BRICS

Basic Research in Computer Science

BRICS RS-95-28   Aceto & Groote: Equational Axiomatization for MPA with String Iteration

A Complete Equational Axiomatization for MPA with String Iteration

Luca Aceto
Jan Friso Groote

BRICS Report Series RS-95-28

ISSN 0909-0878 May 1995

See back inner page for a list of recent publications in the BRICS Report Series. Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK - 8000 Aarhus C
Denmark

Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@brics.dk

BRICS publications are in general accessible through WWW and anonymous FTP:

`http://www.brics.dk/`
`ftp ftp.brics.dk (cd pub/BRICS)`

A Complete Equational Axiomatization for MPA with String Iteration

Luca Aceto[*]
BRICS[†]
Department of Mathematics and Computer Science, Aalborg University
Fredrik Bajersvej 7E, 9220 Aalborg Ø, Denmark
luca@iesd.auc.dk

Jan Friso Groote[‡]
Department of Philosophy, University of Utrecht
Heidelberglaan 8, 3584 CS Utrecht, The Netherlands
jfg@phil.ruu.nl

Abstract

We study equational axiomatizations of bisimulation equivalence for the language obtained by extending Milner's basic CCS with string iteration. String iteration is a variation on the original binary version of the Kleene star operation p^*q obtained by restricting the first argument to be a non-empty sequence of atomic actions. We show that, for every positive integer k, bisimulation equivalence over the set of processes in this language with loops of length at most k is finitely axiomatizable. We also offer a countably infinite equational theory that completely axiomatizes bisimulation equivalence over the whole language. We prove that this result cannot be improved upon by showing that no finite equational axiomatization of bisimulation equivalence over basic CCS with string iteration can exist, unless the set of actions is empty.

AMS Subject Classification (1991): 68Q40, 68Q42.
CR Subject Classification (1991): D.3.1, F.3.2, F.4.2.
Keywords and Phrases: Minimal Process Algebra, Kleene star, string iteration, equational logic, bisimulation equivalence.

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, because of their own nature, non-terminating, all process algebraic

[*] On leave from the School of Cognitive and Computing Sciences, University of Sussex. Partially supported by the HCM project EXPRESS.

[†] Basic Research in Computer Science, Centre of the Danish National Research Foundation.

[‡] This research has been partly carried out during a visit to Aalborg University.

specification formalisms (cf., e.g., [2, 20, 27, 4]) 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. For example, the recursion equation

$$X \overset{\text{def}}{=} \mathit{send} \cdot \mathit{receive} \cdot X \tag{1}$$

describes a system that is willing to perform alternatively the acts of sending and receiving *ad infinitum*. 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 [27, 4], the approximation induction principle [7] and ω-induction [19].)

The research literature on process theory has recently witnessed a resurgence of interest in the study of Kleene star-like operations as an alternative, purely algebraic, way of introducing infinite behaviours in process algebras. (Cf., e.g., the papers [14, 5, 6, 12, 10, 31, 9]). Some of these studies, notably [6], 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 [22, 8] rather than by means of systems of recursion equations. For example, using the original, binary version of the Kleene star operation from [22] studied in [6], the system described by the recursion equation (1) can alternatively be denoted by the term $(\mathit{send} \cdot \mathit{receive})^*\delta$, and, as shown in [6], any regular process can be specified in the axiom system ACP_τ with Kleene star using only handshake communication. (Interestingly, as already noted by Milner in [26, 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 spurred a flurry of research into the expressive power of equational logic to (finitely) axiomatize behavioural equivalences over simple languages incorporating variations on the Kleene star operation. Because of its central nature in process theory, bisimulation equivalence [29, 27] and variations on its theme have been prime candidates for this type of investigation. (Examples of contributions along this line of research may be found in, e.g., [14, 31, 12, 1, 13, 15].) A notable positive result in this direction was obtained by Fokkink and Zantema, who showed in [14] that the finite equational theory for the language of basic process algebra with iteration BPA^* proposed in [6] is indeed complete for bisimulation equivalence over that language. This remarkable result is in sharp contrast with a negative one later obtained by Sewell in [31]. Sewell shows that bisimulation equivalence can*not* be finitely axiomatized over the language BPA^*_δ obtained by adding the stopped process δ to the signature of BPA^*.

The catastrophic effect, for what concerns finite axiomatizability of bisimulation equivalence, of the addition of the, apparently innocuous, stopped process δ to BPA^* may be slightly disconcerting. Terms in the language BPA^* can denote process graphs with arbitrarily long loops, i.e., sequences of transitions leading back to the state they started from. However, any BPA^* term is *normed* in the sense of [3], i.e., any term is capable of terminating by performing a finite sequence of transitions. This implies that perpetual behaviours like, for instance, that specified by the recursion equation (1) cannot be denoted by any BPA^* term. It is the normedness of terms that is lost when δ is added to BPA^*.

In an attempt to reconcile finite axiomatizability of bisimulation with the presence of perpetual processes, Fokkink showed in [12] that bisimulation equivalence is finitely axiomatizable over a language obtained by adding prefix iteration to Milner's basic CCS. The language considered by Fokkink in the aforementioned reference allows for the specification of perpetual behaviours, but such behaviours must have loops of length exactly one.

It is our thesis that the cause of the impossibility of finitely axiomatizing bisimulation equivalence over BPA^*_δ is the combination of the loss of normedness and of the existence of unboundedly long loops in behaviours. Intuitively, unlike the case of normed processes, two perpetual behaviours can be bisimilar even if the lengths of their loops are very different. For example, the equality

$$(a^n)^*\delta \quad = \quad a^*\delta \tag{2}$$

holds for every positive integer n with respect to any equivalence in the linear time-branching time spectrum [16]. However, as it will be made clear by the proof of Thm. 35, any finite equational theory can only change the length of finitely many loops in terms, and thus cannot prove every equivalence of the form (2).

To provide a formal justification for our thesis, we have chosen to study the expressive power of equational logic in characterizing bisimulation equivalence over the simplest extension of Milner's basic CCS in which it is possible to express perpetual processes with unboundedly long loops. More precisely, we consider the subset of CCS consisting of the basic operations to denote finite synchronization trees, extended with *string iteration* w^*P. Intuitively, for a non-empty string of actions w, the term w^*P denotes a process that can execute the string of actions w repeatedly, and that, after each cycle along the w-loop, can decide to behave like P, if P can exhibit any transition at all. We denote the resulting language by $\mathsf{MPA}^{s*}(A)$. An example of a perpetual $\mathsf{MPA}^{s*}(A)$ term is the term $(\mathit{send} \cdot \mathit{receive})^*\delta$.

Our first main result of the paper is that bisimulation equivalence can be equationally axiomatized over the language $\mathsf{MPA}^{s*}(A)$ (cf. Thm. 29). We also prove that, for every positive integer k, bisimulation equivalence over the set of $\mathsf{MPA}^{s*}(A)$ terms with loops of length at most k has a finite equational axiomatization. Thus it is indeed the case that, at least over $\mathsf{MPA}^{s*}(A)$, bisimulation equivalence between perpetual processes with bounded loops can be finitely axiomatized. On the other hand, if the length of loops in process behaviours is not bounded from above, the equational theory that characterizes bisimulation equivalence over the whole of the language $\mathsf{MPA}^{s*}(A)$ is countably infinite. We show that this result cannot be improved upon by proving that no finite equational axiomatization of bisimulation equivalence over the language $\mathsf{MPA}^{s*}(A)$ can exist, unless the set of actions A is empty. (Cf. Thm. 35.) These positive and negative results are then extended to the language $\mathsf{BPA}^{s*}_{\delta\epsilon}(A)$ obtained by extending $\mathsf{MPA}^{s*}(A)$ with general sequential composition in lieu of action prefixing, and with the empty process from [23].

In the process of establishing the aforementioned results, we also obtain solutions to problems of independent interest. In particular, we present a novel, detailed analysis of bisimulation equivalence over a class of simple cyclic terms which we refer to as *terminal cycles* (cf. Lem. 22). As a technical tool for the study of equalities involving terminal cycles, we also prove a unique decomposition theorem over finite strings (cf. Thm. 2) which may be of independent string-theoretic interest.

We conclude this introduction by providing a brief road-map to the contents of this paper. Section 2 introduces the mathematical results and notations that will be used throughout the paper. In that section we also present a unique decomposition theorem for finite strings that will find application in the later developments of the paper. The language of minimal process algebra with string iteration and its operational semantics are introduced in Sect. 3. Section 4 is entirely devoted to detailed proofs of our completeness theorems for bisimulation equivalence over $\mathsf{MPA}^{s*}(A)$. Perhaps surprisingly, the proofs of the completeness theorems are rather involved, and, for this reason, we have chosen to present them in great detail in a style which is inspired by [24]. The non-existence of a finite equational axiomatization for bisimulation equivalence over the language $\mathsf{MPA}^{s*}(A)$ is presented in Sect. 5. Finally, Sect. 6 is devoted to extensions of our main results to the language $\mathsf{BPA}^{s*}_{\delta\epsilon}(A)$.

2 Preliminaries on Strings

We begin with some mathematical preliminaries. In particular, we present a basic result about finite strings which will find application in the remainder of this paper.

Let K be any set. The set of (possibly empty) finite sequences of elements of K, usually referred to as *strings*, is denoted by K^*. We shall use the symbols s, u, v, w, possibly subscripted and/or superscripted, to range over K^*. String concatenation will be denoted by juxtaposition; λ will be used to denote the empty string, and the sequence containing one element $a \in K$ will be written a. We use w^p, where w is a string in K^* and p is a non-negative integer, to denote the string $\underbrace{w \cdots w}_{p\text{-times}}$. By convention, w^0 will stand for the empty string λ. As it is standard practice, the set of non-empty strings over K will be denoted by K^+. The length of a string $w \in K^*$ will be denoted by $\mathsf{length}(w)$. We use $\mathbb{N} = \{1, 2, 3, \ldots\}$ for the set of natural numbers. Throughout the paper, the symbol $\triangleq$ will stand for "equals by definition". For integers p and q, not both zero, we shall write $\gcd(p, q)$ for the greatest common divisor of p and q.

The following notions about strings will be useful in the remainder of this paper.

Definition 1. Let $w \in K^*$. A *root decomposition* of w is a pair $(v, n) \in K^* \times \mathbb{N}$ such that $w = v^n$. The string w is a *prime root* iff $(w, 1)$ is its only root decomposition. A *prime root decomposition* of w is a root decomposition (v, n) for it with v a prime root string.

As an example, consider the string $w = abab$. Both $(w, 1)$ and $(ab, 2)$ are root decompositions of w. In fact, they are the only root decompositions for that string. As w admits a non-trivial root decomposition, i.e. $(ab, 2)$, it is not a prime root. On the other hand, the string ab is a prime root, and $(ab, 2)$ is therefore the unique prime root decomposition for w.

We shall now prove that, as it was the case in the above example, every string $w \in K^*$ has a *unique* prime root decomposition. Apart from its intrinsic string-theoretic interest, this result will be useful to establish an important lemma about the language for processes studied in this paper (cf. Lem. 19).

Theorem 2 *(Unique Prime Root Decomposition Theorem). Every string $s \in K^*$ has a unique prime root decomposition.*

The remainder of this section will be devoted to a proof of the above theorem. In the proof of this result, we shall make use of a result from number theory that may be found in, e.g., [28, Chapter 2]. Before stating this number-theoretic result, we recall a basic notion from number theory.

Definition 3. 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 $p \equiv q \pmod{m}$. In other words, $p \equiv q \pmod{m}$ iff $p = q + km$ for some integer k.

The following result pertaining to the solution of congruence equations will find an important application in the proof of Thm. 2. The interested reader is referred to, e.g., [28, Corollary 2.9] for its proof.

Theorem 4. *Let p, q, r be integers with p and q relatively prime, i.e. with $\gcd(p, q) = 1$, and with $q > 0$. Then the equation*

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

has a solution $x_1 \triangleq p^{\phi(q)-1}r$, where $\phi(q)$ is the positive integer associated with q by Euler's ϕ-function[1]*. All solutions are given by $x = x_1 + jq$, where $j = 0, \pm 1, \pm 2, \ldots$.*

Our application of the above number-theoretic result will be in the proof of the following result about strings.

Lemma 5. *Let $s \in K^+$. Assume that (u, n) and (v, m) are root decompositions of s. Then (w, k) is a root decomposition of s for some string w of length $\gcd(\mathsf{length}(u), \mathsf{length}(v))$.*

Proof. Let $s \in K^+$ and assume that (u, n) and (v, m) are both root decompositions of s. Let $p \triangleq \mathsf{length}(u)/\gcd(\mathsf{length}(u), \mathsf{length}(v))$ and $q \triangleq \mathsf{length}(v)/\gcd(\mathsf{length}(u), \mathsf{length}(v))$. Note, first of all, that p and q are well-defined positive integers because, as $s \in K^+$, both u and v are non-empty strings. Now, both u and v can be expressed (uniquely) as concatenations of, respectively, p and q substrings of length $\gcd(\mathsf{length}(u), \mathsf{length}(v))$. Therefore, let

$$u = u_1 \cdots u_p \tag{3}$$

$$v = v_1 \cdots v_q \tag{4}$$

[1] As stated in, e.g., [28, Thm. 2.5], $\phi(q)$ is the number of positive integers less than or equal to q that are relatively prime to q.

where each u_i $(1 \leq i \leq p)$ and v_j $(1 \leq j \leq q)$ is of length $\gcd(\mathsf{length}(u), \mathsf{length}(v))$. As (u, n) and (v, m) are both root decompositions of s, it follows from (3) and (4) above that, for every $l \geq 0$,

$$(u_1 \cdots u_p)^{ln} \quad = \quad (v_1 \cdots v_q)^{lm} \ . \tag{5}$$

We claim that the strings u_i and v_j are equal for every $1 \leq i \leq p$ and $1 \leq j \leq q$. To see that this indeed the case, consider arbitrary $i \in \{1, \ldots, p\}$ and $j \in \{1, \ldots, q\}$. The substring u_i occurs at positions

$$i + px \quad (0 \leq x \leq ln - 1)$$

in the sequence of strings $(u_1 \cdots u_p)^{ln}$. Similarly, the substring v_j occurs at positions

$$j + qy \quad (0 \leq y \leq lm - 1)$$

in the sequence of strings $(v_1 \cdots v_q)^{lm}$. If, for some non-negative integers x and y in the given ranges, we have that:

$$i + px \quad = \quad j + qy \tag{6}$$

then there is a position in the string $s^l = (u_1 \cdots u_p)^{ln} = (v_1 \cdots v_q)^{lm}$ at which the substrings u_i and v_j both occur. As $\mathsf{length}(u_i) = \mathsf{length}(v_j)$, u_i and v_j must be equal, as desired. So, all we have to do to prove that u_i is equal to v_j is to solve (6). This is where Thm. 4 comes into play. In fact, assume, without loss of generality, that $j \geq i$. Then, for all integers x and y,

$$\begin{aligned} i + px = j + qy \quad &\Leftrightarrow \quad px = (j - i) + qy \\ &\Leftrightarrow \quad px \equiv (j - i) \pmod{q} \ . \end{aligned}$$

Now, because of the way p and q are defined, it follows that $\gcd(p, q) = 1$. Therefore we may apply Thm. 4 to infer that the congruence equation

$$px \equiv (j - i) \pmod{q}$$

has an infinite family of solutions given by $x = x_1 + hq$ for $h = 0, \pm 1, \pm 2, \ldots$, where $x_1 \overset{\triangle}{=} p^{\phi(q)-1}(j - i)$. Now, pick non-negative integers $x \overset{\triangle}{=} x_1$ and y such that $px_1 = (j - i) + qy$. (Note that, as $x_1 \geq 0$ and $0 \leq j - i \leq q - 1$, it must be the case that $y \geq 0$.) Next, pick $l \geq 0$ such that $i + px_1 \leq ln$ and $j + qy \leq lm$. As x_1 and y are solutions of Eqn. 6, the strings u_i and v_j occur at the same position in the sequence

$$s^l = (u_1 \cdots u_p)^{ln} = (v_1 \cdots v_q)^{lm}$$

and are therefore equal. As this holds for every $i \in \{1, \ldots, p\}$ and $j \in \{1, \ldots, q\}$, it follows that all the u_is and v_js are equal. To finish the proof, it is thus sufficient to take $w \overset{\triangle}{=} u_1 (= u_2 = \cdots = u_p = v_1 = \cdots = v_q)$ and $k \overset{\triangle}{=} np (= mq)$. □

Having established the above result, we are now in a position to prove the promised unique prime root decomposition theorem.

