A Complete Axiomatization for Prefix Iteration in Branching Bisimulation

Wan FOKKINK

Utrecht University, Department of Philosophy, Heidelberglaan 8, 3584 CS Utrecht,

The Netherlands, fokkink@phil.ruu.nl

Abstract

This paper studies the interaction of prefix iteration with the silent step in the setting of branching bisimulation. We present a finite equational axiomatization for Basic Process Algebra with deadlock, empty process and the silent step, extended with prefix iteration, and prove that this axiomatization is complete with respect to rooted branching bisimulation equivalence.

1 Introduction

Kleene [15] 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. 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 [8]. Kleene formulated several equations for his operator, for example, $x^*y = x(x^*y) + y$.

This paper considers the prefix counterpart μ^*x of iteration in process algebra, where the argument μ ranges over the set of constants. Our setting is Basic Process Algebra (BPA) from Bergstra and Klop [7], with the deadlock δ and the empty process ϵ and the silent step τ, extended with prefix iteration.

Milner [16] was the first to study iteration in strong bisimulation equivalence, in a process algebra equivalent to $\text{BPA}_{\delta\epsilon}$ extended with the Kleene star. Milner proposed an axiomatization for this process algebra, including a conditional axiom for iteration from Salomaa [19], and he raised the question whether his axiomatization is complete. This question is still open. Bergstra, Bethke and Ponse [6] considered BPA with the Kleene star, and they suggested a finite equational axiomatization for this algebra. Fokkink and Zantema [10] proved that this axiomatization is complete with respect to strong bisimulation equivalence.

Sewell [20] showed that there does not exist a complete finite equational axiomatization for BPA_δ with the Kleene star modulo strong bisimulation, due to the fact that $x^*\delta$ and $(x^n)^*\delta$ are strongly bisimilar for $n \geq 2$. In order to obtain an equational axiomatization nevertheless, the range of terms that are allowed to occur at the left-hand side of the binary Kleene star is to be restricted. Fokkink [9] proposed a complete finite equational axiomatization for *prefix* iteration a^*x, where a ranges over the atomic actions, in a process algebra equivalent to basic CCS. Aceto and Groote [2] generalized this result to *string* iteration w^*x, where w ranges over strings of atomic actions of length smaller than some N.

Fokkink and Zantema [11] studied a rewrite system that stems from axioms for prefix iteration in $\mathrm{BPA}_{\delta\epsilon}$. In order to prove termination of this rewrite system, they generalized a termination theorem from Zantema and Geser [22], based on abstract commutation, to the setting of rewriting modulo equations. As a side result, they obtained a mild variant of the result in [9], namely a complete axiomatization for prefix iteration in $\mathrm{BPA}_{\delta\epsilon}$, instead of in basic CCS. In a revision of Baeten and Bergstra [4], this axiomatization has been applied in an extension of BPA_{δ} with discrete time.

Aceto and Ingólfsdóttir [3] studied prefix iteration in the presence of the silent step τ, in Milner's observation congruence. They extended the axiomatization from [9] with two standard equations for the silent step, and with three new equations which describe the interplay between the silent step and prefix iteration. They proved that their equational axiomatization is complete with respect to observation congruence.

This paper results from an attempt to try and shorten the long and technical completeness proof in [3]. Although this attempt was unsuccessful for observation congruence, it did yield a considerably shorter completeness proof for prefix iteration together with the silent step in rooted branching bisimulation equivalence from Van Glabbeek and Weijland [13]. It turns out that two axioms are sufficient in order to describe the relation between prefix iteration and the silent step in this context:

$$\begin{aligned} \tau^*x &= \tau x + x \\ a \cdot a^*(\tau \cdot a^*(x+y) + x) &= a \cdot a^*(x+y) \end{aligned}$$

The first axiom is based on Koomen's Fair Abstraction Rule as formulated in Baeten and Van Glabbeek [5]. The second axiom is, to our knowledge, new. We prove that these two axioms, together with the standard axioms for $\mathrm{BPA}_{\delta\epsilon\tau}$, make a complete finite equational axiomatization for $\mathrm{BPA}_{\delta\epsilon\tau}$ extended with prefix iteration, with respect to rooted branching bisimulation equivalence.

