ELSEVIER

Information Processing Letters 52 (1994) 333-337

Information Processing Letters

A complete equational axiomatization for prefix iteration

Wan Fokkink [1]

Centrum voor Wiskunde en Informatica, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

Communicated by H. Ganzinger; received 28 February 1994; revised 30 June 1994

Abstract

Prefix iteration a^*x is added to *Minimal Process Algebra* (MPA_δ), which is a subalgebra of BPA_δ equivalent to Milner's basic CCS. We present a finite equational axiomatization for MPA_δ^*, and 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 show that bisimilar terms have the same normal form.

Keywords: Concurrency; Formal languages; Programming calculi; Basic CCS; Iteration; Complete equational axioms

1. Introduction

Kleene [7] defined a binary operator $\_^*\_$ in the context of finite automata, called *Kleene star* or *iteration*. Intuitively, the expression p^*q yields a solution for the recursive equation $X = p \cdot X + q$. In other words, p^*q can choose to execute either p, after which it evolves into p^*q again, or q, after which it terminates.

Milner [11] studied the unary version p^* of the Kleene star in the setting of (strong) bisimulation equivalence, and raised the question whether there exists a complete axiomatization for it. Bergstra, Bethke and Ponse [1] incorporated the binary Kleene star into Basic Process Algebra (BPA) [2], and they suggested three equational axioms for iteration. Fokkink and Zantema [5] proved that these three axioms, together with the five standard axioms for BPA, are a complete axiomatization for BPA* modulo bisimulation.

In this paper, we add the deadlock δ to the syntax. Sewell [14] proved that there does not exist a complete finite equational axiomatization for BPA_δ^*. In order to prove a completeness result, nevertheless, we restrict the binary sequential composition $x \cdot y$ to its unary prefix version $a \cdot x$, to obtain *Minimal Process Algebra* MPA_δ, equivalent to basic CCS [10]. Likewise, we add prefix iteration a^*x to the syntax, resulting in the algebra MPA_δ^*. This algebra is less expressive than BPA_δ^*. For instance, it cannot express a simple process such as $(a+b)^*c$. On the other hand, it contains processes which can be expressed neither in BPA* nor in BPA_δ, such as $a^*\delta$.

We propose two simple equational axioms for iteration, which are actually instantiations of the first and the third axiom for the binary Kleene star. We prove that these two axioms, together with the four standard axioms of MPA_δ, are a complete axiomatization for MPA_δ^* with respect to bisimulation. The proof consists of producing a term rewriting system from the axioms, and showing that bisimilar normal forms are equal modulo AC. This method yields an algorithm to decide whether or not two terms are bisimilar.

[1] Email: wan@cwi.nl.

SSDI 0020-0190(94)00163-4

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

$$a \cdot x \xrightarrow{a} x$$

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

Fig. 1. Action rules for MPA^*_δ.

A1	$x + y = y + x$
A2	$(x + y) + z = x + (y + z)$
A3	$x + x = x$
A6	$x + \delta = x$
MI1	$a \cdot a^* x + x = a^* x$
MI2	$a^*(a^* x) = a^* x$

Fig. 2. Axioms for MPA^*_δ.

2. Minimal Process Algebra with iteration

We assume an alphabet A of atomic actions. The signature of the algebra $\mathrm{MPA}^*_\delta(A)$, or MPA^*_δ for short, consists of a constant δ, which represents deadlock, together with the binary alternative composition $x + y$, and the unary prefix sequential composition $a \cdot x$ and prefix iteration $a^* x$, for $a \in A$. Fig. 1 presents an operational semantics for MPA^*_δ in Plotkin style [13]. Prefix iteration $a^* x$ can choose to execute either a, after which it evolves into $a^* x$ again, or x.

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

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

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

Process terms are considered modulo (*strong*) *bisimulation equivalence* [12]. Intuitively, two process terms are bisimilar if they have the same branching structure.

