Information Processing Letters 42 (1992) 167–171
North-Holland
25 May 1992

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

Groote, J.F., A short proof of the decidability of bisimulation for normed BPA-processes, Information Processing Letters 42 (1992) 167–171.

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

$$p ::= a \mid X \mid p_1 + p_2 \mid p_1 \cdot p_2$$

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 ε, with the convention that $\varepsilon \cdot q = q$.

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:

$$\Delta = \{X_i = p_i \mid 1 \leqslant i \leqslant k\},$$

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 Δ. We use letters α, β, γ and ζ 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 Δ, is a transition relation $\rightarrow_\Delta$ containing the transitions provable by the following rules:

$$\frac{p \rightarrow^a p'}{p + q \rightarrow^a p'} \qquad \frac{q \rightarrow^a q'}{p + q \rightarrow^a q'}$$

$$\frac{p \rightarrow^a p'}{pq \rightarrow^a p'q} \qquad a \rightarrow^a \varepsilon \quad a \in Act$$

Correspondence to: J.F. Groote, Department of Philosophy, University of Utrecht, P.O. Box 80.126, 3508 TC Utrecht, Netherlands.

* The author is supported by the European Communities under RACE project no. 1046 (SPECS) and ESPRIT Basic Research Action 3006 (CONCUR).

$$\frac{p \to^a p'}{X \to^a p'} \quad X = p \in \Delta$$

We omit the subscript Δ 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 iff for all $(p, q) \in R$ it holds that

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

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

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 (σ represents a sequence of actions):

$$|p| = \min(\{length(\sigma) \mid p \to^\sigma \varepsilon\} \cup \{\infty\}).$$

Let Δ be a guarded recursive specification. The *norm* of Δ is $\max(\{|X| \mid X \in Var(\Delta)\})$. Δ 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 $p \cdot q \underline{\leftrightarrow} p' \cdot q$ then $p \underline{\leftrightarrow} p'$, and if $q \cdot p \underline{\leftrightarrow} q \cdot p'$ then $p \underline{\leftrightarrow} p'$.*

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 σ such that $q \cdot p \to^\sigma p$. The only way for $q \cdot p'$ to mimic this is by letting q perform the trace σ, i.e. $q \cdot p \to^\sigma p'$. The results must be bisimilar and hence, $p \underline{\leftrightarrow} p'$. □

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

$$\Delta' = \left\{ X_i = \sum_{j=1}^{n_i} a_{ij}\alpha_{ij} \mid 1 \leqslant i \leqslant m \right\},$$

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

In the sequel we assume that Δ is a guarded recursive specification in restricted GNF.

Definition 5. A function

$$f: Var(\Delta) \to Var(\Delta)^+$$

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 ($n \geqslant 0$):

$$f(X_1 \cdots X_n) = f(X_1) \cdots f(X_n).$$

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 $\alpha, \beta \in Var(\Delta)^*$:

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

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

$$f(\alpha) = f(\beta) \quad \Rightarrow \quad \alpha \underline{\leftrightarrow} \beta.$$

Proof. It is sufficient to show that

$$R = \{\langle \alpha, \beta\rangle \in Var(\Delta)^* \times Var(\Delta)^* \mid f(\alpha) = f(\beta)\}$$

is a bisimulation relation. This is trivial when $\alpha = \varepsilon$ or $\beta = \varepsilon$. So, consider nonempty sequences α and β such that $f(\alpha) = f(\beta)$ and suppose $\alpha \to^a \alpha'$. First we show that for appropriate γ, $f(\alpha) \to^a \gamma$ and $f(\alpha') = f(\gamma)$.