Van Glabbeek [12] showed that the completeness result for prefix iteration in observation congruence from [3] follows from the completeness result for rooted branching bisimulation that is presented in this paper. The combination of these two proofs leads to a considerably shorter completeness proof for prefix iteration in observation congruence than the one presented in [3]. Recently, the four authors have merged their three papers into one paper, [1], which deals at once with weak, branching, delay, and η-bisimulation. Among other things, that paper presents an improved completeness proof for prefix iteration in branching bisimulation, which has the advantage that it is self-contained, while the the proof that is presented in this paper uses the completeness result from [11].

Acknowledgements. This research was initiated by discussions with Luca Aceto. Rob van Glabbeek provided useful comments.

2 Preliminaries

2.1 The syntax and semantics

Assume a non-empty alphabet A of atomic actions, and three special constants δ and ϵ and τ, which represent deadlock and empty process and the silent step respectively.

In the sequel we use the following notations for constants:

- a, b range over A,
- α ranges over $A \cup \{\tau\}$,
- μ ranges over $A \cup \{\delta, \epsilon, \tau\}$.

The signature of the process algebra $\mathrm{BPA}^{p*}_{\delta\epsilon\tau}(A)$ is built from constants μ, alternative composition $x + y$, sequential composition $x \cdot y$, and prefix iteration $\mu^* x$.

Table 1 presents an operational semantics for $\mathrm{BPA}^{p*}_{\delta\epsilon\tau}(A)$ in the style of Plotkin [18]. The binary transition relation $p \xrightarrow{\alpha} q$ denotes that the process p can evolve into q after executing action α, and the unary relation $p \downarrow$ expresses that the process p can terminate successfully.

$$\epsilon \downarrow \qquad\qquad \alpha \xrightarrow{\alpha} \epsilon$$

$$\frac{x \downarrow}{x + y \downarrow \quad y + x \downarrow} \qquad \frac{x \xrightarrow{\alpha} x'}{x + y \xrightarrow{\alpha} x' \quad y + x \xrightarrow{\alpha} x'}$$

$$\frac{x \downarrow \quad y \downarrow}{x \cdot y \downarrow} \qquad \frac{x \downarrow \quad y \xrightarrow{\alpha} y'}{x \cdot y \xrightarrow{\alpha} y'} \qquad \frac{x \xrightarrow{\alpha} x'}{x \cdot y \xrightarrow{\alpha} x' \cdot y}$$

$$\alpha^* x \xrightarrow{\alpha} \alpha^* x \qquad \frac{x \downarrow}{\mu^* x \downarrow} \qquad \frac{x \xrightarrow{\alpha} x'}{\mu^* x \xrightarrow{\alpha} x'}$$

Table 1: Action rules for $\mathrm{BPA}^{p*}_{\delta\epsilon\tau}(A)$

Our model for $\mathrm{BPA}^{p*}_{\delta\epsilon\tau}(A)$ consists of all the closed terms that can be constructed from the constants in $A \cup \{\delta, \epsilon, \tau\}$ together with the three operators. That is, the BNF grammar for the collection of process terms is as follows:

$$p \ ::= \ \mu \mid p + p \mid p \cdot p \mid \mu^* p.$$

As binding convention, * binds stronger than $\cdot$, which binds stronger than $+$. Often, $p \cdot q$ will be abbreviated to pq.

The notation $p \Rightarrow p'$ in the following definition of branching bisimulation expresses that there exists a sequence $p \xrightarrow{\tau} \cdots \xrightarrow{\tau} p'$ of (zero or more) τ-transitions.

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

1. *if $r \xrightarrow{\alpha} r'$ and $r\mathcal{B}s$, then either*
   - *$\alpha = \tau$ and $r'\mathcal{B}s$,*
   - *or $s \Rightarrow s' \xrightarrow{\alpha} s''$ with $r\mathcal{B}s'$ and $r'\mathcal{B}s''$, for some s' and s'',*
2. *if $r \downarrow$ and $r\mathcal{B}s$, then $s \Rightarrow s' \downarrow$ with $r\mathcal{B}s'$, for some s'.*

Branching bisimulation is reflexive, symmetric and transitive. The following lemma is a standard result for branching bisimulation equivalence.