Proof of Theorem 2. If s is the empty string, then $(s, 1)$ is its unique root decomposition, which is prime. Assume then that $s \in K^+$. Let $\{(v_1, n_1), \ldots, (v_k, n_k)\}$ be the non-empty, finite set of root decompositions of s. Choose $1 \leq i \leq k$ such that n_i is the maximum of the non-empty set of natural numbers $\{n_1, \ldots, n_k\}$. It follows that v_i is a prime root and that (v_i, n_i) is a prime root decomposition of s. This establishes the existence of such a root decomposition.

We now argue for the uniqueness of such a prime root decomposition. Assume, towards a contradiction, that (v_j, n_j) is a different prime root decomposition of s. It is immediate to note that $v_i \neq v_j$ and $n_i \neq n_j$. Moreover, as $n_i = \max\{n_1, \ldots, n_k\}$, it must be the case that $n_j < n_i$ and $\mathsf{length}(v_i) < \mathsf{length}(v_j)$. Then, as $s = v_i^{n_i} = v_j^{n_j}$, Lem. 5 gives the existence of a root decomposition (w, k) of s, with w a string of length $\gcd(\mathsf{length}(v_i), \mathsf{length}(v_j))$. As $\mathsf{length}(v_i) < \mathsf{length}(v_j)$, we have also that $\mathsf{length}(w) < \mathsf{length}(v_j)$ and that $v_j = w^p$ for some positive integer $p > 1$. This implies that (w, p) is a non-trivial root decomposition of v_j, contradicting the assumption that v_j were a prime root. This establishes the uniqueness of the prime root decomposition of s. □

To conclude this section, we present a result about prime root strings that will find application in the remainder of this paper. (Cf. Lem. 20.)

Lemma 6. *Let $s \in K^*$ and $a \in A$. Assume that the string as is a prime root. Then so is sa.*

Proof. We show the contrapositive statement. To this end, let $s \in K^*$, and $a \in A$. Assume that sa is not a prime root. We shall prove that as is not a prime root either.

If sa is not a prime root, then there exist a string $v \in K^+$ and a natural number $n > 1$ such that $sa = v^n$. As $sa = v^n$, v must be of the form ua for some string $u \in A^*$. Moreover it holds that $s = (ua)^{n-1}u$. Therefore $as = a(ua)^{n-1}u = (au)^n$. As $n > 1$, it follows immediately that as cannot be a prime root. □

3 Minimal Process Algebra with String Iteration

We assume a finite set A of observable actions. The symbols a, b, c will be used as typical members of this set. We also assume the existence of a countably infinite set Var of process variables, ranged over by x, y, z. The language of minimal process algebra with string iteration $\mathsf{MPA}^{s*}(A, \mathsf{Var})$ is given by the following grammar:

$$E ::= x \mid \delta \mid aE \mid E + E \mid w^*E$$

where $x \in \mathsf{Var}$, $a \in A$ and $w \in A^+$. We shall use E, F, G, possibly subscripted and/or superscripted to range over $\mathsf{MPA}^{s*}(A, \mathsf{Var})$. Instead of repeated action prefix, e.g. $a_1a_2 \ldots a_nE$, we shall often write wE with $w = a_1a_2 \ldots a_n$. By convention, $\lambda E \equiv E$, where the symbol $\equiv$ stands for syntactic equality of terms. When writing $\mathsf{MPA}^{s*}(A, \mathsf{Var})$ terms, we shall sometimes adopt the following parsing conventions: * binds stronger than a, which in turn binds stronger than $+$.

For a positive integer k, we say that a term E in $\mathsf{MPA}^{s*}(A, \mathsf{Var})$ is k-bounded if the length of any string w occurring in sub-terms of E of the form w^*F is smaller than or equal to k. For a term E, we shall sometimes refer to the smallest k for which E is k-bounded as the *iteration bound* of E, notation $\mathsf{IB}(E)$. For example, the iteration bound of the term $(ab)^*x + a^*y$ is 2.

The set of variables occurring in a term $E \in \mathsf{MPA}^{s*}(A, \mathsf{Var})$ will be denoted by $\mathsf{Var}(E)$. We shall write $\mathsf{MPA}^{s*}(A)$ for the set of closed terms, i.e. terms without occurrences of variables, in the language $\mathsf{MPA}^{s*}(A, \mathsf{Var})$. The symbols P, Q, R, T, possibly subscripted and/or superscripted, will be used to range over closed terms.

A (closed) substitution σ is a mapping from variables in Var to (closed) terms in $\mathsf{MPA}^{s*}(A, \mathsf{Var})$. For every $E \in \mathsf{MPA}^{s*}(A, \mathsf{Var})$ and substitution σ, we shall write $E\sigma$ for the result of substituting each variable x occurring in E with the term $\sigma(x)$. For every substitution σ, variable $x \in \mathsf{Var}$ and term $E \in \mathsf{MPA}^{s*}(A, \mathsf{Var})$, we shall use $\sigma[x \mapsto E]$ to denote the substitution mapping x to E, and agreeing with σ on all the other variables.

Definition 7. The *size* $|P|$ of an $\mathsf{MPA}^{s*}(A)$ term P is defined by structural recursion as follows:

$$\begin{aligned} |\delta| &\triangleq 1 \\ |w^*P| &\triangleq \mathsf{length}(w) + |P| \\ |aP| &\triangleq 1 + |P| \\ |P + Q| &\triangleq |P| + |Q| + 1 \end{aligned}$$

The operational semantics of the language $\mathsf{MPA}^{s*}(A)$ is given by the labelled transition system [21, 30]

$$(\mathsf{MPA}^{s*}(A), A, \{\xrightarrow{a} \mid a \in A\})$$

where the transition relations $\xrightarrow{a} \subseteq \mathsf{MPA}^{s*}(A) \times \mathsf{MPA}^{s*}(A)$ are the least ones satisfying the operational rules in Table 1.

The following trivial lemma ensures that, for every $k \in \mathbb{N}$, the language of k-bounded terms is closed under transitions. This property will be implicitly used in the proof of the completeness result for this set of terms.

Lemma 8. *Let P be a k-bounded $\mathsf{MPA}^{s*}(A)$ term. Assume that $P \xrightarrow{a} Q$. Then Q is also k-bounded.*

For each $w \in A^*$, we define the derived transition relation $\xrightarrow{w}$ by induction on $\mathsf{length}(w)$ in the standard way as follows:

- $P \xrightarrow{\lambda} Q$ iff $P \equiv Q$, and
- $P \xrightarrow{aw} Q$ iff $P \xrightarrow{a} P' \xrightarrow{w} Q$ for some $\mathsf{MPA}^{s*}(A)$ term P'.

$$
\begin{array}{|cc|}
\hline
aP \xrightarrow{a} P & \\[2ex]
\dfrac{P \xrightarrow{a} P'}{P + Q \xrightarrow{a} P'} & \dfrac{Q \xrightarrow{a} Q'}{P + Q \xrightarrow{a} Q'} \\[3ex]
(aw)^*P \xrightarrow{a} w(aw)^*P & \dfrac{P \xrightarrow{b} P'}{(aw)^*P \xrightarrow{b} P'} \\
\hline
\end{array}
$$

Table 1: The operational rules for $\mathsf{MPA}^{s*}(A)$ $(a, b \in A)$

For $\mathsf{MPA}^{s*}(A)$ terms P, Q, we say that Q is a *state* of P iff there exists a string $w \in A^*$ such that $P \xrightarrow{w} Q$. Not surprisingly, a straightforward argument by structural induction on $\mathsf{MPA}^{s*}(A)$ terms gives that:

Lemma 9. *For every $P \in \mathsf{MPA}^{s*}(A)$, the set of states of P is finite.*

The notion of behavioural equivalence over $\mathsf{MPA}^{s*}(A)$ terms that we shall consider in this paper is that of bisimulation equivalence, due to Park and Milner [29, 27].

Definition 10. A binary relation $\Re$ over $\mathsf{MPA}^{s*}(A)$ is a *bisimulation* iff it is symmetric and whenever $P \;\Re\; Q$, for all $a \in A$, $P' \in \mathsf{MPA}^{s*}(A)$:

$$\text{if } P \xrightarrow{a} P', \text{ then, for some } Q' \in \mathsf{MPA}^{s*}(A),\ Q \xrightarrow{a} Q' \text{ and } P' \;\Re\; Q'.$$

Two $\mathsf{MPA}^{s*}(A)$ terms P, Q are *bisimulation equivalent*, denoted by $P \underline{\leftrightarrow} Q$, iff there is a bisimulation relation $\Re$ such that $P \;\Re\; Q$.

Bisimulation equivalence can be extended to all of the language $\mathsf{MPA}^{s*}(A, \mathsf{Var})$ in the following standard fashion:

Definition 11. For all $E, F \in \mathsf{MPA}^{s*}(A, \mathsf{Var})$, $E \underline{\leftrightarrow} F$ iff $E\sigma \underline{\leftrightarrow} F\sigma$, for every closed substitution σ.

It is well known that $\underline{\leftrightarrow}$ is an equivalence relation, and, as the rules in Table 1 are in *tyft/tyxt*-format [17], it is also a congruence relation. Therefore it makes sense to form the quotient algebra $\mathsf{MPA}^{s*}(A)/\underline{\leftrightarrow}$. The following lemma states a cardinality property of the algebra $\mathsf{MPA}^{s*}(A)/\underline{\leftrightarrow}$ that will be useful in the developments in Sect. 5 (cf. Lem. 32).

Lemma 12. *If the set of actions A is non-empty, then the (carrier of the) quotient algebra $\mathsf{MPA}^{s*}(A)/\underline{\leftrightarrow}$ is countably infinite.*

Proof. If A is non-empty, then we can pick an action $a \in A$. For each natural number n, we may then define the term $P_n \triangleq a^n\delta$. It is trivial to see that, for all $n, m \in \mathbb{N}$, $P_n \underline{\leftrightarrow} P_m$ iff $n = m$. □

A1	$x + y = y + x$
A2	$(x + y) + z = x + (y + z)$
A3	$x + x = x$
A6	$x + \delta = x$
S1	$x + w(w^*x) = w^*x$
S2	$w^*(w^*x) = w^*x$
S3	$(w^n)^*\delta = w^*\delta$
S4	$a(ua)^*\delta = (au)^*\delta$

Table 2: The axioms for $\mathsf{MPA}^{s*}(A)$ ($w \in A^+$, $u \in A^*$ and $n \in \mathbb{N}$)

The main aim of this paper will be to provide a complete equational axiomatization of bisimulation equivalence over the language $\mathsf{MPA}^{s*}(A)$, and the whole of the next section will be devoted to this end.

4 An Equational Axiomatization for $\mathsf{MPA}^{s*}(A)$

The equational axioms in Table 2 will be shown to completely characterize the relation of bisimulation equivalence between $\mathsf{MPA}^{s*}(A)$ terms. We shall refer to this equational theory as MPA^{s*}. By analogy with our terminology for $\mathsf{MPA}^{s*}(A, \mathsf{Var})$ terms, in case the length of the strings w^n, w, ua and au mentioned in axioms S1–S5 does not exceed k we call these axioms k-bounded. In this case we refer to the resulting set of axioms as MPA^{s*}_k. Note that, as the set of actions A is finite, the family of axioms MPA^{s*}_k is finite, for every positive integer k. The equational theory MPA^{s*}, however, consists of a countably infinite collection of equations.

Definition 13. For an equational theory $\mathcal{E}$ over the signature of $\mathsf{MPA}^{s*}(A)$, we write $\mathcal{E} \vdash P = Q$ iff the equality $P = Q$ can be derived from those in $\mathcal{E}$ using the rules of equational logic. (For ease of reference, the inference rules of equational logic are collected in Table 3.) We also write $\mathcal{E} \vdash Q \subseteq P$, read "$Q$ is a provable summand of P" or "Q can be absorbed into P", iff $\mathcal{E} \vdash P + Q = P$. By convention, all the equational theories we shall consider in this section will contain axioms A1 and A2.

In equational proofs to follow, we shall find it useful to have the following equations at hand:

$$\begin{array}{lrcl} \text{S5} & (w^n)^*w^*\delta & = & w^*\delta \\ \text{S6} & w^*x & = & w^*x + x \\ \text{S7} & w^*x & = & w^*x + w(w^*x) \end{array}$$

Equations S6 and S7 are easily seen to be derivable from S1 and A3. Finally, note that equation S5 is derivable from axioms S2 and S3 as follows:

$$(w^n)^*w^*\delta \overset{\text{S3}}{=} (w^n)^*(w^n)^*\delta \overset{\text{S2}}{=} (w^n)^*\delta \overset{\text{S3}}{=} w^*\delta \ .$$

$$\frac{}{E = E} \qquad \frac{F = E}{E = F} \qquad \frac{E = F \quad F = G}{E = G}$$

$$\frac{}{E\sigma = F\sigma} \quad (E = F) \in \mathcal{E},\ \sigma \text{ a substitution}$$

$$\frac{E_i = F_i \ (1 \leq i \leq n)}{f(E_1, \ldots, E_n) = f(F_1, \ldots, F_n)} \quad f \text{ an operation symbol of arity } n$$

Table 3: The inference rules of equational logic

The first main result of this paper can now be stated.

Theorem 14. *The axiom system in Table 2 is sound and complete for bisimulation equivalence over the language* $\mathsf{MPA}^{s*}(A)$*, i.e. for all* $\mathsf{MPA}^{s*}(A)$ *terms* P *and* Q*,*

$$P \underline{\leftrightarrow} Q \quad \Leftrightarrow \quad \mathsf{MPA}^{s*} \vdash P = Q.$$

Moreover, for k*-bounded* $\mathsf{MPA}^{s*}(A)$*-terms* P *and* Q*, we find*

$$P \underline{\leftrightarrow} Q \quad \Leftrightarrow \quad \mathsf{MPA}^{s*}_k \vdash P = Q.$$

The remainder of this section will be devoted to a detailed proof of the above result. The proof we present is similar in spirit to those for related results given in, e.g., [18, 1, 13, 15], but the details will be rather different and, perhaps surprisingly, quite involved.

