Axiomatizing Prefix Iteration with Silent Steps*

Luca Aceto†
BRICS, Aalborg University

Wan Fokkink§
Utrecht University

Rob van Glabbeek‡
Stanford University

Anna Ingólfsdóttir¶
BRICS, Aalborg University

Abstract

Prefix iteration is a variation on the original binary version of the Kleene star operation P^*Q, obtained by restricting the first argument to be an atomic action. The interaction of prefix iteration with silent steps is studied in the setting of Milner's basic CCS. Complete equational axiomatizations are given for four notions of behavioural congruence over basic CCS with prefix iteration, viz. branching congruence, η-congruence, delay congruence and weak congruence. The completeness proofs for η-, delay, and weak congruence are obtained by reduction to the completeness theorem for branching congruence. It is also argued that the use of the completeness result for branching congruence in obtaining the completeness result for weak congruence leads to a considerable simplification with respect to the only direct proof presented in the literature. The preliminaries and the completeness proofs focus on open terms, i.e., terms that may contain process variables. As a byproduct, the ω-completeness of the axiomatizations is obtained as well as their completeness for closed terms.

AMS Subject Classification (1991): 68Q10, 68Q40, 68Q55.
CR Subject Classification (1991): D.3.1, F.1.2, F.3.2.
Keywords and Phrases: Concurrency, process algebra, basic CCS, prefix iteration, branching bisimulation, η-bisimulation, delay bisimulation, weak bisimulation, equational logic, complete axiomatizations.

1 Introduction

The research literature on process theory has recently witnessed a resurgence of interest in the study of Kleene star-like operations (cf., e.g., the papers [8, 18, 15, 13, 34, 12, 16, 3, 2]).

*In *Information and Computation* 127(1), 1996, pp. 26–40.

†Department of Mathematics and Computer Science, Aalborg University, Fr. Bajersvej 7E, 9220 Aalborg Ø, Denmark. On leave from the School of Cognitive and Computing Sciences, University of Sussex, Brighton BN1 9QH, UK. Partially supported by HCM project EXPRESS. Email: `luca@iesd.auc.dk`.

‡Computer Science Department, Stanford University, Stanford, CA 94305-9045, USA. Partially supported by ONR under grant number N00014-92-J-1974. Email: `rvg@cs.stanford.edu`.

§Utrecht University, Department of Philosophy, Heidelberglaan 8, 3584 CS Utrecht, The Netherlands. Email: `fokkink@phil.ruu.nl`.

¶Department of Mathematics and Computer Science, Aalborg University, Fr. Bajersvej 7E, 9220 Aalborg Ø, Denmark. Email: `annai@iesd.auc.dk`.

Some of these studies, notably [8], have investigated the expressive power of variations on standard process description languages in which infinite behaviours are defined by means of Kleene's star operation [28, 11] rather than by means of systems of recursion equations. Some others (see, e.g., [18, 34, 15, 2, 17]) have studied the possibility of giving finite equational axiomatizations of strong bisimulation equivalence [32, 30] over simple process algebras that include variations on Kleene's star operation. De Nicola and his co-workers [13, 12] have instead focused on the study of tree-based models for what they call "nondeterministic Kleene algebras", and on the proof systems these models support to reason about regular expressions and more expressive languages built on top of those.

This paper aims at giving a contribution to the study of complete equational axiomatizations for Kleene star-like operations from the point of view of process theory. Our starting point is the work presented in [15]. In that reference, a *finite*, complete equational axiomatization of strong bisimulation equivalence has been given for $\mathrm{T(BCCS)}^{p*}(A_\tau)$, i.e., the language of closed terms obtained by extending the fragment of Milner's CCS [30] containing the basic operations needed to express finite synchronization trees with prefix iteration. Prefix iteration is a variation on the original binary version of the Kleene star operation P^*Q [28] obtained by restricting the first argument to be an atomic action. Intuitively, at any time the process term a^*P can decide to perform action a and evolve to itself, or an action from P, by which it exits the a-loop. The behaviour of a^*P is captured very clearly by the rules that give its Plotkin-style structural operational semantics:

$$\frac{}{a^*P \stackrel{a}{\to} a^*P} \qquad \frac{P \stackrel{b}{\to} P'}{a^*P \stackrel{b}{\to} P'}$$

In [15] it is shown that, in strong bisimulation semantics, such an operation can be characterized by the standard equations for CCS summation (cf. [30] and Table 1) and the following two natural laws:

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

The reader familiar with Hennessy's work on complete axiomatizations for the delay operation of Milner's SCCS [24, 25] will have noticed the similarity between the above laws and those presented in [24] (see also [1, Page 40]). This is not surprising as such a delay operation is an instance of the prefix iteration construct.

1.1 Results

In this paper, we extend the results in [15] to a setting with the unobservable action τ. More precisely, we consider four versions of bisimulation equivalence that, to different degrees, abstract away from the internal evolution of processes (viz. delay equivalence [29], weak equivalence [30], η-equivalence [5] and branching equivalence [21]), and provide complete equational axiomatizations for each of the congruences they induce over the language $\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$ of *open* terms over the signature of $\mathrm{T(BCCS)}^{p*}(A_\tau)$. The axiomatizations we present are obtained by extending the axiom system from [15] with the relevant τ-laws known from the literature for each of the congruences we consider (cf. [22] for a discussion of these laws), and with collections of laws that describe the interplay between the silent nature of τ and prefix iteration. For instance, the axiomatization of weak congruence uses Milner's well-known τ-laws [30] and the following axioms describing the interaction of prefix iteration with the silent action τ:

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

The first of these equations was introduced in [8] under the name of *Fair Iteration Rule*, and expresses a fundamental property of weak congruence, namely the abstraction from τ-loops, that underlies the soundness of *Koomen's Fair Abstraction Rule* [4]. The other two equations are from [3], and describe a rather subtle interplay between prefix iteration and the silent action τ. All the axiomatizations we present are *finite*, if so is the set of observable actions, and irredundant.

The strategy we adopt in establishing the completeness results is based upon the use of branching equivalence in the analysis of weak, delay and η-equivalence advocated in [20]. Following *op. cit.*, complete axiomatizations for weak, delay and η-congruence can be obtained from one for branching congruence by:

1. identifying a collection of process terms on which branching congruence coincides with the congruence one aims at axiomatizing, and
2. finding an axiom system that allows for the reduction of every process term to one of the required form.

For example, the completeness result for weak congruence is obtained by proving that branching and weak congruence coincide over the collection of w-saturated process terms (cf. Defn. 4.5), and that, using the axiom system for weak congruence, every term is provably equal to a w-saturated one.

The completeness results for η- and delay congruence are new, while those for weak and branching congruence were first proven in [3] and [16], respectively. However, the proofs for these last two results that are presented in this paper are new, and we consider them to be an improvement on the original ones. In particular, unlike the one given in [16], the proof for branching congruence does not rely on the completeness result for strong bisimulation presented in [15]. Perhaps surprisingly, the proof for weak congruence presented here is simpler than the one given in [3] which only uses properties of weak congruence. The direct proof method employed in *op. cit.* yields a long proof with many case distinctions, while the indirect proof via branching congruence, which we present here, is considerably shorter, and relies on a general relationship between the two congruences. All the authors' attempts to obtain a direct proof of the completeness theorem for weak congruence which is simpler than the one presented in [3] have been to no avail. It should be noted, however, that delicate case analyses appear to be inescapable components of completeness proofs for equational axiomatizations of behavioural congruences over variations on Kleene algebras (cf., e.g., the proofs in [18, 15, 3, 16, 2, 17]), and they are present in our completeness proofs just as well.

Another notable feature of the proofs of the completeness theorems we offer is that, unlike those in [3, 16], they apply to open terms directly, and thus yield the ω-completeness of the axiomatizations as well as their completeness for closed terms. Following [31, 19], this is achieved by defining a structural operational semantics and notions of bisimulation directly on open terms. For all the notions of bisimulation equivalence so defined for open terms in the language $\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$, we prove that two terms are equivalent iff all their closed instantiations are. This ensures that our definitions are in agreement with the standard ones in the literature on process theory.

The ω-completeness of the axiomatizations for branching, η- and delay congruence are all new. The axiomatization for weak congruence was first shown to be ω-complete in [3] in the presence of a countably infinite set of observable actions, using a technique from Groote [23]. Our result in this paper sharpens the one in the aforementioned reference in that, like the ones for branching, η- and delay congruence, it only requires that the set of observable actions be non-empty.

1.2 Outline of the paper

The paper is organized as follows. Section 2 introduces the language of basic CCS with prefix iteration, $\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$, and its operational semantics. In that section we also give the definition of branching, η-, delay and weak congruence over open terms, and show that two open terms are related by any of those congruences iff all their closed instantiations are. Section 2 concludes with a study of several properties of the congruence relations we consider that will be used in the remainder of the paper. The axiom systems that will be shown to completely characterize the aforementioned congruences over $\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$ are analyzed in Section 3. Detailed proofs of the completeness of our axiom systems with respect to the relevant congruences over $\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$ are presented in Section 4.

2 Basic CCS with Prefix Iteration

We assume a non-empty, countable set A of observable actions not containing the distinguished symbol τ. Following Milner [30], the symbol τ will be used to denote an internal, unobservable action of a system. We define $A_\tau \triangleq A \cup \{\tau\}$, and use a, b to range over A and α, β, γ to range over A_τ. We also assume a countably infinite set of process variables Var, ranged over by x, y, z, that is disjoint from A_τ. The meta-variable ξ will stand for a typical member of the set $A_\tau \cup \mathsf{Var}$.

The language of basic CCS with prefix iteration, denoted by $\mathrm{BCCS}^{p*}(A_\tau)$, is given by the following BNF grammar:

$$P ::= x \mid 0 \mid \alpha.P \mid P + P \mid \alpha^* P$$

where $x \in \mathsf{Var}$ and $\alpha \in A_\tau$. The set of (open) terms over $\mathrm{BCCS}^{p*}(A_\tau)$ is denoted by $\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$, and the set of closed terms, i.e., terms that do not contain occurrences of process variables, by $\mathrm{T}(\mathrm{BCCS})^{p*}(A_\tau)$. We shall use P, Q, R, S, T to range over

$\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$. In writing terms over the above syntax, we shall always assume that the operations α^* and $\alpha.\_$ bind stronger than $+$. We shall use the symbol $\equiv$ to stand for syntactic equality of terms. The set of process variables occurring in a term P will be written $\mathsf{Var}(P)$.

A (closed) substitution is a mapping from process variables to (closed) terms over $\mathrm{BCCS}^{p*}(A_\tau)$. For every term P and (closed) substitution σ, the (closed) term obtained by replacing every occurrence of a variable x in P with the (closed) term $\sigma(x)$ will be written $P\sigma$. We shall use $[x \mapsto P]$ to stand for the substitution mapping x to P, and acting like the identity on all the other variables.

