A Finite Basis for Failure Semantics

Wan Fokkink[1,2] and Sumit Nain[3]

[1] Vrije Universiteit Amsterdam, Department of Theoretical Computer Science, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands, `wanf@cs.vu.nl`
[2] CWI, Department of Software Engineering, PO Box 94079, 1090 GB Amsterdam, The Netherlands
[3] BRICS, Department of Computer Science, Aalborg University, Fr. Bajersvej 7E, 9220 Aalborg Ø, Denmark, `nain@cs.aau.dk`

Abstract. We present a finite ω-complete axiomatization for the process algebra BCCSP modulo failure semantics, in case of a finite alphabet. This solves an open question by Groote [12].

1 Introduction

Labeled transition systems model processes by explicitly describing their states and their transitions from state to state, together with the actions that produce these transitions. Several notions of behavioral equivalence have been proposed, with the aim to identify those states of labeled transition systems that afford the same observations.

Van Glabbeek [10, 11] presented the linear time - branching time spectrum of behavioral equivalences for finitely branching, concrete, sequential processes. In this paper we focus on *failure semantics* [6, 7], which distinguishes a process by its "failure pairs", which consist of a finite (partial) trace together with a set of actions that cannot be executed at the ultimate state of this trace. Other semantics in the spectrum are based on (bi)simulation and on (decorated) traces. Figure 1 depicts the linear time - branching time spectrum, where a directed edge from one equivalence to another means that the source of the edge is finer than the target.

Van Glabbeek [10, 11] studied the semantics in his spectrum in the setting of the process algebra BCCSP, which contains only basic process algebraic operators from CCS and CSP, but is sufficiently powerful to express all finite synchronization trees. Van Glabbeek gave (sound and complete) axiomatizations for semantics in the spectrum, meaning that two *closed* BCCSP terms (i.e., terms that do not contain variables) can be equated if and only if they are equivalent.

An axiomatization E is ω*-complete* when an equation can be derived from E if (and only if) all its closed instantiations can be derived from E. In theorem proving applications, it is convenient if an axiomatization has this property, because it means that proofs by (structural) induction can be avoided in favor of purely equational reasoning; see [16]. In [14] it was argued that ω-completeness is desirable for the partial evaluation of programs.

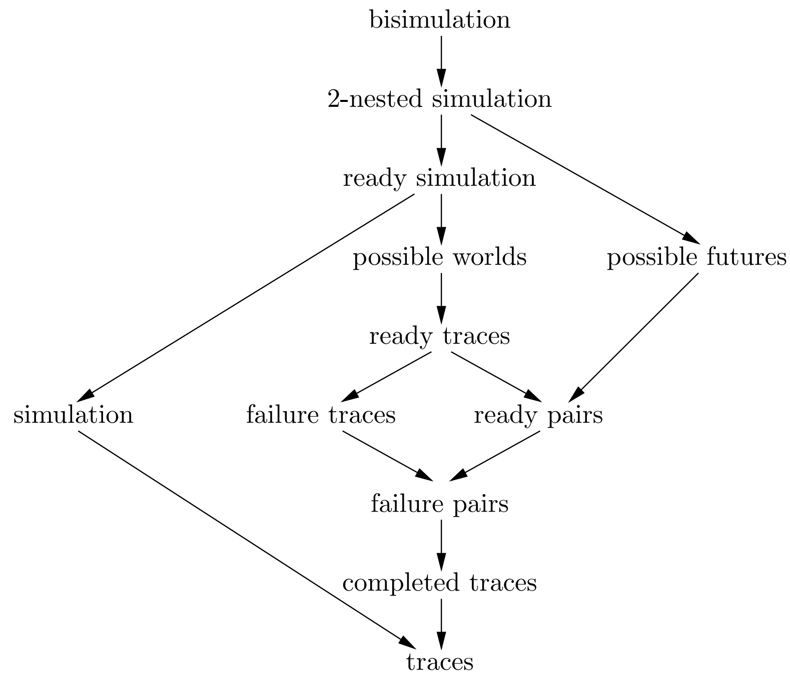

Fig. 1. The linear time - branching time spectrum

Notable examples of ω-incomplete axiomatizations in the literature are the $\lambda K\beta\eta$-calculus (see [27]) and the equational theory of CCS [20]. Therefore laws such as commutativity of parallelism, which are valid in the initial model but which cannot be derived, are often added to the latter equational theory. For such extended equational theories, ω-completeness results were presented in the setting of CCS [22] and ACP [8]. Another negative result, for basic process algebra with the binary Kleene star, was reported in [2]: semantics no coarser than completed trace equivalence and no finer than ready simulation equivalence have no finite (sound and complete) axiomatization, so by default no finite ω-complete axiomatization.

