On the Completeness of the Equations for the Kleene Star in Bisimulation

Wan Fokkink
Utrecht University, Department of Philosophy
Heidelberglaan 8, 3584 CS Utrecht, The Netherlands
fokkink@phil.ruu.nl

Abstract

A classical result from Redko [14] says that there does not exist a complete finite equational axiomatization for the Kleene star modulo trace equivalence. Fokkink and Zantema [9] showed that there does exist a complete finite equational axiomatization for the Kleene star up to strong bisimulation equivalence. Their proof is based on a sophisticated term rewriting analysis.

In this paper, we present a much simpler and shorter completeness proof. Furthermore, we show that the three equations for the Kleene star are all essential for this completeness result.

1 Introduction

Kleene [10] defined a binary operator x^*y in the context of finite automata, which denotes the iterate of x and y. Intuitively, the expression x^*y can choose to execute either x, after which it evolves into x^*y again, or y, after which it terminates. An advantage of the Kleene star is that on the one hand it can express recursion, but that on the other hand one can capture this operator in equational laws. Hence, one does not need meta-principles such as the Recursive Specification Principle from Bergstra and Klop [5]. Kleene formulated several equations for his operator, notably $x^*y = x(x^*y) + y$.

Redko [14] (see also Conway [7]) proved that there does not exist a complete finite equational axiomatization for the Kleene star in language theory. Redko's proof can be transposed to the setting of Basic Process Algebra with the binary Kleene star, denoted by BPA*, modulo trace equivalence. Salomaa [15] proposed a complete finite axiomatization for the Kleene star in language theory which includes one conditional axiom.

Bergstra, Bethke and Ponse [4] suggested a finite equational axiomatization for BPA* in the setting of bisimulation equivalence. Fokkink and Zantema [9] proved that this axiomatization is complete, by means of a sophisticated term rewriting analysis.

The completeness proof in [9] is deplorably long and complicated. Therefore, the completeness result itself was presented in the recent handbook chapter of Baeten and Verhoef [3], but its proof was omitted, because it was considered beyond the scope of that paper. Here, we will propose a much simpler completeness proof, which is based on induction on the structure of process terms. This proof can be presented in a handbook, or at an advanced process algebra course.

Furthermore, we will prove that the completeness result is lost if either one of the three equations that are devoted to the Kleene star is removed from the axiomatization. The proof strategy is to find a model for the axioms minus one of the equations for the Kleene star.

Sewell [16] proved that if the deadlock δ is added to BPA*, then a complete finite equational axiomatization does not exist. Milner [11] formulated an axiomatization for this process algebra, including the conditional axiom from Salomaa [15]. He asked whether his axiomatization is complete. This question is still open!

2 BPA with Binary Kleene Star

This section introduces the basic notions.

2.1 The syntax

We assume a non-empty alphabet A of atomic actions, together with three binary operators: alternative composition $+$, sequential composition $\cdot$, and the Kleene star *. Table 1 presents an operational semantics for BPA* in Plotkin style [13]. The special symbol $\surd$ represents (successful) termination.

$$
\begin{array}{cc}
a \xrightarrow{a} \surd & \\[2ex]
\dfrac{x \xrightarrow{a} \surd}{x+y \xrightarrow{a} \surd \xleftarrow{a} y+x} & \dfrac{x \xrightarrow{a} x'}{x+y \xrightarrow{a} x' \xleftarrow{a} y+x} \\[3ex]
\dfrac{x \xrightarrow{a} \surd}{x \cdot y \xrightarrow{a} y} & \dfrac{x \xrightarrow{a} x'}{x \cdot y \xrightarrow{a} x' \cdot y} \\[3ex]
\dfrac{x \xrightarrow{a} \surd}{x^*y \xrightarrow{a} x^*y} & \dfrac{x \xrightarrow{a} x'}{x^*y \xrightarrow{a} x'(x^*y)} \\[3ex]
\dfrac{y \xrightarrow{a} \surd}{x^*y \xrightarrow{a} \surd} & \dfrac{y \xrightarrow{a} y'}{x^*y \xrightarrow{a} y'}
\end{array}
$$

Table 1: Action rules for BPA*

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

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

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

2.2 BPA* modulo trace equivalence

Redko [14] proved that there does not exist a complete finite equational axiomatization for the Kleene star in language theory. We observe that Redko's proof can be transposed to BPA* modulo trace equivalence. This observation is not immediate, because Redko studies the Kleene star in the presence of the special constants 0 and 1 from language theory. However, Redko's proof does not use these constants; the basic idea is that x^* is trace equivalent with $(x^n)^*(x + x^2 + \ldots + x^{n-1})$ for each $n \geq 2$, and this infinite number of equivalences cannot be expressed in finitely many equations. This reasoning is also valid in BPA*.

