Deriving Complete Inference Systems for a Class of GSOS Languages Generating Regular Behaviours

Luca Aceto*
School of Cognitive and Computing Studies,
University of Sussex,
Falmer, Brighton BN1 9QH, England
email: luca@cogs.sussex.ac.uk

Abstract

Many process algebras are defined by structural operational semantics (SOS). Indeed, most such definitions are nicely structured and fit the GSOS format of [13]. In [2] B. Bloom, F. Vaandrager and I presented a procedure for converting any GSOS language definition to a finite complete equational axiom system which precisely characterizes strong bisimulation of processes. For recursion theoretic reasons, such a complete equational axiom system included, in general, one infinitary induction principle — essentially a reformulation of the Approximation Induction Principle (AIP) [7, 6].

However, it is well-known that AIP and other infinitary proof rules are not necessary for the axiomatization of, *e.g.*, strong bisimulation over regular behaviours (see [29, 8, 31]). In this paper, following [1], I characterize a class of *infinitary* GSOS specifications, obtained by relaxing some of the finiteness constraints of the original format of Bloom, Istrail and Meyer, which generate regular processes. I then show how the techniques of [2] can be adapted to give a procedure for converting any such language definition to a complete equational axiom system for strong bisimulation of processes which does not use infinitary proof rules. Equalities between recursive, regular processes can be established in the resulting inference systems by means of standard axioms to unwind recursive definitions, and the so-called *Recursive Specification Principle* (RSP) [6].

It turns out that not all the infinitary GSOS systems from [1] which generate finite labelled transition systems are amenable to the development of a corresponding equational theory à là [2]. A new class of infinitary GSOS systems which afford a nice algebraic treatment emerges from this study. I believe that this new class of infinitary GSOS specifications has some independent interest.

1991 Mathematics Subject Classification: 68Q05, 68Q55.
1991 CR Categories: D.1.3, D.3.1, F.1.1, F.3.1, F.3.2, F.3.3.
Keywords and Phrases: Structural Operational Semantics (SOS), GSOS format, labelled transition systems, process graphs, bisimulation equivalence, process algebra, regular processes, equational logic, complete axiomatizations, recursive specification principle.

1 Introduction

Since de Simone's pioneering work on the expressivity of the calculi SCCS [28] and MEIJE [3] (cf. the references [35, 36]), there has been considerable interest in the metatheory of process algebras. As

*The work reported in this paper was carried out while the author was at Aalborg University Centre, 9220 Aalborg Ø, Denmark, and was supported by the Danish Natural Science Research Council under project DART. The support of a grant under the British Council academic collaboration programme is also gratefully acknowledged.

many process algebras are defined by structural operational semantics (SOS) [34], this way of giving semantics to programming and specification languages has been a natural handle for proving results for classes of languages. In particular, several formats for SOS rules have emerged in the literature (see, *e.g.*, [35, 13, 20, 19, 14, 37, 5, 40, 10]) and a wealth of properties that hold for *all* languages specified in terms of rules which fit these formats have been established. (In addition to the previous references, the interested reader may wish to consult, *e.g.*, [9, 33, 26, 38, 2, 4, 39, 11, 12, 15, 1] for examples of this kind of metatheoretic results).

In [2] I gave a contribution to this line of research by presenting, together with B. Bloom and F. Vaandrager, a procedure for converting any language definition in the GSOS format of Bloom, Istrail and Meyer [13, 9] to a finite complete equational axiom system which precisely characterizes strong bisimulation of processes. Such a complete equational axiom system included, in general, one infinitary induction principle — essentially a reformulation of the Approximation Induction Principle (AIP) [7, 6]. An infinitary proof rule like AIP is indeed necessary to obtain completeness for arbitrary GSOS systems because, as shown in [2], testing bisimulation over GSOS systems is Π^0_1-complete.

However, it is well-known that AIP and other infinitary proof rules are not necessary for the axiomatization of, *e.g.*, strong bisimulation over regular behaviours (see the classic references [29, 8, 31]). Thus it should be possible to fine tune the methods of [2] to produce complete inference systems for strong bisimulation over classes of GSOS systems that generate regular behaviours which do not rely on infinitary proof rules like AIP. This is the aim of this paper.

1.1 Results

In this paper, I give a procedure for extracting from a GSOS specification that generates regular processes a complete axiom system for strong bisimulation equivalence. This axiom system is equational, except for one conditional equation, and does not rely on infinitary proof rules.

First of all, following [1], I characterize a class of *infinitary* GSOS specifications, obtained by relaxing some of the finiteness constraints of the original format of Bloom, Istrail and Meyer [13, 9], which has some of the basic sanity properties of the original GSOS format. For example, it will ensure that the transition relation induced by the rules will be finitely branching. Syntactic restrictions are then imposed on the rules in these infinitary GSOS specifications to ensure that the semantics of processes is given by finite process graphs. The result is a class of infinitary GSOS systems that includes most of the standard operations used in the literature on process algebras.

I then show how the techniques of [2] can be adapted to give a procedure for converting any such language definition to a complete equational axiom system for strong bisimulation of processes which does not use infinitary proof rules. Equalities between recursive, regular processes can be established in the resulting inference systems by means of standard axioms to unwind recursive definitions, and the so-called *Recursive Specification Principle* (RSP) [6].

Interestingly, it turns out that not all the infinitary GSOS systems from [1] which generate finite labelled transition systems are amenable to the development of a corresponding equational theory à là [2]. In particular, operations that use their arguments in unboundedly many different ways can still generate finite process graphs, but cannot be axiomatized in finitary fashion — at least not using the techniques of [2]. As an example, consider the operation f with rules (one such rule for each $i \in \omega$ and $j \leq i$):

$$\frac{x \stackrel{a_i}{\rightarrow} y}{f(x) \stackrel{a_j}{\rightarrow} \mathbf{0}} \qquad (1)$$

For each process P that can initially perform only a finite number of different actions, the semantics of $f(P)$ is given by a finite process graph. However, such an operation cannot be axiomatized in finitary fashion using the techniques of [2] because there is no upper bound on the number of different rules for it which have the same hypothesis. (See Proposition 5.10). Similarly, operations that have no upper bound on the number of positive hypotheses for their arguments, *i.e.* antecedents like $x \xrightarrow{a_i} y$ in rule (1), do not lend themselves to a clean algebraic description using the methods of [2]. (See Proposition 5.14).

However, for GSOS systems whose operations are defined by rules without negative hypotheses, it is possible to give a reasonably aesthetic axiomatization of operations like the one given by the rules (1). A revised strategy that can be used to axiomatize these operations is presented in Section 7.1. When applied to the operation f described by the rules (1), the revised strategy produces the following natural equations:

$$\begin{aligned} f(\mathbf{0}) &= \mathbf{0} \\ f(x+y) &= f(x)+f(y) \\ f(a_i.x) &= \sum_{1\leq j\leq i} a_j.\mathbf{0} \qquad (a_i \in \mathsf{Act}) \end{aligned}$$

A new class of infinitary GSOS systems which afford a nice algebraic treatment emerges from this study. I believe that this new class of infinitary GSOS specifications has some independent interest, and will form the basis for a general treatment of infinitary GSOS languages which enjoy most of the sanity properties of the original proposal of Bloom, Istrail and Meyer.

1.2 Outline of the Paper

The paper is organized as follows. Section 2 is devoted to a review of background material from the theory of structural operational semantics and process algebras that will be needed in this study. Section 3 introduces the class of regular infinitary GSOS systems that will be axiomatized in Section 5. This is a subclass of the infinitary simple GSOS systems from [1] which afford a clean algebraic treatment. Section 5 presents an adaptation of the techniques from [2] to regular infinitary GSOS systems and two simple impossibility results which motivate the restriction to the class of systems under consideration. The algorithm of Section 5 produces an equational theory which is strongly head normalizing for all processes, *i.e.*, that allows one to prove that a process P is equivalent to the sum of its initial derivatives. In Section 6, I discuss the completeness of the inference system obtained by extending the resulting equational theory with the recursive specification principle. Section 7.1 presents an alternative strategy that can be used to axiomatize positive GSOS operations, *i.e.* operations whose operational semantics is defined by rules without negative hypotheses. This revised strategy can be used to axiomatize operations that, like the one given by the rules (1), are bounded, but not uniformly bounded, in the sense of Definition 3.2. Finally, in Section 7.2, I discuss some directions for further research.

Familiarity with [2, 1] will be helpful, but not necessary, in reading the paper. As this is not an introductory paper on deriving complete axiomatizations from SOS rules, I shall often refer the reader to the literature for examples and motivations. Precise pointers to the literature will be given wherever necessary.

2 Preliminaries

I assume that the reader is familiar with the basic notions of process algebra and structural operational semantics; see, *e.g.*, [25, 23, 30, 6, 34, 20, 13, 9] for more details and extensive motivations.

Let Var be a denumerable set of *process variables* ranged over by x, y. (For technical convenience, I shall assume throughout that the set Var can always be extended). A *signature* Σ consists of a set of *operation symbols*, disjoint from Var, together with a function *arity* that assigns a natural number to each operation symbol. The set $\mathbb{T}(\Sigma)$ of *terms* over Σ is the least set such that

- Each $x \in \mathsf{Var}$ is a term.
- If f is an operation symbol of arity l, and $P_1, \ldots, P_l$ are terms, then $f(P_1, \ldots, P_l)$ is a term.

I shall use $P, Q, \ldots$ to range over terms and the symbol $\equiv$ for the relation of syntactic equality on terms. $\mathrm{T}(\Sigma)$ is the set of *closed* terms over Σ, *i.e.*, terms that do not contain variables. Constants, *i.e.* terms of the form $f()$, will be abbreviated as f. A (closed) Σ-substitution is a mapping σ from the set of variables Var to the set of (closed) terms over Σ. The notation $\{P_1/x_1, \ldots, P_n/x_n\}$, where the P_is are terms and the x_is are distinct variables, will often be used to denote the substitution that maps each x_i to P_i, and leaves all the other variables unchanged.

A Σ-*context* $C[\vec{x}]$ is a term in which at most the variables $\vec{x}$ appear. $C[\vec{P}]$ is $C[\vec{x}]$ with x_i replaced by P_i wherever it occurs. In this paper, substitutions of open terms for variables will only be used in the absence of binding operations. For this reason, I take the liberty of using this simple definition of substitution, and omit the standard details of the formal definition.

Besides terms I have *actions*, elements of some given countable[1] set Act, which is ranged over by a, b, c.

