A Finite Equational Base for CCS with Left Merge and Communication Merge*

Luca Aceto[1,4], Wan Fokkink[2,5], Anna Ingolfsdottir[1,4], and Bas Luttik[3,5]

[1] Department of Computer Science, Reykjavík University, Iceland
luca@ru.is, annai@ru.is
[2] Department of Computer Science, Vrije Universiteit Amsterdam, The Netherlands
wanf@cs.vu.nl
[3] Department of Mathematics and Computer Science, Technische Universiteit Eindhoven, The Netherlands
s.p.luttik@tue.nl
[4] **BRICS**, Department of Computer Science, Aalborg University, Denmark
[5] Department of Software Engineering, CWI, The Netherlands

Abstract. Using the left merge and communication merge from ACP, we present an equational base (i.e., a ground-complete and ω-complete set of valid equations) for the fragment of CCS without restriction and relabelling. Our equational base is finite if the set of actions is finite.

1 Introduction

One of the first detailed studies of the equational theory of a process algebra was performed by Hennessy and Milner [9]. They considered the equational theory of the process algebra that arises from the recursion-free fragment of CCS (see [11]), and presented a set of equational axioms that is complete in the sense that all valid *closed* equations (i.e., equations in which no variables occur) are derivable from it in equational logic [15]. For the elimination of parallel composition from closed terms, Hennessy and Milner proposed the well-known *Expansion Law*, an axiom schema that generates infinitely many axioms. Thus, the question arose whether a finite complete set of axioms exists. With their axiom system ACP, Bergstra and Klop demonstrated in [3] that it does exist if two auxiliary operators are used: the left merge and the communication merge. It was later proved by Moller [13] that without using at least one auxiliary operator a finite complete set of axioms does not exist.

The aforementioned results pertain to the closed fragments of the equational theories discussed, i.e., to the subsets consisting of the closed valid equations only. Many valid equations, such as the equation $(x \parallel y) \parallel z \approx x \parallel (y \parallel z)$ expressing that parallel composition is associative, are not derivable (by means of equational logic) from the axioms in [3] or [9]. In this paper we shall not neglect the variables and contribute to the study of full equational theories of process algebras. We take the fragment of CCS without recursion, restriction and

* The first and third author were partly supported by the project "The Equational Logic of Parallel Processes" (nr. 060013021) of The Icelandic Research Fund.

M. Bugliesi et al. (Eds.): ICALP 2006, Part II, LNCS 4052, pp. 492–503, 2006.

relabelling, and consider the full equational theory of the process algebra that is obtained by taking the syntax modulo bisimilarity [14]. Our goal is then to present an *equational base* (i.e., a set of valid equations from which every other valid equation can be derived) for it, which is finite if the set of actions is finite. Obviously, Moller's result about the unavoidability of the use of auxiliary operations in a finite complete axiomatisation of the closed fragment of the equational theory of CCS a fortiori implies that auxiliary operations are needed to achieve our goal. So we add left merge and communication merge from the start.

Moller [12] considers the equational theory of the same fragment of CCS, except that his parallel operator implements pure interleaving instead of CCS-communication and the communication merge is omitted. He presents a set of valid axiom schemata and proves that it generates an equational base if the set of actions is infinite. Groote [6] does consider the fragment including communication merge, but, instead of the CCS-communication mechanism, he assumes an uninterpreted communication function. His axiom schemata also generate an equational base provided that the set of actions is infinite. We improve on these results by considering the communication mechanism present in CCS, and by proving that our axiom schemata generate an equational base also if the set of actions is finite. Moreover, our axiom schemata generate a finite equational base if the set of actions is finite.

Our equational base consists of axioms that are mostly well-known. For parallel composition ($\|$), left merge ($\between$) and communication merge ($|$) we adapt the axioms of ACP, adding from Bergstra and Tucker [4] a selection of the axioms for *standard concurrency* and the axiom $(x \mid y) \mid z \approx \mathbf{0}$, which expresses that the communication mechanism is a form of *handshaking communication*.

Our proof follows the classic two-step approach: first we identify a set of normal forms such that every process term has a provably equal normal form, and then we demonstrate that for distinct normal forms there is a distinguishing valuation that proves that they should not be equated. (We refer to the survey [2] for a discussion of proof techniques and an overview of results and open problems in the area. We remark in passing that one of our main results in this paper, viz. Corollary 31, solves the open problem mentioned in [2, p. 362].) Since both associating a normal form with a process term and determining a distinguishing valuation for two distinct normal forms are easily seen to be computable, as a corollary to our proof we get the decidability of the equational theory. Another consequence of our result is that our equational base is complete for the set of valid closed equations as well as ω-complete [7].

The positive result that we obtain in Corollary 31 of this paper stands in contrast with the negative result that we have obtained in [1]. In that article we proved that there does not exist a finite equational base for CCS if the auxiliary operation $\between$ of Hennessy [8] is added instead of Bergstra and Klop's left merge and communication merge. Furthermore, we conjecture that a finite equational base fails to exist if the unary action prefixes are replaced by binary sequential composition. (We refer to [2] for an infinite family of valid equations that we believe cannot all be derivable from a single finite set of valid equations.)