Definition 1. Two processes p_0 and q_0 are called *bisimilar*, denoted by $p_0 \leftrightarroweq q_0$, if there exists a symmetric relation $\mathcal{B}$ on processes such that $p_0 \mathcal{B} q_0$, and if $p \xrightarrow{a} p'$ and $p \mathcal{B} q$, then there is a transition $q \xrightarrow{a} q'$ with $p' \mathcal{B} q'$.

The action rules in Fig. 1 are in the *tyft/tyxt* format of Groote and Vaandrager [6]. Hence, bisimulation equivalence is a congruence with respect to all the operators, i.e. if $p \leftrightarroweq p'$ and $q \leftrightarroweq q'$, then $p + q \leftrightarroweq p' + q'$ and $a \cdot p \leftrightarroweq a \cdot p'$ and $a^* p \leftrightarroweq a^* p'$. See [6] for the definition of the tyft/tyxt format, and for a proof of this congruence result. (This proof uses the extra assumption that the rules are *well-founded*. In [4] it is shown that this requirement can be dropped.)

Furthermore, the three rules for MPA_δ are *pure* and *well-founded*, and the two rules for iteration incorporate the Kleene star in the left-hand side of their conclusions. Hence, MPA^*_δ is an operationally conservative extension of MPA_δ, i.e. the action rules for iteration do not influence the transition systems of MPA_δ terms. See [6] for the definitions, and for a proof of this conservativity result.

Fig. 2 contains an axiom system for MPA^*_δ, which consists of the four axioms from MPA_δ together with two axioms for iteration. In the sequel, $p = q$ will mean that this equality can be derived from these axioms. Our axiomatization for MPA^*_δ is sound with respect to bisimulation equivalence, i.e. if $p = q$ then $p \leftrightarroweq q$. Since bisimulation is a congruence, this can be verified by checking soundness for each axiom separately. In this paper it is proved that the axiomatization is complete with respect to bisimulation, i.e. if $p \leftrightarroweq q$ then $p = q$.

3. A Term Rewriting System

Our aim is to prove that the axioms in Fig. 2 are complete for our model of MPA^*_δ modulo bisimulation. A standard scheme for such a proof is to set up a Term Rewriting System (TRS) from the axioms as follows.

1. Turn the axioms into rewrite rules.
2. Apply the *Knuth–Bendix* completion algorithm [9], which yields extra rewrite rules to make the TRS *weakly confluent*. That is, if a term p has

$$a^{\oplus}x \xrightarrow{a} a^{\oplus}x + x$$

PMI1 $a \cdot (a^{\oplus}x + x) = a^{\oplus}x$

PMI2 $a^{\oplus}(a^{\oplus}x + x) = a^{\oplus}x$

Fig. 3. Semantics and axioms for proper iteration.

one-step reductions p' and p'', then both terms can be reduced to a term q.

3. Check that the resulting TRS is *terminating*, which means that there are no infinite reductions.

If the TRS is weakly confluent and terminating, Newman's Lemma says that it supplies each term with a unique normal form. The construction of the TRS ensures that all its rules can be deduced from the axioms. The final step in the completeness proof is to show that bisimilar normal forms are syntactically equal.

See [3,8] for an overview of the field of term rewriting.

3.1. Proper iteration

We want to define a TRS for process terms that reduces bisimilar terms to the same normal form. However, it is not so easy to construct such a TRS for MPA^*_δ. Namely, the terms $a^*x + x$ and a^*x are bisimilar, so they should reduce to the same normal form. A rule $a^*x \to a^*x + x$ does not terminate, so we need the rule

$$a^*x + x \longrightarrow a^*x.$$

However, this rule is not yet sufficient, because it does not deal with the case $a^*(b^*x) + x \leftrightarrows a^*(b^*x)$. Hence, for this case we must introduce an extra rewrite rule. But this rule does not cover the case $a^*(b^*(c^*x)) + x \leftrightarrows a^*(b^*(c^*x))$, etc. So in order to obtain unique normal forms modulo bisimulation for MPA^*_δ, apparently we need an infinite number of rewrite rules.