Definition 2.1 (GSOS Rules and Infinitary GSOS Systems) *Suppose Σ is a signature. A* GSOS rule *ρ over Σ is an inference rule of the form*[2]*:*

$$\frac{\bigcup_{i=1}^{l}\left\{x_i \stackrel{a_{ij}}{\rightarrow} y_{ij} \mid 1 \leq j \leq m_i\right\} \;\cup\; \bigcup_{i=1}^{l}\left\{x_i \stackrel{b_{ik}}{\not\rightarrow} \mid 1 \leq k \leq n_i\right\}}{f(x_1, \ldots, x_l) \stackrel{c}{\rightarrow} C[\vec{x}, \vec{y}]} \qquad (2)$$

where all the variables are distinct, $m_i, n_i \geq 0$, f is an operation symbol from Σ with arity l, $C[\vec{x}, \vec{y}]$ is a Σ-context, and the a_{ij}, b_{ik}, and c are actions in Act.
An infinitary GSOS system *is a pair $G = (\Sigma_G, R_G)$, where Σ_G is a countable signature and R_G is a countable set of GSOS rules over Σ_G.*

It is useful to name components of rules of the form (2). The operation symbol f is the *principal operation* of the rule, and the term $f(\vec{x})$ is the *source*. $C[\vec{x}, \vec{y}]$ is the *target* (sometimes denoted by $\mathsf{target}(\rho)$); c is the *action*; the formulas above the line are the *antecedents* (sometimes denoted by $\mathsf{ante}(\rho)$); and the formula below the line is the *consequent*. If, for some i, $m_i > 0$ then I say that *ρ tests its i-th argument positively*. Similarly if $n_i > 0$ then I say that *ρ tests its i-th argument negatively*. An operation *f tests its i-th argument positively (resp. negatively)* if it occurs as principal operation of a rule that tests its i-th argument positively (resp. negatively).

GSOS systems have been introduced and studied in depth in [13, 9]. The reader familiar with those references may have noticed that infinitary GSOS systems, unlike the GSOS systems in [13, 9], are not required to consist of a finite signature and a finite set of GSOS rules. This slight generalization of the original definition will allow me to deal with calculi which, like CCS [30] and

[1] A set X is countable if it is empty or if there exists an enumeration of X, that is a surjective mapping from the set of positive integers onto X.

[2] The format for GSOS rules considered in this paper is the original one of Bloom, Istrail and Meyer. However, all the results in this paper hold for a generalized version of GSOS rule in which an infinite number of negative antecedents is allowed.

MEIJE [3], postulate an infinite action set. In the setting of this paper, it will also be natural to treat languages with a denumerable set of operations. (See, *e.g.*, Section 4).

Intuitively, an infinitary GSOS system gives a language, whose constructs are the operations in the signature Σ_G, together with a Plotkin-style structural operational semantics [34] for it defined by the set of conditional rules R_G. Informally, the intent of a GSOS rule is as follows. Suppose that we are wondering whether $f(\vec{P})$ is capable of taking a c-step. We look at each rule with principal operation f and action c in turn. We inspect each positive antecedent $x_i \stackrel{a_{ij}}{\rightarrow} y_{ij}$, checking if P_i is capable of taking an a_{ij}-step for each j and if so calling the a_{ij}-children Q_{ij}. We also check the negative antecedents; if P_i is *in*capable of taking a b_{ik}-step for each k. If so, then the rule *fires* and $f(\vec{P}) \stackrel{c}{\rightarrow} C[\vec{P}, \vec{Q}]$. Roughly, this means that the transition relation associated with an infinitary GSOS system, notation $\rightarrow_G$, is the one defined by structural induction on terms using the rules in R_G. This essentially ensures that a transition $f(\vec{P}) \stackrel{a}{\rightarrow}_G Q$ exists between the closed terms $f(\vec{P})$ and Q iff there exist a closed substitution σ, and a rule for f whose antecedents hold when instantiated with σ, and whose instantiated target yields Q. The interested reader is referred to [13, 9] for the details of the formal definition of $\rightarrow_G$.

As usual, the operational semantics for the closed terms over Σ_G will be given in terms of the notion of labelled transition system.

Definition 2.2 (Labelled Transition Systems and Process Graphs) *Let A be a set of labels. A* labelled transition system (lts) *is a pair $(S, \rightarrow)$, where S is a set of* states *and $\rightarrow \subseteq S \times A \times S$ is the* transition relation. *As usual, I shall write $s \stackrel{a}{\rightarrow} t$ in lieu of $(s, a, t) \in \rightarrow$. A state t is* reachable *from state s if there exist states $s_0, \ldots, s_n$ and labels $a_1, \ldots, a_n$ such that*

$$s = s_0 \stackrel{a_1}{\rightarrow} s_1 \stackrel{a_2}{\rightarrow} \cdots \stackrel{a_n}{\rightarrow} s_n = t$$

The set of states which are reachable from s, also known as the set of derivatives *of s, will be denoted by $\mathsf{der}(s)$.*
A process graph *is a triple $(r, S, \rightarrow)$, where $(S, \rightarrow)$ is an lts, $r \in S$ is the* root, *and each state in S is reachable from r. If $(S, \rightarrow)$ is an lts and $s \in S$ then $\mathsf{graph}(s, (S, \rightarrow))$ is the process graph obtained by taking s as the root and restricting $(S, \rightarrow)$ to the part reachable from s. I shall write $\mathsf{graph}(s)$ for $\mathsf{graph}(s, (S, \rightarrow))$ whenever the underlying lts $(S, \rightarrow)$ is understood from the context.*
An lts $(S, \rightarrow)$ is finite *iff S and $\rightarrow$ are finite sets. A process graph $\mathsf{graph}(s, (S, \rightarrow))$ is finite if the restriction of $(S, \rightarrow)$ to the part reachable from s is.*

The lts specified by an infinitary GSOS system G is then given by $\mathsf{lts}(G) = (\mathrm{T}(\Sigma_G), \rightarrow_G)$ and the process graph defining the operational semantics of a closed term P is $\mathsf{graph}(P, \mathsf{lts}(G))$ (abbreviated to $\mathsf{graph}(P)$ when the infinitary GSOS system G is clear from context).

The basic notion of equivalence among terms of an infinitary GSOS system I shall consider in this paper is that of *bisimulation* [32, 28].

Definition 2.3 *Suppose G is an infinitary GSOS system. A binary relation $\sim \; \subseteq \mathrm{T}(\Sigma_G) \times \mathrm{T}(\Sigma_G)$ over closed terms is a* bisimulation *if $P \sim Q$ implies, for all $a \in \mathsf{Act}$,*

1. *If $P \stackrel{a}{\rightarrow}_G P'$ then, for some Q', $Q \stackrel{a}{\rightarrow}_G Q'$ and $P' \sim Q'$.*

2. *If $Q \stackrel{a}{\rightarrow}_G Q'$ then, for some P', $P \stackrel{a}{\rightarrow}_G P'$ and $P' \sim Q'$.*

I write $P \leftrightarrows_G Q$ if there exists a bisimulation $\sim$ relating P and Q. The subscript G is omitted when it is clear from context.

Lemma 2.4 *Suppose G is an infinitary GSOS system. Then $\leftrightarroweq_G$ is an equivalence relation and a congruence for all operation symbols f of G,* i.e., $(\forall i : P_i \leftrightarroweq_G Q_i) \Rightarrow f(\vec{P}) \leftrightarroweq_G f(\vec{Q})$.

Proof: Infinitary GSOS systems are well-founded, stratifiable transition systems specifications in *ntyft* format in the sense of Bol and Groote [19, 14], with the number of operation symbols in a term providing the stratification. The result then follows immediately from the congruence theorems given in the aforementioned references. □

For an infinitary GSOS system G, I shall write $\mathsf{Bisim}(G)$ for the quotient algebra of closed Σ_G-terms modulo bisimulation. That is, for $P, Q \in \mathbb{T}(\Sigma_G)$,

$$\mathsf{Bisim}(G) \models P = Q \quad \Leftrightarrow \quad (\forall \text{ closed } \Sigma_G\text{-substitutions } \sigma : P\sigma \leftrightarroweq_G Q\sigma).$$

The following notion from [2] will be useful in the remainder of this paper.

Definition 2.5 *An infinitary GSOS system H is a* disjoint extension *of an infinitary GSOS system G, notation $G \sqsubseteq H$, if the signature and rules of H include those of G, and H introduces no new rules for operations of G.*

If H disjointly extends G then H introduces no new outgoing transitions for terms of G. This means in particular that, for $P, Q \in \mathrm{T}(\Sigma_G)$, $P \leftrightarroweq_G Q \Leftrightarrow P \leftrightarroweq_H Q$.

The notion of disjoint extension of an infinitary GSOS system is closely related to that of sum of two transition system specifications due to Groote and Vaandrager [20, Definition 7.3].

Definition 2.6 *Let Σ_0 and Σ_1 be signatures whose arity functions agree over their common operation symbols. The* sum *of Σ_0 and Σ_1, notation $\Sigma_0 \oplus \Sigma_1$, is the signature whose operation symbols are those occurring in Σ_0 or Σ_1, and whose function arity is the one with graph given by the union of the graphs of the arity functions of Σ_0 and Σ_1.*

Let $G_i = (\Sigma_{G_i}, R_{G_i})$ $(i = 0, 1)$ be two infinitary GSOS systems with $\Sigma_0 \oplus \Sigma_1$ defined. The sum of G_0 and G_1, notation $G_0 \oplus G_1$, is the infinitary GSOS system:

$$G_0 \oplus G_1 \;=\; (\Sigma_{G_0} \oplus \Sigma_{G_1}, R_{G_0} \cup R_{G_1})$$

It is immediate to see that if an infinitary GSOS system H disjointly extends G then $H = G \oplus H$. Conversely, if $H = G \oplus H$ and H introduces no new rules for operations of G, then $G \sqsubseteq H$.

In this paper, I shall be interested in equations which are preserved by taking disjoint extensions of infinitary GSOS systems. Following [2], I thus introduce, for G an infinitary GSOS system, the class $\mathsf{BISIM}(G)$ of all algebras $\mathsf{Bisim}(G')$, for G' a disjoint extension of G. Thus we have, for $P, Q \in \mathbb{T}(\Sigma_G)$,

$$\mathsf{BISIM}(G) \models P = Q \quad \Leftrightarrow \quad (\forall G' : G \sqsubseteq G' \;\Rightarrow\; \mathsf{Bisim}(G') \models P = Q).$$

3 Regular Infinitary GSOS Systems

In this section, I shall briefly show how to impose syntactic restrictions on the format of rules in an infinitary GSOS system G which ensure that $\mathsf{graph}(P)$ is a finite process graph for each $P \in \mathrm{T}(\Sigma_G)$. Some of the results to follow are from [1], and I refer the reader to that reference for intuition and examples. The class of infinitary GSOS systems which will be considered in this paper will be a subclass of the simple GSOS systems of [1] which allows for the development of a clean algebraic theory.

In order to obtain that $\mathsf{graph}(P)$ is a finite process graph for each closed term P, it is necessary to impose restrictions on the class of infinitary GSOS systems under consideration, ensuring that the transition relation be finitely branching and that the set of states reachable from P be finite. Finite branching of the transition relation $\rightarrow_G$ is one of the basic sanity properties of the original GSOS format of Bloom, Istrail and Meyer [13, 9]. However, in the presence of a possibly infinite action set and signature, it is easy to specify operations which give rise to infinitely branching process graphs, and explicit constraints ruling out this pathology must be imposed on infinitary GSOS systems.

Definition 3.1 *The* positive trigger *of rule (2) is the l-tuple over* 2^{Act} $\langle e_1, \ldots, e_l \rangle$*, where*

$$e_i = \{a_{ij} | 1 \leq j \leq m_i\}$$

When writing positive triggers, I shall often identify a singleton set $\{a\}$ with the action a.

