Basic Process Algebra with Iteration: Completeness of its Equational Axioms

WAN FOKKINK* AND HANS ZANTEMA†

*CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands

†Utrecht University, Padualaan 14, 3508 TB Utrecht, The Netherlands

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 commutativity and associativity of the $+$.

1. INTRODUCTION

Kleene [8] defined a binary operator x^*y in the context of finite automata, which denotes the iterate of x and y. He 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 Kleene's star in the presence of an empty word. The unary Kleene star has been studied extensively ever since.

Redko [13] (see also [5]) proved for the unary Kleene star that a complete finite axiomatization for language equality does not exist. Salomaa [14] presented a complete finite axiomatization which incorporates one conditional axiom, namely (translated to our setting): $x = y \cdot x + z$ implies $x = y^*z$ if y does not incorporate the empty word ϵ, or in other words if $y + \epsilon$ is not equivalent to y. According to Kozen [9] this last property is not algebraic, in the sense that it is not preserved under substitution of terms for actions. He has proposed two alternative conditional axioms which do not have this drawback. However, these axioms are not sound in the setting of (strong) bisimulation equivalence.

Milner [10] 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 into Basic Process Algebra (BPA). They suggested three axioms BKS1-3 for BPA*, where axiom BKS1 is the one from Kleene, while their most advanced axiom BKS3 originates from [16]:

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

In this paper we will prove that BKS1-3, together with the axioms A1-5 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 this paper; we will apply the strategy of semantic labelling from [17]. Finally, we will show that bisimilar normal forms are syntactically equal modulo commutativity and associativity of the $+$. These results together imply that the axiomatization for BPA* from [3] is complete with respect to bisimulation equivalence. Moreover, the applied method yields an algorithm to decide in finite time whether or not two terms are bisimilar.

Sewell [15] has proved that if the deadlock δ is added to our syntax, then a complete finite equational axiomatization does not exist. In [7], however, it is shown that if sequential composition and iteration are replaced by their prefix counterparts, then six simple equational axioms are complete for this algebra.

2. BPA WITH BINARY KLEENE STAR

This section introduces the basic notions. For more details see [3]. BPA* assumes an alphabet A of atomic actions, together with three binary operators: alternative composition $+$, sequential composition $\cdot$, and binary Kleene star *. Table 1 presents an operational semantics for BPA* in Plotkin style [12]. The special symbol $\surd$ in this table represents (successful) termination.

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 $a \in A$:

$$p \;::=\; a \mid p + p \mid p \cdot p \mid p^*p$$

In the sequel the operator $\cdot$ will often be omitted, so pq

TABLE 1. Action rules for BPA*

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

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

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

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

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

TABLE 2. Axioms for BPA*

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	$x \cdot x^*y+y$	$=$	x^*y
BKS2	$x^*y \cdot z$	$=$	$x^*(yz)$
BKS3	$x^*(y \cdot (x+y)^*z+z)$	$=$	$(x+y)^*z$

denotes $p \cdot q$. As binding convention, * binds stronger than $\cdot$, which in turn binds stronger than $+$.

Process terms are considered modulo (strong) bisimulation equivalence [11]. Intuitively, two process terms are bisimilar if they have the same branching structure.

Definition 2.1 *Two processes p_0 and q_0 are called* bisimilar, *denoted by $p_0 \leftrightarrows q_0$, if there exists a symmetric relation R between processes such that:*

1. *$R(p_0, q_0)$,*
2. *if $R(p,q)$ and $p \xrightarrow{a} p'$, then there is a transition $q \xrightarrow{a} q'$ such that $R(p',q')$,*
3. *if $R(p,q)$ and $p \xrightarrow{a} \surd$, then $q \xrightarrow{a} \surd$.*

Since the action rules in Table 1 are in *path* format [2], it follows that bisimulation equivalence is a congruence with respect to all the operators, which means that if $p \leftrightarrows p'$ and $q \leftrightarrows q'$, then $p+q \leftrightarrows p'+q'$ and $pq \leftrightarrows p'q'$ and $p^*q \leftrightarrows p'^*q'$.

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 from [3]. Axiom BKS3 stems from [16]. In the sequel, $p = q$ will mean that this equality can be derived from the axioms for BPA*.