2.3 Bisimulation

In this paper, process terms are considered modulo (strong) bisimulation equivalence from Park [12]. Intuitively, two process terms are bisimilar if they have the same branching structure.

Definition 2.1 *Two processes p and q are* bisimilar*, denoted by $p \underline{\leftrightarrow} q$, if there exists a symmetric binary relation B on processes which relates p and q, such that:*

- *if rBs and $r \xrightarrow{a} r'$, then there is a transition $s \xrightarrow{a} s'$ such that $r'Bs'$,*

- *if rBs and $r \xrightarrow{a} \surd$, then $s \xrightarrow{a} \surd$.*

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

2.4 An axiomatization for BPA* modulo bisimulation

Table 2 contains an axiom system for BPA*. It consists of the standard axioms A1-5 for BPA, together with three axioms BKS1-3 for iteration. The most advanced axiom BKS3 originates from Troeger [17]. In the sequel, $p = q$ will mean that this equality can be derived from the axioms.

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

In the sequel, terms are considered modulo associativity and commutativity of the $+$, and we write $p =_{\mathrm{AC}} q$ if p and q can be equated by axioms A1,2. As usual, $\sum_{i=1}^{n} p_i$ represents $p_1 + \ldots + p_n$. In the sequel, we will take care to avoid empty sums (where $\sum_{i\in\emptyset} p_i + p$ is not considered empty).

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

Table 2: Axioms for BPA*

For each process term p, its collection of possible transitions is non-empty and finite, say $\{p \xrightarrow{a_i} p_i \mid i=1,...,m\} \cup \{p \xrightarrow{b_j} \surd \mid j=1,...,n\}$. We call

$$\sum_{i=1}^{m} a_i p_i + \sum_{j=1}^{n} b_j$$

the *expansion* of p. The terms $a_i p_i$ and b_j are called the *summands* of p.

Lemma 2.2 *Each process term is provably equal to its expansion.*

Proof. Straightforward, by induction on term structure, using axioms A4,5 and BKS1.

3 An Efficient Completeness Proof

3.1 A comparison of proof strategies

First, we discuss the strategy of the completeness proof from Fokkink and Zantema [9]. That proof is based on a standard rewriting technique, which means a quest for unique ground normal forms. They note that this strive cannot be fulfilled for the Kleene star. Therefore, they replace this operator by $p^{\oplus}q$, which represents $p(p^*q)$, and the axioms BKS1-3 are adopted for this new operator. They turn the axioms into conditional rewrite rules, which are applied modulo AC of the $+$. Four rewrite rules are added to make the rewrite system weakly confluent, that is, if there are one-step reductions from a term p to terms p' and p'', then both p' and p'' can be reduced to a term q.

Their next aim is to prove that the resulting conditional rewrite system is terminating, which means that there are no infinite reductions. In this particular case, deducing termination is a complicated matter, due to the occurrence of a rewrite rule where the left-hand side can be obtained from the right-hand side by the elimination of function symbols. Termination is obtained by means of the advanced technique of semantic labelling from Zantema [18]. Hence, each process term is provably equal to a ground normal form, which cannot be reduced by the conditional rewrite system.

Finally, a painstaking case analysis learns that if two ground normal forms are bisimilar, then they are the same modulo AC of the $+$. This observation yields the desired completeness result.

In this paper, we present a completeness proof based on induction on term structure. This strategy turns out to be much more convenient than the term rewriting analysis sketched above. We determine a subset $\mathcal{B}$ of *basic* process terms, such that each process term is provably equal to a basic term. Next, we determine a sophisticated ordering on $\mathcal{B}$. Finally, we prove by induction on this ordering that bisimilar basic terms are provably equal.

3.2 A lemma for normed processes

Processes in BPA* are *normed*, which means that they are able to terminate in finitely many transitions. The *norm* of a process yields the length of the shortest termination trace of this process; this notion stems from [1]. Norm can be defined inductively as follows.

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

Note that bisimilar processes have the same norm. The following lemma is typical for normed processes.

Lemma 3.1 *Let $pq \underline{\leftrightarrow} rs$. By symmetry we may assume $|q| \leq |s|$. We can distinguish two cases:*

- *either $p \underline{\leftrightarrow} r$ and $q \underline{\leftrightarrow} s$,*
- *or there is a substate p' of p such that $p \underline{\leftrightarrow} rp'$ and $p'q \underline{\leftrightarrow} s$.*

Proof. We prove this lemma from the following facts A and B. Fact B originates from Caucal [6].

A. If $pq \underline{\leftrightarrow} rs$ and $|q| \leq s$, then either $q \underline{\leftrightarrow} s$, or there is a substate p' of p such that $p'q \underline{\leftrightarrow} s$.

