A Menagerie of Non-Finitely Based Process Semantics over BPA*

From Ready Simulation to Completed Traces

Luca Aceto* Wan Fokkink† Anna Ingólfsdóttir‡

Abstract

Fokkink and Zantema ((1994) *Computer Journal* **37**:259–267) have shown that bisimulation equivalence has a finite equational axiomatization over the language of Basic Process Algebra with the binary Kleene star operation (BPA*). In the light of this positive result on the mathematical tractability of bisimulation equivalence over BPA*, a natural question to ask is whether any other (pre)congruence relation in van Glabbeek's linear time/branching time spectrum is finitely (in)equationally axiomatizable over it. In this paper, we prove that, unlike bisimulation equivalence, none of the preorders and equivalences in van Glabbeek's linear time/branching time spectrum, whose discriminating power lies in between that of ready simulation and that of completed traces, has a finite equational axiomatization. This we achieve by exhibiting a family of (in)equivalences that holds in ready simulation semantics, the finest semantics that we consider, whose instances cannot all be proven by means of any finite set of (in)equations that is sound in completed trace semantics, which is the coarsest semantics that is appropriate for the language BPA*. To this end, for every finite collection of (in)equations that are sound in completed trace semantics, we build a model in which some of the (in)equivalences of the family under consideration fail. The construction of the model mimics the one used by Conway ((1971) *Regular Algebra and Finite Machines*, page 105) in his proof of a result, originally due to Redko, to the effect that infinitely many equations are needed to axiomatize equality of regular expressions.

Our non-finite axiomatizability results apply to the language BPA* over an arbitrary non-empty set of actions. In particular, we show that completed trace equivalence is not finitely based over BPA* even when the set of actions is a singleton. Our proof of this result may be easily adapted to the standard language of regular expressions to yield a solution to an open problem posed by Salomaa ((1969) *Theory of Automata*, page 143).

*__BRICS__ (**B**asic **R**esearch in **C**omputer **S**cience), Centre of the Danish National Research Foundation, Department of 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 the Human Capital and Mobility project Express. Email: `luca@iesd.auc.dk`. Fax: +45 9815 9889.

†Utrecht University, Department of Philosophy, Heidelberglaan 8, 3584 CS Utrecht, The Netherlands. Email: `fokkink@phil.ruu.nl`. Fax: +31 30 253 2816.

‡__BRICS__ (**B**asic **R**esearch in **C**omputer **S**cience), Centre of the Danish National Research Foundation, Department of Computer Science, Aalborg University, Fr. Bajersvej 7E, 9220 Aalborg Ø, Denmark. Email: `annai@iesd.auc.dk`. Fax: +45 9815 9889.

Another semantics that is usually considered in process theory is trace semantics. Trace semantics is, in general, not preserved by sequential composition, and is therefore inappropriate for the language BPA*. We show that, if the set of actions is a singleton, trace equivalence and preorder are preserved by all the operators in the signature of BPA*, and coincide with simulation equivalence and preorder, respectively. In that case, unlike all the other semantics considered in this paper, trace semantics have finite, complete equational axiomatizations over closed terms.

AMS Subject Classification (1991): 08A70, 03C05, 68Q10, 68Q40, 68Q45, 68Q55, 68Q68, 68Q70.
CR Subject Classification (1991): D.3.1, F.1.1, F.1.2, F.3.2, F.3.4, F.4.1.
Keywords and Phrases: Concurrency, process algebra, Basic Process Algebra (BPA), Kleene star, bisimulation, ready simulation, simulation, completed trace semantics, ready trace semantics, failure trace semantics, readiness semantics, failures semantics, trace semantics, equational logic, complete axiomatizations.

1 Introduction

Process theory aims at providing a framework for the description and analysis of reactive systems, i.e., systems that compute by reacting to stimuli from their environment. As such systems tend to be non-terminating, all process algebraic specification formalisms (cf., e.g., [5, 41, 52, 8]) include facilities for the specification and analysis of infinite behaviours. The description of such behaviours has been traditionally achieved in process theory by means of systems of recursion equations or of variations on Milner's μ-expressions [51, 53]. For example, the recursion equation

$$X \stackrel{\text{def}}{=} (\mathit{send} \cdot \mathit{receive} \cdot X) + \mathit{fail} \qquad (1)$$

describes a system that is willing to perform alternatively the acts of sending and receiving *ad infinitum*, but may *fail* after iterating the sequence $\mathit{send} \cdot \mathit{receive}$ any finite number of times. In order to extend axiomatic verification methods to reason about processes specified by means of recursion equations, several inference rules for proving equalities involving infinite processes have been studied in the literature. (Cf., e.g., rules like unique fixed-point induction in its various flavours [52, 8], the approximation induction principle [11] and ω-induction [39].)

An alternative, purely algebraic, way of introducing infinite behaviours in process algebras is to augment them with variations on the Kleene star operation familiar from the theory of regular algebra—cf., e.g., the papers [32, 9, 10, 29, 24, 22]. Some of these studies, notably [10], 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 [44, 21] rather than by means of systems of recursion equations. For example, using the original, binary version of the Kleene star operation from [44] studied in [10], the system described by the recursion equation (1) can alternatively be denoted by the term $(send \cdot receive)^* fail$, and, as shown in [10], any regular process can be specified in the axiom system ACP_τ [12] with Kleene star using handshake communication. (Interestingly, as already noted by Milner in [51, Sect. 6], not every process defined using finite-state systems of recursion equations can be described, up to bisimulation equivalence, using only regular expressions.)

The possibility of describing infinite behaviours in a purely algebraic syntax has been the motivation for intense research on the use of equational logic to (finitely) axiomatize behavioural equivalences over languages incorporating variations on the Kleene star operation. (Examples of contributions along this line of research may be found in, e.g., [32, 64, 29, 3, 4, 30, 31, 33, 1].) A notable positive result in this direction was obtained by Fokkink and Zantema, who showed in [32] that the finite equational axiom system for the language BPA^* proposed in [10] is indeed complete for bisimulation equivalence. This result is in sharp contrast with a negative one later obtained by Sewell in [64]. Sewell shows that bisimulation equivalence has *no* finite equational axiomatization over the language BPA^*_δ obtained by adding the stopped process δ to the signature of BPA^*. (A discussion of the completeness result by Fokkink and Zantema vis-à-vis Sewell's non-finite axiomatizability result may be found in [2].)

In the light of Fokkink and Zantema's positive result on the mathematical tractability of bisimulation equivalence over BPA^*, a natural question to ask is whether any other (pre)congruence relation in van Glabbeek's linear time/branching time spectrum (cf. [35], where pointers to the original literature may also be found) has a finite (in)equational axiomatization over it. In this paper, we begin to address this question by showing that, unlike bisimulation equivalence, none of the process semantics in the linear time/branching time spectrum lying in between ready simulation and completed traces is finitely based. More precisely, we show that there is a family of (in)equivalences that holds in ready simulation semantics, and *a fortiori* with respect to any behavioural relation that is coarser than it, whose instances cannot all be proven by means of any finite set of (in)equations that is sound in completed trace semantics, which is the coarsest semantics in the linear time/branching time spectrum that is appropriate for the language BPA^*. The family of (in)equivalences that we use in our proof is an adaptation of an axiom schema familiar from the theory of regular algebra (cf., e.g., the equation schema C14.n in [20, page 25]). Consider the equation schema

$$\mathrm{E}.n \quad a^*(a^n) + (a^n)^*(a + \cdots + a^n) \quad = \quad (a^n)^*(a + \cdots + a^n)$$

and the inequation schema

$$\mathrm{I}.n \quad a^*(a^n) \quad \leq \quad (a^n)^*(a + \cdots + a^n)$$

where a is an action, n is a positive integer, and we write a^i for a sequence of a actions of length i. Each of the instances of I.n and E.n is valid in ready simulation semantics.

The crux of the proof of our main result is the construction, for every finite set of (in)equations that is sound in completed trace semantics, of a model in which some of the inequivalences I.n, and some of the equivalences E.n, fail. The model we use for this purpose is based on an adaptation of a beautiful construction due to Conway (cf. [20, Thm. 2, page 105]), who used it to obtain a new proof of a theorem, originally due to Redko [58] (see also [62, Chapter 3 §6] and the references therein), to the effect that equality of regular expressions cannot be axiomatized using a finite number of equations.

The construction of our model relies heavily on the use of prime numbers, as do related arguments presented in, e.g., [20, 26, 47, 64]. Conway's proof of the non-finite axiomatizability of equality of regular expressions is based upon an argument showing that no finite set of regular tautologies can prove all the instances of the aforementioned equality C14.p, for p a prime number. (A generalization of Conway's proof that applies to regular expressions with multiplicities over an arbitrary positive semiring—cf. [25, Chapter 6]—may be found in [47].) In [26], Ésik shows that *iteration theories*, that are a general framework that aims at formalizing the equational logic of iterative processes (cf. the encyclopedic [17] for details), have no finite equational axiomatization. His proof of this result uses the following form of Conway's equation C14.p, for every prime number p:

$$(f^p)^\dagger \;=\; f^\dagger \qquad (f : n \to n + p)$$

where † denotes the iteration operation used in iteration theories. The above identities bear striking resemblance to those employed by Sewell in his proof of the non-existence of finite equational axiomatizations for bisimulation equivalence over regular CCS [52] and over BPA^*_δ [64]. Again, his argument rests on the idea that no finite collection of equations can prove all the equivalences of the form

$$(a^p)^*\delta = a^*\delta$$

where p is a prime number. The similarity amongst these results appears to be more than coincidental. Indeed, Ésik [27] has recently proven that regular languages do have a finite equational axiomatization over iteration theories, i.e., relative to the general set of identities for fixed-point operations. This result, together with the completeness theorems presented in [16, 18], seems to indicate that the deep reason underlying all the aforementioned non-finite axiomatizability results, as well as those presented in this paper, is that the general equational theory of fixed-points is not finitely based. The analysis provided in *op. cit.* also suggests that all the aforementioned negative results have their roots in the original work by Conway and Redko for regular algebra. A related result, whose proof is based on an explicit reduction to the non-existence of a finite equational axiomatization for regular languages, is presented in [23]. In *op. cit.* the authors show that the variety of inversion-free Kleene algebras is not finitely based, thus settling a problem posed by Jónsson in [42].

Our non-finite axiomatizability results apply to the language BPA^* over an arbitrary non-empty set of actions. In particular, we prove that completed trace equivalence over (closed) BPA^* terms is not finitely based even when the set of actions is a singleton. In [62, page 143] Salomaa asked whether the equational theory of (closed) regular expressions over a singleton alphabet is finitely based. Our proof of the non-existence of a finite equational axiomatization of completed traced trace equivalence over (closed) BPA* terms may be easily adapted to the standard language of regular expressions to yield a solution to this question posed by Salomaa. As communicated to us by Salomaa [63], this problem has been open since 1969, the year of publication of [62].

Another semantics that is usually considered in process theory is trace semantics. Trace semantics is, in general, not preserved by sequential composition, and is therefore inappropriate for languages that, like BPA*, include such an operator. However, if the set of actions is a singleton, trace equivalence and preorder *are* preserved by all the operators in the signature of BPA*, and coincide with simulation equivalence and preorder, respectively. In that rather peculiar case, we show that, unlike all the semantics lying in between ready simulation and completed traces, trace equivalence and preorder do have finite, complete equational axiomatizations over *closed* terms in the language BPA*. The reason underlying the existence of these finite axiomatizations is that trace semantics considers *all* the sequences of actions that a process may perform—not only the completed ones. Therefore, if a is the only action, every term containing occurrences of the binary Kleene star operation has the set of all finite sequences of a actions as its set of traces. This means, in particular, that any two terms involving occurrences of the binary Kleene star operation are equivalent in trace semantics. Such terms must have the same denotation in every model for trace equivalence.

We conclude this introduction by providing a brief road-map to the contents of this paper. We begin by introducing the basic notions from process theory that will be needed in the remainder of this study (Sect. 2). The language of Basic Process Algebra with binary Kleene star and its operational semantics are discussed in Sect. 3. We then present the proof of our main result, which is articulated as follows. In Sect. 4 we introduce the family of (in)equivalences on which our argument rests, and reduce the proof of our main result to that of a theorem to the effect that no finite set of (in)equations, that are sound in completed trace semantics, can suffice to prove all of their instances (Thm. 4.5). A proof of Thm. 4.5 is then presented in Sect. 4.1. We begin by studying a normal form for the terms in the language BPA* modulo completed trace equivalence (Sect. 4.1.1). Finally, for every finite set of inequations sound in completed trace semantics, we show how to build a model in which the inequation I.p fails for some prime number p (Sect. 4.1.2). This is sufficient to ensure that the inequality I.p cannot be proven from the inequations under consideration. We then go on to present an analysis of some axiomatic questions on trace semantics over the language BPA*, when the set of actions is a singleton (Sect. 5). More precisely, we show that, unlike all the other process semantics considered in the paper, trace equivalence and preorder do have finite, complete equational axiomatizations over closed terms. We also present evidence that the finite axiom systems that completely characterize trace semantics over closed terms are *not* powerful enough to prove all valid equations between open terms (Propn. 5.5). The paper concludes with a discussion of our results vis-à-vis the completeness theorem by Fokkink and Zantema for bisimulation equivalence—cf. Sect. 6, where the reader will also find more pointers to related literature, and suggestions for further research.

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 [43, 57] that abstracts from the operational semantics of many concurrent calculi.

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

- Proc *is a set of* states*, ranged over by* s*, possibly subscripted or superscripted;*
- Act *is a set of* actions*, ranged over by* a*, possibly subscripted;*
- $\stackrel{a}{\to} \subseteq \mathsf{Proc} \times \mathsf{Proc}$ *is a* transition relation*, for every* $a \in \mathsf{Act}$*. As usual, we shall use the more suggestive notation* $p \stackrel{a}{\to} q$ *in lieu of* $(p, q) \in \stackrel{a}{\to}$*, and write* $s \stackrel{a}{\nrightarrow}$ *iff* $s \stackrel{a}{\to} s'$ *for no state* s'*.*

For $n \geq 0$ and $\varsigma = a_1 \ldots a_n \in \mathsf{Act}^*$, we write $s \stackrel{\varsigma}{\to} s'$ iff there exist states $s_0, \ldots, s_n$ such that $s = s_0 \stackrel{a_1}{\to} s_1 \stackrel{a_2}{\to} \cdots s_{n-1} \stackrel{a_n}{\to} s_n = s'$. 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:

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

We say that s is *deadlocked* iff $\mathsf{initials}(s)$ is empty. An action a will be called a *termination action* of a state s iff $s \stackrel{a}{\to} s'$ for some deadlocked state s'.

