Basic Process Algebra with Iteration: Completeness of its Equational Axioms

Wan Fokkink
CWI
Kruislaan 413, 1098 SJ Amsterdam, The Netherlands

Hans Zantema
Utrecht University
Padualaan 14, 3508 TB Utrecht, The Netherlands

Abstract

Bergstra, Bethke and Ponse proposed an axiomatization for Basic Process Algebra extended with (binary) iteration. In this paper, we prove that this axiomatization is complete with respect to strong bisimulation equivalence. To obtain this result, we will set up a term rewriting system, based on the axioms, and prove that this term rewriting system is terminating, and that bisimilar normal forms are syntactically equal modulo AC.

1 Introduction

Kleene [10] defined a binary operator x^*y in the context of finite automata, which denotes the iterate of x and y. Intuitively, the expression x^*y can choose to execute either x, after which it evolves into x^*y again, or y, after which it terminates. Kleene formulated some algebraic laws for this operator, notably (in our notation) $x^*y = x \cdot x^*y + y$. Copi, Elgot and Wright [6] proposed a simplification of Kleene's setting, e.g. they defined a unary version of the Kleene star in the presence of an empty word. The unary Kleene star has been studied extensively ever since.

Redko [15] (see also [5]) proved for the unary Kleene star that a complete finite axiomatization for language equality does not exist. Salomaa [16] presented a complete finite axiomatization which incorporates one conditional axiom, namely (in our notation) $x = y \cdot x + z$ implies $x = y^*z$ if y does not incorporate the empty word. According to Kozen [11] this last property is not algebraic, in the sense that it is not preserved under substitution of terms for actions. He proposed two alternative conditional axioms which do not have this drawback. These axioms however are not sound in the setting of (strong) bisimulation equivalence.[1]

Milner [12] studied the Kleene star in the setting of bisimulation equivalence, and raised the question whether there exists a complete axiomatization for it. Bergstra, Bethke and Ponse [3] incorporated the binary Kleene star in Basic Process Algebra (BPA). They suggested three axioms BKS1-3 for BPA*, where axiom BKS1 is the defining axiom from Kleene, while their most advanced axiom BKS3 originates from Troeger [18]:

$$x^*(y \cdot (x+y)^*z + z) \;=\; (x+y)^*z$$

In this paper we prove that BKS1-3, together with the five standard axioms for BPA, form a complete axiomatization for BPA* with respect to bisimulation equivalence. For this purpose, we will replace iteration by proper iteration $x^\oplus y$. This construct executes x at least one time, or in other words, $x^\oplus y$ is equivalent to $x \cdot x^* y$. The axioms BKS1-3 are adapted to this new setting, and we will define a term rewriting system based on the axioms of BPA$^\oplus$. Deducing termination of this TRS is a key step in the completeness proof; we will apply the strategy of semantic labelling from one of the authors [20]. Finally, we will show that bisimilar normal forms are syntactically equal modulo AC. These results together imply that the axiomatization for BPA* is complete with respect to bisimulation equivalence. Moreover, the applied method yields an efficient algorithm to decide whether or not two terms are bisimilar.

Sewell [17] proved that if the deadlock δ is added to BPA*, then a complete finite equational axiomatization does not exist. In [7] it has been shown that if sequential composition and iteration are replaced by their prefix counterparts, then six simple equational axioms are complete for this algebra.

Acknowledgements. Jan Bergstra is thanked for his enthusiastic support, and Jos van Wamel for many stimulating discussions.

2 BPA with Binary Kleene Star

This section introduces the basic notions. We assume an alphabet A of atomic actions, together with three binary operators: alternative composi-

[1]For example, one of Kozen's axioms is $x+y\cdot x+z=x \Longrightarrow x+y^*z=x$, which induces $(a+b)^*c+a^*c=(a+b)^*c$.

tion $+$, sequential composition $\cdot$, and binary Kleene star *. Table 1 presents an operational semantics for BPA* in Plotkin style [14]. The special symbol $\surd$ represents (successful) termination.

$$a \xrightarrow{a} \surd$$

$$\frac{x \xrightarrow{a} \surd}{x + y \xrightarrow{a} \surd \xleftarrow{a} y + x} \qquad \frac{x \xrightarrow{a} x'}{x + y \xrightarrow{a} x' \xleftarrow{a} y + x}$$

$$\frac{x \xrightarrow{a} \surd}{x \cdot y \xrightarrow{a} y} \qquad \frac{x \xrightarrow{a} x'}{x \cdot y \xrightarrow{a} x' \cdot y}$$

$$\frac{x \xrightarrow{a} \surd}{x^*y \xrightarrow{a} x^*y} \qquad \frac{x \xrightarrow{a} x'}{x^*y \xrightarrow{a} x' \cdot x^*y}$$

$$\frac{y \xrightarrow{a} \surd}{x^*y \xrightarrow{a} \surd} \qquad \frac{y \xrightarrow{a} y'}{x^*y \xrightarrow{a} y'}$$

Table 1: Action rules for BPA*

Our model for BPA* consists of all the closed terms that can be constructed from the atomic actions and the three binary operators. That is, the BNF grammar for the collection of process terms is as follows, where $a \in A$:

$$p \quad ::= \quad a \mid p + p \mid p \cdot p \mid p^*p.$$

In the sequel the operator $\cdot$ will often be omitted, so pq denotes $p \cdot q$. As binding convention, * and $\cdot$ bind stronger than $+$. Often, $p \cdot q$ will be abbreviated to pq.

Process terms are considered modulo (strong) bisimulation equivalence from Park [13]. Intuitively, two process terms are bisimilar if they have the same branching structure.

Definition 1 *Two processes p and q are* bisimilar, *denoted by $p \leftrightarroweq q$, if there exists a symmetric binary relation $\mathcal{B}$ on processe which relates p and q, such that:*

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

The action rules in Table 1 are in the 'path' format of Baeten and Verhoef [2]. Hence, bisimulation equivalence is a congruence with respect to all the operators, which means that if $p \underline{\leftrightarrow} p'$ and $q \underline{\leftrightarrow} q'$, then $p+q \underline{\leftrightarrow} p'+q'$ and $pq \underline{\leftrightarrow} p'q'$ and $p^*q \underline{\leftrightarrow} p'^*q'$. See [2] for the definition of the path format, and for a proof of this congruence result. (This proof uses the extra assumption that the rules are *well-founded*. Fokkink and van Glabbeek [8] showed that this requirement can be dropped.)

Furthermore, the action rules for BPA are 'pure', which is a syntactic criterion from Groote and Vaandrager (1992), and the two rules for iteration incorporate the Kleene star in the left-hand side of their conclusions. Hence, BPA* is an operationally conservative extension of BPA, i.e. the action rules for iteration do not influence the transition systems of BPA terms. See Verhoef [19] for the definitions, and for a proof of this conservativity result.