Proof. We apply induction on $|p|$. First, let $|p| = 1$. Then $p \xrightarrow{a} \surd$ for some a, so $pq \xrightarrow{a} q$. Since $pq \underline{\leftrightarrow} rs$, we have two options:

- $r \xrightarrow{a} \surd$ and $q \underline{\leftrightarrow} s$. Then we are done.
- $r \xrightarrow{a} r'$ and $q \underline{\leftrightarrow} r's$. This leads to a contradiction: $|q| \leq |s| < |r's| = |q|$.

Next, suppose that we have proved the case for $|p| \leq n$, and let $|p| = n+1$. Then there is a p' with $|p'| = n$ and $p \xrightarrow{a} p'$, which implies $pq \xrightarrow{a} p'q$. Since $pq \underline{\leftrightarrow} rs$, we have two options:

- $r \xrightarrow{a} \surd$ and $p'q \underline{\leftrightarrow} s$. Then we are done.
- $r \xrightarrow{a} r'$ and $p'q \underline{\leftrightarrow} r's$. Since $|p'| = n$, induction yields either $q \underline{\leftrightarrow} s$ or $p''q \underline{\leftrightarrow} s$ for a substate p'' of p'. Again, we are done.

B. If $pq \underline{\leftrightarrow} rq$, then $p \underline{\leftrightarrow} r$.

Proof. Define a binary relation B between closed terms by tBu if $tq \underline{\leftrightarrow} uq$. We show that B constitutes a bisimulation relation between p and r:

- Since $\underline{\leftrightarrow}$ is symmetric, so is B.
- $pq \underline{\leftrightarrow} rq$, so pBr.
- Suppose that tBu and $t \xrightarrow{a} t'$. Then $tq \xrightarrow{a} t'q$, so $tq \underline{\leftrightarrow} uq$ implies that this transition can be mimicked by a transition from uq. This cannot be a transition $uq \xrightarrow{a} q$, because $|t'q| > |q|$, so apparently there is a transition $u \xrightarrow{a} u'$ with $t'q \underline{\leftrightarrow} u'q$. Hence, $t'Bu'$.
- Similarly, we find that if tBu and $t \xrightarrow{a} \surd$, then $u \xrightarrow{a} \surd$.

Finally, we show that facts A and B together prove the lemma. Let $pq \underline{\leftrightarrow} rs$ with $|q| \leq |s|$. According to fact A we can distinguish two cases:

- $q \underline{\leftrightarrow} s$. Then $pq \underline{\leftrightarrow} rs \underline{\leftrightarrow} rq$, so fact B yields $p \underline{\leftrightarrow} r$.
- $p'q \underline{\leftrightarrow} s$ for some substate p' of p. Then $pq \underline{\leftrightarrow} rs \underline{\leftrightarrow} rp'q$, so fact B yields $p \underline{\leftrightarrow} rp'$. □

3.3 Basic terms

We construct a set $\mathcal{B}$ of *basic* process terms, such that each process term is provably equal to a basic term. We will prove the completeness theorem by showing that bisimilar basic terms are provably equal.

$(x+y)z$	$\longrightarrow$	$xz+yz$
$(xy)z$	$\longrightarrow$	$x(yz)$
$(x^*y)z$	$\longrightarrow$	$x^*(yz)$

Table 3: The rewrite system R

Table 3 presents a rewrite system R, which consists of directions of the axioms A4,5 and BKS2, pointing from left to right. The rules in R are to be interpreted modulo AC of the $+$. R is terminating, which means that there are no infinite reductions. This follows from the following weight function w in the natural numbers.

$$\begin{aligned} w(a) &= 2 \\ w(p+q) &= w(p)+w(q) \\ w(pq) &= w(p)^2 w(q) \\ w(p^*q) &= w(p)+w(q). \end{aligned}$$

It is easy to see that if R reduces p to p', then $w(p) > w(p')$. Since the ordering on the natural numbers is well-founded, we can conclude that R is terminating.

Let $\mathcal{N}$ denote the collection of *ground normal forms* of R, i.e., the collection of process terms that cannot be reduced by rules in R. The elements in $\mathcal{N}$ are defined by the following BNF grammar.

$$p \ ::= \ a \mid p+p \mid ap \mid p^*q.$$

Since R is terminating, and since its rules are directions of axioms, it follows that each process term is provably equal to a process term in $\mathcal{N}$.

Definition 3.2 *p' is a* substate *of p if p can evolve into p' by one or more transitions.*

$\mathcal{N}$ is not yet our desired set of basic terms, due to the fact that there exist process terms in $\mathcal{N}$ which have a substate outside $\mathcal{N}$. We give an example.

Example 3.3 *Let $A = \{a, b, c\}$. Clearly, $(a^*b)^*c \in \mathcal{N}$, and*

$$(a^*b)^*c \xrightarrow{a} (a^*b)((a^*b)^*c).$$

However, the substate $(a^*b)((a^*b)^*c)$ is not in $\mathcal{N}$, because the third rule in R reduces this term to $a^*(b((a^*b)^*c))$.

