \\ X &\xrightarrow{dec} \epsilon. \end{aligned}$$

# Line of Investigation

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 **P**, we study its algorithmic aspect.

# Technique

Decomposition, bisimulation base, tableau, . . .

Defender's forcing, computable bound, . . .

Dickson Lemma, Presburger Arithmetics, . . .

# Computable Bound

Write $\gamma \rightarrow \lambda$ if $\gamma \xrightarrow{\tau} \lambda \simeq \gamma$.

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

# Computable Bound

Write $\gamma \rightarrow \lambda$ if $\gamma \stackrel{\tau}{\longrightarrow} \lambda \simeq \gamma$.

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

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

# Technique

Decomposition, bisimulation base, tableau, ...

Defender's forcing, computable bound, ...

Dickson Lemma, Presburger Arithmetics, ...

# Bisimulation Base

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.

# Bisimulation Base

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 $\beta_0 \mathcal{B}^{-1} \alpha_0 \longrightarrow \alpha_1 \longrightarrow \ldots \longrightarrow \alpha_n \stackrel{\ell}{\longrightarrow} \alpha'$ then there are $\beta_1, \ldots, \beta_n, \beta'$ such that $\mathcal{B} \vdash \alpha_1 = \beta_1$, ..., $\mathcal{B} \vdash \alpha_n = \beta_n$ 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 $\beta_i \longrightarrow \beta_{i+1}$, or there are $\beta_i^1, \ldots, \beta_i^{k_i}$ st $\beta_i \longrightarrow \beta_i^1 \longrightarrow \ldots \longrightarrow \beta_i^{k_i} \longrightarrow \beta_{i+1}$ and $\mathcal{B} \vdash \alpha_i = \beta_i^1$, ..., $\mathcal{B} \vdash \alpha_i = \beta_i^{k_i}$.

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

(iii) If $\beta_0 = \epsilon$ then $\alpha_0 \longrightarrow \alpha_1 \longrightarrow \ldots \longrightarrow \alpha_k \longrightarrow \epsilon$ for some $\alpha_1, \ldots, \alpha_k$ with $k \geq 0$ such that $\mathcal{A} \vdash \alpha_1 = \epsilon$, ..., $\mathcal{A} \vdash \alpha_k = \epsilon$.

# Bisimulation Base

**Lemma**. If $\mathcal{B}$ is a bisimulation base, then $\mathcal{B} \subseteq \simeq$.

# Technique

Decomposition, bisimulation base, tableau, ...

Defender's forcing, computable bound, ...

Dickson Lemma, Presburger Arithmetics, ...

# Tableau

A tableau system is way of constructing bisimulation base.

# Tableau

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**. $\simeq_{\mathbf{nBPA}}$ is semidecidable.

# Checking Equality for **nBPA**

**Theorem**. $\simeq_{\mathbf{nBPA}}$ is decidable.

# A Bird's View of Existing Results

# PN: Beyond Decidability

| | **PN** | **nPN** |
|---|---|---|
| $\sim$ | $\Pi^0_1$-complete [JS08]<br>Undecidable [Jan95] | Undecidable [Jan95] |
| $\simeq$ | **?**<br>Undecidable [Jan95] | Undecidable [Jan95] |
| $\approx$ | $\Sigma^1_1$-complete [JS08]<br>Undecidable [Jan95] | Undecidable [Jan95] |

Where is $\simeq_{\mathbf{PN}}$?

# BPP: Dickson Lemma, Redei Lemma

| | **BPP** | **nBPP** |
|---|---|---|
| $\sim$ | Decidable [CHM93]<br>PSPACE [Jan03]<br>PSPACE-hard [Srb02a] | Decidable [CHM93]<br>P [HJM96b]<br>P-hard [BGS92] |
| $\simeq$ | **?**<br>PSPACE-hard [Srb02a] | Decidable [CHL11] |
| $\approx$ | ?<br>PSPACE-hard [Srb03] | ?<br>PSPACE-hard [Srb03] |

Is $\simeq_{\mathbf{BPP}}$ decidable?

# PDA: between the Decidable and the Undecidable

| | **PDA** | **nPDA** |
|---|---|---|
| $\sim$ | Decidable [Sén98]<br>EXPTIME-hard [KM02] | Decidable [Sti98]<br>EXPTIME-hard [KM02] |
| $\simeq$ | ? | **?** |
| $\approx$ | $\Sigma_1^1$-complete [JS08]<br>Undecidable [Srb02c] | $\Sigma_1^1$-complete [JS08]<br>Undecidable [Srb02c] |

Is $\simeq_{\mathbf{nPDA}}$ decidable?

# BPA: Exploiting Transition Tree

| | BPA | nBPA |
|---|---|---|
| $\sim$ | Decidable [CHS92]<br>2-EXPTIME [BCS95]<br>EXPTIME-hard [Kie12]<br>PSPACE-hard [Srb02b] | Decidable [HS91]<br>P-complete [BGS92][HJM96a] |
| $\simeq$ | **?**<br>EXPTIME-hard [May03] | Decidable<br>? |
| $\approx$ | ?<br>EXPTIME-hard [May03]<br>PSPACE-hard [Stř98] | ?<br>EXPTIME-hard [May03]<br>PSPACE-hard [Stř98] |

Is $\simeq_{\textbf{BPA}}$ 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).

