An Axiomatization for the Terminal Cycle

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

Abstract

Milner proposed an axiomatization for the Kleene star in basic process algebra, in the presence of deadlock and empty process, modulo bisimulation equivalence. In this paper, Milner's axioms are adapted to the terminal cycle x^ω, which executes x infinitely many times in a row, and it is shown that this axiomatization is complete for the terminal cycle in basic process algebra with deadlock and empty process modulo bisimulation.

1 Introduction

Kleene [15] 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. A feature of the Kleene star is that on the one hand it can express recursion, while on the other hand it can be captured in equational laws. Hence, one does not need meta-principles such as the Recursive Specification Principle from Bergstra and Klop [10]. Kleene formulated several equations for his operator, notably the defining equation $x^*y = x(x^*y) + y$. Copi, Elgot and Wright [11] proposed a simplification of Kleene's setting, e.g. they defined a unary version x^* of the Kleene star. In the presence of the empty process, the unary and the binary Kleene star are equally expressive.

Salomaa [20] presented a finite complete axiomatization for the Kleene star in language theory, modulo completed trace equivalence, which incorporates one conditional axiom, namely, in our notation,

if $x = y \cdot x + z$, and y cannot terminate immediately, then $x = y^*z$.

The requirement 'y cannot terminate immediately' can be defined inductively on the syntax. However, according to Kozen [16] this requirement is not algebraic, in the sense that it is not preserved under substitution of terms for actions. He proposed an alternative complete finite axiomatization, again with conditional axioms, which does not have this drawback. Kozen's adaptations of Salomaa's conditional axiom, however, are not sound with respect to bisimulation equivalence.

Milner [17] was the first to study the (unary) Kleene star modulo bisimulation, and proposed an axiomatization for it, being an adaptation of Salomaa's axiom system.

Namely, he removed two axioms which are typical for completed traces, but which are not sound with respect to bisimulation (namely $x(y+z)=xy+xz$ and $x\delta=\delta$), and added one axiom which can be derived in Salomaa's setting ($\delta x=\delta$). Milner raised the question whether his axiomatization is complete for the Kleene star in process theory, modulo bisimulation equivalence. Milner [17, page 461] remarks that this question may be hard to answer: "The difficulty is that the method (...) of Salomaa's original completeness proof cannot be applied directly, since -in contrast with the case of languages- an arbitrary system of guarded equations (...) cannot in general be solved in star expressions". Therefore, a new proof method is required, and these last few years I made several attempts to find such a method, but in vain. In order to try and find a solution for Milner's question in several steps, I decided to study a natural instantiation of the binary Kleene star, namely $x^*\delta$. Since the deadlock δ blocks all behaviour, this construct executes x an infinite number of times in a row. In [4, Definition 17], the construct $x^*\delta$ is called 'terminal cycle', motivated by the fact that, modulo bisimulation, it is impossible to escape from loops $x^*\delta$, so that they lie at the leaves of the tree expansion of a process. We will denote the unary operator $x^*\delta$ by x^ω. The terminal cycle is closely related to the Kleene star, and shares several of its characteristics. In this paper, the terminal cycle is studied in the setting of Basic Process Algebra [9] with deadlock and empty process, and the process algebra considered is denoted by $\mathrm{BPA}^\omega_{\delta\varepsilon}$.

The terminal cycle is of interest on its own right. For example, the Kleene star is incorporated in the ToolBus [8], where it is used almost exclusively in the form of a terminal cycle. Another typical example of the terminal cycle is the failure-driven loop in Prolog [22]. Generally speaking, the terminal cycle can be used to formally describe programs that repeat a certain procedure without end.

The three axioms for the unary Kleene star in Milner's axiom system (being Kleene's defining equation, Salomaa's conditional axiom and an equation which describes the interplay of Kleene star and empty process) have obvious counterparts for the terminal cycle. It will be shown that these three axioms, together with the standard axioms for $\mathrm{BPA}_{\delta\varepsilon}$, make a complete axiomatization for $\mathrm{BPA}^\omega_{\delta\varepsilon}$ modulo bisimulation. The proof is based on a strategy that originates from [12]. It also uses several new techniques, which will be described in detail. Hopefully, these techniques will turn out to be applicable in a possible proof of Milner's conjecture.

Sewell [21] proved that there does not exist a complete finite equational axiomatization for the Kleene star in combination of the deadlock δ modulo bisimulation, due to the equivalences $a^*\delta \leftrightarrow (a^p)^*\delta$ for prime numbers p, which hold with respect to bisimulation. Since these equivalences are also present in $\mathrm{BPA}^\omega_{\delta\varepsilon}$, Sewell's argument can be copied to conclude that neither does there exist a complete finite equational axiomatization for $\mathrm{BPA}^\omega_{\delta\varepsilon}$. Hence, the adaptation of Salomaa's conditional axiom for the terminal cycle is essential for the obtained completeness result.

Bergstra, Bethke and Ponse [7] suggested a finite equational axiomatization for basic process algebra with the binary Kleene star, denoted by BPA^*, modulo bisimulation. Their conjecture that it is complete was solved by Fokkink and Zantema [14]. (In contrast with this result, Aceto, Fokkink and Ingólfsdóttir [3] showed that there does not exist a complete finite equational axiomatization for BPA^* modulo any process semantics in between ready simulation and completed traces.) In [12], a new proof for the completeness result from [14] was presented. The proof technique that was introduced in [12] was applied successfully not only in this paper, but also in [1, 2].

Acknowledgements. This research was initiated by a question from Alban Ponse. Luca Aceto provided useful comments, and Jan Bergstra and Alban Ponse are thanked for stimulating discussions.

2 Terminal Cycles in Process Algebra

This section introduces the basic notions.

2.1 Syntax

We assume a non-empty alphabet A of atomic actions, with typical elements a, b, c. We also assume two special constants δ, which represents deadlock, and ε, which represents the empty process, and ξ and η will range over $\{\delta, \varepsilon\}$. Furthermore, we have two binary operators: alternative composition $x + y$, which combines the behaviours of x and y, and sequential composition $x \cdot y$, which puts the behaviours of x and y in sequence. Finally, we have the unary terminal cycle x^ω, which executes x infinitely many times in a row. The language $\mathrm{BPA}^{\omega}_{\delta\varepsilon}(A)$, with typical elements $p, q, ..., w$, consists of all the terms that can be constructed from the atomic actions, the two special constants, the two binary composition operators, and the terminal cycle. That is, the BNF grammar for the collection of process terms is as follows:

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

Remark: In fact, the presence of the special constant δ in the syntax is redundant, because it can already be expressed in $\mathrm{BPA}^{\omega}_{\varepsilon}$ modulo bisimulation; we will see that ε^ω is bisimilar with δ.

We will use $\equiv$ to denote syntactic equality. In the sequel the sequential composition operator will often be omitted, so pq denotes $p \cdot q$. As binding convention, alternative composition binds weaker than sequential composition and the terminal cycle.

2.2 Operational Semantics

Table 1 presents an operational semantics for $\mathrm{BPA}^{\omega}_{\delta\varepsilon}(A)$ in Plotkin style [19]. The binary relation $x \stackrel{a}{\longrightarrow} x'$ represents that process x can evolve into process x' by the execution of action a, while the unary predicate $x\downarrow$ denotes that process x can terminate immediately.

Definition 2.1 *p' is a* derivative *of p if p can evolve into p' by zero or more transitions.*
p' is a proper *derivative of p if p can evolve into p' by one or more transitions.*

In the sequel, p' and p'' will denote derivatives of process term p. The following lemma can easily be deduced, using structural induction.

Lemma 2.2 *Each process term in $\mathrm{BPA}^{\omega}_{\delta\varepsilon}(A)$ has only finitely many derivatives.*

$$\varepsilon\downarrow \qquad\qquad a \xrightarrow{a} \varepsilon$$

$$\frac{x\downarrow}{(x+y)\downarrow \quad (y+x)\downarrow} \qquad \frac{x \xrightarrow{a} x'}{x+y \xrightarrow{a} x' \quad y+x \xrightarrow{a} x'}$$

$$\frac{x\downarrow \quad y\downarrow}{(x\cdot y)\downarrow} \qquad \frac{x\downarrow \quad y \xrightarrow{a} y'}{x\cdot y \xrightarrow{a} y'} \qquad \frac{x \xrightarrow{a} x'}{x\cdot y \xrightarrow{a} x'\cdot y}$$

$$\frac{x \xrightarrow{a} x'}{x^\omega \xrightarrow{a} x'\cdot(x^\omega)}$$

Table 1: Transition rules for $\mathrm{BPA}^\omega_{\delta\varepsilon}(A)$

Process terms are considered modulo bisimulation equivalence from Park [18]. Intuitively, two processes are bisimilar if they have the same branching structure.