2.2 From Ready Simulation to Completed Traces

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 [35, 37] (see also [34, Chapter I]), where 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. In this study, we shall investigate a fragment of the notions of equivalence and preorder from [35]. 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 a is an action:*

  - *if $s_1 \xrightarrow{a} s_1'$, then there is a transition $s_2 \xrightarrow{a} s_2'$ 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 a is an action:*

  - *if $s_1 \stackrel{a}{\nrightarrow}$, then $s_2 \stackrel{a}{\nrightarrow}$.*

- *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 [56, 50] relates two states in a labelled transition system precisely when they have the same branching structure. Simulation (see, e.g., [56]) and ready simulation [14, 48] relax this requirement to different degrees. The following notions, which are all based on decorated versions of traces, are induced by yet further ways of abstracting from the full branching structure of processes.

Definition 2.3 (Decorated Trace Semantics)

- *We say that a sequence of actions ς is a* completed trace *of a state s iff $s \xrightarrow{\varsigma} s'$ for some deadlocked state s'. The set of completed traces of a state s will be denoted by* completed-traces(s). *For states s, s' we write $s \precsim_{CT} s'$ iff the set of completed traces of s is included in that of s'.*

- *We say that a sequence $X_0 a_1 X_1 \ldots a_n X_n$, where $n \geq 0$, the X_i are subsets of Act and the a_i are actions, is a* ready trace *of a state s iff there exist states $s_0, \ldots, s_n$ such that $s = s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \cdots s_{n-1} \xrightarrow{a_n} s_n = s'$, and $\mathsf{initials}(s_i) = X_i$ for every i. The set of ready traces of a state s will be denoted by* ready-traces(s). *For states s, s' we write $s \precsim_{RT} s'$ iff the set of ready traces of s is included in that of s'.*

- *Let X be a subset of Act. For states s, s', we write $s \xrightarrow{X} s'$ iff $s = s'$ and $\mathsf{initials}(s) \cap X = \varnothing$. The relations $\xrightarrow{X}$ will be called the* refusal relations. *The* failure trace *relations $\xrightarrow{\varsigma}$, for $\varsigma \in \left(\mathsf{Act} \cup 2^{\mathsf{Act}}\right)^*$, are defined as the reflexive and transitive closures of the refusal and transition relations. We say that a sequence $\varsigma \in \left(\mathsf{Act} \cup 2^{\mathsf{Act}}\right)^*$, is a* failure trace *of a state s iff $s \xrightarrow{\varsigma} s'$ for some state s'. The set of failure traces of a state s will be denoted by* failure-traces(s). *For states s, s' we write $s \precsim_{FT} s'$ iff the set of failure traces of s is included in that of s'.*

- *For a state s we define:*

$$\begin{array}{rcl} \mathrm{readies}(s) & \triangleq & \left\{(\varsigma, X) \mid \varsigma \in \mathsf{Act}^*, X \subseteq \mathsf{Act} \textit{ and } \exists s' : s \xrightarrow{\varsigma} s' \textit{ and } \mathsf{initials}(s') = X\right\} \\ \mathrm{failures}(s) & \triangleq & \left\{(\varsigma, X) \mid \varsigma \in \mathsf{Act}^*, X \subseteq \mathsf{Act} \textit{ and } \exists s' : s \xrightarrow{\varsigma} s' \textit{ and } \mathsf{initials}(s') \cap X = \varnothing\right\} \end{array}$$

For states s, s' we write $s \precsim_F s'$ iff $\mathrm{failures}(s)$ *is included in* $\mathrm{failures}(s')$*, and $s \precsim_R s'$ iff* $\mathrm{readies}(s)$ *is included in* $\mathrm{readies}(s')$*.*

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

The following result is a standard one in process theory (cf., e.g., [35] where a very informative discussion of the equivalences and preorders in the linear time/branching time spectrum may be found).

Proposition 2.4 *In any transition system,*

$$\begin{array}{ccccccccccc} & & & & \precsim_S & & \precsim_{FT} & & & & \\ & & & \nearrow & & \nearrow & & \searrow & & & \\ \underline{\leftrightarrow} & \rightarrow & \precsim_{RS} & \rightarrow & \precsim_{RT} & & & & \precsim_F & \rightarrow & \precsim_{CT} \\ & & & & & \searrow & & \nearrow & & & \\ & & & & & & \precsim_R & & & & \end{array}$$

where a directed edge from one relation to another means that the source of the edge is included in the target. The same inclusions hold for the kernels of the preorders.

Remark: All the inclusions presented in the previous proposition are proper if the labelled transition system under consideration includes, modulo bisimulation equivalence, the synchronization trees [49] used in the examples presented in [35].

3 BPA with Binary Kleene Star

We begin by presenting the language of Basic Process Algebra (BPA) [11] with binary Kleene star [44] and its operational semantics.

3.1 The Syntax

We assume a non-empty alphabet Act of atomic actions, with typical elements a, b, c, and a countably infinite set Var of process variables, disjoint from Act, with typical elements x, y, z. We shall use α to range over $\mathsf{Act} \cup \mathsf{Var}$.

The language $\mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$ of Basic Process Algebra with binary Kleene star is given by the following BNF grammar:

$$P ::= \alpha \mid P + P \mid P \cdot P \mid P^* P \ .$$

The set of closed terms, i.e., terms that do not contain occurrences of process variables, is denoted by $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$. We shall use P, Q, R, S, T to range over $\mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$. In writing terms over the above syntax, we shall always assume that the operator $\cdot$ binds stronger than $+$, and occurrences of $\cdot$ will often be omitted. With these conventions, the term $PQ + R$ stands for $(P \cdot Q) + R$. 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 $\mathsf{Var}(P)$, and we shall use $\mathsf{StarVar}(P)$ to stand for the set of process variables occurring on the left-hand side of a star in P.

Intuitively, closed terms stand for agents whose behaviour is completely specified, whereas terms containing occurrences of process variables denote agents with partially specified behaviour. For example, an atomic action a stands for a process that can only perform itself in one computational step and terminate in doing so; on the other hand, the term $a + x$ denotes a partially specified process, whose behaviour depends in part on that of the process term that is substituted for the variable x.

Apart from actions and variables, the signature of the language $\mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$ includes the binary operators of alternative composition $+$ and sequential composition $\cdot$ familiar from the theory of Basic Process Algebra [11, 8], and the original binary version of the Kleene star operator introduced in [44]. The term P^*Q stands for a process whose behaviour is specified by the following defining equation:

$$P^*Q \quad = \quad P(P^*Q) + Q \ .$$

A (closed) substitution is a mapping from process variables to (closed) terms in the language $\mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$. 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 the notation $[P/x]$ to denote the substitution mapping the variable x to P, and acting like the identity on all the other variables.

Notation 3.1 *For $I = \{i_1, \ldots, i_n\}$ a finite,* non-empty *index set, we write $\sum_{i \in I} P_i$ for $P_{i_1} + \cdots + P_{i_n}$.*

For a term P and a positive integer n, we write

$$P^n \quad \triangleq \quad \underbrace{P \cdot P \cdots P}_{n\text{-}times}$$

and use $P^{\leq n}$ as a short-hand for $P + P^2 + \cdots + P^n$.

3.2 Operational Semantics

The operational semantics for the language of closed terms $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$ is given by the labelled transition system

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

where the transition relations $\stackrel{a}{\to}$ are the least binary relations over $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act})) \cup \{\checkmark\}$ 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. The special symbol $\checkmark$ represents (successful) termination; therefore the interpretation of the statement $P \xrightarrow{a} \checkmark$ is that the process term P can terminate by performing the atomic action a. Note that $\checkmark$ is the only deadlocked state in the labelled transition system for T(BPA*(Act)).

$$\frac{}{a \xrightarrow{a} \checkmark}$$

$$\frac{P \xrightarrow{a} \checkmark}{P+Q \xrightarrow{a} \checkmark} \qquad \frac{Q \xrightarrow{a} \checkmark}{P+Q \xrightarrow{a} \checkmark} \qquad \frac{P \xrightarrow{a} P'}{P+Q \xrightarrow{a} P'} \qquad \frac{Q \xrightarrow{a} Q'}{P+Q \xrightarrow{a} Q'}$$

$$\frac{P \xrightarrow{a} \checkmark}{P \cdot Q \xrightarrow{a} Q} \qquad \frac{P \xrightarrow{a} P'}{P \cdot Q \xrightarrow{a} P' \cdot Q}$$

$$\frac{P \xrightarrow{a} \checkmark}{P^*Q \xrightarrow{a} P^*Q} \qquad \frac{P \xrightarrow{a} P'}{P^*Q \xrightarrow{a} P'(P^*Q)} \qquad \frac{Q \xrightarrow{a} \checkmark}{P^*Q \xrightarrow{a} \checkmark} \qquad \frac{Q \xrightarrow{a} Q'}{P^*Q \xrightarrow{a} Q'}$$

Table 1: Transition Rules

With the above definitions, the language T(BPA*(Act)) inherits all the notions of equivalence and preorder over processes defined in Sect. 2.2. The following result is standard.

Proposition 3.2 *For $\aleph \in \{RS, CT, RT, FT, F, R\}$, the relations $\lesssim_\aleph$ and $\simeq_\aleph$ are preserved by the operators in the signature of* $\mathbb{T}$(BPA*(Act)). *The same holds for bisimulation equivalence.*

Proof: The congruence result for bisimulation equivalence is well-known (cf., e.g., [32]).

The congruence property for the relations $\lesssim_{RS}$ and $\simeq_{RS}$ is easily established using the fact that, as $\checkmark$ is the only deadlocked state in the labelled transition system for T(BPA*(Act)),

$$\text{if } P \lesssim_{RS} Q \text{ and } P \xrightarrow{a} \checkmark, \text{ then } Q \xrightarrow{a} \checkmark.$$

For each of the relations introduced in Def. 2.3, the set of relevant traces of a composite process can be defined uniformly from those of its components. For example, the set of ready traces of a term $P \in$ T(BPA*(Act)) can be inductively defined thus:

- The set of ready traces of $a \in \mathsf{Act}$ is $\{\{a\}, \{a\}a\emptyset\}$.
- The sequence $X_0 a_1 X_1 \ldots a_n X_n$ is contained in ready-traces$(P+Q)$ iff $X_0 = \mathsf{initials}(P) \cup \mathsf{initials}(Q)$ and
  1. $n = 0$, or
  2. $n > 0$, and $\mathsf{initials}(P) a_1 X_1 \ldots a_n X_n$ is a ready trace of P or $\mathsf{initials}(Q) a_1 X_1 \ldots a_n X_n$ is a ready trace of Q.
- The sequence $X_0 a_1 X_1 \ldots a_n X_n$ is contained in ready-traces(PQ) iff one of the following conditions hold:
  1. $X_0 a_1 X_1 \ldots a_n X_n$ is contained in ready-traces(P) and X_n is non-empty, or
  2. there exists $0 < i \leq n$ such that

(a) $X_0 a_1 X_1 \dots a_i \varnothing$ is a ready trace of P, and
(b) $X_i a_{i+1} X_{i+1} \dots a_n X_n$ is a ready trace of Q.

- The sequence $X_0 a_1 X_1 \dots a_n X_n$ is contained in ready-traces(P^*Q) iff there exist
  1. a non-negative integer k,
  2. ready traces of P of the form
  $$X_{i,0} a_{i,1} X_{i,1} \dots a_{i,n_i} \varnothing \quad (0 < i \leq k, n_i \geq 0)$$
  and
  3. a ready trace of P or Q
  $$X_{k+1,0} a_{k+1,1} X_{k+1,1} \dots a_{k+1,n_{k+1}} X_{k+1,n_{k+1}} \quad (n_{k+1} \geq 0)$$
  with the property that $X_{k+1,n_{k+1}}$ is non-empty if the above is a ready trace of P,

  such that

  $$X_0 a_1 X_1 \dots a_n X_n = X a_{1,1} X_{1,1} \dots a_{1,n_1} X \dots X a_{k,1} X_{k,1} \dots a_{k,n_k} X a_{k+1,1} X_{k+1,1} \dots a_{k+1,n_{k+1}} X_{k+1,n_{k+1}}$$

  where $X = \mathsf{initials}(P) \cup \mathsf{initials}(Q)$.

The result follows immediately from this observation. □

Remark: In [36], van Glabbeek has presented a format of operational rules in Plotkin's SOS style [57] with the property that every operation specified using rules in that format is guaranteed to preserve ready trace equivalence. One of the requirements that such rules have to satisfy is related to the notion of *connectedness*. Connectedness is the smallest equivalence relation over bound variables in a rule, in the sense of [38], such that x and y are connected iff the rule has an antecedent of the form $x \xrightarrow{a} y$. One of the requirements for van Glabbeek's ready trace format is that no two occurrences of variables in the target of a rule are connected in that rule. This requirement is *not* met by the rule

$$\frac{P \xrightarrow{a} P'}{P^*Q \xrightarrow{a} P'(P^*Q)}$$

because the variables P and P' are connected in the above rule, and both occur in the term $P'(P^*Q)$. On the other hand, as shown above, the binary Kleene star operation preserves $\precsim_{RT}$ and, *a fortiori*, ready trace equivalence.

Unlike all the semantics considered in Propn. 3.2, the simulation preorder $\precsim_S$ and its kernel are *not* preserved by the operators of sequential composition and binary Kleene star, at least if the set of actions contains two distinct elements. For example, as ✓ is the least element with respect to the simulation preorder, it follows that $a \precsim_S aa$. However, the reader will find it easy to check that neither $ab \precsim_S (aa)b$ nor $a^*b \precsim_S (aa)^*b$ holds.

Remark: As we shall see in Sect. 5 (cf. Propn. 5.3), if the set of actions is a singleton, then the simulation preorder *is* preserved by the operators in the signature of the language T(BPA*(Act)), and coincides with the preorder induced by trace inclusion [41]. This semantics will have a rather peculiar place in the technical developments of this paper. We refer the impatient reader to Sect. 5 for details.