First of all, we establish the soundness of the equational theory MPA^{s*}.

Lemma 15 *(Soundness). For all* $\mathsf{MPA}^{s*}(A, \mathsf{Var})$ *terms* E, F*,* $\mathsf{MPA}^{s*} \vdash E = F$ *implies* $E \underline{\leftrightarrow} F$*.*

Proof. As bisimulation equivalence is a congruence with respect to the operators of the language $\mathsf{MPA}^{s*}(A, \mathsf{Var})$, the soundness of the axiom system MPA^{s*} can be shown by establishing that each axiom in MPA^{s*} is sound. This easy, but somewhat tedious, verification is left to the reader. □

We remark here that the soundness of equation S3 depends crucially on the fact that the argument of the string iteration construct is a stopped process, i.e., a process that cannot exhibit any transition. In fact, in general, $w^*P \not\underline{\leftrightarrow} (w^n)^*P$. As an example, consider $w \overset{\triangle}{=} a$, $n \overset{\triangle}{=} 2$ and $P \equiv b\delta$. Then $a^*P \not\underline{\leftrightarrow} (aa)^*P$ because $a^*P \xrightarrow{ab} \delta$, whereas $(aa)^*P$ cannot perform the string of actions ab.

The rest of this section is devoted to the much more challenging proof of the completeness of the equational theory MPA^{s*} with respect to bisimulation equivalence, which we shall approach in several intermediate stages.

Lemma 16. *Let* P *be an* $\mathsf{MPA}^{s*}(A)$ *term such that* $P \underline{\leftrightarrow} \delta$*. Then* $A6 \vdash P = \delta$*.*

Proof. A straightforward induction on the structure of P. □

The following definition introduces a class of processes in $\mathsf{MPA}^{s*}(A)$ that will play an important role in the proof of our completeness theorem.

Definition 17. An $\mathsf{MPA}^{s*}(A)$ term P is called a *terminal cycle* if $P \underline{\leftrightarrow} w^*\delta$ for some string $w \in A^+$. An $\mathsf{MPA}^{s*}(A)$ term P is *terminal* iff $P \underline{\leftrightarrow} \delta$ or P is a terminal cycle.

For example, an $\mathsf{MPA}^{s*}(A)$ term of the form $u(wu)^*\delta$, where at least one of the strings u, v is non-empty, is a terminal cycle because it is bisimulation equivalent to the term $(uw)^*\delta$.

Lemma 18. *Let P be an $\mathsf{MPA}^{s*}(A)$ term and $u \in A^*$. If the term uP is a terminal cycle, then so is P.*

Proof. We prove the thesis by induction on the length of the string u. If u is empty, then the lemma follows immediately by convention. Otherwise, assume that $u = au'$ and uP is a terminal cycle. This means that $uP \underline{\leftrightarrow} v^*\delta$ for some $v \in A^+$. As $uP \xrightarrow{a} u'P$, it must be the case that $v = av'$ for some $v' \in A^*$, and that $u'P \underline{\leftrightarrow} v'(av')^*\delta$. Repeated applications of the sound equation S4 give that $u'P \underline{\leftrightarrow} (v'a)^*\delta$. Hence, the term $u'P$ is itself a terminal cycle, and, by the inductive hypothesis, so is P. □

We remark here that the property of being a terminal cycle is not preserved by the action prefixing operation, i.e. the converse of the above lemma does not hold. For example, the term $a^*\delta$ is obviously a terminal cycle, but $b(a^*\delta)$ is not.

The following result, whose proof uses Thm. 2 in an essential way, states an important property of terminal cycles that will be used in proving the completeness of our axiom system for that sub-class of processes.

Lemma 19. *Let $v, w \in A^+$ and $P \in \mathsf{MPA}^{s*}(A)$. Suppose that $v^*P \underline{\leftrightarrow} w^*\delta$. Then there exist a prime root string $s \in A^+$ and natural numbers k, h such that $v = s^k$ and $w = s^h$.*

Proof. Assume that $v^*P \underline{\leftrightarrow} w^*\delta$. Let $n \triangleq \mathsf{length}(w)$ and $m \triangleq \mathsf{length}(v)$. As $v^*P \underline{\leftrightarrow} w^*\delta$, it follows that $v^n = w^m$. By Thm. 2, the strings v and w have unique prime root decompositions (s_v, k_v) and (s_w, k_w), respectively. Therefore the string $v^n (= u^m)$ has prime root decompositions (s_v, nk_v) and (s_w, mk_w). Again by Thm. 2, these two root decompositions must be equal, i.e., it must be the case that $s_v = s_w$ and $nk_v = mk_w$. To establish the thesis, it is thus sufficient to take $s \triangleq s_v (= s_w)$, $k \triangleq k_v$ and $h \triangleq k_w$. □

Lemma 20. *Let P be a k-bounded $\mathsf{MPA}^{s*}(A)$ term. Suppose that P is a terminal cycle. Then $P \underline{\leftrightarrow} s^*\delta$ for some prime root string s of length at most k.*

Proof. Assume that P is a k-bounded terminal cycle. Then there exists a string $v \in A^+$ such that $P \underline{\leftrightarrow} v^*\delta$. Let (s, n) be the unique prime root decomposition for v given by Thm. 2. Using the sound equation S3, we derive that $P \underline{\leftrightarrow} s^*\delta$. The thesis now follows from the following claim:

Claim. For every k-bounded term P and prime root string s,

$$P \leftrightarroweq s^*\delta \text{ implies } \mathsf{length}(s) \leq k \ .$$

This claim can be proven by structural induction on P. We only examine two of the possible forms P may take.

- Case: $P \equiv aQ$.

  As $P \equiv aQ \leftrightarroweq s^*\delta$, it is not hard to see that it must be the case that $s = as'$ and $Q \leftrightarroweq (s'a)^*\delta$ for some string s'. By Lem. 6, the string $s'a$ is also a prime root. As Q is itself k-bounded, we may apply the inductive hypothesis to derive that $\mathsf{length}(as') = \mathsf{length}(s'a) \leq k$.

- Case: $P \equiv v^*Q$.

  As $P \equiv v^*Q \leftrightarroweq s^*\delta$ and s is a prime root, we may apply Lem. 19 to derive that $v = s^h$ for some natural number h. As P is k-bounded, the length of the string v is at most k. Thus $\mathsf{length}(s) \leq k$ as desired.

□

Using the above results, we are now in a position to establish the completeness of our axiom system for terminal cycles.

Notation 21. We shall use $\mathsf{TerCycles}$ to denote the equational theory consisting of equations A3,A6,S2,S3,S4. As usual, for every positive integer k, $\mathsf{TerCycles}_k$ will stand for the equational theory consisting of the k-bounded instances of equations in $\mathsf{TerCycles}$.

Lemma 22. *Let $P \in \mathsf{MPA}^{s*}(A)$ and $w \in A^+$. Assume that $P \leftrightarroweq w^*\delta$. Then $\mathsf{TerCycles} \vdash P = w^*\delta$. Moreover, if P and $w^*\delta$ are k-bounded, then $\mathsf{TerCycles}_k \vdash P = w^*\delta$.*

Proof. Assume that $P \leftrightarroweq w^*\delta$ for some $w \in A^+$. We prove the thesis by induction on the structure of P. We proceed by a case analysis on the form P may take.

- Case: $P \equiv \delta$.

  Vacuous, as $P \not\leftrightarroweq w^*\delta$.

- Case: $P \equiv aP_1$.

  As $P \equiv aP_1 \leftrightarroweq w^*\delta$, the string w must be of the form aw' for some $w' \in A^*$. In this case, it is not hard to see that $P_1 \leftrightarroweq (w'a)^*\delta$. By the inductive hypothesis, it follows that $\mathsf{TerCycles} \vdash P_1 = (w'a)^*\delta$. So, $\mathsf{TerCycles} \vdash P = a(w'a)^*\delta \overset{\text{S4}}{=} (aw')^*\delta$, as desired.

- Case: $P \equiv P_1 + P_2$.

  As $P \leftrightarroweq w^*\delta$, it is easy to see that one of the following three cases must hold:

  1. $P_1 \leftrightarroweq \delta$ and $P_2 \leftrightarroweq w^*\delta$, or

2. $P_2 \leftrightarroweq \delta$ and $P_1 \leftrightarroweq w^*\delta$, or
3. both P_1 and P_2 are bisimulation equivalent to $w^*\delta$.

The first two cases can be easily dealt with using Lem. 16, the inductive hypothesis and axiom A6. In the last case, using the induction hypothesis, we have that $\mathsf{TerCycles} \vdash P_i = w^*\delta$ for $i = 1, 2$. So, $\mathsf{TerCycles} \vdash P \equiv P_1 + P_2 = w^*\delta + w^*\delta \overset{\text{A3}}{=} w^*\delta$.

- Case: $P \equiv v^*P_1$, for some $v \in A^+$.

  As $P \leftrightarroweq w^*\delta$, by Lem. 19 there exists a prime root string s and natural numbers k, h such that $v = s^k$ and $w = s^h$. We proceed by distinguishing two cases depending on whether P_1 has any transition or not.

  1. Case: $P_1 \xrightarrow{a} P_1'$ for no $a \in A$ and $P_1' \in \mathsf{MPA}^{s*}(A)$.
     In this case $P_1 \leftrightarroweq \delta$. Now Lem. 16 gives that $A6 \vdash P_1 = \delta$. Therefore,

     $$\begin{array}{rcl} \mathsf{TerCycles} \vdash \ P & \overset{\text{A6}}{=} & v^*\delta \\ & = & (s^k)^*\delta \\ & \overset{\text{S3}}{=} & s^*\delta \\ & \overset{\text{S3}}{=} & (s^h)^*\delta \\ & = & w^*\delta \end{array}$$

     and we are done.
  2. Case: $P_1 \xrightarrow{a} P_1'$ for some $a \in A$ and $P_1' \in \mathsf{MPA}^{s*}(A)$.
     We claim that, in this case, $P_1 \leftrightarroweq w^*\delta$. To see that this does hold, note that, as $P_1 \xrightarrow{a} P_1'$ and $v^*P_1 \leftrightarroweq w^*\delta$, it must be the case that $w = aw'$ for some $w' \in A^*$, and that $P_1' \leftrightarroweq w'(aw')^*\delta \leftrightarroweq (w'a)^*\delta$. Indeed, whenever $P_1 \xrightarrow{b} P'$ for some action b and $\mathsf{MPA}^{s*}(A)$ process P', it follows that $a = b$ and $P' \leftrightarroweq (w'a)^*\delta$. This implies that $P_1 \leftrightarroweq a(w'a)^*\delta \leftrightarroweq (aw')^*\delta$, as claimed.
     Now, as $P_1 \leftrightarroweq w^*\delta$, an application of the inductive hypothesis gives $\mathsf{TerCycles} \vdash P_1 = w^*\delta$. Therefore,

     $$\begin{array}{rcl} \mathsf{TerCycles} \vdash P & = & v^*(w^*\delta) \\ & = & (s^k)^*((s^h)^*\delta) \\ & \overset{\text{S3}}{=} & (s^k)^*(s^*\delta) \\ & \overset{\text{S5}}{=} & s^*\delta \\ & \overset{\text{S3}}{=} & (s^h)^*\delta \\ & = & w^*\delta \end{array}$$

     and we are done.

An inspection of the above proof shows that only k-bounded instances of the equations S2–S4 need be used in the proof if P and $w^*\delta$ are both k-bounded. This completes the inductive argument, and the proof of the lemma. □

As an easy corollary of the above lemma, we can now prove that the equational theory $\mathsf{TerCycles}$ completely characterizes bisimulation equivalence over terminal cycles.

Corollary 23. *Let P, Q be terms in $\mathsf{MPA}^{s*}(A)$. Assume that $P \leftrightarroweq Q$ and P is a terminal cycle. Then $\mathsf{TerCycles} \vdash P = Q$. Moreover, if P and Q are k-bounded, $\mathsf{TerCycles}_k \vdash P = Q$.*

Proof. Assume that $P \leftrightarroweq Q$ and P is a terminal cycle. By Def. 17, this means that $P \leftrightarroweq w^*\delta$ for some string $w \in A^+$. By Lem. 22 and the fact that bisimulation equivalence is an equivalence relation, we have that $\mathsf{TerCycles} \vdash P = w^*\delta = Q$, from which the thesis follows immediately by transitivity. An identical argument shows that if P and Q are k-bounded, then $\mathsf{TerCycles}_k \vdash P = Q$. In fact, by Lem. 20, in that case we can choose w to be a prime root string of length at most k. □

Terminal cycles are processes with very simple cyclic behaviour. Surprisingly, however, to the best of our knowledge, the results we have presented so far give the first systematic investigation of their properties in the literature. In particular, the completeness result in Corollary 23 appears to be new. As terminal cycles are *strongly determinate* processes in the sense of [27, Def. 11.2], by the main result in [11] and [27, Propn. 11.5], all the equivalences in the linear time-branching time spectrum considered in [16] coincide over the set of such processes. As a result, Corollary 23 gives a complete axiomatization of all the equivalences in van Glabbeek's study [16] over terminal cycles.

The reader might have noticed that in establishing our completeness theorem for bisimulation equivalence over terminal cycles, we have never needed to use equation S1. This equation will, however, play a crucial rôle in the extension of the completeness result to the whole of the language $\mathsf{MPA}^{s*}(A)$. This we now proceed to present.

Definition 24. For every $\mathsf{MPA}^{s*}(A)$ term P we define its minimum distance to a terminal term by:

$$|P|_{min} \quad \stackrel{\triangle}{=} \quad \min\{\mathsf{length}(w) \mid \exists P' : \; P \xrightarrow{w} P' \text{ and } P' \text{ is a terminal term}\} \; .$$

For example, $|P|_{min} = 0$ iff P is a terminal term, and $|aa\delta + a^*\delta|_{min} = 1$ because $aa\delta + a^*\delta \xrightarrow{a} a^*\delta$ is the shortest sequence of transitions from $aa\delta + a^*\delta$ leading to a terminal term.

The following lemma collects some basic properties of the minimum distance to terminal terms that will find application in the proof of the completeness theorem.

Lemma 25.

1. *For every $\mathsf{MPA}^{s*}(A)$ term P, $|P|_{min} \in \mathbb{N}$, i.e. $|\cdot|_{min}$ is well defined.*
2. *Let P and Q be $\mathsf{MPA}^{s*}(A)$ terms. If $P \leftrightarroweq Q$, then $|P|_{min} = |Q|_{min}$.*
3. *Let P be an $\mathsf{MPA}^{s*}(A)$ term and let $w \in A^*$. Assume that $|P|_{min} > 0$. Then $|wP|_{min} = \mathsf{length}(w) + |P|_{min}$.*

Proof. Statements 1 and 2 of the lemma are easily seen to hold. Here we present a proof of statement 3 by induction on the length of the string w.

