Basic Research in Computer Science

Bisimilarity is not Finitely Based over BPA with Interrupt

Luca Aceto
Willem Jan Fokkink
Anna Ingólfsdóttir
Sumit Nain

BRICS Report Series **RS-04-24**

ISSN 0909-0878 **October 2004**

**See back inner page for a list of recent BRICS Report Series publications.
Copies may be obtained by contacting:**

**BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark
Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@brics.dk**

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

`http://www.brics.dk`
`ftp://ftp.brics.dk`
This document in subdirectory `RS/04/24/`

Bisimilarity is not Finitely Based over BPA with Interrupt

Luca Aceto[*,†] Wan Fokkink[‡] Anna Ingolfsdottir[*,§]
Sumit Nain[*]

Abstract

This paper shows that bisimulation equivalence does not afford a finite equational axiomatization over the language obtained by enriching Bergstra and Klop's Basic Process Algebra with the interrupt operator. Moreover, it is shown that the collection of closed equations over this language is also not finitely based.

AMS SUBJECT CLASSIFICATION (1991): 68Q15, 68Q70.
CR SUBJECT CLASSIFICATION (1991): D.3.1, F.1.1, F.4.1.
KEYWORDS AND PHRASES: Concurrency, process algebra, Basic Process Algebra (BPA), interrupt, bisimulation, equational logic, complete axiomatizations, non-finitely based algebras, expressiveness.

1 Introduction

Programming and specification languages often include constructs to specify mode switches (see, e.g., [7, 10, 22, 23, 25]). Indeed, some form of mode transfer in computation appears in the time-honoured theory of operating systems in the guise of, e.g., interrupts, in programming languages as exceptions, and in the behaviour of control programs and embedded systems as discrete "mode switches" triggered by changes in the state of their environment.

[*] **BRICS** (**B**asic **R**esearch **i**n **C**omputer **S**cience), Centre of the Danish National Research Foundation, Department of Computer Science, Aalborg University, Fr. Bajersvej 7B, 9220 Aalborg Ø, Denmark. Email: luca@cs.aau.dk (Luca Aceto), annai@cs.aau.dk (Anna Ingolfsdottir), nain@cs.aau.dk (Sumit Nain).

[†] School of Computer Science, Reykjavík University, Ofanleiti 2, 103 Reykjavík, Iceland.

[‡] Vrije Universiteit Amsterdam, Department of Computer Science, Section Theoretical Computer Science, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands. Email: wanf@cs.vu.nl.

[§] Department of Computer Science, University of Iceland, 107 Reykjavík, Iceland. Email: annaing@hi.is.

In light of the ubiquitous nature of mode changes in computation, it is not surprising that classic process description languages either include primitive operators to describe mode changes—for instance, LOTOS [14, 22] offers the so-called *disruption operator*—or have been extended with variations on mode transfer operators. For instance, examples of such operators that may be added to CCS are discussed by Milner in [24, pp. 192–193], and the reference [16] offers some discussion of the benefits of adding one of those, viz. the *checkpointing operator*, to that language.

In the setting of Basic Process Algebra (BPA), as introduced by Bergstra and Klop in [11], some of these extensions, and their relative expressiveness, have been discussed in the early paper [10]. That preprint of Bergstra's has later been revised and extended in [6]. *Ibidem*, Baeten and Bergstra study the equational theory and expressiveness of BPA_δ (the extension of BPA with a constant δ to describe "deadlock") enriched with two mode transfer operators, viz. the *disrupt* and *interrupt* operators. In particular, they offer an equational axiomatization of bisimulation equivalence [24, 28] over the resulting extension of the language BPA_δ. This axiomatization is finite, if so is the underlying set of actions—a state of affairs that is most pleasing for process algebraists.

However, the axiomatization of bisimulation equivalence offered by Baeten and Bergstra in *op. cit.* relies on the use of four auxiliary operators—two per mode transfer operator. Although the use of auxiliary operators in the axiomatization of behavioral equivalences over process description languages has been well established since Bergstra and Klop's axiomatization of parallel composition using the left and communication merge operators [12], to our mind, a result like the aforementioned one always begs the question whether the use of auxiliary operators is necessary to obtain a finite axiomatization of bisimulation equivalence.

For the case of parallel composition, Moller showed in [26, 27] that strong bisimulation equivalence is not finitely based over CCS [24] and PA [12] without the left merge operator. (The process algebra PA [12] contains a parallel composition operator based on pure interleaving without communication and the left merge operator.) Thus auxiliary operators are necessary to obtain a finite axiomatization of parallel composition. But, is the use of auxiliary operators necessary to give a finite axiomatization of bisimulation equivalence over the language BPA enriched with the mode transfer operators studied by Baeten and Bergstra in [6]?

We address the above natural question in this paper. In particular, we focus on BPA enriched with the interrupt operator. Intuitively, "p interrupted by q" describes a process that normally behaves like p. However, at each point of the computation before p terminates, q can interrupt it, and begin its execution. If this happens, p resumes its computation upon termination of q.

We show that, in the presence of two distinct actions, bisimulation equivalence

is *not* finitely based over BPA with the interrupt operator. Moreover, we prove that the collection of closed equations over this language is also not finitely based. This result provides some evidence that the use of auxiliary operators in the technical developments presented in [6] is indeed necessary in order to obtain a finite axiomatization of bisimulation equivalence.

Our main result adds the interrupt operator to the list of operators whose addition to a process algebra spoils finite axiomatizability modulo bisimulation equivalence; see, e.g., [3, 5, 13, 15, 19, 29, 30] for other examples of non-finite axiomatizability results over process algebras, and some of their precursors in the setting of formal language theory. Of special relevance for concurrency theory are the aforementioned results of Moller's to the effect that the process algebras CCS and PA without the auxiliary left merge operator from [11] do not have a finite equational axiomatization modulo bisimulation equivalence [26, 27]. Recently, in collaboration with Luttik, the first three authors have shown in [4] that the process algebra obtained by adding Hennessy's merge operator from [21] to CCS does not have a finite equational axiomatization modulo bisimulation equivalence. Fokkink and Luttik have shown in [17] that the process algebra PA [12] affords an ω-complete axiomatization that is finite if so is the underlying set of actions. Aceto, Ésik and Ingolfsdottir proved in [2] that there is no finite equational axiomatization that is ω-complete for the max-plus algebra of the natural numbers, a result whose process algebraic implications are discussed in [1]. Fokkink and Nain have shown in [18] that no congruence over the language BCCSP, a basic formalism to express finite process behaviour, that is included in possible worlds equivalence, and includes ready trace equivalence, affords a finite ω-complete equational axiomatization.

The paper is organized as follows. We begin by presenting the language BPA with the interrupt operator, its operational semantics and preliminaries on equational logic in Section 2. *Ibidem* we also show that the interrupt operator is not definable in BPA modulo bisimilarity. The general structure of the proof of our main result, to the effect that bisimilarity is not finitely based over the language we consider in this paper, is presented in Section 3. There we also show how to reduce the proof of our main result to that of a technical statement describing a key property of closed instantiations of sound equations that is preserved under equational derivations (Proposition 3.2). We conclude the paper by offering a proof of Proposition 3.2 in Section 4.

2 Preliminaries

We begin by introducing the basic definitions and results on which the technical developments to follow are based. The interested reader is referred to [6, 11] for

more information.

2.1 The Language $\mathrm{BPA}_{\mathbf{int}}$

We assume a non-empty alphabet A of atomic actions, with typical elements a, b. The language for processes we shall consider in this paper, henceforth referred to as $\mathrm{BPA}_{\mathrm{int}}$, is obtained by adding the interrupt operator from [6] to Bergstra and Klop's BPA [11]. This language is given by the following grammar:

$$t ::= x \mid a \mid t \cdot t \mid t + t \mid t \rhd t \ ,$$

where x is a variable drawn from a countably infinite set V and a is an action. In the above grammar, we use the symbol $\rhd$ for the *interrupt operator*. We shall use the meta-variables t, u, v, w to range over process terms, and write $var(t)$ for the collection of variables occurring in the term t. The *size* of a term is the number of operator symbols in it. A process term is *closed* if it does not contain any variables. Closed terms will be typically denoted by p, q, r, s. As usual, we shall often write tu in lieu of $t \cdot u$, and we assume that $\cdot$ binds stronger than $+$.

A (closed) substitution is a mapping from process variables to (closed) $\mathrm{BPA}_{\mathrm{int}}$ terms. For every term t and (closed) substitution σ, the (closed) term obtained by replacing every occurrence of a variable x in t with the (closed) term $\sigma(x)$ will be written $\sigma(t)$. In what follows, we shall use the notation $\sigma[x \mapsto p]$, where σ is a closed substitution and p is a closed $\mathrm{BPA}_{\mathrm{int}}$ term, to stand for the substitution mapping x to p, and acting like σ on all of the other variables in V.

In the remainder of this paper, we let a^1 denote a, and a^{m+1} denote $a(a^m)$, and terms are considered modulo associativity and commutativity of $+$. In other words, we do not distinguish $t + u$ and $u + t$, nor $(t + u) + v$ and $t + (u + v)$. This is justified because $+$ is associative and commutative with respect to the notion of equivalence we shall consider over $\mathrm{BPA}_{\mathrm{int}}$. (See axioms A1, A2 in Table 3 on page 11.) In what follows, the symbol $=$ will denote equality modulo associativity and commutativity of $+$.

We say that a term t *has* $+$ *as head operator* if $t = t_1 + t_2$ for some terms t_1 and t_2. For example, $a + b$ has $+$ as head operator, but $(a + b)a$ does not.

For $k \geq 1$, we use a *summation* $\sum_{i \in \{1,\ldots,k\}} t_i$ to denote $t_1 + \cdots + t_k$. It is easy to see that every $\mathrm{BPA}_{\mathrm{int}}$ term t has the form $\sum_{i \in I} t_i$, for some finite, non-empty index set I, and terms t_i ($i \in I$) that do not have $+$ as head operator. The terms t_i ($i \in I$) will be referred to as the *(syntactic) summands* of t. For example, the term $(a + b)a$ has only itself as (syntactic) summand.

