A short proof of the decidability of bisimulation for normed BPA-processes

Jan Friso Groote *

Department of Software Technology, CWI, P.O. Box 4079, 1009 AB Amsterdam, Netherlands

Communicated by D. Gries
Received 19 December 1991
Revised 29 January 1992

Abstract


The decidability of bisimulation for normed processes was first proven by J.C.M. Baeten et al. (1987) and subsequently, using other proof techniques, by D. Caucal (1990) and H. Hüttel and C. Stirling (1991). We provide a short and straightforward proof.

Keywords: Formal semantics, Basic Progress Algebra, bisimulation, context-free processes, decidability

BPA (Basic Process Algebra) process expressions or BPA processes [1] are given by the abstract syntax

\[ p ::= a | X | p_1 + p_2 | p_1 \cdot p_2 \]

Here \( a \) ranges over a set \( Act \) of atomic actions, and \( X \) over a set \( Var \) of variables. In BPA the symbol + is interpreted as nondeterministic choice while \( p_1 \cdot p_2 \) represents sequential composition of \( p_1 \) and \( p_2 \) (we often omit the " \( \cdot \) "). For technical convenience, we also introduce the process \( E \), with the convention that \( E \cdot q = q \).

We say that a process expression is guarded iff every variable occurrence in \( p \) occurs in a subexpression \( aq \) of \( p \). Recursive processes are defined by guarded recursive specifications:

\[ \Delta = \{ X_i = p_i | 1 \leq i \leq k \}, \]

where the \( X_i \) are distinct variables, and the \( p_i \) are guarded BPA process expressions with free variables in \( Var(\Delta) = \{ X_1, \ldots, X_k \} \). The variable \( X_1 \) is called the root of \( \Delta \). We use letters \( \alpha, \beta, \gamma \) and \( \zeta \) to range over possibly empty sequences of variables, i.e. \( \alpha, \beta, \gamma, \zeta \in Var(\Delta)^* \). The function length gives the number of variables in a sequence.

The operational semantics of a BPA process expression, given a guarded recursive specification \( \Delta \), is a transition relation \( \rightarrow_\Delta \) containing the transitions provable by the following rules:

\[
\begin{align*}
  p \rightarrow^a p' & \quad q \rightarrow^a q' \\
  p + q \rightarrow^a p' & \quad p + q \rightarrow^a q' \\
  p \rightarrow^a p' & \quad p q \rightarrow^a p' q \\
  a \rightarrow^a \epsilon & \quad a \in Act
\end{align*}
\]
\[
\frac{p \rightarrow^p p'}{X \rightarrow^X p'} \quad X = p \in \Delta
\]

We omit the subscript $\Delta$ if it is clear from the context.

Generally, two processes are considered equivalent if they are bisimilar [5]:

**Definition 1.** A relation $R$ on processes is called a strong bisimulation relation if for all $(p, q) \in R$ it holds that

- If $p \rightarrow^p q'$, then there is a $q'$ such that $q \rightarrow^q q'$ and $p' R q'$.
- If $q \rightarrow^q q'$, then there is a $p'$ such that $p \rightarrow^p p'$ and $p' R q'$.

Two processes $p$ and $q$ are strongly bisimilar, notation $p \leftrightarrow q$, iff there is some bisimulation relation $R$ such that $p R q$.

**Lemma 2.** Strong bisimulation is a congruence relation with respect to $+$ and $\cdot$.

In this paper we restrict our attention to normed BPA process expressions.

**Definition 3.** The norm of a process $p$ is defined by ($\sigma$ represents a sequence of actions):

\[
|p| = \min\{|\sigma| \mid p \rightarrow^p \sigma\} \cup \{\infty\}.
\]

Let $\Delta$ be a guarded recursive specification. The norm of $\Delta$ is $\max\{|X| \mid X \in Var(\Delta)\}$. $\Delta$ is normed iff its norm is finite. A BPA process is called normed, if it has been generated via a normed guarded recursive specification. Note that bisimilar processes have the same norm.

**Lemma 4.** Let $p$, $p'$ and $q$ be normed BPA processes. If $p \cdot q \leftrightarrow p' \cdot q$ then $p \leftrightarrow p'$, and if $q \cdot p \leftrightarrow q' \cdot p'$ then $p \leftrightarrow p'$.

**Proof.** For the first fact, note that every step that can be done by $p$ in $p \cdot q$ must be mimicked by $p'$ in $p' \cdot q$. For the second one, note that there is some smallest trace $\sigma$ such that $q \cdot p \rightarrow^q p$. The only way for $q \cdot p'$ to mimic this is by letting $q$ perform the trace $\sigma$, i.e. $q \cdot p \rightarrow^q p'$. The results must be bisimilar and hence, $p \leftrightarrow p'$.