The paper is organised as follows. In Sect. 2 we introduce a class of algebras of processes arising from a process calculus à la CCS, present a set of equations that is valid in all of them, and establish a few general properties needed in the remainder of the paper. Our class of process algebras is parametrised by a communication function. It is beneficial to proceed in this generality, because it allows us to first consider the simpler case of a process algebra with pure interleaving (i.e., no communication at all) instead of CCS-like parallel composition. In Sect. 3 we prove that an equational base for the process algebra with pure interleaving is obtained by simply adding the axiom $x \mid y \approx \mathbf{0}$ to the set of equations introduced in Sect. 2. The proof in Sect. 3 extends nicely to a proof that, for the more complicated case of CCS-communication, it is enough to replace $x \mid y \approx \mathbf{0}$ by $x \mid (y \mid z) \approx \mathbf{0}$; this is discussed in Sect. 4.

2 Algebras of Processes

We fix a set $\mathcal{A}$ of *actions*, and declare a special action τ that we assume is not in $\mathcal{A}$. We denote by $\mathcal{A}_\tau$ the set $\mathcal{A} \cup \{\tau\}$. Generally, we let a and b range over $\mathcal{A}$ and α over $\mathcal{A}_\tau$. We also fix a countably infinite set $\mathcal{V}$ of *variables*. The set $\mathcal{P}$ of *process terms* is generated by the following grammar:

$$P ::= \; x \;\mid\; \mathbf{0} \;\mid\; \alpha.P \;\mid\; P + P \;\mid\; P \between P \;\mid\; P \mid P \;\mid\; P \parallel P \;,$$

with $x \in \mathcal{V}$, and $\alpha \in \mathcal{A}_\tau$. We shall often simply write α instead of $\alpha.\mathbf{0}$. Furthermore, to be able to omit some parentheses when writing terms, we adopt the convention that $\alpha.$ binds stronger, and $+$ binds weaker, than all the other operations.

Table 1. The operational semantics

$$\frac{}{\alpha.P \xrightarrow{\alpha} P} \qquad \frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'} \qquad \frac{Q \xrightarrow{\alpha} Q'}{P + Q \xrightarrow{\alpha} Q'}$$

$$\frac{P \xrightarrow{\alpha} P'}{P \between Q \xrightarrow{\alpha} P' \parallel Q} \qquad \frac{P \xrightarrow{\alpha} P'}{P \parallel Q \xrightarrow{\alpha} P' \parallel Q} \qquad \frac{Q \xrightarrow{\alpha} Q'}{P \parallel Q \xrightarrow{\alpha} P \parallel Q'}$$

$$\frac{P \xrightarrow{a} P',\ Q \xrightarrow{b} Q',\ \gamma(a,b)\!\downarrow}{P \mid Q \xrightarrow{\gamma(a,b)} P' \parallel Q'} \qquad \frac{P \xrightarrow{a} P',\ Q \xrightarrow{b} Q',\ \gamma(a,b)\!\downarrow}{P \parallel Q \xrightarrow{\gamma(a,b)} P' \parallel Q'}$$

A process term is *closed* if it does not contain variables; we denote the set of all closed process terms by $\mathcal{P}_0$. We define on $\mathcal{P}_0$ binary relations $\xrightarrow{\alpha}$ ($\alpha \in \mathcal{A}_\tau$) by means of the transition system specification in Table 1. The last two rules in Table 1 refer to a *communication function* γ, i.e., a commutative and associative

partial binary function $\gamma : \mathcal{A} \times \mathcal{A} \rightharpoonup \mathcal{A}_\tau$. We shall abbreviate the statement '$\gamma(a,b)$ is defined' by $\gamma(a,b)\downarrow$ and the statement '$\gamma(a,b)$ is undefined' by $\gamma(a,b)\uparrow$. We shall in particular consider the following communication functions:

1. The *trivial communication function* is the partial function $f : \mathcal{A} \times \mathcal{A} \rightharpoonup \mathcal{A}_\tau$ such that $f(a,b)\uparrow$ for all $a, b \in \mathcal{A}$.
2. The CCS *communication function* $h : \mathcal{A} \times \mathcal{A} \rightharpoonup \mathcal{A}_\tau$ presupposes a bijection $\bar{\cdot}$ on $\mathcal{A}$ such that $\overline{\overline{a}} = a$ and $\overline{a} \neq a$ for all $a \in \mathcal{A}$, and is then defined by $h(a,b) = \tau$ if $\overline{a} = b$ and undefined otherwise.

Definition 1. A *bisimulation* is a symmetric binary relation $\mathcal{R}$ on $\mathcal{P}_0$ such that $P \mathcal{R} Q$ implies

$$\text{if } P \xrightarrow{\alpha} P', \text{ then there exists } Q' \in \mathcal{P}_0 \text{ such that } Q \xrightarrow{\alpha} Q' \text{ and } P' \mathcal{R} Q'.$$

Closed process terms $P, Q \in \mathcal{P}_0$ are said to be *bisimilar* (notation: $P \leftrightarrows_\gamma Q$) if there exists a bisimulation $\mathcal{R}$ such that $P \mathcal{R} Q$.

