BRICS RS-05-33 Aceto et al.: Bisimilarity is not Finitely Based over BPA with Interrupt

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-05-33**

ISSN 0909-0878 **October 2005**

Copyright © 2005, **Luca Aceto & Willem Jan Fokkink & Anna Ingólfsdóttir & Sumit Nain.**
BRICS, Department of Computer Science
University of Aarhus. All rights reserved.

Reproduction of all or part of this work is permitted for educational or research use on condition that this copyright notice is included in any copy.

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/05/33/`

Bisimilarity is not Finitely Based over BPA with Interrupt

Luca Aceto [c,a,*], Wan Fokkink [e,b], Anna Ingolfsdottir [c,a]
Sumit Nain [d]

[a] **BRICS** *(**B**asic **R**esearch in **C**omputer **S**cience), Centre of the Danish National Research Foundation, Department of Computer Science, Aalborg University, Fr. Bajersvej 7B, 9220 Aalborg Ø, Denmark.*

[b] *CWI, Department of Software Engineering, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands.*

[c] *Department of Computer Science, School of Science and Engineering, Reykjavík University, Ofanleiti 2, 103 Reykjavík, Iceland.*

[d] *Department of Computer Science, Mail Stop 132, Rice University, 6100 S. Main Street, Houston, TX 77005-1892, USA.*

[e] *Vrije Universiteit Amsterdam, Department of Computer Science, Section Theoretical Computer Science, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands.*

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. In sharp contrast to these results, the collection of closed equations over the language BPA enriched with the disrupt operator is proven to be finitely based.

Key words: Concurrency, process algebra, Basic Process Algebra (BPA), interrupt, disrupt, bisimulation, equational logic, complete axiomatizations, non-finitely based algebras, expressiveness
1991 MSC: 08A70, 03B45, 03C05, 68Q10, 68Q45, 68Q55, 68Q70

* Corresponding author.
Email addresses: luca@ru.is, luca@cs.aau.dk (Luca Aceto), wanf@cs.vu.nl (Wan Fokkink), annai@ru.is, annai@cs.aau.dk (Anna Ingolfsdottir), sumitnain@yahoo.com (Sumit Nain).
URLs: http://www.cs.aau.dk/~luca (Luca Aceto), http://www.cs.vu.nl/~wanf (Wan Fokkink), http://www.cs.aau.dk/~annai (Anna Ingolfsdottir).

1 Introduction

Programming and specification languages often include constructs to specify mode switches (see, e.g., [8,11,23,24,26]). 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.

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 example, LOTOS [15,23] 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 [25, pp. 192–193], and the reference [17] 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 [12], some of these extensions, and their relative expressiveness, have been discussed in the early paper [11]. That preprint of Bergstra's has later been revised and extended in [7]. There, 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 [25,29] 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 [7] relies on the use of four auxiliary operators—two per mode transfer operator. (Two of those auxiliary operators are, however, redundant since they are derived BPA operators.) 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 [13], 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 [27,28] that strong bisimulation equivalence is not finitely based over CCS [25] and PA [13] without the left merge operator. (The process algebra PA [13] 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 [7]?

We address the above natural question in this paper. In particular, we mostly 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 a single action, 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 evidence that the use of auxiliary operators in the technical developments presented in [7] 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., [3,4,14,16,20,30,31] 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 [12] do not have a finite equational axiomatization modulo bisimulation equivalence [27,28]. Recently, in collaboration with Luttik, the first three authors have shown in [5] that the process algebra obtained by adding Hennessy's merge operator from [22] to CCS does not have a finite equational axiomatization modulo bisimulation equivalence. This result is in sharp contrast with a theorem established by Fokkink and Luttik in [18] to the effect that the process algebra PA [13] 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 [19] 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.

Having established that the addition of the interrupt operator to BPA spoils finite axiomatizability modulo bisimulation equivalence, it is natural to ask ourselves whether the same holds true for the disrupt operator from [7]. Intuitively, "p disrupted by q" describes a process that normally behaves like p. However, at each point of the computation before p terminates, q can pre-empt it, and begin its execution. If this happens, p never resumes its computation.

We show that, perhaps surprisingly, in sharp contrast to the main result of the paper,

the use of auxiliary operators is *not* necessary in order to obtain a finite axiomatization of bisimulation equivalence over closed terms in the language obtained by enriching BPA with the disrupt operator. The key to this positive result is the distributivity of the disrupt operator with respect to the non-deterministic choice operator of BPA in its first argument—a property that is not afforded by the interrupt operator.

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. There 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 BPA with the interrupt operator, is presented in Section 3. In that section, 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 13). We offer a proof of Proposition 13 in Section 4. We conclude the paper by presenting in Section 5 an axiomatization of bisimulation equivalence over closed terms in the language obtained by enriching BPA with the disrupt operator from [7]. Such an axiomatization is finite in the presence of a finite set of actions, and does not employ auxiliary operators.

An extended abstract of this paper appeared as [6]. There we announced without proof our main result (namely, Theorem 9) under the assumption that the set of actions contains two distinct actions. The present version of the paper sharpens Theorem 9 in that it now applies to any non-empty set of actions, and offers the full proof of our main result (all of the material in Section 4). Moreover, Proposition 20 in the current paper is new.

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 [7,12] for more information.

2.1 The Language BPA_{int}

We assume a non-empty alphabet A of atomic actions, with typical element a. The language for processes we shall consider in the main body of this paper, henceforth referred to as $\mathrm{BPA}_{\mathrm{int}}$, is obtained by adding the interrupt operator from [7] to Bergstra and Klop's BPA [12]. 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*.

Intuitively, a term of the form $p \rhd q$ 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. An alternative composition $p+q$ nondeterministically behaves as either p or q. A sequential composition $p{\cdot}q$ first behaves as p, and upon termination of p behaves as q.

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 symbols in it. Formally,

- the size of variables and actions is 1, and
- that of $t \cdot u$, $t + u$ and $t \rhd u$ is one plus the sum of the sizes of t and u.

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 $+$ and $\rhd$.

A (closed) substitution is a mapping from process variables to (closed) BPA_{int} terms. For every term t and substitution σ, the term obtained by replacing every occurrence of a variable x in t with the term $\sigma(x)$ will be written $\sigma(t)$. Note that $\sigma(t)$ is closed, if so is σ. In what follows, we shall use the notation $\sigma[x \mapsto p]$, where σ is a closed substitution and p is a closed BPA_{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 $a(a^m)$. Moreover, we consider terms 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 BPA_{int}. (See axioms A1, A2 in Table 3 on page 12.) 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 BPA_{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 operational semantics for the language BPA_{int} is given by the labelled transition

$$\frac{}{a \xrightarrow{a} \checkmark}$$

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

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

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

Table 1
Transition Rules for BPA_{int}

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 $t \xrightarrow{a} u$.