Lemma 2.2 *(Stuttering Lemma.) If* $p_0 \xrightarrow{\tau} \cdots \xrightarrow{\tau} p_n$*, and if* $p_n \leftrightarrow_b p_0$*, then* $p_i \leftrightarrow_b p_0$ *for* $i = 1, ..., n-1$.

Proof. See [13].

Branching bisimulation equivalence is not a congruence. That is, equivalences $p \leftrightarrow_b p'$ and $q \leftrightarrow_b q'$ do not always imply that $p + q \leftrightarrow_b p' + q'$. For example, $\tau a \leftrightarrow_b a$, but $\tau a + b \not\leftrightarrow_b a + b$. In order to turn branching bisimulation into a congruence, we need a rootedness condition.

Definition 2.3 *Two processes* p *and* q *are* rooted branching bisimilar*, denoted by* $p \leftrightarrow_{rb} q$*, if*

1. $p \xrightarrow{\alpha} p'$ *if and only if* $q \xrightarrow{\alpha} q'$ *with* $p' \leftrightarrow_b q'$*,*
2. $p \downarrow$ *if and only if* $q \downarrow$*.*

It is not hard to see that rooted branching bisimulation equivalence is a congruence with respect to the operators, i.e. if $p \leftrightarrow_{rb} p'$ and $q \leftrightarrow_{rb} q'$, then $p + q \leftrightarrow_{rb} p' + q'$ and $p \cdot q \leftrightarrow_{rb} p' \cdot q'$ and $\mu^* p \leftrightarrow_{rb} \mu^* p'$. Furthermore, rooted branching bisimulation implies branching bisimulation. Process terms will be considered modulo rooted branching bisimulation equivalence.

The action rules for $\mathrm{BPA}_{\delta\epsilon\tau}(A)$ are 'pure' and 'well-founded', which are syntactic criteria from Groote and Vaandrager [14]. Moreover, the action rules for prefix iteration incorporate the Kleene star in the left-hand sides of their conclusions. Hence, $\mathrm{BPA}^{p*}_{\delta\epsilon\tau}(A)$ is an operationally conservative extension of $\mathrm{BPA}_{\delta\epsilon\tau}(A)$, i.e. the action rules for prefix iteration do not influence the transition systems of terms in $\mathrm{BPA}_{\delta\epsilon\tau}(A)$. See Verhoef [21] for a proof of this conservativity result.

2.2 The axioms

Table 2 contains the ten standard axioms for $\mathrm{BPA}_{\delta\epsilon\tau}(A)$, together with seven axioms for prefix iteration. The axioms MI1,2,3,6 stem from [6], and MI4,5 from [11]. The axiom MI7 is new.

In the sequel, $p = q$ will mean that the equality can be derived from the axioms in Table 2. The axiomatization is sound with respect to rooted branching bisimulation equivalence, i.e. if $p = q$ then $p \leftrightarrow_{rb} q$. Since rooted branching bisimulation is a congruence, this can be verified by checking soundness for each axiom separately, which is left to the reader. In this paper it is proved that the axiomatization is complete with respect to bisimulation, i.e. if $p \leftrightarrow_{rb} q$ then $p = q$.

The following lemma will be used in the completeness proof.

Lemma 2.4 *For each process term* p*, the collection of transitions from* p *is finite, say* $\{p \xrightarrow{\alpha_i} p_i \mid i = 1, ..., n\}$*, and* $p = \sum_{i=1}^{n} \alpha_i p_i + t(\delta, \epsilon)$*, where* $t(\delta, \epsilon)$ *is either* ϵ *if* $p \downarrow$*, or* δ *otherwise.*

Proof sketch. A straightforward exercise by induction on the size of p, i.e. on the number of function symbols in p, learns that the collection of transitions from p is finite, say

A1	$x+y$	$=$	$y+x$
A2	$(x+y)+z$	$=$	$x+(y+z)$
A3	$x+x$	$=$	x
A4	$(x+y)z$	$=$	$xz+yz$
A5	$(xy)z$	$=$	$x(yz)$
A6	$x+\delta$	$=$	x
A7	δx	$=$	δ
A8	$x\epsilon$	$=$	x
A9	ϵx	$=$	x
BE2	$\alpha(\tau(x+y)+x)$	$=$	$\alpha(x+y)$
MI1	$a\cdot a^*x+x$	$=$	a^*x
MI2	$(a^*x)y$	$=$	$a^*(xy)$
MI3	$a^*(a^*x)$	$=$	a^*x
MI4	δ^*x	$=$	x
MI5	ϵ^*x	$=$	x
MI6	τ^*x	$=$	$\tau x+x$
MI7	$\tau\cdot a^*(\tau\cdot a^*(x+y)+x)$	$=$	$\tau\cdot a^*(x+y)$