Suppose that P is an $\mathsf{MPA}^{s*}(A)$ term such that $|P|_{min} > 0$. Let $w \in A^*$. We aim at showing that $|wP|_{min} = \mathsf{length}(w) + |P|_{min}$. This is immediate if w is the empty string λ. For the inductive step, assume that $w = aw'$ for some action a and string w'. By the inductive hypothesis, we infer that:

$$|w'P|_{min} \quad = \quad \mathsf{length}(w') + |P|_{min} \ . \tag{7}$$

Note moreover that, as $|P|_{min} > 0$, by Lem. 18 the term $aw'P$ cannot be terminal. We can now reason as follows:

$$\begin{aligned}
|aw'P|_{min} \quad &= \quad \min\{\mathsf{length}(v) \mid \exists P' :\ aw'P \xrightarrow{v} P' \text{ and } P' \text{ is a terminal term}\} \\
&\qquad\quad (\text{Definition of } |\cdot|_{min}) \\
&= \quad \min\{\mathsf{length}(av) \mid \exists P' :\ w'P \xrightarrow{v} P' \text{ and } P' \text{ is a terminal term}\} \\
&\qquad\quad (aw'P \text{ is not a terminal term}) \\
&= \quad \min\{\mathsf{length}(aw'u) \mid \exists P' :\ P \xrightarrow{u} P' \text{ and } P' \text{ is a terminal term}\} \\
&\qquad\quad (7) \\
&= \quad \mathsf{length}(aw') + |P|_{min} \\
&\qquad\quad (\text{Definition of } |\cdot|_{min})
\end{aligned}$$

and the inductive step follows. □

It is interesting to note that statement 3 in the previous lemma does not hold if the term P is a terminal cycle. As an example witnessing this fact, consider the term $a(a^*\delta)$. Then $|a(a^*\delta)|_{min} = 0 \neq 1 + |a^*\delta|_{min}$.

The following result is a variation on the so-called *absorption lemma*, a standard tool in proofs of completeness theorems for process algebras (cf., e.g., [18, 27]), and we shall use it heavily in the proof of our main result.

Lemma 26 *(Operational Completeness). Let P be a $\mathsf{MPA}^{s*}(A)$ term. If $P \xrightarrow{a} P'$ then the two following statements hold:*

- *$A3, S1 \vdash aP' \subseteq P$, and*
- *if $|P'| \geq |P|$ then there exist a string $w \in A^*$ and an $\mathsf{MPA}^{s*}(A)$ term P'' such that $P' \equiv wP''$, $A3, S1 \vdash P'' \subseteq P$ and $|P''| \leq |P|$.*

Moreover, if P is k-bounded, then only k-bounded instances of axiom S1 need be used in the equational proofs.

Proof. We prove both statements by induction on the length of the proof of the transition $P \xrightarrow{a} P'$. We proceed by a case analysis on the last rule used in such a proof.

- Case: $P \equiv a.P' \xrightarrow{a} P'$.

  Trivially, $A3 \vdash aP' \subseteq P$ and this establishes statement 1. Statement 2 is vacuous as $|P'| < |P|$.

- Case: $P \equiv P_1 + P_2 \xrightarrow{a} P'$ because $P_i \xrightarrow{a} P'$ for some $i \in \{1, 2\}$.

  Assume without loss of generality that $P_1 \xrightarrow{a} P'$. To prove that statement 1 holds it is sufficient to note that the inductive hypothesis for this statement gives that $A3, S1 \vdash aP' \subseteq P_1$. By the form of P, $A3, S1 \vdash aP' \subseteq P$ follows immediately.

  We now show that statement 2 also holds. Assume that $|P'| \geq |P|$. Trivially, $|P'| > |P_1|$ also holds. We may thus apply the inductive hypothesis for statement 2 to derive that there exist a string w and a term P'' such that $P' \equiv wP''$, $A3, S1 \vdash P'' \subseteq P_1$ and $|P''| \leq |P_1|$. Clearly, it follows that $A3, S1 \vdash P'' \subseteq P_1 \subseteq P$ and $|P''| \leq |P_1| < |P|$.

- Case: $P \equiv (av)^* P_1 \xrightarrow{a} vP \equiv P'$.

  Statement 1 follows immediately by equation S7. To prove statement 2, it is sufficient to take $P'' \triangleq P$ and $w \triangleq v$.

- Case: $P \equiv v^* P_1 \xrightarrow{a} P'$ because $P_1 \xrightarrow{a} P'$.

  Statement 1 is immediate by induction and equation S6. To prove statement 2, assume that $|P'| \geq |P|$. As $|P| > |P_1|$, we may apply the inductive hypothesis to infer that there exist a string w and a term P'' such that $P' \equiv wP''$, $A3, S1 \vdash P'' \subseteq P_1$ and $|P''| \leq |P_1|$. Therefore, as $S6 \vdash P_1 \subseteq P$, we conclude that $A3, S1 \vdash P'' \subseteq P$ and $|P''| \leq |P_1| < |P|$ as desired.

An inspection of the above cases shows that if P is k-bounded, then only k-bounded instances of equation S1 need be used in the proof. This completes the inductive argument and the proof of the lemma. □

We now establish a decomposition property of string iteration with respect to the relation of bisimulation equivalence. A similar decomposition property for the delay operation of Milner's SCCS [25] with respect to a notion of strong bisimulation preorder was, to our knowledge, first shown by Hennessy in [18]. Our proof follows the one given in [1, Lem. 4.3].

Lemma 27. *Let $P, Q \in \mathsf{MPA}^{s*}(A)$ and $w \in A^+$. Then $w^*P \leftrightarrow w^*Q$ iff $w^*P \leftrightarrow Q$ or $P \leftrightarrow w^*Q$ or $P \leftrightarrow Q$.*

Proof. The "if" implication follows immediately from the fact that $\leftrightarrow$ is a congruence and the soundness of equation S2. To show the "only if" implication, it is sufficient to prove that:

$$w^*P \leftrightarrow w^*Q \text{ and } w^*P \not\leftrightarrow Q \text{ and } P \not\leftrightarrow w^*Q \quad \Rightarrow \quad P \leftrightarrow Q \ . \tag{8}$$

We prove that (8) holds. To this end, let us assume that $w^*P \leftrightarrow w^*Q$, $w^*P \not\leftrightarrow Q$ and $P \not\leftrightarrow w^*Q$. By symmetry, to prove that $P \leftrightarrow Q$ must hold, it is sufficient to show that, for all $a \in A$, $P' \in \mathsf{MPA}^{s*}(A)$:

$$P \xrightarrow{a} P' \quad \Rightarrow \quad \exists Q' : \ Q \xrightarrow{a} Q' \text{ and } P' \leftrightarrow Q' \ . \tag{9}$$

This we now proceed to show. Assume that $P \xrightarrow{a} P'$. By the operational semantics for $\mathsf{MPA}^{s*}(A)$, this implies that $w^*P \xrightarrow{a} P'$. As $w^*P \underline{\leftrightarrow} w^*Q$, it follows that $w^*Q \xrightarrow{a} Q'$ for some Q' such that $P' \underline{\leftrightarrow} Q'$. If $Q \xrightarrow{a} Q'$, then we are done. Otherwise, it must be case that, for some string w', $w = aw'$ and $Q' \equiv w'(w^*Q)$. We show that this leads to a contradiction.

Assume that $P \xrightarrow{a} P' \underline{\leftrightarrow} w'(w^*Q)$, with $w = aw'$. Then:

$$\begin{array}{lcll}
P & \underline{\leftrightarrow} & P + aP' & \text{(By Lemmas 26(1) and 15)} \\
 & \underline{\leftrightarrow} & P + aw'(w^*Q) & (P' \underline{\leftrightarrow} w'(w^*Q)) \\
 & \underline{\leftrightarrow} & P + aw'(w^*P) & (w^*P \underline{\leftrightarrow} w^*Q) \\
 & \underline{\leftrightarrow} & w^*P & \text{(Soundness of axiom S1 and } w = aw'\text{)} \\
 & \underline{\leftrightarrow} & w^*Q & (w^*P \underline{\leftrightarrow} w^*Q)
\end{array}$$

This contradicts the assumption that $P \not\underline{\leftrightarrow} w^*Q$. □

The last stepping stone towards the proof of the completeness theorem we are after is the following lemma, which states some properties of bisimulation equivalence over $\mathsf{MPA}^{s*}(A)$ terms which are not terminal.

Lemma 28. *Let $P, Q \in \mathsf{MPA}^{s*}(A)$. The following statements hold:*

1. *Assume that $|P|_{min} > 0$, $v \in A^*$ and $avP \underline{\leftrightarrow} Q$. Then Q cannot have the form w^*Q_1 for any $w \in A^+$ and $\mathsf{MPA}^{s*}(A)$ term Q_1.*

2. *Let $v, w \in A^*$. Assume that $Q \equiv (aw)^*R$ for some $\mathsf{MPA}^{s*}(A)$ term R, $|Q|_{min} > 0$ and $vP \underline{\leftrightarrow} wQ$. Then $\mathsf{length}(v) \leq \mathsf{length}(w)$ and $P \underline{\leftrightarrow} uQ$ for some $u \in A^*$ such that $w = vu$.*

Proof. We prove the two statements separately.

1. Suppose that $|P|_{min} > 0$, $v \in A^*$ and $avP \underline{\leftrightarrow} Q$. We prove that Q cannot have the form w^*Q_1 for any $w \in A^+$ and $\mathsf{MPA}^{s*}(A)$ term Q_1. To this end, assume, towards a contradiction, that $Q \equiv w^*Q_1$. First of all, note that, as $|P|_{min} > 0$, Lem. 25(3) gives that:

   $$(10) \qquad |avP|_{min} \;=\; \mathsf{length}(av) + |P|_{min} \;.$$

   As $avP \underline{\leftrightarrow} Q$, by Lem. 25(2) we derive that:

   $$(11) \qquad |avP|_{min} \;=\; |Q|_{min} \;.$$

   Note that, from the above equalities, we can derive immediately that $|Q|_{min} > 0$. Furthermore, again from $avP \underline{\leftrightarrow} Q$, it follows that $w = aw'$, for some $w' \in A^*$, and that $vP \underline{\leftrightarrow} w'Q$. Using this information we can now derive that:

   $$\begin{array}{lcll}
   |Q|_{min} & = & |avP|_{min} & (11) \\
    & = & \mathsf{length}(av) + |P|_{min} & (10) \\
    & = & 1 + |vP|_{min} & \text{(Lem. 25(3))} \\
    & > & |w'Q|_{min} & \text{(Lem. 25(2) and } vP \underline{\leftrightarrow} w'Q\text{)} \\
    & \geq & |Q|_{min} & \text{(By Lem. 25(3) as } |Q|_{min} > 0\text{)} \;.
   \end{array}$$

   This is clearly a contradiction.

2. Suppose that $Q \equiv (aw)^* R$ for some $\mathsf{MPA}^{s*}(A)$ term R, $|Q|_{min} > 0$ and $vP \leftrightarrows wQ$. We prove, first of all, that $\mathsf{length}(v) \leq \mathsf{length}(w)$. To this end, assume, towards a contradiction, that $\mathsf{length}(v) > \mathsf{length}(w)$. As $vP \leftrightarrows wQ$, w must be a proper prefix of v. This means that there exists a non-empty string u such that $v = wu$ and $uP \leftrightarrows Q$. Because of the form Q takes, $uP \leftrightarrows Q$ implies that $u = (aw)^k au'$ and $P \leftrightarrows u''Q$ for some $k \geq 0$ and strings u', u'' such that $w = u'u''$. By statements (2) and (3) of Lemma 25, $P \leftrightarrows u''Q$ and $|Q|_{min} > 0$ imply that $|P|_{min} = |u''Q|_{min} > 0$. Thus we have that $uP \leftrightarrows Q \equiv (aw)^* R$, $u \in A^+$ and $|P|_{min} > 0$. This contradicts statement 1 of this lemma. Therefore it must be the case that $\mathsf{length}(v) \leq \mathsf{length}(w)$. The fact that $P \leftrightarrows uQ$ for some $u \in A^*$ such that $w = vu$ is now immediate.

□

We can now tackle the proof of the promised completeness theorem. Unfortunately, the proof of this result is combinatorial in nature and consists of the examination of a fairly large number of cases. For this reason, we have chosen to present the proof in a structured style following the spirit, albeit not the letter, of the proposal in [24]. We hope that this type of presentation will help the reader understand easily the details of the proof and judge its correctness, if he/she wishes to do so.

Theorem 29 *(Completeness). For all* $\mathsf{MPA}^{s*}(A)$ *terms* P *and* Q*,* $P \leftrightarrows Q$ *implies* $\mathsf{MPA}^{s*} \vdash P = Q$*. Moreover, if* P *and* Q *are* k*-bounded then* $\mathsf{MPA}^{s*}_k \vdash P = Q$*.*

Proof. We shall show that the following three statements hold for all $P, Q \in \mathsf{MPA}^{s*}(A)$:

1. Suppose that $w \in A^+$, $|P|_{min} > 0$ and $wP \leftrightarrows Q$. Then there exists an $\mathsf{MPA}^{s*}(A)$ term Q' such that $P \leftrightarrows Q'$, $\mathsf{MPA}^{s*} \vdash Q = wQ'$ and $|Q'| + \mathsf{length}(w) \leq |Q|$.
2. If $P \leftrightarrows P + Q$ then $\mathsf{MPA}^{s*} \vdash Q \subseteq P$.
3. If $P \leftrightarrows Q$ then $\mathsf{MPA}^{s*} \vdash P = Q^2$.

We prove these three statements simultaneously by complete induction on $|P| + |Q|$. To this end, let us assume, as our inductive hypothesis, that statements 1–3 hold for all $\mathsf{MPA}^{s*}(A)$ terms P', Q' such that $|P'| + |Q'| < |P| + |Q|$. We now prove that they hold for P and Q. We examine each statement in turn.

1. Suppose that $w \in A^+$, $|P|_{min} > 0$ and $wP \leftrightarrows Q$. We show that the thesis for statement 1 above holds by a case analysis on the form Q may take. Throughout the proof for this statement, we let $w = aw'$ for some $w' \in A^*$.
   - Case: $Q \equiv \delta$.
     The claim is vacuously true because $aw'P$ cannot be bisimilar to δ.

[2] Strictly speaking, it is not necessary to isolate this statement for the proof below to go through. In fact, all the uses of statement 3 made in the proof are implied by statement 2. However, we believe that the current formulation makes it easier to follow our arguments in detail.

- Case: $Q \equiv bQ_1$.

  As $aw'P$ is bisimulation equivalent to Q, it must be the case that $a = b$ and $w'P \leftrightarrow Q_1$. We now proceed by distinguishing two cases, depending on whether the string w' is empty or not.

  - Case: $w' = \lambda$.

    The claim follows immediately by taking $Q' \overset{\triangle}{=} Q_1$.
  - Case: $w' \in A^+$.

    In this case, we have that $w'P \leftrightarrow Q_1$, $w' \in A^+$ and $|P|_{min} > 0$. As $|P| + |Q_1| < |P| + |Q|$, we may apply the inductive hypothesis for statement 1 to derive that, for some $\mathsf{MPA}^{s*}(A)$ term Q',

    $$P \leftrightarrow Q' \text{ and } \mathsf{MPA}^{s*} \vdash Q_1 = w'Q' \text{ and } |Q'| + \mathsf{length}(w') \leq |Q_1| \ .$$

    It is now a simple matter to show that this term Q' satisfies all the constraints in the thesis of statement 1.

  This completes the proof for this case.

- Case: $Q \equiv Q_1 + Q_2$.