The relation $\leftrightarrows_\gamma$ is an equivalence relation on $\mathcal{P}_0$; we denote the equivalence class containing P by $[P]$, i.e.,

$$[P] = \{Q \in \mathcal{P}_0 : P \leftrightarrows_\gamma Q\} \ .$$

The rules in Table 1 are all in de Simone's format [5] if P, P', Q and Q' are treated as variables ranging over closed process terms and the last two rules are treated as rule schemata generating a rule for every a, b such that $\gamma(a,b)\downarrow$. Hence, $\leftrightarrows_\gamma$ has the substitution property for the syntactic constructs of our language of closed process terms, and therefore the constructs induce an algebraic structure on $\mathcal{P}_0/\leftrightarrows_\gamma$, with a constant $\mathbf{0}$, unary operations $\alpha.$ ($\alpha \in \mathcal{A}_\tau$) and four binary operations $+$, $\between$, $|$ and $\|$ defined by $\mathbf{0} = [\mathbf{0}]$, $\alpha.[P] = [\alpha.P]$, and $[P] \star [Q] = [P \star Q]$ for $\star \in \{+, \between, |, \|\}$.

Henceforth, we denote by $\mathbf{P}_\gamma$ (for γ an arbitrary communication function) the algebra obtained by dividing out $\leftrightarrows_\gamma$ on $\mathcal{P}_0$ with constant $\mathbf{0}$ and operations $\alpha.$ ($\alpha \in \mathcal{A}_\tau$), $+$, $\between$, $|$, and $\|$ as defined above. The elements of $\mathbf{P}_\gamma$ are called *processes*, and will be ranged over by p, q and r.

2.1 Equational Reasoning

We can use the full language of process expressions to reason about the elements of $\mathbf{P}_\gamma$. A *valuation* is a mapping $\nu : \mathcal{V} \to \mathbf{P}_\gamma$; it induces an *evaluation mapping*

$$[\![\_]\!]_\nu : \mathcal{P} \to \mathbf{P}_\gamma$$

inductively defined by $[\![x]\!]_\nu = \nu(x)$, $[\![\mathbf{0}]\!]_\nu = \mathbf{0}$, $[\![\alpha.P]\!]_\nu = \alpha.[\![P]\!]_\nu$ and $[\![P \star Q]\!]_\nu = [\![P]\!]_\nu \star [\![Q]\!]_\nu$ for $\star \in \{+, \between, |, \|\}$. A *process equation* is a formula $P \approx Q$ with P and Q process terms; it is said to be *valid* (in $\mathbf{P}_\gamma$) if $[\![P]\!]_\nu = [\![Q]\!]_\nu$ for all $\nu : \mathcal{V} \to \mathbf{P}_\gamma$. If $P \approx Q$ is valid in $\mathbf{P}_\gamma$, then we shall also write $P \leftrightarrows_\gamma Q$. The *equational theory* of the algebra $\mathbf{P}_\gamma$ is the set of all valid process equations, i.e.,

$$EqTh(\mathbf{P}_\gamma) = \{P \approx Q : [\![P]\!]_\nu = [\![Q]\!]_\nu \text{ for all } \nu : \mathcal{V} \to \mathbf{P}_\gamma\} \ .$$

The precise contents of the set $EqTh(\mathbf{P}_\gamma)$ depend to some extent on the choice of γ. For instance, the process equation $x \mid y \approx \mathbf{0}$ is only valid in $\mathbf{P}_\gamma$ if γ is the trivial communication function f; if γ is the CCS communication function h, then $\mathbf{P}_\gamma$ satisfies the weaker equation $x \mid (y \mid z) \approx \mathbf{0}$.

Table 2. Process equations valid in every $\mathbf{P}_\gamma$

A1	$x + y$	$\approx y + x$	C1	$\mathbf{0} \mid x$	$\approx \mathbf{0}$	
A2	$(x + y) + z$	$\approx x + (y + z)$	C2	$a.x \mid b.y$	$\approx \gamma(a,b).(x \parallel y)$	if $\gamma(a,b)\downarrow$
A3	$x + x$	$\approx x$	C3	$a.x \mid b.y$	$\approx \mathbf{0}$	if $\gamma(a,b)\uparrow$
A4	$x + \mathbf{0}$	$\approx x$	C4	$(x + y) \mid z$	$\approx x \mid z + y \mid z$	
			C5	$x \mid y$	$\approx y \mid x$	
L1	$\mathbf{0} \between x$	$\approx \mathbf{0}$	C6	$(x \mid y) \mid z$	$\approx x \mid (y \mid z)$	
L2	$\alpha.x \between y$	$\approx \alpha.(x \parallel y)$	C7	$(x \between y) \mid z$	$\approx (x \mid z) \between y$	
L3	$(x + y) \between z$	$\approx x \between z + y \between z$				
L4	$(x \between y) \between z$	$\approx x \between (y \parallel z)$	P1	$x \parallel y$	$\approx (x \between y + y \between x) + x \mid y$	
L5	$x \between \mathbf{0}$	$\approx x$				

Table 2 lists process equations that are valid in $\mathbf{P}_\gamma$ independently of the choice of γ. (The equations L2, C2 and C3 are actually axiom schemata; they generate an axiom for all $\alpha \in \mathcal{A}_\tau$ and $a, b \in \mathcal{A}$. Note that if $\mathcal{A}$ is finite, then these axiom schemata generate finitely many axioms.) Henceforth whenever we write an equation $P \approx Q$, we shall mean that it is derivable from the axioms in Table 2 by means of equational logic. It is well-known that the rules of equational logic preserve validity. We therefore obtain the following result.

Proposition 2. For all process terms P and Q, if $P \approx Q$, then $P \leftrightarrows_\gamma Q$.