# PN

| | **PN** | **nPN** |
|---|---|---|
| $\sim$ | Decidable [JE96]<br>PSPACE-hard [Srb02a] | EXPSAPCE [Rac78]<br>EXPSPACE-hard [Lip76] |
| $\simeq$ | **?** | ? |
| $\approx$ | Undecidable [JE96]<br>EXPSPACE-hard [Lip76] | ?<br>EXPSPACE-hard [Lip76] |

# BPP

| | **BPP** | **nBPP** |
|---|---|---|
| $\sim$ | Decidable [JE96]<br>PSPACE-hard [Srb02a] | NL [Kuč96]<br>NL-hard [Srb02a] |
| $\simeq$ | ? | **?** |
| $\approx$ | ?<br>PSPACE-hard [Srb03] | ?<br>PSPACE-hard [Srb03] |

# PDA

| | **PDA** | **nPDA** |
|---|---|---|
| $\sim$ | **?**<br>EXPTIME-hard [*,*] | P [EHRS00]<br>NL-hard [Srb02b] |
| $\simeq$ | ? | ? |
| $\approx$ | ?<br>EXPTIME-hard [*,*] | ?<br>EXPTIME-hard [*,*] |

[*,*]= [KM02, Srb02b]

# BPA

| | **BPA** | **nBPA** |
|---|---|---|
| $\sim$ | Decidable [BCS95, BCS96]<br>PSPACE-hard [Srb02a] | NL-complete [Srb02a][Kuč96] |
| $\simeq$ | ?<br>EXPTIME-hard [May03] | Decidable |
| $\approx$ | ?<br>EXPTIME-hard [May03] | ?<br>NP-hard [Srb03, Stř98] |

# Remark

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.

# Checking Regularity for **nBPA**

**Theorem**. The regularity of $\simeq_{\mathbf{nBPA}}$ is decidable.

# Thanks

O. Burkart, D. Caucal, and B. Steffen.
An elementary bisimulation decision procedure for arbitrary context freeprocesses.
*MFCS'95*, pages 423–433, 1995.

O. Burkart, D. Caucal, and B. Steffen.
Bisimulation collapse and the process taxonomy.
*CONCUR 1996*, pages 247–262, 1996.

J. Balcazar, J. Gabarro, and M. Santha.
Deciding bisimilarity is p-complete.
*Formal Aspects of Computing*, 4:638–648, 1992.

W. Czerwiński, P. Hofman, and S. Lasota.
Decidability of branching bisimulation on normed commutative context-free processes.
In *CONCUR 2011*, pages 528–542. Lecture Notes in Computer Science 6901, Springer, 2011.

S. Christensen, Y. Hirshfeld, and F. Moller.

Bisimulation equivalence is decidable for basic parallel processes.
In *Prodeedings of the 4th International Conference on Concurrency Theory (CONCUR'93)*, pages 143–157. Springer, 1993.

S. Christensen, H. Hüttel, and C. Stirling.
Bisimulation equivalence is decidable for all context-free processes.
pages 138–147. Lecture Notes in Computer Science 630, Springer, 1992.

J. Esparza, D. Hansel, P. Rossmanith, and S. Schwoon.
Efficient algorithms for model checking pushdown systems.
In *Computer Aided Verification*, pages 232–247. Springer, 2000.

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

*Theoretical Computer Science*, 158(1-2):143–159, 1996.

Y. Hirshfeld, M. Jerrum, and F. Moller.
A polynomial-time algorithm for deciding bisimulation equivalence of normed basic parallel processes.
*Mathematical Structures in Computer Science*, 6(03):251–259, 1996.

H. Hüttel and C. Stirling.
Actions speak louder than words: Proving bisimilarity for context-free processes.
In *LICS 1991*, pages 376–386, 1991.

P. Jančar.
Undecidability of bisimilarity for petri nets and some related problems.
*Theoretical Computer Science*, 148(2):281–301, 1995.

P. Jancar.
Strong bisimilarity on basic parallel processes in pspace-complete.

In *Logic in Computer Science, 2003. Proceedings. 18th Annual IEEE Symposium on*, pages 218–227. IEEE, 2003.

P. Jančar and J. Esparza.
Deciding finiteness of petri nets up to bisimulation.
*Automata, Languages and Programming*, pages 478–489, 1996.

P. Jančar and J. Srba.
Undecidability of bisimilarity by defender's forcing.
*Journal of ACM*, 55(1), 2008.

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

A. Kučera and R. Mayr.
On the complexity of semantic equivalences for pushdown automata and bpa.
*Mathematical Foundations of Computer Science 2002*, pages 433–445, 2002.

A. Kučera.
Regularity is decidable for normed bpa and normed bpp processes in polynomial time.
In *SOFSEM'96*, pages 377–384. Springer, 1996.

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.
*Theoretical Computer Science*, 6(2):223–231, 1978.

G. Sénizergues.

Decidability of bisimulation equivalence for equational graphs of finite out-degree.
In *Foundations of Computer Science, 1998. Proceedings. 39th Annual Symposium on*, pages 120–129. IEEE, 1998.

J. Srba.
Strong bisimilarity and regularity of basic parallel processes is pspace-hard.
In *STACS 2002*, pages 733–733. Springer, 2002.

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.
In *ONCUR 2002*, volume 2421 of *LNCS*, pages 579–593. Springer-Verlag, 2002.

J. Srba.

Complexity of weak bisimilarity and regularity for BPA and BPP.
*Mathematical Structures in Computer Science*, 13:567–587, 2003.

C. Stirling.
Decidability of bisimulation equivalence for normed pushdown processes.
*Theoretical Computer Science*, 195(2):113–131, 1998.

J. Stříbrná.
Hardness results for weak bisimilarity of simple process algebras.
*MFCS'98. ENTCS*, 18:179–190, 1998.