Following Milner [49], we consider the largest precongruence over T(BPA*(Act)) that is included in the simulation preorder.

Definition 3.3 *A* context *is a term* $R \in \mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$ *containing at most the variable* x. *The relation* $\lesssim_S^c$ *over* $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$ *is defined thus:*

$$P \lesssim_S^c Q \quad \triangleq \quad R[P/x] \lesssim_S R[Q/x], \text{ for every context } R \ .$$

It is easy to see that the relation $\lesssim_S^c$ is indeed the largest precongruence over the language $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$ which is included in $\lesssim_S$. We now proceed to characterize this precongruence explicitly, in the case that the set of actions Act is infinite.

Definition 3.4 *The relation* $\lesssim_{SC}$ *is the largest one over* $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$ *such that* $P \lesssim_{SC} Q$ *iff for every action* a*,*

- *if* $P \xrightarrow{a} P'$, *then there is a transition* $Q \xrightarrow{a} Q'$ *such that* $P' \lesssim_{SC} Q'$*;*
- *if* $P \xrightarrow{a} \checkmark$, *then* $Q \xrightarrow{a} \checkmark$.

Proposition 3.5 *The relation* $\lesssim_{SC}$ *is a precongruence over the language* $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$. *Moreover, if the set of actions* Act *is infinite,* $\lesssim_{SC}$ *coincides with* $\lesssim_S^c$.

Proof: It is easy to check that $\lesssim_{SC}$ is a precongruence over the language $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$ which is included in $\lesssim_S$. It follows that $\lesssim_{SC}$ is included in $\lesssim_S^c$ because the latter is the largest relation with these properties. We now show that the converse inclusion also holds, under the assumption that Act is infinite. To this end, assume that $P \lesssim_S^c Q$. Choose an action b not occurring in P and Q. (Note that, as Act is infinite, such an action may always be found.) By the definition of $\lesssim_S^c$ it follows that $Pb \lesssim_S Qb$. Note now that the relation

$$\mathcal{R} \quad \triangleq \quad \{(R,S) \mid Rb \lesssim_S Sb, \ b \text{ not occurring in } R \text{ and } S\}$$

satisfies the defining clauses of $\lesssim_{SC}$. This is easily checked, using the fact that, as b does not occur in R and S, if $Rb \lesssim_S Sb$ then the termination actions of R are included in those of S. Hence, $\mathcal{R}$ is included in $\lesssim_{SC}$. Since the pair (P,Q) is contained in $\mathcal{R}$, it follows that $P \lesssim_{SC} Q$. □

Remark: If the set of actions is finite, then the preorder $\lesssim_{SC}$ is *strictly* included in $\lesssim_S^c$. To see that this is indeed the case, let us assume that $\mathsf{Act} = \{a_1, \ldots, a_n\}$ for some positive integer n. Consider the term $P \equiv (a_1 + \ldots + a_n)^*(a_1 a_1)$. As $P \xrightarrow{a_i} P$ for every $i \in \{1, \ldots, n\}$, the term P dominates every other term in the language $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$ with respect to the simulation preorder, i.e., $Q \lesssim_S P$ holds for every $Q \in \mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$. We shall now argue that the following inequality holds:

$$a_1 \quad \lesssim_S^c \quad P \ . \tag{2}$$

To this end, we begin by studying the effect of substituting the term P for the variable x in a context R. Let $\cong$ denote the least congruence over $\mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$ that satisfies the following axioms:

$$\begin{aligned} x + y &= y + x \\ (x + y) + z &= x + (y + z) \\ (x + y)z &= xz + yz \\ (xy)z &= x(yz) \\ x(x^*y) + y &= x^*y \end{aligned}$$

We say that a context R is *initially closed* iff $R \cong \sum_{i\in I} R'_i R''_i$, for some finite index set I, and terms $R'_i \in \mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$ and $R''_i \in \mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$. Intuitively, if R is initially closed, then, for every substitution $[Q/x]$, the initial transitions of the term $R[Q/x]$ do not depend upon those of Q.

Lemma 3.6 *Suppose that* $\mathsf{Act} = \{a_1, \ldots, a_n\}$. *Let* $P \equiv (a_1 + \ldots + a_n)^*(a_1 a_1)$. *Then, for every context* $R \in \mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$,

- *either* $R[P/x] \simeq_S P$,
- *or* R *is initially closed.*

Proof: First of all, note that, for every closed term $Q \in \mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$,

- $Q + P \simeq_S P + Q \simeq_S P$,
- $PQ \simeq_S P$ and
- $Q^*P \simeq_S P^*Q \simeq_S P$.

The claim now follows by a straightforward induction on the structure of the context R. □

Using the above lemma, we are now in a position to prove that (2) holds.

Proposition 3.7 *Suppose that* $\mathsf{Act} = \{a_1, \ldots, a_n\}$. *Let* $P \equiv (a_1 + \ldots + a_n)^*(a_1 a_1)$. *Then* $a_1 \sqsubseteq^c_S P$.

Proof: Consider the relation $\mathcal{R}$ defined thus:

$$\begin{array}{rcl}\mathcal{R} & \triangleq & \big(\,\underline{\leftrightarrow} \circ \{(R[a_1/x], R[P/x]) \mid \ R \text{ an initially closed context}\} \circ \underline{\leftrightarrow}\,\big) \\ & & \cup \big\{(Q, P') \mid Q, P' \in \mathrm{T}(\mathrm{BPA}^*(\mathsf{Act})),\ P \sqsubseteq_S P'\big\} \ .\end{array}$$

Note, first of all, that the pair of terms $(R[a_1/x], R[P/x])$ is contained in $\mathcal{R}$ for every context R (Lem. 3.6). To prove the claim, it is therefore sufficient to show that $\mathcal{R}$ is a simulation. To this end, assume that $S\ \mathcal{R}\ T$ and that $S \xrightarrow{a} S'$. We shall now prove that there exists a term T' such that $T \xrightarrow{a} T'$ and $S'\ \mathcal{R}\ T'$.

As $S\ \mathcal{R}\ T$, the definition of $\mathcal{R}$ yields that

1. either $S \,\underline{\leftrightarrow}\, R[a_1/x]$ and $T \,\underline{\leftrightarrow}\, R[P/x]$ for some initially closed context R,
2. or $P \sqsubseteq_S T$.

We proceed with the proof by considering these cases in turn.

1. Assume that $S \,\underline{\leftrightarrow}\, R[a_1/x]$ and $T \,\underline{\leftrightarrow}\, R[P/x]$ for some initially closed context R. As R is initially closed, it follows that $R \cong \sum_{i\in I} R'_i R''_i$, for some terms $R'_i \in \mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$ and $R''_i \in \mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$. Moreover, since the defining axioms of $\cong$ are sound with respect to bisimulation equivalence (cf. Thm. 4.1 below), the terms $R[a_1/x]$ and $\sum_{i\in I} R'_i(R''_i[a_1/x])$ are bisimilar, and so are $R[P/x]$ and $\sum_{i\in I} R'_i(R''_i[P/x])$.

   As $S \xrightarrow{a} S'$ and $S \,\underline{\leftrightarrow}\, R[a_1/x] \,\underline{\leftrightarrow}\, \sum_{i\in I} R'_i(R''_i[a_1/x])$, there exists a term R' such that
   $$\sum_{i\in I} R'_i(R''_i[a_1/x]) \xrightarrow{a} R' \ \text{ and } \ S' \,\underline{\leftrightarrow}\, R' \ .$$

   Since each R'_i is closed, the term R' can only have one of the following two forms:

   - $R' \equiv R''_i[a_1/x]$ for some index $i \in I$ such that $R'_i \xrightarrow{a} \checkmark$, or

- $R' \equiv \hat{R}'_i(R''_i[a_1/x])$ for some index $i \in I$ and closed term $\hat{R}'_i$ such that $R'_i \xrightarrow{a} \hat{R}'_i$.

We continue with the proof by examining the forms R' may take.

- Assume that $R' \equiv R''_i[a_1/x]$ for some index $i \in I$ such that $R'_i \xrightarrow{a} \checkmark$. As $R'_i \xrightarrow{a} \checkmark$, it follows that

$$\sum_{i \in I} R'_i(R''_i[P/x]) \xrightarrow{a} R''_i[P/x] \ .$$

  Moreover, as $T \leftrightarrow R[P/x] \leftrightarrow \sum_{i \in I} R'_i(R''_i[P/x])$, there exists a term T' such that $T \xrightarrow{a} T'$ and $T' \leftrightarrow R''_i[P/x]$. Using Lem. 3.6, we infer that either R''_i is initially closed, or $T' \leftrightarrow R''_i[P/x] \simeq_S P$. In both cases, it follows that $S' \mathcal{R} T'$.

- Assume that $R' \equiv \hat{R}'_i(R''_i[a_1/x])$ for some index $i \in I$ and closed term $\hat{R}'_i$ such that $R'_i \xrightarrow{a} \hat{R}'_i$. As $R'_i \xrightarrow{a} \hat{R}'_i$, it follows that

$$\sum_{i \in I} R'_i(R''_i[P/x]) \xrightarrow{a} \hat{R}'_i(R''_i[P/x]) \ .$$

  Moreover, as $T \leftrightarrow R[P/x] \leftrightarrow \sum_{i \in I} R'_i(R''_i[P/x])$, there exists a term T' such that $T \xrightarrow{a} T'$ and $T' \leftrightarrow \hat{R}'_i(R''_i[P/x])$. The fact that $S' \mathcal{R} T'$ now follows immediately because the context $\hat{R}'_i R''_i$ is initially closed.

2. Assume that $P \lesssim_S T$. As $S \xrightarrow{a} S'$ and $a \in \{a_1, \ldots, a_n\}$, it follows that $P \xrightarrow{a} P$. Therefore $T \xrightarrow{a} T'$ for some term T' such that $P \lesssim_S T'$. By the definition of $\mathcal{R}$, it follows immediately that $S' \mathcal{R} T'$.

We have therefore shown that $\mathcal{R}$ is indeed a simulation relation. We previously noted that the pair of terms $(R[a_1/x], R[P/x])$ is contained in $\mathcal{R}$ for every context R. Hence $a_1 \lesssim_S^c P$ follows. □

However, $a_1 \not\lesssim_{SC} P$, because a_1 can terminate in one step whereas P cannot.

The example discussed in the previous remark is, to our mind, rather peculiar, and reinforces our belief that $\lesssim_{SC}$ is the variation on the simulation preorder that is appropriate for the language $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$. The reader familiar with [10, 32] will also realize that standard bisimulation equivalence over $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$ is the largest symmetric relation included in $\lesssim_{SC}$. For these reasons, in the technical developments to follow we shall only consider the preorder $\lesssim_{SC}$. For later use, we now proceed to study its relationships with the other semantics considered in this paper.

Proposition 3.8 *Over the language* $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$*, the preorder* $\lesssim_{SC}$ *is included in* $\lesssim_{CT}$*, and includes* $\lesssim_{RS}$*. Moreover,* $\lesssim_{SC}$ *coincides with* $\lesssim_{RS}$ *iff* Act *is a singleton.*

Proof: The fact that $\lesssim_{SC}$ is included in $\lesssim_{CT}$, and includes $\lesssim_{RS}$, follows immediately from the definitions of these relations.

We now argue that $\lesssim_{SC}$ coincides with $\lesssim_{RS}$ iff Act is a singleton. To this end, note that the constraint on the set of actions is certainly necessary. In fact, if $a \neq b$, then $a \lesssim_{SC} a + b$, but $a \not\lesssim_{RS} a + b$. To see that it is also sufficient, note that, if the set of actions is a singleton, then $\lesssim_{SC}$ is a ready simulation. □

In the light of Propns. 3.2 and 3.5, for $\aleph \in \{RS, SC, CT, RT, FT, F, R\}$, we can construct the algebra $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))/\simeq_\aleph$ of closed $\mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$-terms modulo $\simeq_\aleph$. That is, for $P, Q \in \mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$,

$$\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))/\simeq_\aleph \;\models P = Q \;\Leftrightarrow\; (\text{for all closed substitutions } \sigma : P\sigma \simeq_\aleph Q\sigma) \;.$$

Each of these algebras has, in fact, the structure of an ordered algebra, in the sense of [15, 39], and, for $P, Q \in \mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$,

$$\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))/\simeq_\aleph \;\models P \leq Q \;\Leftrightarrow\; (\text{for all closed substitutions } \sigma : P\sigma \precsim_\aleph Q\sigma) \;.$$

In both cases, we say that the relevant equation (resp. inequation) is *valid*, or *sound*, with respect to $\simeq_\aleph$ (resp. $\precsim_\aleph$). We shall now proceed to show that none of these (ordered) algebras has a finite (in)equational axiomatization.

Remark: A precongruence relation $\precsim$ over the algebra $\mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$ is *fully invariant*, or *substitutive*, if $P \precsim Q$ implies $P\sigma \precsim Q\sigma$, for every substitution σ. For $\aleph \in \{RS, SC, CT, RT, FT, F, R\}$, we extend the preorder $\precsim_\aleph$ to the whole of $\mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$ thus:

$$P \precsim_\aleph Q \;\;\triangleq\;\; \mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))/\simeq_\aleph \;\models P \leq Q \;.$$

It is easy to see that the precongruence $\precsim_\aleph$ so defined is fully invariant. Similar remarks apply to the congruence relation $\simeq_\aleph$.

4 Non-Finitely Based Process Semantics

In the setting of bisimulation equivalence over the language $\mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$, the following result was first obtained by Fokkink and Zantema [32] for closed terms only, and has later been extended to open terms in [31], where an alternative proof of the original completeness theorem of Fokkink and Zantema is also given.

Theorem 4.1 (Fokkink and Zantema) *The axiom system in Table 2 completely axiomatizes bisimulation equivalence over* $\mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$.

Thus bisimulation equivalence has a finite equational axiomatization over the language $\mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$. In the light of this positive result on the mathematical tractability of bisimulation equivalence over the language of Basic Process Algebra with binary Kleene star, a natural question to ask is whether any other (pre)congruence relation in the linear time/branching time spectrum is finitely (in)equationally axiomatizable over it. We shall now show that, unlike bisimulation equivalence, none of the other preorders and equivalences presented in Sect. 2.2 are finitely based—the one peculiar exception being simulation semantics over closed terms when the set of actions is a singleton (cf. Sect. 5).