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 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} term t iff there exists a term t' such that

$$t \xrightarrow{a_1 \cdots a_{k-1}} t' \xrightarrow{a_k}\checkmark \ .$$

If $t \xrightarrow{a_1 \cdots a_k} t'$ holds for some BPA_{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 [9].*

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

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

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

Note that the depth and the norm of each closed BPA_{int} term are positive, and therefore that the norm of each closed term of the form pq is at least 2. This simple, but useful, observation will be used repeatedly in the remainder of this study.

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_{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_{int} term t.

To fully describe these situations, we introduce the auxiliary notion of configuration of a BPA_{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 *The collection of* BPA_{int} *configurations is given by the grammar:*

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

where t is a BPA_{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 $\mathrm{BPA}_{\mathrm{int}}$ 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 $\sigma(t) \xrightarrow{a} \sigma(t')$ 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 transition relations 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 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 3 *For each* BPA_{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 4 (Operational Correspondence) *Assume that* t *is a* BPA_{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 $t \stackrel{a}{\rightarrow} t'$*, then* $\sigma(t) \stackrel{a}{\rightarrow} \sigma(t')$*.*
(3) If $t \stackrel{x}{\rightarrow}\surd$ *and* $\sigma(x) \stackrel{a}{\rightarrow}\surd$*, then* $\sigma(t) \stackrel{a}{\rightarrow}\surd$*.*
(4) If $t \stackrel{x}{\rightarrow} t'$ *and* $\sigma(x) \stackrel{a}{\rightarrow}\surd$*, then* $\sigma(t) \stackrel{a}{\rightarrow} \sigma(t')$*.*
(5) Assume that $t \stackrel{x_s}{\rightarrow} c$ *and* $\sigma(x) \stackrel{a}{\rightarrow} p$*, for some closed term* p*. Then* $\sigma(t) \stackrel{a}{\rightarrow} \sigma[x_d \mapsto p](c)$*.*
(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:*
- $t \stackrel{a}{\rightarrow} t'$ *for some term* t' *such that* $p = \sigma(t')$*,*
- $t \stackrel{x}{\rightarrow} t'$*,* $\sigma(x) \stackrel{a}{\rightarrow}\surd$ *and* $p = \sigma(t')$*, for some term* t' *and variable* x*, 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 4 uses statement 3. 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 [25,29].

Definition 5 *Two closed* BPA_{int} *terms* p *and* q *are bisimilar, denoted by* $p \leftrightarrow q$*, 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* $\leftrightarrow$ *will be referred to as bisimulation equivalence or bisimilarity.*

It is well known that $\leftrightarrow$ is an equivalence relation, and that it is the largest bisimulation [25,29]. Moreover, the transition rules in Table 1 are in the 'path' format of Baeten and Verhoef [10]. 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 the same depth.

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

Definition 6 *Let* t, u *be* BPA_{int} *terms. Then* $t \leftrightarrow u$ *iff* $\sigma(t) \leftrightarrow \sigma(u)$ *for every closed substitution* σ*.*

For instance, we have that

$$x \rhd y \leftrightarrow (x \rhd y) + yx$$

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.,

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

On the other hand, neither of the equivalences

$$(x + y) \rhd z \leftrightarrow (x \rhd z) + (y \rhd z) \quad \text{and}$$
$$x \rhd (y + z) \leftrightarrow (x \rhd y) + (x \rhd z)$$

holds. Indeed, as our readers can easily check,

$$(a + a^2) \rhd a \not\leftrightarrow (a \rhd a) + (a^2 \rhd a) \quad \text{and}$$
$$a^2 \rhd (a + a^2) \not\leftrightarrow (a^2 \rhd a) + (a^2 \rhd a^2) \ .$$

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 7 *There is no* BPA_{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

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

we have that $\sigma_a(t) \stackrel{a}{\rightarrow}\surd$. Lemma 4(6) yields that either $t \stackrel{a}{\rightarrow}\surd$ or there is a variable z such that $t \stackrel{z}{\rightarrow}\surd$ and $\sigma_a(z) \stackrel{a}{\rightarrow}\surd$. 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 and Lemma 4(1), we may infer that

$$\sigma_a[x \mapsto a^2](t) \stackrel{a}{\rightarrow}\surd \ .$$

This implies that $t \not\leftrightarrow x \rhd y$, because $a^2 \rhd a$ does not have termination traces of length 1.

Assume now that there is a variable z such that $t \stackrel{z}{\rightarrow}\surd$ and $\sigma_a(z) \stackrel{a}{\rightarrow}\surd$. 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 $t \stackrel{z}{\rightarrow}\surd$. 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,

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

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

If $t \leftrightarrow x + u$, then $\sigma_a[x \mapsto a^2](t) \stackrel{a}{\rightarrow} p$ 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

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 $\mathrm{BPA}_{\mathrm{int}}$

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):

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

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

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, the equation $\sigma(t) \approx \sigma(u)$ 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 (see, e.g., Sect. 2 in [21]) 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 8 *An equation $t \approx u$ over the language BPA_{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.*

A collection of equations over the language BPA that is sound and complete with respect to $\underline{\leftrightarrow}$ is given in Table 3. Those equations stem from [12].

In [7], Baeten and Bergstra gave a sound and complete axiomatization of bisimilarity over BPA_{δ} (the extension of BPA with a constant δ to describe "deadlock") enriched with the interrupt operator, using an auxiliary binary operator, which we

denote by H. Intuitively, $p\,H\,q$ behaves as $p \rhd q$, with the restriction that it must take its first action from p. The axioms from [7] for the interrupt operator and its help operator are given below (except for one axiom that involves δ).

$$\begin{aligned} x \rhd y &\approx (x\ H\ y) + (yx) \\ a\ H\ x &\approx a \qquad (a \in A) \\ (ax)\ H\ y &\approx a(x \rhd y) \qquad (a \in A) \\ (x+y)\ H\ z &\approx (x\ H\ z) + (y\ H\ z)\ . \end{aligned}$$

Observe that, in the presence of a finite set of actions, this collection of equations is finite. Note, furthermore, that, unlike the interrupt operator, the auxiliary operator H is distributive with respect to $+$ in its first argument. As we shall also remark in Section 5, this property is very useful for achieving a finite equational axiomatization of bisimilarity. Indeed, the absence of distributivity with respect to $+$ casts doubts as to the possibility that a finite axiom system be powerful enough to "expand" the initial behaviour of terms of the form $p \rhd q$ when the number of non-bisimilar summands in p grows sufficiently large. This observation lies at the heart of the proof of our main result in this study (Theorem 9). This we now proceed to present.