A number of positive and negative results regarding finite ω-complete axiomatizations for BCCSP occur in the literature. Moller [22] proved that the finite axiomatization for BCCSP modulo bisimulation equivalence is ω-complete. Groote [12] presented a similar result for completed trace equivalence, for trace equivalence (in case of an alphabet with more than one element), and for readiness, failure and failure trace equivalence (in case of an infinite alphabet). Blom, Fokkink and Nain [4] proved that in case of an infinite alphabet, BCCSP modulo ready trace equivalence does not have a finite (sound and complete) axiomatization. Aceto, Fokkink and Ingólfsdóttir [3] proved a similar negative result for

2-nested simulation equivalence, independent of the cardinality of the alphabet.[1] Fokkink and Nain [9] showed that in case of a finite alphabet with more than one element, BCCSP modulo any semantics no coarser than readiness equivalence and no finer than possible worlds equivalence does not have a finite ω-complete axiomatization.

A *basis* of an equational theory is a set of axioms from which all equations in the theory can be derived. The existence of a finite basis for an equational theory is a classic topic of study in universal algebra (see, e.g., [19]), dating back to Lyndon [17]. Murskiĭ [24] proved that "almost all" finite algebras (namely all quasi-primal ones) are finitely based, while in [23] he presented an example of a three-element algebra that has no finite basis. Henkin [15] showed that the algebra of naturals with addition and multiplication is finitely based, while Gurevič [13] showed that after adding exponentiation the algebra is no longer finitely based. McKenzie [18] settled Tarski's Finite Basis Problem in the negative, by showing that the general question whether a finite algebra is finitely based is undecidable.

In this paper we present a finite ω-complete axiomatization for BCCSP modulo failure semantics. This provides a positive answer to an open question from [12]. The axiomatization consists of the standard axioms A1-4 for bisimulation, two standard axioms F1-2 for failure semantics, and a new axiom F3 that requires a finite alphabet. The latter axiom was obtained by considering *cover equations*, which aim to obtain a full coverage of the equational theory for (in this case) failure semantics. The central idea is that it is sufficient to only consider equations of the form $at + u \approx u$ and $x + u \approx u$ (where a denotes an action, x a variable, and t, u BCCSP terms). We classified the sound equations of this form, which we call *cover equations*. Now one can proceed in two ways. Either one can determine an infinite family of cover equations that obstructs a finite basis; this approach we took in [9] to prove the absence of a finite basis for a range of process semantics. Or one can determine a finite basis among the cover equations; this approach is followed in the current paper. We only present A1-4 and F1-3 together with a proof that this axiomatization is ω-complete, as the full classification of cover equations is quite involved and not needed for the proof.

We also present a proof that A1-4 together with F1-2 are ω-complete in case of an infinite alphabet. This proof is considerably simpler than the proof in [12], and moreover it is a sub-proof of the proof that A1-4 together with F1-3 are ω-complete in case of a finite alphabet. Last but not least, our axioms F1-2 are simplifications with respect to the axioms for failure equivalence as presented in [11, 12].

Groote [12] also asked whether in case of a finite alphabet, BCCSP modulo failure trace semantics has a finite ω-complete axiomatization. This question remains open.

[1] In case of an infinite alphabet, occurrences of action names in axioms should be interpreted as variables, as else most of the axiomatizations mentioned in this paragraph would be infinite.

2 Preliminaries

Syntax of BCCSP BCCSP(A) is a basic process algebra for expressing finite process behavior. Its syntax consists of closed (process) terms p, q that are constructed from a constant $\mathbf{0}$, a binary operator $\_+\_$ called *alternative composition*, and unary *prefix* operators $a\_$, where a ranges over a nonempty set A of actions, called the *alphabet* (with typical elements a, b, c). Open terms t, u can moreover contain variables from a countably infinite set V (with typical elements x, y, z). A (closed) substitution maps variables in V to (closed) terms. For every term t and substitution σ, the term $\sigma(t)$ is obtained by replacing every occurrence of a variable x in t by $\sigma(x)$.

Transition rules Intuitively, closed terms represent finite process behaviors, where $\mathbf{0}$ does not exhibit any behavior, $p + q$ is the nondeterministic choice between the behaviors of p and q, and ap executes action a to transform into p. This intuition is captured, in the style of Plotkin [26], by the transition rules below, which give rise to A-labeled transitions between closed terms.

$$\frac{}{ax \xrightarrow{a} x} \qquad \frac{x \xrightarrow{a} x'}{x + y \xrightarrow{a} x'} \qquad \frac{y \xrightarrow{a} y'}{x + y \xrightarrow{a} y'}$$

The *depth* of a term t, denoted by $depth(t)$, is the maximal number of transitions in sequence that t can exhibit. It is defined by: $depth(\mathbf{0}) = 0$, $depth(x) = 0$, $depth(t + u) = \max\{depth(t), depth(u)\}$, and $depth(at) = depth(t) + 1$.

For a closed term p, $\mathcal{I}(p)$ denotes the set of actions a for which there exists a transition $p \xrightarrow{a} p'$.