The operational semantics for the language $\mathrm{BCCS}^{p*}(A_\tau)$ is given by the labelled transition system [27, 33]

$$\left(\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau), \left\{\stackrel{\xi}{\rightarrow} \mid \xi \in A_\tau \cup \mathsf{Var}\right\}\right)$$

where the transition relations $\stackrel{\xi}{\rightarrow}$ are the least subsets of $\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau) \times \mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$ satisfying the rules in Fig. 1. Intuitively, a transition $P \stackrel{\alpha}{\rightarrow} Q$ ($\alpha \in A_\tau$) means that the system represented by the term P can perform the action α, thereby evolving into Q, whereas $P \stackrel{x}{\rightarrow} P'$ means that the initial behaviour of P may depend on the term that is substituted for the process variable x. It is not hard to see that if $P \stackrel{x}{\rightarrow} P'$ then $P' \equiv x$.

$$\frac{}{x \stackrel{x}{\rightarrow} x} \qquad \frac{}{\alpha.P \stackrel{\alpha}{\rightarrow} P} \qquad \frac{P \stackrel{\xi}{\rightarrow} P'}{P+Q \stackrel{\xi}{\rightarrow} P'} \qquad \frac{Q \stackrel{\xi}{\rightarrow} Q'}{P+Q \stackrel{\xi}{\rightarrow} Q'}$$

$$\frac{}{\alpha^*P \stackrel{\alpha}{\rightarrow} \alpha^*P} \qquad \frac{P \stackrel{\xi}{\rightarrow} P'}{\alpha^*P \stackrel{\xi}{\rightarrow} P'}$$

Figure 1: Transition rules

The derived transition relations $\stackrel{\varepsilon}{\Rightarrow}$ and $\stackrel{\xi}{\Rightarrow}$ ($\xi \in A_\tau \cup \mathsf{Var}$) are defined in the standard way as follows: $\begin{cases} \stackrel{\varepsilon}{\Rightarrow} \text{ is the reflexive, transitive closure of } \stackrel{\tau}{\rightarrow}, \\ P \stackrel{\xi}{\Rightarrow} Q \text{ iff } \exists P_1, P_2 : \ P \stackrel{\varepsilon}{\Rightarrow} P_1 \stackrel{\xi}{\rightarrow} P_2 \stackrel{\varepsilon}{\Rightarrow} Q \ . \end{cases}$

Definition 2.1 *The set* $\mathsf{der}(P)$ *of* derivatives *of P is the least set containing P that is closed under action-transitions. Formally,* $\mathsf{der}(P)$ *is the least set satisfying:*

1. *$P \in \mathsf{der}(P)$;*
2. *if $Q \in \mathsf{der}(P)$ and $Q \stackrel{\alpha}{\rightarrow} Q'$ for some $\alpha \in A_\tau$, then $Q' \in \mathsf{der}(P)$.*

The following basic fact can be easily shown by structural induction on terms:

Fact 2.2 *For every $P \in \mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$, the set of derivatives of P is finite.*

A fundamental semantic equivalence in the study of reactive systems is *bisimulation equivalence* [32, 30]. In this study, we shall consider four versions of this notion which, to different degrees, abstract away from invisible actions, viz. branching equivalence [21], η-equivalence [5], delay equivalence [29] and weak equivalence [30]. These we now proceed to define for the sake of completeness. The interested reader is referred to the aforementioned references and to [22, 31, 19] for discussion and motivation.

Definition 2.3 (Branching Equivalence) *A binary relation $\mathcal{B}$ over $\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$ is a* branching bisimulation*, or* b-bisimulation *for short, iff it is symmetric and, whenever $P \mathrel{\mathcal{B}} Q$, for all $\xi \in A_\tau \cup \mathsf{Var}$,*

if $P \xrightarrow{\xi} P'$ then

- *$\xi = \tau$ and $P' \mathrel{\mathcal{B}} Q$, or*
- *$Q \stackrel{\varepsilon}{\Rightarrow} Q_1 \xrightarrow{\xi} Q_2 \stackrel{\varepsilon}{\Rightarrow} Q'$ for some Q_1, Q_2, Q' such that $P \mathrel{\mathcal{B}} Q_1$, $P' \mathrel{\mathcal{B}} Q_2$ and $P' \mathrel{\mathcal{B}} Q'$.*

Two process terms P, Q are branching equivalent*, denoted by $P \leftrightarrows_b Q$, iff there exists a branching bisimulation $\mathcal{B}$ such that $P \mathrel{\mathcal{B}} Q$.*

The notions of η-, delay, and weak bisimulation are obtained by relaxing (some of) the constraints imposed by branching bisimulation on the way that two processes can match each other's behaviours. Compare the following definitions:

Definition 2.4 (η-, Delay and Weak Equivalence) *The notion of* η-bisimulation *is defined just as a branching bisimulation above, but without the requirement $P' \mathrel{\mathcal{B}} Q_2$. Two process terms P, Q are* η-equivalent*, denoted by $P \leftrightarrows_\eta Q$, iff there exists an η-bisimulation $\mathcal{B}$ such that $P \mathrel{\mathcal{B}} Q$.*

Likewise, a delay bisimulation*, or* d-bisimulation *for short, is defined just as a branching bisimulation, but omitting the requirement $P \mathrel{\mathcal{B}} Q_1$. Two process terms P, Q are* delay equivalent*, denoted by $P \leftrightarrows_d Q$, iff there exists a delay bisimulation $\mathcal{B}$ such that $P \mathrel{\mathcal{B}} Q$.*

Finally, a weak bisimulation*, or* w-bisimulation*, lacks both the requirements $P \mathrel{\mathcal{B}} Q_1$ and $P' \mathrel{\mathcal{B}} Q_2$, and two process terms P, Q are* weakly equivalent*, denoted by $P \leftrightarrows_w Q$, iff there exists a weak bisimulation $\mathcal{B}$ such that $P \mathrel{\mathcal{B}} Q$.*

Remark: It is easy to see that in the definitions of both branching and delay bisimulation the existence requirement of a term Q' such that $Q_2 \stackrel{\varepsilon}{\Rightarrow} Q'$ and $P' \mathrel{\mathcal{B}} Q'$ is redundant.

The notions of delay and weak equivalence were originally both introduced by Milner under the name of *observation(al) equivalence*.

Proposition 2.5 *Each of the relations $\leftrightarrows_\aleph$ $(\aleph \in \{b, \eta, d, w\})$ is an equivalence relation and the largest $\aleph$-bisimulation. Furthermore, for all P, Q,*

1. *If $P \leftrightarrows_b Q$, then $P \leftrightarrows_\eta Q$ and $P \leftrightarrows_d Q$;*
2. *If $P \leftrightarrows_\eta Q$ or $P \leftrightarrows_d Q$, then $P \leftrightarrows_w Q$.*

Proof: For $\aleph \in \{\eta, d, w\}$, the identity relation, the converse of a $\aleph$-bisimulation and the symmetric closure of the composition of two $\aleph$-bisimulations are all $\aleph$-bisimulations. Hence $\leftrightarrow_\aleph$ is an equivalence relation. This argument does not apply for $\aleph = b$ because the symmetric closure of the composition of two b-bisimulations need not be a b-bisimulation, but in [7] it is shown that also $\leftrightarrow_b$ is an equivalence relation.

That $\leftrightarrow_\aleph$ is the largest $\aleph$-bisimulation (for $\aleph \in \{b, \eta, d, w\}$) follows immediately from the observation that the set of $\aleph$-bisimulations is closed under arbitrary unions. The implications hold by definition. □

The reader familiar with the literature on process theory might have noticed that, in the above definitions, we have departed from the standard approach followed in, e.g., [30] in that we have defined notions of bisimulation equivalence that apply to open terms directly. Indeed, with the exception of studies like [31, 19], bisimulation equivalences like those presented in Defs. 2.3–2.4 are usually defined for closed process expressions only, and are extended to open process expression thus ($\aleph \in \{b, \eta, d, w\}$):

$$P \leftrightarrow_\aleph Q \text{ iff } P\sigma \leftrightarrow_\aleph Q\sigma \text{ for every closed substitution } \sigma.$$

By the following result, first shown in [19] for branching bisimulation over basic CCS with recursion, both approaches yield the same equivalence relation over open terms in the language $\mathrm{BCCS}^{p*}(A_\tau)$.

Proposition 2.6 *For all $P, Q \in \mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$ and $\aleph \in \{b, \eta, d, w\}$,*

$$P \leftrightarrow_\aleph Q \text{ iff } P\sigma \leftrightarrow_\aleph Q\sigma \text{ for every closed substitution } \sigma : \mathsf{Var} \to \mathrm{T}(\mathrm{BCCS})^{p*}(A_\tau).$$

Proof: In the proof of this result, we shall make use of the following, easily established, facts, which relate the transitions of a term $P\sigma$ to those of P and those of the terms $\sigma(x)$:

1. If $P \xrightarrow{\alpha} P'$, then $P\sigma \xrightarrow{\alpha} P'\sigma$.
2. If $P \xrightarrow{x} x$ and $\sigma(x) \xrightarrow{\xi} Q$, then $P\sigma \xrightarrow{\xi} Q$.
3. If $P\sigma \xrightarrow{\xi} Q$, then either
   (a) $\xi \in A_\tau$ and there exists a P' such that $P \xrightarrow{\xi} P'$ and $Q \equiv P'\sigma$, or
   (b) there exists an $x \in \mathsf{Var}$ such that $P \xrightarrow{x} x$ and $\sigma(x) \xrightarrow{\xi} Q$.

We now prove the two implications in the statement of the proposition separately.

- 'Only If Implication'. Assume that $P \leftrightarrow_\aleph Q$ ($\aleph \in \{b, \eta, d, w\}$). We shall show that $P\sigma \leftrightarrow_\aleph Q\sigma$ for every closed substitution $\sigma : \mathsf{Var} \to \mathrm{T}(\mathrm{BCCS})^{p*}(A_\tau)$. To this end, it is sufficient to prove that the relation:
  $$\mathcal{B}_\aleph \triangleq \{(S\sigma, T\sigma) \mid S \leftrightarrow_\aleph T,\ \sigma \text{ a closed substitution}\}$$
  is a $\aleph$-bisimulation. This is straightforward using facts 1–3 above.
- 'If Implication'. Let $\aleph \in \{b, \eta, d, w\}$. Assume that $P\sigma \leftrightarrow_\aleph Q\sigma$ for every closed substitution σ. We shall show that $P \leftrightarrow_\aleph Q$ holds. This we prove by induction on the number of variables occurring in P or Q, i.e., on the cardinality of $\mathsf{Var}(P) \cup \mathsf{Var}(Q)$.

– Basis: $\mathsf{Var}(P) \cup \mathsf{Var}(Q) = \emptyset$. In this case, P and Q are closed terms, and the claim follows immediately.
– Inductive Step: $\mathsf{Var}(P) \cup \mathsf{Var}(Q) \neq \emptyset$. Choose a variable x in $\mathsf{Var}(P) \cup \mathsf{Var}(Q)$. As the set of observable actions A is non-empty, we can pick $a \in A$. It is easy to see that, for positive integers n, m,

$$a^n.0 \leftrightarrows_{\aleph} a^m.0 \Leftrightarrow n = m \ .$$

By Fact 2.2, $\mathsf{der}(P) \cup \mathsf{der}(Q)$ is a finite set of process terms. Therefore it is possible to choose a positive integer n such that, for every $R \in \mathsf{der}(P) \cup \mathsf{der}(Q)$,

$$a^n.0 \quad \not\leftrightarrows_{\aleph} \quad R \ . \tag{1}$$

Note that the above inequality implies that, for every $R \in \mathsf{der}(P) \cup \mathsf{der}(Q)$,

$$a^n.0 \quad \not\leftrightarrows_{\aleph} \quad R[x \mapsto a^{n+1}.0] \ . \tag{2}$$

This is immediate by (1) if x does not occur in R. Otherwise, x occurs in R, and it is not hard to see that $R[x \mapsto a^{n+1}.0]$ can perform a sequence of transitions leading to 0 that has a suffix consisting of at least $n + 1$ a-transitions, whereas $a^n.0$ cannot.