  As $aw'P \leftrightarrow Q$, one of the following three cases must hold:

  - $Q_1 \leftrightarrow \delta$ and $Q_2 \leftrightarrow aw'P$, or
  - $Q_2 \leftrightarrow \delta$ and $Q_1 \leftrightarrow aw'P$, or
  - $Q_1 \leftrightarrow Q_2 \leftrightarrow aw'P$.

  In fact, as the first two cases are symmetric, we may restrict ourselves, without loss of generality, to considering only the first and the third.

  - Case: $Q_1 \leftrightarrow \delta$ and $Q_2 \leftrightarrow aw'P$.

    As $Q_1 \leftrightarrow \delta$, an application of Lem. 16 gives that

    $$(12) \qquad A6 \vdash Q_1 = \delta \ .$$

    As $|P| + |Q_2| < |P| + |Q|$, $Q_2 \leftrightarrow aw'P$ and $|P|_{min} > 0$, we may apply the inductive hypothesis for statement 1 to derive that there exists an $\mathsf{MPA}^{s*}(A)$ term Q' such that $P \leftrightarrow Q'$, $|Q'| + \mathsf{length}(aw') \leq |Q_2|$ and

    $$(13) \qquad \mathsf{MPA}^{s*} \vdash Q_2 = aw'Q' \ .$$

    Now,

    $$\begin{array}{rcll} \mathsf{MPA}^{s*} \vdash \quad Q & = & Q_1 + Q_2 & (Q \equiv Q_1 + Q_2) \\ & = & \delta + aw'Q' & (\text{By (12) and (13)}) \\ & \overset{\text{A6}}{=} & aw'Q' & \end{array}$$

    and $|Q'| + \mathsf{length}(aw') \leq |Q_2| < |Q|$. This completes the proof for this case.
  - Case: $Q_1 \leftrightarrow Q_2 \leftrightarrow aw'P$.

In this case, as $|P| + |Q_i| < |P| + |Q|$ for $i = 1, 2$, we may apply the inductive hypothesis for statement 1 to derive that, for $i = 1, 2$, there exist $\mathsf{MPA}^{s*}(A)$ terms Q'_i such that $P \leftrightarroweq Q'_i$, $|Q'_i| + \mathsf{length}(aw') \leq |Q_i|$ and

$$\mathsf{MPA}^{s*} \vdash Q_i = aw'Q'_i \ . \tag{14}$$

Now, as $|Q'_i| + \mathsf{length}(aw') \leq |Q_i| < |Q|$ $(i = 1, 2)$, it follows that $|P| + |Q'_i| < |P| + |Q|$ for $i = 1, 2$. We may therefore apply the inductive hypothesis for statement 3 to the equivalences $P \leftrightarroweq Q'_1$ and $P \leftrightarroweq Q'_2$ to derive that:

$$\mathsf{MPA}^{s*} \vdash P = Q'_1 = Q'_2 \ . \tag{15}$$

Choose $Q' \stackrel{\triangle}{=} Q'_1$. We have already seen that $P \leftrightarroweq Q'$ and that $|Q'| + \mathsf{length}(aw') < |Q|$. Moreover,

$$\begin{array}{llll} \mathsf{MPA}^{s*} \vdash \ Q & = & Q_1 + Q_2 & (Q \equiv Q_1 + Q_2) \\ & = & aw'Q'_1 + aw'Q'_2 & (14) \\ & = & aw'Q'_1 + aw'Q'_1 & (15) \\ & \stackrel{\text{A3}}{=} & aw'Q'_1 & \end{array}$$

This completes the proof for this case.

The proof for the case $Q \equiv Q_1 + Q_2$ is now complete.

- Case: $Q \equiv v^*Q_1$ for some $v \in A^+$.
  This case is vacuous by Lem. 28(1).

The proof of the inductive step for statement 1 is now complete.

2. Assume that $P \leftrightarroweq P + Q$. We shall show that $\mathsf{MPA}^{s*} \vdash Q \subseteq P$. To this end, we consider the following three cases:

   (a) P is a terminal term, i.e. $|P|_{min} = 0$, or

   (b) $|P|_{min} > 0$ and $|Q|_{min} = 0$, or

   (c) $|P|_{min} > 0$ and $|Q|_{min} > 0$.

   We examine each of these cases in turn.

   (a) Case: P is a terminal term, i.e. $|P|_{min} = 0$.
   As $|P|_{min} = 0$, we have that either $P \leftrightarroweq \delta$ or P is a terminal cycle. If $P \leftrightarroweq \delta \leftrightarroweq P + Q$, then the claim is an immediate corollary of Lem. 16 and transitivity. If P is a terminal cycle, then the claim follows immediately from Corollary 23.

   (b) Case: $|P|_{min} > 0$ and $|Q|_{min} = 0$.
   Again, as $|Q|_{min} = 0$, it follows that either $Q \leftrightarroweq \delta$ or Q is a terminal cycle. We proceed by examining these possibilities in turn.
   If $Q \leftrightarroweq \delta$, then

$$\begin{array}{llll} \mathsf{MPA}^{s*} \vdash \ P & \stackrel{\text{A3}}{=} & P + \delta & \\ & = & P + Q & \text{(By Lem. 16)} \end{array}$$

and we are done.

If Q is a terminal cycle, then $Q \leftrightarrows (aw)^* \delta$ for some action $a \in A$ and string $w \in A^*$. (Note that, by Lem. 20, aw can be chosen to be a prime root string whose length does not exceed the iteration bound of Q.) By Lem. 22, it follows that:

(16) $$\mathsf{TerCycles} \vdash Q = (aw)^* \delta \ .$$

As $\leftrightarrows$ is a congruence, we infer that $P \leftrightarrows P + (aw)^* \delta$. Since $P + (aw)^* \delta \xrightarrow{a} w(aw)^* \delta$, there exists an $\mathsf{MPA}^{s*}(A)$ term P' such that

$$P \xrightarrow{a} P' \text{ and } P' \leftrightarrows w(aw)^* \delta \ .$$

Now, it is easy to see that $P' \leftrightarrows (wa)^* \delta$, i.e. P' is itself a terminal cycle. By Lem. 22, it follows that:

(17) $$\mathsf{TerCycles} \vdash P' = (wa)^* \delta \ .$$

Now we reason as follows:

$$\begin{array}{llll} \mathsf{MPA}^{s*} \vdash \ P & = & P + a.P' & \text{(By Lem. 26(1), as } P \xrightarrow{a} P') \\ & = & P + a.(wa)^* \delta & (17) \\ & \overset{\text{S4}}{=} & P + (aw)^* \delta & \\ & = & P + Q & (16) \end{array}$$

This completes the proof for this case.

(c) Case: $|P|_{min} > 0$ and $|Q|_{min} > 0$.

So far, induction has not played any rôle in the proof of this statement and we have been able to make do with general results about terminal $\mathsf{MPA}^{s*}(A)$ terms. The inductive hypotheses will, however, play a crucial role in the proof of the inductive step for this case. We proceed by a case analysis on the form a term Q with $|Q|_{min} > 0$ may take.

- Case: $Q \equiv Q_1 + Q_2$.

  First of all, note that $P \leftrightarrows P + Q$ and $Q \equiv Q_1 + Q_2$ imply that $P \leftrightarrows P + Q_1$ and $P \leftrightarrows P + Q_2$. As $|P| + |Q_i| < |P| + |Q|$ for $i = 1, 2$, we may apply the inductive hypothesis for statement 2 to both these equalities to derive that, for $i = 1, 2$:

  (18) $$\mathsf{MPA}^{s*} \vdash \ P \ = \ P + Q_i \ .$$

  We now argue as follows:

  $$\begin{array}{llll} \mathsf{MPA}^{s*} \vdash \ P & = & P + Q_2 & (18) \\ & = & P + Q_1 + Q_2 & (18) \\ & = & P + Q & (Q \equiv Q_1 + Q_2) \end{array}$$

  and we are done.

- Case: $Q \equiv a.R$.
  As $P {\leftrightarrows} P+Q$ and $P+Q \xrightarrow{a} R$, there exists an $\mathsf{MPA}^{s*}(A)$ term P' such that $P \xrightarrow{a} P'$ and $P' {\leftrightarrows} R$. We proceed by examining the relationship between the size of P' and that of P.
  - Case: $|P'| \leq |P|$.
    In this case, it follows that $|P'| + |R| < |P| + |Q|$. Therefore we may apply the inductive hypothesis for statement 3 to the equivalence $P' {\leftrightarrows} R$ to derive that:

    $$\mathsf{MPA}^{s*} \vdash P' = R \ . \qquad (19)$$

    We may now simply complete the proof thus:

    $$\begin{array}{lcll} \mathsf{MPA}^{s*} \vdash \quad P & = & P + a.P' & \text{(By Lem. 26(1), as } P \xrightarrow{a} P'\text{)} \\ & = & P + a.R & (19) \\ & = & P + Q & (Q \equiv a.R) \end{array}$$

  - Case: $|P'| > |P|$.
    We proceed by a case analysis on the form P may take. As $P \xrightarrow{a} P'$ and $|P'| > |P|$, there are only two possible cases to consider, namely $P \equiv P_1 + P_2$ and $P \equiv w^*T$. We shall examine these in turn.
    - ∗ Case: $P \equiv P_1 + P_2 \xrightarrow{a} P' {\leftrightarrows} R$.
      Assume, without loss of generality, that $P_1 \xrightarrow{a} P'$. Then it is easy to see that $P_1 {\leftrightarrows} P_1 + a.R$. As $|P_1| + |aR| < |P| + |aR|$, we may apply the inductive hypothesis for statement 2 to the equivalence $P_1 {\leftrightarrows} P_1 + a.R$ to derive that:

      $$\mathsf{MPA}^{s*} \vdash a.R \subseteq P_1$$

      from which it follows immediately that

      $$\mathsf{MPA}^{s*} \vdash a.R \subseteq P \ .$$

    - ∗ Case: $P \equiv w^*T \xrightarrow{a} P' {\leftrightarrows} R$.
      We proceed by examining the possible form of the transition $P \equiv w^*T \xrightarrow{a} P'$. By the operational semantics for $\mathsf{MPA}^{s*}(A)$, there are two possibilities to consider:
      - · $T \xrightarrow{a} P'$, or
      - · $w = aw'$ and $P' \equiv w'P$, for some $w' \in A^+$. (Note that, as $|P'| > |P|$, the string w' can*not* be empty.)

      We proceed by examing these two possibilities in turn.
      - · Case: $T \xrightarrow{a} P'$.
        As $T \xrightarrow{a} P' {\leftrightarrows} R$, it follows that $T {\leftrightarrows} T + a.R$. As $|T| + |aR| < |P| + |aR|$, we may apply the inductive hypothesis for statement 2 to the equivalence $T {\leftrightarrows} T + a.R$ to derive that:

        $$\mathsf{MPA}^{s*} \vdash a.R \subseteq T \ .$$

Using the derived equation S6, it follows immediately that

$$\mathsf{MPA}^{s*} \vdash a.R \subseteq P \ .$$

· Case: $w = aw'$ and $P' \equiv w'P$, for some $w' \in A^+$.
In this case, we have that $w' \in A^+$, $|P|_{min} > 0$, and $w'P \leftrightarrow R$. As $|R| < |Q|$, we have that $|R| + |P| < |Q| + |P|$. We may therefore apply the inductive hypothesis for statement 1 to derive that, for some $\mathsf{MPA}^{s*}(A)$ term R',

$$P \leftrightarrow R' \text{ and } \mathsf{MPA}^{s*} \vdash R = w'R' \text{ and } |R'| + \mathsf{length}(w') \leq |R| \ .$$

As $|R'| < |R| < |Q|$, it follows that $|P| + |R'| < |P| + |Q|$. We may therefore apply the inductive hypothesis for statement 3 to the equivalence $P \leftrightarrow R'$ to derive that:

(20) $$\mathsf{MPA}^{s*} \vdash P = R' \ .$$

We now argue as follows:

$$\begin{array}{llcll} \mathsf{MPA}^{s*} \vdash & P & = & (aw')^*T & (P \equiv (aw')^*T) \\ & & \overset{\text{S7}}{=} & P + aw'P & \\ & & = & P + aw'R' & (20) \\ & & = & P + aR & (\mathsf{MPA}^{s*} \vdash R = w'R') \\ & & = & P + Q & (Q \equiv aR) \end{array}$$

and we are done.

The proof for the case $P \equiv (aw)^*T$ is now complete.

We have now examined all the possible forms that P may take when $P \xrightarrow{a} P'$ and $|P'| > |P|$.

This completes the proof for the case $Q \equiv aR$.

- Case: $Q \equiv (aw)^*R$.
First of all, note that, as $P \leftrightarrow P+Q$, it follows that $P \leftrightarrow P+R$. As $|P|+|R| < |P| + |Q|$, we may apply the inductive hypothesis for statement 2 to the equivalence $P \leftrightarrow P + R$ to obtain that:

(21) $$\mathsf{MPA}^{s*} \vdash R \subseteq P \ .$$

This equality will be used repeatedly in the arguments to follow.
As $P \leftrightarrow P + Q$ and $Q \xrightarrow{a} wQ$, there exists an $\mathsf{MPA}^{s*}(A)$ term P' such that $P \xrightarrow{a} P'$ and $P' \leftrightarrow wQ$. We proceed by considering two cases depending on whether $|P'| < |P|$ or not.

  – Case: $|P'| < |P|$.
  We proceed by considering two sub-cases, depending on whether w is the empty string or not. We examine these two possibilities in turn.

∗ Case: The string w is empty.
In this case, we have that $P' \underline{\leftrightarrow} Q \equiv a^*R$. As $|P'| < |P|$, it follows that $|P'| + |Q| < |P| + |Q|$. We may therefore apply the inductive hypothesis for statement 3 to the equivalence $P' \underline{\leftrightarrow} Q$ to derive that:

$$\text{(22)} \qquad \mathsf{MPA}^{s*} \vdash P' = Q \ .$$

We may now argue as follows:

$$\begin{array}{lrcll}
\mathsf{MPA}^{s*} \vdash & P & = & P + aP' & (\text{By Lem. 26(1), as } P \xrightarrow{a} P') \\
 & & = & P + aQ & (22) \\
 & & = & P + R + aQ & (21) \\
 & & \overset{\text{S1}}{=} & P + Q &
\end{array}$$

and we are done.

∗ Case: $w \in A^+$.
In this case, we have that $w \in A^+$, $P' \underline{\leftrightarrow} wQ$ and $|Q|_{min} > 0$. As $|P'| + |Q| < |P| + |Q|$, we may apply the inductive hypothesis for statement 1 to derive that, for some $\mathsf{MPA}^{s*}(A)$ term P'',

$$\begin{array}{lcl}
\text{(23)} & P'' \underline{\leftrightarrow} Q & \text{and} \\
\text{(24)} & \mathsf{MPA}^{s*} \vdash P' = wP'' & \text{and} \\
\text{(25)} & |P''| + \mathsf{length}(w) \leq |P'| &
\end{array}$$

By (25) and the fact that $|P'| < |P|$, it follows that $|P''| < |P|$. Therefore $|P''| + |Q| < |P| + |Q|$, and we may apply the inductive hypothesis for statement 3 to (23) to derive that:

$$\text{(26)} \qquad \mathsf{MPA}^{s*} \vdash P'' = Q \ .$$

Now we argue that:

$$\begin{array}{lrcll}
\mathsf{MPA}^{s*} \vdash & P & = & P + aP' & (\text{By Lem. 26(1), as } P \xrightarrow{a} P') \\
 & & = & P + awP'' & (24) \\
 & & = & P + awQ & (26) \\
 & & = & P + R + awQ & (21) \\
 & & \overset{\text{S1}}{=} & P + Q &
\end{array}$$

and we are done.