Definition 1. *A pair $(a_1 \cdots a_k, B)$ with $B \subseteq A$ and $k \geq 0$ is a* failure pair *of p_0 if $p_0 \xrightarrow{a_1} p_1 \cdots \xrightarrow{a_k} p_k$ with $\mathcal{I}(p_k) \cap B = \emptyset$. Two closed terms p and q are* failure equivalent*, denoted by $p \sim_{\mathrm{F}} q$, if they have exactly the same failure pairs.*

Failure equivalence is a *congruence* for BCCSP(A), meaning that $p_1 \sim_{\mathrm{F}} q_1$ and $p_2 \sim_{\mathrm{F}} q_2$ implies $ap_1 \sim_{\mathrm{F}} aq_1$ for $a \in A$ and $p_1 + p_2 \sim_{\mathrm{F}} q_1 + q_2$. This follows from the fact that the transition rules above are in the failure format from [5].

Axiomatization An *(equational) axiomatization* E for BCCSP(A) is a collection of equations $t \approx u$. We write $E \vdash t \approx u$ if this equation can be derived from the equations in E using the standard rules of equational logic, and $E \vdash F$ if $E \vdash t \approx u$ for all $t \approx u \in F$. An axiomatization E is *sound* modulo an equivalence $\sim$ on closed terms if $(E \vdash p \approx q) \Rightarrow p \sim q$, and it is *complete* modulo $\sim$ if $p \sim q \Rightarrow (E \vdash p \approx q)$, for all closed terms p and q. An axiomatization E is *ω-complete* if for each equation $t \approx u$ with $E \vdash \sigma(t) \approx \sigma(u)$ for all closed substitutions σ, we have $E \vdash t \approx u$.

The core axioms A1-4 below from [20] are sound and complete for BCCSP(A) modulo bisimulation equivalence [25], which is the finest semantics in the linear time - branching time spectrum (see Figure 1).

$$
\begin{array}{ll}
\text{A1} & x + y \approx y + x \\
\text{A2} & (x + y) + z \approx x + (y + z) \\
\text{A3} & x + x \approx x \\
\text{A4} & x + \mathbf{0} \approx x
\end{array}
$$

In the remainder of this paper, process terms are considered modulo A1-4.

We use *summation* $\sum_{i=1}^{k} t_i$ or $\sum_{i \in \{1,\ldots,k\}} t_i$, with $k \geq 0$, to denote $t_1 + \cdots + t_k$, where the empty sum denotes $\mathbf{0}$. Each process term is of the form $\sum_{i=1}^{k} a_i t_i + \sum_{j=1}^{\ell} x_j$. The $a_i t_i$ and x_j are called the *summands* of this process term.

As binding convention, alternative composition and summation bind weaker than prefixing.

3 A Finite Basis for Failure Semantics

3.1 Cover Equations

The central idea is that for bisimulation semantics, and thus for all process semantics in the linear time - branching time spectrum, axiom A3 is sound. So if an equation $t \approx u$ is sound, then $u + t \approx t$ and $t + u \approx u$ are sound too; and from the last two equations one can derive $t \approx u$, using axiom A1. Hence, by A2 and A4, it is sufficient to only consider sound equations of the form $at + u \approx u$ and $x + u \approx u$. We call these the *cover equations*. We present three lemmas that limit the form that cover equations can have.

Lemma 1. *If $t \approx u$ is sound modulo $\sim_{\mathrm{F}}$, then t and u have the same variable summands.*

Proof. Let $x \in V$ be a summand of t. We define $\sigma(x) = a^{depth(u)+1}\mathbf{0}$ for some $a \in A$, and $\sigma(y) = \mathbf{0}$ for $y \neq x$. Then $(a^{depth(u)+1}, A)$ is a failure pair of $\sigma(t)$, so it must be a failure pair of $\sigma(u)$. This implies that x is a summand of u. $\square$

Remark 1. Lemma 1 fails for trace equivalence. Namely, let $A = \{a\}$. Then $x + ax \approx ax$ is sound modulo trace equivalence. Lemma 1 does hold for completed trace equivalence.

Lemma 2. *If $at + u + bv \approx u + bv$ with $a \neq b$ is sound modulo $\sim_{\mathrm{F}}$, then $at + u \approx u$ is sound modulo $\sim_{\mathrm{F}}$.*

Proof. Since $at + u + bv \approx u + bv$ is sound modulo $\sim_{\mathrm{F}}$ and $a \neq b$, for each closed substitution σ:

- $\mathcal{I}(\sigma(at + u)) = \mathcal{I}(\sigma(u))$, and
- each failure pair $(ab_1 \cdots b_k, B)$ of $a\sigma(t)$ is a failure pair of $\sigma(u)$.

