Deciding Branching Bisimilarity between BPA and Finite-State Systems

Hongfei Fu

BASICS Laboratory
Department of Computer Science
Shanghai Jiao Tong University

APLAS 2009 Paper
Background
Formal Verification

- formal specification: finite-state system
- real implementation: infinite-state system
Bisimulation Semantics:

- strong bisimulation (Park)
- weak bisimulation (Milner)
- branching bisimulation (van Glabbeek and Weijland)
Polynomial time algorithms deciding branching bisimilarity between:

- BPA (Basic Process Algebra) and finite-state systems (FS)
- Normed BPP (Basic Parallel Processes) and finite-state systems
Definitions
Let $V$ be a finite alphabet of symbols. Symbols of $V$ are ranged over by $X, Y, Z$ . . . .

The set of words over $V$ is denoted $V^*$. 
We presume a set of actions $Act_{\tau}$.

We always use

- $\Gamma$ to refer to a FS
- $State(\Gamma)$ to refer to the state set of $\Gamma$
- $f, g, h \ldots$ to range over $State(\Gamma)$
A BPA system is a tuple \((V, \Delta)\) where

- \(V\) is a finite alphabet of symbols.
- \(\Delta\) is a finite set of *rules* for which each rule has the form \(X \xrightarrow{a} \alpha\)
  where \(X \in V\), \(\alpha \in V^*\) and \(a \in \text{Act}\).
A BPA system $(V, \Delta)$ defines an LTS where
- states are elements of $V^*$.
- for $\alpha, \beta \in V^*$, $\alpha \xrightarrow{a} \beta$ if $\alpha = Y\gamma$, $Y \xrightarrow{a} \gamma' \in \Delta$ and $\beta = \gamma'\gamma$.

A state (word) $\alpha \in V^*$ is **normed** if $\alpha \rightarrow^* \varepsilon$.
$\alpha \in V^*$ is **unnormed** if $\alpha$ is not normed.
BPA: An Example

\[ V = \{I, Z\} \]

\[ \Delta = \{Z \xrightarrow{Z} Z, Z \xrightarrow{i} IZ, I \xrightarrow{i} II, I \xrightarrow{d} \varepsilon\} \]

\[ Z \xrightarrow{i} IZ \xrightarrow{i} IIIZ \xrightarrow{i} IIIIZ \ldots \]

Hongfei Fu (Shanghai Jiao Tong University)
Let \((V, \Delta)\) be a BPA system and \(\Gamma\) be a FS system.

A binary relation \(R \subseteq V^* \times \text{State}(\Gamma)\) is a \textit{branching bisimulation} if whenever \((\alpha, f) \in R\) then for each \(a \in \text{Act}\):

\[
\begin{align*}
\alpha & \rightarrow_R a \rightarrow a, & \alpha & \rightarrow_R a, & \alpha & \rightarrow_R a, \\
\end{align*}
\]

or

\[
\begin{align*}
\alpha & \rightarrow_R \tau, & \alpha & \rightarrow_R \tau, & \alpha & \rightarrow_R \tau, \\
\end{align*}
\]

The \textit{branching bisimilarity} \(\approx_{br}\) is the largest branching bisimulation.
Bisimulation Base Technique
an effective technique developed by D. Cauca to decide bisimilarity concerning infinite-state systems

- a finite relation "bisimulation base" from which the whole bisimilarity relation can be effectively generated.

- We can decide the bisimulation problem if we can compute the corresponding bisimulation base.
Previous Application

- strong bisimulation over BPA/BPP
  S. Christensen, H. Hüttel, and C. Stirling 1992/
  S. Christensen, Y. Hirshfeld, and F. Moller 1993

- strong bisimulation over normed BPA/normed BPP
Previous Application

- weak bisimulation between FS and BPA/normed BPP (†)
- various bisimulations between FS and pushdown processes.
Our Application

- the application on the branching bisimilarity between a BPA system and a FS system.
- We follow and rely on the scheme of the previous work on weak bisimulation (†).
The Bisimulation Base between BPA and FS