In order to overcome this complication, we will extend $\mathcal{N}$ with the following collection of process terms:

$$\mathcal{H} = \{p^*q,\ p'(p^*q) \mid p^*q \in \mathcal{N} \wedge p' \text{ substate of } p\}.$$

We define an equivalence relation $\cong$ on $\mathcal{H}$ by putting $p'(p^*q) \cong p^*q$ for substates p' of p, and taking the reflexive, symmetric, transitive closure of $\cong$.

Finally, the set $\mathcal{B}$ of *basic* terms is the union of $\mathcal{N}$ and $\mathcal{H}$.

Lemma 3.4 *If $p \in \mathcal{B}$ and $p \xrightarrow{a} q$, then $q \in \mathcal{B}$.*

Proof. Straightforward, by induction on the structure of p.

3.4 An ordering on basic terms

Norm does not constitute a nice ordering on process terms, because it does not respect term size, for example, $|aa+a| < |aa|$. L-value, from Fokkink and Zantema [9], induces an ordering which does not have this drawback. It is defined as follows:

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

Since norm is preserved under bisimulation, it follows that the same holds for L-value.

Lemma 3.5 *If $p \underline{\leftrightarrow} q$, then $L(p) = L(q)$.*

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

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

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

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

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

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

$$\{|p'(p^*q)| \mid p' \text{ substate of } p\} \ \cup \ \{|p^*q|\} \ \cup \ \{|q'| \mid q' \text{ substate of } q\},$$

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

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

Note that $L(p) < L(pq)$ and $L(p) < L(p^*q)$.

We define an ordering on $\mathcal{B}$ as follows:

- $p < q$ if $L(p) < L(q)$,

- $p < q$ if p is a substate of q but q is not a substate of p,

and we take the transitive closure of $<$.

Note that if $p \cong q$, then p and q have the same substates, and so $L(p) = L(q)$. These observations imply that the ordering $<$ on $\mathcal{B}$ respects the equivalence $\cong$ on $\mathcal{H}$, that is, if $p \cong q < r \cong s$, then $p < s$.

Lemma 3.6 *$<$ is a well-founded ordering on $\mathcal{B}$.*

Proof. If p is a substate of q, then all substates of p are substates of q, so $L(p) \leq L(q)$. Hence, if $p < q$ then $L(p) \leq L(q)$.

Suppose that $<$ is not well-founded, so there exists an infinite chain $p_0 > p_1 > p_2 > \cdots$. Then $L(p_n) \geq L(p_{n+1})$ for all n, so there is an N such that $L(p_N) = L(p_n)$ for all $n > N$. Since $p_N > p_n$ for $n > N$, it follows that p_n is a substate of p_N for $n > N$. Each process term has only finitely many substates, so there are $m, n > N$ with $m < n$ and $p_m =_{\mathrm{AC}} p_n$. Then $p_m \not> p_n$, so we have found a contradiction. Hence, $>$ is well-founded. □

In the next two lemmas, we need a weight function g in the natural numbers, which is defined inductively as follows:

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

It is not difficult to see, by induction on term structure, that if $p \xrightarrow{a} q$, then $g(p) \geq g(q)$.

Lemma 3.7 *Let $p^*q \in \mathcal{B}$. If q' is a substate of q, then $q' < p^*q$.*

Proof. Since q' is a substate of q, it follows that $g(q') \leq g(q)$. Hence, $g(q') < g(p^*q)$, so p^*q cannot be a substate of q'. On the other hand, q' is a substate of p^*q, so then $q' < p^*q$. □

Lemma 3.8 *If $p \in \mathcal{B}$ and $p \xrightarrow{a} q$, then either $q < p$, or $p, q \in \mathcal{H}$ and $p \cong q$.*

Proof. We will use the following two facts A and B.

A. If $p \in \mathcal{N}$ and $q \notin \mathcal{H}$ and $p \xrightarrow{a} q$, then q is a proper subterm of p, that is, $p =_{\mathrm{AC}} C[q]$ for a non-empty context $C[]$.

Proof. We apply induction on the structure of p. Since $p \in \mathcal{N}$, we have

$$p =_{\mathrm{AC}} \sum_i a_i r_i + \sum_j s_j^* t_j + \sum_k b_k.$$

Since $q \notin \mathcal{H}$, and $p \xrightarrow{a} q$, we find that q is of one of the following forms:

- $q =_{\mathrm{AC}} r_i$ for some i. In this case we are done, because the r_i are proper subterms of p.
- $q =_{\mathrm{AC}} s_j'(s_j^* t_j)$ or $q =_{\mathrm{AC}} s_j^* t_j$ for some j. These cases contradict the assumption that $q \notin \mathcal{H}$.
- $t_j \xrightarrow{a} q$ for some j. In this last case, induction yields that q is a proper subterm of t_j, and thus of p.

