Bisimulation assertions for normed value-passing BPA processes

Paweł Pączkowski*

Institute of Mathematics
University of Gdańsk
ul. Wita Stwosza 57
80–952 Gdańsk, Poland

matpmp@univ.gda.pl

June 2001, revised February 2002

Abstract

Typical proof systems for bisimulation equivalence for value-passing processes use bisimulation assertions that are relativised with respect to valuations of data variables that appear free in processes. In this paper, new relativised bisimulation assertions are introduced for two variants of value-passing BPA, allowing us to show a form of congruence property in each case.

1 Introduction

Behaviours of value-passing processes may depend on the values that the processes receive in communications and, consequently, reasoning about value-passing processes usually involves reasoning about the transmitted data. In technical terms of process calculi formalisms, the dependence of process behaviours on data boils down to the presence of free data variables in process terms and the dependence of semantic transitions on valuations of these variables. Accordingly, typical proof systems for bisimulation equivalence for value-passing processes use bisimulation assertions that are relativised with respect to valuations of data variables that appear free in processes. In [5, 11] this relativisation takes the form

$$b \rhd E \sim E' \tag{1}$$

*This research has been financially supported by KBN Grant 8 T11 C 037 16

which intuitively means that processes E and E' are bisimilar when the initial valuation of free data variables appearing in the processes satisfies the property b expressed in some logic of data expressions. Similarly, [4, 7, 9] provide methods for characterizing such initial valuations of data variables appearing in processes that guarantee bisimulation; this again amounts to the relativisation of the form (1).

In contrast to aforementioned work on process calculi based on CCS-style action prefixing, in this paper a value-passing version of BPA is considered, whose distinguishing feature is the presence of the sequential composition operator.

For pure processes, (strong) bisimulation equivalence is congruent with respect to sequential composition ($E \sim E'$ and $F \sim F'$ implies $EF \sim E'F'$) and in a proof system for normed BPA given in [6] a rule of this kind appears. However, in order to provide a sound counterpart of this rule for the value-passing case, more elaborate forms of relativising bisimulation assertions than (1) need to be considered. In this paper, such relativised bisimulation assertions are introduced for two variants of value-passing BPA and a form of congruence property is shown in each case. A proof system for bisimulation equivalence on normed value-passing BPA processes that uses the introduced bisimulation assertions is currently under investigation.

A related study of value-passing BPA processes appeared in [1, 2, 8] providing a similar characterization of bisimilarity as in [4, 7, 9], however, no recursively defined BPA processes were considered there.

In [3] a transformation technique is shown for a class of value-passing BPA process, which essentially implies that the processes we are dealing with in this paper can be transformed into a prefix based form, which has been already dealt with in [11] or [7, 9]. However, such a transformation encodes a lot of the control structure of processes into the data domain, removing, in effect, the sequential composition of processes. Here, we work with processes where sequential composition is explicit.

The paper is organised as follows. Value-passing BPA processes (introduced in [10]) are described in Section 2, Section 3 contains the main technical contribution — the bisimulation assertions and congruence properties, Section 4 briefly indicates some further work.

2 Value-passing BPA processes

The class of processes we consider, introduced in [10], is obtained by extending BPA processes, as defined in [6], with value-passing primitives along the lines of [9, 7]. We assume 3-Greibach normal form of processes, but the congruence properties do not rely on this presentation of processes.

Let $X, Y, Z \ldots$ be a finite set of process variables. A *value-passing BPA process* is defined by a finite set of equations Δ of the form

$$X_i \stackrel{\mathsf{def}}{=} \sum_{j=1}^{n_i} a_{ij}\beta_{ij}, \qquad i = 1 \ldots m \tag{2}$$

where X_i are process variables, a_{ij} are actions described below and β_{ij} are process variable sequences of length at most two. For every equation Δ we assume some finite set of data variables $var(\Delta)$ ranged over by x. The actions a_{ij} can have two forms:

$$(b\,,\ c?x\,,\ s) \qquad \text{or} \qquad (b\,,\ c!e\,,\ s) \tag{3}$$

where b is a boolean expression, c is a channel name drawn from a fixed set of channel names, x is a data variable, e is an expression from a presupposed set of data expressions, and s is a simultaneous assignment:

$$(x_1, \ldots, x_k) := (e_1, \ldots, e_k)$$

Intuitively, an action $(b\,,\ c?x\,,\ s)$ should be interpreted as an input of a value from channel c, under the condition that boolean b holds (before the new value of x has been received), after which the data variables are updated by performing assignment s. The output action $(b\,,\ c!e\,,\ s)$ differs in that the value of expression e is output to channel c.