Table 2: Axioms for $\mathrm{BPA}^{p*}_{\delta\epsilon\tau}(A)$

$\{p \xrightarrow{\alpha_i} p_i \mid i=1,...,n\}$. By induction on the size of p, we find that $p=\sum_{i=1}^{n}\alpha_i p_i + t(\delta,\epsilon)$, where $t(\delta,\epsilon)$ is either ϵ if $p\downarrow$, or δ otherwise. This deduction uses the axioms A1-9+MI1,4,5 and the equation $\tau\cdot\tau^*x+x=\tau^*x$. This last equation can be derived from the axioms as follows.

$$\tau\cdot\tau^*x+x \overset{\mathrm{MI6}}{=} \tau(\tau x+x)+x \overset{\mathrm{BE2}}{=} \tau x+x \overset{\mathrm{MI6}}{=} \tau^*x. \quad \square$$

3 Completeness of the Axioms

In this section we present the completeness proof. The proof strategy will be to adapt process terms that are rooted branching bisimilar in such a way that they become strongly bisimilar. Then we can finish the proof by applying the completeness result from [11].

3.1 Strong bisimulation

First, we present some definitions and results that involve strong bisimulation equivalence from Park [17]. In this section, we do not consider τ as the special constant 'silent step', but as a regular action in the alphabet.

Definition 3.1 *Two processes p and q are* strongly bisimilar, *denoted by $p \underline{\leftrightarrow} q$, if there exists a symmetric binary relation $\mathcal{B}$ on processes which relates p and q, such that*

1. *if $r \xrightarrow{\alpha} r'$ and $r\mathcal{B}s$, then there is a transition $s \xrightarrow{\alpha} s'$ with $r'\mathcal{B}s'$,*

2. *if* $r\downarrow$ *and* $r\mathcal{B}s$, *then* $s\downarrow$.

It is proved in [11] that if $p \leftrightarrow q$, then p and q are provably equal by the axioms A1-9+MI1-5, where a ranges over $A\cup\{\tau\}$ in the axioms MI1-3. The axioms A1-9 and MI4,5, and the axioms MI1-3 where a ranges over A, are part of the axiom system in Table 2. Moreover, the variants of MI1-3 with τ for a can be derived from MI6 together with A3-5 and BE2. For the derivation of the variant of MI1, $\tau\cdot\tau^*x+x=\tau^*x$, see the proof sketch of Lemma 2.4. The easy derivations for the variants of MI2,3 are left to the reader. Hence, the completeness result from [11] induces the following proposition.

Proposition 3.2 *If* $p \leftrightarrow q$, *then* $p = q$.

Note that the converse is not true, i.e. our axiomatization for $\mathrm{BPA}^{p*}_{\delta\epsilon\tau}(A)$ is not sound with respect to strong bisimulation, due to the axioms BE2 and MI6,7 for the silent step.

The following definition stems from Van Glabbeek and Weijland [13].

Definition 3.3 *A transition* $p \stackrel{\tau}{\longrightarrow} p'$ *is* inert *if* $p \leftrightarrow_b p'$.

The next lemma follows easily from the Stuttering Lemma, together with the definition of branching bisimulation.

Lemma 3.4 *If* $p \leftrightarrow_b q$, *and if the transition systems of* p *and* q *do not contain any inert steps, then* $p \leftrightarrow q$.

3.2 Construction of basic terms

In this section we define a set of *basic terms*, and we show that each process term is provably equal to a basic term.

