Basic process algebra with iteration: completeness of its equational axioms

W.J. Fokkink, H. Zantema

Computer Science/Department of Software Technology

CS-R9368 1993

Basic Process Algebra with Iteration: Completeness of its Equational Axioms

Willem Jan Fokkink
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
e-mail: `wan@cwi.nl`

Hans Zantema
Utrecht University
P.O. Box 80089, 3508 TB Utrecht, The Netherlands
e-mail: `hansz@cs.ruu.nl`

Abstract

Bergstra, Bethke & Ponse [BBP93] proposed an axiomatisation for Basic Process Algebra extended with (binary) iteration. In this paper, we prove that this axiomatisation 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 [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. Copi, Elgot & Wright [CEW58] 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 ([Red64], see also [Con71]) proved for the unary Kleene star that a complete finite axiomatisation for language equality does not exist. Salomaa [Sal66] presented a complete finite axiomatisation which incorporates one conditional axiom, namely (translated to our setting):

$$x = y \cdot x + z \text{ and } y \text{ does not have the empty word property} \implies x = y^*z$$

A process y has the *empty word property* if it incorporates the empty word ϵ, or in other words if $y + \epsilon$ is equivalent to y. According to Kozen [Koz91] this property is not algebraic, in the sense that it is not preserved under substitution of terms for actions. He has presented an alternative complete finite axiomatisation, again with conditional axioms, which does not have this drawback.

Milner [Mil84] studied Kleene's star in the setting of (strong) bisimulation equivalence, and raised the question whether there exists a complete axiomatisation for it.

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

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

originates from [Tro93], 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) $x^\oplus y$. This construct executes x at least one time, or in other words, $x^\oplus y$ is equivalent to $x \cdot x^*y$. The axioms SEI1-3 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 [Zan93]. Finally, we will show that bisimilar normal forms are syntactically equal modulo commutativity and associativity of the $+$. These results together imply that the axiomatisation for BPA* from [BBP93] 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.

To our knowledge, never before a finite equational axiom system was proved complete in the setting of Kleene's star. Sewell [Sew93] has proved that if the deadlock δ is added to our syntax, then a complete finite equational axiomatisation does not exist.

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

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 $\cdot$, and single exit iteration *. Table 1 presents an operational semantics for BPA* in Plotkin style [Plo81], taken from [BBP93]. The special symbol $\surd$ (pronounce 'tick') 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 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* [Par81]. Intuitively, two process terms are bisimilar if they have the same branching structure.

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

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

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

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

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

Table 1: Action rules for BPA*

Definition 2.1 *Two processes p_0 and q_0 are called* bisimilar*, denoted by $p_0 \underline{\leftrightarrow} q_0$, 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 $p \xrightarrow{a} \surd$ and $R(p,q)$, then $q \xrightarrow{a} \surd$.*

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 $p \underline{\leftrightarrow} p'$ and $q \underline{\leftrightarrow} q'$, then $p+q \underline{\leftrightarrow} p'+q'$ and $pq \underline{\leftrightarrow} p'q'$ and $p^*q \underline{\leftrightarrow} p'^*q'$.

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. Axiom SEI3 stems from [Tro93]. 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 $p = q$ 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 axiomatisation is *complete* with respect to bisimulation, i.e. if $p \underline{\leftrightarrow} q$ 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.

A1	$x+y$	$=$	$y+x$
A2	$(x+y)+z$	$=$	$x+(y+z)$
A3	$x+x$	$=$	x
A4	$(x+y)z$	$=$	$xz+yz$
A5	$(xy)z$	$=$	$x(yz)$
SEI1	$x\cdot x^*y+y$	$=$	x^*y
SEI2	$x^*y\cdot z$	$=$	$x^*(yz)$
SEI3	$x^*(y\cdot(x+y)^*z+z)$	$=$	$(x+y)^*z$

Table 2: Axioms for BPA*

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

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

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

$$a\cdot(a+b)^*c+b\cdot(a+b)^*c \twoheadrightarrow (a+b)\cdot(a+b)^*c$$

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. 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:*

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

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 turned round the rule for A4. This is shown by the following example.