A set of valid process equations is an *equational base* for $\mathbf{P}_\gamma$ if all other valid process equations are derivable from it by means of equational logic. The purpose of this paper is to prove that if we add to the equations in Table 2 the equation $x \mid y \approx \mathbf{0}$ we obtain an equational base for $\mathbf{P}_f$, and if, instead, we add $x \mid (y \mid z) \approx \mathbf{0}$ we obtain an equational base for $\mathbf{P}_h$. Both these equational bases are finite, if the set of actions $\mathcal{A}$ is finite.

Definition 3. Let P be a process term. We define the *height* of a process term P, denoted $h(P)$, inductively as follows:

$$h(\mathbf{0}) = 0\ , \qquad h(P + Q) = \max(h(P), h(Q))\ ,$$
$$h(x) = 1\ , \qquad h(P \star Q) = h(P) + h(Q) \text{ for } \star \in \{\between, \mid, \parallel\}.$$
$$h(\alpha.P) = h(P) + 1\ ,$$

Definition 4. We call a process term *simple* if it is not $\mathbf{0}$ and not an alternative composition.

Lemma 5. For every process term P there exists a collection of simple process terms $S_1, \ldots, S_n$ $(n \geq 0)$ such that $h(P) \geq h(S_i)$ for all $i = 1, \ldots, n$ and

$$P \approx \sum_{i=1}^{n} S_i \qquad \text{(by A1, A2 and A4).}$$

We postulate that the summation of an empty collection of terms denotes $\mathbf{0}$. The terms S_i will be called *syntactic summands* of P.

2.2 General Properties of $\mathbf{P}_\gamma$

We collect some general properties of the algebras $\mathbf{P}_\gamma$ that we shall need in the remainder of the paper.

The binary transition relations $\xrightarrow{\alpha}$ $(\alpha \in \mathcal{A}_\tau)$ on $\mathcal{P}_0$, which were used to associate an operational semantics with closed process terms, will play an important rôle in the remainder of the paper. They induce binary relations on $\mathbf{P}_\gamma$, also denoted by $\xrightarrow{\alpha}$, and defined as the least relations such that $P \xrightarrow{\alpha} P'$ implies $[P] \xrightarrow{\alpha} [P']$. Note that we then get, directly from the definition of bisimulation, that for all $P, P' \in \mathcal{P}_0$:

$$[P] \xrightarrow{\alpha} [P'] \text{ iff for all } Q \in [P] \text{ there exists } Q' \in [P'] \text{ such that } Q \xrightarrow{\alpha} Q'.$$

Proposition 6. For all $p, q, r \in \mathbf{P}_\gamma$:

(a) $p = \mathbf{0}$ iff there do not exist $p' \in \mathbf{P}_\gamma$ and $\alpha \in \mathcal{A}_\tau$ such that $p \xrightarrow{\alpha} p'$;
(b) $\alpha.p \xrightarrow{\beta} r$ iff $\alpha = \beta$ and $r = p$;
(c) $p + q \xrightarrow{\alpha} r$ iff $p \xrightarrow{\alpha} r$ or $q \xrightarrow{\alpha} r$;
(d) $p \between q \xrightarrow{\alpha} r$ iff there exists $p' \in \mathbf{P}_\gamma$ such that $p \xrightarrow{\alpha} p'$ and $r = p' \parallel q$; and
(e) $p \mid q \xrightarrow{\alpha} r$ iff there exist actions $a, b \in \mathcal{A}$ and processes $p', q' \in \mathbf{P}_\gamma$ such that $\alpha = \gamma(a, b)$, $p \xrightarrow{a} p'$, $q \xrightarrow{b} q'$, and $r = p' \parallel q'$; and
(f) $p \parallel q \xrightarrow{\alpha} r$ iff $p \between q \xrightarrow{\alpha} r$ or $q \between p \xrightarrow{\alpha} r$ or $p \mid q \xrightarrow{\alpha} r$.

Let $p, p' \in \mathbf{P}_\gamma$; we write $p \to p'$ if $p \xrightarrow{\alpha} p'$ for some $\alpha \in \mathcal{A}_\tau$ and call p' a *residual* of p.

It is easy to see from Table 1 that if $P \xrightarrow{\alpha} P'$, then P' has fewer symbols than P. Consequently, the length of a transition sequence starting with a process $[P]$ is bounded from above by the number of symbols in P.

Definition 7. The *depth* $|p|$ of an element $p \in \mathbf{P}_\gamma$ is defined as

$$|p| = \max\{n \geq 0 : \ \exists p_n, \ldots, p_0 \in \mathbf{P}_\gamma \text{ s.t. } p = p_n \to \cdots \to p_0\}.$$

The *branching degree* $bdeg(p)$ of an element $p \in \mathbf{P}_\gamma$ is defined as

$$bdeg(p) = |\{(\alpha, p') : p \xrightarrow{\alpha} p'\}| \ .$$

We establish some useful properties of parallel composition on $\mathbf{P}_\gamma$.

Lemma 8. For all $p, q \in \mathbf{P}_\gamma$, $|p \parallel q| = |p| + |q|$.

According to the following lemma and Proposition 2, $\mathbf{P}_\gamma$ is a commutative monoid with respect to $\parallel$, with $\mathbf{0}$ as the identity element.