$(x+y)z$	$\longrightarrow$	$xz+yz$
$(xy)z$	$\longrightarrow$	$x(yz)$
δx	$\longrightarrow$	δ
ϵx	$\longrightarrow$	x
$(a^*x)y$	$\longrightarrow$	$a^*(xy)$
δ^*x	$\longrightarrow$	x
ϵ^*x	$\longrightarrow$	x
τ^*x	$\longrightarrow$	$\tau x+x$

Table 3: A term rewriting system

Table 3 contains a term rewriting system (TRS) which reduces sequential composition to its prefix counterpart and which eliminates expressions δ^*x and ϵ^*x and τ^*x. Define a weight function on terms as follows:

$$\begin{aligned} w(\mu) &= 2 \\ w(p+q) &= w(p)+w(q) \\ w(pq) &= w(p)^2w(q) \\ w(\mu^*p) &= 6w(p). \end{aligned}$$

It is easy to see that the weight of terms always strictly decreases under application of the rewrite rules. Hence, the TRS in Table 3 is terminating, i.e. there do not exist any infinite reductions. So each process term reduces to a normal form, which cannot be reduced by the TRS. It is not difficult to prove that normal forms are characterized by the following, inductively defined set.

$$p \ ::= \ \mu \mid p+p \mid \alpha p \mid a^*p.$$

We restrict the set of normal forms a bit further, in order to obtain the set of basic terms. Namely, we desire that the silent step and prefix iteration occur in the form $\tau p+q$ and $a \cdot a^*x$, respectively.

Definition 3.5 *The set of* basic terms *is defined by*

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

Lemma 3.6 *For each term p there is a basic term p' such that $p = p'$.*

Proof. The rewrite rules in the TRS are all directions of axioms, so each process term is provably equal to its normal forms. We prove by induction on size that each normal form is provably equal to a basic term.

First, we deal with the case of normal forms of size 1. Normal forms δ and ϵ and a are basic. Furthermore, $\tau \overset{\text{A6,8}}{=} \tau\epsilon + \delta$ is basic.

Next, suppose that we have proved the case for normal forms of size $\leq n$.

1. Consider a normal form $p+q$ of size $n+1$. By induction, $p = p'$ and $q = q'$ where p' and q' are basic. Hence, $p+q = p'+q'$ is basic.

2. Consider a normal form ap of size $n+1$. By induction, $p = p'$ where p' is basic. Hence, $ap = ap'$ is basic.

3. Consider a normal form τp of size $n+1$. By induction, $p = p'$ where p' is basic. Hence, $\tau p = \tau p' \overset{\text{A6}}{=} \tau p' + \delta$ is basic.

4. Consider a normal form a^*p of size $n+1$. By induction, $p = p'$ where p' is basic. Hence, $a^*p = a^*p' \overset{\text{MI1}}{=} a \cdot a^*p' + p'$ is basic. □

The next lemma will be applied in the completeness theorem. It follows immediately from the definition of a basic term, using induction on the size of the context $C[]$.

Lemma 3.7 *If $C[\tau r + s]$ is basic, then r and s and $C[r+s]$ are basic.*

3.3 The completeness theorem

From now on, process terms are considered modulo associativity and commutativity of the $+$, and this equivalence is denoted by $p =_{\text{AC}} q$.

Theorem 3.8 *The axiomatization* A1-9+BE2+MI1-7 *for* $\text{BPA}^{p*}_{\delta\epsilon\tau}(A)$ *is complete with respect to rooted branching bisimulation equivalence.*

Proof. First, we prove the following two statements in parallel, by induction on size.

A. For each basic term p there is a term p' with $\tau p = \tau p'$, such that the transition system of p' does not contain any inert steps.

B. If p and q are basic terms with $p \leftrightarrow_b q$, then $\tau p = \tau q$.

Let A_n denote statement A for basic terms p of size $\leq n$, and let B_n denote statement B for basic terms p and q of sizes $\leq n$.

Suppose that we have already proved A_{n-1} and B_{n-1} for some $n \geq 0$; we prove A_n and B_n. First we prove A_n, so assume a basic term p of size n. Suppose that the transition system of p contains an inert step, for else we are done.

Recall that the basic term p is a sum of terms δ, ϵ, a, $\alpha p'$, and $a(a^* p')$, with p' a basic term, where summands $\tau p'$ always occur as an argument of alternative composition. Hence, the inert step in the transition system of p is caused by a subterm $\tau r + s$ of p which occurs in p in one of the following three forms.