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

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

Theorem 9 *Bisimilarity is not finitely based over the language* BPA_{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* $t \underline{\leftrightarrow} u$*. Moreover, the same holds true if we restrict ourselves to the collection of closed equations over* BPA_{int} *that hold modulo* $\underline{\leftrightarrow}$*.*

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

Theorem 10 *Let* E *be a finite collection of equations over the language* BPA_{int} *that hold modulo* $\underline{\leftrightarrow}$*. Let* $n > 3$ *be larger than the size of each term in the equations in* E*. Then* $E \not\vdash e_n$*, where the family of equations* e_n *(*$n \geq 1$*) is defined thus:*

$$e_n:\ \Phi_n \rhd a \approx a + \sum_{i=2}^{n} a((a^{i-1} + a^3 + a) \rhd a) + a\Phi_n\ . \tag{1}$$

In the above family, $\Phi_n = \sum_{i=1}^{n} p_i$ *where* $p_1 = a$ *and* $p_i = a(a^{i-1} + a^3 + a)$ *for* $i > 1$*.*

Note that the term $\sum_{i=2}^{n} a((a^{i-1} + a^3 + a) \rhd a)$ is only present on the right-hand side of equation e_n if $n > 1$. Observe, furthermore, 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 9 is an immediate consequence of Theorem 10.

The following simple properties of the closed terms Φ_n for $n \geq 1$ and p_i for $1 \leq i \leq n$ will find repeated application in what follows.

Lemma 11

(1) The norm of p_i is 1 if $i = 1$, and 2 otherwise. The depth of p_i is 1 if $i = 1$, and $\max\{i, 4\}$ otherwise.
(2) The norm of $\Phi_n \rhd a$ is 1. Its depth is 2 if $n = 1$, 5 if $n = 2$ or $n = 3$, and $n + 1$ if $n > 3$.
(3) Each a-derivative of Φ_n or $\Phi_n \rhd a$ has norm 1.
(4) Assume that $1 \leq i < j$. Then $p_i \leftrightarrow p_j$ if, and only if, $i = 2$ and $j = 4$. Therefore Φ_n has $n - 1$ non-bisimilar summands if $n > 3$.

PROOF. We limit ourselves to presenting the proof of the former claim in statement (4). The latter claim in that statement is an immediate consequence of the former.

If $i = 2$ and $j = 4$, then $p_i \leftrightarrow p_j$ follows immediately from the definition of the terms p_ℓ ($\ell \geq 1$). Conversely, assume that $p_i \leftrightarrow p_j$ and $1 \leq i < j$. Observe that $i > 1$ by statement (1) of the lemma. Since $p_i \leftrightarrow p_j$, the terms p_i and p_j have the same set of terminating traces, namely

$$\{a^i, a^4, a^2\} = \{a^j, a^4, a^2\} \ .$$

Since $i < j$ by the proviso of the statement, it follows that $i, j \in \{2, 4\}$. Again using $i < j$, we derive that $i = 2$ and $j = 4$, which was to be shown. □

In the remainder of this study, we shall offer a proof of Theorem 10. In order to prove this theorem, it will be sufficient to establish the following technical result:

Proposition 12 *Let E be a finite axiom system over the language* BPA_{int} *that is sound modulo bisimilarity. Let $n > 3$ be larger than the size of each term in the equations in E. Assume, furthermore, that*

- $E \vdash p \approx q$,
- $p \leftrightarrow \Phi_n \rhd a$, *and*
- p *has a summand bisimilar to* $\Phi_n \rhd a$.

Then q has a summand bisimilar to $\Phi_n \rhd a$.

Indeed, assuming Proposition 12, we can prove Theorem 10, and therefore Theorem 9, as follows.

Proof of Theorem 10: Assume that E is a finite axiom system over the language BPA_{int} that is sound modulo bisimilarity. Pick $n > 3$ and larger than the size of the terms in the equations in E. Assume that, for some closed term q,

$$E \vdash \Phi_n \rhd a \approx q \ .$$

By Proposition 12, we have that q has a summand bisimilar to $\Phi_n \rhd a$. Using Lemma 11(2) it is easy to see that the summands of the right-hand side of equation e_n, viz.

$$a + \sum_{i=2}^{n} a((a^{i-1} + a^3 + a) \rhd a) + a\Phi_n \ ,$$

are not bisimilar to $\Phi_n \rhd a$, and thus that

$$q \neq a + \sum_{i=2}^{n} a((a^{i-1} + a^3 + a) \rhd a) + a\Phi_n \ .$$

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 12. 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 12 holds for closed instantiations of axioms in E.

Proposition 13 *Let $t \approx u$ be an equation over the language BPA_{int} that holds modulo bisimilarity. Let σ be a closed substitution, $p = \sigma(t)$ and $q = \sigma(u)$. Assume that*

- *$n > 3$ and the size of t is smaller than n,*
- *$p \leftrightarrows \Phi_n \rhd a$, and*
- *p has a summand bisimilar to $\Phi_n \rhd a$.*

Then q has a summand bisimilar to $\Phi_n \rhd a$.

Indeed, let us assume for the moment that the above result holds. Using it, we can prove Proposition 12 thus:

Proof of Proposition 12: Assume that E is a finite axiom system over the language BPA_{int} that is sound with respect to bisimulation equivalence, and that the following hold, for some closed terms p and q and positive integer $n > 3$ that is larger than the size of each term in the equations in E:

(1) $E \vdash p \approx q$,

(2) $p \underline{\leftrightarrow} \Phi_n \rhd a$, and
(3) p has a summand bisimilar to $\Phi_n \rhd a$.

We prove that q also has a summand bisimilar to $\Phi_n \rhd a$ 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 $n > 3$ is larger than the size of each term mentioned in equations in E, the claim follows by Proposition 13.
- 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 $\Phi_n \rhd a$, we have that so does either p' or p''. Assume, without loss of generality, that p' has a summand bisimilar to $\Phi_n \rhd a$. Since p is bisimilar to $\Phi_n \rhd a$, so is p'. The inductive hypothesis now yields that q' has a summand bisimilar to $\Phi_n \rhd a$. Hence, q has a summand bisimilar to $\Phi_n \rhd a$, 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 $p \underline{\leftrightarrow} \Phi_n \rhd a$, 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 9 is to show Proposition 13. The next section of this paper will be entirely devoted to a proof of that result.

4 Proof of Proposition 13

We begin our proof of Proposition 13 by stating a few auxiliary results that will find application in the technical developments to follow.

Lemma 14 *For $n > 1$, $2 \leq j \leq n$ and closed* BPA_{int} *term q, the term $\Phi_n \rhd a$ is not bisimilar to closed terms that have a summand $\Big((a^{j-1} + a^3 + a) \rhd a\Big) \rhd q$.*

PROOF. Observe that $\Big((a^{j-1}+a^3+a)\rhd a\Big)\rhd q \stackrel{a}{\to} \Big(a^2\rhd a\Big)\rhd q$. The claim now follows immediately by Lemma 11(3). $\Box$

Lemma 15 *Let $n \geq 1$. Assume that $p \rhd q \leftrightarrow \Phi_n \rhd a$, for closed* BPA$_{int}$ *terms p and q. Then $p \leftrightarrow \Phi_n$ and $q \leftrightarrow a$.*

PROOF. Since $p \rhd q \leftrightarrow \Phi_n \rhd a$ and $\Phi_n \rhd a \stackrel{a}{\to} \Phi_n$, there is a closed term r such that $p \rhd q \stackrel{a}{\to} r$ and $r \leftrightarrow \Phi_n$.