Lemma 9. The following equations are derivable from the axioms in Table 2:

$$\begin{array}{lll} \text{P2} & (x \parallel y) \parallel z & \approx x \parallel (y \parallel z) \\ \text{P3} & x \parallel y & \approx y \parallel x \\ \text{P4} & x \parallel \mathbf{0} & \approx x \ . \end{array}$$

An element $p \in \mathbf{P}_\gamma$ is *parallel prime* if $p \neq \mathbf{0}$, and $p = q \parallel r$ implies $q = \mathbf{0}$ or $r = \mathbf{0}$. Suppose that p is an arbitrary element of $\mathbf{P}_\gamma$; a *parallel decomposition* of p is a finite multiset $[p_1, \ldots, p_n]$ of parallel primes such that $p = p_1 \parallel \cdots \parallel p_n$. (The process $\mathbf{0}$ has as decomposition the empty multiset, and a parallel prime process p has as decomposition the singleton multiset $[p]$.) The following theorem is a straightforward consequence of the main result in [10].

Theorem 10. Every element of $\mathbf{P}_\gamma$ has a unique parallel decomposition.

The following corollary follows easily from the above unique decomposition result.

Corollary 11 (Cancellation). Let $p, q, r \in \mathbf{P}_\gamma$. If $p \parallel q = p \parallel r$, then $q = r$.

Lemma 12. For all $p, q \in \mathbf{P}_\gamma$, $bdeg(p \parallel q) \geq bdeg(p), bdeg(q)$.

We define a sequence of parallel prime processes with special properties that make them very suitable as tools in our proofs in the remainder of the paper:

$$\varphi_i = \tau.\mathbf{0} + \cdots + \tau^i.\mathbf{0} \qquad (i \geq 1). \tag{1}$$

Lemma 13. (i) For all $i \geq 1$, the processes φ_i are parallel prime.
(ii) The processes φ_i are all distinct, i.e., $\varphi_k = \varphi_l$ implies that $k = l$.
(iii) For all $i \geq 1$, the process φ_i has branching degree i.

3 An Equational Base for $\mathbf{P}_f$

In this section, we prove that an equational base for $\mathbf{P}_f$ is obtained if the axiom

$$\text{F} \quad x \mid y \ \approx \ \mathbf{0}$$

is added to the set of axioms generated by the axiom schemata in Table 2. The resulting equational base is finite if $\mathcal{A}$ is finite.

Henceforth, whenever we write $P \approx_{\text{F}} Q$, we shall mean that the equation $P \approx Q$ is derivable from the axioms in Table 2 and the axiom F.

Proposition 14. For all process terms P and Q, if $P \approx_{\text{F}} Q$, then $P \leftrightarrows_f Q$.

To prove that adding F to the axioms in Table 2 suffices to obtain an equational base for $\mathbf{P}_f$, we need to establish that $P \leftrightarrows_f Q$ implies $P \approx_{\mathrm{F}} Q$ for all process terms P and Q. First, we identify a set of normal forms $\mathcal{N}_{\mathrm{F}}$ such that every process term P can be rewritten to a normal form by means of the axioms.

Definition 15. The set $\mathcal{N}_{\mathrm{F}}$ of F-*normal forms* is generated by:

$$N \ ::= \ \mathbf{0} \ \mid \ N + N \ \mid \ \alpha.N \ \mid \ x \between N \qquad (x \in \mathcal{V},\ \alpha \in \mathcal{A}_\tau).$$

Lemma 16. For all $P \in \mathcal{P}$ there is an $N \in \mathcal{N}_{\mathrm{F}}$ s.t. $P \approx_{\mathrm{F}} N$ and $h(P) \geq h(N)$.

It remains to prove that for every two F-normal forms N_1 and N_2 there exists a *distinguishing valuation*, i.e., a valuation $*$ such that if N_1 and N_2 are *not* provably equal, then the $*$-interpretations of N_1 and N_2 are distinct. Stating it contrapositively, for every two F-normal forms N_1 and N_2, it suffices to establish the existence of a valuation $* : \mathcal{V} \to \mathbf{P}_f$ such that

$$\text{if } [\![N_1]\!]_* = [\![N_2]\!]_*, \text{ then } N_1 \approx_{\mathrm{F}} N_2. \tag{2}$$

The idea is to use a valuation $*$ that assigns processes to variables in such a way that much of the original syntactic structure of N_1 and N_2 can be recovered by analysing the behaviour of $[\![N_1]\!]_*$ and $[\![N_2]\!]_*$. To recognize variables, we shall use the special processes φ_i ($i \geq 1$) defined in Eqn. (1) on p. 498. Recall that the processes φ_i have branching degree i. We are going to assign to every variable a distinct process φ_i. By choosing i larger than the maximal 'branching degrees' occurring in N_1 and N_2, the behaviour contributed by an instantiated variable is distinguished from behaviour already present in the F-normal forms themselves.

Definition 17. We define the *width* $w(N)$ of an F-normal form N as follows:

(i) if $N = \mathbf{0}$, then $w(N) = 0$;
(ii) if $N = N_1 + N_2$, then $w(N) = w(N_1) + w(N_2)$;
(iii) if $N = \alpha.N'$, then $w(N) = \max(w(N'), 1)$;
(iv) if $N = x \between N'$, then $w(N) = \max(w(N'), 1)$.

