A Cook's Tour of Equational Axiomatizations for Prefix Iteration

Luca Aceto* Wan Fokkink† Anna Ingólfsdóttir‡

1 Introduction

Equationally based proof systems play an important role in both the practice and the theory of process algebras. From the point of view of practice, equationally based proof systems can be used to perform system verifications in a purely syntactic way, and form the basis of axiomatic verification tools like, e.g., PAM [12]. From the theoretical point of view, complete axiomatizations of behavioural equivalences capture the essence of different notions of semantics for processes in terms of a basic collection of identities, and this often allows one to compare semantics which may have been defined in very different styles and frameworks. Some researchers also measure the naturalness of a process semantics by using the existence of a finite complete axiomatization for it over, say, finite behaviours as an acid test.

An excellent example of the unifying role played by equational axiomatizations of process semantics may be found in [8]. *Ibidem* van Glabbeek presents the so-called linear time/branching time spectrum, i.e., the lattice of all the known behavioural equivalences over labelled transition systems ordered by inclusion. The different identifications made by these semantic equivalences over finite synchronization trees are beautifully characterized by the author of *op. cit.* in terms of a few simple axioms. This permits an illuminating comparison of these semantics within a uniform axiomatic framework. However, despite the complete inference systems for bisimulation-based equivalences over regular processes presented in, e.g., [13, 9] and years of intense research, little is still known on the topic of effective complete axiomatizations of the notions of semantics studied in [8] over iterative processes.

In this study, we shall present a contribution to this line of research by investigating a significant fragment of the notions of equivalence and preorder from [8] over Milner's basic CCS (henceforth referred to as BCCS) [14] extended with prefix iteration. Prefix iteration [7] is a variation on the original binary version of the Kleene star operation P^*Q [11], obtained by restricting the first argument to be an atomic action, and yields simple iterative behaviours that can be equationally characterized by means of finite collections of axioms. Furthermore, prefix iteration combines better with the action prefixing operator of CCS than the more general binary Kleene star. A significant advantage of iteration over recursion, as a means to express infinite processes, is that it does not involve a parametric process defini-

*BRICS (Basic Research in Computer Science), Centre of the Danish National Research Foundation, Department of Computer Science, Aalborg University, Fr. Bajersvej 7E, 9220 Aalborg Ø, Denmark. Partially supported by the Human Capital and Mobility project EXPRESS. Email: luca@cs.auc.dk. Fax: +45 9815 9889.

†Department of Computer Science, University of Wales Swansea, Singleton Park, Swansea SA2 8PP, Wales. Email: W.J.Fokkink@swansea.ac.uk. Fax: +44 1792 295708.

‡Dipartimento di Sistemi ed Informatica, Università di Firenze, Via Lombroso 6/17, 50134 Firenze, Italy. Supported by a grant from the Danish National Research Foundation. Email: annai@dsi2.ing.unifi.it. Fax: +39 55 4796730.

tion, because the development of process theory is easier if parameterization does not have to be taken as primitive (see, e.g., Milner [15, page 212]).

Our study of equational axiomatizations for BCCS with prefix iteration has so far yielded complete equational axiomatizations for all the main notions of bisimulation equivalence [7, 1]. In this paper, we continue this research programme by studying axiomatic characterizations for more abstract semantics over this language than those based on variations of bisimulation. More precisely, we consider ready simulation, simulation, readiness, trace and language semantics, and provide complete (in)equational axiomatizations for each of these notions over BCCS with prefix iteration. All of the axiom systems we present are finite, if so is the set of atomic actions under consideration. Although the high level structure of the proofs of our main results follows standard lines in the literature on process theory, the actual details of the arguments are, however, rather subtle (cf., e.g., the proofs of Thms. 4.7 and 4.12). To our mind, this shows how the analysis of the collection of valid identities for the semantics considered in this paper already becomes difficult even in the presence of very simple iterative behaviours, like those that can be expressed using prefix iteration.

The paper is organized as follows. After a brief review of the basic notions from process theory needed in the remainder of the paper (Sect. 2), we present the language BCCS with prefix iteration and its labelled transition system semantics (Sect. 3). Sect. 4 is devoted to a guided tour of our completeness results. The paper concludes with a mention of further results that will be presented in a full account of this work, and a discussion of ongoing research (Sect. 5).

2 Preliminaries

In this section we present the basic notions from process theory that will be needed in the remainder of this study.

2.1 Labelled Transitions Systems

We begin by reviewing the model of labelled transition systems [10] that abstracts from the operational semantics of many concurrent calculi.

Definition 2.1 (Labelled Transition Systems) *A* labelled transition system *is a triple* $(\mathsf{Proc}, \mathsf{Lab}, \{\stackrel{\ell}{\to} \mid \ell \in \mathsf{Lab}\})$*, where:*

- Proc *is a set of* states*, ranged over by s, possibly subscripted or superscripted;*
- Lab *is a set of* labels*, ranged over by ℓ, possibly subscripted;*
- $\stackrel{\ell}{\to} \subseteq \mathsf{Proc} \times \mathsf{Proc}$ *is a* transition relation*, for every $\ell \in \mathsf{Lab}$. As usual, we shall use the more suggestive notation $s \stackrel{\ell}{\to} s'$ in lieu of $(s, s') \in \stackrel{\ell}{\to}$, and write $s \stackrel{\ell}{\not\to}$ iff $s \stackrel{\ell}{\to} s'$ for no state s'.*

All the labelled transition systems we shall consider in this paper will have a special label $\checkmark$ in their label set—used to represent successful termination—, and will enjoy the following property: if $s \stackrel{\checkmark}{\to} s'$, then $s' \stackrel{\ell}{\not\to}$ for every label ℓ.

For $n \geq 0$ and $\varsigma = \ell_1 \ldots \ell_n \in \mathsf{Lab}^*$, we write $s \stackrel{\varsigma}{\to} s'$ iff there exist states $s_0, \ldots, s_n$ such that $s = s_0 \stackrel{\ell_1}{\to} s_1 \stackrel{\ell_2}{\to} \cdots s_{n-1} \stackrel{\ell_n}{\to} s_n = s'$. In that case, we say that ς is a *trace* (of length n) of the state s. For a state $s \in \mathsf{Proc}$ we define:

$$\mathsf{initials}(s) \quad \triangleq \quad \left\{\ell \in \mathsf{Lab} \mid \exists s' :\ s \stackrel{\ell}{\to} s'\right\} \ .$$

2.2 From Ready Simulation to Language Equivalence

Labelled transition systems describe the operational behaviour of processes in great detail. In order to abstract from irrelevant information on the way processes compute, a wealth of notions of behavioural equivalence or approximation have been studied in the literature on process theory. A systematic investigation of these notions is presented in [8], where van Glabbeek studies the so-called linear time/branching time spectrum, i.e., the lattice of all the known behavioural equivalences over labelled transition systems ordered by inclusion. In this study, we shall investigate a significant fragment of the notions of equivalence and preorder from [8]. These we now proceed to present for the sake of completeness.