Definition 2.3 *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 $r\ B\ s$ and $r \xrightarrow{a} r'$, then there is a transition $s \xrightarrow{a} s'$ such that $r'\ B\ s'$;*
- *if $r\ B\ s$ and $r\downarrow$, then $s\downarrow$.*

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 $p^\omega \underline{\leftrightarrow} (p')^\omega$.

Proposition 2.4 *Bisimulation equivalence is a congruence over* $\mathrm{BPA}^\omega_{\delta\varepsilon}(A)$*.*

Namely, the transition rules in Table 1 are in the 'path' format of Baeten and Verhoef [6], which guarantees that the generated bisimulation equivalence is a congruence. See [6] for the definition of the path format, and for a proof of this congruence result.

Remark: The congruence proof in [6] uses the extra assumption that the transition rules are 'well-founded'; Fokkink and Van Glabbeek [13] showed that this requirement can be omitted.

2.3 Axiomatization

The axiomatization for $\mathrm{BPA}^\omega_{\delta\varepsilon}(A)$ in Table 2 consists of the standard axioms A1-9 for $\mathrm{BPA}_{\delta\varepsilon}(A)$, the defining equation TC1 for the terminal cycle, an equation TC2 which describes the interplay of the terminal cycle with the empty process, and RSP^ω which is the adaptation of Salomaa's conditional axiom for the terminal cycle. In this last axiom, $y\not\downarrow$ expresses that y cannot terminate immediately. (This condition is similar to the so-called guardedness restriction in the Recursive Specification Principle from Bergstra and Klop [10].) Without this extra condition, RSP^ω would not be sound,

A1	$x + y$	$=$	$y + x$	
A2	$(x + y) + z$	$=$	$x + (y + z)$	
A3	$x + x$	$=$	x	
A4	$(x + y) \cdot z$	$=$	$x \cdot z + y \cdot z$	
A5	$(x \cdot y) \cdot z$	$=$	$x \cdot (y \cdot z)$	
A6	$x + \delta$	$=$	x	
A7	$\delta \cdot x$	$=$	δ	
A8	$x \cdot \varepsilon$	$=$	x	
A9	$\varepsilon \cdot x$	$=$	x	
TC1	$x \cdot (x^\omega)$	$=$	x^ω	
TC2	$(x + \varepsilon)^\omega$	$=$	x^ω	
RSP$^\omega$	$x = y \cdot x \wedge y \not\downarrow \implies x = y^\omega$			

Table 2: Axiom system for $\mathrm{BPA}^\omega_{\delta\varepsilon}(A)$

because then due to the axiom A9, $x = \epsilon x$, it would imply $x = \epsilon^\omega$, which is clearly unsound. We recall that the three defining inference rules for the predicate $\downarrow$ were presented in Table 1. Its negative counterpart $\not\downarrow$ can also be defined inductively.

In the sequel, $p = q$ will mean that this equality can be derived from the axioms in Table 2. The axiomatization is sound for $\mathrm{BPA}^\omega_{\delta\varepsilon}(A)$ with respect to bisimulation equivalence, i.e., if $p = q$ then $p \mathrel{\underline{\leftrightarrow}} q$.

Proposition 2.5 *The axiomatization* A1-9+TC1,2+RSP$^\omega$ *is sound for* $\mathrm{BPA}^\omega_{\delta\varepsilon}(A)$.

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

From now on, terms are considered modulo associativity and commutativity of the $+$, and modulo prefixing with ε, that is, modulo the axioms A1,2,9. We write $p \cong q$ if p and q can be equated by axioms A1,2,9. As usual, $\sum_{i=1}^n p_i$ represents the term $p_1 + \ldots + p_n$, and the p_i are called the summands of this term. We will take care to avoid empty sums, where $\sum_{i\in\emptyset} p_i + q$ is not considered empty.

Definition 2.6 *For each process term p, its collection of possible transitions is finite, say $\{p \xrightarrow{a_i} p_i \mid i = 1, ..., n\}$. The sum*

$$\sum_{i=1}^n a_i p_i + \xi$$

is called the expansion *of p, where the $\xi \equiv \varepsilon$ if $p \downarrow$ and $\xi \equiv \delta$ if $p \not\downarrow$.*

Lemma 2.7 *Each process term p is provably equal to its expansion.*

Proof: Straightforward, by structural induction on p. We only treat the case where p is of the form q^ω. The remaining cases, where p is of the from a, ξ, $q+r$ or qr, are standard and left to the reader.

Let $p \cong q^\omega$. By induction, q equals its expansion:

$$q \;=\; \sum_{i=1}^{n} a_i q_i + \xi,$$

where $\{q \xrightarrow{a_i} q_i \mid i = 1,...,n\}$ is the set of transitions of q. Then according to the transition rule for the terminal cycle in Table 1, $\{q^\omega \xrightarrow{a_i} q_i(q^\omega) \mid i = 1,...,n\}$ is the set of transitions of q^ω. Furthermore, $q^\omega \not\downarrow$, so the expansion of q^ω takes the form

$$\sum_{i=1}^{n} a_i(q_i(q^\omega)) + \delta.$$

In order to show that q^ω equals its expansion, we distinguish two cases: either $\xi \equiv \delta$ or $\xi \equiv \varepsilon$.

- CASE 1: $\xi \equiv \delta$. Then

$$q^\omega \stackrel{\text{TC1}}{=} q(q^\omega) = (\sum_{i=1}^{n} a_i q_i + \delta)(q^\omega) \stackrel{\text{A4,5,7}}{=} \sum_{i=1}^{n} a_i(q_i(q^\omega)) + \delta.$$

- CASE 2: $\xi \equiv \varepsilon$. Then

$$q^\omega = (\sum_{i=1}^{n} a_i q_i + \varepsilon)^\omega \stackrel{\text{A6,TC2}}{=} (\sum_{i=1}^{n} a_i q_i + \delta)^\omega \stackrel{\text{TC1}}{=} (\sum_{i=1}^{n} a_i q_i + \delta)((\sum_{i=1}^{n} a_i q_i + \delta)^\omega)$$

$$\stackrel{\text{TC2,A6}}{=} (\sum_{i=1}^{n} a_i q_i + \delta)(q^\omega) \stackrel{\text{A4,5,7}}{=} \sum_{i=1}^{n} a_i(q_i(q^\omega)) + \delta. \qquad \Box$$

2.4 Normed Processes

The following terminology stems from [5].

Definition 2.8 *A process term p is called* normed *if it can terminate in finitely many transitions, that is,* $p \xrightarrow{a_1} p_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} p_n \downarrow$.

The class of normed processes can be defined inductively as follows:

- $a \in A$ and ε are normed;
- if p or q is normed, then $p+q$ is normed;
- if p and q are normed, then pq is normed.

We derive an equation from the axioms which involves (closed) process terms that are not normed. In this derivation we need the following lemma.

Lemma 2.9 *If $p\downarrow$, then $p = q + \varepsilon$ where $q \not\downarrow$.*

This lemma follows easily by structural induction on p; the details are left to the reader.

Lemma 2.10 *If p is not normed, then $pq \;=\; p$.*

Proof: We apply structural induction on p.

1. $p \equiv \delta$. Then $\delta q \stackrel{\text{A7}}{=} \delta$.

2. $p \cong p_0 + p_1$ with $p_0 \not\downarrow$ and $p_1 \not\downarrow$. Then induction yields

$$(p_0 + p_1)q \stackrel{\text{A4}}{=} p_0 q + p_1 q = p_0 + p_1.$$

3. $p \cong p_0 p_1$ with $p_0 \not\downarrow$ or $p_1 \not\downarrow$. We distinguish two cases.

   - $p_0 \not\downarrow$. Then induction yields

$$(p_0 p_1)q \stackrel{\text{A5}}{=} p_0(p_1 q) = p_0 = p_0 p_1.$$

   - $p_1 \not\downarrow$. Then induction yields

$$(p_0 p_1)q \stackrel{\text{A5}}{=} p_0(p_1 q) = p_0 p_1.$$

4. $p \cong p_0^\omega$. We distinguish two cases.

   - $p_0 \not\downarrow$.

$$(p_0^\omega)q \stackrel{\text{TC1}}{=} (p_0(p_0^\omega))q \stackrel{\text{A5}}{=} p_0((p_0^\omega)q).$$

     Since $p_0 \not\downarrow$, RSP$^\omega$ then yields $(p_0^\omega)q = p_0^\omega$.