B. If $p \in \mathcal{H}$ and $p \xrightarrow{a} q$, then either $g(p) > g(q)$, or $q \in \mathcal{H}$ and $p \cong q$.

Proof. Since $p \in \mathcal{H}$, either $p =_{\mathrm{AC}} r'(r^* s)$ or $p =_{\mathrm{AC}} r^* s$ for certain r and s. Hence, either $q =_{\mathrm{AC}} r^* s$, or $q =_{\mathrm{AC}} r''(r^* s)$ for a substate r'' of r, or $q =_{\mathrm{AC}} s'$ for a substate s' of s. In the first two cases $q \in \mathcal{H}$ and $p \cong q$, and in the last case $g(q) = g(s') \leq g(s) < g(r^* s) = g(p)$.

Now, we are ready to prove the lemma. Let $p_0 \xrightarrow{a} p_1$ with $p_1 \not< p_0$; we prove that $p_0, p_1 \in \mathcal{H}$ and $p_0 \cong p_1$.

Since p_1 is a substate of p_0 and $p_1 \not< p_0$, apparently p_0 is a substate of p_1. So there exists a sequence

$$p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} p_n =_{\mathrm{AC}} p_0, \qquad n \geq 1.$$

Suppose that $p_k \notin \mathcal{H}$ for all k. Then according to fact A, p_{k+1} is a proper subterm of p_k for $k = 0, ..., n-1$, so $p_n =_{\mathrm{AC}} p_0$ is a proper subterm of p_0; contradiction. Hence, $p_l \in \mathcal{H}$ for some l.

Since each p_k is a substate of each $p_{k'}$, we have $g(p_k) \leq g(p_{k'})$ for k and k', so $g(p_k)$ must be the same for all k. Then it follows from fact B, together with $p_l \in \mathcal{H}$, that $p_k \in \mathcal{H}$ for all k and $p_0 \cong p_1 \cong \cdots \cong p_n$. $\square$

In the sequel, we write $p \leq q$ for $p < q \vee p \cong q$. The ordering $<$ on $\mathcal{B}$ is extended to $\mathcal{B} \times \mathcal{B}$ as expected: $(p, q) < (r, s)$ if either $p \leq r$ and $q < s$, or $p < r$ and $q \leq s$.

3.5 The main theorem

Now we are ready to prove the desired completeness result.

Theorem 3.9 *If $p \underline{\leftrightarrow} q$, then* A1-5+BKS1-3 $\vdash p = q$.

Proof. Each process term is provably equal to a basic term, so it is sufficient to show that bisimilar basic terms are provably equal. Assume $p, q \in \mathcal{B}$ with $p \underline{\leftrightarrow} q$; we show that $p = q$, by induction on the ordering $<$ on $\mathcal{B} \times \mathcal{B}$. So suppose that we have already dealt with pairs of bisimilar basic terms that are smaller than (p, q).

First, assume that p or q is not in $\mathcal{H}$, say $p \notin \mathcal{H}$. Since $p \underline{\leftrightarrow} q$, we can write the expansions of p and q as

$$p = \sum_{i=1}^{m} a_i p_i + \sum_{j=1}^{n} b_j, \qquad q = \sum_{i=1}^{m} a_i q_i + \sum_{j=1}^{n} b_j,$$

where $p_i \underline{\leftrightarrow} q_i$ for $i = 1, ..., m$. Since $p \notin \mathcal{H}$, Lemma 3.8 says that $p_i < p$ for $i = 1, ..., m$. Furthermore, Lemma 3.8 says that $q_i \leq q$ for $i = 1, ..., m$. Then $(p_i, q_i) < (p, q)$, so induction yields $p_i = q_i$ for $i = 1, ..., m$. Hence, $p = q$.

Next, assume $p, q \in \mathcal{H}$. We distinguish three cases.

1. Let $p =_{\mathrm{AC}} r^*s$ and $q =_{\mathrm{AC}} t^*u$. We prove that $r^*s \underline{\leftrightarrow} t^*u$ implies $r^*s = t^*u$.

   We spell out the expansions of r and t:

   $$r = \sum_{i \in I} r_i, \qquad t = \sum_{j \in J} t_j,$$

   where the r_i and the t_j are of the form either av or a.

   Clearly, the summands of t^*u are the summands of $t(t^*u)$ together with the summands of u. Hence, since $r^*s \underline{\leftrightarrow} t^*u$, each term $r_i(r^*s)$ for $i \in I$ is bisimilar either to $t_j(t^*u)$ for a $j \in J$ or to a summand of u. We distinguish these two cases.