Definition 2.2 (Simulation, Ready Simulation and Bisimulation)

- *A binary relation $\mathcal{R}$ on states is a* simulation *iff whenever $s_1 \mathcal{R} s_2$ and ℓ is a label:*

  - *if $s_1 \stackrel{\ell}{\to} s_1'$, then there is a transition $s_2 \stackrel{\ell}{\to} s_2'$ such that $s_1' \mathcal{R} s_2'$.*

- *A binary relation $\mathcal{R}$ on states is a* ready simulation *iff it is a simulation with the property that, whenever $s_1 \mathcal{R} s_2$ and ℓ is a label:*

  - *if $s_1 \stackrel{\ell}{\not\to}$, then $s_2 \stackrel{\ell}{\not\to}$.*

- *A* bisimulation *is a symmetric simulation.*

Two states s and s' are bisimilar, *written $s \underline{\leftrightarrow} s'$, iff there is a bisimulation that relates them. Henceforth the relation $\underline{\leftrightarrow}$ will be referred to as* bisimulation equivalence. *We write $s \precsim_S s'$ (resp. $s \precsim_{RS} s'$) iff there is a simulation (resp. a ready simulation) $\mathcal{R}$ with $s \mathcal{R} s'$.*

Bisimulation equivalence [16] relates two states in a labelled transition system precisely when they have the same branching structure. Simulation (see, e.g., [16]) and ready simulation [4] relax this requirement to different degrees. The following notion, which is based on a version of decorated traces, is induced by yet another way of abstracting from the full branching structure of processes.

Definition 2.3 (Readiness Semantics) *For a state s we define:*

$$\text{readies}(s) \quad \triangleq \quad \left\{(\varsigma, X) \mid \varsigma \in \mathsf{Lab}^*, X \subseteq \mathsf{Lab} \text{ and } \exists s' : s \stackrel{\varsigma}{\to} s' \text{ and } \mathsf{initials}(s') = X\right\}$$

For states s, s' we write $s \precsim_R s'$ iff readies(s) *is included in* readies(s').

The classical notion of language equivalence for finite state automata may be readily defined over labelled transition systems. To this end, it is sufficient to consider the states from which a ✓-labelled transition is possible as accept states.

Definition 2.4 (Language and Trace Semantics)

- *We say that a sequence of labels ς is accepted by a state s iff $s \stackrel{\varsigma\checkmark}{\to} s'$ for some state s'. For states s, s' we write $s \precsim_L s'$ iff every sequence accepted by s is also accepted by s'.*

- *For states s, s' we write $s \precsim_T s'$ iff the set of traces of s is included in that of s'.*

For $\Theta \in \{S, RS, L, R, T\}$, the relation $\precsim_\Theta$ is a preorder over states of an arbitrary labelled transition system; its kernel will be denoted by $\simeq_\Theta$.

3 BCCS with Prefix Iteration

We begin by presenting the language of Basic CCS (henceforth often abbreviated to BCCS) with prefix iteration [7], together with its operational semantics.

3.1 The Syntax

We assume a non-empty alphabet Act of atomic actions, with typical elements a, b, c. The language $\mathrm{T}(\mathrm{BCCS}^{p*}(\mathsf{Act}))$ of Basic CCS with prefix iteration is given by the following BNF grammar:

$$P ::= \mathbf{0} \mid \mathbf{1} \mid a.P \mid P + P \mid a^*P \ .$$

We shall use P, Q, R, S, T to range over $\mathrm{T}(\mathrm{BCCS}^{p*}(\mathsf{Act}))$. In writing terms over the above syntax, we shall always assume that the operator $a.\_$ binds stronger than $+$. We shall use the symbol $\equiv$ to stand for syntactic equality of terms. The expression $P[+Q]$ will be used to denote the fact that Q is an optional summand.

Remark: The reader familiar with Milner's CCS [14], and with some of our previous work on prefix-iteration [7, 2, 1], might have noticed that the syntax for the language $\mathrm{T}(\mathrm{BCCS}^{p*}(\mathsf{Act}))$ presented above includes two distinguished constants, viz. $\mathbf{0}$ and $\mathbf{1}$. Intuitively, the term $\mathbf{0}$ will stand for a deadlocked process, whereas $\mathbf{1}$ will stand for a process that can only terminate immediately with success. Our choice of notation is in keeping with a standard one for regular expressions, cf., e.g., [6].

3.2 Operational Semantics

Let $\checkmark$ be a distinguished symbol not contained in Act. We shall use $\checkmark$ to stand for the action performed by a process as it reports its successful termination. The meta-variable ξ will range over the set $\mathsf{Act} \cup \{\checkmark\}$. The operational semantics for the language $\mathrm{T}(\mathrm{BCCS}^{p*}(\mathsf{Act}))$ is given by the labelled transition system

$$\left(\mathrm{T}(\mathrm{BCCS}^{p*}(\mathsf{Act})), \mathsf{Act} \cup \{\checkmark\}, \left\{ \stackrel{\xi}{\to} \mid \xi \in \mathsf{Act} \cup \{\checkmark\} \right\}\right)$$

where the transition relations $\stackrel{\xi}{\to}$ are the least binary relations over $\mathrm{T}(\mathrm{BCCS}^{p*}(\mathsf{Act}))$ satisfying the rules in Table 1. Intuitively, a transition $P \stackrel{a}{\to} Q$ means that the system represented by the term P can perform the action a, thereby evolving into Q. On the other hand, $P \stackrel{\checkmark}{\to} Q$ means that P can terminate immediately with success; the reader will immediately realize that, in that case, $Q \equiv \mathbf{0}$.

$$\frac{}{a.P \stackrel{a}{\to} P} \qquad \frac{}{\mathbf{1} \stackrel{\checkmark}{\to} \mathbf{0}}$$

$$\frac{P \stackrel{\xi}{\to} P'}{P + Q \stackrel{\xi}{\to} P'} \qquad \frac{Q \stackrel{\xi}{\to} Q'}{P + Q \stackrel{\xi}{\to} Q'}$$

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

Table 1: Transition Rules

With the above definitions, the language $\mathrm{T}(\mathrm{BCCS}^{p*}(\mathsf{Act}))$ inherits all the notions of equivalence and preorder over processes defined in Sect. 2.2. The following result is standard.