Table 2 contains an axiom system for BPA*. It consists of the standard axioms A1-5 for BPA, together with three axioms BKS1-3 for iteration. In the sequel, $p = q$ will mean that the equality can be derived from these axioms.

The axiomatization for BPA* is sound with respect to bisimulation equivalence, i.e. if $p = q$ then $p \underline{\leftrightarrow} q$. Since bisimulation equivalence is a congruence, this can be verified by checking soundness for each axiom separately, which is left to the reader. The purpose of this paper is to prove that the axiomatization is complete with respect to bisimulation, i.e. if $p \underline{\leftrightarrow} q$ then $p = q$.

A1	$x+y$	$=$	$y+x$
A2	$(x+y)+z$	$=$	$x+(y+z)$
A3	$x+x$	$=$	x
A4	$(x+y)z$	$=$	$xz+yz$
A5	$(xy)z$	$=$	$x(yz)$
BKS1	$x(x^*y)+y$	$=$	x^*y
BKS2	$(x^*y)z$	$=$	$x^*(yz)$
BKS3	$x^*(y((x+y)^*z)+z)$	$=$	$(x+y)^*z$

Table 2: Axioms for BPA*

3 A Conditional Term Rewriting System

Our aim is to define a Term Rewriting System (TRS) for process terms in BPA* that reduces each term to a unique normal form, such that if two terms are bisimilar, then they have the same normal form. However, we shall see that one cannot hope to find such a TRS for iteration. Therefore, we will replace it by a new, equivalent operator $p^{\oplus}q$, representing the behaviour of $p(p^*q)$, and we will develop a TRS for the algebra BPA$^{\oplus}$.

We want our TRS to be terminating, so we cannot add the axioms A1,2 as rewrite rules. Therefore, process terms are considered modulo AC, that is, modulo associativity and commutativity of the $+$.

3.1 Turning round two rules for BPA

The axiom A3 yields the expected rewrite rule

$$\mathbf{x} + \mathbf{x} \longrightarrow \mathbf{x}.$$

Usually, in BPA, the axiom A4 as a rewrite rule aims from left to right. However, in BPA* we need this rewrite rule in the opposite direction. For example, in order to reduce the term $a((a+b)^*c) + b((a+b)^*c) + c$ to the term $(a+b)^*c$, we need the reduction

$$a(a+b)^*c) + b((a+b)^*c) \longrightarrow (a+b)((a+b)^*c).$$

Hence, we define the rewrite rule for A4 the other way round.

$$\mathbf{xz} + \mathbf{yz} \longrightarrow (\mathbf{x} + \mathbf{y})\mathbf{z}.$$

In BPA the axiom A5 aims from left to right too, but since we have reversed A4, we must do the same for A5, otherwise the TRS would not be confluent. For example, the term $(ab)d + (ac)d$ would have two different normal forms:

$$a(bd) + a(cd) \text{ and } (ab + ac)d.$$

So we opt for the rule

$$\mathbf{x}(\mathbf{yz}) \longrightarrow (\mathbf{xy})\mathbf{z}.$$

3.2 Proper iteration

Although we have already defined part of a TRS that should reduce terms that are bisimilar to the same normal form, we shall see now that such a TRS does not exist at all.

Since $x^*y + z \underline{\leftrightarrow} x^*y$ if $y + z \underline{\leftrightarrow} y$, such terms should have the same normal form. Therefore, one would expect a rule

$$x^*y + z \longrightarrow x^*y \qquad \text{if } y + z \twoheadrightarrow y.$$

However, this rule does not yield unique normal forms, because we have reversed the rule for A4. For example, the term $a^*(b + ce) + ce + de$ would have two different normal forms:

$$a^*(b + ce) + de \;\text{ and }\; a^*(b + ce) + (c + d)e.$$

To avoid this complication, we replace iteration by an operator $x^{\oplus}y$, called *proper* iteration, which displays the behaviour of $x(x^*y)$.[2] The operational semantics and the axiomatization for proper iteration are given in Tables 3 and 4. They are obtained from the action rules and axioms for iteration, using the equivalences $x^*y \underline{\leftrightarrow} x^{\oplus}y + y$ and $x^{\oplus}y \underline{\leftrightarrow} x(x^*y)$. Note that

$$\begin{array}{llll} \text{BPA}^* & + & (x^{\oplus}y = x(x^*y)) & \vdash \text{PI1-3}, \\ \text{BPA}^{\oplus} & + & (x^*y = x^{\oplus}y + y) & \vdash \text{BKS1-3}. \end{array}$$

So we find that the axiomatization in Table 4 is complete for BPA$^{\oplus}$ if and only if the axiomatization in Table 2 is complete for BPA*.

$$\frac{x \xrightarrow{a} x'}{x^{\oplus}y \xrightarrow{a} x'(x^{\oplus}y + y)} \qquad \frac{x \xrightarrow{a} \surd}{x^{\oplus}y \xrightarrow{a} x^{\oplus}y + y}$$

Table 3: Action rules for proper iteration

PI1	$x(x^{\oplus}y + y)$	$=$	$x^{\oplus}y$
PI2	$(x^{\oplus}y)z$	$=$	$x^{\oplus}(yz)$
PI3	$x^{\oplus}(y((x + y)^{\oplus}z + z) + z)$	$=$	$x((x + y)^{\oplus}z + z)$

Table 4: Axioms for proper iteration

[2]The standard notation for this construct would be x^+y, but we want to avoid ambiguous use of the $+$.

3.3 One rule for axiom PI2

Now that we have replaced iteration by proper iteration, we can continue to define rewrite rules for this new operator. We start with the one for axiom PI2. The question is whether it should rewrite from left to right or vice versa. If it would rewrite from left to right, it would clash with the rule for A4. For example, then the term $(a^{\oplus}b)c+dc$ would have two different normal forms:

$$a^{\oplus}(bc)+dc \;\text{ and }\; (a^{\oplus}b+d)c.$$

Hence, PI2 yields the rule

$$\mathbf{x}^{\oplus}(\mathbf{yz}) \;\longrightarrow\; (\mathbf{x}^{\oplus}\mathbf{y})\mathbf{z}.$$

3.4 Four rules for axiom PI1

The next rule stems from axiom PI1:

$$\mathbf{x}(\mathbf{x}^{\oplus}\mathbf{y}+\mathbf{y}) \;\longrightarrow\; \mathbf{x}^{\oplus}\mathbf{y}.$$

This rewrite rule causes serious complications concerning confluence; it turns out that we need three extra rules to obtain this property.