Now, note that, for every closed substitution σ,

$$\big(P[x \mapsto a^{n+1}.0]\big)\sigma \quad \leftrightarrows_{\aleph} \quad \big(Q[x \mapsto a^{n+1}.0]\big)\sigma \ . \tag{3}$$

As the set of variables occurring in $P[x \mapsto a^{n+1}.0]$ or $Q[x \mapsto a^{n+1}.0]$ is strictly contained in $\mathsf{Var}(P) \cup \mathsf{Var}(Q)$, we may apply the inductive hypothesis to (3) to infer that:

$$P[x \mapsto a^{n+1}.0] \quad \leftrightarrows_{\aleph} \quad Q[x \mapsto a^{n+1}.0] \ . \tag{4}$$

We prove that this implies $P \leftrightarrows_{\aleph} Q$, as required. To this end, in view of (4), it is sufficient to show that the symmetric closure of the relation

$$\mathcal{B}_{\aleph} \quad \triangleq \quad \big\{(S,T) \mid (S,T) \in \mathsf{der}(P) \times \mathsf{der}(Q) \text{ and } S[x \mapsto a^{n+1}.0] \leftrightarrows_{\aleph} T[x \mapsto a^{n+1}.0]\big\}$$

is a $\aleph$-bisimulation. The details of this verification are straightforward, using facts 1–3 above and (2). In particular, condition (2) ensures that whenever $S \ \mathcal{B}_{\aleph} \ T$ and $S \xrightarrow{x} x$, then $T \stackrel{x}{\Rightarrow} x$.

This completes the proof of the inductive step, and thereby of the 'if' implication.

The proof of the proposition is now complete. □

Remark: The reader may have noticed that the 'if' implication in the above statement would *not* hold if the set of observable actions A were empty. In fact, in that, admittedly uninteresting, case, the universal relation over $\mathrm{T(BCCS)}^{p*}(A_\tau)$ would be a branching bisimulation. This would imply, for instance, that, for every closed substitution σ and variables x, y,

$$x\sigma \leftrightarrows_b y\sigma \ .$$

On the other hand, x is not branching equivalent to y.

For the standard reasons explained at length in, e.g., Milner's textbook [30], none of the aforementioned equivalences is a congruence with respect to the summation operation.

In fact, it is also the case that none of the aforementioned equivalences is preserved by the prefix iteration operation. As a simple example of this phenomenon, consider the terms $b.0$ and $\tau.b.0$. As it is well-known, $b.0 \leftrightarroweq_{\aleph} \tau.b.0$ ($\aleph \in \{b, \eta, d, w\}$); however, it is not difficult to check that $a^*(b.0) \not\leftrightarroweq_{\aleph} a^*(\tau.b.0)$. Following Milner [30], the solution to these congruence problems is by now standard; it is sufficient to consider, for each equivalence $\leftrightarroweq_{\aleph}$, the largest congruence over $\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$ contained in it. We now proceed to characterize the resulting congruences explicitly.

Definition 2.7 *We say that:*

- P *and* Q *are* branching congruent, *written* $P \leftrightarroweq_b^c Q$, *iff for all* $\xi \in A_\tau \cup \mathsf{Var}$,
  1. *if* $P \xrightarrow{\xi} P'$, *then* $Q \xrightarrow{\xi} Q'$ *for some* Q' *such that* $P' \leftrightarroweq_b Q'$;
  2. *if* $Q \xrightarrow{\xi} Q'$, *then* $P \xrightarrow{\xi} P'$ *for some* P' *such that* $P' \leftrightarroweq_b Q'$.
- P *and* Q *are* η-congruent, *written* $P \leftrightarroweq_\eta^c Q$, *iff for all* $\xi \in A_\tau \cup \mathsf{Var}$,
  1. *if* $P \xrightarrow{\xi} P'$, *then* $Q \xrightarrow{\xi} Q_1 \stackrel{\varepsilon}{\Rightarrow} Q'$ *for some* Q_1, Q' *such that* $P' \leftrightarroweq_\eta Q'$;
  2. *if* $Q \xrightarrow{\xi} Q'$, *then* $P \xrightarrow{\xi} P_1 \stackrel{\varepsilon}{\Rightarrow} P'$ *for some* P_1, P' *such that* $P' \leftrightarroweq_\eta Q'$.
- P *and* Q *are* delay congruent, *written* $P \leftrightarroweq_d^c Q$, *iff for all* $\xi \in A_\tau \cup \mathsf{Var}$,
  1. *if* $P \xrightarrow{\xi} P'$, *then* $Q \stackrel{\varepsilon}{\Rightarrow} Q_1 \xrightarrow{\xi} Q'$ *for some* Q_1, Q' *such that* $P' \leftrightarroweq_d Q'$;
  2. *if* $Q \xrightarrow{\xi} Q'$, *then* $P \stackrel{\varepsilon}{\Rightarrow} P_1 \xrightarrow{\xi} P'$ *for some* P_1, P' *such that* $P' \leftrightarroweq_d Q'$.
- P *and* Q *are* weakly congruent, *written* $P \leftrightarroweq_w^c Q$, *iff for all* $\xi \in A_\tau \cup \mathsf{Var}$,
  1. *if* $P \xrightarrow{\xi} P'$, *then* $Q \stackrel{\xi}{\Rightarrow} Q'$ *for some* Q' *such that* $P' \leftrightarroweq_w Q'$;
  2. *if* $Q \xrightarrow{\xi} Q'$, *then* $P \stackrel{\xi}{\Rightarrow} P'$ *for some* P' *such that* $P' \leftrightarroweq_w Q'$.

Proposition 2.8 *For every* $\aleph \in \{b, \eta, d, w\}$, *the relation* $\leftrightarroweq_{\aleph}^c$ *is the largest congruence over* $\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$ *contained in* $\leftrightarroweq_{\aleph}$.

Proof: It is straightforward to check that $\leftrightarroweq_{\aleph}^c$ is an equivalence relation for $\aleph \in \{b, \eta, d, w\}$, using that this is the case for $\leftrightarroweq_{\aleph}$. Moreover, it is trivial to see that $\leftrightarroweq_{\aleph}^c$ is included in $\leftrightarroweq_{\aleph}$.

That $\leftrightarroweq_{\aleph}^c$ is a congruence relation over $\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$ follows easily from Definition 2.7, using that the relation

$$\{(\alpha^* P, \alpha^* Q) \mid \alpha \in A_\tau, P \leftrightarroweq_{\aleph}^c Q\} \cup \leftrightarroweq_{\aleph}$$

is a $\aleph$-bisimulation. Here it is essential that, unlike $\leftrightarroweq_{\aleph}$, the relations $\leftrightarroweq_{\aleph}^c$ require that an initial τ-transition in a process can*not* be matched by the other staying idle.

To see that $\leftrightarroweq_{\aleph}^c$ is indeed the largest congruence relation over $\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$ contained in $\leftrightarroweq_{\aleph}$, assume that $=_{\aleph}$ is another relation with these properties and that $P =_{\aleph} Q$. We show that $P \leftrightarroweq_{\aleph}^c Q$ holds.

As A is non-empty, we can pick an action $a \in A$. By Fact 2.2, $\mathsf{der}(P) \cup \mathsf{der}(Q)$ is a finite set of process terms. Therefore it is possible to choose a positive integer n such that, for every $R \in \mathsf{der}(P) \cup \mathsf{der}(Q)$,

$$a^n.0 \quad \not\leftrightarroweq_{\aleph} \quad R\ .$$

As $P =_{\aleph} Q$ and $=_{\aleph}$ is a congruence relation contained in $\leftrightarroweq_{\aleph}$, it follows that $P + a^{n+1}.0 \leftrightarroweq_{\aleph} Q + a^{n+1}.0$. For every $\aleph \in \{b, \eta, d, w\}$, this implies that $P \leftrightarroweq^c_{\aleph} Q$. Consider, for instance, the case $\aleph = b$. Let $P \xrightarrow{\xi} P'$ for some $\xi \in A_\tau \cup \mathsf{Var}$. As $P' \not\leftrightarroweq_b Q + a^{n+1}.0$, it must be that $Q + a^{n+1}.0 \overset{\varepsilon}{\Rightarrow} Q_1 \xrightarrow{\xi} Q'$ with $P + a^{n+1}.0 \leftrightarroweq_b Q_1$ and $P' \leftrightarroweq_b Q'$. Moreover, as $P + a^{n+1}.0$ cannot be branching equivalent to a derivative of Q, it follows that $Q_1 \equiv Q + a^{n+1}.0$. Finally $P' \not\leftrightarroweq_b a^n.0$, so $Q \xrightarrow{\xi} Q'$, even when $\xi = a$. By symmetry, it follows that $P \leftrightarroweq^c_b Q$, which was to be shown. □

Remark: Again, note that, if the set of observable actions A were empty, then the relations $\leftrightarroweq^c_{\aleph}$ ($\aleph \in \{b, \eta, d, w\}$) would *not* be the largest congruences contained in $\leftrightarroweq_{\aleph}$ over $\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$. In fact, in that case, $\leftrightarroweq_{\aleph}$ itself would be a congruence, and it is easy to see that, e.g., $\tau.0 \leftrightarroweq_{\aleph} 0$, but $\tau.0 \not\leftrightarroweq^c_{\aleph} 0$.

Remark: Bloom [10] has formulated the 'RWB cool' and 'RBB cool' formats for transition rules, which ensure that the relations $\leftrightarroweq^c_w$ and $\leftrightarroweq^c_b$, respectively, are congruences.

Although both $\leftrightarroweq^c_w$ and $\leftrightarroweq^c_b$ are congruences for $\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$, the transition rules for $\mathrm{BCCS}^{p*}(A_\tau)$ do *not* fit the RWB and RBB cool formats. In particular, Bloom's formats require that operators for which weak or branching equivalence is not a congruence are not to occur in the right-hand sides of conclusions of transition rules. However, we already remarked that weak and branching equivalence are not congruences for prefix iteration, but this operator does occur at the right-hand side of the transition rule $a^*P \xrightarrow{a} a^*P$.

Hence, we obtain a positive answer to the fourth open question at the end of [10], namely whether there exist transition rules outside the RWB and RBB cool formats which define 'interesting' operators for which $\leftrightarroweq^c_w$ and $\leftrightarroweq^c_b$ are congruences.

The following result is the counter-part of Propn. 2.6 for the aforementioned congruence relations.