So $at + u \approx u$ is sound modulo $\sim_{\mathrm{F}}$. $\square$

Lemma 3. *If $t + x \approx u + x$ is sound modulo $\sim_{\mathrm{F}}$, and x is not a summand of $t + u$, then $t \approx u$ is sound modulo $\sim_{\mathrm{F}}$.*

Proof. Suppose that $t \approx u$ is not sound modulo $\sim_{\mathrm{F}}$; we prove that then $t + x \approx u + x$ is not sound modulo $\sim_{\mathrm{F}}$.

$\sigma(t) \not\sim_{\mathrm{F}} \sigma(u)$ for some closed substitution σ. Without loss of generality we can assume that some failure pair $(b_1 \cdots b_k, B_0)$ of $\sigma(t)$ is not a failure pair of $\sigma(u)$. The (fixed) set B_0 will play a crucial role in the construction below. We distinguish two cases.

1. $k = 0$.
   Then $\mathcal{I}(\sigma(t)) \neq \mathcal{I}(\sigma(u))$. Let $\sigma'(x) = \mathbf{0}$ and $\sigma'(y) = \sigma(y)$ for $y \neq x$. Since x is not a summand of $t+u$, $\mathcal{I}(\sigma'(t+x)) = \mathcal{I}(\sigma(t))$ and $\mathcal{I}(\sigma'(u+x)) = \mathcal{I}(\sigma(u))$. Then $\mathcal{I}(\sigma'(t+x)) \neq \mathcal{I}(\sigma'(u+x))$, and so $\sigma'(t+x) \not\sim_{\mathrm{F}} \sigma'(u+x)$.
2. $k > 0$.
   We define a substitution σ' such that $\sigma'(y) = \sigma(y)$ for $y \neq x$, and $\sigma'(x)$ has the same failure pairs $(c_1 \cdots c_\ell, B_0)$ as $\sigma(x)$ for $\ell < k$, while it does not have failure pairs $(c_1 \cdots c_k, B_0)$. We obtain $\sigma'(x)$ from $\sigma(x)$ by replacing subterms ap at depth $k-1$ by $\mathbf{0}$ if $a \notin B_0$, or by $aa\mathbf{0}$ if $a \in B_0$. That is,

   $$\sigma'(x) = \mathit{chop}_{k-1}(\sigma(x))$$

   where

   $$\begin{array}{ll} \mathit{chop}_m(\mathbf{0}) & = \mathbf{0} \\ \mathit{chop}_m(p+q) & = \mathit{chop}_m(p) + \mathit{chop}_m(q) \\ \mathit{chop}_0(ap) & = \begin{cases} \mathbf{0} & \text{if } a \notin B_0 \\ aa\mathbf{0} & \text{if } a \in B_0 \end{cases} \\ \mathit{chop}_{m+1}(ap) & = a\,\mathit{chop}_m(p) \end{array}$$

   We prove the two desired properties concerning the failure pairs of $\mathit{chop}_m(p)$, for $m \geq 0$ and closed terms p. Let $c_1, \ldots, c_{m+1}$ range over A.

   I For $\ell \leq m$, p and $\mathit{chop}_m(p)$ have the same failure pairs $(c_1 \cdots c_\ell, B_0)$.
   We use induction on ℓ. *Base case:* Since the summands of $\mathit{chop}_0(p)$ are $aa\mathbf{0}$ with $a \in \mathcal{I}(p) \cap B_0$, $\mathcal{I}(p) \cap B_0 = \emptyset$ if and only if $\mathcal{I}(\mathit{chop}_0(p)) \cap B_0 = \emptyset$. *Inductive case:* Let $\ell + 1 \leq m$. By induction, for closed terms q, q and $\mathit{chop}_{m-1}(q)$ have the same failure pairs $(c_2 \cdots c_{\ell+1}, B_0)$. Since $m > 0$, the transitions of $\mathit{chop}_m(p)$ are $\mathit{chop}_m(p) \xrightarrow{c_1} \mathit{chop}_{m-1}(p')$ for $p \xrightarrow{c_1} p'$. Hence p and $\mathit{chop}_m(p)$ have the same failure pairs $(c_1 \cdots c_{\ell+1}, B_0)$.

   II $\mathit{chop}_m(p)$ does not have failure pairs $(c_1 \cdots c_{m+1}, B_0)$.
   We use induction on m. *Base case:* Since the summands of $\mathit{chop}_0(p)$ are $aa\mathbf{0}$ with $a \in \mathcal{I}(p) \cap B_0$, $\mathit{chop}_0(p)$ does not have a failure pair (c_1, B_0). *Inductive case:* By induction, for closed terms q, $\mathit{chop}_m(q)$ does not have failure pairs $(c_2 \cdots c_{m+2}, B_0)$. Since the transitions of $\mathit{chop}_{m+1}(p)$ are $\mathit{chop}_{m+1}(p) \xrightarrow{c_1} \mathit{chop}_m(p')$ for $p \xrightarrow{c_1} p'$, it follows that $\mathit{chop}_{m+1}(p)$ does not have failure pairs $(c_1 \cdots c_{m+2}, B_0)$.