The following observation, whose simple proof is omitted, will find application in the subsequent technical developments.

Lemma 2.1 Let t be a BPA_{int} term, and let σ be a substitution. Assume that t is neither a variable nor a term of the form $t_1 + t_2$ for some t_1, t_2. Then t and $\sigma(t)$ have the same number of summands.

The operational semantics for the language BPA_{int} is given by the labelled transition system

$$\left(\text{BPA}_{\text{int}}, \left\{\xrightarrow{a} \mid a \in A\right\}, \left\{\xrightarrow{a}\checkmark \mid a \in A\right\}\right) \ ,$$

where the transition relations $\xrightarrow{a}$ and the unary predicates $\xrightarrow{a}\checkmark$ are, respectively, the least subsets of $\text{BPA}_{\text{int}} \times \text{BPA}_{\text{int}}$ and BPA_{int} satisfying the rules in Table 1. Intuitively, a transition $t \xrightarrow{a} u$ means that the system represented by the term t can perform the action a, thereby evolving into u. The special symbol $\checkmark$ stands for (successful) termination; therefore the interpretation of the statement $t \xrightarrow{a}\checkmark$ is that the process term t can terminate by performing a. Note that, for every closed term p, there is some action a for which either $p \xrightarrow{a} p'$ holds for some p', or $p \xrightarrow{a}\checkmark$ does.

For terms t, u, and action a, we say that u is an *a-derivative* of t if $t \xrightarrow{a} u$.

$$\frac{}{a \xrightarrow{a}\checkmark} \qquad \frac{t \xrightarrow{a}\checkmark}{t + u \xrightarrow{a}\checkmark} \qquad \frac{u \xrightarrow{a}\checkmark}{t + u \xrightarrow{a}\checkmark} \qquad \frac{t \xrightarrow{a} t'}{t + u \xrightarrow{a} t'} \qquad \frac{u \xrightarrow{a} u'}{t + u \xrightarrow{a} u'}$$

$$\frac{t \xrightarrow{a}\checkmark}{t \cdot u \xrightarrow{a} u} \qquad \frac{t \xrightarrow{a} t'}{t \cdot u \xrightarrow{a} t' \cdot u}$$

$$\frac{t \xrightarrow{a}\checkmark}{t \rhd u \xrightarrow{a}\checkmark} \qquad \frac{t \xrightarrow{a} t'}{t \rhd u \xrightarrow{a} t' \rhd u} \qquad \frac{u \xrightarrow{a}\checkmark}{t \rhd u \xrightarrow{a} t} \qquad \frac{u \xrightarrow{a} u'}{t \rhd u \xrightarrow{a} u' \cdot t}$$

Table 1: Transition Rules for BPA_{int}

The transition relations $\xrightarrow{a}$ naturally compose to determine the possible effects that performing a sequence of actions may have on a BPA_{int} term.

Definition 2.1 For a sequence of actions $a_1 \cdots a_k$ ($k \geq 0$), and BPA_{int} terms t, t', we write $t \xrightarrow{a_1 \cdots a_k} t'$ iff there exists a sequence of transitions

$$t = t_0 \xrightarrow{a_1} t_1 \xrightarrow{a_2} \cdots \xrightarrow{a_k} t_k = t' \ .$$

Similarly, we say that $a_1 \cdots a_k$ ($k \geq 1$) is a termination trace of a BPA_{int} terms t iff there exists a sequence of transitions

$$t = t_0 \xrightarrow{a_1} t_1 \xrightarrow{a_2} \cdots \xrightarrow{a_k}\checkmark \ .$$

If $t \stackrel{a_1 \cdots a_k}{\longrightarrow} t'$ holds for some BPA$_{\rm int}$ term t', or $a_1 \cdots a_k$ is a termination trace of t, then $a_1 \cdots a_k$ is a *trace* of t.

The *depth* of a term t, written $depth(t)$, is the length of the longest trace it affords.

The *norm* of a term t, denoted by $norm(t)$, is the length of its shortest termination trace; this notion stems from [8].

The depth and the norm of closed terms can also be characterized inductively thus:

$$\begin{array}{rcl} depth(a) & = & 1 \\ depth(p+q) & = & \max\{depth(p), depth(q)\} \\ depth(pq) & = & depth(p) + depth(q) \\ depth(p \rhd q) & = & depth(p) + depth(q) \end{array}$$

$$\begin{array}{rcl} norm(a) & = & 1 \\ norm(p+q) & = & \min\{norm(p), norm(q)\} \\ norm(pq) & = & norm(p) + norm(q) \\ norm(p \rhd q) & = & norm(p) \ . \end{array}$$

Note that the depth and the norm of each closed BPA$_{\rm int}$ term are positive.

In what follows, we shall sometimes need to consider the possible origins of a transition of the form $\sigma(t) \stackrel{a}{\rightarrow} p$, for some action a, closed substitution σ, BPA$_{\rm int}$ term t and closed term p. Naturally enough, we expect that $\sigma(t)$ affords that transition if $t \stackrel{a}{\rightarrow} t'$, for some t' such that $p = \sigma(t')$. However, the above transition may also derive from the initial behaviour of some closed term $\sigma(x)$, provided that the collection of initial moves of $\sigma(t)$ depends, in some formal sense, on that of the closed term substituted for the variable x. Similarly, we shall sometimes need to consider the possible origins of a transition of the form $\sigma(t) \stackrel{a}{\rightarrow}\surd$, for some action a, closed substitution σ and BPA$_{\rm int}$ term t.

To fully describe these situations, we introduce the auxiliary notion of configuration of a BPA$_{\rm int}$ term. To this end, we assume a set of symbols

$$V_d = \{x_d \mid x \in V\}$$

disjoint from V. Intuitively, the symbol x_d (read "during x") will be used to denote that the closed term substituted for variable x has begun executing, but has not yet terminated.

Definition 2.2 The collection of BPA$_{int}$ *configurations* is given by the following grammar:

$$c ::= t \mid x_d \mid c \cdot t \mid c \rhd t \ ,$$

where t is a BPA$_{\rm int}$ term, and $x_d \in V_d$.

$$\frac{}{x \xrightarrow{x_s} x_d} \qquad \frac{}{x \xrightarrow{x} \surd}$$

$$\frac{t \xrightarrow{x} t'}{t + u \xrightarrow{x} t'} \qquad \frac{t \xrightarrow{x_s} c}{t + u \xrightarrow{x_s} c} \qquad \frac{t \xrightarrow{x} \surd}{t + u \xrightarrow{x} \surd}$$

$$\frac{u \xrightarrow{x} u'}{t + u \xrightarrow{x} u'} \qquad \frac{u \xrightarrow{x_s} c}{t + u \xrightarrow{x_s} c} \qquad \frac{u \xrightarrow{x} \surd}{t + u \xrightarrow{x} \surd}$$

$$\frac{t \xrightarrow{x} t'}{tu \xrightarrow{x} t'u} \qquad \frac{t \xrightarrow{x_s} c}{tu \xrightarrow{x_s} cu} \qquad \frac{t \xrightarrow{x} \surd}{tu \xrightarrow{x} u}$$

$$\frac{t \xrightarrow{x} t'}{t \rhd u \xrightarrow{x} t' \rhd u} \qquad \frac{t \xrightarrow{x_s} c}{t \rhd u \xrightarrow{x_s} c \rhd u} \qquad \frac{t \xrightarrow{x} \surd}{t \rhd u \xrightarrow{x} \surd}$$

$$\frac{u \xrightarrow{x} u'}{t \rhd u \xrightarrow{x} u't} \qquad \frac{u \xrightarrow{x_s} c}{t \rhd u \xrightarrow{x_s} ct} \qquad \frac{u \xrightarrow{x} \surd}{t \rhd u \xrightarrow{x} t}$$

Table 2: SOS Rules for the Auxiliary Transitions $\xrightarrow{x}$, $\xrightarrow{x_s}$ and $\xrightarrow{x}\surd$ ($x \in V$)

For example, the configuration $x_d \cdot (a \rhd x)$ is meant to describe a state of the computation of some term in which the (closed term substituted for the) occurrence of variable x on the left-hand side of the $\cdot$ operator has begun its execution (and has not terminated), but the one on the right-hand side has not. Note that each configuration contains at most one occurrence of an $x_d \in V_d$.

We shall consider the symbols x_d as variables, and use the notation $\sigma[x_d \mapsto p]$, where σ is a closed substitution and p is a closed BPA_{int} term, to stand for the substitution mapping x_d to p, and acting like σ on all of the other variables.

The way in which the initial behaviour of a term may depend on that of the variables that occur in it is formally described by three auxiliary transition relations whose elements have the following forms:

- $t \xrightarrow{x_s} c$ (read "t can start executing x and become c in doing so"), where t is a term, x is a variable, and c is a configuration,
- $t \xrightarrow{x} t'$, where t and t' are terms and x is a variable, or
- $t \xrightarrow{x}\surd$, where t is a term.

The first of these types of transitions will be used to account for those transitions of the form $\sigma(t) \xrightarrow{a} p$ that are due to a-labelled transitions of the closed term $\sigma(x)$

that do not lead to its termination. The second will describe the origin of transitions of the form $\sigma(t) \stackrel{a}{\rightarrow} \sigma(t')$ that are due to a-labelled transitions of the closed term $\sigma(x)$ that lead to its termination. Finally, transitions of the third kind will allow us to describe the origin of termination transitions of the form $\sigma(t) \stackrel{a}{\rightarrow}\surd$ that are due to a-labelled termination transitions of the closed term $\sigma(x)$.

The SOS rules defining these transitions are given in Table 2. In those rules, the meta-variables t, u, t' and u' denote $\mathrm{BPA}_{\mathrm{int}}$ terms, and c ranges over the collection of configurations that contain one occurrence of a symbol of the form x_d. The attentive reader might have already noticed that the left-hand sides of the rules in Table 2 are always $\mathrm{BPA}_{\mathrm{int}}$ terms, and therefore that no (auxiliary) transitions are possible from configurations that contain one occurrence of a symbol of the form x_d. This is in line with our aim in defining the auxiliary transition relations $\stackrel{x}{\rightarrow}$, $\stackrel{x_s}{\rightarrow}$ and $\stackrel{x}{\rightarrow}\surd$ ($x \in V$), viz. to describe the possible origins of the *initial* transitions of a term of the form $\sigma(t)$, with t a $\mathrm{BPA}_{\mathrm{int}}$ term and σ a closed substitution.