Proposition 2.9 *For $P, Q \in \mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$ and $\aleph \in \{b, \eta, d, w\}$,*

$$P \leftrightarroweq^c_{\aleph} Q \text{ iff } P\sigma \leftrightarroweq^c_{\aleph} Q\sigma \text{ for every closed substitution } \sigma.$$

Proof: A straightforward modification of the proof of Propn. 2.6. □

We end this section with two lemmas that will be of use in the completeness proof for branching congruence. (Cf. the proof of Propn. 4.3.) The first of these lemmas is a standard result for branching bisimulation equivalence, whose proof may be found in [22, 14].

Lemma 2.10 (Stuttering Lemma) *If $P_0 \xrightarrow{\tau} \cdots \xrightarrow{\tau} P_n$ and $P_n \leftrightarroweq_b P_0$, then $P_i \leftrightarroweq_b P_0$ for $i = 1, ..., n-1$.*

The following result about the expressiveness of the language $\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$ stems from [3].

Lemma 2.11

1. *If $P_n \overset{a_n}{\Rightarrow} P_{n+1}$ for $n = 0, 1, 2, ...$, then there is an N such that $a_n = a_N$ for $n > N$.*
2. *Let $a, b \in A$. If $a^*P \leftrightarroweq_{\aleph} b^*Q$ ($\aleph \in \{b, \eta, d, w\}$), then $a = b$.*

Proof: The proof of of the first statement is an easy exercise by structural induction on terms, which is left to the reader. To show statement 2, note that, in light of Propn. 2.5, it is sufficient to deal with the case $\aleph = w$. Assume, towards a contradiction, that $a^*P \leftrightarroweq_w b^*Q$ and $a \neq b$. Then there exist terms P', Q' such that:

- $a^*P \stackrel{b}{\Rightarrow} P' \leftrightarroweq_w b^*Q$, and
- $b^*Q \stackrel{a}{\Rightarrow} Q' \leftrightarroweq_w a^*P$.

This implies that a^*P and b^*Q both exhibit, for example, an infinite sequence where a and b alternate, i.e., $\stackrel{a}{\Rightarrow}\stackrel{b}{\Rightarrow}\stackrel{a}{\Rightarrow}\stackrel{b}{\Rightarrow}\cdots$. This would contradict statement 1 of the lemma. $\square$

3 Axiom Systems

The main aim of this study is to provide complete equational axiomatizations for branching, η-, delay, and weak congruence over the language $\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$. In this section, we present the axiom systems that will be shown to completely characterize these congruence relations over $\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$, and prove their soundness. We also present a proposition on the inter-derivability of these axiom systems that will be useful in the proofs of the promised completeness theorems, and address the issue of the irredundancy of the axiom systems.

3.1 The axioms

Table 1 presents the axiom system $\mathcal{F}$, which was shown in [15] to characterize strong bisimulation over $\mathrm{T}(\mathrm{BCCS})^{p*}(A)$. In addition to the axioms in $\mathcal{F}$, the axiom systems $\mathcal{E}_\aleph$ ($\aleph \in \{b, \eta, d, w\}$) include equations which express the unobservable nature of the τ action. These equations may be found in Tables 2–5; they reflect the different ways in which the congruences we consider abstract away from internal computations in process behaviours. The axiom system $\mathcal{E}_b$ is obtained by adding the axioms presented in Table 2 to $\mathcal{F}$, and $\mathcal{E}_\eta$ extends $\mathcal{E}_b$ with the equations in Table 4. The set of axioms $\mathcal{E}_d$ includes the equations in $\mathcal{F}$ and those in Table 3. Finally, $\mathcal{E}_w$ extends $\mathcal{E}_d$ with the laws in Table 5.

A1	$x + y$	$=$	$y + x$
A2	$(x + y) + z$	$=$	$x + (y + z)$
A3	$x + x$	$=$	x
A4	$x + 0$	$=$	x
PA1	$a.(a^*x) + x$	$=$	a^*x
PA2	$a^*(a^*x)$	$=$	a^*x

Table 1: The axiom system $\mathcal{F}$

B1	$\alpha.(\tau.(x+y)+x)$	$=$	$\alpha.(x+y)$
PB1	τ^*x	$=$	$\tau.x+x$
PB2	$\tau.a^*(\tau.a^*(x+y)+x)$	$=$	$\tau.a^*(x+y)$

Table 2: Axioms for $\mathcal{E}_b$ and for $\mathcal{E}_\eta$

T1	$\alpha.\tau.x$	$=$	$\alpha.x$
T2	$\tau.x$	$=$	$\tau.x+x$
PT1	τ^*x	$=$	$\tau.x$
PT2	$\tau.(a^*x)$	$=$	$a^*(\tau.(a^*x))$

Table 3: Axioms for $\mathcal{E}_d$ and for $\mathcal{E}_w$

The law B1 and the equations T1-3, AT3 are standard characterizations of the silent action τ in branching and weak congruence, respectively. (Note that AT3 is the instance of T3 with $\alpha \in A$. We distinguish the laws T3 and AT3 in order to obtain an irredundant axiom system for weak congruence. Cf. Propn. 3.4 and the subsequent remark for more details.) The origins of the five remaining axioms, which describe the interplay between τ and prefix iteration, are as follows. The equations PB1 and PB2 stem from [16], where a complete axiomatization for branching congruence over closed terms in the language BPA [9] with prefix iteration was presented. (For the sake of precision, we remark here that equation PB2 was formulated in [16] thus:

$$a.a^*(\tau.a^*(x+y)+x) \quad = \quad a.a^*(x+y) \ .$$

The two versions of equation PB2 are easily shown to be inter-derivable; each of them proves their common generalization

$$\text{PB2}' \quad \gamma.a^*(\tau.a^*(x+y)+x) \quad = \quad \gamma.a^*(x+y)$$

T3	$\alpha.(x+\tau.y)$	$=$	$\alpha.(x+\tau.y)+\alpha.y$
PT3	$a^*(x+\tau.y)$	$=$	$a^*(x+\tau.y+a.y)$

Table 4: Extra axioms for $\mathcal{E}_\eta$

AT3	$a.(x+\tau.y)$	$=$	$a.(x+\tau.y)+a.y$
PT3	$a^*(x+\tau.y)$	$=$	$a^*(x+\tau.y+a.y)$

Table 5: Extra axioms for $\mathcal{E}_w$

for $\gamma \in A_\tau$, using laws A1, A2, A4, PA1 and B1.) The equation PT1 was introduced in [8] under the name of FIR_1 (*Fair Iteration Rule*). In [8] it was also noted that this law is an equational formulation of *Koomen's Fair Abstraction Rule* [4]. (To be precise, Koomen's Fair Abstraction Rule is a general name for a family of proof rules KFAR_n, $n \geq 1$. PT1 corresponds to KFAR_1.) The laws PT2 and PT3 originate from [3], where the axiom system $\mathcal{E}_w$ was shown to be complete for weak congruence over $\mathrm{T}(\mathrm{BCCS})^{p*}(A_\tau)$, and ω-complete in the presence of a denumerable set of observable actions A.

Note that each of the axiom systems $\mathcal{E}_\aleph$ ($\aleph \in \{b, \eta, d, w\}$) is finite if so is the set of actions A.

Notation 3.1 *For an axiom system $\mathcal{T}$, we write $\mathcal{T} \vdash P = Q$ iff the equation $P = Q$ is provable from the axiom system $\mathcal{T}$ using the rules of equational logic. For axiom systems $\mathcal{T}, \mathcal{T}'$, we write $\mathcal{T} \vdash \mathcal{T}'$ iff $\mathcal{T} \vdash P = Q$ for every equation $(P = Q) \in \mathcal{T}'$. For a collection of equations X over the signature of* $\mathrm{BCCS}^{p*}(A_\tau)$*, we write $P \stackrel{\mathrm{X}}{=} Q$ as a short-hand for* A1,A2,X $\vdash P = Q$.

For $I = \{i_1, \ldots, i_n\}$ a finite index set, we write $\sum_{i \in I} P_i$ or $\sum\{P_i \mid i \in I\}$ for $P_{i_1} + \cdots + P_{i_n}$. By convention, $\sum_{i \in \emptyset} P_i$ stands for 0.

We establish the soundness of the axiom systems.

Proposition 3.2 *Let $\aleph \in \{b, \eta, d, w\}$. If $\mathcal{E}_\aleph \vdash P = Q$, then $P \leftrightarrow^c_\aleph Q$.*

Proof: As $\leftrightarrow^c_\aleph$ ($\aleph \in \{b, \eta, d, w\}$) is a congruence, it is sufficient to show that each equation in $\mathcal{E}_\aleph$ is sound with respect to it. The equations in the axiom system $\mathcal{F}$ are known to be sound with respect to strong bisimulation equivalence over $\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$; therefore they are, *a fortiori*, sound with respect to each of the congruences we consider. The soundness of the axioms B1, T1-3 and AT3 is well-known, and that of PB1-2 and PT1-3 is easy to check. $\square$

3.2 Expressiveness of the axiom systems

For use in the promised completeness theorems, we now study the relative expressive power of the axiom systems.

Proposition 3.3 *$\mathcal{E}_w \vdash \mathcal{E}_d \vdash \mathcal{E}_b$ and $\mathcal{E}_w \vdash \mathcal{E}_\eta \vdash \mathcal{E}_b$.*

Proof: Since $\mathcal{E}_w$ incorporates $\mathcal{E}_d$, and $\mathcal{E}_\eta$ incorporates $\mathcal{E}_b$, the statements $\mathcal{E}_w \vdash \mathcal{E}_d$ and $\mathcal{E}_\eta \vdash \mathcal{E}_b$ are trivially true. In order to prove the remaining two statements, $\mathcal{E}_d \vdash \mathcal{E}_b$ and $\mathcal{E}_w \vdash \mathcal{E}_\eta$, it suffices to show that the three axioms in Table 2 and the instance of T3 for $\alpha = \tau$ are derivable from $\mathcal{E}_d$. First of all, note that

$$\tau.(x+y) \stackrel{\text{A3,T2}}{=} \tau.(x+y) + x \ . \tag{5}$$

The derivability of the instance of T3 with $\alpha = \tau$ from $\mathcal{E}_d$ follows immediately by observing that, modulo commutativity of $+$, that equality is a substitution instance of (5). In deriving the laws in Table 2 from $\mathcal{E}_d$, we shall make use of the following derived equation:

$$a^*x + x \stackrel{\text{A3,PA1}}{=} a^*x \ . \tag{6}$$

The derivation of the three axioms in Table 2 from $\mathcal{E}_d$ now proceeds as follows:

B1 $\quad \alpha.(\tau.(x+y)+x) \stackrel{(5)}{=} \alpha.\tau.(x+y) \stackrel{\text{T1}}{=} \alpha.(x+y)$.