1. A term $x(y^{\oplus}z+z)+y(y^{\oplus}z+z)$ has two different reductions:

   $$x(y^{\oplus}z+z)+y^{\oplus}z \;\text{ and }\; (x+y)(y^{\oplus}z+z).$$

   So for the sake of confluence, one of these two reducts should reduce to the other. If we would add the rule $(x+y)(y^{\oplus}z+z) \longrightarrow x(y^{\oplus}z+z)+y^{\oplus}z$ to the TRS, then the term $(ac+bc)((bc)^{\oplus}d+d)$ would have two different normal forms:

   $$(ac)((bc)^{\oplus}d+d)+(bc)^{\oplus}d \;\text{ and }\; ((a+b)c)((bc)^{\oplus}d+d).$$

   Hence, we opt for the rule

   $$\mathbf{x}(\mathbf{y}^{\oplus}\mathbf{z}+\mathbf{z})+\mathbf{y}^{\oplus}\mathbf{z} \;\longrightarrow\; (\mathbf{x}+\mathbf{y})(\mathbf{y}^{\oplus}\mathbf{z}+\mathbf{z}).$$

2. A term $x(y(y^{\oplus}z+z))$ has two different reductions:

   $$x(y^{\oplus}z) \;\text{ and }\; (xy)(y^{\oplus}z+z).$$

   A rule $(xy)(y^{\oplus}z+z) \longrightarrow x(y^{\oplus}z)$ clashes with the rule for A5, because then the term $(a(bc))((bc)^{\oplus}d+d))$ would get two different normal forms:

   $$a((bc)^{\oplus}d) \;\text{ and }\; ((ab)c)((bc)^{\oplus}d+d)).$$

Therefore, we define

$$\mathbf{x}(\mathbf{y}^{\oplus}\mathbf{z}) \longrightarrow (\mathbf{x}\mathbf{y})(\mathbf{y}^{\oplus}\mathbf{z}+\mathbf{z}).$$

3. Finally, a term $x^{\oplus}(y(y^{\oplus}z+z))$ has two different reductions:

   $$x^{\oplus}(y^{\oplus}z) \text{ and } (x^{\oplus}y)(y^{\oplus}z+z).$$

   Since a rule $(x^{\oplus}y)(y^{\oplus}z+z) \longrightarrow x^{\oplus}(y^{\oplus}z)$ would clash with the rule for PI2, we opt for

   $$\mathbf{x}^{\oplus}(\mathbf{y}^{\oplus}\mathbf{z}) \longrightarrow (\mathbf{x}^{\oplus}\mathbf{y})(\mathbf{y}^{\oplus}\mathbf{z}+\mathbf{z}).$$

3.5 Two conditional rules for axiom PI3

The obvious interpretation of axiom PI3 as a rewrite rule,

$$x^{\oplus}(x'((x+x')^{\oplus}z+z)+z) \longrightarrow x((x+x')^{\oplus}z+z),$$

obstructs confluence. For if x and x' are normal forms, while the expression $x+x'$ is not, then after reducing $x+x'$ we can no longer apply this rule. Therefore, we translate PI3 to a conditional rule:

$$\mathbf{x}^{\oplus}(\mathbf{x}'(\mathbf{y}^{\oplus}\mathbf{z}+\mathbf{z})+\mathbf{z}) \longrightarrow \mathbf{x}(\mathbf{y}^{\oplus}\mathbf{z}+\mathbf{z}) \text{ if } \mathbf{x}+\mathbf{x}' \twoheadrightarrow \mathbf{y}.$$

Again, this rule leads to a TRS that is not confluent, because a term $x^{\oplus}(y(y^{\oplus}z+z)+z)$ with $x+y \twoheadrightarrow y$ has two reductions:

$$x^{\oplus}(y^{\oplus}z+z) \text{ and } x(y^{\oplus}z+z).$$

So in order to obtain confluence, we add one last conditional rule to the TRS:

$$\mathbf{x}^{\oplus}(\mathbf{y}^{\oplus}\mathbf{z}+\mathbf{z}) \longrightarrow \mathbf{x}(\mathbf{y}^{\oplus}\mathbf{z}+\mathbf{z}) \quad \text{if } \mathbf{x}+\mathbf{y} \twoheadrightarrow \mathbf{y}.$$

3.6 The entire TRS

The entire TRS is given once again in Table 5. The rules are to be interpreted modulo AC. It is easy to see that all rules can be deduced from BPA$^{\oplus}$.

The usual strategy for deducing that each term has a unique normal form, is to prove that the TRS is both *weakly confluent*, (i.e. if a term p has reductions $p' \longleftarrow p \longrightarrow p''$, then there exists a q such that $p' \twoheadrightarrow q \twoheadleftarrow p''$), and *terminating* (i.e. there are no infinite reductions). Newman's Lemma says that such a TRS reduces each term to a unique normal form, which does not reduce any further.

1.	$x + x$	$\longrightarrow$	x
2.	$xz + yz$	$\longrightarrow$	$(x + y)z$
3.	$x(yz)$	$\longrightarrow$	$(xy)z$
4.	$x^{\oplus}(yz)$	$\longrightarrow$	$(x^{\oplus}y)z$
5.	$x(x^{\oplus}y + y)$	$\longrightarrow$	$x^{\oplus}y$
6.	$x(y^{\oplus}z + z) + y^{\oplus}z$	$\longrightarrow$	$(x + y)(y^{\oplus}z + z)$
7.	$x(y^{\oplus}z)$	$\longrightarrow$	$(xy)(y^{\oplus}z + z)$
8.	$x^{\oplus}(y^{\oplus}z)$	$\longrightarrow$	$(x^{\oplus}y)(y^{\oplus}z + z)$
9.	$x^{\oplus}(x'(y^{\oplus}z + z) + z)$	$\longrightarrow$	$x(y^{\oplus}z + z)$
			if $x + x' \twoheadrightarrow y$
10.	$x^{\oplus}(y^{\oplus}z + z)$	$\longrightarrow$	$x(y^{\oplus}z + z)$
			if $x + y \twoheadrightarrow y$

Table 5: Rewrite rules for BPA$^{\oplus}$

Although our choice of rewrite rules has been motivated by the wish for a weakly confluent TRS, it is not so easy to deduce this property yet, due to the presence of conditional rules. The next example shows that the usual method for checking weak confluence of a TRS, namely verifying this property for all overlapping redexes, does not work in a conditional setting.

Example 2 *Consider the TRS which consists of the rules*

$$\begin{array}{rcll} f(x) & \longrightarrow & b & \text{if } x \twoheadrightarrow a, \\ a & \longrightarrow & c. & \end{array}$$

There are no overlapping redexes, but this TRS is not weakly confluent: $f(c) \longleftarrow f(a) \longrightarrow b$.

However, it will turn out that the confluence property is not needed in the proof of the main theorem, which states that bisimilar normal forms are equal modulo AC. Hence, confluence will simply be a consequence of this theorem.