   - $p_0 \downarrow$. Then according to Lemma 2.9 $p_0 = r + \varepsilon$ with $r \not\downarrow$.

$$(r^\omega)q \stackrel{\text{TC1}}{=} (r(r^\omega))q \stackrel{\text{A5}}{=} r((r^\omega)q).$$

     Since $r \not\downarrow$, RSP$^\omega$ then yields $(r^\omega)q = r^\omega$. So, finally,

$$(p_0^\omega)q = ((r+\varepsilon)^\omega)q \stackrel{\text{TC2}}{=} (r^\omega)q = r^\omega \stackrel{\text{TC2}}{=} (r+\varepsilon)^\omega = p_0^\omega. \qquad \square$$

3 Preliminaries

This section presents preliminaries that will be used in the completeness proof. Most of the definitions in this section originate from [12].

3.1 An Ordering on Pairs of Terms

The following weight function will be used to formulate an ordering on pairs of terms.

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

Note that g-value is invariant under axioms A1,2,9. The following lemma can easily be deduced, using structural induction.

Lemma 3.1 *If p' is a derivative of p, then $g(p') \leq g(p)$.*

We consider pairs of process terms modulo commutativity. The ordering $<$ on pairs of process terms, which is taken from [12], is defined as follows.

Definition 3.2 *The ordering $<$ on pairs of terms is obtained by taking the transitive closure of the three relations below.*

1. $(r, s) < (p, q)$ *if* $g(r) < g(p)$ *and* $g(s) < g(p)$;
2. $(r, s) < (p, q)$ *if* $g(r) < g(p)$ *and* $g(s) \leq g(q)$;
3. $(p', q') < (p, q)$ *if p' is a derivative of p, and not vice versa, and q' is a derivative of q.*

In the proof of the completeness theorem we will apply induction with respect to this ordering, so we need to know that it is well-founded.

Lemma 3.3 *The ordering $<$ on pairs of process terms is well-founded modulo $\cong$.*

Proof: First, we collect some properties for this ordering.

1. If $(r, s) < (p, q)$ because $g(r) < g(p)$ and $g(s) < g(p)$, then

   $$\max\{g(r), g(s)\} < \max\{g(p), g(q)\}.$$

2. If $(r, s) < (p, q)$ because $g(r) < g(p)$ and $g(s) \leq g(q)$, then

   $$\max\{g(r), g(s)\} \leq \max\{g(p), g(q)\};$$
   $$g(r) + g(s) < g(p) + g(q).$$

3. Let $(p', q') < (p, q)$ because p' is a derivative of p, and not vice versa, and q' is a derivative of q. According to Lemma 3.1, $g(p') \leq g(p)$ and $g(q') \leq g(q)$, so

   $$\max\{g(p'), g(q')\} \leq \max\{g(p), g(q)\};$$
   $$g(p') + g(q') \leq g(p) + g(q).$$

   Furthermore, since p' is a derivative of p, but not vice versa, the number of derivatives is strictly smaller than the number of derivatives of p. Hence, the number of derivatives of p' together with the number derivatives of q' is strictly smaller than the number of derivatives of p together with the number of derivatives of q.

4. If $p \cong q$, then $g(p) = g(q)$, and p and q share the same derivatives modulo $\cong$.

Suppose that $(p_i, q_i) > (p_{i+1}, q_{i+1})$ for $i = 1, 2, \ldots$. Owing to the previous observations, such a chain has to be finite. Namely:

1. $\max\{g(p_i), g(q_i)\}$ strictly decreases under application of the first item of Definition 3.2, and it does not increase under application of the second and third item, and is invariant modulo $\cong$. Hence, there can be only finitely many applications of this first item.

2. $g(p_i) + g(q_i)$ strictly decreases under application of the second item of Definition 3.2, and it does not increase under application of the third item and is invariant modulo $\cong$. Hence, there can be only finitely many applications of this second item.

3. The sum of the number of derivatives of p_i and the number derivatives of q_i strictly decreases under application of the third item of Definition 3.2, and is invariant modulo $\cong$. Hence, there can be only finitely many applications of this third item. $\Box$

3.2 Basic Terms

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

Definition 3.4 *The set $\mathbb{B}$ of basic process terms is defined inductively as follows:*

1. *if $a_1, \ldots, a_n \in A$ and $p_1, \ldots, p_n \in \mathbb{B}$ and $\xi \in \{\delta, \varepsilon\}$, then* $\sum_{i=1}^{n} a_i p_i + \xi \in \mathbb{B}$*;*

2. *if $p \in \mathbb{B}$ and $p \not\downarrow$ and p' is a proper derivative of p, then* $p'(p^\omega) \in \mathbb{B}$.

For notational convenience, we distinguish the following set $\mathbb{C}$ of cycles in $\mathbb{B}$.

Definition 3.5 $\mathbb{C} = \{p'(p^\omega) \mid p'(p^\omega) \in \mathbb{B}\}$.

The definitions for $\mathbb{B}$ and $\mathbb{C}$ originate from [12]. We now deduce several results for basic terms which will be needed in the completeness proof.

Lemma 3.6 1. *If $p \in \mathbb{C}$ and $p \xrightarrow{a} p'$, then $p' \in \mathbb{C}$.*

2. *If $p \in \mathbb{B}$ and $p \xrightarrow{a} p'$, then $p' \in \mathbb{B}$.*

3. *If $p \in \mathbb{B}$ and $p \xrightarrow{a} p'$ and p is a derivative of p', then $p \in \mathbb{C}$.*

Proof of 3.6.1: Let $p \cong q'(q^\omega)$. According to the transition rules for sequential composition in Table 1, there are two ways in which $q'(q^\omega)$ can make a transition.

- CASE 1: If $q' \downarrow$ and $q^\omega \xrightarrow{a} r$, then $q'(q^\omega) \xrightarrow{a} r$.

  According to the transition rule for the terminal cycle in Table 1, there is only one way in which q^ω can make a transition: if $q \xrightarrow{a} q''$ then $q^\omega \xrightarrow{a} q''(q^\omega)$. Hence, $r \cong q''(q^\omega) \in \mathbb{C}$.

- CASE 2: If $q' \xrightarrow{a} q''$ then $q'(q^\omega) \xrightarrow{a} q''(q^\omega)$. The derivative $q''(q^\omega)$ is in $\mathbb{C}$. $\square$

Proof of 3.6.2: We distinguish two cases: either $p \in \mathbb{C}$ or $p \notin \mathbb{C}$.

- CASE 1: If $p \in \mathbb{C}$, then according to Lemma 3.6.1 $p' \in \mathbb{C} \subset \mathbb{B}$.

- CASE 2: If $p \notin \mathbb{C}$, then it is of the form $\sum_i a_i p_i + \xi$, where the p_i are basic terms. Then p' is of the form p_i for some i (since we consider terms modulo A9), so p' is basic. Note that p' has smaller size than p. $\square$

Proof of 3.6.3: Since p and p' are derivatives of each other, there exists a derivation

$$q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} q_n \qquad n \geq 1.$$

where $q_0 \cong p$ and $q_1 \cong p'$ and $q_n \cong p$. Since $q_0 \in \mathbb{B}$, Lemma 3.6.2 yields $q_i \in \mathbb{B}$ for $i = 0, ..., n$.

Suppose that $q_i \notin \mathbb{C}$ for all i. We observed in the proof of Lemma 3.6.2 that then q_{i+1} has smaller size than q_i for $k = 0, ..., n-1$, so q_n has smaller size than q_0. This contradicts the fact that $q_0 \cong q_n$. Hence, $q_j \in \mathbb{C}$ for some j. Since the derivatives of a term in $\mathbb{C}$ are all in $\mathbb{C}$ (Lemma 3.6.1), it follows that $q_i \in \mathbb{C}$ for all i, and so in particular $p \cong q_0 \in \mathbb{C}$. $\square$

Lemma 3.7 *For each term p there exists a basic term q with $g(q) \leq g(p)$ and $p = q$.*

Proof: The rewrite system R in Table 3 reduces each occurrence of sequential composition in which the left-hand side is not an atomic action. It applies to process terms modulo AC of the + (but not modulo prefixing with ε).

$(x+y)z$	$\rightarrow$	$xz + yz$
$(xy)z$	$\rightarrow$	$x(yz)$
δx	$\rightarrow$	δ
εx	$\rightarrow$	x
$(x^\omega)y$	$\rightarrow$	x^ω

Table 3: The rewrite system R

The rewrite system R is terminating. Namely, consider the following weight function h in the natural numbers:

$$\begin{aligned} h(a) &= 2 \\ h(\xi) &= 2 \\ h(p+q) &= h(p) + h(q) \\ h(pq) &= h(p)^2 h(q) \\ h(p^\omega) &= h(p). \end{aligned}$$

It is easy to see that h-value strictly decreases under application of rewrite rules in R, and that it is invariant modulo AC of the $+$. Since the ordering on the natural numbers is well-founded, it follows that R is terminating.