Definition 3.2 (Boundedness and Uniform Boundedness) *An operation f in an infinitary GSOS system is* bounded *iff for each positive trigger, the corresponding set of rules for f is finite. An infinitary GSOS system is* bounded *iff each of its operations is.*

An operation in an infinitary GSOS system is uniformly bounded *iff there exists an upper bound n_f on the number of distinct rules for f having the same positive trigger. An infinitary GSOS system is* uniformly bounded *iff each of its operations is.*

The notion of bounded infinitary GSOS system is from [1], and is inspired by ideas developed by Vaandrager [39, Definition 3.2] for de Simone systems. The notion of uniform boundedness is new in this paper, and will play an important role in Section 5.1. (In particular, it will be crucial in the proof of Proposition 5.9). Of course, every uniformly bounded operation is also bounded. The following example shows that the converse is not true.

Example: Let $\mathsf{Act} = \{a_i \mid i \geq 1\}$ be a denumerable set of actions. Consider an infinitary GSOS system G comprising a unary operation f and constant $\mathbf{0}$, with rules (one such rule for each $i \in \omega$ and $j \leq i$):

$$\frac{x \xrightarrow{a_i} y}{f(x) \xrightarrow{a_j} \mathbf{0}} \tag{3}$$

Then f is bounded, as there are exactly i rules with trigger a_i for each $i \geq 1$. However, f is not uniformly bounded.

All the standard operations used in the literature on process algebras are uniformly bounded.

A bounded infinitary GSOS system associates a finitely branching process graph with each term. (*A fortiori*, this property is also true of uniformly bounded infinitary GSOS systems).

Lemma 3.3 ([1]) *For each infinitary, bounded GSOS system G, the transition relation $\rightarrow_G$ is finitely branching,* i.e. *for all $P \in \mathrm{T}(\Sigma_G)$, the set*

$$\left\{Q \mid \exists a \in \mathsf{Act} : \ P \xrightarrow{a}_G Q\right\}$$

is finite.

In order to characterize an interesting class of infinitary GSOS systems which generate finite process graphs, I shall now introduce a further restriction on infinitary GSOS systems that ensures that processes have a finite set of derivatives. The following notions are from [1].

Definition 3.4 *A GSOS rule of the form (2) is* simple *iff* $C[\vec{x}, \vec{y}]$ *is either a variable in* $\vec{x}, \vec{y}$ *or it is of the form* $g(z_1, \ldots, z_n)$ *where each* z_i *is a variable in* $\vec{x}, \vec{y}$. *An operation is* simple *iff all the rules for it are. An infinitary GSOS system* $G = (\Sigma_G, R_G)$ *is* simple *iff each operation in* Σ_G *is.*

Definition 3.5 *Let* $G = (\Sigma_G, R_G)$ *be a simple, infinitary GSOS system. The* operator dependency graph *associated with* G *is the directed graph with*

- Σ_G *as set of nodes, and*
- *set of edges* E *given by:* $(f, g) \in E$ *iff there exists a rule* $\rho \in R_G$ *with* f *as principal operation and target* $g(z_1, \ldots, z_n)$, *for some* $z_1, \ldots, z_n \in \mathsf{Var}$.

I shall write $f \prec_G g$ *iff* $f E^\star q$ *in the operator dependency graph for* G, *where* $E^\star$ *denotes the reflexive and transitive closure of* E.

Definition 3.6 (Very Simple GSOS Systems) *Let* $G = (\Sigma_G, R_G)$ *be a simple, infinitary GSOS system. An operation* $f \in \Sigma_G$ *is* very simple *iff it is uniformly bounded and* $\{g \mid f \prec_G g\}$ *is finite. A simple, infinitary GSOS system is* very simple *iff every operation in its signature is.*

The following result can be shown by structural induction on closed terms following the lines of [1, Theorem 5.5].

Proposition 3.7 *Let* $G = (\Sigma_G, R_G)$ *be a very simple GSOS system. Then, for all* $P \in \mathrm{T}(\Sigma_G)$, $\mathsf{graph}(P)$ *is a finite process graph.*

The following definition introduces the subclass of very simple GSOS systems that I shall study in the remainder of this paper. The following definition is new in this paper, and will play an important role in Section 5.2. (In particular, it will be crucial in the proof of Proposition 5.13).

Definition 3.8 (Regular GSOS Systems) *Let* $G = (\Sigma_G, R_G)$ *be an infinitary GSOS systems. An operation* $f \in \Sigma_G$ *has* limited fan-in *iff for every argument* i *for* f *there exists an upper bound* $m_{(f,i)}$ *on the number of distinct positive antecedents for argument* i *in the rules for* f *in* R_G. *I say that an operation* $f \in \Sigma_G$ *is* regular *iff it is very simple, and has limited fan-in. An infinitary GSOS systems is* regular *iff every operation in its signature is.*

Most of the standard operations used in the literature on process algebras are regular. An example of a very simple operation which is *not* regular is presented below.

Example: Let $\mathsf{Act} = \{a_i \mid i \geq 1\}$ be a denumerable set of actions. Consider the very simple GSOS operation g with rules (one such rule for each $i \in \omega$):

$$\frac{\left\{x \xrightarrow{a_j} y_j \mid 1 \leq j \leq i\right\}}{g(x) \xrightarrow{a_i} \mathbf{0}} \tag{4}$$

Then g is not regular as it has a rule with n positive antecedents for its one argument for each $n \in \omega$.

The following result is an immediate corollary of the previous theory.

Corollary 3.9 *Let* $G = (\Sigma_G, R_G)$ *be a regular GSOS system. Then, for all* $P \in \mathrm{T}(\Sigma_G)$, $\mathsf{graph}(P)$ *is a finite process graph.*

4 The Problem

The main problem addressed in [2] was to find a complete axiomatization of bisimulation on closed terms – that is, equality in $\mathsf{Bisim}(G)$ – for an arbitrary GSOS system specification G. In that reference it was shown how to find a finite (conditional) equational theory T such that for all closed terms $P, Q \in \mathrm{T}(\Sigma_G)$,

$$T \vdash P = Q \quad \Leftrightarrow \quad \mathsf{Bisim}(G) \models P = Q.$$

The theory T generated by the methods in [2] was purely equational, apart from the presence of one infinitary conditional equation which is a reformulation of the *Approximation Induction Principle* (AIP) familiar from the literature on ACP [6]. Indeed, by recursion theoretic considerations spelled out in [2], it is not possible to do without an infinitary proof rule like AIP for general GSOS specifications.

However, for classes of GSOS systems generating regular behaviours, it should be possible to obtain complete axiomatizations of bisimulation on closed terms that do not rely on infinitary proof rules. In the remainder of this paper, I shall present a way of obtaining such complete axiomatizations of bisimulation for the regular GSOS specifications introduced in Definition 3.8. The presentation will follow [2] quite closely, and the reader will be referred to that paper for some of the details, and examples of applications of the theory.

In [2], it was shown how to reduce the completeness problem for arbitrary GSOS specifications to that for FINTREE, a simple fragment of CCS suitable for expressing finite trees, which was solved by Hennessy and Milner in [24]. Here I shall follow the same approach, by showing how to reduce the completeness problem for *regular* GSOS specifications to that for finite process graphs, which was solved by Milner in [29]. (See also [8, 31]).

The infinitary GSOS system FINTREE has a constant symbol **0** denoting the null process; unary operation symbols $a(\cdot)$, one for each action in Act, denoting action prefixing; and a binary symbol $+$ for nondeterministic choice. The null process is incapable of taking any action, and consequently has no rules. For each action a there is a rule $ax \xrightarrow{a} x$. The operational semantics of $P + Q$ is defined by the rules (one pair of rules for each $a \in \mathsf{Act}$):

$$\frac{x \xrightarrow{a} x'}{x + y \xrightarrow{a} x'} \qquad \frac{y \xrightarrow{a} y'}{x + y \xrightarrow{a} y'} \tag{5}$$

The set of *guarded* FINTREE terms is the least subset of $\mathbb{T}(\Sigma_{\mathsf{FINTREE}})$ such that:

- **0** is a guarded FINTREE term;
- $a.P$ is a guarded FINTREE term for all $P \in \mathbb{T}(\Sigma_{\mathsf{FINTREE}})$;
- $P + Q$ is a guarded FINTREE term if P and Q are.

Intuitively, a FINTREE term P is guarded if every occurrence of a variable in it is within the scope of an action prefixing operation.

A *recursive specification* E is a set of equations $\{x = P_x \mid x \in V_E\}$ with V_E a finite set of variables and every P_x a guarded FINTREE term with variables in V_E.

The infinitary GSOS system RCCS is obtained by disjointly extending FINTREE with constants of the form $\langle x \mid E \rangle$ for each recursive specification E and variable $x \in V_E$. The variables in V_E are bound in $\langle x \mid E \rangle$. Intuitively, $\langle x \mid E \rangle$ denotes the x-component of a solution of E. Let $E = \{x = P_x \mid x \in V_E\}$ be a recursive specification, and P be an RCCS term. Then, following the standard ACP practice (see, *e.g.*, [6, 18]), $\langle P \mid E\rangle$ denotes the term obtained by replacing each free occurrence of $x \in V_E$ in P by $\langle x \mid E\rangle$, *i.e.*, $\langle P \mid E\rangle$ stands for $P\{\langle x \mid E\rangle/x \mid x \in V_E\}$. The operational semantics of each constant $\langle x \mid E\rangle$ is given by the (finite) set of axioms, one such axiom for each $(a, Q) \in \mathsf{init}(P_x)$:

$$\langle x \mid E\rangle \xrightarrow{a} \langle Q \mid E\rangle$$

where, for every guarded $\mathsf{FINTREE}$ term P, $\mathsf{init}(P)$ is given by structural recursion on P as:

- $\mathsf{init}(\mathbf{0}) = \emptyset$;
- $\mathsf{init}(a.Q) = \{(a, Q)\}$;
- $\mathsf{init}(Q + R) = \mathsf{init}(Q) \cup \mathsf{init}(R)$.

With the above definitions, it is easy to see that, for all $Q \in \mathrm{T}(\Sigma_{\mathsf{RCCS}})$ and $a \in \mathsf{Act}$,

$$\langle x \mid E\rangle \xrightarrow{a}_{\mathsf{RCCS}} Q \Leftrightarrow \langle P_x \mid E\rangle \xrightarrow{a}_{\mathsf{RCCS}} Q$$

The following fact is well-known (see, *e.g.*, [29, 8]):

Lemma 4.1 *For every $P \in \mathrm{T}(\Sigma_{\mathsf{RCCS}})$, $\mathsf{graph}(P)$ is finite.*

Bisimulation equivalence over regular processes has been completely axiomatized by Milner [29], and Bergstra and Klop [8] without the use of infinitary proof rules like AIP. To pave the way to the extension of these results to regular GSOS systems, I shall now show that the axioms in Figure 1 are complete for equality in $\mathsf{Bisim}(\mathsf{RCCS})$ by adapting Milner's proof to the language RCCS. In view of axioms (S1)–(S2) in Figure 1, for $I = \{i_1, \ldots, i_n\}$ a finite index set, I write $\sum_{i\in I} P_i$ for $P_{i_1} + \cdots + P_{i_n}$. By convention, $\sum_{i\in\emptyset} P_i$ stands for $\mathbf{0}$.

$$\begin{array}{rcll}
x + y & = & y + x & \text{(S1)}\\
(x + y) + z & = & x + (y + z) & \text{(S2)}\\
x + x & = & x & \text{(S3)}\\
x + \mathbf{0} & = & x & \text{(S4)}\\[1ex]
\langle x \mid E\rangle & = & \langle P_x \mid E\rangle & \text{(Rec)}
\end{array}$$

$$\frac{E}{x \;=\; \langle x \mid E\rangle}\; E \text{ guarded} \quad \text{(RSP)}$$

Figure 1: The theory T_{RCCS}

The following notion from [29, 31] will be important in the remainder of the paper.