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

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

The first case is impossible because the norm of $r = q'p$ is at least 2, whereas the norm of Φ_n is 1. This contradicts $r \leftrightarrow \Phi_n$.

In the second case, we have that $p \leftrightarrow \Phi_n$. Therefore, as $\leftrightarrow$ is a congruence,

$$p \rhd q \leftrightarrow \Phi_n \rhd q \leftrightarrow \Phi_n \rhd a \ .$$

We claim that $q \leftrightarrow a$, which was to be shown. In fact, observe that the depth of q is 1. Moreover, q can only perform action a, or else the terms $\Phi_n \rhd q$ and $\Phi_n \rhd a$ would not afford the same traces. It follows that $q \leftrightarrow a$ as claimed.

Finally, assume that the third case applies. Observe, first of all, that, since

$$p' \rhd q \leftrightarrow \Phi_n \ ,$$

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

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

On the other hand, each a-derivative of the term Φ_n has norm 1 (Lemma 11(3)). This contradicts

$$p' \rhd q \leftrightarrow \Phi_n \ .$$

Thus $q \leftrightarrow a$ and, using congruence of $\leftrightarrow$ and the assumption of the statement of the lemma,

$$p \rhd a \leftrightarrow \Phi_n \rhd a \ . \tag{2}$$

If $n = 1$, then we can immediately conclude that $p \leftrightarrow a = p_1$, and we are done. Assume therefore that $n \geq 2$. Since $p \rhd a \stackrel{a}{\to} p$, we may infer from (2) that

- either $p \leftrightarrow \Phi_n$

- or $p \leftrightarrow \underline{\quad} \left(a^{j-1} + a^3 + a\right) \rhd a$ for some $j \in \{2, \ldots, n\}$.

In the former case, we are done. To complete our argument, we now show that the latter case leads to a contradiction. To this end, assume that

$$p \leftrightarrow \underline{\quad} \left(a^{j-1} + a^3 + a\right) \rhd a \ .$$

Using congruence of $\leftrightarrow \underline{\quad}$ and (2), we may derive that

$$\left(\left(a^{j-1} + a^3 + a\right) \rhd a\right) \rhd a \leftrightarrow \underline{\quad} \Phi_n \rhd a \ .$$

This contradicts Lemma 14.

The proof of the lemma is now complete. □

The following observation will find a key application in the subsequent technical developments.

Lemma 16 *Let t be a* BPA$_{int}$ *term that does not have $+$ as head operator. Assume that σ is a closed substitution, and that*

$$\sigma(t) \leftrightarrow \underline{\quad} p_{i_1} + \cdots + p_{i_m} \ ,$$

for some $m > 2$ and $1 \leq i_1 < \ldots < i_m$. Then $t = x$, for some variable x.

PROOF. Assume, towards a contradiction, that t is not a variable. We proceed by a case analysis on the possible form this term may have.

(1) CASE $t = a$. This case is vacuous because, since $m > 2$ and $1 \leq i_1 < i_m$, the depth of $p_{i_1} + \cdots + p_{i_m}$ is greater than 1.

(2) CASE $t = t't''$ FOR SOME TERMS t', t''. Then

$$\sigma(t) = \sigma(t')\sigma(t'') \leftrightarrow \underline{\quad} p_{i_1} + \cdots + p_{i_m} \ .$$

Observe, first of all, that $i_1 > 1$ and $\sigma(t') \leftrightarrow \underline{\quad} a$, for otherwise either $p_{i_1} + \cdots + p_{i_m}$ would have norm 1 or $\sigma(t')\sigma(t'')$ would have an a-derivative whose norm is at least 2, contradicting the above equivalence.

Using congruence of $\leftrightarrow \underline{\quad}$,

$$a\sigma(t'') \leftrightarrow \underline{\quad} p_{i_1} + \cdots + p_{i_m} \ .$$

It follows that $p_{i_2} \leftrightarrow \underline{\quad} p_{i_m}$. As $2 \leq i_1 < i_2 < i_m$ (for $m > 2$ by the assumption of the lemma), this contradicts Lemma 11(4).

(3) CASE $t = t' \rhd t''$ FOR SOME TERMS t', t''. Then

$$\sigma(t) = \sigma(t') \rhd \sigma(t'') \leftrightarrow \underline{\quad} p_{i_1} + \cdots + p_{i_m} \ .$$

Observe, first of all, that $\sigma(t'') \underline{\leftrightarrow} a$, for otherwise $\sigma(t') \rhd \sigma(t'')$ would have an a-derivative whose norm is at least 2, contradicting the above equivalence.

Using congruence of $\underline{\leftrightarrow}$,

$$\sigma(t') \rhd a \underline{\leftrightarrow} p_{i_1} + \cdots + p_{i_m} \ .$$

It follows that, for some $j \in \{1, \ldots, m\}$,

$$\sigma(t') \underline{\leftrightarrow} (a^{i_j - 1} + a^3 + a) \ .$$

Again using congruence of $\underline{\leftrightarrow}$, we may now infer that

$$\left(a^{i_j - 1} + a^3 + a\right) \rhd a \underline{\leftrightarrow} p_{i_1} + \cdots + p_{i_m} \ .$$

This is a contradiction because

$$\left(a^{i_j - 1} + a^3 + a\right) \rhd a \xrightarrow{a} a^2 \rhd a \text{ and } norm(a^2 \rhd a) = 2 \ ,$$

whereas each a-derivative of $p_{i_1} + \cdots + p_{i_m}$ has norm 1.

We may therefore conclude that t must be a variable, which was to be shown. □

Remark 17 *The proviso that m be larger than 2 in the statement of the above result is necessary. In fact, if $m = 2$, $i_1 = 2$ and $i_2 = 4$ then*

$$p_2 + p_4 \underline{\leftrightarrow} a(a^3 + a) \ .$$

It follows that $\sigma(ax) \underline{\leftrightarrow} p_2 + p_4$ if $\sigma(x) = a^3 + a$.

The following observations will be used repeatedly in the proof of Proposition 13.

Lemma 18 *Let t be a* BPA$_{int}$ *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 BPA$_{int}$ *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 $x \in var(t)$, 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 BPA_{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 BPA_{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 BPA_{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 BPA_{int} term u that does not contain occurrences of x. □

Lemma 19 *Let $t \approx u$ be an equation over the language BPA_{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,

$$\sigma_a(t) \underline{\leftrightarrow} \sigma_a(u) \ .$$

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

$$\sigma(t) \leftrightarrows \sigma(u) \ .$$

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

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

By Lemma 4(7) and the definition of σ, 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 can conclude that either $depth(p) \geq m+1$ if $x \in var(u')$, or $depth(p) < m$ otherwise. This contradicts $p \leftrightarrows a^m$. 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 3). Moreover, if $c \neq x_d$ then it is easy to see that $depth(\sigma[x_d \mapsto q](c)) > m$, again contradicting $p \leftrightarrows a^m$. 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 3), which was to be shown. □

We are finally in a position to conclude our technical developments by offering a proof of Proposition 13.