To avoid this complication, we replace iteration by an equivalent operator $a^{\oplus}x$, which represents the behaviour of $a \cdot a^*x$. The construct $a^{\oplus}x$ is called *proper* iteration. (Its standard notation would be a^+x, but we want to avoid ambiguous use of the $+$.) The operational semantics and the axiomatization for proper iteration are given in Fig. 3. They are obtained from the action rules and axioms for MPA^*_δ, using the obvious equivalence $a^*x \leftrightarrows a^{\oplus}x + x$. Conversely $a^{\oplus}x \leftrightarrows a \cdot a^*x$, and

$$\mathrm{MPA}^*_\delta + (a^{\oplus}x = a \cdot a^*x) \vdash \mathrm{PMI}1,2,$$
$$\mathrm{MPA}^{\oplus}_\delta + (a^*x = a^{\oplus}x + x) \vdash \mathrm{MI}1,2.$$

So we find that the axiomatization in Fig. 3 is complete for $\mathrm{MPA}^{\oplus}_\delta$ if and only if the axiomatization in Fig. 2 is complete for MPA^*_δ.

1.	$x + x \longrightarrow x$
2.	$x + \delta \longrightarrow x$
3.	$a \cdot (a^{\oplus}x + x) \longrightarrow a^{\oplus}x$
4.	$a^{\oplus}(a^{\oplus}x + x) \longrightarrow a^{\oplus}x$
5.	$a \cdot (a^{\oplus}\delta) \longrightarrow a^{\oplus}\delta$
6.	$a^{\oplus}(a^{\oplus}\delta) \longrightarrow a^{\oplus}\delta$

Fig. 4. Rewrite rules for $\mathrm{MPA}^{\oplus}_\delta$.

3.2. The TRS for $\mathrm{MPA}^{\oplus}_\delta$

We want to find a TRS for $\mathrm{MPA}^{\oplus}_\delta$ that reduces bisimilar terms to the same normal form. In particular, the TRS should be terminating. Axioms A1,2 obstruct this property, so from now on process terms are considered modulo AC (that is, modulo associativity and commutativity of the $+$).

Fig. 4 contains a TRS for $\mathrm{MPA}^{\oplus}_\delta$, which is obtained in two steps. First, axioms A3,6 and MI1,2 are turned into rewrite rules, aiming from left to right. Next, the Knuth-Bendix completion algorithm is applied, which yields Rules 5 and 6. The resulting TRS in Fig. 4 is weakly confluent, and all its rules can be deduced from the axioms for $\mathrm{MPA}^{\oplus}_\delta$. Furthermore, in each rule the term at the left-hand side contains more symbols than the term at the right-hand side, so clearly the TRS is terminating. Thus, Newman's Lemma ensures that the TRS reduces each term to a unique normal form, modulo AC.

4. Normal forms decide bisimilarity

We have developed a TRS for $\mathrm{MPA}_\delta^\oplus$ that reduces terms to a unique normal form. Furthermore, all its rules can be deduced from the axioms of $\mathrm{MPA}_\delta^\oplus$. Therefore, all the rules are sound with respect to bisimulation equivalence, so each term is bisimilar with its normal form. Hence, in order to determine completeness of the axiomatization for $\mathrm{MPA}_\delta^\oplus$ with respect to bisimulation, it is sufficient to prove that if two normal forms are bisimilar, then they are equal modulo AC.

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

We present the proof of the completeness theorem, which is in fact a simplified version of the completeness proof in [5], with some minor extra cases to deal with deadlock. In the proof we apply induction on the following weight function on terms:

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

Theorem 2. *If two normal forms p and q are bisimilar, then $p =_{\mathrm{AC}} q$.*

Proof. We apply induction on $g(p)+g(q)$. If $g(p)+g(q) = 0$, then both p and q must be sums of δ. Since p and q are normal forms, Rule 1 ensures that both p and q are of the form δ, so $p =_{\mathrm{AC}} q$.