In processes defined above, data variables can appear in data expressions and in boolean expressions. Thus, behaviours of processes depend on a global state, i.e. a valuation of data variables. The actions can modify the state by performing inputs and assignments; the output values depend on the current valuation of data variables. The set of equations Δ describes the context-free control of the process execution.

The semantics of processes is defined by the following two-step procedure. Given a set of equations Δ, we define first the *symbolic transitions* of the form

$$\alpha \xrightarrow{a} \beta$$

where α, β are sequences of process variables and a is an action as described by (3). The following rule is used to derive these transitions:

$$X\alpha \xrightarrow{a_j} \beta_j\alpha \quad \text{if } \Delta \text{ contains } X \stackrel{\mathsf{def}}{=} \sum_{j=1}^{n} a_j\beta_j$$

This definitions, essentially, follows [6] if the actions are treated just as symbols, disregarding their structure.

Next, the symbolic transitions are interpreted with respect to valuations of data variables to derive *semantic transitions* of the form

$$\langle\sigma, \alpha\rangle \xrightarrow{c?v} \langle\rho, \beta\rangle \qquad \text{or} \qquad \langle\sigma, \alpha\rangle \xrightarrow{c!v} \langle\rho, \beta\rangle$$

where σ, ρ are valuations of data variables, i.e. functions from data variables to a presupposed set of data values, c is a channel name and v ranges over data values. The actions $c?v$ and $c!v$ which label semantic transitions represent, respectively, an input of value v from channel c and output of value v on channel c.

The following two rules are used to derive semantic transitions. We use the notation $\sigma[s]$ for a valuation obtained from σ by performing assignment s and we denote by $[\![e\sigma]\!]$ the value of expression e under valuation σ. As usually, $\sigma \models b$ stands for 'boolean b is satisfied under valuation σ'.

$$\frac{\alpha \xrightarrow{b\,,\,c!e\,,\,s} \beta \qquad \sigma \models b}{\langle \sigma, \alpha \rangle \xrightarrow{c![\![e\sigma]\!]} \langle \sigma[s], \beta \rangle} \qquad\qquad \frac{\alpha \xrightarrow{b\,,\,c?x\,,\,s} \beta \qquad \sigma \models b}{\langle \sigma, \alpha \rangle \xrightarrow{c?v} \langle \sigma[x \mapsto v][s], \beta \rangle}, \quad \text{for any value } v$$

A process defined by (2) is called *normed* if for every variable X in Δ there is a finite sequence of symbolic transitions that starts from X and ends on ε.

3 Bisimulation assertions

Let Δ and Δ' be two disjoint sets of equations such that $var(\Delta) \cap var(\Delta') = \emptyset$. In the sequel we will be concerned with bisimulation relations on transition systems consisting of the semantic transitions of Δ and Δ'. Assuming common set of data values, the labels of semantic transitions of Δ and Δ' are drawn from the same set so the standard notion of bisimulation applies and the standard notation $\langle \sigma, \alpha \rangle \sim \langle \sigma', \alpha' \rangle$ will be used to say that there is a (strong) bisimulation relation containing $(\langle \sigma, \alpha \rangle, \langle \sigma', \alpha' \rangle)$. We will often use the notational convention of adorning with a prime all of the symbols related to Δ'

Let σ, σ' be two valuations of (all) variables, including $var(\Delta) \cup var(\Delta')$. The juxtaposition $\sigma\sigma'$ will denote a new valuation which differs from σ in that $(\sigma\sigma')(x') = \sigma'(x')$ for every $x' \in var(\Delta')$.

Proposition 1 *Let σ, δ be two valuations of variables. If $\sigma(x) = \delta(x)$ for every $x \in var(\Delta)$ then $\langle \sigma, \alpha \rangle \sim \langle \delta, \alpha \rangle$ for any sequence α of process variables of Δ.*

It follows that $\langle \sigma, \alpha \rangle \sim \langle \sigma\sigma', \alpha \rangle$ and $\langle \sigma', \alpha' \rangle \sim \langle \sigma\sigma', \alpha' \rangle$ for any sequences α and α' of variables of Δ and Δ', respectively.