- We fix a BPA system \((V, \Delta)\) and a FS \(\Gamma\).
- We construct the BPA system \((V', \Delta') = (V \cup State(\Gamma), \Delta \cup \Gamma)\)
- Special attention on words of the form \(\alpha f\) with \(\alpha \in V^*, f \in State(\Gamma)\).
We also define:

- $\mathcal{G}_\Gamma$ to be the $\approx_{br}$ over $\Gamma \cup \{\varepsilon\}$.
- $\text{Normed}(V) = \{X \in V | X \text{ is normed}\}$. 
A relation $K$ is well-formed if $G_\Gamma \subseteq K$ and $K$ is a subset of the relation $G$ defined by:

$$G = ((\text{Normed}(V) \cdot \text{State}(\Gamma)) \times \text{State}(\Gamma)) \cup (V \times \text{State}(\Gamma)) \cup G_\Gamma$$

$G$ is the largest well-formed relation.
The bisimulation base $\mathcal{B}$, is a well-formed relation as follows:

$$\mathcal{B} = \{(Yf, g) \in \text{Normed}(V) \cdot \text{State}(\Gamma) | Yf \approx_{br} g, Y \in \text{Normed}(V)\} \cup \{(Y, g) \in V \times \text{State}(\Gamma) | Y \approx_{br} g\} \cup \mathcal{G}_\Gamma$$
Let $K$ be a well-formed relation. The closure of $K$, denoted $\text{Cl}(K)$, is the least relation $M$ such that:

$$
\begin{align*}
K \subseteq M & \quad (Yf, g) \in K, (\alpha, f) \in M \quad (Y\alpha, g) \in M \\
& \quad (Yf, g) \in K, (\alpha h, f) \in M \quad (Y\alpha h, g) \in M \\
& \quad (\alpha, g) \in M, \alpha \text{ is unnormed} \\
& \quad (\alpha \beta, g), (\alpha \beta h, g) \in M \text{ for every } \beta \in V^* \text{ and } h \in \text{State}(\Gamma)
\end{align*}
$$

$\text{Cl}(K)$ will only contain pairs of two form: $(\alpha, g)$ and $(\alpha f, g)$. 
The Idea Behind Closure

The basic properties w.r.t sequential computation:

- $Yf \approx_{br} g$, $\alpha \approx_{br} f$ implies $Y\alpha \approx_{br} g$.
- $\alpha \approx_{br} g$, $\alpha$ is unnormed implies $\alpha\beta \approx_{br} g$, $\alpha\beta h \approx_{br} g$
Let $K$ be a well-formed relation. For each $g \in \text{State}(\Gamma)$ there is a finite automaton $A^K_g$ constructible in polynomial time such that

$$L(A^K_g) = \{ \alpha | (\alpha, g) \in Cl(K) \} \cup \{ \alpha f | (\alpha f, g) \in Cl(K) \}$$

$$(\alpha, g) \in Cl(K) \text{ iff } \alpha \in L(A^K_g).$$
The Bisimulation Base: The Key Property

**Theorem**

\[ \text{Cl}(B) = \{(\alpha, g) | \alpha \approx_{br} g\} \cup \{(\alpha f, g) | \alpha f \approx_{br} g\}. \]

\[ \alpha \approx_{br} g \text{ iff } \alpha \in A_g^B \]
Computing The Bisimulation Base
First we develop an expansion function $\text{Exp}$ over well-formed relations such that $\text{Exp}(K) \subseteq K$ for every $K$.

Then we iteratively apply $\text{Exp}$ to $G$:
$$B^0 = G, \quad B^1 = \text{Exp}(B^0), \ldots, \quad B^{k+1} = \text{Exp}(B^k), \ldots$$

Finally we obtain a fixed point of $\text{Exp}$ which is exactly the bisimulation base.
The Expansion Function: The Idea

The idea naturally follows the definition of branching bisimulation.