The valuation $*$ that we now proceed to define is parametrised with a natural number W; in Theorem 21 we shall prove that it serves as a distinguishing valuation (i.e., satisfies Eqn. (2)) for all F-normal forms N_1 and N_2 such that $w(N_1), w(N_2) \leq W$. Let $\ulcorner \_ \urcorner$ denote an *injective* function

$$\ulcorner \_ \urcorner : \mathcal{V} \to \{n \in \omega : n > W\}$$

that associates with every variable a unique natural number greater than W. We define the valuation $* : \mathcal{V} \to \mathbf{P}_f$ for all $x \in \mathcal{V}$ by

$$*(x) = \tau.\varphi_{\ulcorner x \urcorner} \ .$$

The τ-prefix is to ensure the following property.

Lemma 18. For every F-normal form N, $bdeg([\![N]\!]_*) \leq w(N)$.

Lemma 19. Let S be a simple F-normal form, let $\alpha \in \mathcal{A}_\tau$, and let p be a process such that $[\![S]\!]_* \xrightarrow{\alpha} p$. Then the following statements hold:

(i) if $S = \beta.N$, then $\alpha = \beta$ and $p = [\![N]\!]_*$;
(ii) if $S = x \between N$, then $\alpha = \tau$ and $p = \varphi_{\ulcorner x \urcorner} \parallel [\![N]\!]_*$.

An important property of $*$ is that it allows us to distinguish the different types of simple F-normal forms by classifying their residuals according to the number of parallel components with a branching degree that exceeds W. Let us say that a process p is of *type* n ($n \geq 0$) if its unique parallel decomposition contains precisely n parallel prime components with a branching degree $> W$.

Corollary 20. Let S be a simple F-normal form such that $w(S) \leq W$.

(i) If $S = \alpha.N$, then the unique residual $[\![N]\!]_*$ of $[\![S]\!]_*$ is of type 0.
(ii) If $S = x \between N$, then the unique residual $\varphi_{\ulcorner x \urcorner} \parallel [\![N]\!]_*$ of $[\![S]\!]_*$ is of type 1.

Theorem 21. For every two F-normal forms N_1, N_2 such that $w(N_1), w(N_2) \leq W$ it holds that $[\![N_1]\!]_* = [\![N_2]\!]_*$ only if $N_1 \approx N_2$ modulo A1–A4.

Proof. By Lemma 5 we may assume that N_1 and N_2 are summations of collections of simple F-normal forms. We assume $[\![N_1]\!]_* = [\![N_2]\!]_*$ and prove that then $N_1 \approx N_2$ modulo A1–A4, by induction on the sum of the heights of N_1 and N_2.

We first prove that for every syntactic summand S_1 of N_1 there is a syntactic summand S_2 of N_2 such that $S_1 \approx S_2$ modulo A1–A4. To this end, let S_1 be an arbitrary syntactic summand of N_1; we distinguish cases according to the syntactic form of S_1.

1. Suppose $S_1 = \alpha.N_1'$; then $[\![S_1]\!]_* \xrightarrow{\alpha} [\![N_1']\!]_*$. Hence, since $[\![N_1]\!]_* = [\![N_2]\!]_*$, there exists a syntactic summand S_2 of N_2 such that $[\![S_2]\!]_* \xrightarrow{\alpha} [\![N_1']\!]_*$. By Lemma 18 the branching degree of $[\![N_1']\!]_*$ does not exceed W, so $[\![S_2]\!]_*$ has a residual of type 0, and therefore, by Corollary 20, there exist $\beta \in \mathcal{A}_\tau$ and a normal form N_2' such that $S_2 = \beta.N_2'$. Moreover, since $[\![S_2]\!]_* \xrightarrow{\alpha} [\![N_1']\!]_*$, it follows by Lemma 19(i) that $\alpha = \beta$ and $[\![N_1']\!]_* = [\![N_2']\!]_*$. Hence, by the induction hypothesis, we conclude that $N_1' \approx N_2'$ modulo A1–A4, so $S_1 = \alpha.N_1' \approx \beta.N_2' = S_2$.
2. Suppose $S_1 = x \between N_1'$; then $[\![S_1]\!]_* \xrightarrow{\tau} \varphi_{\ulcorner x \urcorner} \parallel [\![N_1']\!]_*$. Hence, since $[\![N_1]\!]_* = [\![N_2]\!]_*$, there exists a summand S_2 of N_2 such that $[\![S_2]\!]_* \xrightarrow{\tau} \varphi_{\ulcorner x \urcorner} \parallel [\![N_1']\!]_*$. Since S_2 has a residual of type 1, by Corollary 20 there exist a variable y and a normal form N_2' such that $S_2 = y \between N_2'$. Now, since $[\![S_2]\!]_* \xrightarrow{\tau} \varphi_{\ulcorner x \urcorner} \parallel [\![N_1']\!]_*$, it follows by Lemma 19(ii) that

$$\varphi_{\ulcorner x \urcorner} \parallel [\![N_1']\!]_* = \varphi_{\ulcorner y \urcorner} \parallel [\![N_2']\!]_* \ . \tag{3}$$

Since $[\![N_1']\!]_*$ and $[\![N_2']\!]_*$ are of type 0, we have that the unique decomposition of $[\![N_1']\!]_*$ (see Theorem 10) does not contain $\varphi_{\ulcorner y \urcorner}$ and the unique decomposition of $[\![N_2']\!]_*$ does not contain $\varphi_{\ulcorner x \urcorner}$. Hence, from (3) it follows that $\varphi_{\ulcorner x \urcorner} = \varphi_{\ulcorner y \urcorner}$ and $[\![N_1']\!]_* = [\![N_2']\!]_*$. From the former we conclude, by Lemma 13(ii) and the injectivity of $\ulcorner . \urcorner$, that $x = y$ and from the latter we conclude by the induction hypothesis that $N_1' \approx N_2'$ modulo A1–A4. So $S_1 = x \between N_1' \approx y \between N_2' = S_2$.