Definition 2 R is a *valuation-merged bisimulation* (vm-bisimulation, in short) if whenever $(\langle \sigma, \alpha \rangle, \langle \sigma, \alpha' \rangle) \in R$ then

(2.1) $\langle \sigma, \alpha \rangle \xrightarrow{u} \langle \delta, \beta \rangle$ implies that there is a transition $\langle \sigma, \alpha' \rangle \xrightarrow{u} \langle \delta', \beta' \rangle$ such that $(\langle \delta\delta', \beta \rangle, \langle \delta\delta', \beta' \rangle) \in R$

(2.2) $\langle \sigma, \alpha' \rangle \xrightarrow{u} \langle \delta', \beta' \rangle$ implies that there is a transition $\langle \sigma, \alpha \rangle \xrightarrow{u} \langle \delta, \beta \rangle$ such that $(\langle \delta\delta', \beta \rangle, \langle \delta\delta', \beta' \rangle) \in R$

Proposition 3 *If R is a vm-bisimulation then $(\langle \sigma, \alpha \rangle, \langle \sigma, \alpha' \rangle) \in R$ implies $\langle \sigma, \alpha \rangle \sim \langle \sigma, \alpha' \rangle$*

Proof By the conclusion from Proposition 1 if follows that R is a bisimulation up to bisimulation, hence the required property holds. $\square$

Now we are ready to formulate the first of our bisimilarity assertions, for a restricted class of processes, where all the boolean conditions in symbolic actions are *true*.

Definition 4 $(p, q) \rhd \beta \sim \beta'$ iff there exists a vm-bisimulation R such that

(4.1) $\sigma \models p$ implies $(\langle\sigma,\beta\rangle,\langle\sigma,\beta'\rangle) \in R$ and

(4.2) $(\langle\sigma,\varepsilon\rangle,\langle\sigma,\varepsilon\rangle) \in R$ implies $\sigma \models q$.

The following congruence property holds for the above bisimulation assertion.

Proposition 5 *Let α, β, α', β' be sequences of process variables generated, respectively, by sets of equations Δ and Δ', whose all symbolic actions have true as the boolean part. If $(p,r) \rhd \alpha \sim \alpha'$ and $(r,q) \rhd \beta \sim \beta'$ then $(p,q) \rhd \alpha\beta \sim \alpha'\beta'$.*

Proof Let R_1 and R_2 be the vm-bisimulations justifying, respectively, $(p,r) \rhd \alpha \sim \alpha'$ and $(r,q) \rhd \beta \sim \beta'$. We show that the relation

$$R = \{(\langle\sigma,\gamma\beta\rangle,\langle\sigma,\gamma'\beta'\rangle) \mid (\langle\sigma,\gamma\rangle,\langle\sigma,\gamma'\rangle) \in R_1\} \cup R_2$$

is a vm-bisimulation justifying the thesis. Let us examine one of possible cases; the others are simpler. Let $(\langle\sigma,\gamma\beta\rangle,\langle\sigma,\gamma'\beta'\rangle) \in R$ and $\langle\sigma,\gamma\beta\rangle \xrightarrow{u} \langle\delta,\beta\rangle$. Then it must be the case that γ is some process variable X and $\langle\sigma,X\rangle \xrightarrow{u} \langle\delta,\varepsilon\rangle$. As R_1 is a vm-bisimulation, γ' is also a process variable, say X', for which $\langle\sigma,X'\rangle \xrightarrow{u} \langle\delta',\varepsilon\rangle$ and $(\langle\delta\delta',\varepsilon\rangle,\langle\delta\delta',\varepsilon\rangle) \in R_1$. But then $\delta\delta' \models r$, hence $(\langle\delta\delta',\beta\rangle,\langle\delta\delta',\beta'\rangle) \in R_2 \subset R$ as required. $\Box$

The bisimulation assertions for unrestricted processes must take into account the possibility of blocking on boolean conditions of symbolic actions before reaching ε.

Definition 6 $(p,q,d,d') \rhd \beta \sim \beta'$ iff there exists a vm-bisimulation R such that

(6.1) $\sigma \models p$ implies $(\langle\sigma,\beta\rangle,\langle\sigma,\beta'\rangle) \in R$,

(6.2) $(\langle\sigma,\varepsilon\rangle,\langle\sigma,\varepsilon\rangle) \in R$ implies $\sigma \models q$,

(6.3) $(\langle\sigma,\gamma\rangle,\langle\sigma,\varepsilon\rangle) \in R$, where $\gamma \not\equiv \varepsilon$, implies $\sigma \models d$ and

(6.4) $(\langle\sigma,\varepsilon\rangle,\langle\sigma,\gamma\rangle) \in R$, where $\gamma \not\equiv \varepsilon$, implies $\sigma \models d'$.