Remark: For termination of R it is essential that it does not apply to terms modulo A9, for else

$$x \cong \varepsilon x \cong (\varepsilon\varepsilon)x \rightarrow \varepsilon(\varepsilon x).$$

Since each rewrite rules in R can be derived from the axioms, and does not increase g-value, it follows that each term is provably equal to a normal form of R, with non-increased g-value. These normal forms are of the form

$$\sum_{i\in I} a_i p_i + \sum_{j\in J} b_j + \sum_{k\in K} q_k^\omega + \sum_{l\in L} \xi_l,$$

where $I \cup J \cup K \cup L \neq \emptyset$. We finish the proof of this lemma by showing that each normal form of R is provably equal to a basic term with the same g-value, using structural induction. By induction, we can assume that the p_i and q_k are provably equal to basic terms r_i and s_k, respectively, with the same g-value.

On the one hand, if $\xi_l \equiv \varepsilon$ for some $l \in L$, then $\sum_{l\in L} \xi_l \stackrel{\text{A3,6}}{=} \varepsilon$. On the other hand, if $\xi_l \equiv \delta$ for all $l \in L$, then using A6, $\sum_{l\in L} \xi_l$ can be replaced by a single summand δ. Hence, $\sum_{l\in L} \xi_l$ can be replaced by a single summand ξ, which has the same g-value, namely zero.

$\sum_{j\in J} b_j + \xi \stackrel{\text{A8}}{=} \sum_{j\in J} b_j\varepsilon + \xi$, which has the same g-value, namely zero.

If $s_k \downarrow$, then in s_k^ω we can replace the summand ε of s_k by a summand δ, using axiom TC2 together with A6. (This adaptation was described in detail in case 2 of the proof of Lemma 2.7.) Thus, s_k^ω is provably equal to a term t_k^ω with $t_k \in \mathbb{B}$ and $g(t_k) = g(s_k)$ and $t_k \not\downarrow$. According to Lemma 2.7, t_k^ω is provably equal to its expansion, which is of the form

$$\sum_{m\in M_k} c_{mk} u_{mk} + \delta$$

where $u_{mk} \in \mathbb{C}$ for all m. Hence, we can replace each summand q_k^ω in the original normal form by $\sum_{m\in M_k} c_{mk} u_{mk}$ (use A6 to get rid of the extra summand δ), which has the same g-value (Lemma 3.1).

Thus, the original normal form is provably equal to the following basic term with the same g-value:

$$\sum_{i\in I} a_i r_i + \sum_{j\in J} b_j \varepsilon + \sum_{k\in K} (\sum_{m\in M_k} c_{mk} u_{mk}) + \xi. \qquad \Box$$

4 The Main Theorem

This section presents the proof of the completeness theorem for $\mathrm{BPA}_{\delta\varepsilon}^\omega(A)$.

4.1 Two Auxiliary Functions: ψ and ϕ

For notational convenience, we introduce the following abbreviation.

Definition 4.1 *$r \sqsubseteq s$ denotes $r + s \underline{\leftrightarrow} s$.*

Before starting with the completeness proof, first we need to develop some theory; the definitions and results in this section are all new. The proposition that will be proved at the end of this section makes an important stepping stone to obtain the desired completeness result for $\mathrm{BPA}^{\omega}_{\delta\varepsilon}(A)$.

A complicating factor in the completeness proof is that from $p'(p^\omega) \underline{\leftrightarrow} p''(p^\omega)$ for proper derivatives p' and p'' of p it cannot be concluded that $p' \underline{\leftrightarrow} p''$. We give an example.

Example 4.2 *Clearly* $\varepsilon((aa)^\omega) \underline{\leftrightarrow} a((aa)^\omega)$, *but* $\varepsilon \not\underline{\leftrightarrow} a$.

In order to solve this ambiguity, we define an operator ψ_p on basic terms, where intuitively the term $\psi_p(q)$ is obtained from the argument q as follows: if q has a derivative $q_0 + q_1$ with $p^\omega \underline{\leftrightarrow} q_1(p^\omega)$, then in $\psi_p(q)$ the term q_1 is replaced by ε. We will see that if $p'(p^\omega) \underline{\leftrightarrow} p''(p^\omega)$ then $\psi_p(p') \underline{\leftrightarrow} \psi_p(p'')$. Note for instance, in regard of Example 4.2, that $\psi_{aa}(\varepsilon) = \varepsilon$ and $\psi_{aa}(a) = \varepsilon$. We now proceed to present the precise definition of ψ.

Definition 4.3 *Given* $q \in \mathbb{B}$, *the term* $\psi_p(q)$ *is defined as follows, using structural induction. We distinguish two cases: either* $q \in \mathbb{C}$ *or* $q \notin \mathbb{C}$.

- Case 1: $q \in \mathbb{C}$.

  Then put
$$\psi_p(q) \cong q. \tag{1}$$

- Case 2: $q \notin \mathbb{C}$, *so that*
$$q \cong \sum_{i\in I} a_i q_i + \xi.$$

  Once more, we distinguish two cases: either $p^\omega \not\sqsubseteq q(p^\omega)$ *or* $p^\omega \sqsubseteq q(p^\omega)$.

⋆ Case 2.1: $p^\omega \not\sqsubseteq q(p^\omega)$.

  Then put
$$\psi_p(q) \cong \sum_{i\in I} a_i \psi_p(q_i) + \xi. \tag{2}$$

⋆ Case 2.2: $p^\omega \sqsubseteq q(p^\omega)$.

  Then define
$$I_0 = \{i \in I \mid a_i q_i(p^\omega) \not\sqsubseteq p^\omega\}, \tag{3}$$

  and put
$$\psi_p(q) \cong \sum_{i\in I_0} a_i \psi_p(q_i) + \varepsilon. \tag{4}$$

  Since $p^\omega \sqsubseteq q(p^\omega)$, *and since* $I \backslash I_0$ *contains exactly those* i *for which* $a_i q_i(p^\omega)$ *can be matched with a transition of* p^ω, *and since* $\xi(p^\omega) \sqsubseteq p^\omega$, *it follows that in this case we have*
$$p^\omega \underline{\leftrightarrow} (\sum_{i\in I\backslash I_0} a_i q_i + \xi)(p^\omega). \tag{5}$$

Next, we define a second auxiliary operator ϕ_p on basic terms, where intuitively the term $\phi_p(q)$ is obtained from the argument q by replacing each proper derivative q' of q by $\psi_p(q')$.

Definition 4.4 *Given $q \in \mathbb{B}$, the term $\phi_p(q)$ is defined as follows. We distinguish two cases: either $q \in \mathbb{C}$ or $q \notin \mathbb{C}$.*

- Case 1*: $q \in \mathbb{C}$.*

  Then put
  $$\phi_p(q) \cong q. \tag{6}$$

- Case 2*: $q \notin \mathbb{C}$, so that*
  $$q \cong \sum_{i \in I} a_i q_i + \xi.$$
  Then put
  $$\phi_p(q) \cong \sum_{i \in I} a_i \psi_p(q_i) + \xi. \tag{7}$$

We derive several properties for the functions ψ and ϕ.

Lemma 4.5 *For $q \in \mathbb{B}$ we have:*

1. *$g(\psi_p(q)) \leq g(q)$;*
2. *$g(\phi_p(q)) \leq g(q)$.*

The proof of this lemma is straightforward. Namely, Lemma 4.5.1 follows easily by structural induction on q, and Lemma 4.5.2 follows immediately from Lemma 4.5.1.

Lemma 4.6 *For $q \in \mathbb{B}$ we have:*

1. *if $\psi_p(q) \xrightarrow{a} r$, then $r \cong \psi_p(q')$ where $q \xrightarrow{a} q'$;*
2. *if $p^\omega \not\sqsubseteq q(p^\omega)$ and $q \xrightarrow{a} q'$, then $\psi_p(q) \xrightarrow{a} \psi_p(q')$;*
3. *$\{\phi_p(q) \xrightarrow{a} \psi_p(q') \mid q \xrightarrow{a} q'\}$ is the collection of transitions of $\phi_p(q)$.*

Lemma 4.6 follows immediately from the definitions of the functions ψ and ϕ. It uses that we consider terms modulo A9.

Lemma 4.7 *Assume that*

A. for all terms u with $g(u) < N_0$ we have $p^\omega \not\sqsubseteq u$.

Let $q, r \in \mathbb{B}$ and $g(q + r) < N_0$.