Definition 4.2 *Let G be an infinitary GSOS system that disjointly extends RCCS, and T be an collection of Σ_G-equations. A term $P \in \mathrm{T}(\Sigma_G)$ T-provably satisfies a recursive specification $E = \{x = P_x \mid x \in V_E\}$ in the variable $x_0 \in V_E$ iff there are terms Q_x for $x \in V_E$ with $P \equiv Q_{x_0}$, such that for all $x \in V_E$,*

$$T \vdash \; Q_x = P_x\{Q_y/y \mid y \in V_E\}$$

Proposition 4.3 *The following statements hold:*

1. $\mathsf{BISIM}(\mathsf{RCCS}) \models T_{\mathsf{RCCS}}$;

2. T_{RCCS} *is complete for equality in* $\mathsf{Bisim}(\mathsf{RCCS})$, i.e., *for all* $P, Q \in \mathrm{T}(\Sigma_{\mathsf{RCCS}})$,

$$\mathsf{Bisim}(\mathsf{RCCS}) \models P = Q \Rightarrow T_{\mathsf{RCCS}} \vdash P = Q$$

Proof: (*Sketch.*) The soundness of T_{RCCS} with respect to equality in $\mathsf{Bisim}(G)$, for G a disjoint extension of RCCS, can be shown by adapting the well-known soundness proofs of the axioms with respect to $\mathsf{Bisim}(\mathsf{RCCS})$. (See, *e.g.*, [29, Proposition 4.4]). Here I shall concentrate on sketching the strategy of the proof of completeness. This can be delivered in three steps:

- *Step 1*: For each $P \in \mathrm{T}(\Sigma_{\mathsf{RCCS}})$, it is possible to prove a strong head normalization property for T_{RCCS}, namely

$$T_{\mathsf{RCCS}} \vdash \ P = \sum \left\{ a.Q \mid P \stackrel{a}{\rightarrow}_{\mathsf{RCCS}} Q \right\}$$

  This statement can be easily shown by induction on the number of constants of the form $\langle x \mid E \rangle$ which do not occur within the scope of a prefixing operation in P. (In fact the proof only uses axioms (S1)–(S4) and (Rec)).

- *Step 2*: Following Milner [29, 31], one shows that if $P \leftrightarrows_{\mathsf{RCCS}} Q$ then P and Q T_{RCCS}-provably satisfy a common recursive specification $E = \{x = P_x \mid x \in V_E\}$ in some variable $x_0 \in V_E$.

- *Step 3*: Finally, using (RSP), it is possible to show that if P and Q T_{RCCS}-provably satisfy a common recursive specification E in the variable $x_0 \in V_E$, then $T_{\mathsf{RCCS}} \vdash P = Q$.

□

In the reminder of this paper, I shall mimic the strategy used in the above proof to derive complete inference systems for regular GSOS specifications that do not rely on an infinitary conditional equation like the AIP. The inference systems derived using the methods presented in the remainder of this paper will be equational, apart from the conditional equation (RSP). The equational part of the proof system will allow me to prove an analogue of the strong head normalization result stated in step 1 of the proof of the previous proposition. This will require a variety of methods that will be presented in the following section.

To conclude this section, I now present a result stating that, not surprisingly, one can safely extend RCCS with regular operations while preserving the property that the semantics of every closed term is given by a finite process graph.

Proposition 4.4 *Let G be an infinitary GSOS system obtained by adding a disjoint copy of* RCCS *to a regular GSOS system G'. Then, for every $P \in \mathrm{T}(\Sigma_G)$,* $\mathsf{graph}(P)$ *is finite.*

Proof: Let G be an infinitary GSOS system obtained by adding a disjoint copy of RCCS to a regular GSOS system G'. By construction, it follows that G is uniformly bounded. Lemma 3.3 then gives that $\mathsf{graph}(P)$ is finitely branching for all $P \in \mathrm{T}(\Sigma_G)$.

To prove the claim, it is thus sufficient to show that $\mathsf{der}(P)$ is finite for all $P \in \mathrm{T}(\Sigma_G)$. This can be shown by structural induction on P, and a case analysis on the form P takes. All the cases are straightforward using Lemma 4.1 and induction, apart from the case when $P \equiv f(P_1, \ldots, P_l)$ with f a regular operation in $\Sigma_G - \Sigma_{\mathsf{RCCS}}$. To handle this case, following [1], I prove first that

$$\mathsf{der}(f(P_1, \ldots, P_l)) \quad \subseteq \quad \mathsf{reach}(f(P_1, \ldots, P_l)) \tag{6}$$

where $\mathsf{reach}(f(P_1,\ldots,P_l))$ denotes the set

$$\left(\bigcup_{i=1}^{l} \mathsf{der}(P_i)\right) \cup \{g(R_1,\ldots,R_n) \mid f \prec_{G'} g \text{ and } \forall 1 \le j \le n \; \exists 1 \le i \le l : \; R_j \in \mathsf{der}(P_i)\}$$

The claim then follows by (6), as the P_is have finite derivatives by the inductive hypothesis, and $\{g \in \Sigma_{G'} \mid f \prec_{G'} g\}$ is finite because f is regular. □

5 Axiomatizing Regular GSOS Operations

As mentioned in the previous section, the core of the derivation of complete inference systems for regular GSOS systems will be the generation of a set of equations which allow one to prove an analogue of the strong head normalization result stated in the proof of Proposition 4.3, *viz.*, that each closed term is provably equal to the sum of its initial derivatives. Following [2], I shall first show how to axiomatize a class of well-behaved regular GSOS operations, the smooth operations of [2]. Secondly, I shall extend these results to arbitrary regular GSOS operations.

5.1 The Axiomatization of Regular Smooth Operations

The following definition is from [2], where motivation and examples of smooth operations can be found.

Definition 5.1 *A GSOS rule is* smooth *if it takes the form*

$$\frac{\left\{x_i \xrightarrow{a_i} y_i | i \in I\right\} \cup \left\{x_i \stackrel{b_{ij}}{\not\rightarrow} | i \in K, 1 \le j \le n_i\right\}}{f(x_1,\ldots,x_l) \xrightarrow{c} C[\vec{x},\vec{y}]} \tag{7}$$

where I, K are disjoint sets such that $I \cup K = \{1,\ldots,l\}$, and no x_i with $i \in I$ appears in $C[\vec{x},\vec{y}]$. An operation from an infinitary GSOS system G is smooth *if all the rules for this operation are smooth. G is* smooth *if it contains smooth rules only.*

In order to obtain a strongly head normalizing equational theory for regular, smooth operations, I shall first show how to obtain equations that describe the interplay between such operations and the $\mathsf{FINTREE}$ combinators. Examples of the laws that can be derived using the results below may be found in [2]. Lemmas 5.2–5.6 hold for arbitrary infinitary GSOS systems, and will be stated in full generality even though, in the remainder of the paper, I shall only apply them to obtain equations for regular operations.

The following lemma describes how smooth operations interact with the summation operation of $\mathsf{FINTREE}$, and is a slightly sharpened version of [2, Lemma 4.3].

Lemma 5.2 (Distributivity Laws) *Let f be an l-ary smooth operation of an infinitary GSOS system G that disjointly extends* $\mathsf{FINTREE}$*, and suppose that i is an argument of f for which* each *rule for f has a positive antecedent. Then:*

1. *for every G' that disjointly extends G and every $\Sigma_{G'}$-substitution σ,*

$$f(x_1,\ldots,x_i+y_i,\ldots,x_l)\sigma \xrightarrow{a}_{G'} Q \Leftrightarrow (f(x_1,\ldots,x_i,\ldots,x_l)+f(x_1,\ldots,y_i,\ldots,x_l))\sigma \xrightarrow{a}_{G'} Q \tag{8}$$

for all $a \in \mathsf{Act}$ and $Q \in \mathrm{T}(\Sigma_{G'})$;

2. *f distributes over + in its i-th argument,* i.e.,

$$\mathsf{BISIM}(G) \models f(x_1, \ldots, x_i + y_i, \ldots, x_l) = f(x_1, \ldots, x_i, \ldots, x_l) + f(x_1, \ldots, y_i, \ldots, x_l) \quad (9)$$

Proof: It is sufficient to prove (8), as (9) follows immediately from it. To this end, let G' be a disjoint extension of G, and let $P_1, \ldots, P_l$ and Q_i, $1 \le i \le l$, be closed terms over $\Sigma_{G'}$. Suppose that $f(P_1, \ldots, P_i + Q_i, \ldots, P_l) \xrightarrow{a}_{G'} Q$. Then there exist a rule ρ for f of the form (7) and a closed $\Sigma_{G'}$-substitution σ such that

- $\sigma(x_h) = P_h$, for all $1 \le h \le l$ such that $h \ne i$, and $\sigma(x_i) = P_i + Q_i$,
- $Q \equiv C[\vec{x}, \vec{y}]\sigma$, and
- $\sigma(x_h) \xrightarrow{a_h}_{G'} \sigma(y_h)$, for all $h \in I$, and $\sigma(x_h) \stackrel{b_{hk}}{\nrightarrow}$, for all $h \in K$ and $1 \le k \le n_h$.

As G' disjointly extends G, and f is an operation of G, it follows that ρ is a rule of G. By the hypotheses of the lemma, I have that $i \in I$. Therefore, $P_i + Q_i \xrightarrow{a_i}_{G'} \sigma(y_i)$; that is either $P_i \xrightarrow{a_i}_{G'} \sigma(y_i)$ or $Q_i \xrightarrow{a_i}_{G'} \sigma(y_i)$. Assume, without loss of generality, that the former holds. Consider the substitution τ which maps x_i to P_i, and agrees with σ on all the other variables. Then, ρ and τ can be used to infer that

$$f(P_1, \ldots, P_i, \ldots, P_l) \xrightarrow{a}_{G'} C[\vec{x}, \vec{y}]\tau$$

Moreover, as f is smooth, x_i does not occur in $C[\vec{x}, \vec{y}]$. It follows that $C[\vec{x}, \vec{y}]\tau \equiv Q$, as $\tau = \sigma$ over the set of variables occurring in $C[\vec{x}, \vec{y}]$. Hence, I have that

$$f(P_1, \ldots, P_i, \ldots, P_l) + f(P_1, \ldots, Q_i, \ldots, P_l) \xrightarrow{a}_{G'} Q$$

The converse implication can be shown by a symmetric argument, using the fact that, as f is smooth and its i-th argument is tested positively by every rule for it, argument i is not tested negatively by any rule ρ for f. □

The following lemma, that extends [2, Lemma 4.6] to infinitary GSOS systems, gives *inaction laws* to describe the interaction between arbitrary operations and the FINTREE constant **0**; that is, laws which say when a term $f(\vec{P})$ is bisimilar to **0**.