Proposition 3.1 *For $\Theta \in \{RS, S, L, R, T\}$, the relations $\sqsubseteq_\Theta$ and $\simeq_\Theta$ are preserved by the operators in the signature of* $\mathrm{T}(\mathrm{BCCS}^{p*}(\mathsf{Act}))$*. The same holds for bisimulation equivalence.*

Definition 3.2 *The set of* derivatives *of P is the least set containing P that is closed under the transition relations.*

It is easy to see that the set of derivatives of every $P \in \mathrm{T}(\mathrm{BCCS}^{p*}(\mathsf{Act}))$ is finite. The following variation on a result, first proven in [2] for bisimulation-based equivalences over basic CCS with prefix iteration, will find application in the remainder of the paper (cf. the proof of Thm. 4.12).

Proposition 3.3 *Let $a, b \in \mathsf{Act}$. If $a^*P \simeq_T b^*Q$, then $a = b$.*

4 Equational Axiomatizations

The study of equational axiomatizations of behavioural equivalences and preorders over $\mathrm{T}(\mathrm{BCCS}^{p*}(\mathsf{Act}))$ was initiated in the paper [7]. In *op. cit.* it is shown that the axiom system in Table 2 completely axiomatizes bisimulation equivalence over the language of **1**-free $\mathrm{T}(\mathrm{BCCS}^{p*}(\mathsf{Act}))$ terms. Our aim in the remainder of this study will be to extend this result to the semantics in the linear-time/branching-time spectrum discussed in Sect. 2.2.

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

Table 2: The axiom system $\mathcal{F}$

Notation 4.1 *For an axiom system $\mathcal{T}$, we write $\mathcal{T} \vdash P \leq Q$ iff the inequation $P \leq Q$ is provable from the axioms in $\mathcal{T}$ using the rules of inequational logic. An equation $P = Q$ will be used as a short-hand for the pair of inequations $P \leq Q$ and $Q \leq P$. Whenever we write an inequation of the form $P[+\mathbf{1}] \leq Q[+\mathbf{1}]$, we mean that if the **1** summand appears on the left-hand side of the inequation, then it also appears on the right-hand side.*

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

For a collection of (in)equations X over the signature of $\mathrm{T}(\mathrm{BCCS}^{p*}(\mathsf{Act}))$*, we write $P \overset{(X)}{\leq} Q$ as a short-hand for A1,A2,$X \vdash P \leq Q$.*

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

Henceforth process terms will be considered modulo associativity and commutativity of the +-operation, i.e., modulo axioms A1–2.

We begin the technical developments by noting that the proof of the completeness of the axiom system $\mathcal{F}$ with respect to bisimulation equivalence over the language of **1**-free $\mathrm{T}(\mathrm{BCCS}^{p*}(\mathsf{Act}))$ terms applies *mutatis mutandis* to the whole of the language $\mathrm{T}(\mathrm{BCCS}^{p*}(\mathsf{Act}))$.

Proposition 4.2 *For every* $P, Q \in \mathrm{T}(\mathrm{BCCS}^{p*}(\mathsf{Act}))$, $P \leftrightarrow Q$ *iff* $\mathcal{F} \vdash P = Q$.

The collection of possible transitions of each process term P is finite, say $\{P \xrightarrow{a_i} P_i \mid i = 1, ..., m\} \cup \left\{P \xrightarrow{\checkmark} \mathbf{0} \mid j = 1, \ldots, n\right\}$. We call the term

$$\exp(P) \triangleq \sum_{i=1}^{m} a_i.P_i + \sum_{j=1}^{n} \mathbf{1}$$

the *expansion* of P. The terms $a_i P_i$ and $\mathbf{1}$ will be referred to as the *summands* of P. A straightforward structural induction on terms, using axiom PA1, yields:

Lemma 4.3 *Each process term is provably equal to its expansion.*

We aim at identifying a subset of process terms of a special form, which will be convenient in the proof of the completeness results to follow. 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 $\mathrm{T}(\mathrm{BCCS}^{p*}(\mathsf{Act}))$ including process terms having one of the following two forms:

$$\sum_{i \in I} a_i.P_i[+\mathbf{1}] \qquad \text{or} \qquad a^*(\sum_{i \in I} a_i.P_i[+\mathbf{1}]),$$

where the terms P_i are themselves normal forms, and I is a finite index set. (Recall that the empty sum represents $\mathbf{0}$, and the notation $[+\mathbf{1}]$ stands for optional inclusion of $\mathbf{1}$ as a summand.)

Lemma 4.4 *Each term in* $\mathrm{T}(\mathrm{BCCS}^{p*}(\mathsf{Act}))$ *can be proven equal to a normal form using equations* A3, A4 *and* PA1.

4.1 Ready Simulation

We begin our tour of equational axiomatizations for prefix iteration by presenting a complete axiom system for the ready simulation preorder (cf. Defn. 2.2 for the definition of this relation). The axiom system $\mathcal{E}_{RS}$ consists of the laws for bisimulation equivalence (cf. Table 2) and of those listed in Table 3.

RS1	$a.x$	$\leq$	$a.x + a.y$
RS2	a^*x	$\leq$	$a^*(x + a.y)$

Table 3: The Axioms for Ready Simulation

Theorem 4.5 *For every* $P, Q \in \mathrm{T}(\mathrm{BCCS}^{p*}(\mathsf{Act}))$, $P \sqsubseteq_{RS} Q$ *iff* $\mathcal{E}_{RS} \vdash P \leq Q$.

Proof: We leave it to the reader to check the soundness of the axiom system $\mathcal{E}_{RS}$, and concentrate on its completeness. In view of Lem. 4.4, it is sufficient to show that if $P \sqsubseteq_{RS} Q$ holds for normal forms P and Q, then $\mathcal{E}_{RS} \vdash P \leq Q$. This we now proceed to prove by induction on the sum of the sizes of P and Q.

We proceed by a case analysis on the form the normal forms P and Q may take.

- CASE: $P =_{\mathrm{AC}} \sum_{i \in I} a_i.P_i[+\mathbf{1}]$ and $Q =_{\mathrm{AC}} \sum_{j \in J} b_j.Q_j[+\mathbf{1}]$.

