Basic Process Algebra with Iteration: Completeness of its Equational Axioms

W.J. Fokkink and H. Zantema

Basic Process Algebra with Iteration: Completeness of its Equational Axioms

W.J. Fokkink and H. Zantema

Department of Computer Science
Utrecht University
P.O.Box 80.089
3508 TB Utrecht
The Netherlands

Basic Process Algebra with Iteration:
Completeness of its Equational Axioms

Willem Jan Fokkink* Hans Zantema†
CWI Utrecht University

Abstract

Bergstra, Bethke & Ponse [BBP93] proposed an axiomatisation for Basic Process Algebra extended with iteration. In this paper, we prove that this axiomatisation is complete with respect to 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.

1 Introduction

Kleene [Kle56] defined a binary operator * in the context of finite automata, where E^*F denotes the *iterate* of E and F. He formulated some algebraic laws for this operator, notably, in his notation, $E^*F = F \vee E(E^*F)$. He also noted the correspondence of the constructs $E \vee F$ and EF with the conventions of sum and product respectively in algebra.

Milner [Mil84] studied a unary interpretation of Kleene's star, and raised the question whether there exists a complete axiomatisation for this operator.

Bergstra, Bethke & Ponse [BBP93] incorporated the binary Kleene star into Basic Process Algebra (BPA), and called it *single exit iteration* (SEI). They suggested three axioms SEI1-3 for BPA*, where axiom SEI1 is the one from Kleene, while their most advanced axiom

$$\text{SEI3} \quad x^*(y((x + y)^*z) + z) = (x + y)^*z$$

originates from [Trö93], where this equation was proposed in the setting of a specification language with a construct y *while* x, equivalent to x^*y.

In this paper we will prove that SEI1-3, together with the axioms A1-5 for BPA, form a complete axiomatisation for BPA* with respect to bisimulation equivalence. For this purpose, we will replace SEI by *proper iteration* (PI) $x^\oplus y$, which executes x at least one time; $x^\oplus y$ is defined by $x(x^*y)$. The axioms SEI1-3 are translated 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 [Zan93]. Finally, we will show that bisimilar normal forms are syntactically equal modulo commutativity and associativity of the +.

---
*P.O. Box 94079, 1090 GB Amsterdam, The Netherlands, e-mail: wan@cwi.nl
†P.O. Box 80089, 3508 TB Utrecht, The Netherlands, e-mail: hansz@cs.ruu.nl

2 BPA with Single Exit Iteration

This section introduces the basic notions. For more detailed information we refer to [BBP93].

In BPA*, we assume an alphabet A of atomic actions, together with three binary operators: alternative composition +, sequential composition ·, and single exit iteration *. Table 1 presents an operational semantics for BPA* in Plotkin style [Plo81], taken from [BBP93].

$$a \xrightarrow{a} \sqrt{}$$

$$\frac{x \xrightarrow{a} x'}{x+y \xrightarrow{a} x' \quad y+x \xrightarrow{a} x'} \qquad \frac{x \xrightarrow{a} \sqrt{}}{x+y \xrightarrow{a} \sqrt{} \quad y+x \xrightarrow{a} \sqrt{}}$$

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

$$\frac{x \xrightarrow{a} x'}{x^*y \xrightarrow{a} x'(x^*y)} \qquad \frac{x \xrightarrow{a} \sqrt{}}{x^*y \xrightarrow{a} x^*y}$$

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

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 · will often be omitted, so pq denotes $p \cdot q$. As bracket convention, · and * will bind stronger than +.

Process terms are considered modulo *(strong) bisimulation equivalence* [Par81]. 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 $p_0 \leftrightarrow q_0$, if there exists a symmetric relation R between processes such that:*

1. $R(p_0, q_0)$,

2. if $p \xrightarrow{a} p'$ and $R(p,q)$, then there is a transition $q \xrightarrow{a} q'$ such that $R(p',q')$,

3. if $p \xrightarrow{a} \sqrt{}$ and $R(p,q)$, then $q \xrightarrow{a} \sqrt{}$.