PB1 $\quad \tau^*x \stackrel{\text{PT1}}{=} \tau.x \stackrel{\text{T2}}{=} \tau.x + x$.

$$\begin{aligned}
\text{PB2}\quad \tau.a^*(\tau.a^*(x+y)+x) &\stackrel{(6)}{=} \tau.a^*(\tau.(a^*(x+y)+x+y)+x) \\
&\stackrel{(5)}{=} \tau.a^*(\tau.(a^*(x+y)+x+y)) \\
&\stackrel{(6)}{=} \tau.a^*(\tau.a^*(x+y)) \\
&\stackrel{\text{PT2}}{=} \tau.(\tau.a^*(x+y)) \\
&\stackrel{\text{T1}}{=} \tau.a^*(x+y) \ .
\end{aligned}$$

□

3.3 Irredundancy of the axiom systems

A collection $\mathcal{T}$ of equations is said to be *irredundant* [35, Page 389] iff for every proper subset $\mathcal{T}'$ of $\mathcal{T}$ there exists an equation which is derivable from $\mathcal{T}$, but not from $\mathcal{T}'$.

Experience has shown that axiom systems can contain redundancies; in the field of equational axiomatizations of behavioural congruences this happens for instance in [19]. Therefore, we find it interesting to conclude this section by addressing the issue of the irredundancy of the axiom systems $\mathcal{E}_\aleph$ ($\aleph \in \{b, \eta, d, w\}$).

Proposition 3.4 *For each $\aleph \in \{b, \eta, d, w\}$, the axiom system $\mathcal{E}_\aleph$ is irredundant.*

Proof: To show the irredundancy of the axiom system $\mathcal{E}_\aleph$ ($\aleph \in \{b, \eta, d, w\}$), it is sufficient to prove that, for every axiom $(P = Q) \in \mathcal{E}_\aleph$,

$$\mathcal{E}_\aleph \setminus \{P = Q\} \not\vdash P = Q \ . \tag{7}$$

The standard proof strategy to establish this kind of result is to find a model for the axiom system $\mathcal{E}_\aleph \setminus \{P = Q\}$ in which the equation $P = Q$ is not valid. As the axiom systems $\mathcal{E}_b$ and $\mathcal{E}_d$ are contained in $\mathcal{E}_\eta$ and $\mathcal{E}_w$, respectively, it is sufficient to show (7) for $\mathcal{E}_\eta$ and $\mathcal{E}_w$. In what follows, we limit ourselves to the proofs for the axioms PTn ($n = 1, 2, 3$) and PBn ($n = 1, 2$). We present the model explicitly only for axioms PT2, PB2 and PT3. For axioms PT1 and PB1 we merely give the intuition underlying the construction of an appropriate model. The reader will not have too much trouble in finding models which capture this intuition.

Axioms PT1 and PB1. Intuitively, the reason why equations PT1 and PB1 are not derivable from the axiom systems $\mathcal{E}_w \setminus \{\text{PT1}\}$ and $\mathcal{E}_\eta \setminus \{\text{PB1}\}$, respectively, is that PT1 and PB1 are the only equations that can be used to completely eliminate occurrences of the operation τ^* from terms.

Axioms PT2 and PB2. These axioms can actually be regarded as axiom schemes, in the sense that there is one axiom for each choice of an action $a \in A$. Call these instantiations PT2(a) and PT3(a). We now show that for all $a \in A$

$$\mathcal{E}_w \setminus \{\mathrm{PT2}(a)\} \not\vdash \mathrm{PT2}(a) \quad \text{and} \quad \mathcal{E}_b \setminus \{\mathrm{PB2}(a)\} \not\vdash \mathrm{PB2}(a) \ .$$

Let $a \in A$. We say that a term P is *stable* iff $P \xrightarrow{\tau} P'$ for no P'. A term whose sub-terms of the form a^*P' are stable is said to be *a^*-stable*. Intuitively, the reason why PT2(a) and PB2(a) cannot be derived from the other equations is that PT2(a) and PB2(a) are the only axioms in $\mathcal{E}_w$ and $\mathcal{E}_\eta$, respectively, that can be used to equate an a^*-stable term to one that is not.

Formally, define a denotational semantics for $\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$ in the domain $2^{\{0,1\}}$ by:

$$\begin{array}{rcll}
[\![x]\!]\rho & = & \rho(x) & \\
[\![0]\!]\rho & = & \emptyset & \\
[\![\tau.P]\!]\rho & = & [\![P]\!]\rho \cup \{1\} & \\
[\![b.P]\!]\rho & = & [\![P]\!]\rho \setminus \{1\} & \text{for } b \in A \\
[\![P+Q]\!]\rho & = & [\![P]\!]\rho \cup [\![Q]\!]\rho & \\
[\![\tau^*P]\!]\rho & = & [\![P]\!]\rho \cup \{1\} & \\
[\![b^*P]\!]\rho & = & \left\{ \begin{array}{ll} [\![P]\!]\rho \cup \{0\} & \text{if } b = a \wedge 1 \in [\![P]\!]\rho \\ [\![P]\!]\rho & \text{otherwise} \end{array} \right. &
\end{array}$$

where $\rho : \mathsf{Var} \to 2^{\{0,1\}}$. Here $1 \notin [\![P]\!]\rho$ denotes stability and $0 \notin [\![P]\!]\rho$ denotes a^*-stability. It is now simple to check that this is a model for both the axiom systems $\mathcal{E}_w \setminus \{\mathrm{PT2}(a)\}$ and $\mathcal{E}_\eta \setminus \{\mathrm{PB2}(a)\}$. However, letting $\rho_\emptyset$ map each variable in Var to $\emptyset$,

$$[\![\tau.(a^*x)]\!]\rho_\emptyset = \{1\} \neq \{0,1\} = [\![a^*(\tau.(a^*x))]\!]\rho_\emptyset$$

and

$$[\![\tau.a^*(\tau.a^*(x+y)+x)]\!]\rho_\emptyset = \{0,1\} \neq \{1\} = [\![\tau.a^*(x+y)]\!]\rho_\emptyset \ .$$

Therefore the above is neither a model of $\mathcal{E}_w$ nor one of $\mathcal{E}_\eta$.

Axiom PT3. Again we consider the instantiations PT3(a) and show $\mathcal{E}_w \setminus \{\mathrm{PT3}(a)\} \not\vdash \mathrm{PT3}(a)$. We say that a term P is *a-stable* iff $P \overset{a}{\Rightarrow} P'$ for no P'. Intuitively, the reason why PT3(a) cannot be derived from the other equations is that PT3(a) is the only axiom in $\mathcal{E}_w$ that can be used to equate a term P with a sub-term of the form a^*P' such that P' is a-stable to a term Q that does not have this property.

Formally, define a denotational semantics for $\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$ in the domain $2^{\{0,1\}}$ by:

$$\begin{array}{rcll}
[\![x]\!]\rho & = & \rho(x) & \\
[\![0]\!]\rho & = & \emptyset & \\
[\![\tau.P]\!]\rho & = & [\![P]\!]\rho & \\
[\![a.P]\!]\rho & = & [\![P]\!]\rho \cup \{1\} & \\
[\![b.P]\!]\rho & = & [\![P]\!]\rho \setminus \{1\} & \text{for } b \neq a \\
[\![P+Q]\!]\rho & = & [\![P]\!]\rho \cup [\![Q]\!]\rho & \\
[\![\alpha^*P]\!]\rho & = & [\![P]\!]\rho & \text{for } \alpha \neq a \\
[\![a^*P]\!]\rho & = & \left\{ \begin{array}{ll} \{0,1\} & \text{if } 1 \notin [\![P]\!]\rho \\ [\![P]\!]\rho & \text{otherwise} \end{array} \right. &
\end{array}$$

where $\rho : \mathsf{Var} \to 2^{\{0,1\}}$. Here $1 \notin [\![P]\!]\rho$ denotes a-stability and $0 \in [\![P]\!]\rho$ denotes the property of having a subterm a^*P' with P' a-stable. It is now simple to check that this is a model for the axiom system $\mathcal{E}_w \setminus \{\mathrm{PT3}(a)\}$. However, letting $\rho_\emptyset$ map each variable in Var to $\emptyset$,

$$[\![a^*(x + \tau.y)]\!]\rho_\emptyset = \{0, 1\} \neq \{1\} = [\![a^*(x + \tau.y + a.y)]\!]\rho_\emptyset$$

and so the above is not a model of $\mathcal{E}_w$. □

Remark: In light of (5), the instance of axiom T3 with $\alpha = \tau$ is derivable from the axiom system $\mathcal{E}_d$, and, *a fortiori*, from $\mathcal{E}_w$. Thus defining the axioms for weak congruence to include T3 in lieu of AT3 would lead to a redundant axiomatization, like those presented in, e.g., [26, 31].

4 Completeness

This section is entirely devoted to detailed proofs of the completeness of the axiom systems $\mathcal{E}_\aleph$ ($\aleph \in \{b, \eta, d, w\}$) with respect to $\leftrightarrow^c_\aleph$ over the language of open terms $\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$. A common and, we believe, aesthetically pleasing feature of our completeness proofs for the behavioural congruences $\leftrightarrow^c_\aleph$ ($\aleph \in \{\eta, d, w\}$) is that they are derived in uniform fashion from the corresponding results for branching congruence. Moreover, we shall also argue that the proof of completeness for weak congruence via reduction to the completeness result for branching congruence is considerably shorter than the only direct proof of this result presented in the literature. (Cf. [3].)

Because of the prominent rôle played by the completeness theorem for branching congruence in the developments to follow, we begin by presenting our proof of this result. We remark here that the completeness of the theory $\mathcal{E}_b$ with respect to $\leftrightarrow^c_b$ over the language of closed terms $\mathrm{T}(\mathrm{BCCS})^{p*}(A_\tau)$ was first shown in [16]. The proof presented below is, however, new, and yields the completeness of the axiom system $\mathcal{E}_b$ for the whole of the language $\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$. Moreover it may be argued that, even when restricted to the language of closed terms, our proof improves on the one offered in the aforementioned reference in that, unlike that proof, it does not rely on the completeness result for strong bisimulation from [15].

4.1 Completeness for branching congruence

We aim at identifying a subset of process terms of a special form, which will be convenient in the proof of the completeness result for branching congruence. Following a long-established tradition in the literature on process theory, we shall refer to these terms as *normal forms*. The set of normal forms we are after is the smallest subset of $\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$ including process terms having one of the following two forms:

$$\sum_{i\in I} \alpha_i.P_i + \sum_{j\in J} x_j \qquad \text{or} \qquad a^*(\sum_{i\in I} \alpha_i.P_i + \sum_{j\in J} x_j),$$

where the terms P_i are themselves normal forms, and I, J are finite index sets. (Recall that the empty sum represents 0.)

Lemma 4.1 *Each term in* $\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$ *can be proven equal to a normal form using equations* A4, PA1 *and* PB1*.*

Proof: A straightforward induction on the structure of process terms. For example, the term $\tau^*(a^*x)$ can be reduced to a normal form thus:

$$\begin{aligned} \tau^*(a^*x) &\stackrel{\mathrm{PB1}}{=} \tau.(a^*x) + a^*x \\ &\stackrel{\mathrm{PA1}}{=} \tau.(a^*x) + a.(a^*x) + x \\ &\stackrel{\mathrm{A4}}{=} \tau.(a^*(0+x)) + a.(a^*(0+x)) + x \ . \end{aligned}$$ □

Notation 4.2 *$P =_{\mathrm{AC}} Q$ denotes that P and Q are equal modulo associativity and commutativity of $+$,* i.e., *that* A1,A2 $\vdash P = Q$.