Let $K$ be a well-formed relation. Roughly a pair, say, $(X, g)$ expands in $K$ by the following conditions:

$\alpha' \quad \text{Cl}(K) \quad g'$

$\downarrow a \quad \downarrow a \quad \text{or} \quad \downarrow \tau \quad \text{Cl}(K) \\
\alpha'$

$\downarrow a \quad \text{Cl}(K) \quad \alpha' \quad g'$

$\downarrow a \quad \text{Cl}(K) \quad \alpha''$

$\text{Exp}(K)$ is the set of pairs of $K$ that expands in $K$. 
A binary relation $\mathcal{R} \subseteq V^* \times \text{State}(\Gamma)$ is a *branching bisimulation* if whenever $(\alpha, f) \in \mathcal{R}$ then for each $a \in \text{Act}$:

$$
\begin{align*}
\alpha & \xrightarrow{-\mathcal{R}-} f \\
\downarrow a & \quad \downarrow a \quad \text{or} \quad \downarrow \tau & \quad \mathcal{R} \\
\alpha' & \xrightarrow{-\mathcal{R}-} f'
\end{align*}
$$

The *branching bisimilarity* $\approx_{br}$ is the largest branching bisimulation.
Let $K$ be a well-formed relation. For each $g \xrightarrow{a} h \in \Gamma$, we define an auxiliary language $L^K_{[g,a,h]}$ which approximates the language:

$$L^K_{[g,a,h]} \triangleq \{ E | (E, g) \in \text{Cl}(K) \land \exists E' (E \xrightarrow{a} E' \land (E', h) \in \text{Cl}(K)) \}$$
Let $K$ be a well-formed relation. A pair $(X, g)$ *expands* in $K$ by the following conditions:

\[
\begin{align*}
X & \quad K & \quad g \\
\downarrow a & \quad \downarrow a & \text{or} & \quad \downarrow \tau & \quad Cl(K) \\
\alpha' & \quad Cl(K) & \quad g' & \quad \alpha' & \quad g'
\end{align*}
\]

where $\downarrow a \in Cl(K)$ or $K \downarrow \tau Cl(K)$.
Let $K$ be a well-formed relation. For each $g \xrightarrow{a} h \in \Gamma$, a pair $(X, g), (Xf, g) \in K$ satisfies the condition $\phi^K_{[g,a,h]}$, if:

- For $(X, g)$: $X$ is unnormed and there is some $X \xrightarrow{a} \alpha$ such that $(\alpha, h) \in Cl(K)$
- For $(Xf, g)$: there is some $X \xrightarrow{a} \alpha$ such that $(\alpha f, h) \in Cl(K)$

Then, the (regular) language $L^K_{[g,a,h]}$ is defined as the union of the regular languages below:

- $\{X\} \cdot L(A^K_f)$, for $(Xf, g)$ satisfies $\phi^K_{[g,a,h]}
- \{X\} \cdot V^* + \{X\} \cdot V^* \cdot State(\Gamma)$, for $(X, g)$ satisfies $\phi^K_{[g,a,h]}$
Let $K$ be a well-formed relation. A pair in $K$ expands in $K$ iff

- For $(Y, g)$: whenever $Y \xrightarrow{a} \alpha$, there is some $g \xrightarrow{a} g'$ such that $(\alpha, g') \in Cl(K)$, or $a = \tau$ and $(\alpha, g) \in Cl(K)$; and whenever $g \xrightarrow{a} g'$, there is some $Y \Rightarrow \alpha$ such that $\alpha \in L^K_{[g,a,g']}$. 

- For $(Yf, g)$: . . . .

- For pairs in $G\Gamma$: . . . .

The set $Exp(K)$ is defined as all pairs in $K$ that expand in $K$. 
Consistency with Branching Bisimulation

Lemma

Let $K$ be a well-formed relation such that $\text{Exp}(K) = K$. Then $\text{Cl}(K)$ is a branching bisimulation.
Procedure: $B^0 = \emptyset$, $B^{k+1} = \text{Exp}(B)$.

**Theorem**

*There is a polynomial bounded natural number $j$ such that $B^j = B^{j+1}$. Moreover, $B^j = B$.*
Theorem

Let $K$ be a well-formed relation. The set $\text{Exp}(K)$ can be computed in polynomial time.
The Main Theorem

Theorem

Branching bisimilarity between BPA and FS is polynomial time decidable.
We’ve developed a polynomial time algorithm deciding branching bisimilarity between BPA and FS.

Our algorithm is not new, but improves a previous result by Antonín Kučera and Richard Mayr (2004).

We’ve shown the scheme of the bisimulation base technique.
Questions?