1. *If $q(p^\omega) \underline{\leftrightarrow} r(p^\omega)$ then $\psi_p(q) \underline{\leftrightarrow} \psi_p(r)$.*
2. *If $q(p^\omega) \sqsubseteq r(p^\omega)$ and $q \not\downarrow$ and $r \not\downarrow$, then $\phi_p(q) \sqsubseteq \phi_p(r)$.*

Proof of 4.7.1: Let $q(p^\omega) \underline{\leftrightarrow} r(p^\omega)$; we want to prove $\psi_p(q) \underline{\leftrightarrow} \psi_p(r)$. We apply structural induction with respect to the ordering $<$ on pairs of terms (see Definition 3.2). By symmetry it is sufficient to consider the following two cases: either $q, r \in \mathbb{C}$ or $q \notin \mathbb{C}$.

- Case 1: $q, r \in \mathbb{C}$.

  Then both q and r are not normed, so $q(p^\omega) \underline{\leftrightarrow} q$ and $r(p^\omega) \underline{\leftrightarrow} r$. Furthermore, according to equation (1), $\psi_p(q) \cong q$ and $\psi_p(r) \cong r$. So,

$$\psi_p(q) \underline{\leftrightarrow} q \underline{\leftrightarrow} q(p^\omega) \underline{\leftrightarrow} r(p^\omega) \underline{\leftrightarrow} r \underline{\leftrightarrow} \psi_p(r).$$

- Case 2: $q \notin \mathbb{C}$.

  Again we distinguish two cases: either $p^\omega \not\sqsubseteq q(p^\omega)$ or $p^\omega \sqsubseteq q(p^\omega)$.

⋆ Case 2.1: $p^\omega \not\sqsubseteq q(p^\omega)$, and so $p^\omega \not\sqsubseteq r(p^\omega)$.

Then clearly both $q \not\downarrow$ and $r \not\downarrow$. Then it follows from the definition of ψ (equations (1) and (2)) that $\psi_p(q) \not\downarrow$ and $\psi_p(r) \not\downarrow$. So all that is left to do is to match each transition from $\psi_p(q)$ with a transition from $\psi_p(r)$, and vice versa.

Consider a transition from $\psi_p(q)$; according to Lemma 4.6.1 it is of the form $\psi_p(q) \xrightarrow{a} \psi_p(q')$ where $q \xrightarrow{a} q'$. Since $q(p^\omega) \underline{\leftrightarrow} r(p^\omega)$ and $q(p^\omega) \xrightarrow{a} q'(p^\omega)$, this last transition matches with a transition from $r(p^\omega)$, which, since $r \not\downarrow$, is of the form $r(p^\omega) \xrightarrow{a} r'(p^\omega)$, where $r \xrightarrow{a} r'$. Then $q'(p^\omega) \underline{\leftrightarrow} r'(p^\omega)$, and since $p^\omega \not\sqsubseteq r(p^\omega)$, Lemma 4.6.2 implies that $\psi_p(r) \xrightarrow{a} \psi_p(r')$.

Since $q \notin \mathbb{C}$, Lemma 3.6.3 says that q is not a derivative of q'. Therefore $(q', r') < (q, r)$, by item 3 in Definition 3.2. So, since $q'(p^\omega) \underline{\leftrightarrow} r'(p^\omega)$, induction yields $\psi_p(q') \underline{\leftrightarrow} \psi_p(r')$. Hence, the transition $\psi_p(q) \xrightarrow{a} \psi_p(q')$ matches with the transition $\psi_p(r) \xrightarrow{a} \psi_p(r')$.

Similarly we can show that, vice versa, each transition from $\psi_p(r)$ matches with a transition from $\psi_p(q)$. So $\psi_p(q) \underline{\leftrightarrow} \psi_p(r)$.

⋆ Case 2.2: $p^\omega \sqsubseteq q(p^\omega)$, and so $p^\omega \sqsubseteq r(p^\omega)$.

Since by condition A $p \not\sqsubseteq r$, we have $r(p^\omega) \underline{\not\leftrightarrow} r$. This implies that r is normed, so $r \notin \mathbb{C}$. Since $q, r \notin \mathbb{C}$, we have

$$q \cong \sum_{i \in I} a_i q_i + \xi, \qquad r \cong \sum_{j \in J} b_j r_j + \eta.$$

According to equations (3) and (4)

$$\psi_p(q) \cong \sum_{i \in I_0} a_i \psi_p(q_i) + \varepsilon, \qquad \psi_p(r) \cong \sum_{j \in J_0} b_j \psi_p(r_j) + \varepsilon,$$

where

$$I_0 = \{i \in I \mid a_i q_i(p^\omega) \not\sqsubseteq p^\omega\}, \qquad J_0 = \{j \in J \mid b_j r_j(p^\omega) \not\sqsubseteq p^\omega\}.$$

Note that both $\psi_p(q)\downarrow$ and $\psi_p(r)\downarrow$. So all that is left to do is to match each transition from $\psi_p(q)$ with a transition from $\psi_p(r)$, and vice versa.

Consider a transition from $\psi_p(q)$; it is of the form $\psi_p(q) \xrightarrow{a_i} \psi_p(q_i)$ for some $i \in I_0$. Since $q(p^\omega) \leftrightarrow r(p^\omega)$ and $q(p^\omega) \xrightarrow{a_i} q_i(p^\omega)$, this last transition matches with a transition from $r(p^\omega)$. There are two types of transitions from $r(p^\omega)$.

1. If $\eta \equiv \varepsilon$ and $p \xrightarrow{c} p'$, then $r(p^\omega) \xrightarrow{c} p'(p^\omega)$;
2. $r(p^\omega) \xrightarrow{b_j} r_j(p^\omega)$ for $j \in J$.

Suppose that $q(p^\omega) \xrightarrow{a_i} q_i(p^\omega)$ matches with a transition of the first type, $r(p^\omega) \xrightarrow{c} p'(p^\omega)$. Then $a_i q_i(p^\omega) \leftrightarrow cp'(p^\omega) \sqsubseteq p^\omega$, which contradicts the fact that $i \in I_0$.

So apparently, $q(p^\omega) \xrightarrow{a_i} q_i(p^\omega)$ matches with a transition of the second type, $r(p^\omega) \xrightarrow{b_j} r_j(p^\omega)$ for a $j \in J$. Then $a_i \equiv b_j$, and $q_i(p^\omega) \leftrightarrow r_j(p^\omega)$ so induction yields $\psi_p(q_i) \leftrightarrow \psi_p(r_j)$. Since, $b_j r_j(p^\omega) \leftrightarrow a_i q_i(p^\omega) \not\sqsubseteq p^\omega$, by definition $j \in J_0$, and so $\psi_p(r) \xrightarrow{b_j} \psi_p(r_j)$. Hence, the transition $\psi_p(q) \xrightarrow{a_i} \psi_p(q_i)$ matches with the transition $\psi_p(r) \xrightarrow{b_j} \psi_p(r_j)$.

Similarly we can show that, vice versa, each transition from $\psi_p(r)$ matches with a transition from $\psi_p(q)$. So $\psi_p(q) \leftrightarrow \psi_p(r)$. □

Proof of 4.7.2: Let $q(p^\omega) \sqsubseteq r(p^\omega)$ and $q \not\downarrow$ and $r \not\downarrow$; we want to prove $\phi_p(q) \sqsubseteq \phi_p(r)$.

Since $q \not\downarrow$, the definition of ϕ (equations (6) and (7)) implies $\phi_p(q) \not\downarrow$. So all that is left to do is to match each transition from $\phi_p(q)$ with a transition from $\phi_p(r)$.

Consider a transition from $\phi_p(q)$; according to Lemma 4.6.3 it is of the form $\phi_p(q) \xrightarrow{a} \psi_p(q')$, where $q \xrightarrow{a} q'$. Since $q(p^\omega) \sqsubseteq r(p^\omega)$ and $q(p^\omega) \xrightarrow{a} q'(p^\omega)$, this last transition matches with a transition from $r(p^\omega)$, which, since $r \not\downarrow$, is of the form $r(p^\omega) \xrightarrow{a} r'(p^\omega)$, where $r \xrightarrow{a} r'$. Then $q'(p^\omega) \leftrightarrow r'(p^\omega)$, so Lemma 4.7.1 yields $\psi_p(q') \leftrightarrow \psi_p(r')$. Moreover, since $r \xrightarrow{a} r'$, Lemma 4.6.3 implies that $\phi_p(r) \xrightarrow{a} \psi_p(r')$. Hence, the transition $\phi_p(q) \xrightarrow{a} \psi_p(q')$ matches with the transition $\phi_p(r) \xrightarrow{a} \psi_p(r')$.

Since each transition from $\phi_p(q)$ matches with a transition from $\phi_p(r)$, and $\phi_p(q) \not\downarrow$, it follows that $\phi_p(q) \sqsubseteq \phi_p(r)$. □

Lemma 4.8 *Assume that:*

A. for all terms u with $g(u) < N_0$ we have $p^\omega \not\sqsubseteq u$;

B. for all pairs (u,v) of bisimilar terms with $g(u+v) < N_0$ we have $u = v$.

Let $p, q \in \mathbb{B}$ with $p \not\downarrow$ and $g(p+q) < N_0$. Then

$$q(\phi_p(p)^\omega) = \psi_p(q)(\phi_p(p)^\omega).$$

Proof: We apply structural induction on q. We distinguish two cases: either $q \in \mathbb{C}$ or $q \notin \mathbb{C}$.