3.7 Termination

Proving termination of the TRS in Table 5, modulo AC, is a complicated matter. This is mainly due to the presence of Rule 7, in which the left-hand side can be obtained from the right-hand side by the removal of subterms. A powerful technique for proving termination of TRSs that incorporate such rules is *semantic labelling* [20], where operation symbols that occur in the rewrite rules are supplied with labels, which depend on the semantics of the arguments. Then two TRSs are involved: the original system and the labelled system. The main theorem of [20] states that the labelled system terminates if and only if the original system terminates.

The theory of semantic labelling has been developed for unconditional TRSs. Therefore, we adapt the TRS in Table 5 to an unconditional TRS R, simply by removing the conditions from the last two rules. We shall prove that R is terminating, which immediately implies termination of the conditional TRS in Table 5.

Proposition 3 *The TRS R is terminating.*

Proof. The method from [20] starts with choosing a model, which consists of a set $\mathcal{M}$, and for each function symbol f in the original signature with arity n a mapping $f_{\mathcal{M}} : \mathcal{M}^n \to \mathcal{M}$, such that for every rewrite rule, and for all possible values for its variables in the model, the left-hand side and the right-hand side are equal in the model. Here we choose the model to be the positive natural numbers. Each process p is interpreted by its *norm* $|p|$, being the least number of steps in which it can terminate. This norm can be defined inductively as follows:

$$\begin{array}{rcl} |a| & = & 1 \\ |p+q| & = & \min\{|p|,|q|\} \\ |pq| & = & |p|+|q| \\ |p^{\oplus}q| & = & |p|+|q|. \end{array}$$

Note that norm is associative and commutative with respect to the choice operator, which is essential in order to obtain the termination result modulo AC. Clearly norm is preserved under bisimulation equivalence. Since the Rules 1-8 of R are sound with respect to bisimulation, it follows that norm is preserved under application of these rewrite rules. And it is easy to verify that Rules 9 and 10 of R, which are not sound because they lack their original conditions, preserve norm too.

Next, we select labels for the function symbols. As labels for the operators sequential composition and proper iteration we choose the positive natural numbers, while the atoms and the choice operator remain unchanged.

In each ground term, the occurrences of sequential composition and proper iteration are labelled as follows: we replace $p \cdot q$ by $p\langle|q|\rangle q$ and $p^{\oplus}q$ by $p[|q|]q$.

Finally, for each rule in the TRS we construct a collection of labelled rules. This is done by replacing the variables in the original rule by all possible values in the model, and computing the resulting labels for the operators. This results in the following TRS $\bar{R}$, where the rules are defined for positive natural numbers i and j.

$$\begin{array}{rcl}
x + x & \longrightarrow & x \\
x\langle i\rangle z + y\langle i\rangle z & \longrightarrow & (x+y)\langle i\rangle z \\
x\langle i+j\rangle(y\langle j\rangle z) & \longrightarrow & (x\langle i\rangle y)\langle j\rangle z \\
\\
x[i+j](y\langle j\rangle z) & \longrightarrow & (x[i]y)\langle j\rangle z \\
\\
x\langle i\rangle(x[i]y + y) & \longrightarrow & x[i]y \\
x\langle i\rangle(y[i]z+z) + y[i]z & \longrightarrow & (x+y)\langle i\rangle(y[i]z+z) \\
x\langle i+j\rangle(y[j]z) & \longrightarrow & (x\langle i\rangle y)\langle j\rangle(y[j]z+z) \\
x[i+j](y[j]z) & \longrightarrow & (x[i]y)\langle j\rangle(y[j]z+z) \\
\\
x[i](x'\langle i\rangle(y[i]z+z)+z) & \longrightarrow & x\langle i\rangle(y[i]z+z) \\
x[i](y[i]z+z) & \longrightarrow & x\langle i\rangle(y[i]z+z)
\end{array}$$

Suppose that R admits an infinite reduction. Replace the variables in this reduction by a constant a to obtain an infinite ground reduction in R. For each symbol '$\cdot$' and '$^{\oplus}$' that occurs in this reduction, compute its corresponding label. This way the infinite ground reduction in R transforms into an infinite ground reduction in $\bar{R}$. Hence, termination of $\bar{R}$ implies termination of R.

It remains to prove termination of $\bar{R}$. Although $\bar{R}$ is a TRS with infinitely many rules, this is much easier than proving termination of R. Define a weight function w:

$$\begin{array}{rcl}
w(a) & = & 1 \\
w(p+q) & = & w(p) + w(q) \\
w(p\langle i\rangle q) & = & w(p) + iw(q) \\
w(p[i]q) & = & w(p) + (i+1)w(q)
\end{array}$$

It is easy to verify that for any choice of values for variables in any rule, the weight of the left-hand side is strictly greater than the weight of the right-hand side. For example, in the case of Rule 7 these weights are

$$\begin{array}{rl}
 & w(x) \; + \; (i+j)w(y) \; + \; (i+j)(j+1)w(z) \\
\text{and} & w(x) \; + \; (i+j)w(y) \; + \; j(j+2)w(z)
\end{array}$$

respectively. And $(i+j)(j+1) > j(j+2)$ for $i, j \geq 1$.

Due to the strict monotonic behaviour of w (here it is essential that $i > 0$) we conclude that each reduction step yields a strict decrease of weight. Hence the system $\bar{R}$ is terminating, and so R is terminating. $\square$

4 Normal Forms Decide Bisimilarity

In the previous section we have developed a TRS for BPA$^{\oplus}$ that reduces terms to a normal form. Since all rewrite rules are sound with respect to bisimulation equivalence, it follows that each term is bisimilar with its normal forms. So in order to determine completeness of the axiomatization for BPA$^{\oplus}$ with respect to bisimulation equivalence, it is sufficient to prove that if two normal forms are bisimilar, then they are equal modulo AC.

4.1 An ordering on process terms

As induction base in the proof of our main theorem, we will need a well-founded ordering on process terms that should preferably have the following properties:

1. $$\begin{array}{lll} p \leq p+q & p < pq & p < p^{\oplus}q \\ q \leq p+q & q < pq & q < p^{\oplus}q. \end{array}$$

2. The ordering is preserved under bisimulation.

However, an ordering combining these properties is never well-founded, because for such an ordering we have

$$p^{\oplus}q \leq p^{\oplus}q + q < p(p^{\oplus}q + q)$$

Since $p(p^{\oplus}q+q) \underline{\leftrightarrow} p^{\oplus}q$, it follows that $p^{\oplus}q < p^{\oplus}q$.

The norm, indicating the least number of steps a process must make before it can terminate, induces an ordering that *almost* satisfies all desired properties. The only serious drawback of this ordering is that $|p| \geq |p+q|$. Therefore we adapt it to an ordering induced by L-value, which is defined as follows:

$$L(p) = \max\{|p'| \mid p' \text{ is a proper substate of } p\}$$

where 'proper substate' means that p can evolve into p' by one or more transitions. Since norm is preserved under bisimulation equivalence, the same holds for L.