Our main aim in the remainder of the paper will be to prove the following negative result.

Theorem 4.2 *None of the preorders* $\precsim_\aleph$ *with* $\aleph \in \{RS, SC, CT, RT, FT, F, R\}$ *has a finite inequational axiomatization over* $\mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$. *Similarly, none of*

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

Table 2: The axiom system for bisimulation equivalence

the equivalence relations they induce has a finite equational axiomatization over $\mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$. *These results also hold if we restrict ourselves to axiomatizations of these relations over closed terms only.*

In order to prove this theorem, we shall show that there is a family of (in)equivalences that holds in ready simulation semantics, and *a fortiori* with respect to any behavioural relation that is coarser than it, whose instances cannot all be proven by means of any finite set of (in)equations that are sound in completed trace semantics. The remainder of this paper will be devoted to a formalization of this proof strategy.

Notation 4.3 *For an axiom system $\mathcal{T}$, we write $\mathcal{T} \vdash P = Q$ (resp. $\mathcal{T} \vdash P \leq Q$) iff the equation $P = Q$ (resp. the inequation $P \leq Q$) is provable from the axiom system $\mathcal{T}$ using the rules of equational (resp. inequational) logic.*

In the sequel, an equation $P = Q$ will sometimes be considered as a shorthand for the pair of inequations $P \leq Q$ and $Q \leq P$.

We write $P =_{\mathrm{AC}} Q$ whenever P and Q are equal modulo commutativity and associativity of $+$, i.e.*, whenever* A1,A2 $\vdash P = Q$. *We say that a term Q is a* summand *of P iff $P \equiv Q$ or* A1,A2 $\vdash P = Q + R$ *for some term R.*

The family of (in)equivalences that we are going to use in our proof of Thm. 4.2 is an adaptation of an axiom schema familiar from the theory of regular algebra (cf., e.g., the equation schema C14.n in [20, page 25]). Consider the equation schema

$$\mathrm{E}.n \quad a^*(a^n) + (a^n)^* a^{\leq n} \quad = \quad (a^n)^* a^{\leq n}$$

and the inequation schema

$$\mathrm{I}.n \quad a^*(a^n) \quad \leq \quad (a^n)^* a^{\leq n}$$

where a is an action and n is a positive integer. Note, first of all, that, for n greater than 1, none of the equivalences E.n is sound with respect to bisimulation equivalence. In fact, as $n > 1$, the term $a^*(a^n) + (a^n)^* a^{\leq n}$ has a sequence of two transitions

$$a^*(a^n) + (a^n)^* a^{\leq n} \xrightarrow{a} a^*(a^n) \xrightarrow{a} a^{n-1}$$

leading to a term whose only behaviour is to reach a deadlocked state after having performed $n-1$ a-actions. As $n > 1$, this behaviour cannot be matched by $(a^n)^* a^{\leq n}$. On the other hand, we have that:

Fact 4.4 *For $\aleph \in \{RS, SC, CT, RT, FT, F, R\}$, the inequations* I.$n$ *are sound with respect to $\precsim_\aleph$, and the equations* E.n *are sound with respect to $\simeq_\aleph$.*

Proof: In the light of Propns. 2.4 and 3.8, it is sufficient to show that each instance of E.n and I.n is sound with respect to $\simeq_{RS}$. To this end, check that the relation

$$\begin{aligned}\mathcal{R} \;\triangleq\; & \left\{\left(a^*(a^n), a^i(a^n)^*(a^{\leq n})\right) \mid n \geq 1, 0 \leq i < n\right\} \cup \left\{\left(a^i, a^j(a^n)^*(a^{\leq n})\right) \mid 0 \leq j < i < n\right\} \\ & \cup \left\{\left((a^n)^* a^{\leq n}, a^*(a^n) + (a^n)^* a^{\leq n}\right) \mid n \geq 1\right\} \\ & \cup \left\{\left(a^*(a^n) + (a^n)^* a^{\leq n}, (a^n)^* a^{\leq n}\right) \mid n \geq 1\right\} \cup \mathcal{I}_{\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act})) \cup \{\checkmark\}}\end{aligned}$$

where $\mathcal{I}_{\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act})) \cup \{\checkmark\}}$ denotes the identity relation over the set $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act})) \cup \{\checkmark\}$, is a ready simulation. □

Thm. 4.2 will follow if we can show that no finite set of equations (resp. inequations) that is sound with respect to $\simeq_{CT}$ (resp. $\precsim_{CT}$) can prove all the equalities E.n (resp. all the inequalities I.n). This is the import of the following theorem.

Theorem 4.5

1. *For every finite set of inequations that is sound with respect to $\precsim_{CT}$, there is a prime number p such that the inequality* I.p *is not provable from the inequations in that set. Moreover, this holds even if we add to the inequations in that set all the axioms of the form* I.n *and* E.n *with n not divisible by p.*

2. *For every finite set of equations that is sound with respect to $\simeq_{CT}$, there is a prime number p such that the equality* E.p *is not provable from the equations in that set. Moreover, this holds even if we add to the equations in that set all the axioms of the form* E.n *with n not divisible by p.*

Using the above result on the power of finite (in)equational axiom systems that are sound in completed trace semantics, we can obtain the following corollary.

Corollary 4.6 *No precongruence (resp. congruence) relation over* $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$ *that is included in $\precsim_{CT}$ (resp. $\simeq_{CT}$) and satisfies* I.n *(resp.* E.n*) for all $n \geq 1$ has a finite inequational (resp. equational) axiomatization.*

Proof: Let $\precsim$ be a precongruence relation over $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$ that is included in $\precsim_{CT}$, and satisfies I.n for all $n \geq 1$. Assume, for the sake of contradiction, that there is a finite set of inequations $\mathcal{E}$ that completely axiomatizes $\precsim$ over the language $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$. As $\precsim$ is included in $\precsim_{CT}$, $\mathcal{E}$ is sound with respect to $\precsim_{CT}$. Since $\precsim$ satisfies the closed inequalities I.n for all $n \geq 1$, and $\mathcal{E}$ is complete for $\precsim$ over $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$, it follows that $\mathcal{E} \vdash$ I.n for all $n \geq 1$. This contradicts Thm. 4.5(1).

A similar reasoning shows that no congruence relation over $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$, that satisfies the proviso of the statement, has a finite equational axiomatization. □

Using Corollary 4.6, it is now a simple matter to show Thm. 4.2. To this end, it is sufficient to note that every preorder in the linear time/branching time spectrum which includes $\lesssim_{RS}$, and is included in $\lesssim_{CT}$, is a precongruence over the language $\mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$ (Propns. 3.2 and 3.5) satisfying all the inequalities I.n for $n \geq 1$ (Fact. 4.4). Therefore every such preorder cannot be finitely inequationally axiomatized (Corollary 4.6). A similar reasoning shows that no congruence relation that lies in between $\simeq_{RT}$ and $\simeq_{CT}$ is finitely equationally axiomatizable over $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$. As (in)equality of closed terms cannot be finitely (in)equationally axiomatized, *a fortiori* neither can (in)equality over open terms in the language $\mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$, that is, none of the (ordered) algebras $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))/\simeq_{\aleph}$ ($\aleph \in \{RS, SC, CT, RT, FT, F, R\}$) is finitely based.

In the light of the above discussion, all that we are left to do to prove Thm. 4.2 is to show Thm. 4.5, and the remainder of this section will be devoted to a presentation of a proof of that result.

4.1 A proof of Thm. 4.5

The proof of Thm. 4.5 we now proceed to present is based on an adaptation of a beautiful argument due to Conway (cf. [20, Thm. 2, page 105]). In *op. cit.*, Conway offers two proofs of a theorem, originally due to Redko [58], to the effect that equality of regular expressions cannot be axiomatized using a finite number of equations. The argument we present below is inspired by the second of those proofs (cf. [20, Pages 105–107]), and is model-theoretic in nature. In order to show Thm. 4.5, for every finite set of (in)equations that are valid in completed trace semantics we shall build a model that does not satisfy all of the instances of I.n and E.n. The construction of the model relies heavily on the use of prime numbers, as do related arguments presented in, e.g., [20, 26, 47, 64].

The proof of Thm. 4.5 will be delivered in two steps. We begin by studying a normal form for the terms in the language $\mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$ modulo completed trace equivalence that will be useful in the proof of this result (Sect. 4.1.1). Finally, for every finite set of inequations sound in completed trace semantics, we show how to build a model in which the inequation I.p fails for some prime number p (Sect. 4.1.2). This is sufficient to ensure that the inequality I.p cannot be proven from the inequations under consideration.

4.1.1 Normal Forms

In what follows, it will be convenient to consider a notion of normal form for terms in completed trace semantics.

Definition 4.7 *A term $P \in \mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$ is* +-free *iff it does not contain occurrences of the +-operation. A term P is in* normal form *iff $P =_{\mathrm{AC}} \sum_{i\in I} P_i$ for some finite, non-empty index set I and +-free terms P_i.*

The length of a term P is inductively defined thus:

$$\begin{aligned}
\mathrm{length}(\alpha) &\triangleq 1 \\
\mathrm{length}(P+Q) &\triangleq \mathrm{length}(P) + \mathrm{length}(Q) \\
\mathrm{length}(PQ) &\triangleq \mathrm{length}(P)\mathrm{length}(Q) \\
\mathrm{length}(P^*Q) &\triangleq \mathrm{length}(Q)\ .
\end{aligned}$$

We shall now show that each $\mathbb{T}(\text{BPA}^*(\mathsf{Act}))$ term is completed trace equivalent to a normal form with the same length. (Note that, as the length of every +-free term is 1, the length of a normal form is the number of summands occurring in it.) To obtain this normalization result, it will be convenient to use the equations in Table 3, which are easily seen to be sound with respect to completed trace equivalence.

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

Table 3: Normalization Equations

Lemma 4.8 *Every $P \in \mathbb{T}(\text{BPA}^*(\mathsf{Act}))$ may be proven equal to a normal form, which has the same length and the same variables as P, using the equations in Table 3 as rewrite rules from left to right.*

Proof: A simple induction on the sum of the lengths of Q and R shows that, for normal forms Q and R,

- QR is provably equal to a normal form whose length is length(Q)length(R), and
- Q^*R is provably equal to a normal form whose length is length(R).

The fact that every term P is provably equal to a normal form, whose length is that of P, then follows by a straightforward structural induction on terms. The normalization process preserves the variables in terms because exactly the same variables occur on both sides of each equation in Table 3. □

Notation 4.9 *For a term P, we use* vars(P) *to denote the total number of occurrences of variables in P, and* weight(P) *(the* weight *of the term P) to stand for* $2^{vars(P)}$length(P).

Example: For every positive integer n, the normal form associated with the term $(a^n)^*a^{\leq n}$ is $\sum_{i=1}^{n}(a^n)^*(a^i)$ which has length, and weight, n. □

The crux of our proof of Thm. 4.5 is the construction, for every prime number p, of an ordered algebra $\mathcal{A}_p$ over the signature of the language $\mathbb{T}(\text{BPA}^*(\mathsf{Act}))$ with the following properties:

P1 For every positive integer n, the inequation I.n and the equation E.n fail in $\mathcal{A}_p$ iff p divides n.

P2 Every inequation $P \leq Q$, that is sound in the algebra $\mathrm{T}(\text{BPA}^*(\mathsf{Act}))/\simeq_{CT}$, where Q is a term whose weight is smaller than p, is valid in $\mathcal{A}_p$.

In fact, if we can construct the algebras $\mathcal{A}_p$ satisfying the above properties, then Thm. 4.5 follows thus:

Proof of Thm. 4.5: We prove the two statements separately.

1. Let $\mathcal{E} = \{P_i \leq Q_i \mid i \in I\}$ be a finite set of inequations that is valid in completed trace semantics. Let m be the supremum of the weights of the terms Q_i. Choose p as the least prime number greater than m. Then the inequations in $\mathcal{E}$ and all the instances of I.n and E.n for n not divisible by p are valid in the algebra $\mathcal{A}_p$ (properties P1 and P2). Moreover, the inequation I.p and the equation E.p fail in $\mathcal{A}_p$ (property P1). As $\mathcal{A}_p$ is a model of the axiom system $\mathcal{E} \cup \{\mathrm{I}.n, \mathrm{E}.n \mid n \bmod p \neq 0\}$ in which I.p and E.p fail, it follows that I.p and E.p are not provable from $\mathcal{E} \cup \{\mathrm{I}.n, \mathrm{E}.n \mid n \bmod p \neq 0\}$.
2. Let $\mathcal{E} = \{P_i = Q_i \mid i \in I\}$ be a finite set of equations that is valid in completed trace semantics. Note that any ordered algebra is a model of $\mathcal{E}$ iff it is a model of the finite collection of inequations $\mathcal{E}_{\leq}$ defined thus:

   $$\mathcal{E}_{\leq} \;\triangleq\; \{P_i \leq Q_i, Q_i \leq P_i \mid i \in I\} \ .$$

   The claim now follows immediately by mimicking the proof of statement 1.

The proof of the theorem is now complete. □

In the light of the previous discussion, in order to complete the proof of Thm. 4.5, we are left to construct, for every prime number p, an ordered algebra $\mathcal{A}_p$ having the properties P1 and P2 stated above.

4.1.2 The Algebra $\mathcal{A}_p$

We shall now proceed to build, for every prime number p, an ordered algebra $\mathcal{A}_p$ with the aforementioned properties. The construction we present mimics the one used by Conway in his proof of the non-finite axiomatizability of the theory of regular languages (cf. [20, pp. 105–107]).