Lemma 2.2 For each $\mathrm{BPA}_{\mathrm{int}}$ term t, configuration c and variable x, if $t \stackrel{x_s}{\rightarrow} c$, then x_d occurs in c. Moreover, if $c = x_d$ then x is a summand of t.

The precise connection between the transitions of a term $\sigma(t)$ and those of t is expressed by the following lemma.

Lemma 2.3 [Operational Correspondence] Assume that t is a $\mathrm{BPA}_{\mathrm{int}}$ term, σ is a closed substitution and a is an action. Then the following statements hold:

1. If $t \stackrel{a}{\rightarrow}\surd$, then $\sigma(t) \stackrel{a}{\rightarrow}\surd$.
2. If $t \stackrel{x}{\rightarrow}\surd$ and $\sigma(x) \stackrel{a}{\rightarrow}\surd$, then $\sigma(t) \stackrel{a}{\rightarrow}\surd$.
3. If $t \stackrel{x}{\rightarrow} t'$ and $\sigma(x) \stackrel{a}{\rightarrow}\surd$, then $\sigma(t) \stackrel{a}{\rightarrow} \sigma(t')$.
4. Assume that $t \stackrel{x_s}{\rightarrow} c$ and $\sigma(x) \stackrel{a}{\rightarrow} p$, for some closed term p. Then $\sigma(t) \stackrel{a}{\rightarrow} \sigma[x_d \mapsto p](c)$.
5. If $t \stackrel{a}{\rightarrow} t'$, then $\sigma(t) \stackrel{a}{\rightarrow} \sigma(t')$.
6. Assume that $\sigma(t) \stackrel{a}{\rightarrow}\surd$. Then either $t \stackrel{a}{\rightarrow}\surd$ or there is a variable x such that $t \stackrel{x}{\rightarrow}\surd$ and $\sigma(x) \stackrel{a}{\rightarrow}\surd$.
7. Assume that $\sigma(t) \stackrel{a}{\rightarrow} p$, for some closed term p. Then one of the following possibilities applies:
   - $t \stackrel{x}{\rightarrow} t'$, $\sigma(x) \stackrel{a}{\rightarrow}\surd$ and $p = \sigma(t')$, for some term t' and variable x,
   - $t \stackrel{a}{\rightarrow} t'$ for some t' such that $p = \sigma(t')$, or

- $t \stackrel{x_s}{\rightarrow} c$ and $\sigma(x) \stackrel{a}{\rightarrow} q$, for some variable x, configuration c and closed term q such that $\sigma[x_d \mapsto q](c) = p$.

Proof: Statements 1–5 are proven by induction on the proof of the relevant transitions. The proof of statement 3 uses statement 2. On the other hand, statements 6–7 are proven by induction on the structure of the term t. The proof of statement 7 uses statement 6.

The details are lengthy, but straightforward, and we therefore omit them. □

In this paper, we shall consider the language BPA_{int} modulo bisimulation equivalence [28].

Definition 2.3 Two closed BPA_{int} terms p and q are *bisimilar*, denoted by $p \underline{\leftrightarrow} q$, if there exists a symmetric binary relation $\mathcal{B}$ over closed BPA_{int} terms which relates p and q, such that:

- if $r \mathcal{B} s$ and $r \stackrel{a}{\rightarrow} r'$, then there is a transition $s \stackrel{a}{\rightarrow} s'$ such that $r' \mathcal{B} s'$,
- if $r \mathcal{B} s$ and $r \stackrel{a}{\rightarrow}\surd$, then $s \stackrel{a}{\rightarrow}\surd$.

Such a relation $\mathcal{B}$ will be called a *bisimulation*. The relation $\underline{\leftrightarrow}$ will be referred to as *bisimulation equivalence* or *bisimilarity*.

It is well known that $\underline{\leftrightarrow}$ is an equivalence relation [28]. Moreover, the transition rules in Table 1 are in the 'path' format of Baeten and Verhoef [9]. Hence, bisimulation equivalence is a congruence with respect to all the operators in the signature of BPA_{int}.

Note that bisimilar closed BPA_{int} terms afford the same finite non-empty collection of (termination) traces, and therefore have the same norm and depth.

Bisimulation equivalence is extended to arbitrary BPA_{int} terms thus:

Definition 2.4 Let t, u be BPA_{int} terms. Then $t \underline{\leftrightarrow} u$ iff $\sigma(t) \underline{\leftrightarrow} \sigma(u)$ for every closed substitution σ.

For instance, we have that

$$x \rhd y \underline{\leftrightarrow} (x \rhd y) + yx$$

because, as our readers can easily check, the terms $p \rhd q$ and $(p \rhd q) + qp$ have the same set of initial "capabilities", i.e.,

$$\begin{array}{c} p \rhd q \stackrel{a}{\rightarrow} r \text{ iff } (p \rhd q) + qp \stackrel{a}{\rightarrow} r \ , \text{ for each } a \text{ and } r, \text{ and} \\ p \rhd q \stackrel{a}{\rightarrow}\surd \text{ iff } (p \rhd q) + qp \stackrel{a}{\rightarrow}\surd, \text{ for each } a \ . \end{array}$$

It is natural to expect that the interrupt operator cannot be defined in the language BPA modulo bisimulation equivalence. This expectation is confirmed by the following simple, but instructive, result:

Proposition 2.1 There is no $\mathrm{BPA}_{\mathrm{int}}$ term t such that t does not contain occurrences of the interrupt operator, and $t \leftrightarrow x \rhd y$.

Proof: Assume, towards a contradiction, that t is a $\mathrm{BPA}_{\mathrm{int}}$ term such that t does not contain occurrences of the interrupt operator, and $t \leftrightarrow x \rhd y$.

Consider the closed substitution σ_a mapping each variable to a. Since

$$\sigma_a(t) \leftrightarrow a \rhd a \text{ and } a \rhd a \xrightarrow{a} \surd \ ,$$

we have that $\sigma_a(t) \xrightarrow{a} \surd$. Lemma 2.3(6) yields that either $t \xrightarrow{a} \surd$ or there is a variable z such that $t \xrightarrow{z} \surd$ and $\sigma_a(z) \xrightarrow{a} \surd$. We shall now argue that both of these possibilities imply that $t \not\leftrightarrow x \rhd y$, contradicting our assumption.

Indeed, using the former possibility we may infer that $\sigma_a[x \mapsto a^2](t) \xrightarrow{a} \surd$ (Lemma 2.3(1)). This implies that $t \not\leftrightarrow x \rhd y$, because $a^2 \rhd a$ does not have termination traces of length 1.

Assume now there is a variable z such that $t \xrightarrow{z} \surd$ and $\sigma_a(z) \xrightarrow{a} \surd$. It is not hard to see that $t \leftrightarrow z + u$ for some term u, since t does not contain occurrences of the interrupt operator and $t \xrightarrow{z} \surd$. We claim that

$$\sigma_a[x \mapsto a^2](t) \not\leftrightarrow a^2 \rhd a \ .$$

If $z \neq x$, our claim follows, because, reasoning as above,

$$\sigma_a[x \mapsto a^2](t) \leftrightarrow a + \sigma_a[x \mapsto a^2](u) \xrightarrow{a} \surd$$

whereas $a^2 \rhd a$ does not have termination traces of length 1.

If $t \leftrightarrow x + u$, then $\sigma_a[x \mapsto a^2](t) \xrightarrow{a} p$ for some $p \leftrightarrow a$. On the other hand, the two a-derivatives of $a^2 \rhd a$, namely $a \rhd a$ and a^2, have depth 2, and thus neither of them is bisimilar to a. □

2.2 Equational Logic

An *axiom system* is a collection of equations $t \approx u$ over the language $\mathrm{BPA}_{\mathrm{int}}$. An equation $t \approx u$ is derivable from an axiom system E, notation $E \vdash t \approx u$, if it can be proven from the axioms in E using the rules of equational logic (viz. reflexivity, symmetry, transitivity, substitution and closure under $\mathrm{BPA}_{\mathrm{int}}$ contexts):

$$t \approx t \quad \frac{t \approx u}{u \approx t} \quad \frac{t \approx u \ \ u \approx v}{t \approx v} \quad \frac{t \approx u}{\sigma(t) \approx \sigma(u)}$$

$$\frac{t \approx u \ \ t' \approx u'}{t + t' \approx u + u'} \quad \frac{t \approx u \ \ t' \approx u'}{tt' \approx uu'} \quad \frac{t \approx u \ \ t' \approx u'}{t \rhd t' \approx u \rhd u'} \ .$$

A1	$x+y$	$\approx$	$y+x$
A2	$(x+y)+z$	$\approx$	$x+(y+z)$
A3	$x+x$	$\approx$	x
A4	$(x+y)z$	$\approx$	$(xz)+(yz)$
A5	$(xy)z$	$\approx$	$x(yz)$

Table 3: Some Axioms for $\mathrm{BPA}_{\mathrm{int}}$

Without loss of generality one may assume that substitutions happen first in equational proofs, i.e., that the rule

$$\frac{t \approx u}{\sigma(t) \approx \sigma(u)}$$

may only be used when $(t \approx u) \in E$. In this case $\sigma(t) \approx \sigma(u)$ is called a *substitution instance* of an axiom in E.

Moreover, by postulating that for each axiom in E also its symmetric counterpart is present in E, one may assume that applications of symmetry happen first in equational proofs. In the remainder of this paper, we shall tacitly assume that our equational axiom systems are closed with respect to symmetry.

It is well-known (cf., e.g., Sect. 2 in [20]) that if an equation relating two closed terms can be proven from an axiom system E, then there is a closed proof for it.

Definition 2.5 An equation $t \approx u$ over the language $\mathrm{BPA}_{\mathrm{int}}$ is *sound* with respect to $\underline{\leftrightarrow}$ iff $t \underline{\leftrightarrow} u$. An axiom system is sound with respect to $\underline{\leftrightarrow}$ iff so is each of its equations.