   (a) $r_i(r^*s) \underline{\leftrightarrow} t_j(t^*u)$ for a $j \in J$. Since $r^*s \underline{\leftrightarrow} t^*u$, we find that $r_i(r^*s) \underline{\leftrightarrow} t_j(r^*s)$, so Lemma 3.1 yields $r_i \underline{\leftrightarrow} t_j$.

   (b) $r_i(r^*s) \underline{\leftrightarrow} au'$ for a $u \xrightarrow{a} u'$.

   Thus, we can divide I into the following, not necessarily disjoint, subsets.

   $$\begin{array}{lcl} I_0 & = & \{i \in I \mid \exists j \in J\ (r_i \underline{\leftrightarrow} t_j)\} \\ I_1 & = & \{i \in I \mid \exists u \xrightarrow{a} u'\ (r_i(r^*s) \underline{\leftrightarrow} au')\} \end{array}$$

   Similarly, we can divide J:

   $$\begin{array}{lcl} J_0 & = & \{j \in J \mid \exists i \in I\ (t_j \underline{\leftrightarrow} r_i)\} \\ J_1 & = & \{j \in J \mid \exists s \xrightarrow{a} s'\ (t_j(t^*u) \underline{\leftrightarrow} as')\} \end{array}$$

   We prove an equation.

   Equation 1 $\sum_{i \in I_1} r_i(r^*s) + s = \sum_{j \in J_1} t_j(t^*u) + u$.

   Proof. We show that each separate term in the sum at the left-hand side of the equality sign is provably equal to a term in the sum at the right-hand side. The converse observation follows by symmetry, so then we can conclude that the equality is valid.

By definition of I_1, for each $r_i(r^*s)$ with $i \in I_1$ there is a summand au' of u such that $r_i(r^*s) \underline{\leftrightarrow} au'$. According to Lemma 3.7 $u' < t^*u$, so induction yields $r_i(r^*s) = au'$.

Consider a summand as' of s. Since $r^*s \underline{\leftrightarrow} t^*u$, it follows that as' is bisimilar either with a term $t_j(t^*u)$ with $j \in J$, or with a summand au' of u. In the first case, induction yields $as' = t_j(t^*u)$, and from the definition of J it follows that $j \in J_1$. In the second case, induction yields $as' = au'$ respectively.

Finally, summands a of s correspond with summands a of u.

We continue with the proof of $r^*s = t^*u$. First, suppose that $I_0 = \emptyset$. Then clearly also $J_0 = \emptyset$, and so $I = I_1$ and $J = J_1$. Then $r^*s \overset{\text{BKS1}}{=} \sum_{i \in I_1} r_i(r^*s) + s \overset{\text{Eq. 1}}{=} \sum_{j \in J_1} t_j(t^*u) + u \overset{\text{BKS1}}{=} t^*u$, which is what we want to prove.

So we can assume that I_0 and J_0 are not empty. Put

$$\bar{r} = \sum_{i \in I_0} r_i, \qquad \bar{t} = \sum_{j \in J_0} t_j.$$

Equation 2 $\bar{r} = \bar{t}$.

Proof. By definition of I_0 and J_0, each r_i for $i \in I_0$ is bisimilar to a t_j with $j \in J_0$. Since $L(r_i) \leq L(r) < L(r^*s)$, induction yields $r_i = t_j$. Conversely, each t_j for $j \in J_0$ is provably equal to a r_i with $i \in I_0$. Hence, $\bar{r} = \bar{t}$.

Since $I_0 \cup I_1 = I$, we have $r \overset{\text{A3}}{=} \bar{r} + \sum_{i \in I_1} r_i$. Even so, $t \overset{\text{A3}}{=} \bar{t} + \sum_{j \in J_1} t_j$.

Finally, we can derive $r^*s = t^*u$:

$$\begin{array}{rcl} r^*s & \overset{\text{A3}}{=} & (\bar{r} + \sum_{i \in I_1} r_i)^*s \\ & \overset{\text{BKS3}}{=} & \bar{r}^*(\sum_{i \in I_1} r_i(r^*s) + s) \\ & \overset{\text{Eq. 1,2}}{=} & \bar{t}^*(\sum_{j \in J_1} t_j(t^*u) + u) \\ & \overset{\text{BKS3}}{=} & (\bar{t} + \sum_{j \in J_1} t_j)^*u \\ & \overset{\text{A3}}{=} & t^*u. \end{array}$$

Recall that we are considering a bisimilar pair $p, q \in \mathcal{H}$. We continue with the last two cases.

2. Let $p =_{\text{AC}} r'(r^*s)$ and $q =_{\text{AC}} t^*u$. We prove that $r'(r^*s) \underline{\leftrightarrow} t^*u$ implies $r'(r^*s) = t^*u$.