This completes the proof for the case $|P'| < |P|$.

– Case: $|P'| \geq |P|$.
Assume that $P \xrightarrow{a} P' \underline{\leftrightarrow} wQ$ and $|P'| \geq |P|$. We proceed by a case analysis on the form P may take. Because of the constraints for this case, P may only take one of the following two forms:

∗ $P \equiv P_1 + P_2$ or

∗ $P \equiv (bv)^*T$.

We examine these two cases in turn.

$*$ Case: $P \equiv P_1 + P_2$.

Assume, without loss of generality, that $P_1 \xrightarrow{a} P'$. As $|P'| \geq |P| > |P_1|$, Lem. 26(2) gives, among other things, the existence of an $\mathsf{MPA}^{s*}(A)$ process P'' and of a string $v \in A^*$ such that:

$$P' \equiv vP'' \tag{27}$$

$$|P''| \leq |P_1| \ . \tag{28}$$

As $P' \equiv vP'' \leftrightarroweq wQ$, $|Q|_{min} > 0$ and $Q \equiv (aw)^*R$, by applying Lem. 28(2) we derive that either:

- $v = w$ and $P'' \leftrightarroweq Q$, or
- $vu = w$ and $P'' \leftrightarroweq uQ$ for some string $u \in A^+$.

We proceed by examing these two cases in turn.

- Case: $v = w$ and $P'' \leftrightarroweq Q$

  By (28) and the fact that $P \equiv P_1 + P_2$, we infer that $|P''| < |P|$. Therefore $|P''| + |Q| < |P| + |Q|$, and we may apply the inductive hypothesis for statement 3 to the equivalence $P'' \leftrightarroweq Q$ to derive that:

$$\mathsf{MPA}^{s*} \vdash P'' = Q \ . \tag{29}$$

  We may now argue as follows:

$$\begin{array}{rcll} \mathsf{MPA}^{s*} \vdash \ P & = & P + aP' & (\text{By Lem. 26(1), as } P \xrightarrow{a} P') \\ & = & P + awP'' & (v = w \text{ and } (27)) \\ & = & P + awQ & (29) \\ & = & P + R + awQ & (21) \\ & \overset{\text{S1}}{=} & P + Q & \end{array}$$

  and we are done.

- Case: $vu = w$ and $P'' \leftrightarroweq uQ$ for some string $u \in A^+$.

  By (28) and the fact that $P \equiv P_1 + P_2$, we infer that $|P''| < |P|$. Hence it follows that $|P''| + |Q| < |P| + |Q|$. Therefore, as $u \in A^+$, $P'' \leftrightarroweq uQ$ and $|Q|_{min} > 0$, we may apply the inductive hypothesis for statement 1 to derive that, for some $\mathsf{MPA}^{s*}(A)$ term P''', the following facts hold:

$$P''' \leftrightarroweq Q \tag{30}$$

$$\mathsf{MPA}^{s*} \vdash \ P'' = uP''' \tag{31}$$

$$|P'''| + \mathsf{length}(u) \leq |P''| \ . \tag{32}$$

  By (32) and $|P''| < |P|$, it follows that $|P'''| < |P|$. Therefore $|P'''| + |Q| < |P| + |Q|$, and we may apply the inductive hypothesis for statement 3 to equivalence (30) to derive that:

$$\mathsf{MPA}^{s*} \vdash P''' = Q \ . \tag{33}$$

Thus,

$$\begin{array}{llll}\mathsf{MPA}^{s*} \vdash P & = & P + aP' & \text{(By Lem. 26(1), as } P \xrightarrow{a} P'\text{)} \\ & = & P + avP'' & (27) \\ & = & P + avuP''' & (31) \\ & = & P + awQ & (w = vu \text{ and } (33)) \\ & = & P + R + awQ & (21) \\ & \overset{\mathrm{S1}}{=} & P + Q & \end{array}$$

This completes the proof for the case $P \equiv P_1 + P_2$.

$*$ Case: $P \equiv (bv)^*T$.
In this case, we have that $P \equiv (bv)^*T \xrightarrow{a} P' \leftrightarrows wQ$ and $|P'| \geq |P|$. We proceed by examing the possible form of the transition $P \xrightarrow{a} P'$. By the operational semantics for $\mathsf{MPA}^{s*}(A)$, there are two cases to consider:

$\cdot$ $T \xrightarrow{a} P'$, or

$\cdot$ $b = a$ and $P' \equiv vP$.

We proceed by examining these two cases in turn.

$\cdot$ Case: $T \xrightarrow{a} P'$.
As $|P'| \geq |P| > |T|$, we may apply Lem. 26(2) to the transition $T \xrightarrow{a} P'$ to derive that, among other things, for some string $u \in A^*$ and $\mathsf{MPA}^{s*}(A)$ term P'',

$$(34) \qquad P' \equiv uP''$$
$$(35) \qquad |P''| \leq |T| \ .$$

Since $P' \equiv uP'' \leftrightarrows wQ$, $|Q|_{min} > 0$ and $Q \equiv (aw)^*R$, Lem. 28(2) gives that either
$u = w$ and $P'' \leftrightarrows Q$, or
$w = uu'$ and $P'' \leftrightarrows u'Q$ for some non-empty string $u' \in A^+$.
We proceed by examining these two cases in turn.

Case: $u = w$ and $P'' \leftrightarrows Q$.
By (35) and the form P takes, it follows that $|P''| < |P|$. Therefore $|P''| + |Q| < |P| + |Q|$, and we may apply the inductive hypothesis for statement 3 to the equivalence $P'' \leftrightarrows Q$ to derive that:
$$(36) \qquad \mathsf{MPA}^{s*} \vdash P'' = Q \ .$$

We may now argue as follows:

$$\begin{array}{llll}\mathsf{MPA}^{s*} \vdash P & \overset{\mathrm{S6}}{=} & P + T & \\ & = & P + T + aP' & \text{(By Lem. 26(2), as } T \xrightarrow{a} P'\text{)} \\ & = & P + awP'' & (u = w \text{ and } (34)) \\ & = & P + awQ & (36) \\ & = & P + R + awQ & (21) \\ & \overset{\mathrm{S1}}{=} & P + Q & \end{array}$$

and we are done.

Case: $w = uu'$ and $P'' \underline{\leftrightarrow} u'Q$ for some non-empty string $u' \in A^+$. By (35) and the form P takes, it follows that $|P''| < |P|$. Hence $|P''| + |Q| < |P| + |Q|$. As $P'' \underline{\leftrightarrow} u'Q$, $u' \in A^+$ and $|Q|_{min} > 0$, we may therefore apply the inductive hypothesis for statement 1 to derive that, for some $\mathsf{MPA}^{s*}(A)$ term P''':

$$\begin{array}{lrcl}
(37) & P''' & \underline{\leftrightarrow} & Q \\
(38) & \mathsf{MPA}^{s*} \vdash \quad P'' & = & u'P''' \\
(39) & |P'''| + \mathsf{length}(u') & \leq & |P''| \ .
\end{array}$$

As $|P'''| < |P''| < |P|$, it follows that $|P'''| + |Q| < |P| + |Q|$. We may therefore apply the inductive hypothesis for statement 3 to (37) to infer that:

$$(40) \qquad \mathsf{MPA}^{s*} \vdash P''' = Q \ .$$

Thus:

$$\begin{array}{rcll}
\mathsf{MPA}^{s*} \vdash \quad P & \overset{\text{S6}}{=} & P + T & \\
 & = & P + T + aP' & \text{(By Lem. 26(2), as } T \xrightarrow{a} P'\text{)} \\
 & = & P + auP'' & (34) \\
 & = & P + auu'P''' & (38) \\
 & = & P + awQ & (w = uu' \text{ and } (40)) \\
 & = & P + R + awQ & (21) \\
 & \overset{\text{S1}}{=} & P + Q &
\end{array}$$

and we are done.

This completes the proof for the case $T \xrightarrow{a} P'$.

- Case: $b = a$ and $P' \equiv vP$.

  In this case, we have that $P \equiv (av)^*T$, $Q \equiv (aw)^*R$, $|P|_{min} > 0$, $|Q|_{min} > 0$ and $vP \underline{\leftrightarrow} wQ$. By Lem. 28(2) and symmetry, it must be the case that $v = w$ and $P \underline{\leftrightarrow} Q$. By using Lem. 27, we may now derive, because of the form P and Q take, that:

  $$T \underline{\leftrightarrow} R \text{ or } P \underline{\leftrightarrow} R \text{ or } T \underline{\leftrightarrow} Q \ .$$

  In each of the above cases, we may apply the inductive hypothesis for statement 3 and substitutivity to infer that at least one of the following equalities is provable from the theory MPA^{s*}:

  $$P = Q \text{ or } (aw)^*P = Q \text{ or } P = (aw)^*Q \ .$$

  In each of the above cases we obtain, after possibly applying equation S2, that $\mathsf{MPA}^{s*} \vdash P = Q$, from which $\mathsf{MPA}^{s*} \vdash Q \subseteq P$ follows immediately by A3. This completes the proof for this case.

The proof for the case $P \equiv (bv)^*T \xrightarrow{a} P'$ and $|P'| \geq |P|$ is now complete.

We have therefore examined all the possible cases arising when $P \xrightarrow{a} P' \leftrightarrow wQ$ and $|P'| \geq |P|$.

The proof of the inductive step for statement 2 when $Q \equiv (aw)^* R$ is complete.

We have therefore shown that statement 2 holds for P and Q.

3. Assume that $P \leftrightarrow Q$. Then $P \leftrightarrow P + Q$ and $Q \leftrightarrow Q + P$. By statement 2, we infer that $\mathsf{MPA}^{s*} \vdash Q \subseteq P \subseteq Q$, from which the thesis follows immediately.

An inspection of the above proof, and the fact that the language of k-bounded terms is closed under transitions by Lem. 8 ensure that if P and Q are k-bounded, then only k-bounded equations need be used throughout. The proof of the theorem is now complete.

□

5 Bisimulation Equivalence is not Finitely Axiomatizable over $\mathsf{MPA}^{s*}(A)$

In the previous section we proved that the equational theory MPA^{s*} completely characterizes bisimulation equivalence over the language $\mathsf{MPA}^{s*}(A)$. Moreover, we showed that, for each positive integer k, the *finite* equational theory MPA^{s*}_k gives a sound and complete axiomatization of bisimulation equivalence for terms with iteration bound at most k. This implies that bisimulation equivalence can be finitely axiomatized over $\mathsf{MPA}^{s*}(A)$ terms provided that the length of all loops occurring in terms is bounded from above. The equational theory MPA^{s*} that axiomatizes bisimulation equivalence over the whole of the language $\mathsf{MPA}^{s*}(A)$ consists, however, of a countably infinite collection of equations. This immediately raises the question of whether one can improve upon our completeness result for $\mathsf{MPA}^{s*}(A)$ by exhibiting a finite equational axiomatization of the relation of bisimulation equivalence over this language. We shall now prove that no such axiomatization can exist, unless the set of actions A is empty. To establish this negative result, we shall show that, if A is non-empty, for every finite collection of sound equations $\mathcal{E}$ there is a valid equivalence of the form

$$(a^n)^*\delta \quad \leftrightarrow \quad (a^m)^*\delta \tag{41}$$

that cannot be proven to hold from the equations in $\mathcal{E}$. To this end, for every finite equational theory $\mathcal{E}$, we shall find some property which is enjoyed by every equality $E = F$ which is derivable from $\mathcal{E}$, but not by some instance of (41). The reader familiar with [31] might have noticed that terms like those used in (41) play an important rôle in Sewell's non-finite axiomatizability result for the language BPA^*_δ (cf. [31, Thm. 6.6]). The strategy of our proof will, however, be very different from the one used by Sewell in the aforementioned reference.

To obtain the aforementioned negative result, we shall need to introduce a few technical tools which will be useful for our purposes. First of all, we shall prove an important property satisfied by every equation $E = F$ which is sound with respect to $\leftrightarrow$; namely, we shall show that if $E \underline{\leftrightarrow} F$ and the set of actions A is non-empty, then E and F must have exactly the same variables occurring in them.

Definition 30. The set of strings prefixing occurrences of a variable x in a term $E \in \mathsf{MPA}^{s*}(A, \mathsf{Var})$, notation $\mathsf{Paths}(x, E)$, is defined by structural recursion on terms thus:

$$\begin{aligned}
\mathsf{Paths}(x, \delta) &\stackrel{\triangle}{=} \emptyset \\
\mathsf{Paths}(x, y) &\stackrel{\triangle}{=} \begin{cases} \{\lambda\} & \text{if } x = y \\ \emptyset & \text{otherwise} \end{cases} \\
\mathsf{Paths}(x, aE) &\stackrel{\triangle}{=} \{as \mid s \in \mathsf{Paths}(x, E)\} \\
\mathsf{Paths}(x, E + F) &\stackrel{\triangle}{=} \mathsf{Paths}(x, E) \cup \mathsf{Paths}(x, F) \\
\mathsf{Paths}(x, w^* E) &= \mathsf{Paths}(x, E)
\end{aligned}$$

The following lemma, which can be easily shown by structural induction on terms, gives all the properties of the sets $\mathsf{Paths}(x, E)$ that we shall need for our purposes.

Lemma 31. *For every $E \in \mathsf{MPA}^{s*}(A, \mathsf{Var})$ and $x \in \mathsf{Var}$, the following statements hold:*

1. *$x \in \mathsf{Var}(E)$ iff $\mathsf{Paths}(x, E) \neq \emptyset$.*
2. *Let σ be a closed substitution. Assume that $s \in \mathsf{Paths}(x, E)$ and $\sigma(x) \xrightarrow{a} P$, for some action $a \in A$ and $P \in \mathsf{MPA}^{s*}(A)$. Then $E\sigma \xrightarrow{sa} P$.*

We are now ready to prove that, if the set of actions A is non-empty, then two terms E and F can only be bisimulation equivalent if their sets of variables are identical.

Lemma 32. *If the set of actions A is non-empty, then, for all $E, F \in \mathsf{MPA}^{s*}(A, \mathsf{Var})$, $E \underline{\leftrightarrow} F$ implies $\mathsf{Var}(E) = \mathsf{Var}(F)$.*

Proof. We prove the contrapositive statement. Assume, without loss of generality, that there is a variable $x \in \mathsf{Var}(E) - \mathsf{Var}(F)$. Under this assumption, we shall construct a closed substitution σ such that $E\sigma \not\underline{\leftrightarrow} F\sigma$. This will prove that $E \not\underline{\leftrightarrow} F$.