An example of a collection of equations over the language $\mathrm{BPA}_{\mathrm{int}}$ that are sound with respect to $\underline{\leftrightarrow}$ is given in Table 3. Those equations stem from [11]. Equations dealing with the interrupt operator using two auxiliary operators are offered in [6].

3 Bisimilarity is not Finitely Based over $\mathrm{BPA}_{\mathbf{int}}$

Our order of business in the remainder of this paper will be to show the following theorem:

Theorem 3.1 Bisimilarity is not finitely based over the language $\mathrm{BPA}_{\mathrm{int}}$—that is, there is no finite axiom system that is sound with respect to $\underline{\leftrightarrow}$, and proves all of

the equations $t \approx u$ such that $t \leftrightarrow u$. Moreover, the same holds true if we restrict ourselves to the collection of closed equations over $\mathrm{BPA}_{\mathrm{int}}$ that hold modulo $\leftrightarrow$.

The above theorem is an immediate corollary of the following result:

Theorem 3.2 Let E be a finite collection of equations over the language $\mathrm{BPA}_{\mathrm{int}}$ that hold modulo $\leftrightarrow$. Let $n > 2$ be larger than the size of each term in the equations in E. Then $E \nvdash e_n$, where the family of equations e_n ($n \geq 1$) is defined thus:

$$e_n : \left(\sum_{i=1}^{n} p_i\right) \rhd a \approx b + \sum_{i=2}^{n} b((b^{i-1} + b) \rhd a) + a \sum_{i=1}^{n} p_i \ . \qquad (1)$$

In the above family, $p_1 = b$ and $p_i = b(b^{i-1} + b)$ for $i > 1$.

Observe that, for each $n \geq 1$, the closed equation e_n is sound modulo bisimilarity. Indeed, the left-hand and right-hand sides of the equation have isomorphic labelled transitions systems. Therefore, as claimed above, Theorem 3.1 is an immediate consequence of Theorem 3.2.

The following simple properties of the closed terms mentioned in (1) will find repeated application in what follows.

Lemma 3.1

1. Let $n \geq 1$ and $i \in \{1, \ldots, n\}$. Then, the norm of p_i is 1 if $i = 1$, and 2 otherwise. The depth of p_i is i.
2. For each $n \geq 1$, the norm of $\left(\sum_{i=1}^{n} p_i\right) \rhd a$ is 1, and its depth is $n + 1$.

In the remainder of this study, we shall offer a proof of Theorem 3.2. In order to prove this theorem, it will be sufficient to establish the following technical result:

Proposition 3.1 Let E be a finite axiom system over the language $\mathrm{BPA}_{\mathrm{int}}$ that is sound modulo bisimilarity. Let $n > 2$ be larger than the size of each term in the equations in E. Assume, furthermore, that

- $E \vdash p \approx q$,
- $p \leftrightarrow \left(\sum_{i=1}^{n} p_i\right) \rhd a$ and
- p has a summand bisimilar to $\left(\sum_{i=1}^{n} p_i\right) \rhd a$.

Then q has a summand bisimilar to $\left(\sum_{i=1}^{n} p_i\right) \rhd a$.

Indeed, assuming Proposition 3.1, we can prove Theorem 3.2, and therefore Theorem 3.1, as follows.

Proof of Theorem 3.2: Assume that E is a finite axiom system over the language $\mathrm{BPA}_{\mathrm{int}}$ that is sound modulo bisimilarity. Pick $n > 2$ and larger than the size of the terms in the equations in E. Assume that, for some closed term q,

$$E \vdash \left(\sum_{i=1}^{n} p_i\right) \rhd a \approx q \ .$$

Using Proposition 3.1, we have that q has a summand bisimilar to $\left(\sum_{i=1}^{n} p_i\right) \rhd a$. Note now that the summands of the right-hand side of equation e_n, viz.

$$b + \sum_{i=2}^{n} b((b^{i-1} + b) \rhd a) + a \sum_{i=1}^{n} p_i \ ,$$

are the terms

- b,
- $b((b^{i-1} + b) \rhd a)$, for some $2 \leq i \leq n$, and
- $a \sum_{i=1}^{n} p_i$.

Unlike $\left(\sum_{i=1}^{n} p_i\right) \rhd a$, none of these terms can initially perform both an a and a b action. It follows that no summand of the right-hand side of equation e_n is bisimilar to $\left(\sum_{i=1}^{n} p_i\right) \rhd a$, and thus that

$$q \neq b + \sum_{i=2}^{n} b((b^{i-1} + b) \rhd a) + a \sum_{i=1}^{n} p_i \ .$$

We may therefore conclude that E does not prove equation e_n, which was to be shown. □

Our order of business will now be to provide a proof of Proposition 3.1. Our proof of that result will be proof-theoretic in nature, and will proceed by induction on the depth of equational derivations from a finite axiom system E. The crux in such an induction proof is given by the following proposition, to the effect that the statement of Proposition 3.1 holds for closed instantiations of axioms in E.

Proposition 3.2 Let $t \approx u$ be an equation over the language $\mathrm{BPA}_{\mathrm{int}}$ that holds modulo bisimilarity. Let σ be a closed substitution, $p = \sigma(t)$ and $q = \sigma(u)$. Assume that

- $n > 2$ and the size of t is smaller than n,
- $p \leftrightarrows \left(\sum_{i=1}^{n} p_i\right) \rhd a$ and

- p has a summand bisimilar to $\left(\sum_{i=1}^{n} p_i\right) \rhd a$.

Then q has a summand bisimilar to $\left(\sum_{i=1}^{n} p_i\right) \rhd a$.

Indeed, let us assume for the moment that the above result holds. Using it, we can prove Proposition 3.1 thus:

Proof of Proposition 3.1: Assume that E is a finite axiom system over the language $\mathrm{BPA}_{\mathrm{int}}$ that is sound with respect to bisimulation equivalence, and that the following hold, for some closed terms p and q and positive integer $n > 2$ that is larger than the size of each term in the equations in E:

1. $E \vdash p \approx q$,
2. $p \leftrightarrows \left(\sum_{i=1}^{n} p_i\right) \rhd a$, and
3. p has a summand bisimilar to $\left(\sum_{i=1}^{n} p_i\right) \rhd a$.

We prove that q also has a summand bisimilar to $\left(\sum_{i=1}^{n} p_i\right) \rhd a$ by induction on the depth of the closed proof of the equation $p \approx q$ from E. Recall that, without loss of generality, we may assume that applications of symmetry happen first in equational proofs (that is, E is closed with respect to symmetry).

We proceed by a case analysis on the last rule used in the proof of $p \approx q$ from E. The case of reflexivity is trivial, and that of transitivity follows immediately by using the inductive hypothesis twice. Below we only consider the other possibilities.

- CASE $E \vdash p \approx q$, BECAUSE $\sigma(t) = p$ AND $\sigma(u) = q$ FOR SOME EQUATION $(t \approx u) \in E$ AND CLOSED SUBSTITUTION σ. Since $n > 2$ is larger than the size of each term mentioned in equations in E, the claim follows by Proposition 3.2.

- CASE $E \vdash p \approx q$, BECAUSE $p = p' + p''$ AND $q = q' + q''$ FOR SOME p', q', p'', q'' SUCH THAT $E \vdash p' \approx q'$ AND $E \vdash p'' \approx q''$. Since p has a summand bisimilar to $\left(\sum_{i=1}^{n} p_i\right) \rhd a$, we have that so does either p' or p''. Assume, without loss of generality, that p' has a summand bisimilar to $\left(\sum_{i=1}^{n} p_i\right) \rhd a$. Since p is bisimilar to $\left(\sum_{i=1}^{n} p_i\right) \rhd a$, so is p'. The inductive hypothesis now yields that q' has a summand bisimilar to $\left(\sum_{i=1}^{n} p_i\right) \rhd a$. Hence, q has a summand bisimilar to $\left(\sum_{i=1}^{n} p_i\right) \rhd a$, which was to be shown.

- CASE $E \vdash p \approx q$, BECAUSE $p = p'p''$ AND $q = q'q''$ FOR SOME p', q', p'', q'' SUCH THAT $E \vdash p' \approx q'$ AND $E \vdash p'' \approx q''$. This case is vacuous. In fact, $norm(p) = 1$ by our assumption that $p \leftrightarrows \left(\sum_{i=1}^{n} p_i\right) \rhd a$, whereas the norm of a closed term of the form $p'p''$ is at least 2.

- CASE $E \vdash p \approx q$, BECAUSE $p = p' \rhd p''$ AND $q = q' \rhd q''$ FOR SOME p', q', p'', q'' SUCH THAT $E \vdash p' \approx q'$ AND $E \vdash p'' \approx q''$. The claim is immediate because p and q are their only summands, and E is sound modulo bisimilarity.

This completes the proof. □

In light of our previous discussion, all that we are left to do to complete our proof of Theorem 3.1 is to show Proposition 3.2. The remainder of this paper will be entirely devoted to a proof of that result.

4 Proof of Proposition 3.2

We begin our proof of Proposition 3.2 by stating a few auxiliary results that will find application in the technical developments to follow.

Lemma 4.1 Assume that $n > 2$ and $p \rhd q \leftrightarrows \left(\sum_{i=1}^{n} p_i\right) \rhd a$, for closed $\mathrm{BPA}_{\mathrm{int}}$ terms p and q. Then $p \leftrightarrows \sum_{i=1}^{n} p_i$ and $q \leftrightarrows a$.

Proof: Since $p \rhd q \leftrightarrows \left(\sum_{i=1}^{n} p_i\right) \rhd a$ and

$$\left(\sum_{i=1}^{n} p_i\right) \rhd a \stackrel{a}{\rightarrow} \sum_{i=1}^{n} p_i \ ,$$

there is a closed term r such that $p \rhd q \stackrel{a}{\rightarrow} r$ and $r \leftrightarrows \sum_{i=1}^{n} p_i$.

We proceed by examining the possible origins of the transition $p \rhd q \stackrel{a}{\rightarrow} r$. There are three possibilities to consider, viz.