The following result is the key to the completeness theorem for branching congruence.

Proposition 4.3 *For all $P, Q \in \mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$, if $P \leftrightarrow_b Q$, then, for all $\gamma \in A_\tau$, $\mathcal{E}_b \vdash \gamma.P = \gamma.Q$.*

Proof: First of all, note that, as the equations in $\mathcal{E}_b$ are sound with respect to $\leftrightarrow_b^c$, and, a *fortiori*, for $\leftrightarrow_b$, by Lem. 4.1 it is sufficient to prove that the statement of the proposition holds for branching equivalent normal forms P and Q.

So, let us assume that P and Q be branching equivalent normal forms. We prove that $\mathcal{E}_b \vdash \gamma.P = \gamma.Q$ for all $\gamma \in A_\tau$, by complete induction on the sum of the sizes of P and Q. Recall that normal forms can take the following two forms:

$$\sum_i \alpha_i.P_i + \sum_j x_j \qquad \text{or} \qquad a^*(\sum_i \alpha_i.P_i + \sum_j x_j),$$

where the P_is are themselves normal forms. So, in particular, P and Q have one of these forms. By symmetry, it is sufficient to deal with the following three cases:

1. $P =_{\mathrm{AC}} \sum_i \alpha_i.P_i + \sum_k x_k$ and $Q =_{\mathrm{AC}} \sum_j \beta_j.Q_j + \sum_l y_l$;
2. $P =_{\mathrm{AC}} a^*(\sum_i \alpha_i.P_i + \sum_k x_k)$ and $Q =_{\mathrm{AC}} b^*(\sum_j \beta_j.Q_j + \sum_l y_l)$; and
3. $P =_{\mathrm{AC}} \sum_i \alpha_i.P_i + \sum_k x_k$ and $Q =_{\mathrm{AC}} a^*(\sum_j \beta_j.Q_j + \sum_l y_l)$.

We treat these three cases separately.

1. Case: $P =_{\mathrm{AC}} \sum_i \alpha_i.P_i + \sum_k x_k$ and $Q =_{\mathrm{AC}} \sum_j \beta_j.Q_j + \sum_l y_l$. Consider the following two conditions:

   A. $\alpha_i = \tau$ and $P_i \leftrightarrow_b Q$ for some i;

   B. $\beta_j = \tau$ and $Q_j \leftrightarrow_b P$ for some j.

   We distinguish three sub-cases in the proof, depending on which of the above conditions hold.

   I Suppose that neither A nor B holds. Then, as $P \leftrightarrow_b Q$, each transition $P \xrightarrow{\xi} P'$ must be matched by a transition $Q \xrightarrow{\xi} Q'$ with $P' \leftrightarrow_b Q'$. Hence, each summand $\alpha_i.P_i$ of P matches with a summand $\beta_j.Q_j$ of Q, in the sense that $\alpha_i = \beta_j$ and $P_i \leftrightarrow_b Q_j$. For each such pair of related summands, induction yields

   $$\mathcal{E}_b \vdash \alpha_i.P_i = \alpha_i.Q_j = \beta_j.Q_j \ .$$

   Moreover, each summand x_k of P must be a summand of Q. Hence, possibly using axiom A3, it follows that $\mathcal{E}_b \vdash P + Q = Q$. By symmetry, we infer that $\mathcal{E}_b \vdash P = P + Q = Q$. The fact that $\mathcal{E}_b \vdash \gamma.P = \gamma.Q$ for all $\gamma \in A_\tau$ is now immediate.

II Suppose that both of A and B hold. In this case, there exist i and j such that $\alpha_i = \beta_j = \tau$ and $P_i \leftrightarroweq_b Q \leftrightarroweq_b P \leftrightarroweq_b Q_j$. Applying the inductive hypothesis to the equivalences $P \leftrightarroweq_b Q_j$, $P_i \leftrightarroweq_b Q_j$ and $P_i \leftrightarroweq_b Q$, we infer that, for all $\gamma \in A_\tau$,

$$\mathcal{E}_b \vdash \gamma.P = \gamma.Q_j = \gamma.P_i = \gamma.Q$$

and the inductive step follows.

III Suppose that only one of A and B holds. In the remainder of the proof for this case, we shall assume, without loss of generality, that only A holds. For every summand $\tau.P_i$ of P with $P_i \leftrightarroweq_b Q$ we obtain, by induction, that

$$\mathcal{E}_b \vdash \tau.P_i = \tau.Q \ .$$

Hence, as A holds, by possibly using axioms A3 and/or A4 we infer that

$$\mathcal{E}_b \vdash P = \tau.Q + S$$

where $S =_{\mathrm{AC}} \sum\{\alpha_i.P_i \mid \alpha_i \neq \tau \ \text{ or } \ P_i \not\leftrightarroweq_b Q\} + \sum_k x_k$.
Consider now a summand $\alpha_i.P_i$ of S. As condition B does *not* hold and $P \leftrightarroweq_b Q$, using Lem. 2.10 it is not hard to see that there must exist a summand $\beta_{j_i}.Q_{j_i}$ of Q such that $\alpha_i = \beta_{j_i}$ and $P_i \leftrightarroweq_b Q_{j_i}$. By a similar reasoning, we infer that each one of the variables x_k must be a summand of Q. For related summands $\alpha_i.P_i$ and $\beta_{j_i}.Q_{j_i}$ of S and Q, respectively, induction yields

$$\mathcal{E}_b \vdash \alpha_i.P_i = \beta_{j_i}.Q_{j_i} \ .$$

It follows that $\mathcal{E}_b \vdash Q = R + S$, where

$$R =_{\mathrm{AC}} \sum\{\beta_j.Q_j \mid j \neq j_i \ \text{ for all } i\} + \sum\{y_l \mid y_l \neq x_k \ \text{ for all } k\} \ .$$

Now, for every $\gamma \in A_\tau$,

$$\mathcal{E}_b \vdash \gamma.P = \gamma.(\tau.Q + S) = \gamma.(\tau.(R+S) + S) \stackrel{\mathrm{B1}}{=} \gamma.(R+S) = \gamma.Q$$

and the inductive step follows.

2. Case: $P =_{\mathrm{AC}} a^*(\sum_i \alpha_i.P_i + \sum_k x_k)$ and $Q =_{\mathrm{AC}} b^*(\sum_j \beta_j.Q_j + \sum_l y_l)$. First of all, note that, by Lem. 2.11(2), it must be the case that $a = b$. Consider the following two conditions:

   A. $\alpha_i \in \{\tau, a\}$ and $P_i \leftrightarroweq_b Q$ for some i;

   B. $\beta_j \in \{\tau, a\}$ and $Q_j \leftrightarroweq_b P$ for some j.

We distinguish three sub-cases in the proof, depending on which of the above conditions hold.

I Suppose that neither A nor B holds. Then it is easy to see that

$$\sum_i \alpha_i.P_i + \sum_k x_k \leftrightarroweq_b \sum_j \beta_j.Q_j + \sum_l y_l$$

holds. As these two terms are normal forms whose combined size is smaller than that of P and Q, we may reason exactly as in the previous case 1.I to obtain that $\mathcal{E}_b \vdash \sum_i \alpha_i.P_i + \sum_k x_k = \sum_j \beta_j.Q_j + \sum_l y_l$. Hence $\mathcal{E}_b \vdash P = Q$, and the inductive step follows immediately.

II Suppose that both A and B hold. Then, as in case 1.II above, there exist i and j such that $P_i \leftrightarrow_b Q \leftrightarrow_b P \leftrightarrow_b Q_j$. Applying the inductive hypothesis to the equivalences $P \leftrightarrow_b Q_j$, $P_i \leftrightarrow_b Q_j$ and $P_i \leftrightarrow_b Q$, we infer that, for all $\gamma \in A_\tau$,

$$\mathcal{E}_b \vdash \gamma.P = \gamma.Q_j = \gamma.P_i = \gamma.Q$$

and the inductive step follows.

III Suppose that only one of A and B holds. In the remainder of the proof for this case, we shall assume, without loss of generality, that only A holds. We distinguish three further sub-cases.

IIIa Suppose that A holds for some indices i_1, i_2 with $\alpha_{i_1} = \tau$ and $\alpha_{i_2} = a$.
For every i with $\alpha_i \in \{\tau, a\}$ and $P_i \leftrightarrow_b Q$, induction yields $\mathcal{E}_b \vdash \alpha_i.P_i = \alpha_i.Q$. Hence, possibly using axioms A3 and/or A4, we infer that

$$\mathcal{E} \vdash P = a^*(\tau.Q + a.Q + S)$$

where $S =_{\mathrm{AC}} \sum\{\alpha_i.P_i \mid \alpha_i \notin \{\tau, a\} \text{ or } P_i \not\leftrightarrow_b Q\} + \sum_k x_k$. Reasoning as in case 1.III above, we find that $\mathcal{E}_b \vdash Q = a^*(R + S)$ for some term R. Now

$$\begin{array}{rcl}
\gamma.P & = & \gamma.a^*(\tau.Q + a.Q + S) \\
& = & \gamma.a^*\big(\tau.a^*(R+S) + a.a^*(R+S) + S\big) \\
& \overset{\mathrm{PA2}}{=} & \gamma.a^*\big(\tau.a^*a^*(R+S) + a.a^*(R+S) + S\big) \\
& \overset{\mathrm{PA1}}{=} & \gamma.a^*\Big(\tau.a^*\big(R + a.a^*(R+S) + S\big) + a.a^*(R+S) + S\Big) \\
& \overset{\mathrm{PB2}'}{=} & \gamma.a^*\big(R + a.a^*(R+S) + S\big) \\
& \overset{\mathrm{PA1}}{=} & \gamma.a^*a^*(R+S) \overset{\mathrm{PA2}}{=} \gamma.a^*(R+S) \;=\; \gamma.Q.
\end{array}$$

IIIb Suppose that A holds only for some i with $\alpha_i = \tau$.
Then, reasoning as in the case above, we obtain that $\mathcal{E}_b \vdash P = a^*(\tau.Q + S)$ (i.e., the summand $a.Q$ vanishes), and $\mathcal{E}_b \vdash Q = a^*(R + S)$ for some terms R and S. This yields the following simplification of the argument above:

$$\begin{array}{rcl}
\gamma.P & = & \gamma.a^*(\tau.Q + S) \\
& = & \gamma.a^*(\tau.a^*(R+S) + S) \\
& \overset{\mathrm{PB2}'}{=} & \gamma.a^*(R+S) \\
& = & \gamma.Q\ .
\end{array}$$

IIIc Suppose that A holds only for some i with $\alpha_i = a$.
Then, reasoning as in case IIIa above, we infer that

$$\mathcal{E}_b \vdash P = a^*(a.Q + S)$$

where $S =_{\mathrm{AC}} \sum\{\alpha_i.P_i \mid \alpha_i \neq a \text{ or } P_i \not\leftrightarrow_b Q\} + \sum_k x_k$ (i.e., $\tau.Q$ vanishes). Since B does not hold, it follows that every summand $\beta_j.Q_j$ of Q matches with a summand $\alpha_i.P_i$ of S, every y_l is equal to an x_k, and vice versa. Possibly using axiom A3, it follows that $\mathcal{E}_b \vdash Q = a^*S$ (i.e., R vanishes). Now

$$\gamma.P = \gamma.a^*(a.Q + S) = \gamma.a^*(a.a^*S + S) \overset{\mathrm{PA1}}{=} \gamma.a^*a^*S \overset{\mathrm{PA2}}{=} \gamma.a^*S = \gamma.Q.$$

3. CASE: $P =_{\mathrm{AC}} \sum_i \alpha_i.P_i + \sum_k x_k$ and $Q =_{\mathrm{AC}} a^*(\sum_j \beta_j.Q_j + \sum_l y_l)$. Consider the following two conditions:

A. $\alpha_i \in \{\tau, a\}$ and $P_i \leftrightarrow_b Q$ for some i;

B. $\beta_j \in \{\tau, a\}$ and $Q_j \leftrightarrow_b P$ for some j.

Since $Q \xrightarrow{a} Q$ and $P \leftrightarrow_b Q$, it follows that $P \stackrel{a}{\Rightarrow} P'$ with $P' \leftrightarrow_b Q$, for some P'. By the Stuttering Lemma 2.10, the intermediate states in the derivation $P \stackrel{a}{\Rightarrow} P'$ are all branching equivalent to Q. Hence there exists an index i such that $\alpha_i \in \{\tau, a\}$ and $P_i \leftrightarrow_b Q$. So we know that A holds. We proceed by distinguishing two sub-cases, depending on whether B holds or not.