   $|u| = |t^*u| = |r'(r^*s)| \geq 2$, so u does not have atomic summands, which means that its expansion is of the form $\sum_i a_i u_i$. Since $r'(r^*s) \underline{\leftrightarrow} t^*u$, each u_i is bisimilar to r^*s or to a term $r''(r^*s)$. According to Lemma 3.7 $u_i < t^*u$, and $r^*s \cong r'(r^*s)$ or $r''(r^*s) \cong r'(r^*s)$, so induction yields $u_i = r^*s$ or $u_i = r''(r^*s)$ respectively. This holds for all i, so $u = \sum_i a_i u_i = v(r^*s)$ for some term v. Then $r'(r^*s) \underline{\leftrightarrow} t^*u \underline{\leftrightarrow} (t^*v)(r^*s)$, so Lemma 3.1 implies $r' \underline{\leftrightarrow} t^*v$. Since $L(r') < L(r'(r^*s))$, induction yields $r' = t^*v$. Hence, $r'(r^*s) = (t^*v)(r^*s) \overset{\text{BKS2}}{=} t^*(v(r^*s)) = t^*u$.

3. Let $p =_{\mathrm{AC}} r'(r^*s)$ and $q =_{\mathrm{AC}} t'(t^*u)$. We prove that $r'(r^*s) \underline{\leftrightarrow} t'(t^*u)$ implies $r'(r^*s) = t'(t^*u)$.

   By symmetry we may assume $|r^*s| \leq |t^*u|$. Lemma 3.1 distinguishes two possible cases.

   Either $r' \underline{\leftrightarrow} t'$ and $r^*s \underline{\leftrightarrow} t^*u$. Since $L(r') < L(r'(r^*s))$, induction yields $r' = t'$. Furthermore, we can apply the first construction to $r^*s \underline{\leftrightarrow} t^*u$ in order to obtain $r^*s = t^*u$.

   Or $r' \underline{\leftrightarrow} t'r''$ and $r''(r^*s) \underline{\leftrightarrow} t^*u$ for a substate r'' of r'. Since $L(r') < L(r'(r^*s))$, induction yields $r' = t'r''$. Furthermore, we can apply the second construction to $r''(r^*s) \underline{\leftrightarrow} t^*u$ in order to obtain $r''(r^*s) = t^*u$. Hence, $r'(r^*s) = (t'r'')(r^*s) \overset{\mathrm{BKS2}}{=} t'(r''(r^*s)) = t'(t^*u)$. □

3.6 An example

We give an example as to how the construction in the completeness proof acts on a particular pair of bisimilar process terms.

Example 3.10 Let $A = \{a_1, a_2, b\}$, and consider the following two process terms:

$$p =_{\mathrm{AC}} a_1^*(a_2((a_1+a_2)^*b)+b), \qquad q =_{\mathrm{AC}} a_2^*(a_1((a_1+a_2)^*b)+b).$$

Note that $p \underline{\leftrightarrow} (a_1+a_2)^*b \underline{\leftrightarrow} q$. We show how the construction in the proof of Theorem 3.9 applied to the pair p, q produces a derivation of $p = q$.

Clearly $p, q \in \mathcal{H}$, and we are dealing with the first of the three possible cases for $p, q \in \mathcal{H}$ that were distinguished in the completeness proof. Following the notations that were introduced there, we have $I = \{1\}$ and $J = \{2\}$. Since $a_1 \underline{\not\leftrightarrow} a_2$, it follows that I_0 and J_0 are empty. Then $I_1 = I$ and $J_1 = J$, so in order to prove $p = q$, it is sufficient to prove the version of Equation 1 that is obtained by applying BKS1 to both p and q:

$$a_1p + a_2((a_1+a_2)^*b) + b = a_2q + a_1((a_1+a_2)^*b) + b.$$

This equality follows immediately from the following two equalities, which are smaller than $p = q$ with respect to the ordering on $\mathcal{B} \times \mathcal{B}$.

$$\begin{array}{lcccl} (a_1+a_2)^*b & = & q & =_{\mathrm{AC}} & a_2^*(a_1((a_1+a_2)^*b)+b), \\ (a_1+a_2)^*b & = & p & =_{\mathrm{AC}} & a_1^*(a_2((a_1+a_2)^*b)+b). \end{array}$$

We only deal with the first equation. Again, this equation is in the first of the three categories for bisimilar terms in $\mathcal{H}$ that were distinguished in the completeness proof. In this case, $I_0 = \{2\}$ and $I_1 = \{1\}$ and $J_0 = \{2\}$ and $J_1 = \emptyset$, so the final derivation takes the following simple form:

$$(a_1+a_2)^*b \overset{\mathrm{BKS3}}{=} a_2^*(a_1((a_1+a_2)^*b)+b).$$

4 The Axioms BKS1-3 are Essential for Completeness

In this section, we show that each of the axioms BKS1-3 for the binary Kleene star is essential for the obtained completeness result.