Let $x \in \mathsf{Var}(E) - \mathsf{Var}(F)$. As $x \in \mathsf{Var}(E)$, by Lem. 31(1) there exists a string $s \in A^*$ such that $s \in \mathsf{Paths}(x, E)$. Let σ_δ be the closed substitution mapping all the variables in Var to δ. By Lem. 9, the closed term $F\sigma_\delta$ has finitely many states. As A is non-empty, by Lem. 12 we may therefore find a term $P \in \mathsf{MPA}^{s*}(A)$ which is not a state of $F\sigma_\delta$ up to bisimulation equivalence, i.e., a term P such that, for no state Q of $F\sigma_\delta$, $P \underline{\leftrightarrow} Q$. Define now a substitution σ by:

$$\sigma \stackrel{\triangle}{=} \sigma_\delta[x \mapsto aP]$$

where $a \in A$. We claim that $E\sigma \not\underline{\leftrightarrow} F\sigma$. In fact, as $s \in \mathsf{Paths}(x, E)$ and $\sigma(x) \xrightarrow{a} P$, Lem. 31(2) gives that $E\sigma \xrightarrow{sa} P$. On the other hand, as $x \notin \mathsf{Var}(F)$, it follows that $F\sigma \equiv F\sigma_\delta$, and, by construction, no state of $F\sigma_\delta$ is bisimulation equivalent to P.

We have therefore shown that $E \not\underline{\leftrightarrow} F$, as desired. □

The above result does not hold if the set of actions A is empty. In fact, in that, admittedly pathological, case all the terms in the language $\mathsf{MPA}^{s*}(\emptyset)$ are equivalent to δ, and the equation $x = y$ is sound (and complete) for bisimulation equivalence over that language. It is also interesting to remark that Lem. 32 does not hold in general for trace equivalence and maximal trace equivalence. (The interested reader is invited to consult the encyclopedic reference [16] for information on these equivalences.) For instance, if $A = \{a\}$, then it is not too hard to see that the terms $a^*(a\delta)$ and $a^*(a\delta) + P$ are maximal trace equivalent (and, *a fortiori*, trace equivalent) for every $P \in \mathsf{MPA}^{s*}(A)$. This implies that, if the set of actions A is the singleton $\{a\}$, the equation

$$(42) \qquad a^*(a\delta) \;=\; a^*(a\delta) + x$$

is sound for maximal trace equivalence. As implied by Lem. 32, equation (42) is, instead, not sound with respect to bisimulation equivalence. For instance, the terms $a^*(a\delta)$ and $a^*(a\delta) + aa\delta$ are not bisimilar.

Definition 33. Let $\mathcal{E} = \{E_i = F_i \mid 1 \le i \le n\}$ ($n \in \mathbb{N}$) be a finite equational theory over the signature of $\mathsf{MPA}^{s*}(A, \mathsf{Var})$. The *iteration bound* of $\mathcal{E}$, notation $\mathsf{IB}(\mathcal{E})$, is given by:

$$\mathsf{IB}(\mathcal{E}) \;\stackrel{\triangle}{=}\; \max\{\mathsf{IB}(E_i), \mathsf{IB}(F_i) \mid 1 \le i \le n\} \;.$$

The following lemma is the key to our promised non-finite axiomatizability result. It states a property that is true of all the equalities that are provable from a finite equational theory $\mathcal{E}$ over the signature of $\mathsf{MPA}^{s*}(A, \mathsf{Var})$, but that is not satisfied by all instances of equality (41). Intuitively, the lemma states that two terms E and F whose iteration bound is "large enough" can only be proven equal from the finite theory $\mathcal{E}$ iff they have the same iteration bound.

Lemma 34. *Let A be a non-empty set of actions, and let $\mathcal{E} = \{E_i = F_i \mid 1 \le i \le n\}$ ($n \in \mathbb{N}$) be a sound, finite equational theory over the signature of $\mathsf{MPA}^{s*}(A, \mathsf{Var})$. Let $E, F \in \mathsf{MPA}^{s*}(A, \mathsf{Var})$. Assume that $\mathcal{E} \vdash E = F$. Then the following statements hold:*

1. $\mathsf{IB}(E) > \mathsf{IB}(\mathcal{E})$ *iff* $\mathsf{IB}(F) > \mathsf{IB}(\mathcal{E})$;
2. *if* $\mathsf{IB}(E) > \mathsf{IB}(\mathcal{E})$ *and* $\mathsf{IB}(F) > \mathsf{IB}(\mathcal{E})$, *then* $\mathsf{IB}(E) = \mathsf{IB}(F)$.

Proof. Let A be a non-empty set of actions, and let $\mathcal{E} = \{E_i = F_i \mid 1 \le i \le n\}$ ($n \in \mathbb{N}$) be a sound, finite equational theory over the signature of $\mathsf{MPA}^{s*}(A, \mathsf{Var})$. Assume that $\mathcal{E} \vdash E = F$. We prove that both statements of the lemma hold, simultaneously by induction on the depth of the proof of the equality $E = F$ from the theory $\mathcal{E}$. We proceed by a case analysis on the last inference rule used in the proof. We shall give the details of the proof for all the cases, except those in which the equality $E = F$ follows by reflexivity or symmetry. For each of the cases we consider the two statements in turn.

- Case: The equality $E = F$ is proven by instantiating some equation ($E_i = F_i$) in the theory $\mathcal{E}$, i.e., $E \equiv E_i\sigma$ and $F \equiv F_i\sigma$ for some equation $(E_i = F_i) \in \mathcal{E}$ and substitution σ.

1. First of all, note that, for every $G \in \mathsf{MPA}^{s*}(A, \mathsf{Var})$ and substitution τ,

   $$\mathsf{IB}(G\tau) = \max(\mathsf{IB}(G), \max\{\mathsf{IB}(\tau(x)) \mid x \in \mathsf{Var}(G)\}) \ . \tag{43}$$

   Using the above equality, the claim is immediate from the following chain of logical equivalences:

   $$\begin{aligned}
   \mathsf{IB}(E) > \mathsf{IB}(\mathcal{E}) &\Leftrightarrow \mathsf{IB}(E_i\sigma) > \mathsf{IB}(\mathcal{E}) \\
   &\qquad (E \equiv E_i\sigma) \\
   &\Leftrightarrow \max(\mathsf{IB}(E_i), \max\{\mathsf{IB}(\sigma(x)) \mid x \in \mathsf{Var}(E_i)\}) > \mathsf{IB}(\mathcal{E}) \\
   &\qquad (43) \\
   &\Leftrightarrow \max\{\mathsf{IB}(\sigma(x)) \mid x \in \mathsf{Var}(E_i)\} > \mathsf{IB}(\mathcal{E}) \\
   &\qquad (\mathsf{IB}(\mathcal{E}) \geq \mathsf{IB}(E_i)) \\
   &\Leftrightarrow \max\{\mathsf{IB}(\sigma(x)) \mid x \in \mathsf{Var}(F_i)\} > \mathsf{IB}(\mathcal{E}) \\
   &\qquad (\text{By Lem. 32, as } E_i = F_i \text{ is sound}) \\
   &\Leftrightarrow \max(\mathsf{IB}(F_i), \max\{\mathsf{IB}(\sigma(x)) \mid x \in \mathsf{Var}(F_i)\}) > \mathsf{IB}(\mathcal{E}) \\
   &\qquad (\mathsf{IB}(\mathcal{E}) \geq \mathsf{IB}(F_i)) \\
   &\Leftrightarrow \mathsf{IB}(F_i\sigma) > \mathsf{IB}(\mathcal{E}) \\
   &\qquad (43) \\
   &\Leftrightarrow \mathsf{IB}(F) > \mathsf{IB}(\mathcal{E}) \\
   &\qquad (F \equiv F_i\sigma)
   \end{aligned}$$

2. Assume that $\mathsf{IB}(E)$ and $\mathsf{IB}(F)$ are both strictly larger than $\mathsf{IB}(\mathcal{E})$. We show that $\mathsf{IB}(E) = \mathsf{IB}(F)$. This follows because

   $$\begin{aligned}
   \mathsf{IB}(E) &= \max\{\mathsf{IB}(\sigma(x)) \mid x \in \mathsf{Var}(E_i)\} \\
   &\qquad (\text{By (43) and } \mathsf{IB}(E) > \mathsf{IB}(\mathcal{E}) \geq \mathsf{IB}(E_i)) \\
   &= \max\{\mathsf{IB}(\sigma(x)) \mid x \in \mathsf{Var}(F_i)\} \\
   &\qquad (\text{By Lem. 32, as } E_i = F_i \text{ is sound}) \\
   &= \mathsf{IB}(F) \\
   &\qquad (\text{By (43) and } \mathsf{IB}(F) > \mathsf{IB}(\mathcal{E}) \geq \mathsf{IB}(F_i))
   \end{aligned}$$

- Case: The equality $E = F$ is proven using the transitivity rule, i.e., $\mathcal{E} \vdash E = F$ because, for some $G \in \mathsf{MPA}^{s*}(A, \mathsf{Var})$, $\mathcal{E} \vdash E = G$ and $\mathcal{E} \vdash G = F$ by shorter inferences.

  1. By applying the inductive hypothesis for statement 1 to $\mathcal{E} \vdash E = G$ and $\mathcal{E} \vdash G = F$, we derive that:

     $$\mathsf{IB}(E) > \mathsf{IB}(\mathcal{E}) \ \Leftrightarrow\ \mathsf{IB}(G) > \mathsf{IB}(\mathcal{E}) \tag{44}$$

     $$\mathsf{IB}(G) > \mathsf{IB}(\mathcal{E}) \ \Leftrightarrow\ \mathsf{IB}(F) > \mathsf{IB}(\mathcal{E}) \tag{45}$$

     from which the claim follows immediately.

  2. Assume that $\mathsf{IB}(E) > \mathsf{IB}(\mathcal{E})$ and $\mathsf{IB}(F) > \mathsf{IB}(\mathcal{E})$. By (44) and (45), we also have that $\mathsf{IB}(G) > \mathsf{IB}(\mathcal{E})$. We may therefore apply the inductive hypothesis for statement 2 to the inferences $\mathcal{E} \vdash E = G$ and $\mathcal{E} \vdash G = F$ to derive that $\mathsf{IB}(E) = \mathsf{IB}(G) = \mathsf{IB}(F)$, as desired.

- **Case:** The equality $E = F$ is proven using the substitutivity rule for action prefixing, i.e., $\mathcal{E} \vdash E = F$ because, for some action $a \in A$ and terms $E', F' \in \mathsf{MPA}^{s*}(A, \mathsf{Var})$, $E \equiv aE'$, $F \equiv aF'$ and $\mathcal{E} \vdash E' = F'$ by a shorter inference.

  Both statements follow immediately by induction and the fact that, for every $G \in \mathsf{MPA}^{s*}(A, \mathsf{Var})$, $\mathsf{IB}(aG) = \mathsf{IB}(G)$.

- **Case:** The equality $E = F$ is proven using the substitutivity rule for summation, i.e., $\mathcal{E} \vdash E = F$ because, for some terms $E', E'', F', F'' \in \mathsf{MPA}^{s*}(A, \mathsf{Var})$, $E \equiv E' + E''$, $F \equiv F' + F''$ and the equalities $(E' = F')$ and $(E'' = F'')$ are provable from $\mathcal{E}$ by shorter inferences.

  1. Note, first of all, that, for all $E_1, E_2 \in \mathsf{MPA}^{s*}(A, \mathsf{Var})$,

     $$\mathsf{IB}(E_1 + E_2) = \max(\mathsf{IB}(E_1), \mathsf{IB}(E_2)) \ . \tag{46}$$

     Now we may argue thus:

     $$\begin{array}{rcll} \mathsf{IB}(E) > \mathsf{IB}(\mathcal{E}) & \Leftrightarrow & \mathsf{IB}(E') > \mathsf{IB}(\mathcal{E}) \text{ or } \mathsf{IB}(E'') > \mathsf{IB}(\mathcal{E}) & (46) \\ & \Leftrightarrow & \mathsf{IB}(F') > \mathsf{IB}(\mathcal{E}) \text{ or } \mathsf{IB}(F'') > \mathsf{IB}(\mathcal{E}) & \text{(By induction)} \\ & \Leftrightarrow & \mathsf{IB}(F) > \mathsf{IB}(\mathcal{E}) & (46) \end{array}$$

     and we are done.

  2. Assume that $\mathsf{IB}(E) > \mathsf{IB}(\mathcal{E})$ and $\mathsf{IB}(F) > \mathsf{IB}(\mathcal{E})$. By (46) we may assume, without loss of generality, that $\mathsf{IB}(E) = \mathsf{IB}(E')$. By the inductive hypothesis for statement 1, we derive that $\mathsf{IB}(F') > \mathsf{IB}(\mathcal{E})$. Therefore, using the inductive hypothesis for statement 2, we may infer that $\mathsf{IB}(E) = \mathsf{IB}(E') = \mathsf{IB}(F')$. By (46), the claim will now follow if we prove that

     $$\mathsf{IB}(F'') \ \leq \ \mathsf{IB}(F') \ .$$

     As, by our assumptions $\mathsf{IB}(F') = \mathsf{IB}(E) > \mathsf{IB}(\mathcal{E})$, this is immediate if $\mathsf{IB}(F'') \leq \mathsf{IB}(\mathcal{E})$. Otherwise, we have that $\mathsf{IB}(F'') > \mathsf{IB}(\mathcal{E})$. By the inductive hypothesis for statement 1, we infer that $\mathsf{IB}(E'') > \mathsf{IB}(\mathcal{E})$. We may therefore apply the inductive hypothesis for statement 2 to derive that $\mathsf{IB}(F'') = \mathsf{IB}(E'') \leq \mathsf{IB}(E) = \mathsf{IB}(E') = \mathsf{IB}(F')$, as desired.

     Hence we have proven that $\mathsf{IB}(E) = \mathsf{IB}(F)$.

- **Case:** The equality $E = F$ is proven using the substitutivity rule for string iteration, i.e., $\mathcal{E} \vdash E = F$ because, for some string $w \in A^+$ and terms $E', F' \in \mathsf{MPA}^{s*}(A, \mathsf{Var})$, $E \equiv w^*E'$, $F \equiv w^*F'$ and the equality $(E' = F')$ is provable from $\mathcal{E}$ by a shorter inference.

  1. Note, first of all, that, for all $G \in \mathsf{MPA}^{s*}(A, \mathsf{Var})$,

     $$\mathsf{IB}(w^*G) = \max(\mathsf{length}(w), \mathsf{IB}(G)) \ . \tag{47}$$

     The claim now follows immediately by this fact and the inductive hypothesis for statement 1.

2. Assume that $\mathsf{IB}(E) > \mathsf{IB}(\mathcal{E})$ and $\mathsf{IB}(F) > \mathsf{IB}(\mathcal{E})$. We shall prove that $\mathsf{IB}(E) = \mathsf{IB}(F)$. We proceed by distinguishing two cases depending on whether $\mathsf{IB}(E') > \mathsf{IB}(\mathcal{E})$ or not.
   - Case: $\mathsf{IB}(E') > \mathsf{IB}(\mathcal{E})$.
     In this case, the inductive hypothesis for statement 1 gives that $\mathsf{IB}(F') > \mathsf{IB}(\mathcal{E})$. Therefore we may apply the inductive hypothesis for statement 2 to the inference $\mathcal{E} \vdash E' = F'$ to derive that:

     $$\mathsf{IB}(E') = \mathsf{IB}(F') \ . \tag{48}$$

     The claim now follows from the following chain of equalities:

     $$\begin{array}{rcll} \mathsf{IB}(E) & = & \max(\mathsf{length}(w), \mathsf{IB}(E')) & (47) \\ & = & \max(\mathsf{length}(w), \mathsf{IB}(F')) & (48) \\ & = & \mathsf{IB}(F) & (47) \end{array}$$

   - Case: $\mathsf{IB}(E') \leq \mathsf{IB}(\mathcal{E})$.
     In this case, the inductive hypothesis for statement 1 gives that $\mathsf{IB}(F') \leq \mathsf{IB}(\mathcal{E})$ also holds. The claim now follows because, as $\mathsf{IB}(E)$ and $\mathsf{IB}(F)$ are strictly greater than $\mathsf{IB}(\mathcal{E})$ by the assumptions of the statement, it must be the case that

     $$\mathsf{IB}(E) = \mathsf{length}(w) = \mathsf{IB}(F) \ .$$

This completes the proof of the lemma. □

Using Lem. 34, we can finally prove that bisimulation equivalence cannot be finitely axiomatized over $\mathsf{MPA}^{s*}(A)$, unless the set of actions A is empty.