$$
\begin{array}{lrcl}
\text{A1} & x+y & = & y+x \\
\text{A2} & (x+y)+z & = & x+(y+z) \\
\text{A3} & x+x & = & x \\
\text{A4} & (x+y)z & = & xz+yz \\
\text{A5} & (xy)z & = & x(yz) \\
\\
\text{SEI1} & x(x^*y)+y & = & x^*y \\
\text{SEI2} & x^*(yz) & = & (x^*y)z \\
\text{SEI3} & x^*(y((x+y)^*z)+z) & = & (x+y)^*z
\end{array}
$$

Table 2: Axioms for BPA*

Since the action rules in Table 1 are in *path* format [BV93], it follows that bisimulation equivalence is a *congruence* with respect to all the operators, i.e. if $p \leftrightarrow p'$ and $q \leftrightarrow q'$, then $p+q \leftrightarrow p'+q'$ and $pq \leftrightarrow p'q'$ and $p^*q \leftrightarrow p'^*q'$.

Table 2 contains an axiom system for BPA*, which originates from [BBP93]. It consists of the axioms A1-5 for BPA together with three axioms SEI1-3 for iteration. In the sequel, $p = q$ will mean that this equality can be derived from axioms A1-5 and SEI1-3. This axiomatisation for BPA* is *sound* with respect to bisimulation equivalence, i.e. if the equality $p = q$ can be derived from these axioms, then $p \leftrightarrow q$. 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 axiomatisation is *complete* with respect to bisimulation, i.e. if $p \leftrightarrow q$ 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 SEI. 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$^\oplus$. ¿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

$$x + x \longrightarrow x$$

Usually, in BPA, the axiom A4 as a rewrite rule aims from left to right. However, the following example learns that in BPA* we need this rewrite rule in the opposite direction.

Example 3.1 *Consider the term $a((a+b)^*c) + b((a+b)^*c) + c$. In order to reduce this term to $(a+b)^*c$, we need the reduction*

$$a((a+b)^*c) + b((a+b)^*c) \longrightarrow\!\!\!\rightarrow (a+b)((a+b)^*c)$$

(End example)

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

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

In BPA, the axiom A5 aims from left to right too, but since we have reversed A4, we must do the same for A5. The next example shows that the TRS would not be confluent otherwise.

Example 3.2 *Suppose that A5 rewrites from left to right. Then the term $(ab)d + (ac)d$ has two different normal forms:*

$$a(bd) + a(cd) \longleftarrow (ab)d + (ac)d \longrightarrow (ab+ac)d$$

(End example)

So we opt for the rule

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

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 $x^*y + y \leftrightarrow x^*y$, these terms should have the same normal form. A rule $x^*y \longrightarrow x^*y + y$ does not terminate, so one would expect a rule $x^*y + y \longrightarrow x^*y$. However, this last rule clashes with the rule for A4; we give a small example.

Example 3.3 *If we add the rule $x^*y + y \longrightarrow x^*y$ to our TRS, then the term $a^*(bd) + bd + cd$ has two different normal forms:*

$$a^*(bd) + cd \longleftarrow a^*(bd) + bd + cd \longrightarrow a^*(bd) + (b+c)d$$

(End example)

To avoid this complication, we replace SEI by an operator $x^\oplus y$, called *proper iteration* PI, which has the behaviour of $x(x^*y)$. (The standard notation for this construct would be x^+y, but we want to avoid ambiguous use of the $+$.) The operational semantics and the axiomatisation for PI are given in Tables 3 and 4. They are obtained from the action rules and axioms for SEI, using the obvious equivalence $x^*y \leftrightarrow x^\oplus y + y$. Clearly, the axiomatisation in Table 4 is complete for BPA$^\oplus$ if and only if the axiomatisation in Table 2 is complete for BPA*.

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

Table 3: Action rules for proper iteration

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

Table 4: Axioms for proper iteration

3.3 One rule for axiom PI2

Now that we have replaced SEI by PI, 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. The next example shows that if it would rewrite from right to left, it would clash with the rule for A4.

Example 3.4 *If we add the rule* $(x^{\oplus}y)z \longrightarrow x^{\oplus}(yz)$ *to our TRS, then the term* $(a^{\oplus}b)c + dc$ *has two different normal forms:*

$$a^{\oplus}(bc) + dc \longleftarrow (a^{\oplus}b)c + dc \longrightarrow\!\!\!\rightarrow (a^{\oplus}b + d)c$$

(End example)

Hence, PI2 yields the rule

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

3.4 Four rules for axiom PI1

The next rule stems from axiom PI1.