I Suppose that B holds, so $Q_j \leftrightarrow_b P$ for some j. Then $Q_j \leftrightarrow_b P_i$ also holds, and the inductive hypothesis yields $\mathcal{E}_b \vdash \gamma.P = \gamma.Q_j = \gamma.P_i = \gamma.Q$, for all $\gamma \in A_\tau$, as desired.

II Suppose B does not hold. Reasoning as in case 2.III above, we can distinguish three sub-cases:

- A holds for some indices i_1 and i_2 with $\alpha_{i_1} = \tau$ and $\alpha_{i_2} = a$.
  Then, for some terms R and S, $\mathcal{E}_b \vdash P = \tau.Q + a.Q + S$ and $\mathcal{E}_b \vdash Q = a^*(R + S)$.
- A holds only for some index i such that $\alpha_i = \tau$.
  Then, for some terms R and S, $\mathcal{E}_b \vdash P = \tau.Q + S$ and $\mathcal{E}_b \vdash Q = a^*(R + S)$.
- A holds only for some index i such that $\alpha_i = a$.
  Then, for some term S, $\mathcal{E}_b \vdash P = a.Q + S$, and, since B does not hold, we find that $\mathcal{E}_b \vdash Q = a^* S$.

In all three cases we obtain $\mathcal{E}_b \vdash \gamma.P = \gamma.Q$, reasoning just as in case 2.III, but skipping the applications of PA2 and using B1 (in the second case with PA1) instead of PB2$'$.

The proof of the inductive step is now complete. □

Theorem 4.4 *Let $P, Q \in \mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$. If $P \leftrightarrow_b^c Q$, then $\mathcal{E}_b \vdash P = Q$.*

Proof: Consider two process terms P and Q that are branching congruent. We shall prove that $\mathcal{E}_b \vdash P = P + Q = Q$, from which the claim follows. In fact, by symmetry, it is sufficient to show that if $P \leftrightarrow_b^c Q$, then $\mathcal{E}_b \vdash P = P + Q$. To this end, note, first of all, that, by Lem. 4.1, P and Q may be proven equal to some normal forms using equations A4, PA1 and PB1. Possibly using equation PA1 again, we may therefore derive that

$$\begin{array}{l} \mathcal{E}_b \vdash P = \sum\{\alpha_i.P_i \mid i \in I\} + \sum\{x_j \mid j \in J\} \quad \text{and} \\ \mathcal{E}_b \vdash Q = \sum\{\beta_k.Q_k \mid k \in K\} + \sum\{y_l \mid l \in L\} \end{array}$$

for some finite index sets I, J, K, L. As $P \leftrightarrow_b^c Q$ and the equations in $\mathcal{E}_b$ are sound with respect to branching congruence, it follows that

1. for every k there exists an index i_k such that $\alpha_{i_k} = \beta_k$ and $P_{i_k} \leftrightarrow_b Q_k$, and
2. for every l there exists an index j_l such that $x_{j_l} \equiv y_l$.

By Propn. 4.3, for every k we may infer that

$$\mathcal{E}_b \vdash \alpha_{i_k}.P_{i_k} = \alpha_{i_k}.Q_k = \beta_k.Q_k \enspace .$$

The fact that $\mathcal{E}_b \vdash P = P + Q$ is now immediate using axiom A3. □

4.2 Completeness for η-, delay, and weak congruence

We now proceed to derive completeness results for η-, delay, and weak congruence from Thm. 4.4. The key to this derivation is the observation that, for certain classes of process terms, these congruence relations coincide with branching congruence. These classes of process terms are defined below.

Definition 4.5 *We say that a term P is:*

- η-saturated *iff for each of its derivatives Q, R and S and $\xi \in A_\tau \cup \mathsf{Var}$ we have that:*
$$Q \xrightarrow{\xi} R \xrightarrow{\tau} S \textit{ implies } Q \xrightarrow{\xi} S.$$
- d-saturated *iff for each of its derivatives Q, R and S and $\xi \in A_\tau \cup \mathsf{Var}$ we have that:*
$$Q \xrightarrow{\tau} R \xrightarrow{\xi} S \textit{ implies } Q \xrightarrow{\xi} S.$$
- w-saturated *iff it is both η- and d-saturated.*

The following theorem was first shown in [22] for process graphs. Here, we present its adaptation to open terms in the language $\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$.

Theorem 4.6 *Let $\aleph \in \{\eta, d, w\}$. If P and Q are $\aleph$-saturated and $P \leftrightarroweq^c_\aleph Q$, then $P \leftrightarroweq^c_b Q$.*

Proof: We only present the proof for weak congruence. The proofs for η- and delay congruence are simple variations on this theme, and the interested reader will have no difficulty in reconstructing them.

Note, first of all, that any two w-saturated terms that are weakly equivalent are also branching equivalent. This follows because the relation

$$\mathcal{B} \quad \triangleq \quad \{(S,T) \mid S \leftrightarroweq_w T, \quad S,T \ w\text{-saturated}\}$$

is a branching bisimulation.

Now, assume that $P \leftrightarroweq^c_w Q$ and that $P \xrightarrow{\xi} P'$. Then, there exists a Q' such that $Q \stackrel{\xi}{\Rightarrow} Q'$ and $P' \leftrightarroweq_w Q'$. As Q is w-saturated, it follows that $Q \xrightarrow{\xi} Q'$. Since P' and Q' are w-saturated and weakly equivalent, we infer that $P' \leftrightarroweq_b Q'$. Therefore, by symmetry, we finally obtain that $P \leftrightarroweq^c_b Q$, which was to be shown. □

Proposition 4.7 *Let $\aleph \in \{\eta, d, w\}$. For each term P, $\mathcal{E}_\aleph \vdash P = P'$ for some $\aleph$-saturated term P'.*

Proof: Again, we only present the details of the proof for $\aleph = w$. The proofs of η- and d-saturation are simple variations on this theme, and the interested reader will have no difficulty in reconstructing them.

A term P is *in head normal form* if it has the following form, where I, J are finite index sets:

$$P =_{\mathrm{AC}} \sum_{i \in I} \alpha_i . P_i + \sum_{j \in J} x_j \quad .$$

By induction on the structure of a process term T, we show that T can be proven equal to a process term that is both w-saturated and in head normal form, using the axiom system $\mathcal{E}_w$. The cases $T \equiv x$ and $T \equiv 0$ are trivial.

- Case: $T \equiv P + Q$. By the inductive hypothesis, P and Q can be transformed into w-saturated terms P' and Q' in head normal form, respectively. Then T is provably equal to $P' + Q'$, which is a w-saturated term, and may be turned into head normal form by possibly using A4.
- Case: $T \equiv a.P$. By the inductive hypothesis, P can be proven equal to a w-saturated term

  $$P' =_{\mathrm{AC}} \sum_{i \in I} b_i.P_i + \sum_{j \in J} \tau.Q_j + \sum_{k \in K} x_k \ .$$

  By AT3, T is provably equal to

  $$T' =_{\mathrm{AC}} a.P' + \sum_{j \in J} a.Q_j \ .$$

  We show that T' is w-saturated. Since P' and its derivatives are w-saturated, we only need to check the w-saturation condition for T' itself. Note that the case $T' \xrightarrow{\tau} R \xrightarrow{\xi} S$ does not apply. Assume that $T' \xrightarrow{\xi} R \xrightarrow{\tau} S$. Then $\xi = a$, and R is either P' or Q_j for some $j \in J$.

  If $R \equiv P'$, then $S \equiv Q_{j'}$ for some $j' \in J$. Therefore $T' \xrightarrow{a} S$ follows.

  If $R \equiv Q_j$, then $P' \xrightarrow{\tau} R \xrightarrow{\tau} S$. Since P' is w-saturated, it follows that $P' \xrightarrow{\tau} S$. Hence $S \equiv Q_{j'}$ for some $j' \in J$, and $T' \xrightarrow{a} S$ follows.
- Case: $T \equiv \tau.P$. By induction, P is provably equal to a w-saturated term

  $$P' =_{\mathrm{AC}} \sum_{i \in I} \alpha_i.P_i + \sum_{j \in J} x_j \ .$$

  By T2, T is provably equal to $T' \equiv \tau.P' + P'$. We show that T' is w-saturated. Since P' and its derivatives are w-saturated, we only need to check the w-saturation condition for transitions emanating from T' itself. We distinguish three possibilities.

  Assume that $T' \xrightarrow{\tau} P' \xrightarrow{\xi} Q$. Then $T' \xrightarrow{\xi} Q$ follows immediately.

  Assume that $T' \xrightarrow{\tau} P_i \xrightarrow{\xi} Q$ for some $i \in I$ with $\alpha_i = \tau$. Then $P' \xrightarrow{\tau} P_i \xrightarrow{\xi} Q$, and, as P' is w-saturated, it follows that $P' \xrightarrow{\xi} Q$. Hence $T' \xrightarrow{\xi} Q$, as desired.

  Assume that $T' \xrightarrow{\alpha_i} P_i \xrightarrow{\tau} Q$ for some $i \in I$. Then $P' \xrightarrow{\alpha_i} P_i \xrightarrow{\tau} Q$. As P' is w-saturated, it follows that $P' \xrightarrow{\alpha_i} Q$. Thus $T' \xrightarrow{\alpha_i} Q$, as desired.
- Case: $T \equiv a^*P$. By induction, P is provably equal to a w-saturated term

  $$P' =_{\mathrm{AC}} \sum_{i \in I} b_i.P_i + \sum_{j \in J} \tau.Q_j + \sum_{k \in K} x_k \ .$$

  Now,

  $$T \overset{\mathrm{PT3}}{=} a^*(P' + \sum_{j \in J} a.Q_j) \overset{\mathrm{PA1}}{=} a.a^*(P' + \sum_{j \in J} a.Q_j) + P' + \sum_{j \in J} a.Q_j \triangleq_{\mathrm{AC}} T' \ .$$

  We show that T' is w-saturated. All derivatives other than T' itself and $S \equiv a^*(P' + \sum_{j \in J} a.Q_j)$ are w-saturated by assumption, so we only need deal with these two cases. First, we deal with T'.