Lemma 5.3 (Inaction Laws) *Suppose f is an l-ary smooth operation of an infinitary GSOS system G that disjointly extends* FINTREE*, and suppose that, for $1 \le i \le l$, term P_i is of the form* **0***, x_i, ax_i or $ax_i + y_i$. Suppose further that for each rule for f of the form (7) there is an index i such that either (1) $i \in I$ and $P_i \equiv$* **0** *or $P_i \equiv ax_i$ for some $a \ne a_i$, or (2) $i \in K$ and $P_i \equiv b_{ij}x_i + y_i$ for some $1 \le j \le n_i$. Then:*

1. *for every G' that disjointly extends G and every $\Sigma_{G'}$-substitution σ,*

$$f(\vec{P})\sigma \xrightarrow{a} Q \quad \textit{for no } a \in \mathsf{Act} \textit{ and } Q \in \mathrm{T}(\Sigma_{G'}) \quad (10)$$

2. *for every G' that disjointly extends G and every $\Sigma_{G'}$-substitution σ, $f(\vec{P})$ is bisimilar to* **0**, i.e.,

$$\mathsf{BISIM}(G) \models f(\vec{P}) = \mathbf{0} \quad (11)$$

To complete the series of results giving equations dealing with the interplay between smooth operations and the FINTREE combinators, I shall now present results corresponding to the action laws of [2]. In particular, I shall follow the "alternative" approach given in [2, Section 6] in which the interplay between prefixing and smooth operations is described by means of so called "peeling" and "action laws".

Definition 5.4 ([2]) *A smooth operation f from an infinitary GSOS system G is* distinctive *if, for each argument i, either all rules for f test i positively or none of them does, and moreover for each pair of different rules for f there is an argument for which both rules have a positive antecedent, but with a different action.*

The following lemma gives the so-called peeling laws. These are laws that can be used to reduce the arguments that are tested negatively by a smooth and distinctive operation to a form in which either action laws or inaction laws can be applied.

Lemma 5.5 (Peeling Laws) *Suppose f is a distinctive smooth operation of a disjoint extension G of* FINTREE, *with a rule ρ of the form*

$$\frac{\left\{x_i \xrightarrow{a_i} y_i | i \in I\right\} \cup \left\{x_i \stackrel{b_{ij}}{\not\rightarrow} | i \in K, 1 \leq j \leq n_i\right\}}{f(\vec{x}) \xrightarrow{c} C[\vec{x}, \vec{y}]}$$

Let $k \in K$ be such that x_k does not occur in $C[\vec{x}, \vec{y}]$, and $b \notin \{b_{kj} | 1 \leq j \leq n_k\}$. Take

$$P_i \equiv \begin{cases} a_i y_i & i \in I \\ b x'_k + x''_k & i = k \\ x_i & i \in K \wedge i \neq k \end{cases} \qquad \textit{and} \qquad Q_i \equiv \begin{cases} a_i y_i & i \in I \\ x''_k & i = k \\ x_i & i \in K \wedge i \neq k \end{cases}$$

Then:

1. *for every G' that disjointly extends G and every $\Sigma_{G'}$-substitution σ,*

$$f(\vec{P})\sigma \xrightarrow{a} S \Leftrightarrow f(\vec{Q})\sigma \xrightarrow{a} S \tag{12}$$

   for all $a \in$ Act and $S \in \mathrm{T}(\Sigma_{G'})$;

2. *the equality $f(\vec{P}) = f(\vec{Q})$ is valid in every G' that disjointly extends G,* i.e.,

$$\mathsf{BISIM}(G) \models f(\vec{P}) = f(\vec{Q}) \tag{13}$$

Proof: It is sufficient to prove the first statement as the second is an immediate corollary of it. Let G' be a disjoint extension of G. Now note that, for any closed $\Sigma_{G'}$-substitution σ, rule ρ fires from $f(\vec{P})\sigma$ iff it fires from $f(\vec{Q})\sigma$. By the distinctiveness of f, ρ is the only rule that can possibly fire from these terms. Moreover, as x_k does not occur in $C[\vec{x}, \vec{y}]$, it is easy to check that if ρ fires, then the targets of the matching transitions from $f(\vec{P})\sigma$ and $f(\vec{Q})\sigma$ are syntactically equal. $\Box$

Lemma 5.6 (Action Laws) *Suppose f is a distinctive smooth operation of a disjoint extension G of* FINTREE, *with a rule ρ of the form*

$$\frac{\left\{x_i \xrightarrow{a_i} y_i | i \in I\right\} \cup \left\{x_i \stackrel{b_{ij}}{\not\rightarrow} | i \in K, 1 \leq j \leq n_i\right\}}{f(\vec{x}) \xrightarrow{c} C[\vec{x}, \vec{y}]}$$

Let

$$P_i \equiv \begin{cases} a_i y_i & i \in I \\ \mathbf{0} & i \in K \wedge n_i > 0 \\ x_i & \textit{otherwise} \end{cases}$$

Then:

1. *for every G' that disjointly extends G and every $\Sigma_{G'}$-substitution σ,*

$$f(\vec{P})\sigma \xrightarrow{a}_{G'} Q \Leftrightarrow c.\Big(C[\vec{P},\vec{y}]\sigma\Big) \xrightarrow{a}_{G'} Q \tag{14}$$

for all $a \in \mathsf{Act}$ and $Q \in \mathrm{T}(\Sigma_{G'})$;

2. *the equality $f(\vec{P}) = c.C[\vec{P},\vec{y}]$ is valid in every G' that disjointly extends G,* i.e.,

$$\mathsf{BISIM}(G) \models f(\vec{P}) = c.C[\vec{P},\vec{y}] \tag{15}$$

Proof: I only prove statement (14). Let G' be a disjoint extension of G, and σ be a $\Sigma_{G'}$-substitution. Assume that ρ is the rule for f given by the proviso of the lemma, and that $f(\vec{P})\sigma \xrightarrow{a}_{G'} Q$. Then there exist a rule $\rho' \in R_{G'}$ for f and a $\Sigma_{G'}$-substitution τ such that:

- $\tau(x_i) = a_i.\sigma(y_i)$ for all $i \in I$, $\tau(x_i) = \mathbf{0}$ for all $i \in K$ with $n_i > 0$, and $\tau(x_i) = \sigma(x_i)$ for all $i \in K$ with $n_i = 0$;
- τ satisfies the antecedents of ρ'; and
- Q can be obtained by applying the substitution τ to the target of ρ'.

As $G \sqsubseteq G'$ and f is distinctive in G, it follows that $\rho = \rho'$. Thus I have that $a = c$. Moreover, as τ satisfies the antecedents of ρ, $\tau(x_i) = a_i.\sigma(y_i) \xrightarrow{a_i}_{G'} \tau(y_i) = \sigma(y_i)$ for all $i \in I$. Note now that, as f is smooth, each variable occurring in the context $C[\vec{x},\vec{y}]$ is contained in the set $\{y_i \mid i \in I\} \cup \{x_i \mid i \in K\}$. We have already seen that σ and τ agree over $\{y_i \mid i \in I\} \cup \{x_i \mid i \in K \text{ and } n_i = 0\}$. We also have that $\tau(x_i) = \mathbf{0} = P_i\sigma$ for all $i \in K$ with $n_i > 0$. It follows that $Q \equiv C[\vec{x},\vec{y}]\tau \equiv C[\vec{P},\vec{y}]\sigma$. This implies that $c.\Big(C[\vec{P},\vec{y}]\sigma\Big) \xrightarrow{a}_{G'} Q$.

The converse implication can be shown in similar fashion. □

The combination of peeling laws (13), instantiated action laws (15), distributivity laws (9), and inaction laws (11), gives a theory that is head normalizing for terms built from distinctive smooth operations that are *discarding* [2, Definition 6.2].

Definition 5.7 (Discarding and Good Operations) *A smooth GSOS rule of the form (7) is* discarding *if for no argument i that is tested negatively, x_i occurs in the target. A smooth operation is* discarding *if all the rules for this operation are.*
A smooth operation is good[3] *if it is both discarding and distinctive.*

Theorem 5.8 *Suppose G is an infinitary GSOS system with $\mathsf{RCCS} \sqsubseteq G$. Let $\Sigma \subseteq \Sigma_G - \Sigma_{\mathsf{RCCS}}$ be a collection of good operations of G. Let T be the equational theory that extends $T_{\mathsf{FINTREE}} \cup \{(\mathrm{REC})\}$ with the following axioms, for each operation f from Σ:*

1. *for each argument i of f that is tested positively, a distributivity axiom (9),*
2. *for each rule for f of the form (7), for each argument i that is tested negatively, and for each action $b \notin \{b_{ij} | 1 \leq j \leq n_i\}$, a peeling law (13),*
3. *for each rule for f, an action law (15),*
4. *all the inaction laws (11) for f.*

[3]This terminology has been introduced by Bosscher in [15].

Then $\mathsf{BISIM}(G) \models T$*, and for each* $P \in \mathrm{T}(\Sigma \cup \Sigma_{\mathsf{RCCS}})$

$$T \vdash P = \sum \left\{ a.Q \mid P \xrightarrow{a}_G Q \right\}$$

Proof: The fact that $\mathsf{BISIM}(G) \models T$ follows immediately from the previous lemmas. I now show that for each $P \in \mathrm{T}(\Sigma \cup \Sigma_{\mathsf{RCCS}})$

$$T \vdash P = \sum \left\{ a.Q \mid P \xrightarrow{a}_G Q \right\}$$

The proof will be by structural induction on P. I proceed by a case analysis on the possible forms P can take. The cases $P \equiv \mathbf{0}$ and $P \equiv a.Q$ are trivial, using the fact that G disjointly extends RCCS.

Case $P \equiv \langle x \mid E \rangle$. First of all, note that for every guarded FINTREE term P and recursive specification E the following holds:

$$(\mathrm{S1})\text{–}(\mathrm{S4}) \vdash \langle P \mid E \rangle = \sum \{ a.\langle Q \mid E \rangle \mid (a, Q) \in \mathsf{init}(P) \}$$

The claim then follows by axiom (Rec) and the fact that G disjointly extends RCCS.

Case $P \equiv Q + R$. Immediate by applying the inductive hypothesis to Q and R.

Case $P \equiv f(P_1, \ldots, P_l)$ *for some* $f \in \Sigma$. By induction, $T \vdash P_i = \sum \left\{ a.Q \mid P_i \xrightarrow{a}_G Q \right\}$ for each $1 \leq i \leq l$. I shall now prove that $T \vdash P = \sum \left\{ a.Q \mid P \xrightarrow{a}_G Q \right\}$ by a further induction on the combined sizes of the P_is. There are three main cases to examine.