- CASE 1: $q \in \mathbb{C}$.

  Then the desired equality trivially holds, because according to equation (1) $\psi_p(q) \cong q$.

- CASE 2: $q \notin \mathbb{C}$, so that

$$q \cong \sum_{i \in I} a_i q_i + \xi. \tag{8}$$

  Again we distinguish two cases: either $p^\omega \not\sqsubseteq q(p^\omega)$ or $p^\omega \sqsubseteq q(p^\omega)$.

⋆ CASE 2.1: $p^\omega \not\sqsubseteq q(p^\omega)$.

  Then according to equation (2)

$$\psi_p(q) = \sum_{i \in I} a_i \psi_p(q_i) + \xi, \tag{9}$$

  so that

$$\begin{array}{rcll} \psi_p(q)(\phi_p(p)^\omega) & = & (\sum_{i \in I} a_i \psi_p(q_i) + \xi)(\phi_p(p)^\omega) & \text{(eq. (9))} \\ & = & \sum_{i \in I} a_i(\psi_p(q_i)(\phi_p(p)^\omega)) + \xi(\phi_p(p)^\omega) & \text{(A4, 5)} \\ & = & \sum_{i \in I} a_i(q_i(\phi_p(p)^\omega)) + \xi(\phi_p(p)^\omega) & \text{(induction)} \\ & = & (\sum_{i \in I} a_i q_i + \xi)(\phi_p(p)^\omega) & \text{(A4, 5)} \\ & \cong & q(\phi_p(p)^\omega) & \text{(eq. (8))} \end{array}$$

  .

⋆ CASE 2.2: $p^\omega \sqsubseteq q(p^\omega)$.

  Then according to equations (3) and (4)

$$\psi_p(q) \cong \sum_{i \in I_0} a_i \psi_p(q_i) + \varepsilon \tag{10}$$

  where

$$I_0 = \{i \in I \mid a_i q_i(p^\omega) \not\sqsubseteq p^\omega\}.$$

  For notational convenience, put

$$r \cong \sum_{i \in I \setminus I_0} a_i q_i + \delta. \tag{11}$$

  Equation (7) yields

$$\phi_p(r) \cong \sum_{i \in I \setminus I_0} a_i \psi_p(q_i) + \delta. \tag{12}$$

  First, we derive the equation

$$\phi_p(p)^\omega = (r + \xi)(\phi_p(p)^\omega) \tag{13}$$

  as follows. According to equivalence (5) we have

$$p^\omega \underline{\leftrightarrow} (r + \xi)(p^\omega). \tag{14}$$

  We distinguish two cases: either $\xi \equiv \delta$ or $\xi \equiv \varepsilon$.

- Case 2.2.1: $\xi \equiv \delta$.

  Since $r(p^\omega) \underline{\leftrightarrow} p(p^\omega)$ (equivalence (14)) and $r \not\downarrow$ and $p \not\downarrow$, and since $g(r+p) < N_0$ and requirement A of Lemma 4.7.2 is satisfied, it implies $\phi_p(r) \underline{\leftrightarrow} \phi_p(p)$. Since $g(\phi_p(r) + \phi_p(p)) < N_0$ (Lemma 4.5.2), condition B yields

$$\phi_p(r) = \phi_p(p). \tag{15}$$

  So,

$$\begin{array}{rcll}
(r+\delta)(\phi_p(p)^\omega) & = & (\sum_{i\in I\setminus I_0} a_i q_i + \delta)(\phi_p(p)^\omega) & \text{(eq. (11), A6)} \\
& = & \sum_{i\in I\setminus I_0} a_i(q_i(\phi_p(p)^\omega)) + \delta & \text{(A4, 5, 6)} \\
& = & \sum_{i\in I\setminus I_0} a_i(\psi_p(q_i)(\phi_p(p)^\omega)) + \delta & \text{(induction)} \\
& = & (\sum_{i\in I\setminus I_0} a_i\psi_p(q_i) + \delta)(\phi_p(p)^\omega) & \text{(A4, 5, 6)} \\
& = & \phi_p(r)(\phi_p(p)^\omega) & \text{(eq. (12))} \\
& = & \phi_p(p)(\phi_p(p)^\omega) & \text{(eq. (15))} \\
& = & \phi_p(p)^\omega. & \text{(TC1)}
\end{array}$$

- Case 2.2.2: $\xi \equiv \varepsilon$.

  Since $r(p^\omega) \sqsubseteq p(p^\omega)$ (equivalence (14)), and $r \not\downarrow$ and $p \not\downarrow$, and since $g(r+p) < N_0$ and requirement A of Lemma 4.7.2 is satisfied, it implies $\phi_p(r) \sqsubseteq \phi_p(p)$. Since $g(\phi_p(r) + \phi_p(p)) < N_0$ (Lemma 4.5.2), condition B yields

$$\phi_p(r) + \phi_p(p) = \phi_p(p). \tag{16}$$

  So,

$$\begin{array}{rcll}
(r+\varepsilon)(\phi_p(p)^\omega) & = & (\sum_{i\in I\setminus I_0} a_i q_i + \varepsilon)(\phi_p(p)^\omega) & \text{(eq. (11), A6)} \\
& = & \sum_{i\in I\setminus I_0} a_i(q_i(\phi_p(p)^\omega)) + \phi_p(p)^\omega & \text{(A4, 5, 9)} \\
& = & \sum_{i\in I\setminus I_0} a_i(\psi_p(q_i)(\phi_p(p)^\omega)) + \phi_p(p)^\omega & \text{(induction)} \\
& = & (\sum_{i\in I\setminus I_0} a_i\psi_p(q_i) + \delta)(\phi_p(p)^\omega) + \phi_p(p)^\omega & \text{(A4, 5, 6)} \\
& = & \phi_p(r)(\phi_p(p)^\omega) + \phi_p(p)^\omega & \text{(eq. (12))} \\
& = & (\phi_p(r) + \phi_p(p))(\phi_p(p)^\omega) & \text{(TC1, A4)} \\
& = & \phi_p(p)(\phi_p(p)^\omega) & \text{(eq. (16))} \\
& = & \phi_p(p)^\omega. & \text{(TC1)}
\end{array}$$

  This finishes the proof of equation (13). Now we can derive the desired equality $q(\phi_p(p)^\omega) = \psi_p(q)(\phi_p(p)^\omega)$ as follows.

$$\begin{array}{rcll}
q(\phi_p(p)^\omega) & = & (\sum_{i\in I_0} a_i q_i + r + \xi)(\phi_p(p)^\omega) & \text{(eqs. (8), (11), A6)} \\
& = & \sum_{i\in I_0} a_i(q_i(\phi_p(p)^\omega)) + (r+\xi)\phi_p(p)^\omega & \text{(A4, 5)} \\
& = & \sum_{i\in I_0} a_i(q_i(\phi_p(p)^\omega)) + \phi_p(p)^\omega & \text{(eq. (13))} \\
& = & \sum_{i\in I_0} a_i(\psi_p(q_i)(\phi_p(p)^\omega)) + \phi_p(p)^\omega & \text{(induction)} \\
& = & (\sum_{i\in I_0} a_i\psi_p(q_i) + \varepsilon)(\phi_p(p)^\omega) & \text{(A4, 5, 9)} \\
& = & \psi_p(q)(\phi_p(p)^\omega). & \text{(eq. (10))}
\end{array}$$

This finishes the proof of Lemma 4.8. □

Finally, the functions ψ and ϕ and their properties are used to prove the following proposition, which will be a crucial ingredient of the completeness proof.