Proof of Proposition 13: 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) $n > 3$ and the size of t is smaller than n,
(3) σ is a closed substitution, $p = \sigma(t)$ and $q = \sigma(u)$,
(4) $p \leftrightarrows \Phi_n \rhd a$, and
(5) p has a summand bisimilar to $\Phi_n \rhd a$.

We shall prove that q also has a summand bisimilar to $\Phi_n \rhd a$.

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

$$t = \sum_{i \in I} t_i \quad \text{and} \tag{3}$$

$$u = \sum_{j \in J} u_j \ , \tag{4}$$

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 $\Phi_n \rhd a$, 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 $\Phi_n \rhd a$. 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, $\sigma(x)$ has a summand bisimilar to $\Phi_n \rhd a$, 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 19). Thus there is an index $j \in J$ such that $u_j = x$, and, modulo bisimulation, $\sigma(u)$ has $\Phi_n \rhd a$ 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,

$$\sigma(t_i) = \sigma(t')\sigma(t'') \leftrightarrows \Phi_n \rhd a \ .$$

This is a contradiction because

$$norm(\Phi_n \rhd a) = 1 < 2 \leq norm(\sigma(t')\sigma(t'')) = norm(\sigma(t_i)) \ .$$

(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

$$\sigma(t_i) = \sigma(t') \rhd \sigma(t'') \leftrightarrows \Phi_n \rhd a \ .$$

By Lemma 15, this yields that

$$\sigma(t') \leftrightarrows \Phi_n \quad \text{and} \tag{5}$$
$$\sigma(t'') \leftrightarrows a \ . \tag{6}$$

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 $2k < n - 1$. Indeed, the size of

$$t_i = t' \rhd t'' = (w_1 + \cdots + w_k) \rhd t''$$

is at least $2k + 1$ and n is larger than the size of t, and therefore of t_i.

Hence, since Φ_n has $n - 1$ inequivalent summands (Lemma 11(4)) and

$$\sigma(t') \leftrightarrows \Phi_n \ ,$$

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

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

for some $m > 2$ and $1 \leq i_1 < \ldots < i_m \leq n$. By Lemma 16, it follows that w_h can only be a variable x and thus that

$$\sigma(x) \underline{\leftrightarrow} p_{i_1} + \cdots + p_{i_m} \ . \tag{7}$$

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 (6).

Our order of business will now be to use the information collected so far in this case of the proof to argue that the term $\sigma(u)$ has a summand that is bisimilar to $\Phi_n \rhd a$. To this end, consider the substitution

$$\sigma' = \sigma[x \mapsto a(\Phi_n \rhd a)] \ .$$

We have that

$$\begin{aligned}
\sigma'(t_i) &= \sigma'(t') \rhd \sigma'(t'') \\
&= (\sigma'(x) + \sigma'(t''')) \rhd \sigma'(t'') \quad (\text{As } t' = x + t''') \\
&= (\sigma'(x) + \sigma'(t''')) \rhd \sigma(t'') \quad (\text{As } x \notin var(t'')) \\
&\underline{\leftrightarrow} (a(\Phi_n \rhd a) + \sigma'(t''')) \rhd a \quad (\text{As } \sigma(t'') \underline{\leftrightarrow} a) \ .
\end{aligned}$$

Thus, for some p',

$$\sigma'(t_i) \xrightarrow{a} p' \underline{\leftrightarrow} (\Phi_n \rhd a) \rhd a \ .$$

By (3), $\sigma'(t) \xrightarrow{a} p'$ also holds. Since $t \approx u$ is sound with respect to $\underline{\leftrightarrow}$, it follows that $\sigma'(t) \underline{\leftrightarrow} \sigma'(u)$. Hence, by (4), there exist an index $j \in J$ and a q' such that

$$\sigma'(u_j) \xrightarrow{a} q' \underline{\leftrightarrow} (\Phi_n \rhd a) \rhd a \ . \tag{8}$$

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

$$\sigma(u) \underline{\leftrightarrow} \Phi_n \rhd a \ ,$$

and thus $\sigma(u)$ has depth $n+1$ because $n > 3$ (Lemma 11(2)). On the other hand, by (8),

$$depth(\sigma'(u_j)) \geq n + 3 \ .$$

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

$$x \in var(u_j) \ . \tag{9}$$

We shall now argue that $\sigma(u_j) \underline{\leftrightarrow} \Phi_n \rhd a$ by a further case analysis on the form a term u_j satisfying (8) and (9) may have.

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

$$\sigma'(u_j) = \sigma'(x) = a(\Phi_n \rhd a) \xrightarrow{a} \Phi_n \rhd a$$

is the only initial transition afforded by $\sigma'(u_j)$. Clearly this contradicts (8).

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

Recall that

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

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

(i) $\sigma'(u') \xrightarrow{a} r$ and $q' = r\sigma'(u'')$, for some r, or

(ii) $\sigma'(u') \xrightarrow{a} \checkmark$ and $q' = \sigma'(u'')$.

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

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

$$\begin{aligned} n+2 &= depth(\sigma'(u'')) \quad \text{(By (8))} \\ &= depth(\sigma(u'')) \quad (\text{As } x \notin var(u'')) \\ &< depth(\sigma(u_j)) \quad (\text{As } u_j = u'u'') \\ &\leq depth(\sigma(u)) \\ &= n+1 \quad (\text{As } \sigma(u) \underline{\leftrightarrow} \Phi_n \rhd a \text{ and } n > 3) \ . \end{aligned}$$

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

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

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

$$q' = \sigma'(u'') \underline{\leftrightarrow} \sigma'(x) = a(\Phi_n \rhd a) \ .$$

This contradicts (8) because $norm(q') = 1$, whereas $norm(\sigma'(x)) = 2$. Lemma 18(2) thus yields that

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

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

$$\begin{aligned} q' &= \sigma'(u'') \\ &\underline{\leftrightarrow} \sigma'(x) + \sigma'(u''') \\ &= a(\Phi_n \rhd a) + \sigma(u''') \quad (\text{As } x \notin var(u''')) \\ &\underline{\leftrightarrow} (\Phi_n \rhd a) \rhd a \quad \text{(By (8))} \ . \end{aligned}$$

Since the transition

$$(\Phi_n \rhd a) \rhd a \xrightarrow{a} \Big((a^{n-1} + a^3 + a) \rhd a\Big) \rhd a$$

can only be matched by a transition of the form

$$\sigma(u''') \xrightarrow{a} r \underline{\leftrightarrow} \Big((a^{n-1} + a^3 + a) \rhd a\Big) \rhd a \ ,$$

for some r (Lemma 14), and $n > 3$ by one of the assumptions of the proposition, we may infer that

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

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 (8),

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

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 3c.i. This case is vacuous because, since by (8), $norm(q') = 1$. On the other hand, the norm of $q''\sigma'(u')$ is at least 2.
- Case 3c.ii. Note, first of all, that, since

  $$q' = q'' \rhd \sigma'(u'') \underline{\leftrightarrow} (\Phi_n \rhd a) \rhd a \ ,$$