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

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.

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

So for the sake of confluence, one of these two reducts should reduce to the other. The next example shows that a rule $(x + y)(y^{\oplus}z + z) \longrightarrow x(y^{\oplus}z + z) + y^{\oplus}z$ would clash with the rule for A4.

Example 3.5 *If we add the rule* $(x+y)(y^\oplus z + z) \longrightarrow x(y^\oplus z + z) + y^\oplus z$ *to our TRS, then the term* $(ac + bc)((bc)^\oplus d + d)$ *has two different normal forms:*

$$(ac)((bc)^\oplus d + d) + (bc)^\oplus d \longleftarrow (ac + bc)((bc)^\oplus d + d) \longrightarrow ((a+b)c)((bc)^\oplus d + d)$$

(End example)

Hence, we opt for the rule

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

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

$$x(y^\oplus z) \longleftarrow x(y(y^\oplus z + z)) \longrightarrow (xy)(y^\oplus z + z)$$

A rule $(xy)(y^\oplus z + z) \longrightarrow x(y^\oplus z)$ would clash with the rule for A5, which is shown by the next example.

Example 3.6 *If we add the rule* $(xy)(y^\oplus z + z) \longrightarrow x(y^\oplus z)$ *to our TRS, then the term* $(a(bc))((bc)^\oplus d + d))$ *has two different normal forms:*

$$a((bc)^\oplus d) \longleftarrow (a(bc))((bc)^\oplus d + d)) \longrightarrow ((ab)c)((bc)^\oplus d + d))$$

(End example)

Therefore, we define

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

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

$$x^\oplus(y^\oplus z) \longleftarrow x^\oplus(y(y^\oplus z + z)) \longrightarrow (x^\oplus y)(y^\oplus z + z)$$

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

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

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 into a conditional rule:

$$\mathbf{x^\oplus(x'(y^\oplus z + z) + z) \longrightarrow x(y^\oplus z + z)} \qquad \mathbf{if\ x + x' \longrightarrow\!\!\!\!\rightarrow y}$$

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

$$x^\oplus(y^\oplus z + z) \longleftarrow x^\oplus(y(y^\oplus z + z) + z) \longrightarrow x(y^\oplus z + z)$$

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

$$\mathbf{x^\oplus(y^\oplus z + z) \longrightarrow x(y^\oplus z + z)} \qquad \mathbf{if\ x + y \longrightarrow\!\!\!\!\rightarrow y}$$

3.6 The complete TRS

The complete TRS is given once again in Table 5. It is easy to see that all the rules can be deduced from BPA*, so they are sound with respect to bisimulation equivalence.

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$^{\oplus}$

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 $p'' \longleftarrow p \longrightarrow p'$, then there exists a term q such that $p'' \twoheadrightarrow q \twoheadleftarrow p'$), 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 verifying this property for all overlapping redexes, does not work in a conditional setting.

Example 3.7 *Consider the TRS consisting of the rules*

$$f(x) \longrightarrow b \quad \text{if } x \twoheadrightarrow a$$
$$a \longrightarrow c$$

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

(End example)

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 our TRS.

3.7   Termination

Proving termination of our TRS is a complicated matter, due to the presence of Rule 7; its left-hand side is in a sense 'smaller' than its right-hand side. A powerful technique for proving termination of TRSs that incorporate such rules is the one of *semantic labelling* [Zan93], where operation symbols occurring in the rewrite rules are supplied with labels, depending on the semantics of their arguments. Then two TRS's are involved: the original system and the labelled system. The main theorem of [Zan93] states that the labelled system terminates if and only if the original system terminates. The theory of semantic labelling is developed for unconditional TRS's. Though it will easily generalise to conditional systems, we do not need this. Let R be the unconditional system obtained by simply removing the conditions of the last two rules. We shall prove that R is terminating, from which termination of the conditional system of Table 5 easily follows.