Lemma 4 *If $p \underline{\leftrightarrow} q$, then $L(p) = L(q)$.*

Proof. If p' is a proper substate of p, then bisimilarity of p and q implies that there is a proper substate q' of q such that $p' \underline{\leftrightarrow} q'$, and so $|p'| = |q'|$. Hence, $L(p) \leq L(q)$, and by symmetry $L(q) \leq L(p)$. $\Box$

We deduce the inductive definition for L-value. $L(p+q)$ is the maximum of the collection

$$\begin{array}{rl} & \{|p'| \mid p' \text{ proper substate of } p\} \\ \cup & \{|q'| \mid q' \text{ proper substate of } q\}, \end{array}$$

so $L(p+q) = \max\{L(p), L(q)\}$. Next, $L(pq)$ is the maximum of the collection

$$\begin{array}{rl} & \{|p'q| \mid p' \text{ proper substate of } p\} \\ \cup & \{|q|\} \quad \cup \quad \{|q'| \mid q' \text{ proper substate of } q\}, \end{array}$$

so $L(pq) = \max\{L(p) + |q|, L(q)\}$. Finally, $L(p^{\oplus}q)$ is the maximum of the collection

$$\begin{array}{rl} & \{|p'(p^{\oplus}q + q)| \mid p' \text{ proper substate of } p\} \\ \cup & \{|p^{\oplus}q + q|\} \quad \cup \quad \{|q'| \mid q' \text{ proper substate of } q\}, \end{array}$$

so $|p^{\oplus}q| = \max\{L(p) + |q|, L(q)\}$. Recapitulating, we have found:

$$\begin{array}{rcl} L(a) & = & 0 \\ L(p+q) & = & \max\{L(p), L(q)\} \\ L(pq) & = & \max\{L(p) + |q|, L(q)\} \\ L(p^{\oplus}q) & = & \max\{L(p) + |q|, L(q)\}. \end{array}$$

Hence, L-value too satisfies almost all the requirements formulated above; only, we have inequalities $L(q) \leq L(pq)$ and $L(q) \leq L(p^{\oplus}q)$, instead of the desired strict inequalities. Therefore, we introduce a second weight function g on process terms, defined by:

$$\begin{array}{rcl} g(a) & = & 0 \\ g(p+q) & = & \max\{g(p), g(q)\} \\ g(pq) & = & g(q) + 1 \\ g(p^{\oplus}q) & = & g(q) + 1. \end{array}$$

Note that g-value is not preserved under bisimulation equivalence. However, the following lemma holds.

Lemma 5 *If $p \longrightarrow q$, then $g(p) \geq g(q)$.*

Proof. For each rewrite rule it is easily checked that the g-value of the left-hand side is greater than or equal than the g-value of the right-hand side. Since the functions that are used in the definition of g are weakly monotonous in their coordinates, we may conclude that g-value is never increased by a rewrite step. $\Box$

In the proof of the main theorem we will apply induction on a lexicographical combination of L-value and g-value.

4.2 Some lemmas

We deduce three lemmas that will be used in the proof of the main theorem. The first lemma is typical for normed processes [1], i.e. for processes that are able to terminate in finitely many transitions. This lemma originates from Caucal [4].

Lemma 6 *If $pr \underline{\leftrightarrow} qr$, then $p \underline{\leftrightarrow} q$.*

Proof. A transition $p'r \xrightarrow{a} p''r$ in pr cannot be mimicked by a transition $q'r \xrightarrow{a} r$ in qr, because $|p''r| > |r|$. Hence, each transition $p'r \xrightarrow{a} p''r$ is mimicked by a transition $q'r \xrightarrow{a} q''r$, and vice versa. This induces a bisimulation relation between p and q; the transition $p' \xrightarrow{a} p''$ in p is mimicked by the transition $q' \xrightarrow{a} q''$ in q, and vice versa. $\Box$

Definition 7 *We say that two process terms p and q have* behaviour in common *if there are p' and q' such that $p \xrightarrow{a} p'$ and $q \xrightarrow{a} q'$ and $p' \underline{\leftrightarrow} q'$.*

Lemma 8 *If two terms pq and rs have behaviour in common, and $|q| \geq |s|$, then either $q \underline{\leftrightarrow} ts$ for some t, or $q \underline{\leftrightarrow} s$.*

Proof. If $pq \xrightarrow{a} q$ and $rs \xrightarrow{a} r's$ with $q \underline{\leftrightarrow} r's$, or if $pq \xrightarrow{a} q$ and $rs \xrightarrow{a} s$ with $q \underline{\leftrightarrow} s$, then we are done. And $pq \xrightarrow{a} p'q$ and $rs \xrightarrow{a} s$ with $p'q \underline{\leftrightarrow} s$ would contradict $|q| \geq |s|$. Thus, the only interesting case is if $pq \xrightarrow{a} p'q$ and $rs \xrightarrow{a} r's$ with $p'q \underline{\leftrightarrow} r's$. The inequality $|q| \geq |s|$ then yields $|p'| \leq |r'|$.

We show, with induction on $|p'|$, that $p'q \underline{\leftrightarrow} r's$ together with $|p'| \leq |r'|$ indicate either $q \underline{\leftrightarrow} ts$ for some t or $q \underline{\leftrightarrow} s$. If $|p'| = 1$, then $p' \xrightarrow{a} \surd$, and so $p'q \xrightarrow{a} q$. Since $p'q \underline{\leftrightarrow} r's$, this transition can be mimicked by a transition $r's \xrightarrow{a} r''s$ or $r's \xrightarrow{a} s$, and so $q \underline{\leftrightarrow} r''s$ or $q \underline{\leftrightarrow} s$ respectively.

Next, let $|p'| = n + 1$. Clearly, there is a transition $p' \xrightarrow{a} p''$ with $|p''| = n$. Since $p'q \underline{\leftrightarrow} r's$, and $p'q \xrightarrow{a} p''q$, there must be a transition $r's \xrightarrow{a} r''s$ with $p''q \underline{\leftrightarrow} r''s$. Since $|r'| \geq |p'| = n+1$ implies $|r''| \geq n = |p''|$, the induction hypothesis learns that either $q \underline{\leftrightarrow} ts$ for some t, or $q \underline{\leftrightarrow} s$. $\Box$

Lemma 9 *If a term rs has normal form q, then pq or $p^{\oplus}q$ is* not *a normal form.*

Proof. Suppose that q is a normal form of a term rs. Each rule in Table 5 that applies to a term of the form tu or $t^{\oplus}u$, reduces it to one of either forms again. So q must be in one of either forms. But Rules 3, 4, 7 and 8 reduce $p(tu)$ and $p^{\oplus}(tu)$ and $p(t^{\oplus}u)$ and $p^{\oplus}(t^{\oplus}u)$ respectively. Hence, pq and $p^{\oplus}q$ are not in normal form. □

4.3 The main theorem

Process terms are considered modulo AC. From now on, this equivalence is denoted by $p =_{\mathrm{AC}} q$, and we say that p and q are of the same form. Clearly, each process term p is a sum of terms of the form a and qr and $q^{\oplus}r$, which are called the *summands* of p.