Let a be an arbitrary action. The cyclic group of rank p generated by a can be depicted thus:

$$1 = a^0 \to a^1 \to a^2 \to \cdots \to a^{p-1} \to a^p = 1 \ .$$

The carrier A_p of the algebra $\mathcal{A}_p$ consists of non-empty formal sums of powers of a, together with the formal symbol a^*, i.e.,

$$A_p \;\triangleq\; \left\{\textstyle\sum_{i\in I} a^i \mid \varnothing \subset I \subseteq \{0,\ldots,p-1\}\right\} \cup \{a^*\} \ .$$

In order to give the set A_p enough structure to serve as a suitable semantic domain for the language $\mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$, we need to define the semantic counterparts of the operations in its signature over it. To this end, we map every action in Act to the symbol a, and stipulate that the semantic counterparts of the binary operations are given by the equations in Table 4, where we use the meta-variables e and e' to range over the set A_p. In order to avoid confusion between syntactic and semantic operations, we shall use circled symbols to denote the operations in the algebra $\mathcal{A}_p$. For example, $\oplus$ stands for the semantic counterpart of the $+$ operation of $\mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$. Note that the operations $\oplus$ and $\odot$ are commutative, but $^{\circledast}$ is not. For example,

$$a^{\circledast}1 = a^* \neq a = 1^{\circledast}a \ .$$

An A_p-*environment* is a mapping ρ from process variables to the set A_p. For a term P and an A_p-environment ρ, we shall use $\mathcal{A}_p[\![P]\!]\rho$ to denote the element of A_p that is associated with the term P by the unique homomorphic extension of ρ to $\mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$. If P is a closed term, then $\mathcal{A}_p[\![P]\!]\rho$ is independent of the environment ρ. In that case, we shall simply write $\mathcal{A}_p[\![P]\!]$ for the denotation of P in the algebra $\mathcal{A}_p$.

SUM1	$\sum_{i\in I} a^i \oplus \sum_{j\in J} a^j$	$=$	$\sum_{h\in I\cup J} a^h$
SUM2	$a^* \oplus e$	$=$	a^*
SUM3	$e \oplus a^*$	$=$	a^*
COMP1	$\sum_{i\in I} a^i \odot \sum_{j\in J} a^j$	$=$	$\sum_{h\in\{(i+j) \bmod p \mid (i,j)\in I\times J\}} a^h$
COMP2	$a^* \odot e$	$=$	a^*
COMP3	$e \odot a^*$	$=$	a^*
STAR	$e^{\circledast} e'$	$=$	$\begin{cases} e' & \text{if } e = 1 \\ a^* & \text{otherwise} \end{cases}$

Table 4: The operation of the algebra $\mathcal{A}_p$

We now define a partial ordering on the set A_p thus:

$$e \sqsubseteq_p e' \quad \triangleq \quad e \oplus e' = e' \ .$$

It is not hard to see that $e \sqsubseteq_p e'$ holds iff

- $e' = a^*$, or
- $e = \sum_{i\in I} a^i$, $e' = \sum_{j\in J} a^j$ and I is included in J.

Note, moreover, that the operations in the algebra $\mathcal{A}_p$ are monotonic with respect to the above defined partial ordering. Therefore we have given $\mathcal{A}_p$ the structure of an ordered algebra over the signature of the language $\mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$, in the sense of [15, 39]. It is not hard to see that the equations in Table 3 are sound in the algebra $\mathcal{A}_p$. Hence, if P is a term, and P_{nf} is a normal form for it, then, for every A_p-environment ρ,

$$\mathcal{A}_p[\![P]\!]\rho \ = \ \mathcal{A}_p[\![P_{\mathrm{nf}}]\!]\rho \ .$$

We now proceed to show that the algebra $\mathcal{A}_p$ meets the requirements P1 and P2 that we set out to achieve. To this end, note, first of all, that the inequations I.n fail in $\mathcal{A}_p$ if n is a multiple of p. In fact, in that case,

$$\mathcal{A}_p[\![a^*(a^n)]\!] = a^* \not\sqsubseteq_p \sum_{i=0}^{p-1} a^i = \mathcal{A}_p\left[\!\!\left[(a^n)^* a^{\leq n}\right]\!\!\right] \ .$$

A fortiori, the equations E.n fail in $\mathcal{A}_p$ if n is a multiple of p. On the other hand, if p does not divide n then the equation E.n is valid in $\mathcal{A}_p$, and, *a fortiori*, so is the inequation I.n. This follows because

$$\mathcal{A}_p[\![a^*(a^n)]\!] = a^* = (a^{n \bmod p})^{\circledast}\mathcal{A}_p\left[\!\!\left[a^{\leq n}\right]\!\!\right] = \mathcal{A}_p[\![a^n]\!]^{\circledast}\mathcal{A}_p\left[\!\!\left[a^{\leq n}\right]\!\!\right] = \mathcal{A}_p\left[\!\!\left[(a^n)^* a^{\leq n}\right]\!\!\right]$$

where the second equality from the left holds because of the assumption that n is not divisible by p.

In the light of the above discussion, it follows that the ordered algebra $\mathcal{A}_p$ satisfies the requirement P1 set out on page 19. Note that the aforementioned examples entail that $\mathcal{A}_p$ does not satisfy all the inequations that are valid in the algebra $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))/\simeq_{CT}$. In particular, the inequation I.p, which is valid in $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))/\simeq_{CT}$, fails in it. As remarked in the example on page 19, the weight of (the normal form for) the term $(a^p)^* a^{\leq p}$ is p. We shall now proceed to show that requirement P2 is met by $\mathcal{A}_p$, i.e., that every inequation $P \leq Q$, with Q a term of weight *smaller* than p, which is sound in the algebra $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))/\simeq_{CT}$, is valid in $\mathcal{A}_p$.

As a stepping stone towards the proof of the fact that $\mathcal{A}_p$ meets requirement P2, we shall now argue that the failure of the inequation I.p in the algebra $\mathcal{A}_p$ is paradigmatic. In fact, if $P \leq Q$ is an inequation that is sound in completed trace semantics, and ρ is an A_p-environment such that $\mathcal{A}_p[\![P]\!]\rho \not\sqsubseteq_p \mathcal{A}_p[\![Q]\!]\rho$, then it must be the case that $\mathcal{A}_p[\![P]\!]\rho = a^*$ and $\mathcal{A}_p[\![Q]\!]\rho = \sum_{i=0}^{p-1} a^i$ (cf. Lem. 4.15(2)). This implies that the algebra $\mathcal{A}_p$ is indeed very close to being a model for completed trace semantics. All that we should need to do to turn $\mathcal{A}_p$ into such a model is to identify the elements a^* and $\sum_{i=0}^{p-1} a^i$.

In the proof of a subsequent lemma (Lem. 4.14), we shall make use of some basic notions from number theory. These we now proceed to recall, for the sake of clarity. The interested reader is referred to, e.g., [54] for more details.

Definition 4.10 *Let p and q be integers. If a positive integer m divides the difference $p - q$, we say that p is congruent to q modulo m and write $p \equiv q \pmod{m}$.*

The following classic result pertaining to the solution of congruence equations (cf., e.g., [54, Corollary 2.9]) will find application in the proof of Lem. 4.14(1).

Theorem 4.11 *Let p, q, r be integers with p and q relatively prime, i.e. with $\gcd(p, q) = 1$, and with $q \neq 0$. Then the equation*

$$px \equiv r \pmod{q}$$

has an integer solution x_1. All solutions are given by $x = x_1 + jq$, where $j = 0, \pm 1, \pm 2, \ldots$.

Notation 4.12 *Let P be a term and a an action. We shall use P_a to denote the term obtained from P by replacing every occurrence of an action in P with a.*

Notation 4.13 *Let* $[0 \mapsto p] : \{0, \dots, p-1\} \to \{1, \dots, p\}$ *be the function that maps* 0 *to* p *and acts like the identify on every other integer in its domain. For an* A_p*-environment* ρ*, let* $\bar{\rho} : \mathsf{Var} \to \mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$ *denote the closed substitution which is defined by*

$$\begin{array}{lcll} \bar{\rho}(x) & \triangleq & \sum_{i \in I} a^{[0 \mapsto p](i)} & \textit{if } \rho(x) = \sum_{i \in I} a^i \\ \bar{\rho}(x) & \triangleq & a^* a & \textit{if } \rho(x) = a^* \ . \end{array}$$

Lemma 4.14

1. *Let* $Q, R \in \mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$ *be terms containing only occurrences of the action* $a \in \mathsf{Act}$*. Suppose that* p *is a prime number* p *and* $i \in \{0, \dots, p-1\}$*. Then* a^{np+i} *is a completed trace of* Q^*R *for some* $n \geq 0$ *iff there exists a non-negative integer* m *such that*
   - *either* a^{mp+i} *is a completed trace of* R*,*
   - *or there exists* $j \in \{1, \dots, p-1\}$ *such that* a^{mp+j} *is a completed trace of* Q*.*
2. *Let* $P \in \mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$ *and let* ρ *be an* A_p*-environment. Suppose that* p *is a prime number. Then, for every* $i \in \{0, ..., p-1\}$*,* $a^i \sqsubseteq_p \mathcal{A}_p[\![P]\!]\rho$ *iff* a^{np+i} *is a completed trace of* $P_a\bar{\rho}$ *for some non-negative integer* n*.*

Proof: We prove the two statements separately.

1. Let $Q, R \in \mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$ and $a \in \mathsf{Act}$. Assume that p is a prime number, and that $i \in \{0, \dots, p-1\}$. We establish the two implications separately.
   - 'Only If Implication'. Suppose that a^{np+i} is a completed trace of Q^*R for some $n \geq 0$. We shall prove that, for some $m \geq 0$, a^{mp+i} is a completed trace of R or there exists $j \in \{1, \dots, p-1\}$ such that a^{mp+j} is a completed trace of Q. To this end, note that it is sufficient to show that if every completed trace of Q has length that is a multiple of p, then a^{mp+i} is a completed trace of R for some $m \geq 0$. The simple proof of this fact is left to the reader.
   - 'If Implication'. Suppose that, for some $m \geq 0$,

     A a^{mp+i} is a completed trace of R, or

     B there exists $j \in \{1, \dots, p-1\}$ such that a^{mp+j} is a completed trace of Q.

     We shall prove that a^{np+i} is a completed trace of Q^*R for some $n \geq 0$.

     The only non-trivial case to consider is when condition B above holds. In this case, we proceed as follows. As the set of completed traces of R is non-empty, and R contains only occurrences of action a, we can choose a completed trace a^h of R, for some positive integer h. Then, for every $k \geq 0$, the term Q^*R has a completed trace $a^{k(mp+j)+h}$. We shall now argue that it is possible to choose k such that, for some $n \geq 0$,

     $$k(mp + j) + h = np + i \ .$$

     To this end, note that such a k can be found iff the congruence equation in the unknown k

     $$jk \equiv i - h \pmod{p}$$

     has a non-negative solution. This is an immediate consequence of Thm. 4.11, because j and p are relatively prime.

This completes the proof of statement 1.

2. Let $P \in \mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$, and let p be a prime number. Assume that $i \in \{0, \ldots, p-1\}$. We prove the statement by induction on the structure of P, and proceed by a case analysis on the form P may take.

   - Case: $P \equiv b$.
     In this case, $a^i \sqsubseteq_p \mathcal{A}_p[\![P]\!]\rho$ holds only for $i = 1$, because $\mathcal{A}_p[\![P]\!]\rho = a$. Moreover, $P_a\bar{\rho} \equiv a$, so a is the only completed trace of $P_a\bar{\rho}$.
   - Case: $P \equiv x$.
     In this case, $\mathcal{A}_p[\![P]\!]\rho = \rho(x)$ and $P_a\bar{\rho} = \bar{\rho}(x)$. It follows easily from the definition of $\bar{\rho}$ that $a^i \sqsubseteq_p \rho(x)$ iff $\bar{\rho}(x)$ has a completed trace a^{np+i} for some $n \in \{0, 1\}$.
   - Case: $P \equiv Q + R$.
     In this case, $\mathcal{A}_p[\![P]\!]\rho = \mathcal{A}_p[\![Q]\!]\rho \oplus \mathcal{A}_p[\![R]\!]\rho$. So $a^i \sqsubseteq_p \mathcal{A}_p[\![P]\!]\rho$ iff either $a^i \sqsubseteq_p \mathcal{A}_p[\![Q]\!]\rho$ or $a^i \sqsubseteq_p \mathcal{A}_p[\![R]\!]\rho$. By induction, this is the case iff either $Q_a\bar{\rho}$ or $R_a\bar{\rho}$ has a completed trace of the form a^{np+i} for some non-negative integer n. Finally, this holds iff $P_a\bar{\rho} \equiv Q_a\bar{\rho} + R_a\bar{\rho}$ has a completed trace of the form a^{np+i}.
   - Case: $P \equiv QR$.
     As $\mathcal{A}_p[\![P]\!]\rho = \mathcal{A}_p[\![Q]\!]\rho \odot \mathcal{A}_p[\![R]\!]\rho$, using the definition of $\odot$ it is not hard to see that $a^i \sqsubseteq_p \mathcal{A}_p[\![P]\!]\rho$ iff $a^j \sqsubseteq_p \mathcal{A}_p[\![Q]\!]\rho$ and $a^k \sqsubseteq_p \mathcal{A}_p[\![R]\!]\rho$, for some $j, k \in \{0, \ldots, p-1\}$ with $(j + k) \bmod p = i$. By induction, this holds iff $Q_a\bar{\rho}$ and $R_a\bar{\rho}$ have completed traces a^{lp+j} and a^{mp+k} for non-negative integers l and m, respectively. Finally, as $(j+k) \bmod p = i$, this is the case iff $P_a\bar{\rho} \equiv (Q_a\bar{\rho})(R_a\bar{\rho})$ has a completed trace a^{np+i} for some non-negative integer n.
   - Case: $P \equiv Q^*R$.
     As $\mathcal{A}_p[\![P]\!]\rho = \mathcal{A}_p[\![Q]\!]\rho^{\circledast}\mathcal{A}_p[\![R]\!]\rho$, using the definition of $^{\circledast}$ it is not hard to see that $a^i \sqsubseteq_p \mathcal{A}_p[\![P]\!]\rho$ iff either $a^j \sqsubseteq_p \mathcal{A}_p[\![Q]\!]\rho$ for some $j \in \{1, ..., p-1\}$ or $a^i \sqsubseteq_p \mathcal{A}_p[\![R]\!]\rho$. By induction, this is the case iff either $Q_a\bar{\rho}$ has a completed trace a^{lp+j} for some non-negative integer l or $R_a\bar{\rho}$ has a completed trace a^{mp+i} for non-negative integer m. Finally, as $Q_a\bar{\rho}$ and $R_a\bar{\rho}$ are closed terms containing only occurrences of action a, by statement 1 of the lemma this holds iff $P_a\bar{\rho} \equiv (Q_a\bar{\rho})^*(R_a\bar{\rho})$ has a completed trace a^{np+i} for some non-negative integer n.

   This completes the proof of statement 2.