Now assume that we have already proved the theorem for bisimilar normal forms p and q with $g(p) + g(q) < n$, for some $n \geqslant 1$. We prove it for $g(p) + g(q) = n$, by showing that the separate bisimilar summands of p and q are of the same form. Since $g(p) + g(q) > 0$, clearly p and q are not bisimilar to δ. Then Rule 2 ensures that they do not contain any summands δ. This leaves the following three possibilities.

(1) First, suppose that summands $a \cdot r$ of p and $a \cdot s$ of q are bisimilar, so $r \leftrightarrows s$. Since $g(r) + g(s) < n$, the induction hypothesis yields $r =_{\mathrm{AC}} s$.

(2) Next, let summands $a \cdot r$ and $a^\oplus s$ be bisimilar, so $r \leftrightarrows a^\oplus s + s$. We deduce a contradiction.

If $s \neq_{\mathrm{AC}} \delta$, then $a^\oplus s + s$ is a normal form, because we cannot apply Rule 1 or 2 to $a^\oplus s + s$, and $a^\oplus s$ and s are normal forms. Moreover, $g(r) + g(a^\oplus s + s) < n$, so the induction hypothesis yields $r =_{\mathrm{AC}} a^\oplus s + s$. Then we can apply Rule 3 to $a \cdot r =_{\mathrm{AC}} a \cdot (a^\oplus s + s)$, so $a \cdot r$ is not a normal form. Contradiction.

If $s =_{\mathrm{AC}} \delta$, then $r \leftrightarrows a^\oplus \delta$, and $g(r) + g(a^\oplus \delta) < n$, so induction yields $r =_{\mathrm{AC}} a^\oplus \delta$. Then we can apply Rule 5 to $a \cdot r =_{\mathrm{AC}} a \cdot (a^\oplus \delta)$. Again, contradiction.

(3) Finally, assume that summands $a^\oplus r$ and $a^\oplus s$ are bisimilar, so $a^\oplus r + r \leftrightarrows a^\oplus s + s$. We prove $r =_{\mathrm{AC}} s$.

If r and s do not contain summands that are bisimilar with $a^\oplus s$ and $a^\oplus r$ respectively, then $a^\oplus r + r \leftrightarrows a^\oplus s + s$ implies $r \leftrightarrows s$. Since $g(r) + g(s) < n$, induction yields $r =_{\mathrm{AC}} s$, and we are done.

So suppose that either r contains a summand bisimilar to $a^\oplus s$, or s contains a summand bisimilar to $a^\oplus r$. We deduce a contradiction.

By symmetry, it is sufficient to deduce a contradiction for the first case only, where r contains a summand bisimilar to $a^\oplus s$. Induction yields that this summand of r is of the form $a^\oplus s$. According to Rule 1, r can contain only one subterm of the form $a^\oplus s$. Hence, either $r =_{\mathrm{AC}} a^\oplus s$, or $r =_{\mathrm{AC}} a^\oplus s + r'$ where the summands of r' are not bisimilar to $a^\oplus s$. Then $a^\oplus r + r \leftrightarrows a^\oplus s + s$ implies that the summands of r' are bisimilar to summands of s.

The term s does not contain any summands bisimilar to $a^\oplus s$ or $a^\oplus r$. For else, induction would yield that this summand is of the form $a^\oplus s$ or $a^\oplus r$ respectively, which would imply that s contains more symbols than s or r respectively. However, clearly s cannot contain more symbols than itself, and since r has a summand $a^\oplus s$, it follows that r contains more symbols than s.

Recall that r is either of the form $a^\oplus s + r'$ or $a^\oplus s$, and if r' occurs, then all its summands are bisimilar to summands of s. Conversely, since $a^\oplus r + r \leftrightarrows a^\oplus s + s$, and since the summands of s are not bisimilar to $a^\oplus s$ or $a^\oplus r$, it follows that they must all be bisimilar to summands of r', or to δ. Hence, either $s \leftrightarrows r'$ if r' occurs, or $s \leftrightarrows \delta$ otherwise. We distinguish the two possibilities.