Theorem 10 *If two normal forms p and q are bisimilar, then $p =_{\mathrm{AC}} q$.*

Proof. In order to prove the theorem, we prove three extra statements in parallel.

A. If two normal forms $p =_{\mathrm{AC}} rs$ and $q =_{\mathrm{AC}} tu$ have common behaviour, then $s =_{\mathrm{AC}} u$.

B. If two normal forms $p =_{\mathrm{AC}} rs$ and $q =_{\mathrm{AC}} t^{\oplus}u$ have common behaviour, then $s =_{\mathrm{AC}} t^{\oplus}u + u$.

C. If two normal forms $p =_{\mathrm{AC}} r^{\oplus}s$ and $q =_{\mathrm{AC}} t^{\oplus}u$ have common behaviour, then $r^{\oplus}s =_{\mathrm{AC}} t^{\oplus}u$.

The statement in the main theorem is labelled D.

If $L(p) = L(q) = 0$, then both p and q must be sums of atoms. So in this case A and B and C are empty statements. And D holds too, because bisimilarity of p and q indicates that they contain exactly the same atoms, and Rule 1 ensures that both terms contain each of these atoms only once.

Next, fix an $m > 0$ and assume that we have already proved the four statements if $L(p)$ and $L(q)$ are smaller than m. We will prove it for the case that they are equal to m. Let A_n and B_n and C_n and D_n denote the assertions for pairs p, q with $\max\{L(p), L(q)\} \leq m$ and $g(p) + g(q) \leq n$. They are proved by induction on n.

The case $n = 0$ corresponds with the case $L(p) = L(q) = 0$, because if $g(p) + g(q) = 0$, then both p and q must be sums of atoms. As induction hypothesis we now assume A_n, B_n, C_n and D_n, and we shall prove $A_{n+1}, B_{n+1}, C_{n+1}$ and D_{n+1}.

1. A_{n+1} is true.

Let normal forms rs and tu have behaviour in common, with $L(rs) \leq m$ and $L(tu) \leq m$ and $g(rs) + g(tu) = n + 1$. We want to prove $s =_{\mathrm{AC}} u$. By symmetry we may assume $|s| \geq |u|$, so Lemma 8 offers two possibilities.

1.1 $s \underline{\leftrightarrow} u$.

$L(s) \leq L(rs) \leq m$ and $L(u) \leq L(tu) \leq m$ and $g(s) + g(u) < g(rs) + g(tu) = n + 1$. Hence, D_n yields $s =_{\mathrm{AC}} u$.

1.2 $s \underline{\leftrightarrow} vu$ for some v.

Let w be a normal form of vu. According to Lemma 5 $g(w) \leq g(vu)$, so $g(s)+g(w) < g(rs)+g(vu) = n+1$. Further, since $s \underline{\leftrightarrow} w$, $L(w) = L(s) \leq m$. Hence, D_n yields $s =_{\mathrm{AC}} w$. However, Lemma 9 says that s cannot be a normal form of a term vu. Contradiction.

2. B_{n+1} is true.

According to the previous point we may assume A_{n+1}. Let normal forms rs and $t^{\oplus}u$ have behaviour in common, with $L(rs) \leq m$ and $L(t^{\oplus}u) \leq m$ and $g(rs) + g(t^{\oplus}u) = n + 1$. We want to prove $s =_{\mathrm{AC}} t^{\oplus}u + u$. Since $t^{\oplus}u \underline{\leftrightarrow} t(t^{\oplus}u + u)$, Lemma 8 offers three possibilities.

2.1 $s \underline{\leftrightarrow} t^{\oplus}u + u$.

The term $t^{\oplus}u + u$ is a normal form, because we cannot apply Rules 1,2 or 6 to it. Moreover, $g(s) + g(t^{\oplus}u + u) = g(s) + g(t^{\oplus}u) = n$, so D_n yields $s =_{\mathrm{AC}} t^{\oplus}u + u$.

2.2 $vs \underline{\leftrightarrow} t^{\oplus}u + u$ for some v.

This implies $v's \underline{\leftrightarrow} u$ for some v'. As in 1.2, we can deduce that then u is a normal form of $v's$, which is a contradiction according to Lemma 9.

2.3 $s \underline{\leftrightarrow} v(t^{\oplus}u + u)$ for some v.

Note that $g(s) + g(v(t^{\oplus}u + u)) = n + 1$, so we cannot yet apply D_n.

Let v be a normal form. If $v =_{\mathrm{AC}} t$ then $s \underline{\leftrightarrow} t^{\oplus}u$, so that D_n yields $s =_{\mathrm{AC}} t^{\oplus}u$. Then Rule 7 reduces rs, which is a contradiction. So apparently v cannot be of the form t. Thus, Rule 5 cannot be applied to $v(t^{\oplus}u + u)$, so this term is a normal form.

First, consider a summand $\alpha\beta$ of s. This term and $v(t^{\oplus}u + u)$ have behaviour in common, so A_{n+1} yields $\beta =_{\mathrm{AC}} t^{\oplus}u + u$.

Next, consider a summand $\alpha^{\oplus}\beta$ of s. This term and $v(t^{\oplus}u+u)$ have behaviour in common. Since $\alpha^{\oplus}\beta \underline{\leftrightarrow} \alpha(\alpha^{\oplus}\beta+\beta)$, Lemma 8 offers three possibilities.

- $\alpha^{\oplus}\beta+\beta \underline{\leftrightarrow} t^{\oplus}u+u$.

  $g(\alpha^{\oplus}\beta+\beta)+g(t^{\oplus}u+u) \leq g(s)+g(t^{\oplus}u) = n$, so D_n implies $\alpha^{\oplus}\beta+\beta =_{\mathrm{AC}} t^{\oplus}u+u$. Since the summands of $\alpha^{\oplus}\beta+\beta$ and $t^{\oplus}u+u$ with greatest size are $\alpha^{\oplus}\beta$ and $t^{\oplus}u$ respectively, it follows that $\alpha^{\oplus}\beta =_{\mathrm{AC}} t^{\oplus}u$.

- $w(\alpha^{\oplus}\beta+\beta) \underline{\leftrightarrow} t^{\oplus}u+u$ for some w.

  Then $w'(\alpha^{\oplus}\beta+\beta) \underline{\leftrightarrow} u$ for some w', and we obtain a contradiction as in 1.2.