Proposition 4.9 *Assume that:*

A. *for all terms u with $g(u) < N_0$ we have $p^\omega \not\sqsubseteq u$;*

B. *for all pairs (u, v) of bisimilar terms with $g(u+v) < N_0$ we have $u = v$.*

Let $g(p+q+r) < N_0$ and $q(p^\omega) \leftrightarrow r(p^\omega)$. Then

$$q(p^\omega) = r(p^\omega).$$

Proof: By Lemma 3.7 $p = s$ with $s \in \mathbb{B}$ and $g(s) \leq g(p) < N_0$. If $s\downarrow$, then in s^ω we can replace the summand ε of s by a summand δ, using axiom TC2 together with A6. (This adaptation was described in detail in case 2 of the proof of Lemma 2.7.) Hence, s^ω is provably equal to a term t^ω with $t \in \mathbb{B}$ and $g(t) = g(s) < N_0$ and $t \not\downarrow$. Thus,

$$p^\omega = t^\omega. \tag{17}$$

Since $t(t^\omega) \leftrightarrow \varepsilon(t^\omega)$, and $g(t) < N_0$, and condition A holds, Lemma 4.7.1 can be applied to derive $\psi_t(t) \leftrightarrow \psi_t(\varepsilon) \leftrightarrow \varepsilon$. Since $g(\psi_t(t)) < N_0$ (Lemma 4.5.1), condition B yields $\psi_t(t) = \varepsilon$. Since conditions A and B hold, Lemma 4.8 can be applied to derive $t(\phi_t(t)^\omega) = \psi_t(t)(\phi_t(t)^\omega) = \varepsilon(\phi_t(t)^\omega) = \phi_t(t)^\omega$. Since $t \not\downarrow$, RSP$^\omega$ then yields

$$t^\omega = \phi_t(t)^\omega. \tag{18}$$

According to Lemma 3.7 there exist basic terms u and v with $g(u) \leq g(q) < N_0$ and $g(v) \leq g(r) < N_0$ and

$$q = u \tag{19}$$

$$r = v. \tag{20}$$

Since $u(t^\omega) \leftrightarrow q(p^\omega) \leftrightarrow r(p^\omega) \leftrightarrow v(t^\omega)$, and since $g(u+v) < N_0$ and requirement A of Lemma 4.7.1 is satisfied, it implies $\psi_t(u) \leftrightarrow \psi_t(v)$. Since $g(\psi_t(u) + \psi_t(v)) < N_0$ (Lemma 4.5.1), condition B yields

$$\psi_t(u) = \psi_t(v). \tag{21}$$

Hence,

$$\begin{aligned} q(p^\omega) &\overset{(17),(19)}{=} u(t^\omega) \overset{(18)}{=} u(\phi_t(t)^\omega) \overset{\text{Lem. 4.8}}{=} \psi_t(u)(\phi_t(t)^\omega) \\ &\overset{(21)}{=} \psi_t(v)(\phi_t(t)^\omega) \overset{\text{Lem. 4.8}}{=} v(\phi_t(t)^\omega) \overset{(18)}{=} v(t^\omega) \overset{(17),(20)}{=} r(p^\omega). \quad \square \end{aligned}$$

4.2 The Completeness Proof

Finally, we are in a position to prove the main theorem.

Theorem 4.10 *The axiomatization* A1-9+TC1,2+RSP$^\omega$ *is complete for* BPA$^\omega_{\delta\varepsilon}(A)$.

Proof: Assume $p, q \in \mathbb{B}$ with $p \leftrightarrow q$; we show that $p = q$, by induction on the well-founded ordering $<$ on pairs of terms. So suppose that we have already dealt with pairs of bisimilar basic terms that are smaller than (p, q). By symmetry it is sufficient to consider the following two cases: either $p \notin \mathbb{C}$ or $p, q \in \mathbb{C}$.

- CASE 1: $p \notin \mathbb{C}$.

  Since $p \leftrightarrow q$, we can adapt the expansions of p and q to the following forms, using axiom A3:

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

  where $p_i \leftrightarrow q_i$ for $i = 1, ..., n$. Since $p \notin \mathbb{C}$, Lemma 3.6.3 says that p is not a derivative of p_i for $i = 1, ..., n$. Since the p_i and the q_i for $i = 1, ..., n$ are derivatives of p and q respectively, it follows that $(p_i, q_i) < (p, q)$ for $i = 1, ..., n$ (by item 3 in Definition 3.2). So induction yields $p_i = q_i$ for $i = 1, ..., n$. Hence, $p = q$.

- CASE 2: $p, q \in \mathbb{C}$, that is,

  $p \cong r'(r^\omega)$, where $r \in \mathbb{B}$ with $r \not\downarrow$, and r' is a proper derivative of r;

  $q = s'(s^\omega)$, where $s \in \mathbb{B}$ with $s \not\downarrow$, and s' is a proper derivative of s.

  By symmetry, it is sufficient to distinguish the following two cases: either r' is not normed, or both r' and s' are normed.

⋆ CASE 2.1: r' is not normed.

Then by Lemma 2.10 $r'(r^\omega) = r'$. Since $g(r') \leq g(r) < g(p)$, item 2 in Definition 3.2 yields $(r', q) < (p, q)$. So, since $r' \leftrightarrow r'(r^\omega) \leftrightarrow q$, induction yields $r' = q$. Hence, $p = r'(r^\omega) = r' = q$.

⋆ CASE 2.2: Both r' and s' are normed.

For convenience of notation put $N_0 = \max\{g(p), g(q)\}$. Again, we consider two cases: either there exists or there does not exist a term t with $g(t) < N_0$ and $p \leftrightarrow t$.

○ CASE 2.2.1: There exists a term t with $g(t) < N_0$ and $p \sqsubseteq t$ (and so $q \sqsubseteq t$).

Since by the assumption at case 2.2 r' is normed, by definition r' has a derivative r'' with $r'' \downarrow$. Since $r''(r^\omega)$ is a derivative of p, and $p \sqsubseteq t$, there exists a derivative t' of t with $r''(r^\omega) \sqsubseteq t'$. Since $r'' \downarrow$ it follows that $r^\omega \sqsubseteq t'$.

Consider the expansion $\sum_{k \in K} c_k t_k + \xi$ of t'. Define $K_0 = \{k \in K \mid c_k t_k \sqsubseteq r^\omega\}$ and put

$$u \cong \sum_{k \in K_0} c_k t_k + \delta.$$

Since $r^\omega \sqsubseteq t'$, and $r^\omega \not\downarrow$, it follows that $r^\omega \leftrightarrow u$, and so $ru \leftrightarrow u$. Furthermore, Lemma 3.1 implies $g(u) \leq g(t) < N_0$, and so $g(ru + u) < N_0$. So after using Lemma 3.7 to reduce ru and u to basic form, we can apply induction, by item 1 in Definition 3.2, to conclude $ru = u$. Since $r \not\downarrow$, RSP$^\omega$ yields $r^\omega = u$. Hence, $p = r'u$. By Lemma 3.7 $r'u = v$ with $v \in \mathbb{B}$ and $g(v) < N_0$. Thus, $p = v$.

Even so, we can derive $q = w$ for some basic term w with $g(w) < N_0$.

Then $v \leftrightarrow p \leftrightarrow q \leftrightarrow w$. Since $g(v + w) < N_0$, induction yields $v = w$. Hence, $p = v = w = q$.

- Case 2.2.2: For each term t, if $g(t) < N_0$ then $p \not\sqsubseteq t$ (and so $q \not\sqsubseteq t$).

  Since $p \underline{\leftrightarrow} q$, the assumption of this case implies $g(p) = g(q)$.

  Note that the requirements A and B for Proposition 4.9 are satisfied, by the assumption at case 2.2.2 together with the induction hypothesis (item 1 of Definition 3.2). So we are allowed to apply Proposition 4.9 in this case.

  We distinguish two cases: either $r^\omega \underline{\leftrightarrow} s^\omega$ or $r^\omega \underline{\not\leftrightarrow} s^\omega$.

- Case 2.2.2.1: $r^\omega \underline{\leftrightarrow} s^\omega$.

  Then $r(s^\omega) \underline{\leftrightarrow} r(r^\omega) \underline{\leftrightarrow} r^\omega \underline{\leftrightarrow} s^\omega \underline{\leftrightarrow} s(s^\omega)$, so since $g(r+s) < N_0$, Proposition 4.9 yields $r(s^\omega) = s(s^\omega) \stackrel{\text{TC1}}{=} s^\omega$. Since $s \not\downarrow$, RSP$^\omega$ yields

$$r^\omega = s^\omega. \tag{22}$$

  Furthermore, $r'(s^\omega) \underline{\leftrightarrow} r'(r^\omega) \underline{\leftrightarrow} s'(s^\omega)$ and $g(r'+s') < N_0$, so Proposition 4.9 yields

$$r'(s^\omega) = s'(s^\omega). \tag{23}$$

  Hence, $p \cong r'(r^\omega) \stackrel{(22)}{=} r'(s^\omega) \stackrel{(23)}{=} s'(s^\omega) \cong q$.