$\square$

The main use of the above technical result will be in the proof of the following lemma, which will be used repeatedly in the proof of Thm. 4.18 to follow.

Lemma 4.15 *Let $P, Q \in \mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$ and let ρ be an A_p-environment. Suppose that* $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))/\simeq_{CT} \models P \leq Q$. *Then:*

1. *If $\mathcal{A}_p[\![P]\!]\rho = a^*$, then either $\mathcal{A}_p[\![Q]\!]\rho = a^*$ or $\mathcal{A}_p[\![Q]\!]\rho = \sum_{i=0}^{p-1} a^i$.*
2. *If $\mathcal{A}_p[\![P]\!]\rho \not\sqsubseteq_p \mathcal{A}_p[\![Q]\!]\rho$, then $\mathcal{A}_p[\![P]\!]\rho = a^*$ and $\mathcal{A}_p[\![Q]\!]\rho = \sum_{i=0}^{p-1} a^i$.*

Proof: Suppose that $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))/\simeq_{CT} \models P \leq Q$. First of all, note that as the inequation $P \leq Q$ is sound in the algebra $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))/\simeq_{CT}$, then so is $P_a \leq Q_a$. Using this observation, we now prove the two statements of the lemma separately.

1. As $\mathcal{A}_p[\![P]\!]\rho = a^*$, it follows that $P_a\bar{\rho}$ has completed traces of the form $a^{n_i p+i}$ for each $i \in \{0, \ldots, p-1\}$ (Lem. 4.14(2)). Since $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))/\simeq_{CT} \models P_a \leq Q_a$, the set of completed traces of $P_a\bar{\rho}$ is included in that of $Q_a\bar{\rho}$. Therefore $Q_a\bar{\rho}$ has each of the completed traces $a^{n_i p+i}$ ($i \in \{0, \ldots, p-1\}$). Again using Lem. 4.14(2), we obtain that $a^i \sqsubseteq_p \mathcal{A}_p[\![Q]\!]\rho$ for every $i \in \{0, ..., p-1\}$. Hence, either $\mathcal{A}_p[\![Q]\!]\rho = a^*$ or $\mathcal{A}_p[\![Q]\!]\rho = \sum_{i=0}^{p-1} a^i$.

2. Suppose that the A_p-environment ρ is such that $\mathcal{A}_p[\![P]\!]\rho \not\sqsubseteq_p \mathcal{A}_p[\![Q]\!]\rho$. We shall show that $\mathcal{A}_p[\![P]\!]\rho = a^*$ and $\mathcal{A}_p[\![Q]\!]\rho = \sum_{i=0}^{p-1} a^i$.

   We begin by proving that $\mathcal{A}_p[\![P]\!]\rho = a^*$. To this end, assume, towards a contradiction, that $\mathcal{A}_p[\![P]\!]\rho = \sum_{i \in I} a^i$ for some non-empty $I \subseteq \{0, ..., p-1\}$. According to Lem. 4.14(2), $P_a\bar{\rho}$ has a completed trace of the form $a^{n_i p+i}$ for each $i \in I$. Since $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))/\simeq_{CT} \models P_a \leq Q_a$, the term $Q_a\bar{\rho}$ also has a completed trace of the form $a^{n_i p+i}$ for each $i \in I$. By Lem. 4.14(2) it follows that $a^i \sqsubseteq_p \mathcal{A}_p[\![Q]\!]\rho$ for each $i \in I$. Hence, $\mathcal{A}_p[\![P]\!]\rho \sqsubseteq_p \mathcal{A}_p[\![Q]\!]\rho$, which contradicts one of the assumptions of the lemma.

   Thus $\mathcal{A}_p[\![P]\!]\rho = a^*$ must hold. Since $\mathcal{A}_p[\![P]\!]\rho \not\sqsubseteq_p \mathcal{A}_p[\![Q]\!]\rho$, it follows that $\mathcal{A}_p[\![Q]\!]\rho \neq a^*$. Hence, statement 1 of the lemma yields $\mathcal{A}_p[\![Q]\!]\rho = \sum_{i=0}^{p-1} a^i$.

The proof of the lemma is now complete. □

In the proof of the fact that the algebra $\mathcal{A}_p$ satisfies requirement P2 on page 19, we shall make use of some properties of the semantic mapping $\mathcal{A}_p[\![\cdot]\!]$. For ease of reference, these are collected in the following lemma.

Lemma 4.16 *For every $P \in \mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$ and A_p-environment ρ, the following statements hold:*

1. *If $\mathcal{A}_p[\![P]\!]\rho \neq a^*$, then $\rho(x) \neq a^*$ for every variable x contained in $\mathsf{Var}(P)$.*

2. *If $\mathcal{A}_p[\![P]\!]\rho = a^i$ for some $0 \leq i \leq p-1$, then $\rho(x)$ is a power of a for every variable x contained in $\mathsf{Var}(P)$.*

3. *Assume that P is $+$-free, $\mathcal{A}_p[\![P]\!]\rho \neq a^*$, and ρ maps every variable occurring in P to a power of a. Then $\mathcal{A}_p[\![P]\!]\rho$ is a power of a.*

4. *If $\mathcal{A}_p[\![P]\!]\rho \neq a^*$, then $\rho(x)$ is a power of a for every variable x contained in $\mathsf{StarVar}(P)$.*

5. *Assume that $\mathcal{A}_p[\![P]\!]\rho = a^*$, that ρ' coincides with ρ over $\mathsf{StarVar}(P)$, and that if $\rho(x) = a^*$ for an $x \in \mathsf{Var}(P)$, then $\rho'(x) = a^*$. Then $\mathcal{A}_p[\![P]\!]\rho' = a^*$.*

6. *Assume that $\mathcal{A}_p[\![P]\!]\rho \neq a^*$, that ρ' coincides with ρ over $\mathsf{StarVar}(P)$, and that $\rho'(x) \neq a^*$ for $x \in \mathsf{Var}(P)$. Then $\mathcal{A}_p[\![P]\!]\rho' \neq a^*$.*

Proof: All the statements can be shown by induction on the structure of the term P. The details are left to the reader. Here we only remark that the proof for statement 4 uses statement 2 to deal with the case in which P has the form Q^*R for some terms Q and R. In fact, if P has that form and $\mathcal{A}_p[\![P]\!]\rho \neq a^*$, then it must be the case that $\mathcal{A}_p[\![Q]\!]\rho = 1$. Statement 2 then yields that $\rho(x)$ maps each variable in Q to a power of a. □

Lemma 4.17 *Let N denote the number of occurrences of the process variable x in the term Q. Let $[(a+a^2)/x]$ denote the substitution mapping x to $a+a^2$, and acting like the identity on all the other variables. Then the length of $Q[(a+a^2)/x]$ is at most 2^N times the length of Q.*

Proof: Straightforward, by induction on the size of Q. □

We are finally in a position to prove that the algebra $\mathcal{A}_p$ satisfies all the inequations $P \leq Q$, with Q a term of weight smaller than p, that are sound in completed trace semantics. This implies that the algebra $\mathcal{A}_p$ does indeed meet requirement P2.

Theorem 4.18 *If* $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))/\simeq_{CT} \models P \leq Q$ *and* $\mathrm{weight}(Q)$ *is smaller than p, then* $\mathcal{A}_p \models P \leq Q$.

Proof: We shall show that if the inequation $P \leq Q$ is sound in the algebra $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))/\simeq_{CT}$, but fails in $\mathcal{A}_p$, then Q must have weight at least p.

Assume that $P \leq Q$ is sound in $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))/\simeq_{CT}$, but not in $\mathcal{A}_p$. Then there exists an A_p-environment ρ such that

$$\mathcal{A}_p[\![P]\!]\rho \not\sqsubseteq_p \mathcal{A}_p[\![Q]\!]\rho \ .$$

By Lem. 4.15(2), it must be the case that

$$\mathcal{A}_p[\![P]\!]\rho = a^* \not\sqsubseteq_p \sum_{i=0}^{p-1} a^i = \mathcal{A}_p[\![Q]\!]\rho \ .$$

As $\mathcal{A}_p[\![Q]\!]\rho \neq a^*$, it follows that ρ maps no variable in Q to a^* (Lem. 4.16(1)), and that ρ maps every variable in $\mathsf{StarVar}(Q)$ to a power of a (Lem. 4.16(4)). We now proceed with the proof by distinguishing two cases, depending on whether $\mathsf{StarVar}(P)$ is included in $\mathsf{StarVar}(Q)$ or not.

- Case: $\mathsf{StarVar}(P) \subseteq \mathsf{StarVar}(Q)$.

  Consider the A_p-environment ρ' that is defined as follows:

$$\begin{array}{lcll} \rho'(x) & \triangleq & \rho(x) & \text{if } x \in \mathsf{StarVar}(Q) \\ \rho'(x) & \triangleq & \rho(x) & \text{if } \rho(x) = a^* \\ \rho'(x) & \triangleq & 1 & \text{otherwise} \ . \end{array}$$

  Since ρ maps no variable in Q to a^*, the same holds for ρ'. Hence, Lem. 4.16(6) gives that $\mathcal{A}_p[\![Q]\!]\rho' \neq a^*$. Furthermore, since $\mathsf{StarVar}(P)$ is included in $\mathsf{StarVar}(Q)$, ρ' coincides with ρ over $\mathsf{StarVar}(P)$. By construction, if $\rho(x) = a^*$ then $\rho'(x) = a^*$. So, by Lem. 4.16(5), we may infer that $\mathcal{A}_p[\![P]\!]\rho' = a^*$. As the inequation $P \leq Q$ fails in $\mathcal{A}_p$ for the A_p-environment ρ', Lem. 4.15(2) yields that

$$\mathcal{A}_p[\![P]\!]\rho' = a^* \not\sqsubseteq_p \sum_{i=0}^{p-1} a^i = \mathcal{A}_p[\![Q]\!]\rho' \ .$$

  Let Q have normal form $\sum_{k=1}^{m} Q_k$, where each Q_k is $+$-free. Since $\mathcal{A}_p \models Q = \sum_{k=1}^{m} Q_k$, we infer that:

$$\sum_{i=0}^{p-1} a^i = \mathcal{A}_p[\![Q]\!]\rho' = \mathcal{A}_p[\![Q_1]\!]\rho' \oplus \cdots \oplus \mathcal{A}_p[\![Q_m]\!]\rho' \ .$$

By Lem. 4.8, Q and $\sum_{k=1}^{m} Q_k$ have the same variables, and the length of Q is m. As ρ' maps each variable in Q to a power of a, Lem. 4.16(3) now gives that, for every index k, $\mathcal{A}_p[\![Q_k]\!]\rho' = a^j$ for some $0 \leq j \leq p-1$. It follows that $m \geq p$. Thus, $p \leq \text{length}(Q) \leq \text{weight}(Q)$, which was to be shown.

- Case: $\mathsf{StarVar}(P) \not\subseteq \mathsf{StarVar}(Q)$.

  Fix a process variable $x_0 \in \mathsf{StarVar}(P) \backslash \mathsf{StarVar}(Q)$. Consider the A_p-environment ρ' that is defined as follows:

$$\begin{array}{rcll} \rho'(x) & \triangleq & \rho(x) & \text{if } x \in \mathsf{StarVar}(Q) \\ \rho'(x_0) & \triangleq & a + a^2 & \\ \rho'(x) & \triangleq & 1 & \text{otherwise} \ . \end{array}$$

  Since ρ maps no variable in Q to a^*, the same holds for ρ'. Hence, an application of Lem. 4.16(6) gives that $\mathcal{A}_p[\![Q]\!]\rho' \neq a^*$. Furthermore, since $\rho'(x_0)$ is not a power of a, Lem. 4.16(4) gives that $\mathcal{A}_p[\![P]\!]\rho' = a^*$. As the inequation $P \leq Q$ fails in $\mathcal{A}_p$ for the A_p-environment ρ', Lem. 4.15(2) yields that

$$\mathcal{A}_p[\![P]\!]\rho' = a^* \not\sqsubseteq_p \sum_{i=0}^{p-1} a^i = \mathcal{A}_p[\![Q]\!]\rho' \ .$$

  Let $[(a + a^2)/x_0]$ stand for the substitution mapping x_0 to the term $a + a^2$, and acting like the identity on all the other variables. Suppose that the term $Q[(a+a^2)/x_0]$ has normal form $\sum_{k=1}^{m} Q_k$, where each Q_k is $+$-free. By Lem. 4.17, it follows that the length m of $Q[(a+a^2)/x_0]$ is at most $2^{\text{vars}(Q)}\text{length}(Q)$, that is the weight of Q. Consider now the A_p-environment ρ'' that is defined as follows:

$$\begin{array}{rcll} \rho''(x_0) & \triangleq & 1 & \\ \rho''(x) & \triangleq & \rho'(x) & \text{otherwise} \ . \end{array}$$

  By the standard interplay between substitutions and the interpretation mapping $\mathcal{A}_p[\![\cdot]\!]$, and using the fact that $\mathcal{A}_p \models Q[(a + a^2)/x_0] = \sum_{k=1}^{m} Q_k$, we infer that:

$$\sum_{i=0}^{p-1} a^i = \mathcal{A}_p[\![Q]\!]\rho' = \mathcal{A}_p\left[\!\left[Q[(a + a^2)/x_0]\right]\!\right]\rho'' = \mathcal{A}_p[\![Q_1]\!]\rho'' \oplus \cdots \oplus \mathcal{A}_p[\![Q_m]\!]\rho'' \ .$$

  By construction, ρ'' maps each variable in Q to a power of a. As the set of variables occurring in the term $\sum_{k=1}^{m} Q_k$ is $\mathsf{Var}(Q)\backslash\{x_0\}$ (Lem. 4.8), an application of Lem. 4.16(3) now gives that, for every index k, $\mathcal{A}_p[\![Q_k]\!]\rho'' = a^j$ for some $0 \leq j \leq p-1$. It follows that m is greater than, or equal to, p. Thus, $p \leq 2^{\text{vars}(Q)}\text{length}(Q)$, which was to be shown.

This completes the proof of the theorem. □

As an immediate corollary of the above theorem, we obtain the following result.