The axiomatization for BPA* is sound with respect to bisimulation equivalence, i.e. if $p = q$ then $p \leftrightarrows q$. 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 $p \leftrightarrows q$ then $p = q$.

3. A CONDITIONAL TRS FOR BPA$^\oplus$

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 $p \cdot p^* q$, and we will develop a TRS for the algebra BPA$^\oplus$. From now on, process terms are considered modulo commutativity and associativity of the $+$.

3.1. Turning round two rules for BPA

The axiom A3 yields the expected rewrite rule

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

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. Because for example, in order to reduce the term $a \cdot (a+b)^*c + b \cdot (a+b)^*c + c$ to the term $(a+b)^*c$ we need the reduction

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

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

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

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

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

So we opt for the rule

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

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 $x^*y + z \leftrightarrow x^*y$ if $y + z \leftrightarrow y$, 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 turned round the rule for A4. For example, the term $a^*(b + ce) + ce + de$ would have two different normal forms:

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

To avoid this complication, we replace iteration by an operator $x^{\oplus}y$, called *proper* iteration, which has the behaviour of $x \cdot x^*y$. (The standard notation for this construct would be x^+y, but we want to avoid ambiguous use of the $+$.) 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 obvious equivalence $x^*y \leftrightarrow x^{\oplus}y + y$. The axiomatization in Table 4 is complete for BPA$^{\oplus}$ if and only if the axiomatization in Table 2 is complete for BPA*.

TABLE 3. Action rules for proper iteration

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

TABLE 4. Axioms for proper iteration

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

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. Because then the term $(a^{\oplus}b)c+dc$ would have two different normal forms:

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

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

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

   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:

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

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

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

   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:

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

   Therefore, we define

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

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

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

   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. Because 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:

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

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

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

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

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

3.6. The entire TRS

The entire TRS is given once again in Table 5. It is easy to see that all rules can be deduced from BPA$^{\oplus}$. As usual, the rules are considered modulo commutativity and associativity of the $+$.

TABLE 5. Rewrite rules for BPA$^{\oplus}$

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 $x+y \twoheadrightarrow y$

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 some term has reductions p'' and p', then there exists a term q that reduces both to p'' and to p'), and *terminating* (i.e. there are no infinite reductions). Because then Newman's Lemma says that the TRS is confluent, so that the TRS yields unique normal forms.

Although our choice of rewrite rules has been motivated by the wish for a 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 to verify this property for all overlapping redexes, does not work in a conditional setting.

Example 3.1 Consider the TRS consisting 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 syntactically equal modulo commutativity and associativity of the $+$. Hence, confluence will simply be a consequence of this main theorem, together with the property of termination for the TRS.

3.7. Termination

Proving termination of the TRS in Table 5, modulo commutativity and associativity of the $+$, is a complicated matter. This is mainly due to the presence of Rule 7, in which the right-hand side can be obtained from the left-hand side by substituting terms for variables. A powerful technique for proving termination of TRSs that incorporate such rules is the one of *semantic labelling* [17], 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 [17] 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.

Theorem 3.1 *The TRS in Table 5 is terminating.*

Proof The method from [17] 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 commutative and associative with respect to the choice operator, which is essential in order to obtain the termination result modulo commutativity and associativity of this operator. 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.

As labels for the operators sequential composition and proper iteration we choose the positive natural numbers. 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 all positive natural numbers i and j.

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

The claim is that termination of $\bar{R}$ implies termination of R. Because 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 $\bar{R}$.

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

$$\begin{array}{ll}
 & w(p) \ + \ (i+j)w(q) \ + \ (i+j)(j+1)w(r) \\
\text{and} & w(p) \ + \ (i+j)w(q) \ + \ j(j+2)w(r)
\end{array}$$

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. ■

4. NORMAL FORMS DECIDE BISIMULATION EQUIVALENCE

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 provably equal by the axioms A1,2.