- Case 2.2.2.2: $r^\omega \underline{\not\leftrightarrow} s^\omega$.

  r' is normed, so it has a derivative r'' with $r'' \downarrow$. Since $r''(r^\omega)$ is a derivative of p, and $p \underline{\leftrightarrow} q$, there exists a derivative $s''(s^\omega)$ of q with $r''(r^\omega) \underline{\leftrightarrow} s''(s^\omega)$. As $r'' \downarrow$, and $r''(r^\omega) \underline{\leftrightarrow} s''(s^\omega)$, we have $r^\omega \sqsubseteq s''(s^\omega)$. Since $(s''s) \not\downarrow$, the expansion of $s''s$ is of the form $\sum_{k\in K} c_k t_k + \delta$. Define $K_0 = \{k \in K \mid c_k t_k(s^\omega) \sqsubseteq r^\omega\}$ and put

$$u \cong \sum_{k\in K_0} c_k t_k + \delta.$$

  Since $r^\omega \sqsubseteq s''s(s^\omega)$, it follows that $r^\omega \underline{\leftrightarrow} u(s^\omega)$. Note that $u \not\downarrow$. Furthermore, Lemma 3.1 implies $g(u) \le g(s''s) < N_0$.

  Even so, $s^\omega \underline{\leftrightarrow} v(r^\omega)$ for some term v with $v \not\downarrow$ and $g(v) < N_0$.

  Since $uv(r^\omega) \underline{\leftrightarrow} u(s^\omega) \underline{\leftrightarrow} r^\omega \underline{\leftrightarrow} r(r^\omega)$, and $g(uv+r) < N_0$, Proposition 4.9 yields $uv(r^\omega) = r(r^\omega) \stackrel{\text{TC1}}{=} r^\omega$. Since $(uv) \not\downarrow$, RSP$^\omega$ then yields

$$r^\omega = (uv)^\omega. \tag{24}$$

  Even so,

$$s^\omega = (vu)^\omega. \tag{25}$$

  Since $u((vu)^\omega) \stackrel{\text{TC1}}{=} u((vu)((vu)^\omega)) \stackrel{\text{A5}}{=} (uv)(u((vu)^\omega))$, and $(uv) \not\downarrow$, RSP$^\omega$ yields

$$u((vu)^\omega) = (uv)^\omega. \tag{26}$$

  Furthermore, $r'u(s^\omega) \underline{\leftrightarrow} r'u((vu)^\omega) \underline{\leftrightarrow} r'((uv)^\omega) \underline{\leftrightarrow} r'(r^\omega) \underline{\leftrightarrow} s'(s^\omega)$, so since $g(r'u+s') < N_0$, Proposition 4.9 yields

$$r'u(s^\omega) = s'(s^\omega). \tag{27}$$

So finally,

$$p \cong r'(r^\omega) \overset{(24)}{=} r'((uv)^\omega) \overset{(26)}{=} r'u((vu)^\omega) \overset{(25)}{=} r'u(s^\omega) \overset{(27)}{=} s'(s^\omega) \cong q.$$

This finishes the proof of Theorem 4.10. □

4.3 Two Examples

We give two examples as to how the construction in the completeness proof acts on particular pairs of bisimilar basic terms. The examples are meant to present the spirit of the construction, not its precise workings. Therefore, we will avoid some detours, which are enforced in the completeness proof for the sake of generality, but which would only be confusing in these particular cases. The syntactic form of basic terms, which is convenient for abstract reasoning, becomes clumsy when applied in practice, so we will not reduce terms to basic form in the examples. The examples show that the two main cases in the completeness proof, 2.2.2.1 and 2.2.2.2, are not imaginary.

Example 4.11 $(a+\varepsilon)((a+b(a+\varepsilon))^\omega) \underline{\leftrightarrow} (b+\varepsilon)((a(b+\varepsilon)+b)^\omega)$.

This equivalence belongs with case 2.2.2.1. It can be derived as follows.

$$\begin{aligned}(a+\varepsilon)((a+b)^\omega) &\overset{\text{A4,9}}{=} a((a+b)^\omega)+(a+b)^\omega\\ &\overset{\text{TC1,A4}}{=} (a+a+b)((a+b)^\omega)\\ &\overset{\text{A3,TC1}}{=} (a+b)^\omega.\end{aligned}$$

Thus we have derived

$$(a+\varepsilon)((a+b)^\omega) = (a+b)^\omega. \tag{28}$$

Even so we can derive

$$(b+\varepsilon)((a+b)^\omega) = (a+b)^\omega. \tag{29}$$

Next,

$$\begin{aligned}(a+b(a+\varepsilon))((a+b)^\omega) &\overset{\text{A4,5}}{=} a((a+b)^\omega)+b((a+\varepsilon)((a+b)^\omega))\\ &\overset{(28)}{=} a((a+b)^\omega)+b((a+b)^\omega)\\ &\overset{\text{A4,TC1}}{=} (a+b)^\omega.\end{aligned}$$

Since $(a+b(a+\varepsilon))\not\downarrow$, RSP$^\omega$ yields

$$(a+b)^\omega = (a+b(a+\varepsilon))^\omega. \tag{30}$$

Even so we can derive

$$(a+b)^\omega = (a(b+\varepsilon)+b)^\omega. \tag{31}$$

So finally,

$$\begin{aligned}(a+\varepsilon)((a+b(a+\varepsilon))^\omega) &\overset{(30)}{=} (a+\varepsilon)((a+b)^\omega)\\ &\overset{(28)}{=} (a+b)^\omega\\ &\overset{(29)}{=} (b+\varepsilon)((a+b)^\omega)\\ &\overset{(31)}{=} (b+\varepsilon)((a(b+\varepsilon)+b)^\omega).\end{aligned}$$

Example 4.12 $(a\delta+b)((c(a\delta+b))^\omega) \underline{\leftrightarrow} (a\delta+bc)^\omega$.

This equivalence belongs with case 2.2.2.2. It can be derived as follows.

$$\begin{array}{rcl}(a\delta+b)((c(a\delta+b))^\omega) & \stackrel{\rm TC1}{=} & (a\delta+b)((c(a\delta+b))((c(a\delta+b))^\omega)) \\ & \stackrel{\rm A4,5}{=} & ((a\delta+b)c)((a\delta+b)((c(a\delta+b))^\omega)).\end{array}$$

Since $((a\delta+b)c)\not\downarrow$, RSP$^\omega$ yields

$$(a\delta+b)((c(a\delta+b))^\omega) = ((a\delta+b)c)^\omega. \qquad (32)$$

So finally,

$$\begin{array}{rcl}(a\delta+b)((c(a\delta+b))^\omega) & \stackrel{(32)}{=} & ((a\delta+b)c)^\omega \\ & \stackrel{\rm A4,5,7}{=} & (a\delta+bc)^\omega.\end{array}$$

References

[1] L. Aceto and W.J. Fokkink. An equational axiomatization for multi-exit iteration. Report RS-96-22, BRICS, Aalborg University, 1996. Available via ftp at brics.aau.dk in the directory pub/BRICS/RS/96/22.

[2] L. Aceto, W.J. Fokkink, R.J. van Glabbeek, and A. Ingólfsdóttir. Axiomatizing prefix iteration with silent steps. *Information and Computation*, 127(1):26–40, 1996.

[3] L. Aceto, W.J. Fokkink, and A. Ingólfsdóttir. A menagerie of non-finitely based process semantics over BPA*: from ready simulation to completed traces. Technical report, BRICS, Aalborg University, 1996.

[4] L. Aceto and J.F. Groote. A complete equational axiomatization for MPA with string iteration. Report RS-95-28, BRICS, Aalborg University, 1995. Available via ftp at brics.aau.dk in the directory pub/BRICS/RS/95/28.

[5] 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.

[6] 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 *LNCS*, pages 477–492. Springer, 1993.

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

[8] J.A. Bergstra and P. Klint. The discrete time toolbus. In M. Wirsing and M. Nivat, editors, *Proceedings 5th Conference on Algebraic Methodology and Software Technology (AMAST'96),* Munich, volume 1101 of *LNCS*, pages 286–305. Springer, 1996.

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

[10] 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,* Wendisch-Rietz, volume 215 of *LNCS*, pages 9–23. Springer, 1985.

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

[12] W.J. Fokkink. On the completeness of the equations for the Kleene star in bisimulation. In M. Wirsing and M. Nivat, editors, *Proceedings 5th Conference on Algebraic Methodology and Software Technology (AMAST'96),* Munich, volume 1101 of *LNCS*, pages 180–194. Springer, 1996.

[13] W.J. Fokkink and R.J. van Glabbeek. Ntyft/ntyxt rules reduce to ntree rules. *Information and Computation*, 126(1):1–10, 1996.

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

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

[16] D. Kozen. A completeness theorem for Kleene algebras and the algebra of regular events. *Information and Computation*, 110(2):366–390, 1994.

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

[18] D.M.R. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, *Proceedings 5th GI Conference,* Karlsruhe, volume 104 of *LNCS*, pages 167–183. Springer, 1981.

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

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

[21] P.M. 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.

[22] L. Sterling and E. Shapiro. *The Art of Prolog.* MIT Press, 1986.