  As $P \sqsubseteq_{RS} Q$, we infer that:

  1. for every i there exists an index j_i such that $a_i = b_{j_i}$ and $P_i \sqsubseteq_{RS} Q_{j_i}$,
  2. $\mathbf{1}$ is a summand of P iff it is a summand of Q, and
  3. the collections of actions $\{a_i \mid i \in I\}$ and $\{b_j \mid j \in J\}$ are equal.

  The induction hypothesis and substitutivity yield that, for every $i \in I$,

  $$\mathcal{E}_{RS} \vdash a_i.P_i \quad \leq \quad b_{j_i}.Q_{j_i} \ .$$

  Again using substitutivity, we obtain that

  $$\mathcal{E}_{RS} \vdash P \quad \leq \quad \sum_i b_{j_i}.Q_{j_i}[+\mathbf{1}] \ .$$

  Note now that, for every index j that is not contained in the set $\{j_i \mid i \in I\}$, there is an index j_l $(l \in I)$ such that $b_j = b_{j_l}$. We can therefore apply axiom RS1 as necessary to infer that

  $$\mathcal{E}_{RS} \vdash \sum_i b_{j_i}.Q_{j_i}[+\mathbf{1}] \quad \leq \quad Q \ .$$

  The provability of the inequation $P \leq Q$ from the axiom system $\mathcal{E}_{RS}$ now follows immediately by transitivity.

- CASE: $P =_{\mathrm{AC}} \sum_{i \in I} a_i.P_i[+\mathbf{1}]$ and $Q =_{\mathrm{AC}} b^*(\sum_{j \in J} b_j.Q_j[+\mathbf{1}])$.
  To deal with this case, begin by applying PA1 to Q to obtain the equality

  $$Q \quad = \quad b.Q + \sum_{j \in J} b_j.Q_j[+\mathbf{1}] \ .$$

  We can now reason as in the first case of the proof to derive that

  $$P \quad \leq \quad b.Q + \sum_{j \in J} b_j.Q_j[+\mathbf{1}] \ .$$

  Transitivity now yields the inequation $P \leq Q$.

- CASE: $P =_{\mathrm{AC}} a^*(\sum_i a_i.P_i[+\mathbf{1}])$ and $Q =_{\mathrm{AC}} \sum_j b_j.Q_j[+\mathbf{1}]$.
  Apply PA1 to P, and reason as in the previous case.

- CASE: $P =_{\mathrm{AC}} a^*(\sum_i a_i.P_i[+\mathbf{1}])$ and $Q =_{\mathrm{AC}} b^*(\sum_j b_j.Q_j[+\mathbf{1}])$.

  As $P \sqsubseteq_{RS} Q$, we infer that:

  1. there exists a Q' such that $Q \xrightarrow{a} Q'$ and $P \sqsubseteq_{RS} Q'$,
  2. for every i there exists a $Q(i)$ such that $Q \xrightarrow{a_i} Q(i)$ and $P_i \sqsubseteq_{RS} Q(i)$,
  3. $\mathbf{1}$ is a summand of P iff it is a summand of Q, and
  4. the collections of actions $\{a_i \mid i \in I\} \cup \{a\}$ and $\{b_j \mid j \in J\} \cup \{b\}$ are equal.

  Because of the form Q takes, Q' and every $Q(i)$ is either Q itself or one of the Q_j's. Therefore we may apply the inductive hypothesis to each of the inequivalences $P_i \sqsubseteq_{RS} Q(i)$ and substitutivity to infer that

  $$\mathcal{E}_{RS} \vdash \sum_i a_i.P_i \quad \leq \quad \sum_i a_i.Q(i) \ . \tag{1}$$

  We proceed with the proof by considering the following two sub-cases:

A. There is an index j such that $a = b_j$ and $P \precsim_{RS} Q_j$;

B. For no index j with $a = b_j$ it holds that $P \precsim_{RS} Q_j$.

We consider these two cases in turn.

A. Assume that there is an index j such that $a = b_j$ and $P \precsim_{RS} Q_j$. In this case, we may apply the inductive hypothesis to derive that

$$\mathcal{E}_{RS} \vdash P \quad \leq \quad Q_j \ . \tag{2}$$

We can now finish the proof of the inequation $P \leq Q$ from the axiom system $\mathcal{E}_{RS}$ as follows:

$$\begin{aligned} P \quad & \stackrel{\text{(PA1)}}{=} \quad a.P + \sum_i a_i.P_i[+\mathbf{1}] \\ & \stackrel{(1),(2)}{\leq} \quad b_j.Q_j + \sum_i a_i.Q(i)[+\mathbf{1}] \\ & \stackrel{\text{(RS1)}}{\leq} \quad b_j.Q_j + \sum_i a_i.Q(i) + \exp(Q)[+\mathbf{1}] \\ & \stackrel{\text{(A3),(PA1)}}{=} \quad Q \ . \end{aligned}$$

B. Assume that for no index j with $a = b_j$ it holds that $P \precsim_{RS} Q_j$. In this case, we infer that $a = b$. We can now reason as follows:

$$\begin{aligned} P \equiv a^*(\sum_i a_i.P_i[+\mathbf{1}]) \quad & \stackrel{(1)}{\leq} \quad a^*(\sum_i a_i.Q(i)[+\mathbf{1}]) \\ & \stackrel{\text{(RS1),(RS2)}}{\leq} \quad a^*(\sum_i a_i.Q(i) + a.Q + \sum_j b_j.Q_j[+\mathbf{1}]) \\ & \stackrel{\text{(A3),(PA1)}}{\leq} \quad a^*Q \\ & \stackrel{\text{(PA2)}}{=} \quad Q \ . \end{aligned}$$

This completes the proof of the theorem. □

4.2 Simulation

The axiom system $\mathcal{E}_S$ consists of the laws for bisimulation equivalence in Table 2 and of the axiom

$$\text{(S)} \qquad x \quad \leq \quad x + y \ .$$

Inequation S is well-known to characterize the simulation preorder over finite synchronization trees. Unlike in the case of ready simulation, no extra law is needed to deal with prefix iteration explicitly.

Theorem 4.6 *For every* $P, Q \in \mathrm{T}(\mathrm{BCCS}^{p*}(\mathsf{Act}))$, $P \precsim_S Q$ *iff* $\mathcal{E}_S \vdash P \leq Q$.

The proof of the completeness theorem for simulation is similar to the one for ready simulation, and is therefore omitted.