In [1] it is shown that any guarded recursive specification $\Delta$ can be effectively presented in the following normal form

\[
\Delta' = \{X_i = \sum_{j=1}^{n_i} a_{ij} \alpha_i \mid 1 \leq i \leq m\},
\]

where $a_{ij}$ is a variable sequence containing at most two variables, such that the root of $\Delta'$ is bisimulation equivalent to that of $\Delta$. Moreover, when $\Delta$ is normed, so is $\Delta'$. By analogy with context-free grammars $\Delta'$ is said to be in restricted GNF (Greibach Normal Form). It is worth noting that $\Delta'$ can be constructed in such a way that its size is polynomial in $\Delta$. For a recursive specification $\Delta$ in restricted GNF and a sequence $\alpha$ it holds that if $\alpha \rightarrow^\alpha p$, then $p$ is again a sequence of variables and $\text{length}(p) \leq \text{length}(\alpha) + 1$.

In the sequel we assume that $\Delta$ is a guarded recursive specification in restricted GNF.

**Definition 5.** A function

\[
f: Var(\Delta) \rightarrow Var(\Delta)^+
\]

is called a $Var(\Delta)$-assignment. Here $Var(\Delta)^+$ is the set of all nonempty sequences of variables from $Var(\Delta)$. The function $f$ is extended to sequences in the expected way $(n \geq 0)$:

\[
f(X_1 \cdots X_n) = f(X_1) \cdots f(X_n).
\]

We say that $f$ is norm-preserving iff $|X| = |f(X)|$ and $f$ is idempotent iff $f(f(X)) = f(X)$. Moreover, we say that $f$ is transfer-preserving iff for all $X \in Var(\Delta)$ and $\alpha, \beta \in Var(\Delta)^+$:

- $X \rightarrow^X \alpha$ \quad $\Rightarrow$ \quad $\exists \beta f(X) \rightarrow^\beta \beta$ and $f(\alpha) = f(\beta)$,
- $f(X) \rightarrow^X \beta$ \quad $\Rightarrow$ \quad $\exists \alpha X \rightarrow^\alpha \alpha$ and $f(\alpha) = f(\beta)$.

**Lemma 6.** Suppose $f$ is an idempotent, transfer-preserving $Var(\Delta)$-assignment. Then for all sequences of variables $\alpha$ and $\beta$:

\[
f(\alpha) = f(\beta) \quad \Rightarrow \quad \alpha \leftrightarrow \beta.
\]
Proof. It is sufficient to show that
\[ R = \{ (\alpha, \beta) \in \text{Var}(\Delta)^* \times \text{Var}(\Delta)^* | f(\alpha) = f(\beta) \} \]
is a bisimulation relation. This is trivial when \( \alpha = \epsilon \) or \( \beta = \epsilon \). So, consider nonempty sequences \( \alpha \) and \( \beta \) such that \( f(\alpha) = f(\beta) \) and suppose \( \alpha \rightarrow^* \alpha' \). First we show that for appropriate \( \gamma \), \( f(\alpha) \rightarrow^* \gamma \) and \( f(\alpha') \rightarrow^* \gamma \).

If \( \alpha = X \), then, as \( f \) is transfer-preserving, \( f(X) \rightarrow^* \gamma \) and \( f(\alpha') \rightarrow^* \gamma \). If \( \alpha = X_1 \alpha_1 \), then \( f(\alpha) = \gamma_1 \gamma_2 \) such that \( f(X_1) = \gamma_1 \) and \( f(\alpha_1) = \gamma_2 \). As \( \alpha \rightarrow^* \alpha' \), it follows that \( X_1 \rightarrow^* \alpha_1' \) and \( \alpha_1 = \alpha_1' \). Hence, \( f(\alpha) \rightarrow^* \gamma_1 \gamma_2'. \) \( f(\alpha') \rightarrow^* \gamma_1 \gamma_2' \). So, we can conclude that \( f(\alpha) \rightarrow^* \gamma_1 \gamma_2' \) and \( f(\alpha') \rightarrow^* \gamma_1 \gamma_2' \).