Example 3.3 *Suppose that we add the proposed rewrite rule to our TRS. Then the term $a^*(b+ce)+ce+de$ has two different normal forms:*

$$a^*(b+ce)+de \;\longleftarrow\; a^*(b+ce)+ce+de \;\longrightarrow\; a^*(b+ce)+(c+d)e$$

To avoid this complication, we replace SEI by an operator $x^\oplus y$, called *proper iteration* PI, 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 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 $x^*y \underline{\leftrightarrow} x^\oplus y+y$. The axiomatisation in Table 4 is complete for BPA$^\oplus$ if and only if the axiomatisation in Table 2 is complete for BPA*.

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

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

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 left to right, 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:*

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

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.

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

   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:*

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

   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:

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

   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:*

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

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.

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

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

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

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:

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

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:

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

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

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

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$^{\oplus}$.

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*

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

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

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$^\oplus$

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. 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* [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, which immediately implies termination of the conditional system of Table 5.

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:

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

Note that norm is commutative and associative with respect to the choice operator, which is essential 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] $S_{\cdot}$ and $S_{\oplus}$ both to be the positive natural numbers, and define $\pi_{\cdot}(x,y) = \pi_{\oplus}(x,y) = y$ 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 '$\cdot$' 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 '$\cdot$' and the corresponding value is i, then the symbol '$\cdot$' 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 '$\cdot$' 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 $\bar{R}$.

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. Define inductively 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 and 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:

$$\begin{array}{rcl} w(p\langle i+j \rangle (q[j]r)) & = & w(p) \; + \; (i+j)w(q) \; + \; (i+j)(j+1)w(r) \\ w((p\langle i \rangle q)\langle j \rangle (q[j]r+r)) & = & w(p) \; + \; (i+j)w(q) \; + \; j(j+2)w(r) \end{array}$$

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.

Theorem 3.8 *The TRS R in Table 5 is terminating.*

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 $\bar{R}$

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. $p \leq p+q \qquad p < pq \qquad p < p^\oplus q$
   $q \leq p+q \qquad q < pq \qquad q < p^\oplus q$

2. The ordering is preserved under bisimulation 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 $p(p^\oplus q+q) \mathrel{\underline{\leftrightarrow}} p^\oplus q$, 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, 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 $p \underline{\leftrightarrow} q$, 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 $L(q) \leq L(p)$. □

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

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

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 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 $\lambda x, y \,.\, y+1$ 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. □

In the proof of the main theorem, in Section 4.3, we will apply induction to process terms using 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 [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 $p'r \xrightarrow{a} p''r$ 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. □

Definition 4.4 *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 4.5 *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 $pq \xrightarrow{a} q$ 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. Thus, the only interesting case is if $pq \xrightarrow{a} p'q$ and $rs \xrightarrow{a} r's$ with $p'q \leftrightarrow r's$. The inequality $|q| \geq |s|$ yields $|p'| \leq |r'|$. We apply induction on $|p'|$.

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 $p'q \leftrightarrow r's$, the transition $p'q \xrightarrow{a} p''q$ can be mimicked by a transition $r's \xrightarrow{a} r''s$. 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 $q \leftrightarrow s$. □

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 on n.

A_0 and B_0 and C_0 are trivially true, since they are empty statements. And D_0 corresponds to 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; 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 $s \underline{\leftrightarrow} u$.

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

1.2 $s \underline{\leftrightarrow} vu$ 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 $s \underline{\leftrightarrow} w$, $L(w) = L(s) \leq m$. 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 $t^{\oplus}u \underline{\leftrightarrow} t(t^{\oplus}u + u)$, Lemma 4.5 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 gives $s \cong t^{\oplus}u + u$.

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

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

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.

If $v \underline{\leftrightarrow} t$ then $s \underline{\leftrightarrow} t^{\oplus}u$, 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.

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

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

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

  Then $w'(\alpha^{\oplus}\beta + \beta) \underline{\leftrightarrow} u$ for some w', and we get a contradiction as in 2.2.