4.3 Readiness

In this section we present a complete axiom system for prefix iteration with respect to the readiness preorder. The axiom system $\mathcal{E}_R$ consists of the collection of laws for ready simulation and of those listed in Table 4.

R1	$a.(b.x+b.y+v)$	$\leq$	$a.(b.x+v)+a.(b.y+w)$
R2	$a.a^*(b.x+b.y+v)$	$\leq$	$a.a^*(b.x+v)+a.a^*(b.y+w)$
R3	$a^*(b.x+b.y+v+a.(b.y+w))$	$=$	$a^*(b.x+v+a.(b.y+w))+b.y$

Table 4: Axioms for Readiness

Theorem 4.7 *For every* $P, Q \in \mathrm{T}(\mathrm{BCCS}^{p*}(\mathsf{Act}))$, $P \precsim_R Q$ *iff* $\mathcal{E}_R \vdash P \leq Q$.

We focus on the completeness of $\mathcal{E}_R$, and leave soundness to the reader. Before proving this completeness theorem, we introduce some auxiliary definitions and results.

Definition 4.8 *A term* P *is* saturated *if for each pair of derivations* $P \xrightarrow{a} Q \xrightarrow{b} Q'$ *and* $P \xrightarrow{a} R$ *with* $b \in \mathsf{initials}(R)$ *we have* $R \xrightarrow{b} R'$ *with* $Q' \precsim_R R'$.

The following lemma stems from [3].

Lemma 4.9 *If* $P \precsim_R Q$ *and* $P \xrightarrow{a} P'$ *and* Q *is saturated, then* $Q \xrightarrow{a} Q'$ *with* $P' \precsim_R Q'$.

Definition 4.10 *A normal form* P *is* strongly saturated *if:*

1. *P is saturated;*
2. *if* $P =_{AC} \sum_{i\in I} a_i.P_i[+\mathbf{1}]$, *then the term* P_i *is strongly saturated, for every* $i \in I$.

Axioms R1–R3 play a crucial role in the proof of the following key result.

Lemma 4.11 *Each term is provably equal, by the axioms in* $\mathcal{E}_R$, *to a strongly saturated normal form, in which each subterm of the form* a^*R *occurs in the context* $a.\_$.

Finally we are in a position to prove Thm. 4.7.

Proof: Suppose that $P \precsim_R Q$; we prove that $\mathcal{E}_R \vdash P \leq Q$. By Lem. 4.11 it is not hard to see that it suffices to establish the claim under the following assumptions:

1. P and Q are normal forms;
2. Q is strongly saturated;
3. proper subterms of P and Q of the form a^*R occur in the context $a.\_$;
4. if $P =_{\mathrm{AC}} a^*R$ and $Q =_{\mathrm{AC}} b^*S$, then $a = b$.

(In fact, according to Lem. 4.11, the last two conditions could be replaced by the stronger condition that *all* subterms of P and Q of the form a^*R occur in the context $a.\_$. However, we shall need the weaker formulation above to be able to satisfy the induction hypothesis.) We derive the desired inequality $P \leq Q$ from $\mathcal{E}_R$ by induction with respect to the following lexicographic ordering on pairs of process terms: $(P, Q) < (R, S)$ if

- either $size(P) < size(R)$;
- or $size(P) = size(R)$ and $size(Q) < size(S)$.

The next two cases distinguish the possible syntactic forms of P.

- CASE 1: $P =_{\mathrm{AC}} \sum_{i\in I} a_i.P_i[+\mathbf{1}]$.

  Since $P \precsim_R Q$, $P \xrightarrow{a_i} P_i$ and Q is saturated, Lem. 4.9 implies that for each $i \in I$ we have $Q \xrightarrow{a_i} Q_i$ for some Q_i such that $P_i \precsim_R Q_i$. According to Lem. 4.11, $\mathcal{E}_R \vdash Q_i = R_i$, with R_i a strongly saturated normal form, in which each subterm of the form c^*S occurs in the context $c.\_$. Moreover, each P_i is a normal form, in which all proper subterms of the form c^*S occur in the context $c.\_$, with $\mathit{size}(P_i) < \mathit{size}(P)$. Hence, we can apply induction to $P_i \precsim_R R_i$ to derive $\mathcal{E}_R \vdash P_i \leq R_i$. Therefore, for each $i \in I$,

$$\mathcal{E}_R \vdash a_i.P_i \leq a_i.R_i = a_i.Q_i \ . \tag{3}$$

  By substitutivity, we have that

$$P =_{\mathrm{AC}} \sum_{i\in I} a_i.P_i[+\mathbf{1}] \overset{(3)}{\leq} \sum_{i\in I} a_i.Q_i[+\mathbf{1}] \ . \tag{4}$$

  Since $P \precsim_R Q$ implies $\mathsf{initials}(P) = \mathsf{initials}(Q)$, it follows that $\mathsf{initials}(Q) \setminus \{\checkmark\}$ is equal to $\{a_i \mid i \in I\}$. Furthermore, $P \precsim_R Q$ implies that P has a summand $\mathbf{1}$ if and only if $Q \xrightarrow{\checkmark} \mathbf{0}$. Hence,

$$\sum_{i\in I} a_i.Q_i[+\mathbf{1}] \overset{(\mathrm{RS1})}{\leq} \exp(Q) \overset{(\mathrm{Lem.4.3})}{=} Q$$

  which together with equation (4) yields $\mathcal{E}_R \vdash P \leq Q$.

- CASE 2: $P =_{\mathrm{AC}} a^*(\sum_{i\in I} a_i.P_i[+\mathbf{1}])$.

  The next two cases distinguish the possible syntactic forms of Q.

- CASE 2.1: $Q =_{\mathrm{AC}} \sum_{j\in J} b_j.Q_j[+\mathbf{1}]$.