- either $p =_{\mathrm{AC}} \tau r + s$, with $r \leftrightarrow_b \tau r + s$,
- or p has a subterm $\alpha(\tau r + s)$, with $r \leftrightarrow_b \tau r + s$,
- or p has a subterm $a \cdot a^*(\tau r + s)$, with $r \leftrightarrow_b a^*(\tau r + s)$.

In the first two cases, $r \leftrightarrow_b \tau r + s$ yields $r \leftrightarrow_b r + s$, and in the last case $r \leftrightarrow_b a^*(\tau r + s)$ yields $r \leftrightarrow_b a^*(r + s)$. We consider the three cases separately.

1. $p =_{\mathrm{AC}} \tau r + s$ and $r \leftrightarrow_b r + s$.

   According to Lemma 3.7, r and s are basic, so $r + s$ is basic. Furthermore, the size of $r + s$ is smaller than the size of $\tau r + s$, which means that the size of $r + s$ is smaller than n. So B_{n-1} together with $r \leftrightarrow_b r + s$ yield $\tau r = \tau(r + s)$. Hence, $\tau p =_{\mathrm{AC}} \tau(\tau r + s) = \tau(\tau(r + s) + s) \stackrel{\mathrm{BE2}}{=} \tau(r + s)$.

   Since $r + s$ is a basic term of size $< n$, A_{n-1} yields that there is a term p' with $\tau(r + s) = \tau p'$, such that the transition system of p' does not contain any inert steps. Since $\tau p = \tau(r + s) = \tau p'$, we are done.

2. $p =_{\mathrm{AC}} C[\alpha(\tau r + s)]$ and $r \leftrightarrow_b r + s$.

   Again, B_{n-1} yields $\tau r = \tau(r + s)$. Hence, $p =_{\mathrm{AC}} C[\alpha(\tau r + s)] = C[\alpha(\tau(r + s) + s)] \stackrel{\mathrm{BE2}}{=} C[\alpha(r + s)]$.

   According to Lemma 3.7 $C[\alpha(r+s)]$ is basic, and its size is smaller than n, so A_{n-1} yields that there is a term p' with $\tau C[\alpha(r + s)] = \tau p'$, such that the transition system of p' does not contain any inert steps. Since $\tau p = \tau C[\alpha(r + s)] = \tau p'$, we are done.

3. $p =_{\mathrm{AC}} C[a \cdot a^*(\tau r + s)]$ with $r \leftrightarrow_b a^*(r + s)$.

   $a^*(r + s)$ is a basic term of size $< n$, so B_{n-1} together with $r \leftrightarrow_b a^*(r + s)$ yield $\tau r = \tau \cdot a^*(r + s)$. Hence, $p =_{\mathrm{AC}} C[a \cdot a^*(\tau r + s)] = C[a \cdot a^*(\tau \cdot a^*(r + s) + s)] \stackrel{\mathrm{MI7}}{=} C[a \cdot a^*(r + s)]$.

   $C[a \cdot a^*(r + s)]$ is a basic term of size $< n$, so A_{n-1} yields that there is a term p' with $\tau C[a \cdot a^*(r + s)] = \tau p'$, such that the transition system of p' does not contain any inert steps. Since $\tau p = \tau C[a \cdot a^*(r + s)] = \tau p'$, we are done.

Next, we prove B_n. Assume basic terms p and q of sizes $\leq n$ with $p \leftrightarrow_b q$. A_n yields that there exist terms p' and q' of which the transition systems do not contain any inert steps, such that $\tau p = \tau p'$ and $\tau q = \tau q'$. Soundness of the axioms yields $p \leftrightarrow_b \tau p \leftrightarrow_{rb} \tau p' \leftrightarrow_b p'$ and $q \leftrightarrow_b \tau q \leftrightarrow_{rb} \tau q' \leftrightarrow_b q'$, so $p' \leftrightarrow_b p \leftrightarrow_b q \leftrightarrow_b q'$. Since the transition systems of p' and q' do not contain any inert steps, Lemma 3.4 yields $p' \leftrightarrow q'$. Hence, Proposition 3.2 yields $p' = q'$, so $\tau p = \tau p' = \tau q' = \tau q$.

Finally, we show that statement B implies the desired completeness result. First, from the axioms we deduce the following equation.

$$\alpha\tau \;=\; \alpha. \tag{1}$$

It follows by substituting δ for x and ϵ for y in axiom BE2, and applying axioms A6,8.