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. $p \leq p+q \qquad p < pq \qquad p < p^{\oplus}q$
   $q \leq p+q \qquad q < pq \qquad q < p^{\oplus}q$
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 $p(p^{\oplus}q+q) \leftrightarrow p^{\oplus}q$, 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 4.1 *If $p \leftrightarrow q$, 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 $p' \leftrightarrow q'$, and so $|p'| = |q'|$. Hence, $L(p) \leq L(q)$, and by symmetry $L(q) \leq L(p)$. ■

Let us deduce the inductive definition for L. Since $L(p+q)$ is the maximum of the collection

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

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

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

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

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

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

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

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:

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

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

Lemma 4.2 *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 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. ■

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 [4].

Lemma 4.3 *If $pr \leftrightarrow qr$, then $p \leftrightarrow q$.*

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; A transition $p' \xrightarrow{a} p''$ in p is mimicked by a transition $q' \xrightarrow{a} q''$ in q, and vice versa. ■

Definition 4.1 *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 $p' \leftrightarrow q'$.*

Lemma 4.4 *If two terms pq and rs have behaviour in common, and $|q| \geq |s|$, then either $q \leftrightarrow ts$ for some t or $q \leftrightarrow s$.*

Proof If $pq \xrightarrow{a} q$ and $rs \xrightarrow{a} r's$ with $q \leftrightarrow r's$, or if $pq \xrightarrow{a} q$ and $rs \xrightarrow{a} s$ with $q \leftrightarrow s$, then we are done. And $pq \xrightarrow{a} p'q$ and $rs \xrightarrow{a} s$ with $p'q \leftrightarrow s$ would contradict $|q| \geq |s|$.

Thus, the only interesting case is $pq \xrightarrow{a} p'q$ and $rs \xrightarrow{a} r's$ with $p'q \leftrightarrow r's$. The inequality $|q| \geq |s|$ yields $|p'| \leq |r'|$. We show, with induction on $|p'|$, that $p'q \leftrightarrow r's$ together with $|p'| \leq |r'|$ indicates either $q \leftrightarrow ts$ for some t or $q \leftrightarrow s$.

If $|p'| = 1$, then $p' \xrightarrow{a} \surd$, and so $p'q \xrightarrow{a} q$. Since $p'q \leftrightarrow r's$, this transition can be mimicked by a transition $r's \xrightarrow{a} r''s$ or $r's \xrightarrow{a} s$, and so $q \leftrightarrow r''s$ or $q \leftrightarrow s$, respectively.

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

Lemma 4.5 *If pq or $p^\oplus q$ is a normal form, then q is* not *a normal form of a term rs.*

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 commutativity and associativity of the $+$. From now on, this equivalence is denoted by $p \cong q$, and we say that p and q are of the same form. Clearly, each process term p is of the form $a_1 + \ldots + a_k + p_1q_1 + \ldots + p_lq_l + r_1{}^\oplus s_1 + \ldots + r_m{}^\oplus s_m$. The terms a_i and p_iq_i and $r_i{}^\oplus s_i$ are called the *summands* of p.

Theorem 4.6 *If two normal forms p and q are bisimilar, then they are of the same form.*

Proof In order to prove the theorem, we will prove three extra statements in parallel, namely:

A. If two normal forms $p \equiv rs$ and $q \equiv tu$ have common behaviour, then $s \cong u$.

B. If two normal forms $p \equiv rs$ and $q \equiv t^\oplus u$ have common behaviour, then $s \cong t^\oplus u + u$.

C. If two normal forms $p \equiv r^\oplus s$ and $q \equiv t^\oplus u$ have common behaviour, then $r^\oplus s \cong t^\oplus u$.

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 for if $L(p)$ and $L(q)$ are smaller than m. We will prove it for the case that they are smaller or equal 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 $s \cong u$. Without loss of generality we may assume $|s| \geq |u|$, so Lemma 4.4 offers two possibilities.

1.1 $s \leftrightarrow u$.

$L(s) \leq L(rs) \leq m$ and $L(u) \leq L(tu) \leq m$ and $g(s) + g(u) < g(rs) + g(tu) = n + 1$. Hence, D_n yields $s \cong u$.

1.2 $s \leftrightarrow vu$ for some v.

Let w be a normal form of vu. According to Lemma 4.2 $g(w) \leq g(vu)$, so $g(s) + g(w) < g(rs) + g(vu) = n + 1$. Further, since $s \leftrightarrow w$, $L(w) = L(s) \leq m$. Hence, D_n yields $s \cong w$. However, Lemma 4.5 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 $s \cong t^{\oplus} u + u$. Since $t^{\oplus}u \leftrightarrow t(t^{\oplus}u + u)$, Lemma 4.4 offers three possibilities.