  Suppose that $P \xrightarrow{c} P'$. Since $P \precsim_R Q$ and Q is saturated, Lem. 4.9 implies that there is a $j \in J$ such that $c = b_j$ and $P' \precsim_R Q_j$. Both P' and Q_j are normal forms, and since Q is strongly saturated, by Defn. 4.10(2) Q_j is strongly saturated too. Furthermore, if $P' =_{\mathrm{AC}} d^*R$ and $Q_j =_{\mathrm{AC}} e^*S$, then $c = d$ and $b_j = e$, owing to property 3 of P and Q, and so $d = c = b_j = e$. Moreover, it is easy to see that property 3 of P and Q implies that the same property holds for P' and Q_j. Finally, $\mathit{size}(P') \leq \mathit{size}(P)$ and $\mathit{size}(Q_j) < \mathit{size}(Q)$. Hence, we can apply induction to $P' \precsim_R Q_j$ to derive $\mathcal{E}_R \vdash P' \leq Q_j$. Substitutivity now yields

$$\mathcal{E}_R \vdash c.P' \leq b_j.Q_j \ . \tag{5}$$

  Hence,

$$P \overset{(\mathrm{Lem.4.3})}{=} \exp(P) \overset{(5)}{\leq} \sum_{j\in J_0} b_j.Q_j[+\mathbf{1}] \tag{6}$$

  for some $J_0 \subseteq J$. It is easy to see that $P \precsim_R Q$ implies $\mathsf{initials}(Q) \setminus \{\checkmark\} = \mathsf{initials}(P) \setminus \{\checkmark\} = \{b_j \mid j \in J_0\}$. Moreover, $P \xrightarrow{\checkmark} \mathbf{0}$ if and only if Q has a summand $\mathbf{1}$. Hence,

$$\sum_{j\in J_0} b_j.Q_j[+\mathbf{1}] \overset{(\mathrm{RS1})}{\leq} \sum_{j\in J} b_j.Q_j[+\mathbf{1}] =_{\mathrm{AC}} Q \ .$$

  Together with equation (6) this yields $\mathcal{E}_R \vdash P \leq Q$.

- CASE 2.2: $Q =_{\mathrm{AC}} a^*(\sum_{j\in J} b_j.Q_j[+\mathbf{1}])$.

  Since $P \precsim_R Q$ and $P \xrightarrow{a_i} P_i$ and Q is saturated, Lem. 4.9 implies that for each $i \in I$

  1. either $a_i = a$ and $P_i \precsim_R Q$,
  2. or there is a j such that $a_i = b_j$ and $P_i \precsim_R Q_j$.

Clearly, each P_i is a normal form in which all proper subterms of the form c^*S occur in the context $c.\_$, and with $size(P_i) < size(P)$.
In the first case, applying induction to $P_i \sqsubseteq_R Q$, we infer that $\mathcal{E}_R \vdash P_i \leq Q$. Therefore, by substitutivity,

$$\mathcal{E}_R \vdash a_i.P_i \leq a.Q \ . \tag{7}$$

In the second case, Lem. 4.11 implies $\mathcal{E}_R \vdash Q_j = R_j$, with R_j a strongly saturated normal form, in which each subterm of the form c^*S occurs in the context $c.\_$. Then by induction $P_i \sqsubseteq_R R_j$ implies $\mathcal{E}_R \vdash P_i \leq R_j$. It follows, by substitutivity, that

$$\mathcal{E}_R \vdash a_i.P_i \leq a_i.R_j = b_j.Q_j \ . \tag{8}$$

Hence, for some $J_0 \subseteq J$:

$$P \overset{\text{(RS2)}}{\leq} a^*(a.Q + \sum_{i \in I} a_i.P_i[+\mathbf{1}]) \overset{(8),(7)}{\leq} a^*(a.Q + \sum_{j \in J_0} b_j.Q_j[+\mathbf{1}]) \ . \tag{9}$$

It is easy to see that $P \sqsubseteq_R Q$ implies that $\mathsf{initials}(Q) \setminus \{\checkmark\} = \{b_j \mid j \in J_0\} \cup \{a\}$, and that $P \xrightarrow{\checkmark} \mathbf{0}$ if and only if $Q \xrightarrow{\checkmark} \mathbf{0}$. Hence

$$a^*(a.Q + \sum_{j \in J_0} b_j.Q_j[+\mathbf{1}]) \overset{\text{(RS1)}}{\leq} a^*(a.Q + \sum_{j \in J} b_j.Q_j[+\mathbf{1}]) \overset{\text{(PA1),(PA2)}}{=} Q \ .$$

Together with equation (9) this yields $\mathcal{E}_R \vdash P \leq Q$.

The proof is now complete. □

4.4 Traces

The axiom system $\mathcal{E}_T$ consists of the laws for bisimulation equivalence in Table 2 and those in Table 5. Axiom T1 is a well-known equation used to characterize trace equivalence over finite synchronization trees, and axiom T2 is the adaptation of this equation to the case of prefix iteration. Finally, T3 is, to the best of our knowledge, a new axiom.

T1	$a.(x+y)$	$=$	$a.x + a.y$
T2	$a^*(x+y)$	$=$	$a^*x + a^*y$
T3	$a^*(a.x)$	$=$	$a.(a^*x)$

Table 5: Axioms for Traces

Theorem 4.12 *For every $P, Q \in \mathrm{T}(\mathrm{BCCS}^{p*}(\mathsf{Act}))$,*

1. *$P \simeq_T Q$ iff $\mathcal{E}_T \vdash P = Q$;*
2. *$P \sqsubseteq_T Q$ iff $\mathcal{E}_T \cup \{(\mathrm{S})\} \vdash P \leq Q$.*

Proof: We prove the two statements separately. We leave it to the reader to check the soundness of the axiom system $\mathcal{E}_T \cup \{(\mathrm{S})\}$, and concentrate on the proofs of the completeness results.

