On Distributed Bisimilarity over Basic Parallel Processes

Petr Jančar   Zdeněk Sawa

Department of Computer Science, FEI
Technical University of Ostrava
17. listopadu 15, Ostrava-Poruba 708 33
Czech Republic

26 November 2005

Presented Result

- The result was presented at AVIS'05 workshop (Automated Verification of Infinite-State Systems, Edinburgh, 2nd-3rd April 2005, a part of ETAPS 2005),
- It will be published in Electronic Notes in Theoretical Computer Science.
- A plan to prepare an overview article on non-interleaving equivalences on Basic Parallel Processes with S. Lasota and S. Fröschle.

Introduction

- **Equivalence checking** – deciding for a pair of systems whether they are equivalent with respect to some notion of equivalence
- **Basic Parallel Processes (BPP)** – very simple class of infinite state systems

Basic Parallel Processes (BPP)

- $Act = \{a, b, c, \ldots\}$ – actions
- $Var = \{X, Y, Z, \ldots\}$ – variables

BPP expressions

$$P \ ::= \ \mathbf{0} \ | \ X \ | \ a.P \ | \ P_1 + P_2 \ | \ P_1 \parallel P_2$$

BPP process definition

$$\Delta = \{X_i \stackrel{\mathrm{def}}{=} P_i \mid 1 \leq i \leq n\}$$

Basic Parallel Processes (BPP)

- $Act = \{a, b, c, \ldots\}$ – actions
- $Var = \{X, Y, Z, \ldots\}$ – variables

BPP expressions

$$P \ ::= \ \mathbf{0} \ | \ X \ | \ a.P \ | \ P_1 + P_2 \ | \ P_1 \parallel P_2$$

BPP process definition

$$\Delta = \{X_i \stackrel{\text{def}}{=} P_i \mid 1 \leq i \leq n\}$$

Example:

$$X \stackrel{\text{def}}{=} a.(X + c.Z) + \mathbf{0}$$
$$Y \stackrel{\text{def}}{=} a.(X \parallel Y) + b.Z$$
$$Z \stackrel{\text{def}}{=} c.(X \parallel Z)$$

Distributed bisimilarity

- one of **non-interleaving** equivalences ('true concurrency' equivalences)

Distributed bisimilarity

- one of **non-interleaving** equivalences ('true concurrency' equivalences)
- on BPP coincides with many other non-interleaving equivalences: (location equivalence, causal equivalence, history preserving bisimilarity, performance equivalence, . . .)

Distributed bisimilarity

- one of **non-interleaving** equivalences ('true concurrency' equivalences)
- on BPP coincides with many other non-interleaving equivalences: (location equivalence, causal equivalence, history preserving bisimilarity, performance equivalence, . . .)
- distinguishes processes such as

$$a.\mathbf{0} \parallel a.\mathbf{0} \qquad \text{and} \qquad a.a.\mathbf{0}$$

Distributed bisimilarity (cont.)

Distributed bisimilarity takes the spatial distribution of (sub)processes into account.

Each transition from a process P goes to a **pair** of processes:

- **local** derivative (P')
- **concurrent** derivative (P'')

Transition

$$P \xrightarrow{a} [P', P'']$$

Distributed bisimilarity (cont.)

Structural operational semantics rules

$$\frac{}{a.P \xrightarrow{a} [P, \mathbf{0}]} \qquad \frac{P_j \xrightarrow{a} [P', P''] \text{ for some } j \in I}{\sum_{i \in I} P_i \xrightarrow{a} [P', P'']}$$

$$\frac{P \xrightarrow{a} [P', P'']}{P \parallel Q \xrightarrow{a} [P', P'' \parallel Q]} \qquad \frac{Q \xrightarrow{a} [Q', Q'']}{P \parallel Q \xrightarrow{a} [Q', P \parallel Q'']}$$

$$\frac{P \xrightarrow{a} [P', P'']}{X \xrightarrow{a} [P', P'']} ((X \stackrel{\text{def}}{=} P) \in \Delta)$$

Distributed bisimilarity (cont.)

Definition

A relation $\mathcal{R}$ is a **distributed bisimulation** iff for each $(P, Q) \in \mathcal{R}$ and each $a \in Act$:

- if $P \xrightarrow{a} [P', P'']$ then $Q \xrightarrow{a} [Q', Q'']$ for some Q', Q'' such that $(P', Q') \in \mathcal{R}$ and $(P'', Q'') \in \mathcal{R}$
- if $Q \xrightarrow{a} [Q', Q'']$ then $P \xrightarrow{a} [P', P'']$ for some P', P'' such that $(P', Q') \in \mathcal{R}$ and $(P'', Q'') \in \mathcal{R}$

Distributed bisimilarity (cont.)

Definition

A relation $\mathcal{R}$ is a **distributed bisimulation** iff for each $(P, Q) \in \mathcal{R}$ and each $a \in Act$:

- if $P \xrightarrow{a} [P', P'']$ then $Q \xrightarrow{a} [Q', Q'']$ for some Q', Q'' such that $(P', Q') \in \mathcal{R}$ and $(P'', Q'') \in \mathcal{R}$
- if $Q \xrightarrow{a} [Q', Q'']$ then $P \xrightarrow{a} [P', P'']$ for some P', P'' such that $(P', Q') \in \mathcal{R}$ and $(P'', Q'') \in \mathcal{R}$

Definition

Processes P and Q are **distributed bisimilar**, denoted $P \sim Q$, iff there is a distributed bisimulation $\mathcal{R}$ such that $(P, Q) \in \mathcal{R}$.