- $\alpha^{\oplus}\beta+\beta \underline{\leftrightarrow} w(t^{\oplus}u+u)$ for some w.

  Then $\beta \underline{\leftrightarrow} w'(t^{\oplus}u+u)$ for some w', and we obtain a contradiction as in 1.2.

So we may conclude $\alpha^{\oplus}\beta =_{\mathrm{AC}} t^{\oplus}u$.

If s contains several summands of the form $\alpha(t^{\oplus}u+u)$ or $t^{\oplus}u$, then we can apply Rule 1,2 or 6 to s. However, s is in normal form, so apparently it consists of a single term $\alpha(t^{\oplus}u+u)$ or $t^{\oplus}u$. Then apply Rule 3 or 7 applies to rs, which again is a contradiction, because rs is in normal form.

3. C_{n+1} is true.

Assume normal forms $r^{\oplus}s$ and $t^{\oplus}u$ that have behaviour in common, with $L(r^{\oplus}s) \leq m$ and $L(t^{\oplus}u) \leq m$ and $g(r^{\oplus}s)+g(t^{\oplus}u) = n+1$. We want to prove $r^{\oplus}s =_{\mathrm{AC}} t^{\oplus}u$. By symmetry we may assume $|r^{\oplus}s| \geq |t^{\oplus}u|$, so Lemma 8 offers two possibilities.

3.1 $r^{\oplus}s+s \underline{\leftrightarrow} v(t^{\oplus}u+u)$ for some v.

Then $s \underline{\leftrightarrow} v'(t^{\oplus}u+u)$ for some v'. This leads to a contradiction as in 2.3.

3.2 $r^{\oplus}s+s \underline{\leftrightarrow} t^{\oplus}u+u$.

First, suppose that s and u have no behaviour in common with $t^{\oplus}u$ and $r^{\oplus}s$ respectively, so that $s \underline{\leftrightarrow} u$ and $r^{\oplus}s \underline{\leftrightarrow} t^{\oplus}u$. Since D_n applies to the first equivalence, we get $s =_{\mathrm{AC}} u$. And the second equivalence yields $r(r^{\oplus}s+s) \underline{\leftrightarrow} t(t^{\oplus}u+u) \underline{\leftrightarrow} t(r^{\oplus}s+s)$, so Lemma 6 implies $r \underline{\leftrightarrow} t$. Since $L(r) = L(t) < m$, statement D then implies $r =_{\mathrm{AC}} t$, and we are done.

So we can suppose that either s and $t^{\oplus}u$ have behaviour in common, or u and $r^{\oplus}s$ have behaviour in common. We deduce a contradiction.

By symmetry it is sufficient to deduce a contradiction for the first case only, where s and $t^{\oplus}u$ have behaviour in common. If a summand $\alpha\beta$ or $\alpha^{\oplus}\beta$ of s has behaviour in common with $t^{\oplus}u$, then B_n or C_n implies $\beta =_{\rm AC} t^{\oplus}u+u$ or $\alpha^{\oplus}\beta =_{\rm AC} t^{\oplus}u$ respectively. If s contains several summands of the form $\alpha(t^{\oplus}u+u)$ or $t^{\oplus}u$, then Rules 1,2 or 6 can be applied to it. However, s is a normal form, so apparently it contains exactly one such summand.

If u and $r^{\oplus}s$ have behaviour in common too, then similarly we can deduce that u has a summand of the form $\beta(r^{\oplus}s+s)$ or $r^{\oplus}s$, which indicates that u has a size greater than s. On the other hand, s has a summand $\alpha(t^{\oplus}u+u)$ or $t^{\oplus}u$, so s has a size greater than u. This cannot be, so u and $r^{\oplus}s$ have no behaviour in common.

And if u has behaviour in common with the summand $\alpha(t^{\oplus}u+u)$ or $t^{\oplus}u$ of s, then it follows from A_n or B_n or C_n that u has a summand of the form $\beta(t^{\oplus}u+u)$ or $t^{\oplus}u$. Again we establish a contradiction; u has greater size than itself.

Hence, we have found that

- $r^{\oplus}s+s \underline{\leftrightarrow} t^{\oplus}u+u$,
- s has a summand $\alpha(t^{\oplus}u+u)$ or $t^{\oplus}u$, and all other summands of s have no behaviour in common with $t^{\oplus}u$,
- u has no behaviour in common with $r^{\oplus}s$, nor with the summand $\alpha(t^{\oplus}u+u)$ or $t^{\oplus}u$ of s.

From these facts it follows that

- $s =_{\rm AC} \alpha(t^{\oplus}u+u)+s'$ or $s =_{\rm AC} t^{\oplus}u+s'$,
- $s' \underline{\leftrightarrow} u$,
- $r^{\oplus}s+\alpha(t^{\oplus}u+u)$ or $r^{\oplus}s+t^{\oplus}u$ is bisimilar to $t^{\oplus}u$.

Since $s' \underline{\leftrightarrow} u$, D_n yields $s' =_{\rm AC} u$. We distinguish the two possible forms of s.

- $s =_{\rm AC} \alpha(t^{\oplus}u+u)+u$.

  Then $r^{\oplus}s+\alpha(t^{\oplus}u+u) \underline{\leftrightarrow} t^{\oplus}u$. Since $r^{\oplus}s+s \underline{\leftrightarrow} t^{\oplus}u+u$, this yields $(r+\alpha)(t^{\oplus}u+u) \underline{\leftrightarrow} t(t^{\oplus}u+u)$. Lemma 6 implies $r+\alpha \underline{\leftrightarrow} t$, so since $L(r+\alpha)=L(t)<m$, we obtain $r+\alpha \longrightarrow\!\!\!\!\to t$. Then Rule 9 can be applied to $r^{\oplus}s =_{\rm AC} r^{\oplus}(\alpha(t^{\oplus}u+u)+u)$. Since $r^{\oplus}s$ is a normal form, this is a contradiction.

- $s =_{\mathrm{AC}} t^{\oplus}u + u$.

  Then $r^{\oplus}s + t^{\oplus}u \leftrightarrow t^{\oplus}u$. Since $r^{\oplus}s + s \leftrightarrow t^{\oplus}u + u$, this yields $(r+t)(t^{\oplus}u+u) \leftrightarrow t(t^{\oplus}u+u)$. Lemma 6 implies $r + t \leftrightarrow t$, so since $L(r+t) = L(t) < m$, we obtain $r + t \twoheadrightarrow t$. Then Rule 10 can be applied to $r^{\oplus}s =_{\mathrm{AC}} r^{\oplus}(t^{\oplus}u + u)$, and once more we have found a contradiction.

4. D_{n+1} is true.

We may assume A_{n+1} and B_{n+1} and C_{n+1}. Let p and q be bisimilar normal forms with $L(p) = L(q) = m$ and $g(p) + g(q) = n + 1$. We want to prove $p =_{\mathrm{AC}} q$.