Let p and q be process terms with $p \leftrightarrow_{rb} q$. According to the definition of rooted branching bisimulation, the sets of possible transitions from p and q are $\{p \xrightarrow{\alpha_i} p_i \mid i = 1, ..., n\}$ and $\{q \xrightarrow{\alpha_i} q_i \mid i = 1, ..., n\}$ with $p_i \leftrightarrow_b q_i$ for $i = 1, ..., n$, and $p\downarrow$ if and only if $q\downarrow$. Hence, Lemma 2.4 yields $p = \sum_{i=1}^n \alpha_i p_i + t(\delta, \epsilon)$ and $q = \sum_{i=1}^n \alpha_i q_i + t(\delta, \epsilon)$, where $t(\delta, \epsilon)$ is either δ or ϵ. According to Lemma 3.6, there exist basic terms p_i' and q_i' such that $p_i = p_i'$ and $q_i = q_i'$. Since $p_i \leftrightarrow_b q_i$, and since the axioms are sound, we find $p_i' \leftrightarrow_{rb} p_i \leftrightarrow_b q_i \leftrightarrow_{rb} q_i'$, so $p_i' \leftrightarrow_b q_i'$. Hence, B yields $\tau p_i' = \tau q_i'$ for $i = 1, ..., n$, so

$$\begin{array}{rcl} p &=& \sum_{i=1}^n \alpha_i p_i + t(\delta,\epsilon) \;=\; \sum_{i=1}^n \alpha_i p_i' + t(\delta,\epsilon) \;\stackrel{(1)}{=}\; \sum_{i=1}^n \alpha_i \tau p_i' + t(\delta,\epsilon) \\ &=& \sum_{i=1}^n \alpha_i \tau q_i' + t(\delta,\epsilon) \;\stackrel{(1)}{=}\; \sum_{i=1}^n \alpha_i q_i' + t(\delta,\epsilon) \;=\; \sum_{i=1}^n \alpha_i q_i + t(\delta,\epsilon) \;=\; q. \;\square \end{array}$$

3.4 Complete axiomatizations for reducts of $BPA^{p*}_{\delta\epsilon\tau}(A)$

We study how the axiomatization for $\mathrm{BPA}^{p*}_{\delta\epsilon\tau}(A)$ is to be adapted in case the deadlock δ and/or the empty process ϵ are removed from the syntax.

In the setting without ϵ, the equality $\alpha\tau = \alpha$ can no longer be deduced from axiom BE2, so it has to be added to the axiom system. Since ϵ is absent, the more general equality $x\tau = x$ is sound. Furthermore, we can replace axiom BE2 by axiom B2, in order to get rid of the variable α in the axioms.

B1	$x\tau$	$=$	x
B2	$\tau(\tau(x+y)+x)$	$=$	$\tau(x+y)$

Finally, without ϵ we do not need axioms A8,9 and MI5. With a similar proof scheme as has been used for the completeness proof for $\mathrm{BPA}^{p*}_{\delta\epsilon\tau}(A)$, with the cases for ϵ omitted, we can obtain the following result.

Theorem 3.9 *The axiomatization* A1-7+B1,2+MI1-4,6,7 *for* $\mathrm{BPA}^{p*}_{\delta\tau}(A)$ *is complete with respect to rooted branching bisimulation equivalence.*

In the setting of $\mathrm{BPA}^{p*}_{\epsilon\tau}(A)$ too, the equality $\alpha\tau = \alpha$ cannot be deduced from the axioms. Axiom B1 is not sound, due to the presence of the empty process, so we add the equality $\alpha\tau = \alpha$ to the axiom system, under the name BE1. Furthermore, without deadlock the following equality can no longer be deduced from axiom MI7, so it has to be added to the axiom system.

MI8	$a \cdot a^*(\tau \cdot a^* x)$	$=$	$a \cdot a^* x$

Finally, without δ we do not need axioms A6,7 and MI4. With a similar proof scheme as has been used for the completeness proof for $\mathrm{BPA}^{p*}_{\delta\epsilon\tau}(A)$, we can obtain the following result.