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

$$f(\alpha') = f(\alpha_1'\alpha_1) = f(\alpha_1')f(f(\alpha_1)) = f(\gamma_1')f(\gamma_2) = f(\gamma_1'\gamma_2).$$

Now we show that if $f(\alpha) \to^a \gamma$, then $\beta \to^a \beta'$ and $f(\gamma) = f(\beta')$. Assume $f(\alpha) \to^a \gamma$. If $\beta = Y$, then $f(Y) = f(\alpha)$. As $f(\alpha) \to^a \gamma$ and f is transfer-preserving, $Y \to^a \beta'$ and $f(\beta') = f(\gamma)$. If $\beta = Y_1\beta_1$, $f(Y_1) = \gamma_1$ and $f(\beta_1) = \gamma_2$, then $f(\alpha) = \gamma_1\gamma_2$. Because $f(\alpha) \to^a \gamma$ it follows that $\gamma_1 \to^a \gamma_1'$ and $\gamma = \gamma_1'\gamma_2$. As f is transfer-preserving, $Y_1 \to^a \beta_1'$ and $f(\beta_1') = f(\gamma_1')$. Hence, $\beta \to^a \beta_1'\beta_1$ and

$$f(\beta_1'\beta_1) = f(\gamma_1')f(f(\beta_1)) = f(\gamma_1')f(\gamma_2) = f(\gamma_1'\gamma_2) = f(\gamma).$$

From the previous two paragraphs it follows that if $\alpha \to^a \alpha'$ then $\beta \to^a \beta'$ and $f(\alpha') = f(\beta')$. The case where β can perform the first step is symmetric. So R is indeed a bisimulation relation. □

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

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

We also use $\leqslant$, $\geqslant$ and $>$ with their obvious meanings.

Definition 7. The $Var(\Delta)$-assignment $f_{\underline{\leftrightarrow}}$ is defined by:

$$f_{\underline{\leftrightarrow}}(X) = \max(\{\alpha \mid X \underline{\leftrightarrow} \alpha\}).$$

Because $\{\alpha \mid X \underline{\leftrightarrow} \alpha\}$ is a nonempty, finite set, $f_{\underline{\leftrightarrow}}$ is well-defined.

Lemma 8. *If Δ is normed, then*:

(1) $f_{\underline{\leftrightarrow}}(\alpha) = \max(\{\gamma \mid \alpha \underline{\leftrightarrow} \gamma\})$.
(2) *If $\alpha \underline{\leftrightarrow} \beta$, then $f_{\underline{\leftrightarrow}}(\alpha) = f_{\underline{\leftrightarrow}}(\beta)$.*
(3) *$f_{\underline{\leftrightarrow}}$ is transfer-preserving.*
(4) *$f_{\underline{\leftrightarrow}}$ is idempotent.*

Proof. (1) Let $\alpha = Z_1 \cdots Z_k$ and define $\beta = \max(\{\gamma \mid \alpha \underline{\leftrightarrow} \gamma\})$. Obviously, as $f_{\underline{\leftrightarrow}}(\alpha) \underline{\leftrightarrow} \beta$, $f_{\underline{\leftrightarrow}}(\alpha) \leqslant \beta$. Assume $\beta > f_{\underline{\leftrightarrow}}(\alpha)$. By contradiction, we show that $\beta \leqslant f_{\underline{\leftrightarrow}}(\alpha)$ and hence that $f_{\underline{\leftrightarrow}}(\alpha) = \beta$. Let $f_{\underline{\leftrightarrow}}(\alpha) = X_1 \cdots X_n$ and $\beta = Y_1 \cdots Y_m$. Note that $m \geqslant n$.

- Suppose that $X_1 \cdots X_n = Y_1 \cdots Y_n$. Then $m > n$. As $|Y_{n+1} \cdots Y_m| > 0$, this means that $|f_{\underline{\leftrightarrow}}(\alpha)| < |\beta|$ and hence $f_{\underline{\leftrightarrow}}(\alpha)$ is not bisimilar to β. Contradiction.
- So it must be the case that there is a $1 \leqslant i \leqslant n$ 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

$$X_i \cdots X_n \underline{\leftrightarrow} Y_i \cdots Y_m. \tag{1}$$

Now assume that $|X_i| \leqslant |Y_i|$. There exists some shortest σ such that $X_i \cdots X_n \to^\sigma X_{i+1} \cdots X_n$. We can conclude that

$$Y_i \cdots Y_m \to^\sigma \zeta \cdot Y_{i+1} \cdots Y_m$$

for some possibly empty sequence of variables ζ, where

$$X_{i+1} \cdots X_n \underline{\leftrightarrow} \zeta \cdot Y_{i+1} \cdots Y_m.$$