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

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 gives $s \cong t^{\oplus}u + u$.

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

This implies $v's \leftrightarrow u$ for some v', and we get a contradiction as in 1.2.

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

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

If $v \leftrightarrow t$ then $s \leftrightarrow t^{\oplus}u$, so that D_n yields $s \cong t^{\oplus}u$. But then Rule 7 reduces rs, so apparently v cannot be bisimilar with t. So if v is a normal form, $v(t^{\oplus}u + u)$ is a normal form too.

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 $\beta \cong t^{\oplus}u + u$.

Next, consider a summand $\alpha^{\oplus}\beta$ of s. This term and $v(t^{\oplus}u + u)$ have behaviour in common, so Lemma 4.4 offers three possibilities.

- $\alpha^{\oplus}\beta + \beta \leftrightarrow t^{\oplus}u + u$.
  We have $g(\alpha^{\oplus}\beta+\beta)+g(t^{\oplus}u+u) \leq g(s)+g(t^{\oplus}u) = n$, so D_n implies $\alpha^{\oplus}\beta+\beta \cong t^{\oplus}u+u$. 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 $\alpha^{\oplus}\beta \cong t^{\oplus}u$.
- $w(\alpha^{\oplus}\beta + \beta) \leftrightarrow t^{\oplus}u + u$ for some w.
  Then $w'(\alpha^{\oplus}\beta + \beta) \leftrightarrow u$ for some w', and we get a contradiction as in 2.2.
- $\alpha^{\oplus}\beta + \beta \leftrightarrow w(t^{\oplus}u + u)$ for some w.
  Then $\beta \leftrightarrow w'(t^{\oplus}u+u)$ for some w', and again we get a contradiction as in 2.2.

So we may conclude $\alpha^{\oplus}\beta \cong t^{\oplus}u$.

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 $t^{\oplus}u$. But then we can apply Rule 3 or 7 to rs, and again this is a contradiction.

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 $r^{\oplus}s \cong t^{\oplus}u$. Without loss of generality we assume $|r^{\oplus}s| \geq |t^{\oplus}u|$, so once more Lemma 4.4 offers two possibilities.

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

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

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

First, suppose that s and u have no behaviour in common with $t^{\oplus} u$ and $r^{\oplus} s$, respectively, so that $s \leftrightarrow u$ and $r^{\oplus}s \leftrightarrow t^{\oplus}u$. Since D_n applies to the first equivalence, we get $s \cong u$. And the second equivalence yields $r(r^{\oplus}s + s) \leftrightarrow t(t^{\oplus}u + u) \leftrightarrow t(r^{\oplus}s + s)$, so Lemma 4.3 implies $r \leftrightarrow t$. Since $L(r) = L(t) < m$, statement D then gives $r \cong t$, and we are done.

So without loss of generality we may assume that s and $t^{\oplus}u$ have behaviour in common. If a summand $\alpha\beta$ or $\gamma^{\oplus}\delta$ of s has behaviour in common with $t^{\oplus}u$, then B_n or C_n implies $\beta \cong t^{\oplus}u + u$ or $\gamma^{\oplus}\delta \cong t^{\oplus}u$, 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 u has a summand of the form $\beta(r^{\oplus}s + s)$ or $r^{\oplus}s$, which indicates that u has greater size than s. But on the other hand, s has a summand $\alpha(t^{\oplus}u + u)$ or $t^{\oplus}u$, so s has size greater than u. This cannot be, so u and $r^{\oplus}s$ can 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 $\gamma(t^\oplus u+u)$ or $t^\oplus u$. Again we establish a contradiction; u has greater size than itself.