Theorem 4.1 *Completeness of the axioms A1-5+BKS1-3 for BPA* modulo bisimulation is lost if one of the axioms BKS1-3 is skipped.*

Proof. A standard technique for proving that some equation e cannot be derived from an equational theory $\mathcal{E}$ is to define a model for $\mathcal{E}$ in which e is not valid. We will apply this technique to show that the axioms BKS1-3 cannot be derived from the other axioms.

In order to show that BKS1 cannot be derived from A1-5+BKS2,3, we define the following interpretation function ϕ of open terms in the natural numbers. It captures the intuition that BKS1 is the only equality that enables to expand the Kleene star. Namely, it does not take into account terms that occur at the right-hand side of a multiplication.

$$\begin{array}{rcl}\phi(a) &=& 0\\ \phi(x) &=& 0\\ \phi(P+Q) &=& \max\{\phi(P),\phi(Q)\}\\ \phi(P\cdot Q) &=& \phi(P)\\ \phi(P^*Q) &=& \max\{\phi(P)+1,\phi(Q)+1\}\end{array}$$

It is easy to see that this interpretation is a model for A1-5+BKS2,3. However, $\phi(a(a^*a)+a)=0$, while $\phi(a^*a)=1$. Hence, the equality $a(a^*a)+a=a^*a$ cannot be derived from A1-5+BKS2,3.

In order to show that BKS2 cannot be derived from A1-5+BKS1,3, we define the following interpretation function ψ of open terms in the natural numbers.

$$\begin{array}{rcl}\psi(a) &=& 0\\ \psi(x) &=& 0\\ \psi(P+Q) &=& \max\{\psi(P),\psi(Q)\}\\ \psi(P\cdot Q) &=& \psi(Q)\\ \psi(P^*Q) &=& \max\{\psi(P)+1,\psi(Q)\}\end{array}$$

It is easy to see that this interpretation is a model for A1-5+BKS1,3. However, $\psi((a^*a)a)=\psi(a)=0$, while $\psi(a^*(aa))=\max\{\psi(a)+1,\psi(aa)\}=1$. Hence, the equality $(a^*a)a=a^*(aa)$ cannot be derived from A1-5+BKS1,3.

Finally, in order to show that BKS3 cannot be derived from A1-5+BKS1,2, we define the following interpretation function η of open terms in sets of natural numbers. It captures the intuition that BKS3 is the only equality that can change the interpretation at the left-hand side of a Kleene star. Namely, $\eta(P)$ collects the norms of subterms that occur as arguments at the left-hand side of a Kleene star.

$$\begin{array}{rcl}\eta(a) &=& \emptyset\\ \eta(x) &=& \emptyset\\ \eta(P+Q) &=& \eta(P)\cup\eta(Q)\\ \eta(P\cdot Q) &=& \eta(P)\cup\eta(Q)\\ \eta(P^*Q) &=& \eta(P)\cup\eta(Q)\cup\{|P|\}\end{array}$$

It is easy to see that this interpretation is a model for A1-5+BKS1,2. However, $\eta((aa)^*(a((aa+a)^*a)+a)) = \{|aa|, |aa+a|\} = \{1,2\}$ while $\eta((aa+a)^*a) = \{|aa+a|\} = \{1\}$. Hence, the equality $(aa)^*(a((aa+a)^*a)+a) = (aa+a)^*a$ cannot be derived from A1-5+BKS1,2. □

References

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

[2] J.C.M. Baeten and C. Verhoef. A congruence theorem for structured operational semantics with predicates. In E. Best, editor, *Proceedings 4th Conference on Concurrency Theory (CONCUR'93),* Hildesheim, volume 715 of *Lecture Notes in Computer Science*, pages 477–492. Springer-Verlag, 1993.

[3] J.C.M. Baeten and C. Verhoef. Concrete process algebra. In S. Abramsky, D.M. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science, Volume IV, Syntactical Methods*, pages 149–268. Oxford University Press, 1995.

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

[5] J.A. Bergstra and J.W. Klop. Verification of an alternating bit protocol by means of process algebra. In W. Bibel and K.P. Jantke, editors, *Proceedings Spring School on Mathematical Methods of Specification and Synthesis of Software Systems '85,* Wendisch-Rietz, volume 215 of *Lecture Notes in Computer Science*, pages 9–23. Springer-Verlag, 1986.

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

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

[8] W.J. Fokkink and R.J. van Glabbeek. Ntyft/ntyxt rules reduce to ntree rules. Technical Note CS-95-17, Stanford University, 1995. To appear in *Information and Computation.*

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

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

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

[12] D.M.R. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, *Proceedings 5th GI Conference,* Karlsruhe, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer-Verlag, 1981.

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

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

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

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

[17] 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.

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