1. Assume that $P \simeq_T Q$. We shall prove that $\mathcal{E}_T \vdash P = Q$.
   By a term rewriting analysis, which is omitted here, it can be shown that each process term is provably equal, by means of the axiom system $\mathcal{E}_T$, to a trace normal form of the form
   $$\sum_{i \in I} a_i^*(\sum_{j \in J_i} b_{ij}.P_{ij} + \sum_{\alpha \in A_i} \mathbf{1}) \; + \; \sum_{k \in K} c_k.Q_k \; + \; \sum_{\beta \in B} \mathbf{1}$$
   where $a_i \neq b_{ij}$ for all $i \in I$ and $j \in J_i$, and all the terms P_{ij} and Q_k are themselves trace normal forms. (Recall that the empty sum stands for $\mathbf{0}$.)
   Suppose that two trace normal forms P and Q are trace equivalent. We prove by induction on the sum of the sizes of P and Q that the equation $P = Q$ can be derived from $\mathcal{E}_T$. Since P and Q are trace normal forms, we have that
   $$P \;=_{\mathrm{AC}}\; \sum_{i \in I} a_i^* R_i \;+\; \sum_{m \in M} a'_m.T_m \;+\; \sum_{\gamma \in C} \mathbf{1}$$
   $$Q \;=_{\mathrm{AC}}\; \sum_{j \in J} b_j^* S_j \;+\; \sum_{n \in N} b'_n.U_n \;+\; \sum_{\delta \in D} \mathbf{1}$$
   where
   $$R_i =_{\mathrm{AC}} \sum_{k \in K_i} c_{ik}.R_{ik} + \sum_{\alpha \in A_i} \mathbf{1} \qquad i \in I$$
   $$S_j =_{\mathrm{AC}} \sum_{\ell \in L_j} d_{j\ell}.S_{j\ell} + \sum_{\beta \in B_j} \mathbf{1} \qquad j \in J$$
   with $a_i \neq c_{ik}$, $b_j \neq d_{j\ell}$ and R_{ik}, $S_{j\ell}$, T_m, U_n trace normal forms for all $i \in I$, $k \in K_i$, $j \in J$, $\ell \in L_j$, $m \in M$ and $n \in N$. We distinguish three cases.
   - Case 1: $I = M = C = \varnothing$.
     In other words, $P \equiv \mathbf{0}$. Since $P \simeq_T Q$, it then follows that $J = N = D = \varnothing$, so that $P \equiv \mathbf{0} \equiv Q$.
   - Case 2: $|I| = 1$ and $M = C = \varnothing$, and $|J| = 1$ and $N = D = \varnothing$; say $I = \{i\}$ and $J = \{j\}$. Since $a_i^* R_i \;\simeq_T\; b_j^* S_j$, it follows that $a_i = b_j$ (Propn. 3.3). Note, furthermore, that
     $$R_i \xrightarrow{\checkmark} \mathbf{0} \text{ iff } P \xrightarrow{\checkmark} \mathbf{0} \text{ iff } Q \xrightarrow{\checkmark} \mathbf{0} \text{ iff } S_j \xrightarrow{\checkmark} \mathbf{0}.$$
     Therefore, since $a_i \neq c_{ik}$ for $k \in K_i$ and $a_i \neq d_{j\ell}$ for $L \in L_j$, we can conclude that $R_i \;\simeq_T\; S_j$. Since the sizes of these two trace normal forms are strictly smaller than those of P and Q, respectively, induction then yields that $\mathcal{E}_T \vdash R_i \;=\; S_j$. As $a_i = b_j$, it follows that $P \;=_{\mathrm{AC}}\; a_i^* R_i \;=\; b_j^* S_j \;=_{\mathrm{AC}}\; Q$ can be derived from $\mathcal{E}_T$.
   - Case 3: $|I| \geq 2$ or $M \neq \varnothing$ or $C \neq \varnothing$.
     For every action $a \in \mathsf{Act}$, define the following index sets:
     $$\begin{array}{rcl} I_a &=& \{i \in I \mid a_i = a\} \\ J_a &=& \{j \in J \mid b_j = a\} \\ K_{ia} &=& \{k \in K_i \mid c_{ik} = a\} \\ L_{ja} &=& \{\ell \in L_j \mid d_{j\ell} = a\} \\ M_a &=& \{m \in M \mid a'_m = a\} \qquad \text{and} \\ N_a &=& \{n \in N \mid b'_n = a\}\ . \end{array}$$
     Note that, due to the restrictions on trace normal forms, if $i \in I_a$ then $K_{ia} = \varnothing$. So using axioms A4 and PA1 and T1,2 we can derive the equation
     $$P = \sum_{a \in \mathsf{initials}(P)} a.\Big(a^*\left(\textstyle\sum_{i \in I_a} R_i\right) + \textstyle\sum_{m \in M_a} T_m + \textstyle\sum_{i \in I \setminus I_a} \textstyle\sum_{k \in K_{ia}} R_{ik}\Big) + \sum_{i \in I} \sum_{\alpha \in A_i} \mathbf{1} + \sum_{\gamma \in C} \mathbf{1}\ . \qquad (10)$$

Similarly, if $j \in J_a$ then $L_{ja} = \varnothing$. Thus, again using axioms A4 and PA1 and T1,2 we can also derive the equation

$$Q = \sum_{a \in \mathsf{initials}(Q)} a.\left(a^*\left(\sum_{j\in J_a} S_j\right) + \sum_{n\in N_a} U_n + \sum_{j\in J\setminus J_a} \sum_{\ell \in L_{ja}} S_{j\ell}\right) + \sum_{j\in J}\sum_{\beta\in B_j} \mathbf{1} + \sum_{\delta\in D} \mathbf{1} \ . \tag{11}$$

Since $P \simeq_T Q$, it follows that $\mathsf{initials}(P) = \mathsf{initials}(Q)$. Fix an $a \in \mathsf{initials}(P)$. Then $P \simeq_T Q$ yields

$$a^*(\sum_{i\in I_a} R_i) + \sum_{m\in M_a} T_m + \sum_{i\in I\setminus I_a}\sum_{k\in K_{ia}} R_{ik} \simeq_T a^*(\sum_{j\in J_a} S_j) + \sum_{n\in N_a} U_n + \sum_{j\in J\setminus J_a}\sum_{\ell\in L_{ja}} S_{j\ell} \ .$$

Since $|I| \geq 2$ or $M \neq \varnothing$ or $C \neq \varnothing$, the term on the left-hand side of the above equality has size strictly smaller than P. Moreover, the term on the right-hand side has size no greater than Q. So it follows by induction that we can derive

$$a^*(\sum_{i\in I_a} R_i) + \sum_{m\in M_a} T_m + \sum_{i\in I\setminus I_a}\sum_{k\in K_{ia}} R_{ik} = a^*(\sum_{j\in J_a} S_j) + \sum_{n\in N_a} U_n + \sum_{j\in J\setminus J_a}\sum_{\ell\in L_{ja}} S_{j\ell} \tag{12}$$

from $\mathcal{E}_T$. Finally, since $P \xrightarrow{\checkmark} \mathbf{0}$ if and only if $Q \xrightarrow{\checkmark} \mathbf{0}$, it follows that $(\cup_{i\in I} A_i) \cup C$ is empty if and only if so is $(\cup_{j\in J} B_j) \cup D$. If both collections are non-empty, then axiom A3 yields

$$\sum_{i\in I}\sum_{\alpha\in A_i} \mathbf{1} + \sum_{\gamma\in C} \mathbf{1} \ = \ \mathbf{1} \ = \ \sum_{j\in J}\sum_{\beta\in B_j} \mathbf{1} + \sum_{\delta\in D} \mathbf{1} \ . \tag{13}$$

Equations (10), (11) and (13), together with equations (12) for every $a \in \mathsf{initials}(P)$, now yield $\mathcal{E}_T \vdash P = Q$. This finishes the proof of the completeness of the axiom system $\mathcal{E}_T$.