   We proceed to relate the failure pairs of $\sigma(t_0)$ and $\sigma'(t_0)$, for terms t_0.

III For $\ell < k$, $\sigma(t_0)$ and $\sigma'(t_0)$ have the same failure pairs $(c_1 \cdots c_\ell, B_0)$.
We apply induction on ℓ. *Base case:* Clearly, $\sigma(x) \cap B_0 = \emptyset$ if and only if $\sigma'(x) \cap B_0 = \emptyset$, and moreover $\sigma(y) = \sigma'(y)$ for $y \neq x$. This implies that $\mathcal{I}(\sigma(t_0)) \cap B_0 = \emptyset$ if and only if $\mathcal{I}(\sigma'(t_0)) \cap B_0 = \emptyset$. *Inductive case:* Let $\ell + 1 < k$. We prove for each summand of t_0 that applying σ or σ' gives rise to the same failure pairs $(c_1 \cdots c_{\ell+1}, B_0)$. By property I, $\sigma(x)$ and $\sigma'(x)$ have the same failure pairs $(c_1 \cdots c_{\ell+1}, B_0)$. Moreover, $\sigma(y) = \sigma'(y)$ for $y \neq x$. Furthermore, by induction, for each summand $c_1 t_1$ of t_0, $\sigma(t_1)$ and $\sigma'(t_1)$ have the same failure pairs $(c_2 \cdots c_{\ell+1}, B_0)$; so $\sigma(c_1 t_1)$ and $\sigma'(c_1 t_1)$ have the same failure pairs $(c_1 \cdots c_{\ell+1}, B_0)$.

IV If t_0 does not have the summand x, then $\sigma(t_0)$ and $\sigma'(t_0)$ have the same failure pairs $(c_1 \cdots c_k, B_0)$.
We prove for each summand of t_0 that applying σ or σ' gives rise to the same failure pairs $(c_1 \cdots c_k, B_0)$. $\sigma(y) = \sigma'(y)$ for $y \neq x$. Furthermore, by property III, for each summand $c_1 t_1$ of t_0, $\sigma(t_1)$ and $\sigma'(t_1)$ have the same failure pairs $(c_2 \cdots c_k, B_0)$; so $\sigma(c_1 t_1)$ and $\sigma'(c_1 t_1)$ have the same failure pairs $(c_1 \cdots c_k, B_0)$.

Recall that $(b_1 \cdots b_k, B_0)$ is a failure pair of $\sigma(t)$ and not of $\sigma(u)$. Since x is not a summand of $t + u$, by property IV, $(b_1 \cdots b_k, B_0)$ is a failure pair of $\sigma'(t)$ and not of $\sigma'(u)$. Since $k > 0$, $(b_1 \cdots b_k, B_0)$ is a failure pair of $\sigma'(t + x)$. By property II, $\sigma'(x) = \mathit{chop}_{k-1}(\sigma(x))$ does not have the failure pair $(b_1 \cdots b_k, B_0)$, so $(b_1 \cdots b_k, B_0)$ is not a failure pair of $\sigma'(u + x)$. Hence $\sigma'(t + x) \not\sim_{\mathrm{F}} \sigma'(u + x)$.

We conclude that $t + x \approx u + x$ is not sound modulo $\sim_{\mathrm{F}}$. $\square$

Remark 2. The condition in Lemma 3 that x is not a summand of $t + u$ is essential. For instance, $x + x \approx \mathbf{0} + x$ is sound modulo $\sim_{\mathrm{F}}$, but $x \approx \mathbf{0}$ is not.

Remark 3. There exist equivalences in between bisimulation and partial traces that are a congruence for BCCSP(A), but for which Lemma 2 and/or Lemma 3 fail. So these lemmas have to be proved for each equivalence in the linear time - branching time spectrum individually.

3.2 ω-Complete Axiomatizations for Failure Semantics

We present two axioms for failure semantics.[2]

$$
\begin{array}{ll}
\mathrm{F1} & a(x+y) + ax + a(y+z) \approx ax + a(y+z) \\
\mathrm{F2} & a(x+by) + a(x+by+bz) \approx a(x+by+bz)
\end{array}
$$

It is not hard to see that F1 and F2 are sound for BCCSP(A) modulo $\sim_{\mathrm{F}}$.

Theorem 1. *[10] A1-4+F1-2 is complete for* BCCSP(A) *modulo* $\sim_{\mathrm{F}}$.

[2] Van Glabbeek [11] and Groote [12] presented a somewhat more complicated version of F2. In [11] it takes the form $a(bx+u) + a(by+v) \approx a(bx+by+u) + a(by+v)$.