Now we show that if \( \alpha \rightarrow^* \gamma \), then \( \beta \rightarrow^* \beta' \) and \( f(\gamma) \rightarrow^* f(\beta') \). Assume \( \alpha \rightarrow^* \gamma \). If \( \beta = Y \), then \( f(Y) = f(\gamma) \). As \( f(\alpha) \rightarrow^* \gamma \) and \( f \) is transfer-preserving, \( Y \rightarrow^* \beta' \) and \( f(\beta') \rightarrow^* f(\gamma) \). If \( \beta = Y_1 \beta_1 \), \( f(Y_1) = \gamma_1 \) and \( f(\beta_1) = \gamma_2 \), then \( f(\alpha) \rightarrow^* \gamma_1 \gamma_2. \) Because \( f(\alpha) \rightarrow^* \gamma \) it follows that \( \gamma_1 \rightarrow^* \gamma_1' \gamma_2' \). As \( f \) is transfer-preserving, \( Y_1 \rightarrow^* \beta_1' \) and \( f(\beta_1') \rightarrow^* f(\gamma_1') \). Hence, \( \beta \rightarrow^* \beta' \beta_1' \) and \( f(\beta_1') \rightarrow^* f(\gamma_1') \). \( f(\beta_1') \rightarrow^* f(\gamma_1') \). From the previous two paragraphs it follows that if \( \alpha \rightarrow^* \alpha' \) then \( \beta \rightarrow^* \beta' \) and \( f(\alpha') \rightarrow^* f(\beta') \). The case where \( \beta \) can perform the first step is symmetric. So \( R \) is indeed a bisimulation relation.

Now we show that if \( \alpha \leftrightarrow \beta \) for normed \( \alpha \) and \( \beta \), then there exists a transfer-preserving \( \text{Var}(\Delta) \)-assignment \( f \) such that \( f(\alpha) = f(\beta) \). In order to do so, we assume a total ordering \( < \) on \( \text{Var}(\Delta) \). This ordering is extended to a total ordering on sequences of variables as follows:

\[
\alpha < \beta \text{ iff } \begin{cases} \text{length}(\alpha) < \text{length}(\beta) \text{ or } \\
\alpha \text{ is lexicographically smaller than } \beta \text{ and } \text{length}(\alpha) = \text{length}(\beta). \end{cases}
\]

We also use \( \leq, \geq \) and \( > \) with their obvious meanings.

Definition 7. The \( \text{Var}(\Delta) \)-assignment \( f \) is defined by:
\[
f(X) = \max(\{\alpha | X \leftrightarrow \alpha\}).
\]
Because \( \{\alpha | X \leftrightarrow \alpha\} \) is a nonempty, finite set, \( f \) is well-defined.

Lemma 8. If \( \Delta \) is normed, then:
(1) \( f_{\pm}(\alpha) = \max(\{\gamma | \alpha \leftrightarrow \gamma\}) \).
(2) If \( \alpha \leftrightarrow \beta \), then \( f_{\pm}(\alpha) = f_{\pm}(\beta) \).
(3) \( f_{\pm} \) is transfer-preserving.
(4) \( f_{\pm} \) is idempotent.

Proof. (1) Let \( \alpha = Z_1 \cdots Z_k \) and define \( \beta = \max(\{\gamma | \alpha \leftrightarrow \gamma\}) \). Obviously, as \( f_{\pm}(\alpha) \leftrightarrow \beta \), \( f_{\pm}(\alpha) \leq \beta \). Assume \( \beta \neq f_{\pm}(\alpha) \). By contradiction, we show that \( f_{\pm}(\alpha) \) and hence that \( f_{\pm}(\alpha) = \beta \). Let \( f_{\pm}(\alpha) = X_1 \cdots X_m \) and \( \beta = Y_1 \cdots Y_n \). Note that \( m \geq n \).
- Suppose that \( X_1 \cdots X_m = Y_1 \cdots Y_n \). Then \( m > n \). As \( |Y_1 \cdots Y_m| > 0 \), this means that \( |f_{\pm}(\alpha)| < |\beta| \) and hence \( f_{\pm}(\alpha) \) is not bisimilar to \( \beta \). Contradiction.
- So it must be the case that there is a \( 1 \leq i \leq n \) such that \( X_i \neq Y_i \). Take such \( i \) minimal, i.e. \( X_1 \cdots X_{i-1} = Y_1 \cdots Y_{i-1} \). By Lemma 4, it follows that
\[
X_i \cdots X_m \leftrightarrow Y_i \cdots Y_m. \tag{1}
\]
Now assume that \( |X_i| < |Y_i| \). There exists some shortest \( \sigma \) such that \( X_i \cdots X_m \rightarrow^* X_{i+1} \cdots X_n \). We can conclude that
\[
Y_i \cdots Y_m \rightarrow^* \zeta \cdot Y_{i+1} \cdots Y_m
\]
for some possibly empty sequence of variables \( \zeta \), where
\[
X_{i+1} \cdots X_n \leftrightarrow \zeta \cdot Y_{i+1} \cdots Y_m.
\]
Substitution in formula (1) and application of Lemma 4 gives that \( X_{i+1} \cdots X_n \leftrightarrow \zeta \cdot Y_{i+1} \cdots Y_m \).
with $f_\to(Z_i) = X_i \cdots X_{i'}$, such that $l \leq i \leq l'$. 
$f_\to(Z_i)$ is not maximal, as $X_i$ can be replaced by $Y_i$.