- $r =_{\mathrm{AC}} a^\oplus s + r'$ and $s \leftrightarrows r'$. Then induction implies $s =_{\mathrm{AC}} r'$, so we can apply Rule 4 to $a^\oplus r =_{\mathrm{AC}} a^\oplus(a^\oplus s + s)$. Contradiction.
- $r =_{\mathrm{AC}} a^\oplus s$ and $s \leftrightarrows \delta$. Then induction implies $s =_{\mathrm{AC}} \delta$, so we can apply Rule 6 to $a^\oplus r =_{\mathrm{AC}} a^\oplus(a^\oplus \delta)$.

Again, contradiction.

Hence, we may conclude that p and q contain exactly the same summands. Rule 1 ensures that both p and q contain each summand only once, so $p =_{AC} q$. □

Corollary 3. *The axiomatization* A1,2,3,6 + MI1,2 *for* MPA_δ^* *is complete with respect to bisimulation equivalence.*

Proof. If two terms in $MPA_\delta^\oplus$ are bisimilar, then according to Theorem 2 their normal forms are of the same form. Since all the rewrite rules can be deduced from A1,2,3,6 + PMI1,2, it follows that this is a complete axiom system for $MPA_\delta^\oplus$. Then clearly A1,2,3,6 + MI1,2 is a complete axiomatization for MPA_δ^*. □

Acknowledgements

Jan Bergstra initiated this research, and Jos van Wamel provided helpful comments.

References

[1] J.A. Bergstra, I. Bethke and A. Ponse, Process algebra with iteration and nesting, *Comput. J.* **37** (4) (1994) 243–258.

[2] J.A. Bergstra and J.W. Klop, Process algebra for synchronous communication, *Inform. and Comput.* **60** (1/3) (1984) 109–137.

[3] N. Dershowitz and J.-P. Jouannaud, Rewrite systems, in: J. van Leeuwen, ed., *Handbook of Theoretical Computer Science, Volume B, Formal Methods and Semantics* (Elsevier, Amsterdam, 1990) 243–320.

[4] W.J. Fokkink, The tyft/tyxt format reduces to tree rules, in: M. Hagiya and J.C. Mitchell, eds., *Proc. 2nd Symp. on Theoretical Aspects of Computer Software (TACS'94)*, Sendai, Japan, Lecture Notes in Computer Science **789** (Springer, Berlin, 1994) 440–453.

[5] W.J. Fokkink and H. Zantema, Basic process algebra with iteration: completeness of its equational axioms, *Comput. J.* **37** (4) (1994) 259–267.

[6] J.F. Groote and F.W. Vaandrager, Structured operational semantics and bisimulation as a congruence, *Inform. and Comput.* **100** (2) (1992) 202–260.

[7] S.C. Kleene, Representation of events in nerve nets and finite automata, in: *Automata Studies* (Princeton University Press, Princeton, NJ, 1956) 3–41.

[8] J.W. Klop, Term rewriting systems, in: S. Abramsky, D.M. Gabbay and T.S.E. Maibaum, eds., *Handbook of Logic in Computer Science, Volume I, Background: Computational Structures* (Oxford University Press, Oxford, 1992) 1–116.

[9] D.E. Knuth and P.B. Bendix, Simple word problems in universal algebras, in: J. Leech, ed., *Computational Problems in Abstract Algebra* (Pergamon, Oxford, 1970) 263–297; reprinted in: *Automation of Reasoning* **2** (Springer, Berlin, 1983) 342–376.

[10] R. Milner, *A Calculus of Communicating Systems*, Lecture Notes in Computer Science **92** (Springer, Berlin, 1980).

[11] R. Milner, A complete inference system for a class of regular behaviours, *J. Comput. System Sci.* **28** (1984) 439–466.

[12] D.M.R. Park, Concurrency and automata on infinite sequences, in: P. Deussen, ed., *Proc. 5th GI Conf.*, Lecture Notes in Computer Science **104** (Springer, Berlin, 1981) 167–183.

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

[14] P. Sewell, Bisimulation is not finitely (first order) equationally axiomatisable, in: *Proc. LICS'94*, Paris (IEEE Computer Society Press, Silver Spring, MD, 1994) 62–70.