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

  $$\begin{aligned} depth(q') &= depth(q'') + depth(\sigma'(u'')) \\ &> depth(\sigma'(u'')) \\ &\geq n+2 \quad \text{(By Lemma 18(1))} \ , \end{aligned}$$

  contradicting the above equivalence. Since $x \notin var(u'')$ and $x \in$

$var(u_j)$ by (9), we may infer that

$$x \in var(u') \ . \tag{10}$$

Recall that, by the assumptions for this sub-case, $\sigma'(u') \xrightarrow{a} q''$. Using Lemma 4(7), 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 4(4), $\sigma(u') \xrightarrow{a} \sigma(w)$. So

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

$depth(\sigma(u_j)) \leq depth(\sigma(u)) = n + 1$, so $depth(\sigma(w) \rhd \sigma(u'')) \leq n$. This implies that $x \in var(w)$. For else, by assumptions of this sub-case,

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

Then q' would have depth at most n, contradicting (8). But, as $x \in var(w)$, Lemma 18(1) yields that

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

again contradicting (8).

The second case is also vacuous because, exactly as in the first case, we can show that $depth(q')$ is no larger than n if $x \notin var(w)$, and is larger than $n + 2$ otherwise. This contradicts (8).

We are therefore left to examine the third possibility. Note that $x \notin var(c)$, or else

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

contradicting (8). We claim that $y = x$. To see that this does hold, assume, towards a contradiction, that $y \neq x$. Then, by the assumptions for this sub-case,

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

Lemma 4(5) and $u' \stackrel{y_s}{\rightarrow} c$ now yield that

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

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

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

As $depth(\sigma(u_j)) \leq depth(\sigma(u)) = n+1$, this implies that $depth(q')$ is no larger than n, contradicting (8). Hence $y = x$ as claimed.

Since $\sigma'(x) \stackrel{a}{\rightarrow} r$, it follows that $r = \Phi_n \rhd a$. By one of the assumptions for this sub-case, and since $x \notin var(u'')$,

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

Since $depth(q') = n + 2$ by (8), x_d occurs in c (Lemma 3), and $depth(r) = n + 1$, this is only possible if

- $c = x_d$ and
- $\sigma(u'') \leftrightarrows a$.

(Indeed, by Definition 2, the only other possible forms of a configuration c containing x_d are $c_1 \cdot w$ and $c_1 \rhd w$ for some configuration c_1 and BPA$_{\mathrm{int}}$ term w. In both of these cases,

$$depth(\sigma'[x_d \mapsto r](c)) \geq n + 2 = depth(q') \ ,$$

contradicting $q' = \sigma'[x_d \mapsto r](c) \rhd \sigma(u'')$.) We shall now argue that

$$\sigma(u_j) \leftrightarrows \Phi_n \rhd a \ , \tag{11}$$

proving that $q = \sigma(u)$ has a summand bisimilar to $\Phi_n \rhd a$, which was to be shown. In fact,

$$\sigma(u_j) \ = \ \sigma(u') \rhd \sigma(u'') \ \leftrightarrows \ \sigma(u') \rhd a \ .$$

We shall now prove that $\sigma(u') \leftrightarrows \Phi_n$. Indeed, since

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

it follows that

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

Recall that $\sigma(u) \leftrightarrows \Phi_n \rhd a$. Therefore, there is an a-derivative of $\Phi_n \rhd a$ that is bisimilar to $\sigma(u')$. This a-derivative of $\Phi_n \rhd a$ can only be Φ_n. In fact, the other a-derivatives of $\Phi_n \rhd a$ have the form

$$\left(a^{j-1} + a^3 + a\right) \rhd a \quad (j \in \{2, \ldots, n\}) \ .$$

If one of those terms were bisimilar to $\sigma(u')$, then using our assumption that $\sigma(u) \leftrightarrow \Phi_n \rhd a$, we could infer that, for some $j \in \{2, \ldots, n\}$,

$$\Phi_n \rhd a \leftrightarrow \Phi_n \rhd a + \Big(\Big(a^{j-1} + a^3 + a\Big) \rhd a\Big) \rhd a \ .$$

This contradicts Lemma 14.

Therefore, $\sigma(u') \leftrightarrow \Phi_n$, as claimed. We may finally conclude that $\sigma(u_j) \leftrightarrow \Phi_n \rhd a$.

The proof for case 3c.ii is now complete.

- Case 3c.iii. Since $\sigma'(u'') \xrightarrow{a} \surd$, using Lemma 4(6) we may infer that
  - $u'' \xrightarrow{a} \surd$, or
  - $u'' \xrightarrow{y} \surd$ and $\sigma'(y) \xrightarrow{a} \surd$, for some variable y.

  In the latter case, as $\sigma'(x) \xrightarrow{a} \surd$ does not hold, $y \neq x$, and so $\sigma(y) = \sigma'(y) \xrightarrow{a} \surd$. Using statements 1 and 3 of Lemma 4, we therefore in either case have that
  $$\sigma(u'') \xrightarrow{a} \surd \ .$$
  This yields that $\sigma(u_j) = \sigma(u') \rhd \sigma(u'') \xrightarrow{a} \sigma(u')$.

  Now, reasoning exactly as in the previous case, we can argue that $\sigma(u') \leftrightarrow \Phi_n$. Therefore, using congruence of $\leftrightarrow$,

  $$\sigma(u_j) \leftrightarrow \Phi_n \rhd \sigma(u'') \ .$$

  This equivalence yields that $depth(\sigma(u_j)) = depth(\sigma(u)) = n + 1$, and that the depth of $\sigma(u'')$ is 1. It follows that $\sigma(u'') \leftrightarrow a$. Hence, $\sigma(u)$ has a summand, namely $\sigma(u_j)$, that is bisimilar to $\Phi_n \rhd a$.

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. □

5 BPA with the Disrupt Operator

As mentioned in Section 1, in their paper [7], Baeten and Bergstra have given a finite axiomatization of bisimilarity over 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, using auxiliary operators. The main result in this paper (Theorem 9) shows that the use of auxiliary operators is indeed necessary in order to obtain a finite axiomatization of bisimulation equivalence over the language BPA_{int}, and that this holds true even if we restrict ourselves to axiomatizing the collection of closed equations over this language.

A natural question to ask at this point is whether this negative result applies also to the language BPA_{dis} obtained by enriching BPA with the disrupt operator. Intu-

itively, "p disrupted by q"—which we shall write $p \blacktriangleright q$ in what follows—describes a process that normally behaves like p. However, at each point of the computation before p terminates, q can begin its execution. If this happens, q takes over, and p never resumes its computation. This intuition is captured formally by the following transition rules:

$$\frac{t \stackrel{a}{\rightarrow}\surd}{t \blacktriangleright u \stackrel{a}{\rightarrow}\surd} \qquad \frac{t \stackrel{a}{\rightarrow} t'}{t \blacktriangleright u \stackrel{a}{\rightarrow} t' \blacktriangleright u} \qquad \frac{u \stackrel{a}{\rightarrow}\surd}{t \blacktriangleright u \stackrel{a}{\rightarrow}\surd} \qquad \frac{u \stackrel{a}{\rightarrow} u'}{t \blacktriangleright u \stackrel{a}{\rightarrow} u'}$$

As was the case for the interrupt operator (see Proposition 7), the disrupt operator cannot be defined in the language BPA modulo bisimulation equivalence.