Theorem 2. *[12] If $|A| = \infty$, then A1-4+F1-2 is ω-complete.*

In case of a finite alphabet, A1-4+F1-2 is not ω-complete. Let $A = \{a_1, \ldots, a_n\}$ for some $n > 0$. Then the following axiom is sound for BCCSP(A) modulo $\sim_{\mathrm{F}}$.

$$\mathrm{F3}_n \quad a(x + \textstyle\sum_{i=1}^{n} a_i z_i) + a(x + y + \sum_{i=1}^{n} a_i z_i) \approx a(x + y + \sum_{i=1}^{n} a_i z_i)$$

$\mathrm{F3}_n$ cannot be derived from A1-4+F1-2. This follows from the fact that A1-4+F1-2 are sound for BCCSP(A) modulo $\sim_{\mathrm{F}}$ in case of an infinite alphabet, while $\mathrm{F3}_n$ is not.

Theorem 3. *If $A = \{a_1, \ldots, a_n\}$, then A1-4+F1-2+F3$_n$ is ω-complete.*

The main aim of this paper is to present a proof of Theorem 3. Furthermore, although Theorems 1 and 2 have already been proved in earlier papers, we provide new proofs here. First of all, since axiom F2 is presented in a simpler form here than in [11, 12], strictly speaking Theorems 1 and 2 have not been proved in [10, 12], but are immediate corollaries from results in those papers and the fact that the earlier axiom $a(bx+u)+a(by+v) \approx a(bx+by+u)+a(by+v)$ can be derived from A1-4+F1-2. More important, Theorems 1 and 2 follow immediately from our proof of Theorem 3. So we obtain the new proofs for free, and moreover they are considerably simpler than the old proofs. In particular, the new proofs are fully equational, and more direct than the old proofs, which involve term rewriting and normal forms.

By abuse of notation, we let a finite set $X \subset V$ denote the term $\sum_{x \in X} x$. From now on, X, Y (possibly subscripted) denote finite subsets of V.

Proof. We derive all equations $t \approx u$ that are sound modulo $\sim_{\mathrm{F}}$, by induction on $\max\{\mathit{depth}(t), \mathit{depth}(u)\}$. Clearly $u + t \approx t$ and $t + u \approx u$ are sound too; and from the last two equations one can derive $t \approx u$. So it suffices to derive all sound equations of the form $at + u \approx u$ and $x + u \approx u$. In view of Lemma 1, soundness of $x + u \approx u$ implies that x is a summand of u, so that $x + u \approx u$ follows from A3. So it suffices to derive all sound equations of the form $at + u \approx u$. In view of Lemmas 2 and 3, we can take u to be of the form $\sum_{j \in J} au_j$. (Note that the equations to-be-proved, which are thus obtained from the original equation $t \approx u$, involve terms with a depth $\leq \max\{\mathit{depth}(t), \mathit{depth}(u)\}$; so the induction order is respected.)

Hence, to prove Theorems 1, 2 and 3, it suffices to prove that each equation $at + \sum_{j \in J} au_j \approx \sum_{j \in J} au_j$ that is sound modulo $\sim_{\mathrm{F}}$ can be derived from A1-4+F1-2 in case of an infinite alphabet, and from A1-4+F1-2+F3$_n$ in case of a finite alphabet with n elements. Consider such a sound equation modulo $\sim_{\mathrm{F}}$:

$$a(X + \sum_{i \in I} b_i t_i) + \sum_{j \in J} a(Y_j + \sum_{k \in K_j} c_k u_k) \approx \sum_{j \in J} a(Y_j + \sum_{k \in K_j} c_k u_k) \qquad (1)$$

As said before, we apply induction on the depth of the terms in the equation. In the base case of the induction, $I = \emptyset$ and $K_j = \emptyset$ for $j \in J$.

The main idea of the proof will be to restrict the syntactic form that the summands in equation (1) can have, by exploiting the fact that it is sound modulo $\sim_{\mathrm{F}}$. Thus the application of a suitable closed substitution can provide information on syntactic relations between the summands.

Suppose, towards a contradiction, that $X \not\subseteq \cup_{j\in J} Y_j$. Let $x \in X \backslash (\cup_{j\in J} Y_j)$. Let σ be a closed substitution with $\sigma(x) = a^{\ell}\mathbf{0}$ where ℓ is greater than the depth of the terms in (1), and $\sigma(y) = \mathbf{0}$ for all $y \neq x$. Then clearly $(a^{\ell+1}, A)$ is a failure pair of $\sigma(a(X + \sum_{i\in I} b_i t_i))$, but not of $\sigma(\sum_{j\in J} a(Y_j + \sum_{k\in K_j} c_k u_k))$ for $j \in J$, contradicting the soundness of (1). Hence $X \subseteq \cup_{j\in J} Y_j$.