Case 1. There is an argument i that is tested positively by f and for which P_i is of the form $P_i' + P_i''$. As f is distinctive, all rules for f test i positively. In this case we can apply one of the distributivity laws (9) to infer

$$T \vdash f(P_1, \ldots, P_i' + P_i'', \ldots, P_l) = f(P_1, \ldots, P_i', \ldots, P_l) + f(P_1, \ldots, P_i'', \ldots, P_l)$$

The sub-inductive hypothesis now gives that

$$\begin{aligned} T \vdash f(P_1, \ldots, P_i', \ldots, P_l) &= \sum \left\{ a.Q \mid f(P_1, \ldots, P_i', \ldots, P_l) \xrightarrow{a}_G Q \right\} \\ T \vdash f(P_1, \ldots, P_i'', \ldots, P_l) &= \sum \left\{ a.Q \mid f(P_1, \ldots, P_i'', \ldots, P_l) \xrightarrow{a}_G Q \right\} \end{aligned}$$

Thus, by (8), it follows that

$$T \vdash f(P_1, \ldots, P_i' + P_i'', \ldots, P_l) = \sum \left\{ a.Q \mid f(P_1, \ldots, P_i' + P_i'', \ldots, P_l) \xrightarrow{a}_G Q \right\}$$

Case 2. There is an argument i that is tested positively by f and for which $P_i \equiv \mathbf{0}$. Since f is distinctive, all rules for f test i positively. Thus T contains an inaction law $f(x_1, \ldots, x_{i-1}, \mathbf{0}, x_{i+1}, \ldots, x_l) = \mathbf{0}$. Instantiation of this law gives $T \vdash f(\vec{P}) = \mathbf{0}$, and the induction step follows.

Case 3. For all arguments k that are tested positively by f, P_k is of the form $a_k P_k'$. There are two subcases to consider.

Case 3.1. For each rule for f with positive trigger $\langle e_1, \ldots, e_l \rangle$, there is an i that is tested positively such that $e_i \neq a_i$. Then T contains an inaction law $f(\vec{Q}) = \mathbf{0}$, where $Q_k \equiv a_k x_k$ if k is tested positively, and $Q_k \equiv x_k$ otherwise. Instantiation of this law gives $T \vdash f(\vec{P}) = \mathbf{0}$, and the induction step follows.

Case 3.2. There exists a rule ρ for f with positive trigger $\langle e_1, \ldots, e_l \rangle$ such that $e_k = a_k$ for all k that are tested positively. Since f is distinctive, ρ is in fact the unique rule with this property. Again there are two subcases.

Case 3.2.1. There is an index j that is not tested positively, and there is an action b such that $x_j \stackrel{b}{\not\to}$ is an antecedent of ρ, and T proves an equation of the form $P_j = bP'_j + P''_j$. Now we note that T contains an inaction law $f(\vec{Q}) = \mathbf{0}$, where $Q_k \equiv a_k x_k$ if k is tested positively, $Q_k \equiv bx_k + y_k$ if $k = j$, and $Q_k \equiv x_k$ otherwise. Application of this law gives $T \vdash f(P_1, \ldots, bP'_j + P''_j, \ldots, P_l) = \mathbf{0}$, and the induction step follows.

Case 3.2.2. For each index n that is not tested positively, P_n is of the form $\sum a_{nj} P_{nj}$ and, for no j, $x_n \stackrel{a_{nj}}{\not\to}$ is an antecedent of ρ. Again, I distinguish two subcases (the last ones).

Case 3.2.2.1. There exists an argument n that is tested negatively by ρ and an action b for which T proves an equation of the form $P_n = b.P'_n + P''_n$. Then, an instance of axiom (13) for ρ, n and b gives $T \vdash f(P_1, \ldots, b.P'_n + P''_n, \ldots, P_l) = f(P_1, \ldots, P''_n, \ldots, P_l)$, where, by (12), the two terms have the same derivatives. The claim now follows by applying the inductive hypothesis to the term $f(P_1, \ldots, P''_n, \ldots, P_l)$.

Case 3.2.2.2. For every argument n that is tested negatively by ρ, $P_n \equiv \mathbf{0}$. Then, an application of the action law (15) for ρ gives the required head normal form.

This completes the subinductive argument and the proof for the case $P \equiv f(P_1, \ldots, P_l)$ with $f \in \Sigma$.

The proof of the theorem is then complete.

□

The next proposition shows how to handle general smooth and discarding operations by expressing them as a *finite* sum of good operations, in very much the same way as the merge operation of ACP is expressed as a sum of the auxiliary operations of left-merge and communication merge (see, *e.g.*, [6] for a textbook presentation). In particular, the following proposition allows for the "discovery" of the auxiliary operations of ACP. See [2] for details and examples.

Proposition 5.9 *Let G be an infinitary GSOS system obtained by adding a disjoint copy of* RCCS *to a regular GSOS system G'. Assume that f is an l-ary smooth and discarding operation of G'. Then there exists a disjoint extension G^* of G' with l-ary good, regular operations $f_1, \ldots, f_n$ not occurring in* RCCS *such that:*

1. *G^* is a regular GSOS system;*

2. *for every disjoint extension G'' of $G^* \oplus$* RCCS, *and every $\Sigma_{G''}$-substitution σ,*

$$f(\vec{x})\sigma \xrightarrow{a}_{G''} Q \Leftrightarrow (f_1(\vec{x}) + \cdots + f_n(\vec{x}))\sigma \xrightarrow{a}_{G''} Q \qquad (16)$$

for all $a \in$ Act *and $Q \in \mathrm{T}(\Sigma_{G''})$;*

3. for all $\vec{x}$ of length l,

$$\mathsf{BISIM}(G^* \oplus \mathsf{RCCS}) \models f(\vec{x}) = f_1(\vec{x}) + \cdots + f_n(\vec{x}) \tag{17}$$

Proof: Assume that f is an l-ary smooth and discarding operation of G'. I shall show how to partition the set R of rules for f in R_G into sets $R_1, \ldots, R_n$ in such a way that that, for all $1 \leq i \leq n$, f is distinctive in the infinitary GSOS system obtained from G by removing all the rules in $R - R_i$. First of all, partition the set of rules for f into sets $R_1, \ldots, R_m$, where, for each $1 \leq j \leq m$ and $\rho, \rho' \in R$, $\rho, \rho' \in R_j$ iff they test the same arguments positively. Note that, even if R were denumerable, $m \leq 2^l$. I now show how to further partition each R_j for which f restricted to R_j is not distinctive.

As G' is regular, f is uniformly bounded. Thus there exists a maximum $n_{(f,j)}$ to the number of distinct rules in R_j which have the same positive trigger. It is then possible to further partition R_j into sets $R_{j1}, \ldots, R_{jn_{(f,j)}}$ in such a way that no two distinct rules in R_j with the same positive trigger are in the same set. The restriction of f to the rules in each R_{jk} trivially yields a good operation.

Define Σ_{G^*} to be the signature obtained by extending $\Sigma_{G'}$ with fresh l-ary operation symbols $f_{11}, \ldots, f_{1n_{(f,1)}}, \ldots, f_{m1}, \ldots, f_{mn_{(f,m)}}$. Next define R_{G^*} to be the set of rules obtained by extending $R_{G'}$, for each (i,j), with rules derived from the rules of R_{ij} by replacing the operation symbol in the source by f_{ij}. It is immediate to see that each operation f_{ij} so defined is good. Moreover, the resulting infinitary GSOS system G^* is regular as G' was. It is routine to check that for all disjoint extensions G'' of $G^* \oplus \mathsf{RCCS}$, for all $P_1, \ldots, P_l, Q \in \mathrm{T}(\Sigma_{G''})$, and $a \in \mathsf{Act}$

$$f(P_1, \ldots, P_l) \xrightarrow{a}_{G''} Q \quad \Leftrightarrow \quad \sum_{i=1}^{m} \sum_{j=1}^{n_{(f,i)}} f_{ij}(P_1, \ldots, P_l) \xrightarrow{a}_{G''} Q$$

from which $\mathsf{BISIM}(G^* \oplus \mathsf{RCCS}) \models f(\vec{x}) = \sum_{i=1}^{m} \sum_{j=1}^{n_{(f,i)}} f_{ij}(\vec{x})$ follows immediately. □

It is interesting to note that the above proposition would *not* hold if regular operations were allowed to be bounded, but not uniformly so. As an example, consider the operation f on page 7 given by the rules (3). This operation is bounded, but not uniformly bounded. I shall now show that, under mild assumptions, it is impossible to express f as a finite sum of unary smooth, distinctive operations.

Proposition 5.10 *Let* $\mathsf{Act} = \{a_i \mid i \geq 1\}$ *be a denumerable set of actions, and G be an infinitary GSOS system which disjointly extends* $\mathsf{FINTREE}$ *comprising a smooth, discarding unary operation f, with rules (one such rule for each $i \in \omega$ and $j \leq i$):*

$$\frac{x \xrightarrow{a_i} y}{f(x) \xrightarrow{a_j} \mathbf{0}}$$

Then there does not exist a disjoint extension G' of G with a family of unary smooth and distinctive operations $f_1, \ldots, f_n$ such that

$$\mathsf{Bisim}(G') \models f(x) = f_1(x) + \cdots + f_n(x)$$

Proof: Assume, towards a contradiction, that there exists a disjoint extension G' of G with a family of unary smooth and distinctive operations $f_1, \ldots, f_n$ such that

$$\mathsf{Bisim}(G') \models f(x) = f_1(x) + \cdots + f_n(x)$$

Take $P \equiv a_{n+1}.\mathbf{0}$. Then $f(P) \xrightarrow{a_l}_{G'} \mathbf{0}$ for each $1 \leq l \leq n+1$. As $f(P) \leftrightarrows_{G'} f_1(P) + \cdots + f_n(P)$, it must be the case that, for some $1 \leq i \leq n$ and $1 \leq j < h \leq n+1$, $f_i(P) \xrightarrow{a_j}_{G'} P_1$ and $f_i(P) \xrightarrow{a_h} P_2$ for some closed terms P_1, P_2 such that $P_1 \leftrightarrows_{G'} \mathbf{0} \leftrightarrows_{G'} P_2$. These two transitions must be provable from two distinct rules

$$\frac{H_1}{f_i(x) \xrightarrow{a_j} Q_1} \qquad \frac{H_2}{f_i(x) \xrightarrow{a_h} Q_2}$$

and two substitutions which map x to P. As f_i is distinctive, these rules test the same arguments positively. There are two cases to examine:

Case 1. Both the above rules have a positive antecedent for x. As both rules fire when x is instantiated to P, these positive antecedents must be of the form $x \xrightarrow{a_{n+1}} y$ for some variable y. It follows that f_i has two rules with the same positive trigger. This contradicts the assumption that f_i is distinctive.

Case 2. Both the rules do not test x positively. Again this contradicts the distinctiveness of f_i because the two rules would then have the same positive trigger.