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

  Then $\beta \underline{\leftrightarrow} w'(t^{\oplus}u + u)$ for some w', and 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 $r^{\oplus}s + s \underline{\leftrightarrow} v(t^{\oplus}u + u)$ 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 $s \cong u$. 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 4.3 implies $r \underline{\leftrightarrow} t$. 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 $s' \underline{\leftrightarrow} u$, 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 $r^{\oplus}s+\alpha(t^{\oplus}u+u) \underline{\leftrightarrow} t^{\oplus}u$, and so, since $r^{\oplus}s+s \underline{\leftrightarrow} t^{\oplus}u+u$, we have $(r+\alpha)(t^{\oplus}u+u) \underline{\leftrightarrow} t(t^{\oplus}u+u)$. Then Lemma 4.3 implies $r+\alpha \underline{\leftrightarrow} t$, 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 $r^{\oplus}s+t^{\oplus}u \underline{\leftrightarrow} t^{\oplus}u$, and so, since $r^{\oplus}s+s \underline{\leftrightarrow} t^{\oplus}u+u$, we have $(r+t)(t^{\oplus}u+u) \underline{\leftrightarrow} t(t^{\oplus}u+u)$. This implies $r+t \underline{\leftrightarrow} t$, 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 $r \underline{\leftrightarrow} t$. 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 $r(t^{\oplus}u+u) \cong rs \underline{\leftrightarrow} t^{\oplus}u \underline{\leftrightarrow} t(t^{\oplus}u+u)$, and Lemma 4.3 implies $r \underline{\leftrightarrow} t$. 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 $p \cong q$. $\square$

Corollary 4.8 *The axiomatisation* A1-5 + SEI1-3 *for* BPA* *is complete with respect to bisimulation equivalence.*

Proof. If two terms in BPA$^{\oplus}$ are bisimilar, then according to Theorem 4.7 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 + SEI1-3 is a complete axiomatisation for BPA*. $\square$

References

[BBK93] 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.

[BBP93] J.A. Bergstra, I. Bethke, and A. Ponse. Process algebra with iteration and nesting. Report P9314b, Programming Research Group, University of Amsterdam, 1993. Revised version of Process algebra with iteration, Report P9314.

[BV93] 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.

[Cau90] D. Caucal. Graphes canoniques et graphes algébriques. *Theoretical Informatics and Applications*, 24(4):339–352, 1990.

[CEW58] I.M. Copi, C.C. Elgot, and J.B. Wright. Realization of events by logical nets. *Journal of the ACM*, 5:181–196, 1958.

[Con71] J.H. Conway. *Regular algebra and finite machines.* Chapman and Hall, 1971.

[Kle56] S.C. Kleene. Representation of events in nerve nets and finite automata. In *Automata Studies*, pages 3–41. Princeton University Press, 1956.

[Koz91] D. Kozen. A completeness theorem for Kleene algebras and the algebra of regular events. In *Proceedings* 6th *Annual Symposium on Logic in Computer Science,* Amsterdam, pages 214–225. IEEE Computer Society Press, 1991.

[Mil84] R. Milner. A complete inference system for a class of regular behaviours. *Journal of Computer and System Sciences*, 28:439–466, 1984.

[Par81] D.M.R. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, 5th *GI Conference*, LNCS 104, pages 167–183. Springer-Verlag, 1981.

[Plo81] G.D. Plotkin. A structural approach to operational semantics. Report DAIMI FN-19, Computer Science Department, Aarhus University, 1981.

[Red64] V.N. Redko. On defining relations for the algebra of regular events. *Ukrainskii Matematicheskii Zhurnal*, 16:120–126, 1964. In Russian.

[Sal66] A. Salomaa. Two complete axiom systems for the algebra of regular events. *Journal of the ACM*, 13(1):158–169, 1966.

[Sew93] P. Sewell. Bisimulation is not (first order) equationally axiomatisable. Technical report, Department of Computer Science, Edinburgh University, 1993. To appear in the proceedings of LICS'94.

[Tro93] 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.

[Zan93] H. Zantema. Termination of term rewriting by semantic labelling. Report RUU-CS-93-24, Department of Computer Science, Utrecht University, 1993.