We have established that every syntactic summand of N_1 is provably equal to a syntactic summand of N_2. Similarly, it follows that every syntactic summand of N_2 is provably equal to a syntactic summand of N_2. Hence, modulo A1–A4, $N_1 \approx N_1 + N_2 \approx N_2$, so the proof of the theorem is complete. □

Note that, by instantiating the parameter W with a sufficiently large value, it follows from the preceding theorem that there exists a distinguishing valuation for *every* pair of F-normal forms N_1 and N_2. Thus, we get the following corollary.

Corollary 22. For all process terms P and Q, $P \approx_{\mathrm{F}} Q$ if, and only if, $P \leftrightarrows_f Q$.

4 An Equational Base for $\mathbf{P}_h$

We now consider the algebra $\mathbf{P}_h$. Note that if $\mathcal{A}$ happens to be the empty set, then $\mathbf{P}_h$ satisfies the axiom F, and it is clear from the proof in the previous section that the axioms generated by the axiom schemata in Table 2 together with F in fact constitute a finite equational base for $\mathbf{P}_h$. We therefore proceed with the assumption that $\mathcal{A}$ is nonempty, and prove that an equational base for $\mathbf{P}_h$ is then obtained if we add the axiom

$$\mathrm{H} \quad x \mid (y \mid z) \approx \mathbf{0}$$

to the set of axioms generated by the axiom schemata in Table 2. Again, the resulting equational base is finite if the set $\mathcal{A}$ is finite.

Henceforth, whenever we write $P \approx_{\mathrm{H}} Q$, we shall mean that the equation $P \approx Q$ is derivable from the axioms in Table 2 and the axiom H.

Proposition 23. For all process terms P and Q, if $P \approx_{\mathrm{H}} Q$, then $P \leftrightarrows_h Q$.

We proceed to adapt the proof presented in the previous section to establish the converse of Proposition 23. Naturally, with H instead of F not every occurrence of $\mid$ can be eliminated from process terms; we therefore need to adapt the notion of normal form.

Definition 24. The set $\mathcal{N}_{\mathrm{H}}$ of H-*normal forms* is generated by:

$$N ::= \mathbf{0} \mid N + N \mid \alpha.N \mid x \between N \mid (x \mid a) \between N \mid (x \mid y) \between N \ ,$$

with $x, y \in \mathcal{V}$, $\alpha \in \mathcal{A}_\tau$ and $a \in \mathcal{A}$.

Lemma 25. For every process term P there exists an H-normal form N such that $P \approx_{\mathrm{H}} N$ and $h(P) \geq h(N)$.

We proceed to establish that for every two H-normal forms N_1 and N_2 there exists a valuation $* : \mathcal{V} \to \mathbf{P}_h$ such that

$$\text{if } [\![N_1]\!]_* = [\![N_2]\!]_*, \text{ then } N_1 \approx_{\mathrm{H}} N_2. \tag{4}$$

The distinguishing valuations $*$ will have a slightly more complicated definition than before, because of the more complicated notion of normal form.

As in the previous section, the definition of $*$ is parametrised with a natural number W. Since $|$ may occur in H-normal forms, we now also need to make sure that whatever process $*$ assigns to variables has sufficient communication abilities. To achieve this, we also parametrise $*$ with a finite subset $\mathcal{A}' = \{a_1, \ldots, a_n\}$ of $\mathcal{A}$ that is closed under the bijection $\overline{\_}$ on $\mathcal{A}$. (Note that every finite subset of $\mathcal{A}$ has a finite superset with the aforementioned property.) Based on W and $\mathcal{A}'$ we define the valuation $* : \mathcal{V} \to \mathbf{P}_h$ by

$$*(x) = a_1.\varphi_{(1 \cdot \ulcorner x \urcorner)} + \cdots + a_n.\varphi_{(n \cdot \ulcorner x \urcorner)} \ .$$

We shall prove that $*$ satisfies Eqn. (4) provided that the actions occurring in N_1 and N_2 are in $\mathcal{A}' \cup \{\tau\}$ and the width of N_1 and N_2, defined below, does not exceed W. We must also be careful to define the injection $\ulcorner \_ \urcorner$ in such a way that the extra factors $1, \ldots, n$ in the definition of $*$ do not interfere with the numbers assigned to variables; we let $\ulcorner \_ \urcorner$ denote an injection

$$\ulcorner \_ \urcorner : \mathcal{V} \to \{m : m \text{ a prime number such that } m > n \text{ and } m > W\}$$

that associates with every variable a prime number greater than the cardinality of $\mathcal{A}'$ and greater than W.

The definition of width also needs to take into account the cardinality of $\mathcal{A}'$ to maintain that the maximal branching degree in $[\![N]\!]_*$ does not exceed $w(N)$.

Definition 26. We define the *width* $w(N)$ of an H-normal form N as follows:

(i)–(iii) see Definition 17(i–iii).
(iv) if $N = x \between N'$, then $w(N) = \max(w(N'), n)$;
(v) if $N = (x \mid \alpha) \between N'$, then $w(N) = \max(w(N'), 1)$; and
(vi) if $N = (x \mid y) \between N'$, then $w(N) = \max(w(N'), n)$.