Theorem 3.10 *The axiomatization* A1-5,8,9+BE1+B2+MI1-3,5-8 *for* $\mathrm{BPA}^{p*}_{\epsilon\tau}(A)$ *is complete with respect to rooted branching bisimulation equivalence.*

Finally, if both ϵ and δ are omitted, then we can deduce the following result.

Theorem 3.11 *The axiomatization* A1-5+B1,2+MI1-3,6-8 *for* $\mathrm{BPA}^{p*}_{\tau}(A)$ *is complete with respect to rooted branching bisimulation equivalence.*

References

[1] L. Aceto, W.J. Fokkink, R.J. van Glabbeek, and A. Ingólfsdóttir. Axiomatizing prefix iteration with silent steps. Technical Report RS-95-56, BRICS, Aalborg University, 1995. Available at `http://www.brics.aau.dk/BRICS/`. To appear in *Information and Computation*.

[2] L. Aceto and J.F. Groote. A complete equational axiomatization for MPA with string iteration. Technical Report RS-95-28, BRICS, Aalborg University, 1995. Available at `http://www.brics.aau.dk/BRICS/`.

[3] L. Aceto and A. Ingólfsdóttir. A complete equational axiomatization for prefix iteration with silent steps. Technical Report RS-95-5, BRICS, Aalborg University, 1995. Available at `http://www.brics.aau.dk/BRICS/`.

[4] J.C.M. Baeten and J.A. Bergstra. Discrete time process algebra. In *Proceedings 3rd Conference on Concurrency Theory (CONCUR'92)*, Stony Brook (1992), W.R. Cleaveland, ed., Lecture Notes in Computer Science 630, Springer, pp. 401–420. Revised version to appear in *Formal Aspects of Computing*.

[5] J.C.M. Baeten and R.J. van Glabbeek. Another look at abstraction in process algebra. In *Proceedings 14th International Colloquium on Automata, Languages and Programming (ICALP'87)*, Karlsruhe (1987), Th. Ottmann, ed., Lecture Notes in Computer Science 267, Springer, pp. 84–94.

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

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

[8] J.A. Bergstra and J.W. Klop. Verification of an alternating bit protocol by means of process algebra. In *Proceedings Spring School on Mathematical Methods of Specification and Synthesis of Software Systems '85*, Wendisch-Rietz (1985), W. Bibel and K.P. Jantke, eds., Lecture Notes in Computer Science 215, Springer, pp. 9–23.

[9] W.J. Fokkink. A complete equational axiomatization for prefix iteration. *Information Processing Letters 52*, 6 (1994), 333–337.

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

[11] W.J. Fokkink and H. Zantema. Prefix iteration in basic process algebra: applying termination techniques. In *Proceedings 2nd Workshop on the Algebra of Communicating Processes (ACP'95)*, Eindhoven (1995), A. Ponse, C. Verhoef, and S.F.M. van Vlijmen, eds., Technical Report CS-95-14, Eindhoven University of Technology, pp. 139–156.

[12] R.J. van Glabbeek. Branching bisimulation as a tool in the analysis of weak bisimulation. Unpublished manuscript, 1995. Available at `http://boole.stanford.edu/~rvg/pub/tool.dvi.gz`.

[13] R.J. van Glabbeek and W.P. Weijland. Branching time and abstraction in bisimulation semantics. In *Proceedings 11th IFIP World Computer Congress*, San Francisco (1989), G.X. Ritter, ed., North-Holland, pp. 613–618. To appear in *Journal of the ACM*.

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

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

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

[17] D.M.R. Park. Concurrency and automata on infinite sequences. In *Proceedings 5th GI Conference*, Karlsruhe (1981), P. Deussen, ed., Lecture Notes in Computer Science 104, Springer, pp. 167–183.

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

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

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

[21] C. Verhoef. A general conservative extension theorem in process algebra. In *Proceedings IFIP Conference on Programming Concepts, Methods and Calculi (PROCOMET'94)*, San Miniato (1994), E.-R. Olderog, ed., IFIP Transactions A-56, Elsevier, pp. 149–168.

[22] H. Zantema and A. Geser. A complete characterization of termination of $0^p1^q \to 1^r0^s$. In *Proceedings 6th Conference on Rewriting Techniques and Applications (RTA'95)*, Kaiserslautern (1995), J. Hsiang, ed., Lecture Notes in Computer Science 914, Springer, pp. 41–55.