Corollary 4.19 *Let* $P, Q \in \mathbb{T}(\text{BPA}^*(\mathsf{Act}))$ *be terms of weight smaller than* p*. Suppose that* $\text{T}(\text{BPA}^*(\mathsf{Act}))/\simeq_{CT} \models P = Q$*. Then* $\mathcal{A}_p \models P = Q$*.*

Proof: Let $P = Q$ be an equation consisting of terms of weight smaller than p. Suppose that $\text{T}(\text{BPA}^*(\mathsf{Act}))/\simeq_{CT} \models P = Q$. Then $\text{T}(\text{BPA}^*(\mathsf{Act}))/\simeq_{CT} \models P \leq Q \leq P$. By the previous theorem, $\mathcal{A}_p \models P \leq Q \leq P$. Therefore $\mathcal{A}_p \models P = Q$. □

In the light of the above discussion, we have finally completed the proof of Thm. 4.5, and therefore of Thm. 4.2.

Remark: As pointed out to us by Ésik [28], the proof that we have just completed does in fact yield a stronger statement than that of Thm. 4.2. To see that this is indeed the case, let us define the preorder $\precsim_{TL}$ over $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$ as follows:

$P \precsim_{TL} Q$ iff the set of the lengths of the completed traces of P is included in that of Q.

It is easy to see that, for every $P, Q \in \mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$ and action a,

$$P \precsim_{TL} Q \text{ iff } P_a \precsim_{CT} Q_a \ .$$

It follows that $\precsim_{TL}$ (resp. $\simeq_{TL}$) is a fully invariant precongruence (resp. congruence) for the language $\mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$.

Using the above observations, it is not hard to see that the proof of Thm. 4.2 that we have presented above can in fact be used to show the following result:

Theorem 4.20 *No precongruence (resp. congruence) relation over* $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$ *that is included in* $\precsim_{TL}$ *(resp.* $\simeq_{TL}$*) and satisfies* I.n *(resp.* E.n*) for all* $n \geq 1$ *has a finite inequational (resp. equational) axiomatization. This also holds if we restrict ourselves to axiomatizations of these relations for closed terms only.*

An example of a preorder over $\mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$, which, under the assumption that Act contains at least two elements, lies strictly in between $\precsim_{CT}$ and $\precsim_{TL}$, is the one considered in *commutative regular algebra* (cf., e.g., [59, 62, 20]). This we now proceed to define, for the sake of completeness.

Let L be a set of sequences over the alphabet Act. We write $c(L)$ to denote the set consisting of all those sequences that can be obtained by permuting the actions in some sequence contained in L. We define

$$P \precsim_{CCT} Q \quad \triangleq \quad c(\text{completed-traces}(P)) \subseteq c(\text{completed-traces}(Q)) \ .$$

If Act contains two distinct actions, then $\precsim_{CCT}$ strictly includes $\precsim_{CT}$, and is strictly included in $\precsim_{TL}$. As $\precsim_{CCT}$ is easily seen to be a precongruence, Thm. 4.20 yields the non-existence of a finite inequational axiomatization for it over the language $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$.

The reader familiar with [40, 55] may have noticed the similarities between the notion of commutative regular algebra and the *counter model* for CSP [41] defined *ibidem*. The main difference between the two notions being that the counter model is based upon, not necessarily completed, traces.

Remark: In [62, page 143] Salomaa pointed out that it was an open problem whether the equational theory of (closed) regular expressions over a singleton alphabet is finitely based. Our results show that completed trace equivalence over $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$, and *a fortiori* over $\mathbb{T}(\mathrm{BPA}^*(\mathsf{Act}))$, is not finitely based, even when the set of actions is a singleton. Indeed our proof can be easily adapted, along the lines of the one given by Conway in [20, Thm. 2], to yield the non-existence of a finite equational axiomatization of equality of (closed) regular expressions over a singleton alphabet, thus answering the aforementioned question of Salomaa's. As communicated to us by Salomaa [63], this problem has been open since 1969, the year of publication of [62].

5 The Peculiar Case of Trace Semantics

The reader familiar with van Glabbeek's linear time/branching time spectrum might have noticed the absence of *trace semantics* [41] from the developments presented in the previous section. We shall now proceed to fill this gap by studying some axiomatic questions concerning trace semantics over $\mathbb{T}(\text{BPA}^*(\mathsf{Act}))$. As we shall see, this leads to some rather peculiar results, at least when compared with those that we obtained for the other semantics considered in this paper.

Definition 5.1 *For states s, s' in any labelled transition system, we write $s \lesssim_T s'$ iff the set of traces of s is included in that of s'. The kernel of the preorder $\lesssim_T$ will be denoted by $\simeq_T$.*

Note that the set of traces of a state s in any labelled transition system is prefix closed, unlike that of its completed traces.

In general, completed trace semantics and trace semantics are incomparable. For example, the $\mathrm{T}(\text{BPA}^*(\mathsf{Act}))$ terms a and aa have disjoint sets of completed traces, but the set of traces of a is included in that of aa. On the other hand, the processes a^ω and b^ω specified by the following recursion equations

$$\begin{aligned} a^\omega &\stackrel{\text{def}}{=} a \cdot a^\omega \\ b^\omega &\stackrel{\text{def}}{=} b \cdot b^\omega \end{aligned}$$

have no completed trace, but disjoint sets of non-empty traces. However, if the labelled transition system under consideration is *normed*, in the sense of [6], then the set of traces of every state s is obtained as the prefix closure of its set of completed traces. This is because every state in a normed transition system has at least one completed trace, and therefore each of its traces is the prefix of a completed one. As the labelled transition system giving the operational semantics to the language $\mathrm{T}(\text{BPA}^*(\mathsf{Act}))$ is normed (cf., e.g., [32]), by the above discussion we obtain that:

Fact 5.2 *The relation $\lesssim_{CT}$ (resp. $\simeq_{CT}$) is strictly included in $\lesssim_T$ (resp. $\simeq_T$) over the language* $\mathrm{T}(\text{BPA}^*(\mathsf{Act}))$.

In general, trace semantics is not appropriate for languages that, like $\mathrm{T}(\text{BPA}^*(\mathsf{Act}))$, include a sequential composition operator. In fact, if the set of actions Act contains at least two distinct actions, then neither $\lesssim_T$ nor $\simeq_T$ are preserved by sequential composition. As an example, consider the $\mathrm{T}(\text{BPA}^*(\mathsf{Act}))$ terms a and aa. We have already remarked that $a \lesssim_T aa$. However, if b is an action that is different from a, then $ab \not\lesssim_T aab$. For this reason, in the previous sections we confined our attention to semantics that are included in completed trace semantics.

In contrast to the general situation depicted above, in the, admittedly rather uninteresting, case in which the set of actions Act is a singleton, we observe the following fact.

Proposition 5.3 *Assume that the set of actions* Act *is a singleton. Then:*

1. *The relations* $\precsim_T$ *and* $\simeq_T$ *are preserved by all the operations in the signature of the language* $\mathbb{T}(\text{BPA}^*(\mathsf{Act}))$.

2. *The simulation preorder* $\precsim_S$ *coincides with* $\precsim_T$.

Proof: Assume that a is the only action contained in Act. We prove the two statements separately.

1. As the set of traces of a term of the form $P+Q$ is the union of those of P and Q, it follows immediately that $\precsim_T$ is preserved by summation. (Indeed, this holds regardless of the cardinality of the set of actions.) We shall now prove that $\precsim_T$ is preserved by sequential composition and binary Kleene star.

   Suppose that P, Q, R, S are terms such that $P \precsim_T Q$ and $R \precsim_T S$. We shall show that $PR \precsim_T QS$ and that $P^*R \simeq_T Q^*S$.

   - We prove, first of all, that $PR \precsim_T QS$.
     Assume that a^n is a trace of the term PR, for some non-negative integer n. We now proceed to argue that a^n is also a trace of the term QS.
     If a^n is a trace of Q, then it is also a trace of QS, in which case we are done. Thus we may assume that a^n is not a trace of Q. Note that, since $P \precsim_T Q$, a^n is not a trace of P either. As a consequence of these assumptions, we infer that:

     (a) $n = h+k$ for two positive integers h and k such that a^h is a *completed* trace of P, and a^k is a trace of R; and

     (b) every trace a^j of Q has length smaller than n.

     In this case, we argue as follows. As the length of the traces of Q is bounded from above by n, we can choose the longest such trace a^j. This trace is a completed trace of Q. As a^h is a trace of Q ($P \precsim_T Q$), and a^j is the longest such trace, it follows that $h \leq j$. Since a^j is a completed trace of Q, and a^k is a trace of S ($R \precsim_T S$), a^{j+k} is a trace of QS. Finally, $j+k \geq h+k = n$, so then a^n is a trace of QS.

   - To prove that $P^*R \simeq_T Q^*S$, it is sufficient to note that every process containing occurrences of the binary Kleene star operator has the set of all finite sequences of a actions as its set of traces.

2. The fact that $\precsim_S$ is included in $\precsim_T$ is a simple consequence of the definitions of these relations. To see that the converse also holds, under the assumption that the only action is a, it is sufficient to check that the relation:

   $$\mathcal{R} \quad \triangleq \quad \left\{(P,Q) \mid P \precsim_T Q\right\} \cup \{(\checkmark, P) \mid P \in \mathrm{T}(\text{BPA}^*(\mathsf{Act}))\}$$

   is a simulation. This is an easy consequence of the fact that, for terms P, Q over action a, $P \precsim_T Q$ iff

   - either Q has an infinite a-computation, i.e., for some terms $Q_1, Q_2, \ldots$,

     $$Q \xrightarrow{a} Q_1 \xrightarrow{a} Q_2 \xrightarrow{a} \cdots$$

   - or the length of the longest completed trace of P is less than, or equal to, that of the longest completed trace of Q.

The proof is now complete. □

In the light of the above congruence result, and of the non-finite axiomatizability results presented in the main body of the paper, it is natural to wonder whether the trace (pre)congruence has a finite (in)equational axiomatization over the language of closed terms T(BPA*(Act)) over a singleton set of actions. We recall that our previous negative results pertaining to the non-finite axiomatizability of several process semantics over T(BPA*(Act)) apply for every non-empty set of actions. We shall now proceed to show that, in contrast to the situation summarized in Thm. 4.2, trace (pre)congruence can be finitely (in)equationally axiomatized over the set of *closed* terms over a singleton action set.

TE1	$x + (y^*z)$	$=$	a^*a
TE2	$x + xy$	$=$	xy
TE3	xy	$=$	yx

Table 5: Characteristic equations for trace equivalence ($\mathsf{Act} = \{a\}$)

Consider the axiom system $\mathcal{E}_T$ consisting of the equations A1–A5 in Table 2 together with the axioms in Table 5. It is not hard to see that the equations in $\mathcal{E}_T$ are sound with respect to trace equivalence over T(BPA*(Act)). The only non-standard equations in $\mathcal{E}_T$ are TE1 and TE3, whose soundness depends crucially upon the assumption that the only action is a.

Theorem 5.4 *Let* $\mathsf{Act} = \{a\}$ *and* $P, Q \in$ T(BPA*(Act)). *Then* $P \simeq_T Q$ *iff* $\mathcal{E}_T \vdash P = Q$. *Moreover* $P \lesssim_T Q$ *iff* $P \leq Q$ *can be proven from the equations in* $\mathcal{E}_T$ *together with the inequation*

(3) $$x \leq x + y \ .$$

Proof: (Sketch.) The soundness of the equations in $\mathcal{E}_T$ and of inequation (3) is easy to check. We shall now argue for the completeness, over closed terms, of the proposed axiomatizations.

A trace normal form is either a^*a or a term of the form $\sum_{i=1}^{n} a^i$ for some positive integer n. A simple structural induction on terms gives that every term in the language T(BPA*(Act)) can be proven equal to a trace normal form. The proof of this fact makes use of all of the equations in $\mathcal{E}_T$, together with the following derived laws:

$$x(y + z) = xy + xz$$
$$(a^*a)x = a^*a \ .$$

The completeness of the axiomatization for trace equivalence now follows immediately because two trace normal forms are equal iff they are identical, modulo commutativity and associativity of summation.

To establish the completeness of the axiomatization for the trace precongruence, note that $P \lesssim_T Q$ iff $P + Q \simeq_T Q$. As $\mathcal{E}_T$ is complete for trace equivalence, it follows that, if $P \lesssim_T Q$, then $\mathcal{E}_T \vdash P + Q = Q$. Now, the inequality $P \leq Q$ can be proven using (3) and transitivity. □

It is interesting to note that the above axiomatizations are *not* complete for open terms in the language $\mathbb{T}$(BPA*(Act)). For example, when the set of actions is a singleton, the equations

(4) $$a + x = x$$

(5) $$xx + yy = xx + yy + xy$$

are sound with respect to trace semantics over $\mathbb{T}(\text{BPA}^*(\mathsf{Act}))$. However, we shall now show that (4) and (5) are *not* provable from the axiom system $\mathcal{E}_T \cup \{(3)\}$.

Proposition 5.5 *Equations* (4) *and* (5) *are not deducible from the axiom systems* $\mathcal{E}_T \cup \{(3)\}$.

Proof: We build a model for the axiom system $\mathcal{E}_T \cup \{(3)\}$ in which (4) and (5) fail. The carrier of the model $\mathcal{M}$ is the poset depicted below:

$$\begin{array}{ccccc} & & 2 & & \\ & & \uparrow & & \\ & & \mathbf{2} & & \\ & \nearrow & & \nwarrow & \\ 1 & & & & \mathbf{1} \end{array}$$

The operators in the signature of the language $\text{T}(\text{BPA}^*(\mathsf{Act}))$ are defined over $\mathcal{M}$ as follows ($e, e' \in \{1, 2, \mathbf{1}, \mathbf{2}\}$):

$$\begin{array}{rcl} a_{\mathcal{M}} & \triangleq & 1 \\ e +_{\mathcal{M}} e' & \triangleq & \sup(e, e') \\ e^{*_{\mathcal{M}}} e' & \triangleq & 2 \\ e \cdot_{\mathcal{M}} e' & \triangleq & \begin{cases} 1 & \text{if } e = e' = 1 \\ \mathbf{1} & \text{if } e = e' = \mathbf{1} \\ 2 & \text{otherwise} \end{cases} \end{array}$$

(Note that the operations so defined are monotonic.) The reader will have no difficulty in verifying that the resulting ordered algebra is a model of $\mathcal{E}_T \cup \{(3)\}$. However (4) and (5) fail in $\mathcal{M}$. Indeed (4) fails in $\mathcal{M}$ because, letting ρ denote an environment mapping x to $\mathbf{1}$,

$$\mathcal{M}[\![x]\!]\rho = \mathbf{1} \neq \mathbf{2} = \mathcal{M}[\![a + x]\!]\rho \ .$$

To see that (5) also fails, let ρ' be an $\mathcal{M}$-environment mapping x to 1, and y to $\mathbf{1}$. Then,

$$\mathcal{M}[\![xx + yy]\!]\rho' = \mathbf{2} \neq 2 = \mathcal{M}[\![xx + yy + xy]\!]\rho' \ .$$

The proof is now complete. $\square$

We leave it as an open question whether there exists a finite (in)equational axiomatization of the (ordered) algebra $\mathbb{T}(\text{BPA}^*(\mathsf{Act}))/ \simeq_T$. This problem is closely related to that of finding a finite ω-complete axiomatization of the algebra of the positive integers with operations of summation and maximum. To the best of our knowledge, this problem is, surprisingly, still awaiting a solution.