1. $q \stackrel{a}{\rightarrow} q'$ and $r = q'p$, for some q',
2. $q \stackrel{a}{\rightarrow} \surd$ and $r = p$, or
3. $p \stackrel{a}{\rightarrow} p'$ and $r = p' \rhd q$.

The first case is impossible because the norm of $r = q'p$ is at least 2, whereas the norm of $\sum_{i=1}^{n} p_i$ is 1. This contradicts $r \leftrightarrows \sum_{i=1}^{n} p_i$.

In the second case, we have that $p \leftrightarrows \sum_{i=1}^{n} p_i$. Therefore

$$p \rhd q \leftrightarrows \left(\sum_{i=1}^{n} p_i\right) \rhd q \leftrightarrows \left(\sum_{i=1}^{n} p_i\right) \rhd a \ .$$

We claim that $q \leftrightarrows a$, which was to be shown. In fact, observe that the depth of q is 1 (Lemma 3.1(2)). Moreover, q can only perform action a, or else the terms

$(\sum_{i=1}^{n} p_i) \rhd q$ and $\left(\sum_{i=1}^{n} p_i\right) \rhd a$ would not afford the same traces. It follows that $q \leftrightarrow a$ as claimed.

Finally, assume that the third case applies. We shall show that this leads to a contradiction. Observe, first of all, that, since

$$p' \rhd q \leftrightarrow \sum_{i=1}^{n} p_i \ ,$$

b is the only action q can perform. We claim that $q \leftrightarrow b$. To see that this claim holds, assume that $q \stackrel{b}{\rightarrow} q'$ for some q'. Then

$$p' \rhd q \stackrel{b}{\rightarrow} q'p' \text{ and } norm(q'p') \geq 2 \ .$$

On the other hand, each b-derivative of the term $\sum_{i=1}^{n} p_i$ has the form $b^{j-1} + b$ for some $j \in \{2, \ldots, n\}$, and thus has norm 1. This contradicts

$$p' \rhd q \leftrightarrow \sum_{i=1}^{n} p_i \ .$$

Thus $q \leftrightarrow b$ and, using congruence of $\leftrightarrow$,

$$p' \rhd b \leftrightarrow \sum_{i=1}^{n} p_i \ . \tag{2}$$

It follows that $depth(p') = n - 1$. Since $p' \rhd b \stackrel{b}{\rightarrow} p'$, and the only b-derivative of $\sum_{i=1}^{n} p_i$ whose depth is $n - 1$ is $b^{n-1} + b$, we may infer that

$$p' \leftrightarrow b^{n-1} + b \ . \tag{3}$$

Using congruence of $\leftrightarrow$ again, together with (2)–(3), yields that

$$\left(b^{n-1} + b\right) \rhd b \leftrightarrow \sum_{i=1}^{n} p_i \ . \tag{4}$$

Since $n > 2$ by one of the assumptions of the lemma, we have that $n - 1 \neq 1$, and therefore $b(b^{n-2} + b)$ is a summand of $\sum_{i=1}^{n} p_i$. Consider now the transition

$$\sum_{i=1}^{n} p_i \stackrel{b}{\rightarrow} b^{n-2} + b \ .$$

Observe that the depth of the target of that transition is $n - 2$. It is now easy to see that no b-derivative of $\left(b^{n-1} + b\right) \rhd b$ has depth $n - 2$, contradicting (4).

The proof of the lemma is now complete. □

Remark 4.1 The proviso that n be larger than 2 in the statement of the above result is necessary. In fact, if $n = 2$ then

$$(b \rhd a) \rhd b \underline{\leftrightarrow} (b + b^2) \rhd a \underline{\leftrightarrow} (p_1 + p_2) \rhd a \ ,$$

but $b \not\underline{\leftrightarrow} a$ and $b \rhd a \not\underline{\leftrightarrow} b + b^2$.

The following observations will be used repeatedly in the proof of Proposition 3.2.

Lemma 4.2 Let t be a $\mathrm{BPA}_{\mathrm{int}}$ term, x be a variable, and σ be a closed substitution. Assume that $x \in var(t)$. Then the following statements hold:

1. $depth(\sigma(t)) \geq depth(\sigma(x))$, and
2. if $depth(\sigma(t)) = depth(\sigma(x))$, then either $t \underline{\leftrightarrow} x$ or $t \underline{\leftrightarrow} x + u$ for some $\mathrm{BPA}_{\mathrm{int}}$ term u that does not contain occurrences of x.

Proof: Both statements are shown by induction on the structure of t. Here we limit ourselves to presenting a proof for statement 2. The case $t = x$ is trivial, and those where $t = t_1 t_2$ or $t = t_1 \rhd t_2$, for some terms t_1, t_2 are vacuous, because $depth(\sigma(t))$ is larger than $depth(\sigma(x))$ for terms t of those forms. We are thus left to examine the case $t = t_1 + t_2$ for some terms t_1, t_2.

Since $x \in var(t)$, we have that either $x \in var(t_1) \cap var(t_2)$ or x occurs in exactly one of t_1 and t_2. We examine these two possibilities in turn.

Assume that $x \in var(t_1) \cap var(t_2)$. We claim that, for $i \in \{1, 2\}$,

$$depth(\sigma(x)) = depth(\sigma(t_i)) \ .$$

Indeed, by statement 1 of the lemma, we have that $depth(\sigma(x)) \leq depth(\sigma(t_i))$ for $i \in \{1, 2\}$. Moreover, for $i \in \{1, 2\}$,

$$\begin{aligned} depth(\sigma(t_i)) &\leq \max\{depth(\sigma(t_1)), depth(\sigma(t_2))\} \\ &= depth(\sigma(t_1 + t_2)) = depth(\sigma(x)) \ . \end{aligned}$$

Therefore, by the induction hypothesis, for $i \in \{1, 2\}$, we may infer that either $t_i \underline{\leftrightarrow} x$ or $t_i \underline{\leftrightarrow} x + u_i$ for some $\mathrm{BPA}_{\mathrm{int}}$ term u_i that does not contain occurrences of x.

If both $t_1 \underline{\leftrightarrow} x$ and $t_2 \underline{\leftrightarrow} x$, then $t_1 + t_2 \underline{\leftrightarrow} x$. Otherwise, $t = t_1 + t_2 \underline{\leftrightarrow} x + u$ for some $\mathrm{BPA}_{\mathrm{int}}$ term u that does not contain occurrences of x.

Assume now, without loss of generality, that $x \in var(t_1)$ and $x \notin var(t_2)$. Reasoning as above, we may apply the inductive hypothesis to t_1 to obtain that either $t_1 \underline{\leftrightarrow} x$ or $t_1 \underline{\leftrightarrow} x + u_1$ for some $\mathrm{BPA}_{\mathrm{int}}$ term u_1 that does not contain occurrences of x. In both cases, it follows that $t = t_1 + t_2 \underline{\leftrightarrow} x + u$ for some $\mathrm{BPA}_{\mathrm{int}}$ term u that does not contain occurrences of x. □

Lemma 4.3 Let $t \approx u$ be an equation over the language $\mathrm{BPA}_{\mathrm{int}}$ that is sound with respect to bisimulation equivalence. Assume that some variable x occurs as a summand in t. Then x also occurs as a summand in u.

Proof: Recall that, for some finite index set I, we can write

$$t \;=\; \sum_{i \in I} t_i \;\;,$$

where none of the t_i ($i \in I$) has $+$ as head operator. Assume that variable x occurs as a summand in t—i.e., there is an $i \in I$ with $t_i = x$. We shall argue that x also occurs as a summand in u.

Consider the substitution σ_a mapping each variable to a. As $t \approx u$ is sound with respect to bisimulation equivalence, we have that

$$\sigma_a(t) \leftrightarrow \sigma_a(u) \;\;.$$

Pick an integer m larger than the depth of $\sigma_a(t)$ and of $\sigma_a(u)$. Let σ be the substitution mapping x to the term a^{m+1} and agreeing with σ_a on all the other variables.

As $t \approx u$ is sound with respect to bisimulation equivalence, we have that

$$\sigma(t) \leftrightarrow \sigma(u) \;\;.$$

Moreover, the term $\sigma(t)$ affords the transition $\sigma(t) \stackrel{a}{\rightarrow} a^m$, for $t_i = x$ and $\sigma(x) = a^{m+1} \stackrel{a}{\rightarrow} a^m$. Hence, for some closed term p,

$$\sigma(u) \stackrel{a}{\rightarrow} p \leftrightarrow a^m \;\;.$$

By Lemma 2.3(7) and the definition of σ, we have that one of the following holds:

- $u \stackrel{y}{\rightarrow} u'$, $\sigma(y) \stackrel{a}{\rightarrow}\surd$ and $p = \sigma(u')$, for some term u' and variable $y \neq x$,
- $u \stackrel{a}{\rightarrow} u'$ for some u' such that $p = \sigma(u')$, or
- $u \stackrel{x_s}{\rightarrow} c$ for some configuration c such that $\sigma[x_d \mapsto a^m](c) = p$.

In the first two cases, we have that either $depth(p) \geq m+1$, if $x \in var(u')$, or $depth(p) < m$, otherwise. This contradicts $p \leftrightarrow a^m$. In the third case, we claim that $c = x_d$ and that x is a summand of u. In fact, x_d occurs in c (Lemma 2.2). Moreover, if $c \neq x_d$ then it is easy to see that $depth(\sigma[x_d \mapsto q](c)) > m$, again contradicting $p \leftrightarrow a^m$. Hence $c = x_d$ as claimed. Since, $u \stackrel{x_s}{\rightarrow} c = x_d$, it follows that x is a summand of u (Lemma 2.2), which was to be shown. □

We are finally in a position to conclude our technical developments by offering a proof of Proposition 3.2.

Proof of Proposition 3.2: Recall that, by the proviso of the proposition,

1. $t \approx u$ is an equation over the language $\mathrm{BPA}_{\mathrm{int}}$ that holds modulo bisimilarity,
2. $n > 2$ and the size of t is smaller than n,
3. σ is a closed substitution, $p = \sigma(t)$ and $q = \sigma(u)$,
4. $p \underline{\leftrightarrow} \left(\sum_{i=1}^{n} p_i\right) \rhd a$ and
5. p has a summand bisimilar to $(\sum_{i=1}^{n} p_i) \rhd a$.