Theorem 35. *Assume that the set of actions A is non-empty. Then no finite collection of sound equations over the signature of $\mathsf{MPA}^{s*}(A, \mathsf{Var})$ can be complete for bisimulation equivalence over the language $\mathsf{MPA}^{s*}(A)$.*

Proof. Assume that A is a non-empty set of actions. Let $\mathcal{E}$ be a finite collection of sound equations over the signature of $\mathsf{MPA}^{s*}(A, \mathsf{Var})$. We exhibit a sound equality $P = Q$ that $\mathcal{E}$ cannot prove, thus showing the incompleteness of the theory $\mathcal{E}$.

Let $k \triangleq \mathsf{IB}(\mathcal{E})$. Consider the pair of terms $P \triangleq (a^{k+1})^*\delta$ and $Q \triangleq (a^{k+2})^*\delta$. It is not hard to see that $P \leftrightarroweq Q$. In fact, this follows from Lem. 15 and the fact that $\mathsf{MPA}^{s*} \vdash P = Q$ by using equation S3 twice. However, by Lem. 34(2), $\mathcal{E} \not\vdash P = Q$ because $\mathsf{IB}(P) > k$ and $\mathsf{IB}(Q) > k$, but $\mathsf{IB}(P) \neq \mathsf{IB}(Q)$. □

6 Extensions to $\mathsf{BPA}^{s*}_{\delta\epsilon}(A)$

The results that we have presented so far can be extended to the language $\mathsf{BPA}^{s*}_{\delta\epsilon}(A)$, obtained by augmenting BPA_δ [4] with the empty process ϵ from [23] and with string

$$
\begin{array}{cc}
a \xrightarrow{a} \epsilon & \epsilon \xrightarrow{\checkmark} \delta \\[2ex]
\dfrac{P \xrightarrow{\mu} P'}{P+Q \xrightarrow{\mu} P'} & \dfrac{Q \xrightarrow{\mu} Q'}{P+Q \xrightarrow{\mu} Q'} \\[3ex]
\dfrac{P \xrightarrow{a} P'}{PQ \xrightarrow{a} P'Q} & \dfrac{P \xrightarrow{\checkmark} P' \quad Q \xrightarrow{\mu} Q'}{PQ \xrightarrow{\mu} Q'} \\[3ex]
(aw)^*P \xrightarrow{a} w(aw)^*P & \dfrac{P \xrightarrow{\mu} P'}{(aw)^*P \xrightarrow{\mu} P'}
\end{array}
$$

Table 4: The operational rules for $\mathsf{BPA}^{s*}_{\delta\epsilon}(A)$ ($a \in A$, $\mu \in A_{\checkmark}$)

iteration. We shall spare the reader the tedious details of the proofs, and only indicate the extra ingredients needed for the proofs presented so far to go through over this language, when there are any.

The language of $\mathsf{BPA}^{s*}_{\delta\epsilon}(A,\mathsf{Var})$ terms is generated by the following grammar:

$$E ::= a \mid \delta \mid \epsilon \mid x \mid E+F \mid EF \mid w^*E$$

where $a \in A$, $w \in A^+$ and $x \in \mathsf{Var}$. The notion of k-bounded term introduced in Sect. 3 applies equally well to terms in the language $\mathsf{BPA}^{s*}_{\delta\epsilon}(A,\mathsf{Var})$, and we shall take the liberty of talking about k-bounded $\mathsf{BPA}^{s*}_{\delta\epsilon}(A,\mathsf{Var})$ terms in the technical statements of this section. The set of closed terms generated by the above grammar will be denoted by $\mathsf{BPA}^{s*}_{\delta\epsilon}(A)$ ($P, Q, P' \ldots \in \mathsf{BPA}^{s*}_{\delta\epsilon}(A)$).

The operational semantics for the language $\mathsf{BPA}^{s*}_{\delta\epsilon}(A)$ is given by the labelled transition system

$$(\mathsf{BPA}^{s*}_{\delta\epsilon}(A), A_{\checkmark}, \{\xrightarrow{\mu} \mid \mu \in A_{\checkmark}\})$$

specified by the Plotkin-style operational rules in Table 4, where $A_{\checkmark} \triangleq A \cup \{\checkmark\}$ and $\checkmark$ is a fresh action symbol used to denote successful termination. The operational rules in Table 4 are by now completely standard; here we only remark that the operational treatment of sequential composition we adopt is taken from [17]. It is easy to see that whenever the transition $P \xrightarrow{\checkmark} Q$ can be derived from the rules in Table 4, then $Q \equiv \delta$.

Bisimulation equivalence over $\mathsf{BPA}^{s*}_{\delta\epsilon}(A)$, denoted by $\leftrightarrow$ with abuse of notation, is defined by extending the notion of bisimulation relation given in Def. 10 with the extra requirement that whenever $P \,\Re\, Q$ and $P \xrightarrow{\checkmark} P'$, then $Q \xrightarrow{\checkmark} Q'$ for some Q'.

As the rules in Table 4 are in *tyft/tyxt*-format [17], bisimulation equivalence is a congruence over the language $\mathsf{BPA}^{s*}_{\delta\epsilon}(A)$. We shall now show how the results on axiomatizations of bisimulation equivalence presented in the previous sections can be extended to the language $\mathsf{BPA}^{s*}_{\delta\epsilon}(A)$.

A4	$(x+y)z = xz + yz$
A5	$(xy)z = x(yz)$
A7	$\delta x = \delta$
A8	$x\epsilon = x$
A9	$\epsilon x = x$
S8	$(w^*x)y = w^*(xy)$

Table 5: The extra axioms for $\mathsf{BPA}^{s*}_{\delta\epsilon}(A)$ ($w \in A^+$)

Let $\mathsf{BPA}^{s*}_{\delta\epsilon}$ denote the equational theory obtained by extending the one in Table 2 with the equations in Table 5. Equations A4, A5 and A7–A9 are familiar from the various flavours of the algebra BPA [4] with or without features like the deadlocked process δ and the empty process ϵ. Equation S8 is an instance of law SEI2 from [6]. Versions of this equation dealing with the so-called *prefix iteration* may be found in [13, 15]. By analogy with our previous terminology, we say that an equation $(E = F)$ in the theory $\mathsf{BPA}^{s*}_{\delta\epsilon}$ is k-bounded if both E and F are k-bounded $\mathsf{BPA}^{s*}_{\delta\epsilon}(A, \mathsf{Var})$ terms.

Theorem 36. *The axiom system $\mathsf{BPA}^{s*}_{\delta\epsilon}$ is sound and complete for bisimulation equivalence over the language $\mathsf{BPA}^{s*}_{\delta\epsilon}(A)$, i.e. for all $\mathsf{BPA}^{s*}_{\delta\epsilon}(A)$ terms P and Q,*

$$P \underline{\leftrightarrow} Q \quad \Leftrightarrow \quad \mathsf{BPA}^{s*}_{\delta\epsilon} \vdash P = Q.$$

Moreover, for k-bounded $\mathsf{BPA}^{s*}_{\delta\epsilon}(A)$-terms P and Q, if $P \underline{\leftrightarrow} Q$ then the equality $P = Q$ can be proved by using only k-bounded equations in the theory $\mathsf{BPA}^{s*}_{\delta\epsilon}$.

Proof. (Sketch) We only give a hint on how the proof of Thm. 29 can be adapted to show the completeness of the theory $\mathsf{BPA}^{s*}_{\delta\epsilon}$ for bisimulation equivalence over the language $\mathsf{BPA}^{s*}_{\delta\epsilon}(A)$.

First of all, note that it is sufficient to prove completeness of the theory $\mathsf{BPA}^{s*}_{\delta\epsilon}$ for bisimulation equivalence over a subset of $\mathsf{BPA}^{s*}_{\delta\epsilon}(A)$, namely that of *basic terms*. A $\mathsf{BPA}^{s*}_{\delta\epsilon}(A)$ term is said to be basic iff it can be generated by the following grammar:

$$P ::= \delta \mid \epsilon \mid aP \mid P + P \mid w^*P \ .$$

Intuitively, basic terms are $\mathsf{BPA}^{s*}_{\delta\epsilon}(A)$ terms in which action prefixing is used in lieu of general sequential composition. A straightforward argument by induction on the size of of $\mathsf{BPA}^{s*}_{\delta\epsilon}(A)$ terms shows that, for every $\mathsf{BPA}^{s*}_{\delta\epsilon}(A)$ term P, there exists a basic term Q such that

$$\text{A4,A5,A7–A9,S8} \vdash P = Q \ .$$

This statement justifies our previous claim that it is sufficient to show completeness for basic terms.

The fact that the theory $\mathsf{BPA}^{s*}_{\delta\epsilon}$ is complete for bisimulation equivalence over basic terms can be shown by a painstaking reworking of the proof of Thm. 29 for this language.

Here we confine ourselves to remarking that in the reworking of the proof of the inductive step for the statement

$$P \leftrightarroweq P + Q \text{ implies } \mathsf{BPA}^{s*}_{\delta\epsilon} \vdash Q \subseteq P$$

we make use of the following addition to Lem. 26, which shows how to absorb ✓-labelled transitions:

$$\text{For all } P, P' \in \mathsf{BPA}^{s*}_{\delta\epsilon}(A), \quad P \xrightarrow{\checkmark} P' \quad \text{implies} \quad \text{A3,A4,A9,S1} \vdash \epsilon \subseteq P \ .$$

□

As it was the case for the language $\mathsf{MPA}^{s*}(A)$, the above result shows that, for every positive integer k, bisimulation equivalence can be finitely axiomatized over the language of k-bounded $\mathsf{BPA}^{s*}_{\delta\epsilon}(A)$ terms. The equational theory $\mathsf{BPA}^{s*}_{\delta\epsilon}$ that axiomatizes bisimulation equivalence over the whole of $\mathsf{BPA}^{s*}_{\delta\epsilon}(A)$ is, however, infinite. A careful reworking of the proof of Thm. 35 shows that, unless the set of actions A is empty, no finite, complete axiomatization of bisimulation equivalence over $\mathsf{BPA}^{s*}_{\delta\epsilon}(A)$ can exist

Theorem 37. *Assume that the set of actions A is non-empty. Then no finite collection of sound equations over the signature of $\mathsf{BPA}^{s*}_{\delta\epsilon}(A, \mathsf{Var})$ can be complete for bisimulation equivalence over the language $\mathsf{BPA}^{s*}_{\delta\epsilon}(A)$.*

Acknowledgements: The authors would like to thank Anna Ingólfsdóttir and Vladimiro Sassone for inspiring discussions on the subject of unique root decompositions for finite strings which led to the proof of Thm. 2.

References

[1] L. Aceto and A. Ingólfsdóttir, *A complete equational axiomatization for prefix iteration with silent steps*, Research Report RS–95–5, BRICS (Basic Research in Computer Science, Centre of the Danish National Research Foundation), Department of Mathematics and Computer Science, Aalborg University, Jan. 1995.

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

[3] 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.

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

[5] J. Bergstra, I. Bethke, and A. Ponse, *Process algebra with combinators*, Report P9319, Programming Research Group, University of Amsterdam, Amsterdam, 1993.

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

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

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

[9] F. Corradini, R. De Nicola, and A. Labella, *Fully abstract models for nondeterministic Kleene algebras (extended abstract)*, 1995. To appear in the *Proceedings of CONCUR 95*.

[10] 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.

[11] J. Engelfriet, *Determinacy → (observation equivalence = trace equivalence)*, Theoretical Comput. Sci., 36 (1985), pp. 21–25.

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

[13] ——, *A complete axiomatization for prefix iteration in branching bisimulation*, Logic Group Preprint Series 126, Dept. of Philosophy, Utrecht University, Jan. 1995.

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

[15] ——, *A complete equational axiomatization for BPA$\delta\epsilon$ with prefix iteration*, 1995. Unpublished manuscript.

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

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

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

[19] ——, *Algebraic Theory of Processes*, MIT Press, Cambridge, Massachusetts, 1988.

[20] C. Hoare, *Communicating Sequential Processes*, Prentice-Hall International, Englewood Cliffs, 1985.

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

[22] 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.

[23] C. Koymans and J. Vrancken, *Extending process algebra with the empty process* ϵ, Logic Group Preprint Series Nr. 1, CIF, State University of Utrecht, 1985.

[24] L. Lamport, *How to write a proof*, Research Report 94, Digital Equipment Corporation, Systems Research Center, Feb. 1993.

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

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

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

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

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

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

[31] 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.

Recent Publications in the BRICS Report Series

RS-95-28 Luca Aceto and Jan Friso Groote. *A Complete Equational Axiomatization for MPA with String Iteration*. May 1995. 39 pp.

RS-95-27 David Janin and Igor Walukiewicz. *Automata for the μ-calculus and related results*. May 1995. 11 pp. To appear in *Mathematical Foundations of Computer Science: 20th Int. Symposium*, MFCS '95 Proceedings, LNCS, 1995.

RS-95-26 Faith Fich and Peter Bro Miltersen. *Tables should be sorted (on random access machines)*. May 1995. 11 pp. To appear in *Algorithms and Data Structures: 4th Workshop*, WADS '95 Proceedings, LNCS, 1995.

RS-95-25 Søren B. Lassen. *Basic Action Theory*. May 1995. 47 pp.

RS-95-24 Peter Ørbæk. *Can you Trust your Data?* April 1995. 15 pp. Appears in Mosses, Nielsen, and Schwartzbach, editors, *Theory and Practice of Software Development. 6th International Joint Conference CAAP/FASE*, TAPSOFT '95 Proceedings, LNCS 915, 1995, pages 575–590.

RS-95-23 Allan Cheng and Mogens Nielsen. *Open Maps (at) Work*. April 1995. 33 pp.

RS-95-22 Anna Ingólfsdóttir. *A Semantic Theory for Value–Passing Processes, Late Approach, Part II: A Behavioural Semantics and Full Abstractness*. April 1995. 33 pp.

RS-95-21 Jesper G. Henriksen, Ole J. L. Jensen, Michael E. Jørgensen, Nils Klarlund, Robert Paige, Theis Rauhe, and Anders B. Sandholm. *MONA: Monadic Second-Order Logic in Practice*. May 1995. 17 pp.

RS-95-20 Anders Kock. *The Constructive Lift Monad*. March 1995. 18 pp.

RS-95-19 François Laroussinie and Kim G. Larsen. *Compositional Model Checking of Real Time Systems*. March 1995. 20 pp.

RS-95-18 Allan Cheng. *Complexity Results for Model Checking*. February 1995. 18pp.