6 Concluding Remarks

In this paper we have shown that none of the process semantics that lie in between ready simulation and completed traces are finitely based over the language BPA*. This result is in sharp contrast with a theorem by Fokkink and Zantema [32] to the effect that bisimulation equivalence has a finite equational axiomatization over BPA*, and the reader might wonder why bisimulation equivalence is finitely based whereas none of the process semantics considered in this paper is. (The only peculiar exception being trace semantics over closed terms when the set of actions is a singleton.) We shall now present our interpretation of the dichotomy between bisimulation and the process semantics considered in this paper.

For every process term $P \in \mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$, let $\mathrm{Loops}(P)$ denote the collection of the completed traces of the sub-terms of P that occur on the left-hand side of a star. Intuitively, $\mathrm{Loops}(P)$ is the set of the sequences of actions labelling the loops in the finite automaton that is associated with P by the operational semantics for $\mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$. We shall now prove a result to the effect that two BPA^* terms can only be bisimilar if they have loops of the same length.

Proposition 6.1 *Let* $P, Q \in \mathrm{T}(\mathrm{BPA}^*(\mathsf{Act}))$*. If* $P \leftrightarrow Q$*, then* $\mathrm{Loops}(P) = \mathrm{Loops}(Q)$.

Proof: As the equations in Table 2 are complete with respect to bisimulation equivalence, it is sufficient to show that if the equation $P = Q$ is deducible from those in Table 2, then $\mathrm{Loops}(P) = \mathrm{Loops}(Q)$. This is easily verified because the statement holds for the equations in the aforementioned table, and is preserved by the rules of equational deduction. □

As witnessed by the family of equivalences E.n, Propn. 6.1 does not hold for any of the semantics considered in this paper, and its loss entails that any complete equational axiom system for any of these semantics should be powerful enough to equate terms with loops of different lengths. Indeed, the import of Thm. 4.5 is that no finite set of (in)equations can prove all the equivalences between terms whose loops have prime length. This is the same reason that leads to the non-existence of a finite equational axiom system for bisimulation equivalence over BPA^*_δ [64], and that underlies the negative results in, e.g., [58, 62, 20, 26, 47].

The results of this paper have shown that all the semantics considered in [35] are not finitely based over the language BPA^*, with the exception of *2-nested simulation equivalence* [38] and *possible-futures equivalence* [60]. Establishing whether these equivalences are finitely based or not is a possible avenue for further research along the lines of this paper. Let us just remark here that, at least to the best of our knowledge, no complete axiomatization for 2-nested simulation equivalence is known even over the basic syntax for synchronization trees. Moreover, possible-futures equivalence is not preserved by sequential composition and binary Kleene star, and therefore this semantics cannot be readily used for a language like BPA^*.

Having established that none of the process semantics considered in this paper has a finite equational axiomatization over BPA^*, a natural topic for further study is the search for effectively presented, infinite equational axiomatizations for them. This is most likely to be a difficult problem, as witnessed by the corresponding developments in the theory of regular expressions. These we now briefly recall for the sake of historical completeness. A theorem of Redko's, whose proof was simplified and corrected by Pilling [20, Chapter 11], gives an infinite, complete system of identities for commutative regular expressions [59]. An infinite equational axiomatization of the theory of regular expressions over a singleton alphabet was given by Redko in [58] (cf. also [20, Chapter

4]). (Variations on the aforementioned results of Redko's that apply to regular expressions over a singleton alphabet with multiplicities over the tropical semiring may be found in [19].) The construction of a complete equational axiomatization for regular expressions over an arbitrary alphabet was addressed by Conway in his seminal monograph [20]. *Ibidem* Conway proposed three conjectures, whose solution would yield the desired complete set of equations. It took many years, and Krob's landmark paper [46], to settle two of these conjectures of Conway's, and to obtain the first complete equational axiom system for the theory of regular expressions. An alternative equational axiomatization for regular expressions, developed within the framework of iteration theories [17], may be found in [16]. Finite implicational proof systems for regular expressions have been developed by, e.g., Salomaa [61, 62] and Kozen [45]. (The interested reader is invited to consult [46, Sect. 15] for a thorough discussion of implicational proof systems for regular languages.) Modifications of these proof systems to yield complete axiom systems based on conditional equations for the process semantics considered in this paper over BPA* are an interesting topic for future research.

Acknowledgements: We should like to thank Zoltán Ésik for his encouragement to pursue the research reported in this paper, and for many insightful comments and pointers to related literature. We are also indebted to Gheorghe Ştefănescu for alerting us to the relevance of Ésik's work on axiomatizations of iteration theories, and to Daniel Krob for providing us with information on his work on axiomatic questions for regular expressions.

References

[1] L. Aceto and W. J. Fokkink, *An equational axiomatization for multi-exit iteration*, Research Report RS-96-22, BRICS (Basic Research in Computer Science, Centre of the Danish National Research Foundation), Department of Computer Science, Aalborg University, June 1996. Available by anonymous ftp at the address `ftp.brics.aau.dk` in the directory `pub/BRICS/RS/96/22`.

[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 at the address `ftp.brics.aau.dk` in the directory `pub/BRICS/RS/95/28`.

[3] L. Aceto and A. Ingólfsdóttir, *An equational axiomatization of observation congruence for prefix iteration*, in Wirsing and Nivat [65], pp. 195–209. Full version available as BRICS Report RS-95-5, January 1995.

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

[5] D. Austry and G. Boudol, *Algèbre de processus et synchronisations*, Theoretical Comput. Sci., 30 (1984), pp. 91–131.

[6] J. Baeten, J. Bergstra, and J. Klop, *Decidability of bisimulation equivalence for processes generating context-free languages*, J. Assoc. Comput. Mach., 40 (1993), pp. 653–682.

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

[8] J. Baeten and W. Weijland, *Process Algebra*, Cambridge Tracts in Theoretical Computer Science 18, Cambridge University Press, 1990.

[9] J. Bergstra, I. Bethke, and A. Ponse, *Process algebra with iteration*, Report CS-R9314, Programming Research Group, University of Amsterdam, 1993. Available by anonymous ftp at the address `fwi.uva.nl` as file `pub/programming-research/reports/1993/P9314.ps.Z`.

[10] ——, *Process algebra with iteration and nesting*, Computer Journal, 37 (1994), pp. 243–258.

[11] J. Bergstra and J. Klop, *Fixed point semantics in process algebras*, Report IW 206, Mathematisch Centrum, Amsterdam, 1982.

[12] ——, *Algebra of communicating processes with abstraction*, Theoretical Comput. Sci., 37 (1985), pp. 77–121.

[13] E. Best, ed., *Proceedings CONCUR 93,* Hildesheim, Germany, vol. 715 of Lecture Notes in Computer Science, Springer-Verlag, 1993.

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

[15] S. L. Bloom, *Varieties of ordered algebras*, J. Comput. System Sci., 13 (1976), pp. 200–212.

[16] S. L. Bloom and Z. Ésik, *Equational axioms for regular sets*, Mathematical Structures in Computer Science, 3 (1993), pp. 1–24.

[17] ——, *Iteration Theories: The Equational Logic of Iterative Processes*, EATCS Monographs on Theoretical Computer Science (W. Brauer, G. Rozenberg and A. Salomaa eds.), Springer-Verlag, 1993.

[18] ——, *Iteration algebras of finite state process behaviours*, Jan. 1994. Unpublished manuscript. Available by anonymous ftp from `menger.eecs.stevens-tech.edu` in the directory `/pub/bloom/papers/processes`.

[19] A. Bonnier-Rigny and D. Krob, *A complete system of identities for one-letter rational expressions with multiplicities in the tropical semiring*, Theoretical Comput. Sci., 134 (1994), pp. 27–50.

[20] J. H. Conway, *Regular algebra and finite machines*, Mathematics Series (R. Brown and J. De Wet eds.), Chapman and Hall, London, United Kingdom, 1971.

[21] I. Copi, C. Elgot, and J. Wright, *Realization of events by logical nets*, J. Assoc. Comput. Mach., 5 (1958), pp. 181–196.

[22] F. Corradini, R. De Nicola, and A. Labella, *Fully abstract models for nondeterministic Kleene algebras (extended abstract)*, in Proceedings CONCUR 94, Philadelphia, PA, USA, I. Lee and S. Smolka, eds., vol. 962 of Lecture Notes in Computer Science, Springer-Verlag, 1995, pp. 130–144.

[23] S. Crvenković and R. Madarász, *On Kleene algebras*, Theoretical Comput. Sci., 108 (1993), pp. 17–24.

[24] R. De Nicola and A. Labella, *A completeness theorem for nondeterministic Kleene algebras*, in Proceedings of MFCS '94, vol. 841 of Lecture Notes in Computer Science, Springer-Verlag, 1994, pp. 536–545.

[25] S. Eilenberg, *Automata, Languages, and Machines*, vol. 59–A of Pure and Applied Mathematics, Academic Press, New York and London, 1974.

[26] Z. Ésik, *Independence of the equational axioms for iteration theories*, J. Comput. System Sci., 36 (1988), pp. 66–76.

[27] ——, *Group axioms for iteration*, 1995. Unpublished manuscript.

[28] ——, *Personal communication*, May 1996.

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

[30] ——, *A complete axiomatization for prefix iteration in branching bisimulation*, Fundamenta Informaticae, 26 (1996), pp. 103–113.

[31] ——, *On the completeness of the equations for the Kleene star in bisimulation*, in Wirsing and Nivat [65], pp. 180–194. A full version of the paper is available by anonymous ftp from `phil.ruu.nl` as `logic/PREPRINTS/preprint141.ps`.

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

[33] ——, *Prefix iteration in basic process algebra: applying termination techniques*, in Proceedings ACP 95, Eindhoven, A. Ponse, C. Verhoef, and B. v. Vlijmen, eds., vol. Report CS-95-14, Eindhoven University of Technology, 1995, pp. 139–156.

[34] R. J. v. Glabbeek, *Comparative Concurrency Semantics and Refinement of Actions*, PhD thesis, Free University, Amsterdam, 1990.

[35] ——, *The linear time – branching time spectrum*, in Baeten and Klop [7], pp. 278–297.

[36] ——, *Full abstraction in structural operational semantics (extended abstract)*, in Proceedings of the Third International Conference on Algebraic Methodology and Software Technology (AMAST '93), Twente, The Netherlands, M. Nivat, C. Rattray, T. Rus, and G. Scollo, eds., Workshops in Computing, Springer-Verlag, June 1993, pp. 77–84.

[37] ——, *The linear time – branching time spectrum II: the semantics of sequential processes with silent moves*, in Best [13], pp. 66–81.

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

[39] M. Hennessy, *Algebraic Theory of Processes*, MIT Press, Cambridge, Massachusetts, 1988.

[40] C. Hoare, *Specifications, programs and implementations*, Technical Monograph PRG–29, Programming Research Group, Oxford University, 1982.

[41] ——, *Communicating Sequential Processes*, Prentice-Hall International, Englewood Cliffs, 1985.

[42] B. Jónsson, *The theory of binary relations*, in Algebraic Logic, Colloquia Mathematica Societatis János Bolyai, vol. 54, North-Holland, Amsterdam, 1988, pp. 245–292.

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

[44] 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.

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

[46] D. Krob, *Complete systems of B-rational identities*, Theoretical Comput. Sci., 89 (1991), pp. 207–343.

[47] ——, *Models of a K-rational identity system*, J. Comput. System Sci., 45 (1992), pp. 396–434.

[48] K. G. Larsen and A. Skou, *Bisimulation through probabilistic testing*, Information and Computation, 94 (1991), pp. 1–28.

[49] R. Milner, *A Calculus of Communicating Systems*, vol. 92 of Lecture Notes in Computer Science, Springer-Verlag, 1980.

[50] ——, *Calculi for synchrony and asynchrony*, Theoretical Comput. Sci., 25 (1983), pp. 267–310.

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

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

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

[54] I. Niven and H. Zuckerman, *An introduction to the theory of numbers (2nd edition)*, John Wiley & Sons, 1960.

[55] E.-R. Olderog and C. Hoare, *Specification-oriented semantics for communicating processes*, Acta Inf., 23 (1986), pp. 9–66.

[56] 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.

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

[58] V. Redko, *On defining relations for the algebra of regular events*, Ukrainskii Matematicheskii Zhurnal, 16 (1964), pp. 120–126. In Russian.

[59] ——, *On the algebra of commutative events*, Ukrainskii Matematicheskii Zhurnal, 16 (1964), pp. 185–195. In Russian.

[60] W. Rounds and S. Brookes, *Possible futures, acceptances, refusals and communicating processes*, in 22^{th} Annual Symposium on Foundations of Computer Science, Nashville, Tennessee, New York, 1981, IEEE, pp. 140–149.

[61] A. Salomaa, *Two complete axiom systems for the algebra of regular events*, J. Assoc. Comput. Mach., 13 (1966), pp. 158–169.

[62] ——, *Theory of Automata*, vol. 100 of International Series of Monographs in Pure and Applied Mathematics (I.N. Sneddon and M. Stark eds.), Pergamon Press, Oxford, 1969.

[63] ——, *Personal communication*, June 1996.

[64] 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.

[65] M. Wirsing and M. Nivat, eds., *Algebraic Methodology and Software Technology: 5th International Conference, AMAST '96, Munich, Germany*, vol. 1101 of Lecture Notes in Computer Science, Springer-Verlag, July 1996.