We shall prove that q also has a summand bisimilar to $\left(\sum_{i=1}^{n} p_i\right) \rhd a$.

We can assume that, for some finite non-empty index sets I, J,

$$t = \sum_{i \in I} t_i \quad \text{and} \tag{5}$$

$$u = \sum_{j \in J} u_j \ , \tag{6}$$

where none of the t_i ($i \in I$) and u_j ($j \in J$) has $+$ as its head operator.

Since $p = \sigma(t)$ has a summand bisimilar to $\left(\sum_{i=1}^{n} p_i\right) \rhd a$, then so does $\sigma(t_i)$ for some index $i \in I$. Our aim is now to show that there is an index $j \in J$ such that $\sigma(u_j)$ has a summand bisimilar to $\left(\sum_{i=1}^{n} p_i\right) \rhd a$, proving that $q = \sigma(u)$ also has a summand bisimilar to $\left(\sum_{i=1}^{n} p_i\right) \rhd a$. This we proceed to do by a case analysis on the form t_i may have.

1. CASE $t_i = x$ FOR SOME VARIABLE x. In this case, we have that $\sigma(x)$ has a summand bisimilar to $\left(\sum_{i=1}^{n} p_i\right) \rhd a$, and t has x as a summand. As $t \approx u$ is sound with respect to bisimulation equivalence, it follows that u also has x as a summand (Lemma 4.3). Thus there is an index $j \in J$ such that $u_j = x$, and, modulo bisimulation, $\sigma(u)$ has $\left(\sum_{i=1}^{n} p_i\right) \rhd a$ as a summand, which was to be shown.

2. CASE $t_i = t't''$ FOR SOME TERMS t', t''. This case is vacuous. Indeed, note, first of all, that $\sigma(t_i) = \sigma(t')\sigma(t'')$ is its only summand. Therefore,

$$\sigma(t_i) = \sigma(t')\sigma(t'') \underline{\leftrightarrow} \left(\sum_{i=1}^{n} p_i\right) \rhd a \ .$$

   This is a contradiction because

$$norm\left(\left(\sum_{i=1}^{n} p_i\right) \rhd a\right) = 1 < 2 \leq norm(\sigma(t')\sigma(t'')) = norm(\sigma(t_i)) \ .$$

3. CASE $t_i = t' \rhd t''$ FOR SOME TERMS t', t''. The analysis of this case is the crux of the proof, and we present the argument in considerable detail.

   Since $\sigma(t_i) = \sigma(t') \rhd \sigma(t'')$ is its only summand, we have that

   $$\sigma(t_i) = \sigma(t') \rhd \sigma(t'') \underline{\leftrightarrow} \left(\sum_{i=1}^{n} p_i\right) \rhd a \ .$$

   By Lemma 4.1, this yields that

   $$\sigma(t') \quad \underline{\leftrightarrow} \quad \sum_{i=1}^{n} p_i \quad \text{and} \tag{7}$$

   $$\sigma(t'') \quad \underline{\leftrightarrow} \quad a \ . \tag{8}$$

   Now, t' can be written thus:

   $$t' = w_1 + \cdots + w_k \quad (k \geq 1) \ ,$$

   where none of the summands w_h has $+$ as head operator. Observe that, since n is larger than the size of t, we have that $k < n$. Hence, since

   $$\sigma(t') \underline{\leftrightarrow} \sum_{i=1}^{n} p_i \ ,$$

   there must be some $h \in \{1, \ldots, k\}$ such that

   $$\sigma(w_h) \underline{\leftrightarrow} p_{i_1} + \cdots + p_{i_m}$$

   for some $m > 1$ and $1 \leq i_1 < \ldots < i_m \leq n$. By Lemma 2.1, it follows that w_h can only be a variable x and thus that

   $$\sigma(x) \quad \underline{\leftrightarrow} \quad p_{i_1} + \cdots + p_{i_m} \ . \tag{9}$$

   Note that, as x is a summand of t',

   $$t' = x + t''' \ , \text{ for some term } t''' \ .$$

   Moreover, we have that $x \notin var(t'')$, or else $\sigma(t'') \not\underline{\leftrightarrow} a$, contradicting (8).

   Our order of business will now be to use the information collected so far in this case of the proof to argue that $\sigma(u)$ has a summand bisimilar to $\left(\sum_{i=1}^{n} p_i\right) \rhd a$. To this end, consider the substitution

   $$\sigma' = \sigma[x \mapsto a(\left(\textstyle\sum_{i=1}^{n} p_i\right) \rhd a)] \ .$$

We have that

$$\begin{array}{rcl}\sigma'(t_i) & = & \sigma'(t') \rhd \sigma'(t'') \\ & = & (\sigma'(x) + \sigma'(t''')) \rhd \sigma'(t'') \quad (\text{As } t' = x + t''') \\ & = & (\sigma'(x) + \sigma'(t''')) \rhd \sigma(t'') \quad (\text{As } x \notin var(t'')) \\ & \underline{\leftrightarrow} & \Big(a\big(\big(\sum_{i=1}^{n} p_i\big) \rhd a\big) + \sigma'(t''')\Big) \rhd a \quad (\text{As } \sigma(t'') \underline{\leftrightarrow} a) \ .\end{array}$$

Thus, for some p',

$$\sigma'(t_i) \stackrel{a}{\rightarrow} p' \underline{\leftrightarrow} \big(\big(\sum_{i=1}^{n} p_i\big) \rhd a\big) \rhd a \ .$$

By (5), we have that $\sigma'(t) \stackrel{a}{\rightarrow} p'$ also holds. Since $t \approx u$ is sound with respect to $\underline{\leftrightarrow}$, it follows that $\sigma'(t) \underline{\leftrightarrow} \sigma'(u)$. Hence, by (6), there exist an index $j \in J$ and a q' such that

$$\sigma'(u_j) \stackrel{a}{\rightarrow} q' \underline{\leftrightarrow} \big(\big(\sum_{i=1}^{n} p_i\big) \rhd a\big) \rhd a \ . \tag{10}$$

Recall that, by one of the assumptions of the proposition,

$$\sigma(u) \underline{\leftrightarrow} \big(\sum_{i=1}^{n} p_i\big) \rhd a \ ,$$

and thus $\sigma(u)$ has depth $n+1$. On the other hand, by (10),

$$depth(\sigma'(u_j)) \geq n + 2 \ .$$

Since σ and σ' differ only in the closed term they map variable x to, it follows that

$$x \in var(u_j) \ . \tag{11}$$

We shall now argue that $\sigma(u_j) \underline{\leftrightarrow} (\sum_{i=1}^{n} p_i) \rhd a$ by a further case analysis on the form a term u_j satisfying (10) and (11) may have.

(a) CASE $u_j = x$. This case is vacuous because

$$\sigma'(u_j) = \sigma'(x) = a(\big(\sum_{i=1}^{n} p_i\big) \rhd a) \stackrel{a}{\rightarrow} \big(\sum_{i=1}^{n} p_i\big) \rhd a = q'$$

is the only initial transition afforded by $\sigma'(u_j)$. This contradicts (10) because

$$\begin{array}{rcl} depth(q') & = & n+1 \\ & < & n+2 \\ & = & depth(((\sum_{i=1}^{n} p_i) \rhd a) \rhd a) \ . \end{array}$$

(b) CASE $u_j = u'u''$ FOR SOME TERMS u', u''. We show that this case also leads to a contradiction.

Recall that

$$\sigma'(u_j) = \sigma'(u')\sigma'(u'') \stackrel{a}{\rightarrow} q' \underline{\leftrightarrow} \left(\left(\sum_{i=1}^{n} p_i\right) \rhd a\right) \rhd a \ .$$

We proceed by a case analysis on the possible origin of this transition. There are two possibilities, viz.

i. $\sigma'(u') \stackrel{a}{\rightarrow} r$ and $q' = r\sigma'(u'')$, for some r, or
ii. $\sigma'(u') \stackrel{a}{\rightarrow} \checkmark$ and $q' = \sigma'(u'')$.

The former case is vacuous because, by (10), $norm(q') = 1$, whereas $norm(r\sigma'(u'')) \geq 2$.

In the latter case, we claim that $x \in var(u'')$. In fact, if $x \notin var(u'')$, then we obtain a contradiction thus:

$$\begin{array}{rcll} n+2 & = & depth(\sigma'(u'')) & \text{(By (10))} \\ & = & depth(\sigma(u'')) & (\text{As } x \notin var(u'')) \\ & < & depth(\sigma(u_j)) & (\text{As } u_j = u'u'') \\ & \leq & depth(\sigma(u)) & \\ & = & n+1 & \left(\text{As } \sigma(u) \underline{\leftrightarrow} \left(\sum_{i=1}^{n} p_i\right) \rhd a\right) \ . \end{array}$$

Thus $x \in var(u'')$, as claimed.

Observe now that, in light of (10), $u'' \underline{\not\leftrightarrow} x$. Indeed, if u'' were bisimilar to x, then we could infer that

$$q' = \sigma'(u'') \underline{\leftrightarrow} \sigma'(x) = a(\left(\sum_{i=1}^{n} p_i\right) \rhd a) \ .$$

Thus $q' \stackrel{b}{\not\rightarrow}$, contradicting (10). Since, by (10),

$$depth(\sigma'(x)) = n+2 = depth(q') = depth(\sigma'(u'')) \ ,$$

Lemma 4.2(2) thus yields that

$$u'' \underline{\leftrightarrow} x + u''' \ ,$$

for some u''' that does not contain x. Hence,

$$\begin{array}{rcll} q' & = & \sigma'(u'') & \\ & \underline{\leftrightarrow} & \sigma'(x) + \sigma'(u''') & \\ & = & a(\left(\sum_{i=1}^{n} p_i\right) \rhd a) + \sigma(u''') & (\text{As } x \notin var(u''')) \\ & \underline{\leftrightarrow} & \left(\left(\sum_{i=1}^{n} p_i\right) \rhd a\right) \rhd a & \text{(By (10))} \ . \end{array}$$

Since the transition

$$\left(\left(\sum_{i=1}^{n} p_i\right) \rhd a\right) \rhd a \xrightarrow{b} \left((b^{n-1} + b) \rhd a\right) \rhd a$$

can only be matched by a transition of the form

$$\sigma(u''') \xrightarrow{b} r \underline{\leftrightarrow} \left((b^{n-1} + b) \rhd a\right) \rhd a \ ,$$

for some r, we may infer that

$$depth(\sigma(u''')) \geq n + 1 \ .$$

We can finally derive a contradiction as follows:

$$\begin{aligned} n+1 &= depth(q) \\ &= depth(\sigma(u)) \\ &\geq depth(\sigma(u_j)) \\ &= depth(\sigma(u')) + depth(\sigma(u'')) \\ &= depth(\sigma(u')) + depth(\sigma(x) + \sigma(u''')) \\ &> n+1 \ . \end{aligned}$$

This completes the proof for the case $u_j = u'u''$.