Substitution in formula (1) and application of Lemma 4 gives that $X_i\zeta \underline{\leftrightarrow} Y_i$. If ζ is not empty, β is not maximal, as replacing $X_i\zeta$ for Y_i in β yields a "larger" sequence. If ζ is empty, then $X_i \underline{\leftrightarrow} Y_i$. If $X_i > Y_i$, then β is not maximal; replace Y_i by X_i. If $X_i < Y_i$, then there is a j

with $f_{\underline{\leftrightarrow}}(Z_j) = X_l \cdots X_{l'}$ such that $l \leqslant i \leqslant l'$. $f_{\underline{\leftrightarrow}}(Z_j)$ 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 $\alpha \underline{\leftrightarrow} \beta$. Then, by (1),

$$\begin{aligned} f_{\underline{\leftrightarrow}}(\alpha) &= \max(\{\gamma \mid \alpha \underline{\leftrightarrow} \gamma\}) \\ &= \max(\{\gamma \mid \beta \underline{\leftrightarrow} \gamma\}) = f_{\underline{\leftrightarrow}}(\beta). \end{aligned}$$

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

(4) As $f_{\underline{\leftrightarrow}}(X) \underline{\leftrightarrow} X$,

$$\begin{aligned} f_{\underline{\leftrightarrow}}(f_{\underline{\leftrightarrow}}(X)) &= \max(\{\alpha \mid f_{\underline{\leftrightarrow}}(X) \underline{\leftrightarrow} \alpha\}) \\ &= \max(\{\alpha \mid X \underline{\leftrightarrow} \alpha\}) = f_{\underline{\leftrightarrow}}(X). \quad \square \end{aligned}$$

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

Proof. ($\Leftarrow$) Lemma 6. ($\Rightarrow$) By Lemma 8 $f_{\underline{\leftrightarrow}}$ suffices. □

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

Proof. Since f is idempotent $f(f(X)) = f(X)$. As f is idempotent and transfer-preserving, $f(X) \underline{\leftrightarrow} X$. 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 $Var(\Delta)$-assignments. By Lemma 10 such $Var(\Delta)$-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 α and β, 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 $Var(\Delta)$-assignment with $f(\alpha) = f(\beta)$ is decidable. By Corollary 9 it follows that it is decidable whether $\alpha \underline{\leftrightarrow} \beta$. □

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 Σ_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 $Var(\Delta)$-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 $Var(\Delta)$-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-

text-free languages, in: J.W. de Bakker, A.J. Nijman and P.C. Treleaven, eds., *Proc. PARLE Conf., Vol. II (Parallel Languages)*, Eindhoven, Lecture Notes in Computer Science **259** (Springer, Berlin, 1987) 94–113.

[2] D. Caucal, Graphes canoniques de graphes algébriques, *Theoret. Inform. and Appl.* **24** (4) (1990) 339–352.

[3] H. Hüttel and C. Stirling, Actions speak louder than words: Proving bisimilarity for context-free processes, in: *Proc. 6th Ann. Symp. on Logic in Computer Science*, Amsterdam, The Netherlands (IEEE Computer Society Press, Silver Spring, MD, 1991) 376–386.

[4] D.T. Huynh and L. Tian, Deciding bisimilarity of normed context-free processes is in Σ_2^p, Tech. Rept. UTDCS-1-92, University of Texas at Dallas, 1992.

[5] D.M.R. Park, Concurrency and automata on infinite sequences, in: P. Deussen, ed., *Proc. 5th GI Conf.*, Lecture Notes in Computer Science **104** (Springer, Berlin, 1981) 167–183.