  - Let $T' \xrightarrow{\xi} Q \xrightarrow{\tau} R$. There are three possibilities:

- $\xi = a$ and $Q \equiv S$. Then $R \equiv Q_j$ for some $j \in J$, and thus $T' \xrightarrow{a} R$.
- $P' \xrightarrow{\xi} Q \xrightarrow{\tau} R$. In that case $P' \xrightarrow{\xi} R$ since P' is w-saturated, and thus $T' \xrightarrow{\xi} R$ follows.
- $\xi = a$ and $Q \equiv Q_j$ for some $j \in J$. In that case $P' \xrightarrow{\tau} Q_j \xrightarrow{\tau} R$. Thus $P' \xrightarrow{\tau} R$, since P' is w-saturated. Hence $R \equiv Q_{j'}$ for some $j' \in J$. Again $T' \xrightarrow{a} R$ follows.

• Let $T' \xrightarrow{\tau} Q \xrightarrow{\xi} R$. Then $P' \xrightarrow{\tau} Q \xrightarrow{\xi} R$. Thus $P' \xrightarrow{\xi} R$, since P' is w-saturated, and $T' \xrightarrow{\xi} R$ follows.

Next, we deal with S.

• Let $S \xrightarrow{\xi} Q \xrightarrow{\tau} R$. There are three possibilities:
  - $\xi = a$ and $Q \equiv S$. Then $R \equiv Q_j$ for some $j \in J$, and $S \xrightarrow{a} R$.
  - $P' \xrightarrow{\xi} Q \xrightarrow{\tau} R$. In that case $S \xrightarrow{\xi} R$ since P' is w-saturated.
  - $\xi = a$ and $Q \equiv Q_j$ for some $j \in J$. In that case $P' \xrightarrow{\tau} Q_j \xrightarrow{\tau} R$; therefore $P' \xrightarrow{\tau} R$, since P' is w-saturated. Hence $R \equiv Q_{j'}$ for some $j' \in J$. Again $S \xrightarrow{a} R$ follows.

• Let $S \xrightarrow{\tau} Q \xrightarrow{\xi} R$. Then $P' \xrightarrow{\tau} Q \xrightarrow{\xi} R$, and, since P' is w-saturated, $S \xrightarrow{\xi} R$ follows.

• CASE: $T \equiv \tau^* P$. Application of PT1 reduces this case to the one $T \equiv \tau.P$.

This completes the inductive argument. □

In light of Propn. 3.3, the results in Thm. 4.6 and Propn. 4.7 effectively reduce the completeness problem for η-, delay, and weak congruence over $\mathbb{T}(\mathrm{BCCS})^{p*}(A_\tau)$ to that for branching congruence.

Corollary 4.8 *Let $\aleph \in \{\eta, d, w\}$. If $P \leftrightarrow^c_\aleph Q$, then $\mathcal{E}_\aleph \vdash P = Q$.*

Proof: Let $\aleph \in \{\eta, d, w\}$. Suppose that $P \leftrightarrow^c_\aleph Q$. Prove P and Q equal to $\aleph$-saturated processes P' and Q', respectively (Propn. 4.7). By the soundness of the axiom system $\mathcal{E}_\aleph$ (Propn. 3.2), P' and Q' are $\aleph$-congruent. It follows that P' and Q' are branching congruent (Thm. 4.6). Hence, by Thm. 4.4, $\mathcal{E}_b \vdash P' = Q'$. The claim now follows because, by Propn. 3.3, the axioms for branching congruence are derivable from the theory $\mathcal{E}_\aleph$. □

References

[1] L. Aceto, B. Bloom and F. Vaandrager, *Turning SOS rules into equations*, Information and Computation 111 (1994), pp. 1–52.

[2] L. Aceto and J.F. Groote, *A complete equational axiomatization for MPA with string iteration*, Research Report RS–95–28, BRICS (Basic Research in Computer Science, Centre of the Danish National Research Foundation), Department of Mathematics and Computer Science, Aalborg University, May 1995. Available by anonymous ftp from `ftp.daimi.aau.dk` in the directory `pub/BRICS/RS/95/28`.

[3] L. Aceto and A. Ingólfsdóttir, *A complete equational axiomatization for prefix iteration with silent steps*, Research Report RS–95–5, BRICS (Basic Research

in Computer Science, Centre of the Danish National Research Foundation), Department of Mathematics and Computer Science, Aalborg University, Jan. 1995. To appear in the Proceedings of AMAST '96. Available by anonymous ftp from `ftp.daimi.aau.dk` in the directory `pub/BRICS/RS/95/5`.

[4] J. Baeten, J. Bergstra and J.W. Klop, *On the consistency of Koomen's fair abstraction rule*, Theoretical Comput. Sci. 51 (1987), pp. 129–176.

[5] J. Baeten and R. v. Glabbeek, *Another look at abstraction in process algebra*, in Proceedings 14^{th} ICALP, Karlsruhe, T. Ottmann, ed., vol. 267 of Lecture Notes in Computer Science, Springer-Verlag, July 1987, pp. 84–94.

[6] J. Baeten and J.W. Klop, eds., *Proceedings CONCUR 90,* Amsterdam, vol. 458 of Lecture Notes in Computer Science, Springer-Verlag, 1990.

[7] T. Basten, *Branching bisimilarity is an equivalence indeed!*, to appear in Information Processing Letters.

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

[9] J. Bergstra and J.W. Klop, *The algebra of recursively defined processes and the algebra of regular processes*, in Proceedings 11^{th} ICALP, Antwerpen, J. Paredaens, ed., vol. 172 of Lecture Notes in Computer Science, Springer-Verlag, 1984, pp. 82–95.

[10] B. Bloom, *Structural operational semantics for weak bisimulations*, Theoretical Comput. Sci. 146 (1995), pp. 25–68.

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

[12] F. Corradini, R. De Nicola and A. Labella, *Fully abstract models for nondeterministic Kleene algebras (extended abstract)*, in Proceedings CONCUR 95, Philadelphia, PA, USA, I. Lee and S. Smolka, eds., vol. 962 of Lecture Notes in Computer Science, Springer-Verlag, 1995, pp. 130–144.

[13] R. De Nicola and A. Labella, *A completeness theorem for nondeterministic Kleene algebras*, in Proceedings of MFCS '94, Košice, Slovakia, I. Prívara, B. Rovan and P. Pužička, eds., vol. 841 of Lecture Notes in Computer Science, Springer-Verlag, 1994.

[14] R. De Nicola, U. Montanari and F. Vaandrager, *Back and forth bisimulations*, in Baeten and Klop [6], pp. 152–165.

[15] W. Fokkink, *A complete equational axiomatization for prefix iteration*, Inf. Process. Lett. 52 (1994), pp. 333–337.

[16] ——, *A complete axiomatization for prefix iteration in branching bisimulation*, Logic Group Preprint Series 126, Dept. of Philosophy, Utrecht University, Jan. 1995. To appear in *Fundamenta Informaticae*. Available by ftp from `ftp.phil.ruu.nl` as `logic/PREPRINTS/preprint126.ps.Z`.

[17] ——, *On the completeness of the equations for the Kleene star in bisimulation*, Logic Group Preprint Series 141, Dept. of Philosophy, Utrecht University, Sept. 1995. To appear in the Proceedings of AMAST '96. Available by ftp from `ftp.phil.ruu.nl` as `logic/PREPRINTS/preprint141.ps.Z`.

[18] W. Fokkink and H. Zantema, *Basic process algebra with iteration: Completeness of its equational axioms*, Computer Journal 37 (1994), pp. 259–267.

[19] R. v. Glabbeek, *A complete axiomatization for branching bisimulation congruence of finite-state behaviours*, in Mathematical Foundations of Computer Science 1993, Gdansk, Poland, A. Borzyszkowski and S. Sokołowski, eds., vol. 711 of Lecture Notes in Computer Science, Springer-Verlag, 1993, pp. 473–484. Available by anonymous ftp from `Boole.stanford.edu` as `pub/DVI/complete.dvi.gz`.

[20] ——, *Branching bisimulation as a tool in the analysis of weak bisimulation.* Available at `ftp://boole.stanford.edu/pub/DVI/tool.dvi.gz`.

[21] R. v. Glabbeek and W.P. Weijland, *Branching time and abstraction in bisimulation semantics (extended abstract)*, in Information Processing 89, G. Ritter, ed., North-Holland, 1989, pp. 613–618. Full paper available as:

[22] ——, *Branching time and abstraction in bisimulation semantics*, Technical Report TUM-I9052, SFB-Bericht Nr. 342/29/90 A, Institut für Informatik, Technische Universität München, Germany, 1990. Extended abstract appeared as [21]. To appear in *Journal of the ACM.*

[23] J.F. Groote, *A new strategy for proving ω–completeness with applications in process algebra*, in Baeten and Klop [6], pp. 314–331.

[24] M. Hennessy, *A term model for synchronous processes*, Information and Control 51 (1981), pp. 58–75.

[25] ——, *Axiomatising finite delay operators*, Acta Inf. 21 (1984), pp. 61–88.

[26] M. Hennessy and R. Milner, *Algebraic laws for nondeterminism and concurrency*, Journal of the ACM 32 (1985), pp. 137–161.

[27] R. Keller, *Formal verification of parallel programs*, Comm. ACM 19 (1976), pp. 371–384.

[28] S. Kleene, *Representation of events in nerve nets and finite automata*, in Automata Studies, C. Shannon and J. McCarthy, eds., Princeton University Press, 1956, pp. 3–41.

[29] R. Milner, *A modal characterisation of observable machine behaviour*, in Proceedings CAAP 81, G. Astesiano and C. Bohm, eds., vol. 112 of Lecture Notes in Computer Science, Springer-Verlag, 1981, pp. 25–34.

[30] ——, *Communication and Concurrency*, Prentice-Hall International, Englewood Cliffs, 1989.

[31] ——, *A complete axiomatisation for observational congruence of finite-state behaviours*, Information and Computation 81 (1989), pp. 227–247.

[32] D. Park, *Concurrency and automata on infinite sequences*, in 5^{th} GI Conference, Karlsruhe, Germany, P. Deussen, ed., vol. 104 of Lecture Notes in Computer Science, Springer-Verlag, 1981, pp. 167–183.

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

[34] P. Sewell, *Bisimulation is not finitely (first order) equationally axiomatisable*, in Proceedings 9^{th} Annual Symposium on Logic in Computer Science, Paris, France, IEEE Computer Society Press, 1994, pp. 62–70.

[35] W. Taylor, *Equational logic*, Appendix 4 of the book Universal Algebra by G. Grätzer, Springer-Verlag, 1979, pp. 378–400.