(c) CASE $u_j = u' \rhd u''$ FOR SOME TERMS u', u''. This is the lengthiest sub-case of case 3 of the proof, and its analysis will occupy us for the next few pages.
Recall that, by (10),

$$\sigma'(u_j) = \sigma'(u') \rhd \sigma'(u'') \xrightarrow{a} q' \underline{\leftrightarrow} (\left(\sum_{i=1}^{n} p_i\right) \rhd a) \rhd a \ .$$

We proceed by a case analysis on the possible origin of this transition. There are three possibilities, namely

i. $\sigma'(u'') \xrightarrow{a} q''$ and $q' = q''\sigma'(u')$, for some q'',
ii. $\sigma'(u') \xrightarrow{a} q''$ and $q' = q'' \rhd \sigma'(u'')$, for some q'', or
iii. $\sigma'(u'') \xrightarrow{a} \checkmark$ and $q' = \sigma'(u')$.

We examine these sub-cases in turn.

- Case 3(c)i. This case is vacuous because, since

  $$q' \underline{\leftrightarrow} (\left(\sum_{i=1}^{n} p_i\right) \rhd a) \rhd a \ ,$$

  we have that $norm(q') = 1$. On the other-hand, the norm of a closed term of the form $q''\sigma'(u')$, for some q'', is at least 2.

- Case 3(c)ii. Note, first of all, that, since

$$q' = q'' \rhd \sigma'(u'') \leftrightarrow \underline{\leftrightarrow} \left(\left(\sum_{i=1}^{n} p_i \right) \rhd a \right) \rhd a \ ,$$

we have that $x \notin var(u'')$. In fact, if $x \in var(u'')$, then we would be able to infer that

$$\begin{array}{rcl} depth(q') & = & depth(q'') + depth(\sigma'(u'')) \\ & > & depth(\sigma'(u'')) \\ & \geq & n+2 \quad \text{(By Lemma 4.2(1))} \ , \end{array}$$

contradicting the above equivalence. Since $x \notin var(u'')$ and $x \in var(u_j)$ by (11), we may infer that

$$x \in var(u') \ . \tag{12}$$

Recall that, by the assumptions for this sub-case, $\sigma'(u') \xrightarrow{a} q''$ and $q' = q'' \rhd \sigma'(u'')$. Using Lemma 2.3(7), we have that one of the following possibilities arises:

i. $u' \xrightarrow{y} w$, $\sigma'(y) \xrightarrow{a} \checkmark$ and $q'' = \sigma'(w)$, for some term w and variable y,

ii. $u' \xrightarrow{a} w$ for some w such that $q'' = \sigma'(w)$, or

iii. $u' \xrightarrow{y_s} c$ and $\sigma'(y) \xrightarrow{a} r$, for some variable y, configuration c and closed term r such that $\sigma'[y_d \mapsto r](c) = q''$.

We consider these possibilities in turn.

The first of these cases is vacuous. In fact, using the assumptions for this case, we can derive a contradiction as follows. Note, first of all, that $y \neq x$ because $\sigma'(y) \xrightarrow{a} \checkmark$. Therefore

$$\sigma(y) = \sigma'(y) \xrightarrow{a} \checkmark \ .$$

Hence, by Lemma 2.3(3), we have that $\sigma(u') \xrightarrow{a} \sigma(w)$. So

$$\sigma(u_j) = \sigma(u') \rhd \sigma(u'') \xrightarrow{a} \sigma(w) \rhd \sigma(u'') \ .$$

Note that $depth(\sigma(w) \rhd \sigma(u'')) \leq n$. This implies that $x \in var(w)$, or else

$$q' = \sigma'(w) \rhd \sigma'(u'') = \sigma(w) \rhd \sigma(u'')$$

would have depth at most n, contradicting (10). But, since $x \in var(w)$, Lemma 4.2(1) yields that

$$depth(q') > depth(\sigma'(w)) \geq depth(\sigma'(x)) = n+2 \ ,$$

again contradicting (10).
The second case is also vacuous because, exactly as in the first case, we can show that $depth(q') \leq n$, if $x \notin var(w)$, and $depth(q') > n + 2$, otherwise. This contradicts (10).
We are therefore left to examine the third possibility. Recall, for the sake of clarity, that, for some variable y, configuration c and closed term r,

- $u' \xrightarrow{y_s} c$,
- $\sigma'(y) \xrightarrow{a} r$,
- $\sigma'[y_d \mapsto r](c) = q''$,
- $x \notin var(u'')$, and
- $q' = q'' \rhd \sigma'(u'') = q'' \rhd \sigma(u'') \underline{\leftrightarrow} \left(\left(\sum_{i=1}^{n} p_i\right) \rhd a\right) \rhd a$ by (10).

Note that $x \notin var(c)$, or else

$$depth(q') > depth(q'') \geq n + 2 \ ,$$

contradicting (10). We claim that $y = x$. To see that this does hold, assume, towards a contradiction, that $y \neq x$. Then

$$\sigma(y) = \sigma'(y) \xrightarrow{a} r \ .$$

Statement 4 in Lemma 2.3 now yields that

$$\sigma(u') \xrightarrow{a} \sigma[y_d \mapsto r](c) = \sigma'[y_d \mapsto r](c) = q'' \ .$$

(The first equality holds because $x \notin var(c)$.) Hence,

$$\sigma(u_j) \xrightarrow{a} q'' \rhd \sigma(u'') = q' \ .$$

This implies that $depth(q') \leq n$, contradicting (10).
To sum up, we have that $y = x$, $r = \left(\sum_{i=1}^{n} p_i\right) \rhd a$, and

$$q' = \sigma'[x_d \mapsto r](c) \rhd \sigma(u'') \ .$$

Since $depth(q') = n + 2$ by (10), x_d occurs in c, and $depth(r) = n + 1$, this is only possible if

- $c = x_d$ and
- $\sigma(u'') \underline{\leftrightarrow} a$.

We shall now argue that

$$\sigma(u_j) \underline{\leftrightarrow} \left(\sum_{i=1}^{n} p_i\right) \rhd a \ , \tag{13}$$

proving that $q = \sigma(u)$ has a summand bisimilar to $\left(\sum_{i=1}^{n} p_i\right) \rhd a$, which was to be shown.
In fact,

$$\begin{aligned} \sigma(u_j) &= \sigma(u') \rhd \sigma(u'') \\ &\underline{\leftrightarrow} \sigma(u') \rhd a \ . \end{aligned}$$

We claim that $\sigma(u') \underline{\leftrightarrow} \sum_{i=1}^{n} p_i$, and thus that (13) holds. Indeed, since $\sigma(u'') \stackrel{a}{\rightarrow}\checkmark$, we have that

$$\sigma(u_j) \stackrel{a}{\rightarrow} \sigma(u') \ .$$

As $\sigma(u_j)$ is a summand of $\sigma(u)$, we obtain that

$$\sigma(u) \stackrel{a}{\rightarrow} \sigma(u')$$

also holds. Recall that $\sigma(u) \underline{\leftrightarrow} \left(\sum_{i=1}^{n} p_i\right) \rhd a$. The only a-labelled transition out of $\left(\sum_{i=1}^{n} p_i\right) \rhd a$ is

$$\left(\sum_{i=1}^{n} p_i\right) \rhd a \stackrel{a}{\rightarrow} \sum_{i=1}^{n} p_i \ .$$

Therefore, $\sigma(u') \underline{\leftrightarrow} \sum_{i=1}^{n} p_i$, as claimed.
The proof for case 3(c)ii is now complete.

- Case 3(c)iii. Recall, for the sake of clarity, that
  i. $\sigma'(u'') \stackrel{a}{\rightarrow}\checkmark$ and
  ii. $q' = \sigma'(u') \underline{\leftrightarrow} ((\sum_{i=1}^{n} p_i) \rhd a) \rhd a$.

  Our order of business will be to show that, under these assumptions,

$$\sigma(u_j) \underline{\leftrightarrow} (\sum_{i=1}^{n} p_i) \rhd a \ , \tag{14}$$

  and thus that $\sigma(u)$ has a summand bisimilar to $\left(\sum_{i=1}^{n} p_i\right) \rhd a$, which was to be shown. Since $\sigma'(u'') \stackrel{a}{\rightarrow}\checkmark$, using statement 6 in Lemma 2.3 we may infer that
  – $u'' \stackrel{a}{\rightarrow}\checkmark$ or

– $u'' \stackrel{y}{\rightarrow}\checkmark$ and $\sigma'(y) \stackrel{a}{\rightarrow}\checkmark$, for some variable y.

In the latter case, as $\sigma'(x) \stackrel{a}{\rightarrow} \checkmark$ does not hold, we have that $y \neq x$, and so $\sigma(y) = \sigma'(y) \stackrel{a}{\rightarrow}\checkmark$. Using statements 1 and 2 of Lemma 2.3, we therefore have that

$$\sigma(u'') \stackrel{a}{\rightarrow}\checkmark \ .$$

This yields that $\sigma(u_j) = \sigma(u') \rhd \sigma(u'') \stackrel{a}{\rightarrow} \sigma(u')$. As $\sigma(u_j)$ is a summand of $\sigma(u)$, and $\sigma(u) \leftrightarrows \left(\sum_{i=1}^{n} p_i\right) \rhd a$, we may therefore infer as above that $\sigma(u') \leftrightarrows \sum_{i=1}^{n} p_i$. Hence

$$\sigma(u_j) \leftrightarrows \left(\sum_{i=1}^{n} p_i\right) \rhd \sigma(u'') \ .$$

This equivalence yields that $depth(\sigma(u_j)) = depth(\sigma(u)) = n + 1$, and that the depth of $\sigma(u'')$ is 1. We claim that $\sigma(u'') \leftrightarrows a$, proving that (14) holds as claimed. In fact, if $\sigma(u'') \stackrel{b}{\rightarrow}\checkmark$, then $\sigma(u_j)$ would afford the trace b^{n+1}, contradicting our assumption that $\sigma(u)$ is bisimilar to $\left(\sum_{i=1}^{n} p_i\right) \rhd a$.