□

In fact, the argument used in the above proof can be used to show that, *mutatis mutandis*, there does not exist a disjoint extension G' of G with a family of smooth and distinctive operations $f_1, \ldots, f_n$ with arities $l_1, \ldots, l_n$, respectively, such that

$$\mathsf{Bisim}(G') \models f(x) = f_1(\underbrace{x, \ldots, x}_{l_1\text{-times}}) + \cdots + f_n(\underbrace{x, \ldots, x}_{l_n\text{-times}})$$

Here I have preferred to present in detail the notationally simpler case in which all the f_is are themselves unary.

Proposition 5.9 is the only result in this paper which would not hold if I allowed regular operations to be bounded, rather than uniformly bounded.

5.2 General Regular Operations

In this subsection I show how to axiomatize GSOS operations that are not both smooth and discarding. The technical development will parallel the results presented in [2, Section 4.2].

Definition 5.11 *For ρ a GSOS rule of the form (2), I write $\rho\sigma$ for the result of substituting $\sigma(x)$ for each x occurring in ρ.*

The following technical lemma is an extension to infinitary GSOS systems of [2, Lemma 4.12], whose proof can be adapted to this larger class of GSOS specifications.

Lemma 5.12 *Suppose G is an infinitary GSOS system and $P = f(\vec{z})$ and $Q = f'(\vec{v})$ are terms over Σ_G with variables that do not occur in R_G. Suppose that there exists a 1-1 correspondence between rules for f and rules for f' such that, whenever a rule ρ for f with source $f(\vec{x})$ is related to a rule ρ' for f' with source $f'(\vec{y})$, we have that, with exception of their sources, $\rho\{\vec{z}/\vec{x}\}$ and $\rho'\{\vec{v}/\vec{y}\}$ are identical. Then:*

1. *for every disjoint extension G' of G and closed $\Sigma_{G'}$-substitution σ,*

$$P\sigma \xrightarrow{a}_{G'} S \;\Leftrightarrow\; Q\sigma \xrightarrow{a}_{G'} S$$

for all $a \in \mathsf{Act}$ and $S \in \mathrm{T}(\Sigma_{G'})$;

2. $\mathsf{BISIM}(G) \models P = Q$.

Proposition 5.13 *Suppose G is a regular GSOS system containing an operation f with arity l that is not both smooth and discarding. Then there exists a regular disjoint extension G' of G with a smooth and discarding operation f' not occurring in* RCCS *with arity l' (possibly different from l), and there exist vectors $\vec{z}$ of l distinct variables, and $\vec{v}$ of l' variables in $\vec{z}$ (possibly repeated), such that:*

1. *for every disjoint extension G'' of G' and $\Sigma_{G''}$-substitution σ,*

$$f(\vec{z})\sigma \xrightarrow{a}_{G''} Q \;\;\Leftrightarrow\;\; f'(\vec{v})\sigma \xrightarrow{a}_{G''} Q \tag{18}$$

   for all $a \in$ Act and $Q \in \mathrm{T}(\Sigma_{G''})$;

2. *the equation $f(\vec{z}) = f'(\vec{v})$ is valid in any disjoint extension of G',* i.e.,

$$\mathsf{BISIM}(G') \models f(\vec{z}) = f'(\vec{v}) \tag{19}$$

Proof: (Following the proof of [2, Lemma 4.13]). In order to determine the arity of f' I first quantify the degree in which f is non-smooth and non-discarding. For ρ a simple GSOS rule of the form (2), and $1 \leq i \leq l$, the *nastiness factor* of ρ and i is defined as

m_i	if $n_i = 0$ and x_i does not occur in the target
$m_i + 1$	if $n_i > 0$ and x_i does not occur in the target, or $n_i = 0$ and x_i occurs in the target
$m_i + 2$	if $n_i > 0$ and x_i occurs in the target

Note that, as f is a regular operation, the nastiness factor of ρ and i is less than or equal to $m_{(f,i)}+2$ for all ρ, where $m_{(f,i)}$ is the maximum number of positive antecedents for i in the rules for f. The *nastiness factor* of f and i, notation $N(f,i)$, is then defined as the maximum over all rules ρ for f of the nastiness factor of f and i. Let $l' = \sum_{i=1}^{l} N(f,i)$ and let f' be a fresh operation symbol. Then $\Sigma_{G'}$ is defined as the signature that extends Σ_G with an l'-ary operation symbol f'. Let $\vec{w} = w_{11}, \ldots, w_{1N(f,1)}, \ldots, w_{l1}, \ldots, w_{lN(f,l)}$ be a vector of l' different variables. Suppose ρ is a rule for f as in (2) and suppose τ is the substitution that maps each variable w_{ij} to x_i and leaves all the other variables unchanged. Let ρ' be the smooth and discarding GSOS rule with source $f'(\vec{w})$, and such that, with exception of their sources $\rho'\tau$ and ρ are identical. In fact, such a ρ' can be obtained from ρ by replacing the source of ρ by $f'(\vec{w})$, and by replacing variables x_i in the antecedents and the target by variables w_{ij} in such a way that the resulting rule is smooth and discarding. This can be done as follows: starting with the positive antecedents, one replaces each occurrence of x_i with a different variable w_{ij}; after that one more w_{ij} is available for the occurrences of x_i in the negative antecedents, and another one for the occurrences of x_i in the target — in case there are such occurrences. Define $R_{G'}$ to be a set of rules that extends R_G with a rule ρ', defined as above, for each rule ρ for f. It is easy to see that, by construction, the resulting infinitary GSOS system is regular, as G was.

Let now $\vec{z} = z_1, \ldots, z_l$ be a vector of different variables, all of them not occurring in R_G, and let $\vec{v} = v_{11}, \ldots, v_{1N(f,1)}, \ldots, v_{l1}, \ldots, v_{lN(f,l)}$ be the vector of length l' given by $v_{ij} = z_i$. It is easy to see that, for each pair ρ, ρ' of corresponding rules, $\rho\{\vec{z}/\vec{x}\}$ and $\rho'\{\vec{v}/\vec{w}\}$ are identical, with exception of their sources. Thus we can apply Lemma 5.12 to obtain that the claims of the lemma hold. ☐

It is interesting to note that the above proposition would *not* hold for very simple GSOS operations that are not regular. As an example, consider the operation g on page 8 given by the rules (4). This operation is very simple, but not regular. I shall now show that, under mild assumptions, it is impossible to express g as a finite sum of smooth operations.

Proposition 5.14 *Let* $\mathsf{Act} = \{a_i \mid i \geq 1\}$ *be a denumerable set of actions, and* G *be an infinitary GSOS system which disjointly extends* FINTREE *comprising a very simple unary operation* g*, with rules (one such rule for each* $i \in \omega$*):*

$$\frac{\left\{x \xrightarrow{a_j} y_j \mid 1 \leq j \leq i\right\}}{g(x) \xrightarrow{a_i} \mathbf{0}}$$

Then there does not exist a disjoint extension G' *of* G *with a family of smooth operations* $g_1, \ldots, g_n$ *with arities* $l_1, \ldots, l_n$*, respectively, such that*

$$\mathsf{Bisim}(G') \models g(x) = g_1(\underbrace{x, \ldots, x}_{l_1\text{-}times}) + \cdots + g_n(\underbrace{x, \ldots, x}_{l_n\text{-}times})$$

Proof: Assume, towards a contradiction, that such a G' exists. Let l be the maximum of $l_1, \ldots, l_n$ and take $P \equiv \sum_{i=1}^{l+1} a_i.\mathbf{0}$. Then $g(P) \xrightarrow{a_{l+1}}_{G'} \mathbf{0}$. As $g(P) \leftrightarrows_{G'} g_1(P^{l_1}) + \cdots + g_n(P^{l_n})$, there exists $1 \leq j \leq n$ and $Q \in \mathrm{T}(\Sigma_{G'})$ such that $g_j(P^{l_j}) \xrightarrow{a_{l+1}}_{G'} Q \leftrightarrows_{G'} \mathbf{0}$. Let ρ be any rule for g_j that can be used to derive this transition. As g_j is smooth and $l_j < l+1$, there exists an index k with $1 \leq k \leq l+1$ such that, for no argument i of g_j, ρ has a positive antecedent of the form $x_i \xrightarrow{a_k} y_i$.

Consider now the term $R \equiv \sum \{a_i.\mathbf{0} \mid i \in \{1, \ldots, l+1\} - \{k\}\}$. Note that $g(R) \stackrel{a_{l+1}}{\not\rightarrow}_{G'}$ as $R \stackrel{a_k}{\not\rightarrow}_{G'}$. On the other hand, I claim that ρ can be used to show that $g_j(R^{l_j}) \xrightarrow{a_{l+1}}_{G'}$. In fact, all the positive antecedents for ρ are met by setting all the arguments of g_j to R, as they were met by P and none of them refers to a_k. Moreover, it is immediate to see that, for all $b \in \mathsf{Act}$, $P \stackrel{b}{\not\rightarrow}_{G'}$ implies that $R \stackrel{b}{\not\rightarrow}_{G'}$. Hence all the negative antecedents of ρ are also met by R as they were met by P. Therefore, $g_j(R^{l_j}) \xrightarrow{a_{l+1}}_{G'}$. It follows that $g(R)$ is not bisimilar to $g_1(R^{l_1}) + \cdots + g_n(R^{l_n})$. This contradicts the assumption that

$$\mathsf{Bisim}(G') \models g(x) = g_1(\underbrace{x, \ldots, x}_{l_1\text{-times}}) + \cdots + g_n(\underbrace{x, \ldots, x}_{l_n\text{-times}})$$

□

Proposition 5.13 is the only result in this paper that does not hold for very simple GSOS systems which contain operations that do not have limited fan-in.

The theory that has been developed so far gives a strongly head normalizing equational theory for all regular GSOS operations.

Theorem 5.15 *Let* G *be a regular GSOS system. Then the infinitary GSOS system* G' *of the form* $G^* \oplus \mathsf{RCCS}$ *such that* G^* *is a regular GSOS system that disjointly extends* G*, and* $\mathsf{RCCS} \sqsubseteq G'$*, together with the equational theory* T *produced by the algorithm of Figure 2 have the property that* $\mathsf{BISIM}(G') \models T$ *and, for all* $P \in \mathrm{T}(\Sigma_{G'})$*,* $T \vdash P = \sum \left\{a.Q \mid P \xrightarrow{a}_{G'} Q\right\}$*.*

Proof: The fact that $\mathsf{BISIM}(G') \models T$ follows immediately from the previous results of this section. I shall now show that, for all $P \in \mathrm{T}(\Sigma_{G'})$,

$$T \vdash P = \sum \left\{a.Q \mid P \xrightarrow{a}_{G'} Q\right\}$$

First of all, it is easy to prove, by structural induction on P, that for all $P \in \mathrm{T}(\Sigma_{G'})$, there exists a $Q \in \mathrm{T}(\Sigma_{G'})$ built only from good operations in Σ_{G^*} and operations in RCCS such that $T \vdash P = Q$ using instances of laws (19) and (17). Moreover, by (18) and (16), I have that

$$\left\{a.R \mid P \xrightarrow{a}_{G'} R\right\} = \left\{a.R \mid Q \xrightarrow{a}_{G'} R\right\} \tag{20}$$

Input	A regular GSOS system G.
Output	An infinitary GSOS system G' of the form $G^* \oplus \mathsf{RCCS}$ such that G^* is a regular GSOS system that disjointly extends G, and $\mathsf{RCCS} \sqsubseteq G'$, together with an equational theory T, such that $\mathsf{BISIM}(G') \models T$ and T is strongly head normalizing for all terms of G'.