The case where $|Y_i| < |X_i|$ goes in the same way, but is slightly simpler.

(2) Suppose $\alpha \leftrightarrow \beta$. Then, by (1),

$$f_\to(\alpha) = \max(\{\gamma | \alpha \leftrightarrow \gamma\})$$

$$= \max(\{\gamma | \beta \leftrightarrow \gamma\}) = f_\to(\beta).$$

(3) Suppose $X \in \text{Var}(\Delta)$ and $\beta = f_\to(X)$. As $f_\to(X) \leftrightarrow \beta$, we have the following. If $X \to^\alpha \alpha'$, then $\exists \beta'$ such that $\beta \to^\alpha \beta'$ and $\alpha' \leftrightarrow \beta'$. By (2) it follows that $f_\to(\alpha') = f_\to(\beta')$. If $\beta \to^\alpha \beta'$, then $\exists \alpha'$ such that $X \to^\alpha \alpha'$ and $\alpha' \leftrightarrow \beta'$. By (2), $f_\to(\alpha') = f_\to(\beta')$.

(4) As $f_\to(\alpha) = f_\to(\beta)$,

$$f_\to(f_\to(X)) = \max(\{\alpha | f_\to(X) \leftrightarrow \alpha\})$$

$$= \max(\{\alpha | X \leftrightarrow \alpha\}) = f_\to(X).$$

Corollary 9. If $\Delta$ is normed, then $\alpha \leftrightarrow \beta$ if and there exists an idempotent and transfer-preserving $\text{Var}(\Delta)$-assignment $f$ such that $f(\alpha) = f(\beta)$.

Proof. ($\Rightarrow$) Lemma 6. ($\Leftarrow$) By Lemma 8 $f_\to$ suffices.

Lemma 10. Let $\Delta$ be normed. Suppose $f$ is an idempotent and transfer-preserving $\text{Var}(\Delta)$-assignment. Then $f$ is norm-preserving.

Proof. Since $f$ is idempotent $f(f(X)) = f(X)$. As $f$ is idempotent and transfer-preserving, $f(X) \leftrightarrow X$. So, $|f(X)| = |X|$.

Theorem 11. Bisimulation is decidable for normed BPA processes.

Proof. By Corollary 9 we must check this for idempotent and transfer-preserving $\text{Var}(\Delta)$-assignments. By Lemma 10 such $\text{Var}(\Delta)$-assignments are norm-preserving. There are only finitely many of these because each variable has a nonzero and finite norm. For any sequence of variables $\alpha$ and $\beta$, it is straightforward to calculate whether $f(\alpha) = f(\beta)$. It can also easily and effectively be checked whether such an $f$ is idempotent and transfer-preserving. So, the existence of a norm- and transfer-preserving $\text{Var}(\Delta)$-assignment with $f(\alpha) = f(\beta)$ is decidable. By Corollary 9 it follows that it is decidable whether $\alpha \leftrightarrow \beta$.

Remark 12. An original motivation for the work as presented here was to determine the complexity of deciding bisimulation for normed BPA processes. The result in this article leads to a nonde­terministic exponential algorithm. Recently, Huynh and Tian have shown that deciding bisim­ulation for normed BPA processes is in $\Sigma_2^P$, and hence in PSPACE [4]. It is an open problem whether a more efficient algorithm exists.

Remark 13. The proof in this paper resembles the proof given in [2]. The main technical difference is in the concept of a transfer-preserving $\text{Var}(\Delta)$-assignment, versus an auto-bisimulable relation in [2], and in the presentation. For an easy compar­ison we indicate the relation between the two most important concepts. The proof in [2] de­pends on the notions of an auto-bisimulable relation and a fundamental relation. A fundamental relation is modulo the difference in representa­tion a norm-preserving and idempotent $\text{Var}(\Delta)$-assignment. An auto-bisimulable relation is a wider notion than transfer-preserving, but they coincide for fundamental relations. The main ar­gument given in [2] is that the reflexive, transitive closure of auto-bisimulable and fundamental re­lations coincides with strong bisimulation equiva­lence, which is in a sense exactly what Corollary 9 says.

Acknowledgment

I thank Didier Caucal, Dung T. Huynh, Jan Willem Klop, Alexandru Mateescu, Alban Ponse, Colin Stirling and Frits Vaandrager for their helpful comments.

References

[1] J.C.M. Baeten, J.A. Bergstra and J.W. Klop, Decidability of bisimulation equivalence for processes generating con-