Lemma 27. For every H-normal form N, $bdeg([\![N]\!]_*) \leq w(N)$.

Lemma 28. Let S be a simple H-normal form, let $\alpha \in \mathcal{A}_\tau$, and let p be a process such that $[\![S]\!]_* \xrightarrow{\alpha} p$. Then the following statements hold:

(i) if $S = \beta.N$, then $\alpha = \beta$ and $p = [\![N]\!]_*$;
(ii) if $S = x \between N$, then $\alpha = a_i$ and $p = \varphi_{i \cdot \ulcorner x \urcorner} \parallel [\![N]\!]_*$ for some $i \in \{1, \ldots, n\}$;
(iii) if $S = (x \mid a) \between N$, then $\alpha = \tau$ and $p = \varphi_{i \cdot \ulcorner x \urcorner} \parallel [\![N]\!]_*$ for the unique $i \in \{1, \ldots, n\}$ such that $\overline{a} = a_i$; and
(iv) if $S = (x \mid y) \between N$, then $\alpha = \tau$ and $p = \varphi_{i \cdot \ulcorner x \urcorner} \parallel \varphi_{j \cdot \ulcorner y \urcorner} \parallel [\![N]\!]_*$ for some $i, j \in \{1, \ldots, n\}$ such that $\overline{a_i} = a_j$.

As in the previous section, we distinguish H-normal forms by classifying their residuals according to the number of parallel components with a branching degree that exceeds W.

Corollary 29. Let S be a simple H-normal form such that $w(S) \leq W$ and such that the actions occurring in S are included in $\mathcal{A}' \cup \{\tau\}$.

(i) If $S = \alpha.N$, then the unique residual of $[\![S]\!]_*$ is of type 0.
(ii) If $S = x \between N$, then all residuals of $[\![S]\!]_*$ are of type 1.
(iii) If $S = (x \mid a) \between N$, then the unique residual of $[\![S]\!]_*$ is of type 1.
(iv) If $S = (x \mid y) \between N$, then all residuals of $[\![S]\!]_*$ are of type 2.

Theorem 30. For every two H-normal forms N_1, N_2 such that $w(N_1), w(N_2) \leq W$ and such that the actions occurring in N_1 and N_2 are included in $\mathcal{A}' \cup \{\tau\}$ it holds that $[\![N_1]\!]_* = [\![N_2]\!]_*$ only if $N_1 \approx N_2$ modulo A1–A4, C5.

Proof. The proof of this theorem is very similar to the proof of Theorem 21, only there are two more cases to consider and the reasoning is slightly more complex due to the more complex definition of $*$. □

Corollary 31. For all process terms P and Q, $P \approx_{\mathrm{H}} Q$ if, and only if, $P \leftrightarrows_h Q$.

References

1. L. Aceto, W. J. Fokkink, A. Ingolfsdottir, and B. Luttik. CCS with Hennessy's merge has no finite equational axiomatization. *Theor. Comput. Sci.*, 330(3): 377–405, 2005.
2. L. Aceto, W. J. Fokkink, A. Ingolfsdottir, and B. Luttik. Finite equational bases in process algebra: Results and open questions. In A. Middeldorp, V. van Oostrom, F. van Raamsdonk, and R. C. de Vrijer, editors, *Processes, Terms and Cycles: Steps on the Road to Infinity*, LNCS 3838, pages 338–367. Springer, 2005.
3. J. A. Bergstra and J. W. Klop. Process algebra for synchronous communication. *Inform. and Control*, 60(1-3):109–137, 1984.
4. J. A. Bergstra and J. V. Tucker. Top-down design and the algebra of communicating processes. *Sci. Comput. Programming*, 5(2):171–199, 1985.
5. R. de Simone. Higher-level synchronising devices in Meije-SCCS. *Theor. Comput. Sci.*, 37:245–267, 1985.
6. J. F. Groote. A new strategy for proving ω-completeness applied to process algebra. In J. C. M. Baeten and J. W. Klop, editors, *Proceedings of CONCUR'90*, LNCS 458, pages 314–331. Springer, 1990.
7. J. Heering. Partial evaluation and ω-completeness of algebraic specifications. *Theoret. Comput. Sci.*, 43(2-3):149–167, 1986.
8. M. Hennessy. Axiomatising finite concurrent processes. *SIAM J. Comput.*, 17(5):997–1017, 1988.
9. M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *J. ACM*, 32(1):137–161, January 1985.
10. B. Luttik and V. van Oostrom. Decomposition orders—another proof of the fundamental theorem of arithmetic. *Theor. Comput. Sci.*, 335(2–3):147–186, 2005.
11. R. Milner. *Communication and Concurrency*. Prentice-Hall International, 1989.
12. F. Moller. *Axioms for Concurrency*. PhD thesis, University of Edinburgh, 1989.
13. F. Moller. The nonexistence of finite axiomatisations for CCS congruences. In *Proceedings of LICS'90*, pages 142–153. IEEE Computer Society Press, 1990.
14. D. M. R. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, 5^{th} *GI Conference*, LNCS 104, pages 167–183. Springer, 1981.
15. W. Taylor. Equational logic. *Houston J. Math.*, (Survey), 1979.