This completes the proof of case 3c, and thus that of case 3.

Since we have examined all the possible forms that t_i can take, the proof of the proposition is now complete. □

References

[1] L. ACETO, Z. ÉSIK, AND A. INGOLFSDOTTIR, *On the two-variable fragment of the equational theory of the max-sum algebra of the natural numbers*, in Proceedings of the 17th International Symposium on Theoretical Aspects of Computer Science, STACS 2000 (Lille), H. Reichel and S. Tison, eds., vol. 1770 of Lecture Notes in Computer Science, Springer-Verlag, Feb. 2000, pp. 267–278.

[2] ——, *The max-plus algebra of the natural numbers has no finite equational basis*, Theoretical Comput. Sci., 293 (2003), pp. 169–188.

[3] L. ACETO, W. FOKKINK, AND A. INGOLFSDOTTIR, *A menagerie of non-finitely based process semantics over BPA*—from ready simulation to completed traces*, Mathematical Structures in Computer Science, 8 (1998), pp. 193–230.

[4] L. ACETO, W. FOKKINK, A. INGOLFSDOTTIR, AND B. LUTTIK, *CCS with Hennessy's merge has no finite equational axiomatization*, Research report RS-03-34, BRICS, Nov. 2003. To appear in *Theoretical Computer Science*.

[5] L. ACETO, W. FOKKINK, R. VAN GLABBEEK, AND A. INGOLFSDOTTIR, *Nested semantics over finite trees are equationally hard*, Information and Computation, 191 (2004), pp. 203–232.

[6] J. C. BAETEN AND J. BERGSTRA, *Mode transfer in process algebra*, Report CSR 00–01, Technische Universiteit Eindhoven, 2000. This paper is an expanded and revised version of [10].

[7] J. C. BAETEN, J. BERGSTRA, AND J. W. KLOP, *Syntax and defining equations for an interrupt mechanism in process algebra*, Fundamenta Informaticae, IX (1986), pp. 127–168.

[8] ——, *Decidability of bisimulation equivalence for processes generating context-free languages*, J. Assoc. Comput. Mach., 40 (1993), pp. 653–682.

[9] J. C. BAETEN AND C. VERHOEF, *A congruence theorem for structured operational semantics*, in Proceedings CONCUR 93, Hildesheim, Germany, E. Best, ed., vol. 715 of Lecture Notes in Computer Science, Springer-Verlag, 1993, pp. 477–492.

[10] J. BERGSTRA, *A mode transfer operator in process algebra*, Report P8808, Programming Research Group, University of Amsterdam, 1988.

[11] J. BERGSTRA AND J. W. KLOP, *Fixed point semantics in process algebras*, Report IW 206, Mathematisch Centrum, Amsterdam, 1982.

[12] ——, *Process algebra for synchronous communication*, Information and Control, 60 (1984), pp. 109–137.

[13] S. BLOM, W. FOKKINK, AND S. NAIN, *On the axiomatizability of ready traces, ready simulation and failure traces*, in Proceedings 30th Colloquium on Automata, Languages and Programming—ICALP'03, Eindhoven, J. C. Baeten, J. K. Lenstra, J. Parrow, and G. J. Woeginger, eds., vol. 2719 of Lecture Notes in Computer Science, Springer-Verlag, 2003, pp. 109–118.

[14] E. BRINKSMA, *A tutorial on LOTOS*, in Proceedings of the IFIP Workshop on Protocol Specification, Testing and Verification, M. Diaz, ed., North-Holland, 1986, pp. 73–84.

[15] J. H. CONWAY, *Regular Algebra and Finite Machines*, Mathematics Series (R. Brown and J. De Wet eds.), Chapman and Hall, London, United Kingdom, 1971.

[16] A. DSOUZA AND B. BLOOM, *On the expressive power of CCS*, in Foundations of Software Technology and Theoretical Computer Science (Bangalore, 1995), P. S. Thiagarajan, ed., vol. 1026 of Lecture Notes in Computer Science, Springer-Verlag, 1995, pp. 309–323.

[17] W. FOKKINK AND B. LUTTIK, *An omega-complete equational specification of interleaving*, in Proceedings 27th Colloquium on Automata, Languages and Programming—ICALP'00, Geneva, U. Montanari, J. Rolinn, and E. Welzl, eds., vol. 1853 of Lecture Notes in Computer Science, Springer-Verlag, July 2000, pp. 729–743.

[18] W. FOKKINK AND S. NAIN, *On finite alphabets and infinite bases: From ready pairs to possible worlds*, in Proceedings of Foundations of Software Science and Computation Structures, 7th International Conference, FOSSACS 2004, I. Walukiewicz, ed., vol. 2897, Springer-Verlag, 2004, pp. 182–194.

[19] J. L. GISCHER, *The equational theory of pomsets*, Theoretical Comput. Sci., 61 (1988), pp. 199–224.

[20] J. F. GROOTE, *A new strategy for proving ω–completeness with applications in process algebra*, in Proceedings CONCUR 90, Amsterdam, J. C. Baeten and J. W. Klop, eds., vol. 458 of Lecture Notes in Computer Science, Springer-Verlag, 1990, pp. 314–331.

[21] M. HENNESSY, *Axiomatising finite concurrent processes*, SIAM J. Comput., 17 (1988), pp. 997–1017.

[22] ISO, *Information processing systems – open systems interconnection – LOTOS – a formal description technique based on the temporal ordering of observational behaviour* ISO/TC97/SC21/N DIS8807, 1987.

[23] S. MAUW, *PSF – A Process Specification Formalism*, PhD thesis, University of Amsterdam, Dec. 1991.

[24] R. MILNER, *Communication and Concurrency*, Prentice-Hall International, Englewood Cliffs, 1989.

[25] R. MILNER, M. TOFTE, R. HARPER, AND D. MACQUEEN, *The Definition of Standard ML (Revised)*, MIT Press, 1997.

[26] F. MOLLER, *The importance of the left merge operator in process algebras*, in Proceedings 17^{th} ICALP, Warwick, M. Paterson, ed., vol. 443 of Lecture Notes in Computer Science, Springer-Verlag, July 1990, pp. 752–764.

[27] ——, *The nonexistence of finite axiomatisations for CCS congruences*, in Proceedings 5^{th} Annual Symposium on Logic in Computer Science, Philadelphia, USA, IEEE Computer Society Press, 1990, pp. 142–153.

[28] D. PARK, *Concurrency and automata on infinite sequences*, in 5^{th} GI Conference, Karlsruhe, Germany, P. Deussen, ed., vol. 104 of Lecture Notes in Computer Science, Springer-Verlag, 1981, pp. 167–183.

[29] V. REDKO, *On defining relations for the algebra of regular events*, Ukrainskii Matematicheskii Zhurnal, 16 (1964), pp. 120–126. In Russian.

[30] P. SEWELL, *Nonaxiomatisability of equivalences over finite state processes*, Annals of Pure and Applied Logic, 90 (1997), pp. 163–191.

Recent BRICS Report Series Publications

RS-04-24 **Luca Aceto, Willem Jan Fokkink, Anna Ingólfsdóttir, and Sumit Nain.** ***Bisimilarity is not Finitely Based over BPA with Interrupt.*** **October 2004. 30 pp.**

RS-04-23 **Hans Hüttel and Jiří Srba.** ***Recursion vs. Replication in Simple Cryptographic Protocols.*** **October 2004.**

RS-04-22 **Gian Luca Cattani and Glynn Winskel.** ***Profunctors, Open Maps and Bisimulation.*** **October 2004. 64 pp. To appear in** ***Mathematical Structures in Computer Science.***

RS-04-21 **Glynn Winskel and Francesco Zappa Nardelli.** ***New-HOPLA—A Higher-Order Process Language with Name Generation.*** **October 2004. 38 pp.**

RS-04-20 **Mads Sig Ager.** ***From Natural Semantics to Abstract Machines.*** **October 2004. 21 pp. Presented at the** ***International Symposium on Logic-based Program Synthesis and Transformation,*** **LOPSTR 2004, Verona, Italy, August 26–28, 2004.**

RS-04-19 **Bolette Ammitzbøll Madsen and Peter Rossmanith.** ***Maximum Exact Satisfiability: NP-completeness Proofs and Exact Algorithms.*** **October 2004. 20 pp.**

RS-04-18 **Bolette Ammitzbøll Madsen.** ***An Algorithm for Exact Satisfiability Analysed with the Number of Clauses as Parameter.*** **September 2004. 4 pp.**

RS-04-17 **Mayer Goldberg.** ***Computing Logarithms Digit-by-Digit.*** **September 2004. 6 pp.**

RS-04-16 **Karl Krukow and Andrew Twigg.** ***Distributed Approximation of Fixed-Points in Trust Structures.*** **September 2004. 25 pp.**

RS-04-15 **Jesús Fernando Almansa.** ***Full Abstraction of the UC Framework in the Probabilistic Polynomial-time Calculus ppc.*** **August 2004.**

RS-04-14 **Jesper Makholm Byskov.** ***Maker-Maker and Maker-Breaker Games are PSPACE-Complete.*** **August 2004. 5 pp.**

RS-04-13 **Jens Groth and Gorm Salomonsen.** ***Strong Privacy Protection in Electronic Voting.*** **July 2004. 12 pp. Preliminary abstract presented at Tjoa and Wagner, editors,** ***13th International Workshop on Database and Expert Systems Applications,*** **DEXA '02 Proceedings, 2002, page 436.**