So, s is of the form $\alpha(t^\oplus u+u)+s'$ or $t^\oplus u+s'$, where $s' \leftrightarrow u$, and $r^\oplus s+\alpha(t^\oplus u+u)$ or $r^\oplus s+t^\oplus u$ is bisimilar with $t^\oplus u$. From D_n it follows that $s' \cong u$. We distinguish the two possible forms of s:

- $s \cong \alpha(t^\oplus u+u)+u$.
  Then $r^\oplus s+\alpha(t^\oplus u+u) \leftrightarrow t^\oplus u$, and so, since $r^\oplus s+s \leftrightarrow t^\oplus u+u$, we have $(r+\alpha)(t^\oplus u+u) \leftrightarrow t(t^\oplus u+u)$. Then Lemma 4.3 implies $r+\alpha \leftrightarrow t$, so since $L(r+\alpha)=L(t)<m$, we obtain $r+\alpha \twoheadrightarrow t$. But then Rule 9 can be applied to $r^\oplus s \cong r^\oplus(\alpha(t^\oplus u+u)+u)$. Since $r^\oplus s$ is a normal form, this is a contradiction.
- $s \cong t^\oplus u+u$.
  Then $r^\oplus s+t^\oplus u \leftrightarrow t^\oplus u$, and so, since $r^\oplus s+s \leftrightarrow t^\oplus u+u$, we have $(r+t)(t^\oplus u+u) \leftrightarrow t(t^\oplus u+u)$. This implies $r+t \leftrightarrow t$, so since $L(r+t)=L(t)<m$, we obtain $r+t \twoheadrightarrow t$. But then Rule 10 can be applied to $r^\oplus s \cong r^\oplus(t^\oplus u+u)$, and once more we have 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 $p \cong q$.

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 now show that each non-atomic summand of p is also bisimilar 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 $u \cong s \cong u'$, so that Rule 2 reduces this pair. And if they are of the form tu and $t'^\oplus u'$, then A_{n+1} and B_{n+1} give $u \cong s \cong t'^\oplus u'+u'$, 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 $t^\oplus u+u \cong s \cong t'^\oplus u'+u'$. This means $t^\oplus u \cong t'^\oplus u'$, 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 with a non-atomic summand of q and vice versa.

- Suppose that summands rs and tu are bisimilar. Then A_{n+1} implies $s \cong u$, so according to Lemma 4.3 $r \leftrightarrow t$. Since $L(r)=L(t)<m$, we obtain $r \cong t$.
- If summands rs and $t^\oplus u$ are bisimilar, then B_{n+1} implies $s \cong t^\oplus u+u$. So $r(t^\oplus u+u) \cong rs \leftrightarrow t^\oplus u \leftrightarrow t(t^\oplus u+u)$, and Lemma 4.3 implies $r \leftrightarrow t$. Since $L(r)=L(t)<m$, this yields $r \cong t$. Hence, $rs \cong t(t^\oplus u+u)$. But then we can apply Rule 5 to rs; contradiction.
- Finally, if summands $r^\oplus s$ of p and $t^\oplus u$ of q 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 $p \cong q$. ■

Corollary 4.7 *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 4.6 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 clearly A1-5 + BKS1-3 is a complete axiomatization for BPA*. ■

ACKNOWLEDGEMENTS

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

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, ed., *Proceedings CONCUR'93,* Hildesheim, *LNCS 715*, pp. 477–492. Springer-Verlag, Berlin 1993.

[3] J. A. Bergstra, I. Bethke and A. Ponse. Process algebra with iteration and nesting. This issue.

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

[5] J. H. Conway. *Regular Algebra and Finite Machines.* Chapman and Hall, London, 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 axiomatisation for prefix iteration. Report CS-R9415, CWI, Amsterdam, 1994.

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

[9] D. Kozen. A completeness theorem for Kleene algebras and the algebra of regular events. In *Proceedings LICS'91,* Amsterdam, pp. 214–225. IEEE Computer Society Press, New York, 1991.

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

[11] D. M. R. Park. Concurrency and automata on infinite sequences. In P. Deussen, ed., *5th GI Conference, LNCS 104*, pp. 167–183. Springer-Verlag, Berlin 1981.

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

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

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

[15] P. Sewell. Bisimulation is not (first order) equationally axiomatisable. Technical report, Department of Computer Science, Edinburgh University, 1993. To appear in *Proceedings LICS'94*.

[16] 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.

[17] H. Zantema. Termination of term rewriting by semantic labelling. Report RUU-CS-92-38, Department of Computer Science, Utrecht University, 1992. Revised version RUU-CS-93-24. To appear in *Fundamenta Informaticæ* (special issue on term rewriting systems).