First, we show that each summand of p is bisimilar to a summand of q, and vice versa. Clearly, each atomic summand a of p corresponds with a summand a of q. We show that each non-atomic summand of p also corresponds to a summand of q.

Suppose that a summand rs of p has behaviour in common with two summands of q. If these summands are of the form tu and $t'u'$, then A_{n+1} implies $u =_{\mathrm{AC}} s =_{\mathrm{AC}} u'$, so that Rule 2 reduces this pair. If they are of the form tu and $t'^{\oplus}u'$, then A_{n+1} and B_{n+1} give $u =_{\mathrm{AC}} s =_{\mathrm{AC}} t'^{\oplus}u' + u'$, so that Rule 6 reduces this pair. Finally, if they are of the form $t^{\oplus}u$ and $t'^{\oplus}u'$, then B_{n+1} implies $t^{\oplus}u + u =_{\mathrm{AC}} s =_{\mathrm{AC}} t'^{\oplus}u' + u'$. This means $t^{\oplus}u =_{\mathrm{AC}} t'^{\oplus}u'$, so Rule 1 reduces this pair.

Similarly, if a summand $r^{\oplus}s$ of p has behaviour in common with two summands of q, we find using B_{n+1} and C_{n+1} that Rule 1, 2 or 6 can be applied to this pair.

So, since q is a normal form, the assumption of a non-atomic summand of p having behaviour in common with two summands of q leads to a contradiction. By symmetry, each non-atomic summand of q too can have behaviour in common with only one summand of p. So apparently, each non-atomic summand of p is bisimilar to a non-atomic summand of q and vice versa.

- Suppose that summands rs and tu are bisimilar. Then A_{n+1} implies $s =_{\mathrm{AC}} u$, so according to Lemma 6 $r \leftrightarrow t$. Since $L(r) = L(t) < m$, we obtain $r =_{\mathrm{AC}} t$.

- If summands rs and $t^{\oplus}u$ are bisimilar, then B_{n+1} implies $s =_{\mathrm{AC}} t^{\oplus}u+u$. Then $r(t^{\oplus}u+u) =_{\mathrm{AC}} rs \leftrightarrow t^{\oplus}u \leftrightarrow t(t^{\oplus}u+u)$, so Lemma 6 implies $r \leftrightarrow t$. Since $L(r) = L(t) < m$, this yields $r =_{\mathrm{AC}} t$. Hence $rs =_{\mathrm{AC}} t(t^{\oplus}u + u)$, so we can apply Rule 5 to rs. Contradiction.

- Finally, if summands $r^{\oplus}s$ and $t^{\oplus}u$ are bisimilar, then C_{n+1} says that they are of the same form.

Hence, p and q contain exactly the same summands. Rule 1 indicates that each of these summands occurs only once in both p and q, so $p =_{\mathrm{AC}} q$. $\square$

Corollary 11 *The TRS in Table 5 is confluent.*

Corollary 12 *The axioms A1-5 + BKS1-3 for BPA* are complete with respect to bisimulation equivalence.*

Proof. If two terms in BPA$^{\oplus}$ are bisimilar, then according to Theorem 10 their normal forms are of the same form. Since all the rewrite rules can be deduced from A1-5 + PI1-3, it follows that this is a complete axiom system for BPA$^{\oplus}$. Then A1-5 + BKS1-3 is a complete axiomatization for BPA*. $\square$

References

[1] J.C.M. Baeten, J.A. Bergstra, and J.W. Klop. Decidability of bisimulation equivalence for processes generating context-free languages. *Journal of the ACM*, 40(3):653–682, 1993.

[2] J.C.M. Baeten and C. Verhoef. A congruence theorem for structured operational semantics with predicates. In E. Best, editor, *Proceedings CONCUR'93,* Hildesheim, *LNCS 715*, pages 477–492. Springer-Verlag, 1993.

[3] J.A. Bergstra, I. Bethke, and A. Ponse. Process algebra with iteration and nesting. *The Computer Journal*, 37(4):243–258, 1994.

[4] D. Caucal. Graphes canoniques de graphes algébriques. *Theoretical Informatics and Applications*, 24(4):339–352, 1990.

[5] J.H. Conway. *Regular Algebra and Finite Machines.* Chapman and Hall, 1971.

[6] I.M. Copi, C.C. Elgot, and J.B. Wright. Realization of events by logical nets. *Journal of the ACM*, 5:181–196, 1958.

[7] W. J. Fokkink. A complete equational axiomatization for prefix iteration. *Information Processing Letters*, 52(6):333–337, 1994.

[8] W. J. Fokkink and R. J. van Glabbeek. Ntyft/tyxt rules reduce to ntree rules. *Information and Computation*, 126(1):1–10, 1996.

[9] J. F. Groote and F. W. Vaandrager. Structured operational semantics and bisimulation as a congruence, *Information and Computation*, 100(2):202–260, 1992.

[10] S.C. Kleene. Representation of events in nerve nets and finite automata. In *Automata Studies*, pages 3–41. Princeton University Press, 1956.

[11] D. Kozen. A completeness theorem for Kleene algebras and the algebra of regular events. *Information and Computation*, 110(2):366–390, 1994.

[12] R. Milner. A complete inference system for a class of regular behaviours. *Journal of Computer and System Sciences*, 28:439–466, 1984.

[13] D.M.R. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, *Proceedings 5th GI Conference*, *LNCS 104*, pages 167–183. Springer-Verlag, 1981.

[14] G.D. Plotkin. A structural approach to operational semantics. Report DAIMI FN-19, Aarhus University, 1981.

[15] V.N. Redko. On defining relations for the algebra of regular events. *Ukrainskii Matematicheskii Zhurnal*, 16:120–126, 1964. In Russian.

[16] A. Salomaa. Two complete axiom systems for the algebra of regular events. *Journal of the ACM*, 13(1):158–169, 1966.

[17] P. Sewell. Bisimulation is not finitely (first order) equationally axiomatisable. In *Proceedings 9th IEEE Symposium on Logic in Computer Science (LICS'94),* Paris, pages 62–70. IEEE Computer Society Press, 1994.

[18] D.R. Troeger. Step bisimulation is pomset equivalence on a parallel language without explicit internal choice. *Mathematical Structures in Computer Science*, 3:25–62, 1993.

[19] C. Verhoef. A general conservative extension theorem in process algebra. In E.-R. Olderog, editor, *Proceedings IFIP Conference on Programming Concepts, Methods and Calculi (PROCOMET'94)*, San Miniato, *IFIP Transactions* A-56, pages 149–168. Elsevier, 1994.

[20] H. Zantema. Termination of term rewriting by semantic labelling. *Fundamenta Informaticae*, 24(1,2):89–105, 1995.