2. Note that, for every $P, Q \in \mathrm{T}(\mathrm{BCCS}^{p*}(\mathsf{Act}))$, the following holds:

$$P \sqsubseteq_T Q \text{ iff } P + Q \simeq_T Q \ .$$

Thus the completeness of the axiom system $\mathcal{E}_T \cup \{(\mathrm{S})\}$ with respect to $\sqsubseteq_T$ is an immediate consequence of the first statement of the theorem.

□

4.5 Language Semantics

The axiom system $\mathcal{E}_L$ consists of the laws for bisimulation equivalence in Table 2, those for trace equivalence in Table 5, and the equations in Table 6. Axiom L1 is an adaptation to action prefixing of a well-known equation from regular algebra, and axiom L2 is the generalization of this equation to the case of prefix iteration.

Theorem 4.13 *For every $P, Q \in \mathrm{T}(\mathrm{BCCS}^{p*}(\mathsf{Act}))$,*

1. $P \simeq_L Q$ *iff* $\mathcal{E}_L \vdash P = Q$;
2. $P \sqsubseteq_L Q$ *iff* $\mathcal{E}_L \cup (S) \vdash P \leq Q$.

L1	$a.\mathbf{0}$	$=$	$\mathbf{0}$
L2	$a^*\mathbf{0}$	$=$	$\mathbf{0}$

Table 6: Axioms for Completed Traces

Proof: We leave it to the reader to check the soundness of the axiom system $\mathcal{E}_L \cup$ (S), and concentrate on the completeness results.

1. Assume that $P \simeq_L Q$. We shall prove that $\mathcal{E}_L \vdash P = Q$. A simple term rewriting analysis (which is omitted here) shows that each process term is provably equal to a term which is either $\mathbf{0}$-free, or of the form $\mathbf{0}$.

   Suppose that two terms P and Q are language equivalent. We distinguish two cases.

   - CASE 1: $P \equiv \mathbf{0}$. Then clearly also $Q \equiv \mathbf{0}$, so $P \equiv \mathbf{0} \equiv Q$.
   - CASE 2: P is $\mathbf{0}$-free. Then clearly Q is also $\mathbf{0}$-free. Since P and Q are $\mathbf{0}$-free and language equivalent, it is not hard to see that they are also trace equivalent. So, according to Thm. 4.12, the equation $P = Q$ can be derived from $\mathcal{E}_T$, which is included in $\mathcal{E}_L$.

2. Note that, for every $P, Q \in \mathrm{T}(\mathrm{BCCS}^{p*}(\mathsf{Act}))$, the following holds:

   $$P \sqsubseteq_L Q \text{ iff } P + Q \simeq_L Q \ .$$

   Thus the completeness of the axiom system $\mathcal{E}_L \cup \{(\mathrm{S})\}$ with respect to $\sqsubseteq_T$ is an immediate consequence of the first statement of the theorem.

□

5 Further Work

The completeness results presented in this paper deal with a significant fragment of the notions of semantics discussed in [8]. To our mind, the most important omission is a complete proof system for failures semantics [5] over BCCS with prefix iteration. We conjecture that a complete axiomatization for the failure preorder can be obtained by adding the laws in Table 7 to those for bisimulation equivalence (cf. Table 2), and we are currently working on the details of such a proof. The crux of the argument is a proof to the effect that the suggested inequations are sufficient to convexly saturate each process term, in the sense of [3].

F1	$a.(x+y)$	$\leq$	$a.x + a.(y+z)$
F2	$a.a^*(x+y)$	$\leq$	$a.a^*x + a.a^*(y+z)$
F3	$a.a^*x$	$\leq$	$a^*a.(x+y)$
F4	$a^*(x+y+a.(y+z))$	$\leq$	$a^*(x+a.(y+z))+y$
RS2	a^*x	$\leq$	$a^*(x+a.y)$

Table 7: Axioms for Failures

We have also obtained irredundancy results for the axioms systems for ready simulation, simulation, trace and language equivalence. These will be presented in the full version of this paper, together with a characterization of the expressive power of BCCS with prefix iteration.

Acknowledgements: The research reported in this paper originates from a question posed by Rocco De Nicola.

References

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

[2] L. Aceto and A. Ingólfsdóttir, *An equational axiomatization of observation congruence for prefix iteration*, in Algebraic Methodology and Software Technology: 5th International Conference, AMAST '96, Munich, Germany, M. Wirsing and M. Nivat, eds., vol. 1101 of Lecture Notes in Computer Science, Springer-Verlag, July 1996, pp. 195–209. Full version available as BRICS Report RS-95-5, January 1995.

[3] J. Bergstra, J. W. Klop, and E.-R. Olderog, *Readies and failures in the algebra of communicating processes*, SIAM J. Comput., 17 (1988), pp. 1134–1177.

[4] B. Bloom, S. Istrail, and A. R. Meyer, *Bisimulation can't be traced*, J. Assoc. Comput. Mach., 42 (1995), pp. 232–268.

[5] S. Brookes, C. Hoare, and A. Roscoe, *A theory of communicating sequential processes*, J. Assoc. Comput. Mach., 31 (1984), pp. 560–599.

[6] J. H. Conway, *Regular Algebra and Finite Machines*, Mathematics Series (R. Brown and J. De Wet eds.), Chapman and Hall, London, United Kingdom, 1971.

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

[8] R. J. v. Glabbeek, *The linear time – branching time spectrum*, in Proceedings CONCUR 90, Amsterdam, J. Baeten and J. Klop, eds., vol. 458 of Lecture Notes in Computer Science, Springer-Verlag, 1990, pp. 278–297.

[9] ——, *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`.

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

[11] 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.

[12] H. Lin, *An interactive proof tool for process algebras*, in 9th Annual Symposium on Theoretical Aspects of Computer Science, vol. 577 of Lecture Notes in Computer Science, Cachan, France, 13–15 Feb. 1992, Springer, pp. 617–618.

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

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

[15] ——, *The polyadic π-calculus: a tutorial*, in Proceedings Marktoberdorf Summer School '91, Logic and Algebra of Specification, NATO ASI Series F94, Springer-Verlag, 1993, pp. 203–246.

[16] 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.