The method of [Zan93] starts from giving a model for the TRS. This is an algebra over the signature with the property that for each rewrite rule and each choice of the variables the interpretation of the left-hand side is equal to the interpretation of the right-hand side. Here we choose the model to be the positive natural numbers, and 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:

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

Note that norm is commutative and associative with respect to the choice operator, which is essential for obtaining 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 9,10 of R, which are not sound because they lack their original conditions, preserve norm too.

If we define in the notation of [Zan93] $S.$ and $S_{\oplus}$ both to be the positive natural numbers, and define $\pi.(x,y) = \pi_{\oplus}(x,y) = y$ then we obtain the infinite TRS presented in Table 6, where i ranges over the positive natural numbers. Here sequential composition labelled by i is denoted by $\langle i \rangle$, and proper iteration labelled by i is denoted by $[i]$.

Now the claim is that termination of R follows from termination of $\bar{R}$. A sketch of the proof as given in [Zan93] can be given as follows. Assume that R admits an infinite reduction. By replacing all variables in this reduction by any constant a, we obtain an infinite ground reduction of R. For each symbol '.' and '$\oplus$' occurring in any term of this reduction, compute the value in the model of its right argument, i.e. the least number of steps in which this right argument can terminate. If the symbol is '.' and the corresponding value is i, then the symbol '.' is replaced by $\langle i \rangle$; if the symbol is '$\oplus$' and the corresponding value is i, then the symbol '$\oplus$' is replaced by $[i]$. If this is done properly for all symbols '.' and '$\oplus$' then it can be checked that each ground reduction step in R transforms to a ground reduction step in $\bar{R}$. Hence the infinite ground reduction of R transforms to an infinite ground reduction of $\bar{R}$, contradicting termination of $\bar{R}$.

1.	$x + x$	$\longrightarrow$	x
2.	$x\langle i\rangle z + y\langle i\rangle z$	$\longrightarrow$	$(x + y)\langle i\rangle z$
3.	$x\langle i + j\rangle(y\langle j\rangle z)$	$\longrightarrow$	$(x\langle i\rangle y)\langle j\rangle z$
4.	$x[i + j](y\langle j\rangle z)$	$\longrightarrow$	$(x[i]y)\langle j\rangle z$
5.	$x\langle i\rangle(x[i]y + y)$	$\longrightarrow$	$x[i]y$
6.	$x\langle i\rangle(y[i]z + z) + y[i]z$	$\longrightarrow$	$(x + y)\langle i\rangle(y[i]z + z)$
7.	$x\langle i + j\rangle(y[j]z)$	$\longrightarrow$	$(x\langle i\rangle y)\langle j\rangle(y[j]z + z)$
8.	$x[i + j](y[j]z)$	$\longrightarrow$	$(x[i]y)\langle j\rangle(y[j]z + z)$
9.	$x[i](x'\langle i\rangle(y[i]z + z) + z)$	$\longrightarrow$	$x\langle i\rangle(y[i]z + z)$
10.	$x[i](y[i]z + z)$	$\longrightarrow$	$x\langle i\rangle(y[i]z + z)$

Table 6: Rewrite rules with semantic labels: the system $\bar{R}$

Now 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. One way to do this is by defining inductively a weight function w:

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

It is easy to verify that for any choice of values for variables and any rule, the weight of the left-hand side is strictly greater than the weight of the right-hand side. Due to the strict monotonic behaviour of this weight function (here it is essential that $i > 0$), we conclude that each reduction step yields a strict decrease of the weight. Hence the system $\bar{R}$ is terminating.

Combining these observations we have proved the following theorem.

Theorem 3.8 *The TRS for* BPA* *in Table 5 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 axiomatisation 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. $\quad p \leq p + q \qquad p < pq \qquad p < p^{\oplus}q$
   $\quad q \leq p + q \qquad q < pq \qquad q < p^{\oplus}q$

2. The ordering is preserved under bisimulation equivalence.

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, defined by

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

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 $L(q) \leq L(p)$. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad$ $\square$

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

$$\{|p'| \mid p' \text{ proper substate of } p\} \ \cup \ \{|q'| \mid q' \text{ proper substate of } q\}$$

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

$$\{|p'q| \mid p' \text{ proper substate of } p\} \ \cup \ \{|q|\} \ \cup \ \{|q'| \mid q' \text{ proper substate of } q\}$$