Suppose, towards a contradiction, that $\{b_i \mid i \in I\} \not\subseteq \cup_{j\in J}\{c_k \mid k \in K_j\}$. Let $b \in \{b_i \mid i \in I\} \backslash \cup_{j\in J} \{c_k \mid k \in K_j\}$. Let σ be the closed substitution with $\sigma(y) = \mathbf{0}$ for all $y \in V$. Then clearly $(ab, \emptyset)$ is a failure pair of $\sigma(a(X+\sum_{i\in I} b_i t_i))$, but not of $\sigma(a(Y_j + \sum_{k\in K_j} c_k u_k))$ for $j \in J$, contradicting the soundness of (1). Hence $\{b_i \mid i \in I\} \subseteq \cup_{j\in J}\{c_k \mid k \in K_j\}$.

Since (1) is sound, clearly

$$X + \sum_{i\in I} b_i t_i + \sum_{j\in J} (Y_j + \sum_{k\in K_j} c_k u_k) \approx \sum_{j\in J} (Y_j + \sum_{k\in K_j} c_k u_k) \tag{2}$$

is also sound modulo $\sim_{\mathrm{F}}$. Let $L_j = \{k \in K_j \mid c_k \in \{b_i \mid i \in I\}\}$. By Lemmas 2 and 3, (2) implies that

$$\sum_{i\in I} b_i t_i + \sum_{j\in J} \sum_{k\in L_j} c_k u_k \approx \sum_{j\in J} \sum_{k\in L_j} c_k u_k \tag{3}$$

is sound modulo $\sim_{\mathrm{F}}$. Hence, by induction on depth (or, in the base case, because $I = \emptyset$ and $L_j = \emptyset$ for $j \in J$), it can be derived from A1-4, F1-2 and, in case of a finite alphabet with n elements, F3$_n$.

We consider two cases.

1. $\{b_i \mid i \in I\} \neq A$.
   Suppose, towards a contradiction, that for all $j \in J$, either $Y_j \not\subseteq X$ or $\{c_k \mid k \in K_j\} \not\subseteq \{b_i \mid i \in I\}$. Let $c \in A\backslash\{b_i \mid i \in I\}$. Let σ be the closed substitution with $\sigma(x) = \mathbf{0}$ for $x \in X$ and $\sigma(y) = c$ for $y \notin X$. Then clearly $(a, A\backslash\{b_i \mid i \in I\})$ is a failure pair of $\sigma(a(X + \sum_{i\in I} b_i t_i))$, but not of $\sigma(a(Y_j + \sum_{k\in K_j} c_k u_k))$ for $j \in J$, contradicting the soundness of (1). Hence there is a $j_0 \in J$ such that $Y_{j_0} \subseteq X$ and $\{c_k \mid k \in K_{j_0}\} \subseteq \{b_i \mid i \in I\}$.
   We start with the term $\sum_{j\in J} a(Y_j + \sum_{k\in K_j} c_k u_k)$. Since $Y_{j_0} \subseteq X \subseteq \cup_{j\in J} Y_j$ and $K_{j_0} = L_{j_0}$, we can use F1 to convert the summand $a(Y_{j_0} + \sum_{k\in K_{j_0}} c_k u_k)$ into $a(X + \sum_{j\in J} \sum_{k\in L_j} c_k u_k)$. By (3) we convert this into $a(X + \sum_{i\in I} b_i t_i + \sum_{j\in J} \sum_{k\in L_j} c_k u_k)$. Finally, since $c_k \in \{b_i \mid i \in I\}$ for $k \in L_j$ and $j \in J$, we can use F2 to convert this into $a(X + \sum_{i\in I} b_i t_i)$. Thus we have derived (1).
2. $\{b_i \mid i \in I\} = A$. Note that in this case A is finite.
   We start with the term $\sum_{j\in J} a(Y_j + \sum_{k\in K_j} c_k u_k)$. Using F1 we can create the summand $a(\sum_{j\in J} Y_j + \sum_{j\in J} \sum_{k\in K_j} c_k u_k)$. Note that $K_j = L_j$ for $j \in J$, because $\{b_i \mid i \in I\} = A$. So we can use (3) to convert this summand into

$a(\sum_{j\in J} Y_j + \sum_{i\in I} b_i t_i + \sum_{j\in J}\sum_{k\in K_j} c_k u_k)$. Finally, since $X \subseteq \cup_{j\in J} Y_j$ and $\{b_i \mid i \in I\} = A$, we can use $\mathrm{F3}_{|A|}$ to convert this into $a(X + \sum_{i\in I} b_i t_i)$. Thus we have derived (1).