Distributed bisimilarity – example

$a.\mathbf{0} \parallel a.\mathbf{0}$ and $a.a.\mathbf{0}$ are not distributed bisimilar, because

$$a.\mathbf{0} \parallel a.\mathbf{0} \xrightarrow{a} [\mathbf{0}, a.\mathbf{0}]$$

but

$$a.a.\mathbf{0} \xrightarrow{a} [a.\mathbf{0}, \mathbf{0}]$$

Main problem

Problem BPP-DBISIM

Instance: A BPP process definition Δ and two variables $X, Y \in Var(\Delta)$

Question: Is $X \sim Y$?

Remark: Question whether $P_1 \sim P_2$ where P_1, P_2 are BPP expressions can be easily transformed to BPP-DBISIM.

Basic processes

Operators $+$ and $\|$ are associative and commutative with respect to $\sim$.

Definition

Basic process is a process of the form

$$X_1 \parallel X_2 \parallel \cdots \parallel X_n$$

where $n \geq 0$ and each $X_i \in Var$

Remark: $\mathbf{0}$ is a basic process where $n = 0$.

Basic processes (cont.)

Due to associativity and commutativity of $\|$, **basic processes** can be viewed as **multisets** of variables and represented more succinctly.

For example:

$$(X \parallel X \parallel X \parallel X \parallel X \parallel X \parallel X \parallel Y \parallel Y \parallel Z \parallel Z \parallel Z)$$

can be represented as:

$$X^7 Y^2 Z^3$$

where numbers as represented in binary

Left merge

Left merge is an auxiliary new operator:

$$P_1 \mathbin{\|\!\!\lfloor} P_2$$

is similar to $P_1 \parallel P_2$ but P_1 must perform an action first.

$$\frac{P \stackrel{a}{\longrightarrow} [P', P'']}{P \mathbin{\|\!\!\lfloor} Q \stackrel{a}{\longrightarrow} [P', P'' \parallel Q]}$$

Normal form

Every BPP process can be transformed into equivalent **normal form** where all equations are of the form

$$X \stackrel{\text{def}}{=} \sum_{i \in I} ((a_i.P_i) \parallel Q_i)$$

where all P_i, Q_i are basic processes.

Remark: Variables in Q_i are syntactically unguarded.

Normal form

Every BPP process can be transformed into equivalent **normal form** where all equations are of the form

$$X \stackrel{\text{def}}{=} \sum_{i \in I} ((a_i.P_i) \,\|\!\!\lfloor\, Q_i)$$

where all P_i, Q_i are basic processes.

Remark: Variables in Q_i are syntactically unguarded.

$Y \prec_1 X$ holds iff $X \stackrel{\text{def}}{=} \sum_{i \in I} ((a.P_i) \,\|\!\!\lfloor\, Q_i)$ and Y occurs in some Q_i

$\prec$ is the transitive closure of $\prec_1$

$\prec$ is irreflexive

Known results

It was shown in

> *Christensen, S.: Decidability and Decomposition in Process Algebras, Ph.D. thesis, The University of Edinburgh, 1993.*

that:

- BPP-DBISIM is decidable.
- Every BPP process definition can be transformed to (a variant of) normal form.

Known results (cont.)

It was shown in

Lasota, S.: A polynomial-time algorithm for deciding true concurrency equivalences of Basic Parallel Processes, MFCS, 2003.

that:

- BPP-DBISIM can be solved in polynomial time.
- BPP process definition can be transformed to normal form in polynomial time.

Remarks on Lasota's algorithm

Lasota's algorithm consists of two steps:

- transformation of BPP process definition to normal form
- algorithm deciding distributed bisimilarity on processes in normal form

Remark: No explicit degrees of the polynomials bounding time complexity of these steps were provided.

Remarks on Lasota's algorithm (cont.)

Our analysis of Lasota's transformation to normal form shows that:

- it can be done in $O(n^3)$
- the size of the constructed process definition can rise to $\Theta(n^3)$

Remarks on Lasota's algorithm (cont.)

Our analysis of Lasota's transformation to normal form shows that:

- it can be done in $O(n^3)$
- the size of the constructed process definition can rise to $\Theta(n^3)$

Lasota's algorithm deciding distributed bisimilarity on processes in normal form:

- repeatedly calls an algorithm for deciding **strong** bisimilarity on **normed** BPP as a subroutine
- has time complexity that seems to be in $O(n^5)$, where n is size its instance (BPP in normal form)

Our contribution

We present an algorithm for deciding distributed bisimilarity on processes in normal form that:

- is conceptually simpler
- is self-contained (does not use any other algorithm as a subroutine)
- allows to bound its time complexity by $O(n^3)$

Algorithm

The algorithm:

- is inspired by ideas introduced by P. Jančar in the proof of PSPACE-completeness of deciding strong bisimilarity on BPP
- constructs a sequence of approximations of $\sim$ on the set of basic processes (elements of $Var^{\oplus}$) from above
- produces as a by-product a semilinear representation of $\sim$ on the set of basic processes

Thank you