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

$$\{|p'(p^{\oplus}q + q)| \mid p' \text{ proper substate of } p\} \ \cup \ \{|p^{\oplus}q + q|\} \ \cup \ \{|q'| \mid q' \text{ proper substate of } q\}$$

Since $|p^{\oplus}q + q| = |q|$, it follows that $|p^{\oplus}q| = max\{L(p) + |q|, L(q)\}$. 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 has normal form 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 max and $\lambda x, y \cdot y + 1$ as used in the definition of g are weakly monotonous in both coordinates, we may conclude that g-value is never increased by a rewrite step. Hence, in a reduction to normal form it does not increase either. $\quad\square$

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 [BBK93], i.e. for processes that are able to terminate in finitely many transitions. This lemma originates from [Cau90].

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

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

Definition 4.4 *We say that two process terms p and q have behaviour in common if either there are p' and q' such that $p \overset{a}{\longrightarrow} p'$ and $q \overset{a}{\longrightarrow} q'$ and $p' \leftrightarrow q'$, or $p \overset{a}{\longrightarrow} \sqrt{}$ and $q \overset{a}{\longrightarrow} \sqrt{}$.*

Lemma 4.5 *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 $pq \overset{a}{\longrightarrow} q$ and $rs \overset{a}{\longrightarrow} r's$ with $q \leftrightarrow r's$, or if $pq \overset{a}{\longrightarrow} q$ and $rs \overset{a}{\longrightarrow} s$ with $q \leftrightarrow s$, then we are done. Thus, the only interesting case is if $pq \overset{a}{\longrightarrow} p'q$ and $rs \overset{a}{\longrightarrow} r's$ with $p'q \leftrightarrow r's$. The inequality $|q| \geq |s|$ yields $|p'| \leq |r'|$. We apply induction to $|p'|$.

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

Next, let $|p'| = n+1$. Clearly, there is a transition $p' \overset{a}{\longrightarrow} p''$ with $|p''| = n$. Since $p'q \leftrightarrow r's$, the transition $p'q \overset{a}{\longrightarrow} p''q$ can be mimicked by a transition $r's \overset{a}{\longrightarrow} r''s$. Then $p''q \leftrightarrow r''s$, and $|r'| \geq n+1$ induces $|r''| \geq n$, so the induction hypothesis learns that either $q \leftrightarrow ts$ for some t or $q \leftrightarrow s$. $\quad\square$

Lemma 4.6 *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, and 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 $a_1 + ... + a_k + p_1q_1 + ... + p_lq_l + r_1{}^{\oplus}s_1 + ... + r_m{}^{\oplus}s_m$. The terms a_i and p_iq_i and $r_i{}^{\oplus}s_i$ are called the *summands* of p.

Theorem 4.7 *If two normal forms p and q are bisimilar, then they are of the same form.*

Proof. If $L(p) = L(q) = 0$, then both p and q must be sums of atoms. 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. Hence $p \cong q$.

Next, fix an $m > 0$ and assume that we have already proved the theorem for bisimilar normal forms p and q with $L(p) = L(q) < m$. We will prove it for $L(p) = L(q) = m$. In order to do so we need the following statements.

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 will be labelled by D. 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 to n.

A_0 and B_0 and C_0 are trivially true, since they are empty statements. And D_0 corresponds with the dealt case $L(p) = L(q) = 0$, because if $g(p) + g(q) = 0$, then both p and q must be sums of atoms.

As the induction hypothesis we assume A_n, B_n, C_n and D_n; 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 4.5 offers two possibilities.

1.1 $s \underleftrightarrow{} u$.

$L(s) \leq L(rs) \leq m$ 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 $s \underleftrightarrow{} vu$ 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 4.6 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 4.5 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 4.5 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 we have 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 4.5 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 $r + \alpha \longrightarrow\!\!\!\!\rightarrow t$. 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 $r + t \longrightarrow\!\!\!\!\rightarrow t$. 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 $p \cong q$.

Since p and q are bisimilar, each summand of p has behaviour in common with a summand of q and vice versa. We show that a summand of p can have behaviour in common with only one summand of q.

A summand a of p can only have behaviour in common with a summand a of q. According to Rule 1, q can contain only one summand a.