Before we formulate the second congruence property, let us introduce a notation $cond(\beta)$ for the disjunction of all boolean conditions appearing in the actions a_j in the defining equation $X \stackrel{\mathsf{def}}{=} \sum_{j=1}^{n} a_j\beta_j$ for the first variable of β. Thus $cond(\beta)$ expresses the condition under which any semantic transition can originate from β.

Proposition 7 *If $(p_1,r,d_1,d_1') \rhd \alpha \sim \alpha'$, $(r,q_2,d_2,d_2') \rhd \beta \sim \beta'$, $d_1 \Rightarrow \neg cond(\beta')$ and $d_1' \Rightarrow \neg cond(\beta)$ then $(p_1,q_2,d_2,d_2') \rhd \alpha\beta \sim \alpha'\beta'$.*

Proof Let R_1 and R_2 be the vm-bisimulations justifying, respectively, $(p_1,r,d_1,d_1') \rhd \alpha \sim \alpha'$ and $(r,q_2,d_2,d_2') \rhd \beta \sim \beta'$. As with Proposition 5 we show that the relation

$$R = \{(\langle\sigma,\gamma\beta\rangle,\langle\sigma,\gamma'\beta'\rangle) \mid (\langle\sigma,\gamma\rangle,\langle\sigma,\gamma'\rangle) \in R_1\} \cup R_2$$

is a vm-bisimulation justifying the thesis. Let us examine $(\langle\sigma,\gamma\beta\rangle,\langle\sigma,\beta'\rangle) \in R$ as an interesting exemplary case. If there were a transition $\langle\sigma,\gamma\beta\rangle \xrightarrow{u} \langle\delta,\gamma_1\beta\rangle$, then R_1 would not have been a vm-bisimulation. Now, suppose there is a transition $\langle\sigma,\beta'\rangle \xrightarrow{u} \langle\delta',\beta_1'\rangle$. But by condition (6.3) we have $\sigma \models d_1$ which combined with $d_1 \Rightarrow \neg cond(\beta')$ excludes the existence of such a transition. $\Box$

4 Concluding remarks

The congruence properties of the previous section can be viewed as sound rules for proving bisimulation for value-passing BPA processes. A proof system containing the rule formulated as Proposition 5 has been already developed for a restricted class of normed value-passing BPA processes. Although the soundness of this proof system can be shown without exploiting Proposition 5, this congruence property helped to justify the choice of bisimulation assertions used in the proof system. This line of work is aimed at extending the results of [10].

References

[1] J.F. Groote and S.P. Luttik, *Undecidability and completeness results for process algebras with alternative quantification over data*, Report SEN-R9806, CWI, Amsterdam, (available at http://www.cwi.nl/~mcrl) 1998.

[2] J.F. Groote and S.P. Luttik, *A complete axiomatisation of branching bisimulation for process algebras with alternative quantification over data*, Report SEN-R9830, CWI, Amsterdam, (available at http://www.cwi.nl/~mcrl) 1998.

[3] J.F. Groote, A. Ponse and Y.S. Usenko, *Linearization in parallel pCRL* Report SEN-R0019, CWI, Amsterdam, (available at http://www.cwi.nl/~mcrl) 2000.

[4] M. Hennessy and H. Lin, *Symbolic bisimulations*, Theoretical Computer Science 138, pp. 353–389, 1995.

[5] M. Hennessy and H. Lin, *Proof systems for message-passing algebras*, Formal aspects of Computer Science, pp. 379–407, 1996

[6] H. Hüttel and C. Stirling, *Actions speak louder than words: proving bisimilarity for context-free processes*, J. Logic Computat. 8 (4), pp. 485-509, 1998.

[7] H. Lin, *Symbolic transition graph with assignment*, in: Proceedings of CONCUR'96, LNCS 1119, Springer-Verlag, 1996.

[8] S.P. Luttik, *Complete axiomatisations of weak-, delay-, and eta-bisimulation for process algebras with alternative quantification over data*, Report SEN-R9914, CWI, Amsterdam, (available at http://www.cwi.nl/~mcrl) 1999.

[9] P. Pączkowski, *Characterizing bisimilarity of value-passing parametrised processes*, Proceedings of INFINITY, Electronic Notes of Theoretical Computer Science 5, 1997. http://www.elsevier.nl/locate/entcs/volume5.html

[10] P. Paczkowski, *Towards characterizing bisimilarity of value-passing processes with context-free control*, Electronic Notes in Theoretical Computer Science 18, 1998. http://www.elsevier.nl/locate/entcs/volume18.html

[11] Julian Rathke, *Symbolic Techniques for Value-Passing Calculi*, D. Phil. thesis, University of Sussex, 1997.