Proposition 20 *There is no* BPA_{dis} *term* t *such that* t *does not contain occurrences of the disrupt operator, and* $t \leftrightarrows x \blacktriangleright y$.

PROOF. Assume, towards a contradiction, that t is a $\mathrm{BPA}_{\mathrm{dis}}$ term such that t does not contain occurrences of the disrupt operator, and $t \leftrightarrows x \blacktriangleright y$.

Consider the closed substitution σ mapping variable x to a and each other variable to a^2. Since

$$\sigma(t) \leftrightarrows a \blacktriangleright a^2 \text{ and } a \blacktriangleright a^2 \stackrel{a}{\rightarrow}\surd \ ,$$

we have that $\sigma(t) \stackrel{a}{\rightarrow}\surd$. Since $\sigma(t) \stackrel{a}{\rightarrow}\surd$ and t does not contain occurrences of the disrupt operator, it is not hard to see that, for some term u, either $t \leftrightarrows a + u$ or $t \leftrightarrows x + u$. Both of these possibilities lead to a contradiction.

Indeed, using the former possibility, we may infer that

$$\sigma[x \mapsto a^2](t) \stackrel{a}{\rightarrow}\surd \ .$$

This implies that $t \not\leftrightarrows x \blacktriangleright y$, because $a^2 \blacktriangleright a^2$ does not have termination traces of length 1.

Assume now that $t \leftrightarrows x + u$. We claim that

$$\sigma[x \mapsto a^2, y \mapsto a^3](t) \not\leftrightarrows a^2 \blacktriangleright a^3 \ .$$

This follows because $\sigma[x \mapsto a^2, y \mapsto a^3](t) \stackrel{a}{\rightarrow} p$ for some $p \leftrightarrows a$, since $t \leftrightarrows x + u$. On the other hand, the two a-derivatives of $a^2 \blacktriangleright a^3$, namely $a \blacktriangleright a^3$ and a^2, have depth at least 2, and thus neither of them is bisimilar to a. □

It is not hard to see that the following equations are sound modulo bisimilarity over

the language $\mathrm{BPA}_{\mathrm{dis}}$:

$$\begin{array}{ll} \text{(D1)} & a \blacktriangleright x \approx a + x \\ \text{(D2)} & ax \blacktriangleright y \approx a(x \blacktriangleright y) + y \quad \text{and} \\ \text{(D3)} & (x + y) \blacktriangleright z \approx (x \blacktriangleright z) + (y \blacktriangleright z) \ . \end{array}$$

In the first two equations above, the symbol a ranges over the set of actions A. Those two identities are therefore equation schemas. Note, however, that such schemas have only finitely many instances if A is finite.

The last of the equations above is particularly important, at least as far as obtaining a finite equational axiomatization of bisimilarity over the collection of closed terms in the language $\mathrm{BPA}_{\mathrm{dis}}$ is concerned. (The interested reader may have already noticed that its soundness modulo bisimulation equivalence depends crucially on the fact that transitions due to moves of the second argument of a disrupt discard the first argument.) Indeed, its repeated use in conjunction with the first two laws allows us to eliminate occurrences of the disrupt operator from closed terms. This effectively reduces the problem of finitely axiomatizing bisimilarity over the collection of closed terms in the language $\mathrm{BPA}_{\mathrm{dis}}$ to that of offering a finite axiomatization of bisimilarity over closed BPA terms. As shown by Bergstra and Klop in [12], the five equations in Table 3 suffice to axiomatize bisimilarity over the language BPA.

In sharp contrast to Theorem 9, we therefore have that:

Theorem 21 *The collection of closed equations over* BPA_{dis} *that hold modulo* $\underline{\leftrightarrow}$ *is axiomatized by (A1)–(A5) in Table 3 together with (D1)–(D3), and is therefore finitely based if* A *is finite.*

It follows that, in the presence of a finite action set, the use of auxiliary operators is *not* necessary in order to obtain a finite axiomatization of bisimulation equivalence over closed terms in the language $\mathrm{BPA}_{\mathrm{dis}}$.

The axiomatization of bisimilarity over closed terms in the language $\mathrm{BPA}_{\mathrm{dis}}$ offered in the theorem above is not ω-complete. For example, the reader can easily check that the disrupt operator is associative modulo bisimilarity, i.e., that the equation

$$(x \blacktriangleright y) \blacktriangleright z \approx x \blacktriangleright (y \blacktriangleright z)$$

holds modulo $\underline{\leftrightarrow}$. This equation is not provable using the equations mentioned in Theorem 21. However, we conjecture that, in the presence of a finite action set, bisimilarity also affords a finite ω-complete axiomatization over $\mathrm{BPA}_{\mathrm{dis}}$. Work on a proof of this conjecture is in progress.

References

[1] L. Aceto, Z. Ésik, A. Ingolfsdottir, On the two-variable fragment of the equational theory of the max-sum algebra of the natural numbers, in: H. Reichel, S. Tison (Eds.), Proceedings of the 17th International Symposium on Theoretical Aspects of Computer Science, STACS 2000 (Lille), Vol. 1770 of Lecture Notes in Computer Science, Springer-Verlag, 2000, pp. 267–278.

[2] L. Aceto, Z. Ésik, A. Ingolfsdottir, The max-plus algebra of the natural numbers has no finite equational basis, Theoretical Comput. Sci. 293 (1) (2003) 169–188.

[3] L. Aceto, W. Fokkink, R. van Glabbeek, A. Ingolfsdottir, Nested semantics over finite trees are equationally hard, Information and Computation 191 (2) (2004) 203–232.

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

[5] L. Aceto, W. Fokkink, A. Ingolfsdottir, B. Luttik, CCS with Hennessy's merge has no finite equational axiomatization, Theoretical Comput. Sci. 330 (3) (2005) 377–405.

[6] L. Aceto, W. Fokkink, A. Ingolfsdottir, S. Nain, Bisimilarity is not finitely based over BPA with interrupt, in: J. L. Fiadeiro, N. Harman, M. Roggenbach, J. J. M. M. Rutten (Eds.), Algebra and Coalgebra in Computer Science: First International Conference, CALCO 2005, Swansea, UK, September 3-6, 2005, Proceedings, Vol. 3629 of Lecture Notes in Computer Science, Springer-Verlag, 2005, pp. 52–66.

[7] J. Baeten, J. Bergstra, Mode transfer in process algebra, Report CSR 00–01, Technische Universiteit Eindhoven, this paper is an expanded and revised version of [11] (2000). Available from `http://alexandria.tue.nl/extra1/wskrap/publichtml/200010731.pdf`.