We note that in case of an infinite alphabet, the axioms $\mathrm{F3}_n$ were not used in the derivation of (1). □

References

1. L. Aceto, W.J. Fokkink, R.J. van Glabbeek, and A. Ingólfsdóttir. Axiomatizing prefix iteration with silent steps. *Information Computation*, 127(1):26–40, 1996.
2. L. Aceto, W.J. Fokkink, and A. Ingólfsdóttir. A menagerie of non-finitely based process semantics over BPA*: From ready simulation to completed traces. *Mathematical Structures in Computer Science*, 8(3):193–230, 1998.
3. L. Aceto, W.J. Fokkink, and A. Ingólfsdóttir. 2-nested simulation is not finitely equationally axiomatizable. In *Proceedings 18th Symposium on Theoretical Aspects of Computer Science (STACS'01)*, Dresden, LNCS 2010, pp. 39–50. Springer, 2001.
4. S.C.C. Blom, W.J. Fokkink, and S. Nain. On the axiomatizability of ready traces, ready simulation and failure traces. In *Proceedings 30th Colloquium on Automata, Languages and Programming (ICALP'03)*, Eindhoven, LNCS. Springer, 2003. To appear.
5. B. Bloom, W.J. Fokkink, and R.J. van Glabbeek. Precongruence formats for decorated trace semantics. *ACM Transactions on Computational Logic*, 5(1):26–78, 2004.
6. S.D. Brookes, C.A.R. Hoare, and A.W. Roscoe. A theory of communicating sequential processes. *Journal of the ACM*, 31(3):560–599, 1984.
7. R. De Nicola and M.C.B. Hennessy. Testing equivalences for processes. *Theoretical Computer Science*, 34(1/2):83–133, 1984.
8. W.J. Fokkink and S.P. Luttik. An ω-complete equational specification of interleaving. In *Proceedings 27th Colloquium on Automata, Languages and Programming (ICALP'00)*, Geneva, LNCS 1853, pp. 729–743. Springer, 2000.
9. W.J. Fokkink and S. Nain. On finite alphabets and infinite bases: from ready pairs to possible worlds. In *Proceedings 7th Conference on Foundations of Software Science and Computation Structures (FOSSACS'04)*, Barcelona, LNCS 2987, pp. 182-194. Springer, 2004.
10. R.J. van Glabbeek. The linear time – branching time spectrum. In *Proceedings 1st Conference on Concurrency Theory (CONCUR'90)*, Amsterdam, LNCS 458, pp. 278–297. Springer 1990
11. R.J. van Glabbeek. The linear time – branching time spectrum I. The semantics of concrete, sequential processes. In J.A. Bergstra, A. Ponse, and S.A. Smolka, eds, *Handbook of Process Algebra*, pp. 3–99. Elsevier, 2001.
12. J.F. Groote. A new strategy for proving ω-completeness with applications in process algebra. In *Proceedings 1st Conference on Concurrency Theory (CONCUR'90)*, Amsterdam, LNCS 458, pp. 314–331. Springer, 1990.
13. R. Gurevič. Equational theory of positive natural numbers with exponentiation is not finitely axiomatizable. *Annals of Pure and Applied Logic*, 49:1–30, 1990.
14. J. Heering. Partial evaluation and ω-completeness of algebraic specifications. *Theoretical Computer Science*, 43:149–167, 1986.

15. L. Henkin. The logic of equality. *American Mathematical Monthly*, 84(8):597–612, 1977.
16. A. Lazrek, P. Lescanne, and J.-J. Thiel. Tools for proving inductive equalities, relative completeness, and ω-completeness. *Information and Computation*, 84(1):47–70, 1990.
17. R.C. Lyndon. Identities in two-valued calculi. *Transactions of the American Mathematical Society*, 71:457–465, 1951.
18. R.N. McKenzie. Tarski's finite basis problem is undecidable. *International Journal of Algebra and Computation*, 6(1):49–104, 1996.
19. R.N. McKenzie, G. McNulty, and W. Taylor. *Algebras, Varieties, Lattices.* Wadsworth & Brooks/Cole, 1987.
20. R. Milner. *Communication and Concurrency.* Prentice Hall, 1989.
21. R. Milner. A complete axiomatisation for observational congruence of finite-state behaviours. *Information and Computation*, 81(2):227–247, 1989.
22. F. Moller. *Axioms for Concurrency.* PhD thesis, University of Edinburgh, 1989.
23. V.L. Murskiĭ. The existence in the three-valued logic of a closed class with a finite basis having no finite complete system of identities. *Doklady Akademii Nauk SSSR*, 163:815–818, 1965. In Russian.
24. V.L. Murskiĭ. The existence of a finite basis of identities, and other properties of "almost all" finite algebras. *Problemy Kibernetiki*, 30:43–56, 1975. In Russian.
25. D.M.R. Park. Concurrency and automata on infinite sequences. In *Proceedings 5th GI (Gesellschaft für Informatik) Conference*, Karlsruhe, LNCS 104, pp. 167–183. Springer, 1981.
26. G.D. Plotkin. A structural approach to operational semantics. Report DAIMI FN-19, Aarhus University, 1981.
27. G.D. Plotkin. The λ-calculus is ω-incomplete. *Journal of Symbolic Logic*, 39:313–317, 1974.