[8] J. Baeten, J. Bergstra, J. W. Klop, Syntax and defining equations for an interrupt mechanism in process algebra, Fundamenta Informaticae IX (2) (1986) 127–168.

[9] J. Baeten, J. Bergstra, J. W. Klop, Decidability of bisimulation equivalence for processes generating context-free languages, J. ACM 40 (3) (1993) 653–682.

[10] J. Baeten, C. Verhoef, A congruence theorem for structured operational semantics, in: E. Best (Ed.), Proceedings CONCUR 93, Hildesheim, Germany, Vol. 715 of Lecture Notes in Computer Science, Springer-Verlag, 1993, pp. 477–492.

[11] J. Bergstra, A mode transfer operator in process algebra, Report P8808, Programming Research Group, University of Amsterdam (1988).

[12] J. Bergstra, J. W. Klop, Fixed point semantics in process algebras, Report IW 206, Mathematisch Centrum, Amsterdam (1982).

[13] J. Bergstra, J. W. Klop, Process algebra for synchronous communication, Information and Control 60 (1/3) (1984) 109–137.

[14] S. Blom, W. Fokkink, S. Nain, On the axiomatizability of ready traces, ready simulation and failure traces, in: J. Baeten, J. K. Lenstra, J. Parrow, G. J. Woeginger (Eds.), Proceedings 30th Colloquium on Automata, Languages and Programming—ICALP'03, Eindhoven, Vol. 2719 of Lecture Notes in Computer Science, Springer-Verlag, 2003, pp. 109–118.

[15] E. Brinksma, A tutorial on LOTOS, in: M. Diaz (Ed.), Proceedings of the IFIP Workshop on Protocol Specification, Testing and Verification, North-Holland, 1986, pp. 73–84.

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

[17] A. Dsouza, B. Bloom, On the expressive power of CCS, in: P. S. Thiagarajan (Ed.), Foundations of Software Technology and Theoretical Computer Science (Bangalore, 1995), Vol. 1026 of Lecture Notes in Computer Science, Springer-Verlag, 1995, pp. 309–323.

[18] W. Fokkink, B. Luttik, An omega-complete equational specification of interleaving, in: U. Montanari, J. Rolinn, E. Welzl (Eds.), Proceedings 27th Colloquium on Automata, Languages and Programming—ICALP'00, Geneva, Vol. 1853 of Lecture Notes in Computer Science, Springer-Verlag, 2000, pp. 729–743.

[19] W. Fokkink, S. Nain, On finite alphabets and infinite bases: From ready pairs to possible worlds, in: I. Walukiewicz (Ed.), Proceedings of Foundations of Software Science and Computation Structures, 7th International Conference, FOSSACS 2004, Vol. 2897, Springer-Verlag, 2004, pp. 182–194.

[20] J. L. Gischer, The equational theory of pomsets, Theoretical Comput. Sci. 61 (1988) 199–224.

[21] J. F. Groote, A new strategy for proving ω–completeness with applications in process algebra, in: J. Baeten, J. W. Klop (Eds.), Proceedings CONCUR 90, Amsterdam, Vol. 458 of Lecture Notes in Computer Science, Springer-Verlag, 1990, pp. 314–331.

[22] M. Hennessy, Axiomatising finite concurrent processes, SIAM J. Comput. 17 (5) (1988) 997–1017.

[23] 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).

[24] S. Mauw, PSF – A Process Specification Formalism, Ph.D. thesis, University of Amsterdam (Dec. 1991).

[25] R. Milner, Communication and Concurrency, Prentice-Hall International, Englewood Cliffs, 1989.

[26] R. Milner, M. Tofte, R. Harper, D. MacQueen, The Definition of Standard ML (Revised), MIT Press, 1997.

[27] F. Moller, The importance of the left merge operator in process algebras, in: M. Paterson (Ed.), Proceedings 17^{th} ICALP, Warwick, Vol. 443 of Lecture Notes in Computer Science, Springer-Verlag, 1990, pp. 752–764.

[28] F. Moller, 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.

[29] D. Park, Concurrency and automata on infinite sequences, in: P. Deussen (Ed.), 5^{th} GI Conference, Karlsruhe, Germany, Vol. 104 of Lecture Notes in Computer Science, Springer-Verlag, 1981, pp. 167–183.

[30] V. Redko, On defining relations for the algebra of regular events, Ukrainskii Matematicheskii Zhurnal 16 (1964) 120–126, in Russian.

[31] P. Sewell, Nonaxiomatisability of equivalences over finite state processes, Annals of Pure and Applied Logic 90 (1–3) (1997) 163–191.

Recent BRICS Report Series Publications

RS-05-33 **Luca Aceto, Willem Jan Fokkink, Anna Ingólfsdóttir, and Sumit Nain.** ***Bisimilarity is not Finitely Based over BPA with Interrupt*****. October 2005. 33 pp. This paper supersedes BRICS Report RS-04-24. An extended abstract of this paper appeared in** ***Algebra and Coalgebra in Computer Science, 1st Conference, CALCO 2005*****, Swansea, Wales, 3–6 September 2005, Lecture Notes in Computer Science 3629, pp. 54–68, Springer-Verlag, 2005.**

RS-05-32 **Anders Møller, Mads Østerby Olesen, and Michael I. Schwartzbach.** ***Static Validation of XSL Transformations*****. October 2005. 50 pp.**

RS-05-31 **Christian Kirkegaard and Anders Møller.** ***Type Checking with XML Schema in*** **XACT. September 2005. 20 pp.**

RS-05-30 **Karl Krukow.** ***An Operational Semantics for Trust Policies*****. September 2005. 38 pp.**

RS-05-29 **Olivier Danvy and Henning Korsholm Rohde.** ***On Obtaining the Boyer-Moore String-Matching Algorithm by Partial Evaluation*****. September 2005. ii+9 pp. To appear in** ***Information Processing Letters*****. This version supersedes BRICS RS-05-14.**

RS-05-28 **Jiří Srba.** ***On Counting the Number of Consistent Genotype Assignments for Pedigrees*****. September 2005. 15 pp. To appear in FSTTCS '05.**

RS-05-27 **Pascal Zimmer.** ***A Calculus for Context-Awareness*****. August 2005. 21 pp.**

RS-05-26 **Henning Korsholm Rohde.** ***Measuring the Propagation of Information in Partial Evaluation*****. August 2005. 39 pp.**

RS-05-25 **Dariusz Biernacki and Olivier Danvy.** ***A Simple Proof of a Folklore Theorem about Delimited Control*****. August 2005. ii+11 pp. To appear in** ***Journal of Functional Programming*****. This version supersedes BRICS RS-05-10.**

RS-05-24 **Małgorzata Biernacka, Dariusz Biernacki, and Olivier Danvy.** ***An Operational Foundation for Delimited Continuations in the CPS Hierarchy*****. August 2005. iv+43 pp. To appear in the journal** ***Logical Methods in Computer Science*****. This version supersedes BRICS RS-05-11.**