Step 1. Add to G a disjoint copy of RCCS.

Step 2. For each regular operation $f \in \Sigma_G$ that is not both smooth and discarding, apply the construction of Proposition 5.13 to extend the system with a regular smooth and discarding version f', in such a way that law (19) holds. Add all the resulting instances of law (19) to $T_{\mathsf{FINTREE}} \cup (\mathsf{Rec})$.

Step 3. For each smooth, discarding and non-distinctive operation $f \notin \Sigma_{\mathsf{RCCS}}$ in the resulting system, apply the construction of Proposition 5.9 to generate good, regular operations $f_1, \ldots, f_n$ in such a way that law (17) is valid. The system so-obtained is the infinitary GSOS system G' we were looking for. Add to the equational theory all the resulting instances of law (17).

Step 4. Add to the equational theory obtained in Step 3 the equations given by applying Theorem 5.8 to all the good operations in $\Sigma_{G'} - \Sigma_{\mathsf{RCCS}}$. The result is the theory T we were looking for.

Figure 2: The algorithm

Next, an application of Theorem 5.8 gives that

$$T \vdash Q = \sum \left\{ a.R \mid Q \xrightarrow{a}_{G'} R \right\}$$

The claim now follows immediately by transitivity and (20). □

6 Completeness

For any regular GSOS system G, the algorithm presented in Figure 2 allows for the generation of a disjoint GSOS extension G' with a strongly head-normalizing equational theory. The reader might recall that this was the first step in the proof of completeness of T_{RCCS} for $\mathsf{Bisim}(\mathsf{RCCS})$. I shall now show how to mimic the remaining two steps in the proof of Proposition 4.3 to obtain completeness for arbitrary regular GSOS specifications.

The following proposition plays the role of step 2 of the proof of Proposition 4.3 in this setting.

Proposition 6.1 *Suppose that G is a regular GSOS system. Let G' and T denote the disjoint extension of G, and the strongly head normalizing equational theory constructed by the algorithm in Figure 2, respectively. Then, for all $P, Q \in \mathrm{T}(\Sigma_{G'})$ such that $\mathsf{Bisim}(G') \models P = Q$, there exists a recursive specification E T-provably satisfied in the same variable x_0 by both P and Q.*

Proof: Let $P, Q \in \mathrm{T}(\Sigma_{G'})$ be such that $\mathsf{Bisim}(G') \models P = Q$. By Proposition 4.4, it follows that $\mathsf{graph}(P)$ and $\mathsf{graph}(Q)$ are finite. (Recall that G' is obtained by adding a disjoint copy of RCCS to a regular GSOS system G^*). A recursive specification E which is T-provably satisfied in the same variable x_0 by both P and Q can now be constructed as follows. Take a fresh set of variables $V_E = \{x_{RS} \mid R \in \mathsf{der}(P),\ S \in \mathsf{der}(Q) \text{ and } R \leftrightarroweq_{G'} S\}$ with x_{PQ} as leading variable. By the definition of bisimulation, it follows that:

1. for each $R \in \mathsf{der}(P)$, there exists $S \in \mathsf{der}(Q)$ such that $R \leftrightarrow_{G'} S$, and

2. for each $S \in \mathsf{der}(Q)$, there exists $R \in \mathsf{der}(P)$ such that $R \leftrightarrow_{G'} S$.

Next define

$$P_{x_{RS}} = \sum \left\{ a.x_{R'S'} \mid R \xrightarrow{a}_{G'} R',\ S \xrightarrow{a}_{G'} S', \text{ and } R' \leftrightarrow_{G'} S' \right\}$$

Note that, for each R' such that $R \xrightarrow{a}_{G'} R'$, $P_{x_{RS}}$ contains a summand of the form $a.x_{R'S'}$ for some S', and that the same also holds for each S' such that $S \xrightarrow{a}_{G'} S'$.

Take $E = \{x_{RS} = P_{x_{RS}} \mid x_{RS} \in V_E\}$. First I show that P T-provably satisfies E. To see that this is indeed the case, consider the substitution $\{R/x_{RS} \mid x_{RS} \in V_E\}$. Then

$$\begin{aligned} T \vdash P_{x_{RS}}\{R/x_{RS} \mid x_{RS} \in V_E\} &= \sum \left\{ a.R' \mid R \xrightarrow{a}_{G'} R' \right\} \\ &= R \text{ by Theorem 5.15} \end{aligned}$$

A symmetric argument gives that Q also T-provably satisfies E. □

The promised completeness result now follows easily from the previous theory.

Theorem 6.2 (Completeness) *Suppose that G is a regular GSOS system. Let G' and T denote the disjoint extension of G, and the strongly head normalizing equational theory constructed by the algorithm in Figure 2, respectively. Then, $T \cup \{(\mathrm{RSP})\}$ is complete for equality in* $\mathsf{Bisim}(G')$.

Proof: The fact that $\mathsf{Bisim}(G') \models T \cup \{(\mathrm{RSP})\}$ follows immediately from Theorem 5.15 and Proposition 4.3. It remains to be shown that for all $P, Q \in \mathrm{T}(\Sigma_{G'})$, $\mathsf{Bisim}(G') \models P = Q$ implies $T \cup \{(\mathrm{RSP})\} \vdash P = Q$. By the previous proposition, if P and Q are such that $\mathsf{Bisim}(G') \models P = Q$, then they both T-provably satisfy a common recursive specification E in the same variable x_0. Then (RSP) can be used to show that $P = \langle x_0 \mid E \rangle$ and $Q = \langle x_0 \mid E \rangle$. Therefore, $T \cup \{(\mathrm{RSP})\} \vdash P = Q$. □

7 Concluding Remarks

In this paper I have presented a class of GSOS specifications, with possibly a denumerable set of operations and rules, which generate regular processes. I have also shown how the techniques of [2] can be adapted to give a procedure for converting any such language definition to a complete equational axiom system for strong bisimulation of processes which, unlike the one presented in the aforementioned reference, does not use infinitary proof rules like AIP. A by-product of this study has been a class of infinitary GSOS systems which generate finite labelled transition systems, and are amenable to the development of a corresponding equational theory à là [2]. This class of infinitary GSOS systems which afford a neat algebraic treatment is obtained by restricting consideration to the simple operations from [1] that are uniformly bounded, in the sense of Definition 3.2, and regular (cf. Definition 3.8).

7.1 An Alternative Axiomatization of Positive GSOS Operations

At least for GSOS operations defined by *positive* rules only, the algorithm presented in [2] for GSOS systems, and adapted in this paper to a class of infinitary GSOS specifications, can be modified to generate a reasonably aesthetic equational axiomatization for a class of bounded operations. Let me recall that the need for the restriction to uniformly bounded operations in the application of the algorithm presented in [2] arises from the fact that operations like the one given by the rules (3) on page 7 cannot be neatly axiomatized in finitary fashion à là [2]. This is because the operation f given by the rules (3) is smooth, but not distinctive; moreover, as shown in Proposition 5.10, under mild assumptions, f cannot be expressed as a finite sum of unary distinctive operations.

However, it is not too difficult to see that f can be axiomatized, without recourse to auxiliary operations, by means of the following equations:

$$f(\mathbf{0}) = \mathbf{0} \tag{21}$$

$$f(x+y) = f(x)+f(y) \tag{22}$$

$$f(a_i.x) = \sum_{1\leq j\leq i} a_j.\mathbf{0} \quad (a_i \in \mathsf{Act}) \tag{23}$$

The point here is that, although not distinctive, f treats its one argument in a uniform way, in the sense that each rule for f has a positive antecedent for it. As a consequence of Lemma 5.2, we have that equation (22) holds for f, and this allows for a rather pleasing axiomatization of this operation.

This discussion leads to the following weakening of the notion of distinctiveness that will turn out to be sufficient for a reasonable axiomatization of a class of positive GSOS operations.

Definition 7.1 *Let f be an l-ary operation in an infinitary GSOS system G. I say that f is* positive *iff in every rule for f of the form (2), $n_i = 0$ for every argument i of f.*

A positive smooth operation f from an infinitary GSOS system G is consistent *iff the rules for f use the same target variables,* i.e., *if $x \xrightarrow{a} y$ is an antecedent of a rule for f, then every rule for f that tests x positively has an antecedent of the form $x \xrightarrow{b} y$ for some $b \in \mathsf{Act}$.*

A positive smooth operation f from an infinitary GSOS system G is weakly distinctive *iff it is consistent, and every rule for f tests the same arguments positively.*

For example, as previously remarked, the operation f given by the rules (3) is weakly distinctive. An interesting example of an operation from the literature on process algebras which is weakly distinctive, but *not* distinctive, is the internal choice operation $\oplus$ used in TCSP [16, 25] and the variant of CCS considered by De Nicola and Hennessy in [22, 17, 23]. (The notation I use is from [22, 17, 23]). The operation $\oplus$ is given by the rules:

$$\frac{}{x \oplus y \xrightarrow{\tau} x} \qquad \frac{}{x \oplus y \xrightarrow{\tau} y}$$

The internal choice operation is not distinctive because it has two distinct rules with positive trigger $\langle \emptyset, \emptyset \rangle$. However, all the rules for $\oplus$ are axioms, and this is sufficient to ensure that $\oplus$ is weakly distinctive.

Note that, by Lemma 5.2, weakly distinctive smooth operations distribute over $+$ in each of the arguments they test positively. (For instance, equation (22) can be obtained by applying Lemma 5.2 to the operation f described by the rules (3)). Furthermore, Lemma 5.3 gives inaction laws for these operations describing their interplay with the stopped process $\mathbf{0}$. (An example of such a law is equation (21)). Hence it is not too difficult to see that all that is needed to axiomatize weakly distinctive smooth operations which, like the one given by the rules (3), are bounded is a set of action laws describing the interplay between these operations and action prefixing. Below I shall give a way of generating such laws for positive weakly distinctive, bounded smooth operations.

Definition 7.2 *Let f be an l-ary operation in an infinitary GSOS system G. Let $\langle e_1, \ldots, e_l \rangle$ be a positive trigger of f. Then $R(f, \langle e_1, \ldots, e_l \rangle)$ denotes the set of rules for f which have $\langle e_1, \ldots, e_l \rangle$*

as positive trigger. For every rule $\rho \in R(f, \langle e_1, \ldots, e_l \rangle)$, c_ρ will denote its action, and $C_\rho[\vec{x}, \vec{y}]$ its target.

Note that if f is a bounded operation, then $R(f, \langle e_1, \ldots, e_l \rangle)$ is a finite set of rules for each trigger $\langle e_1, \ldots, e_l \rangle$ (cf. Definition 3.2).

Proposition 7.3 *Suppose f is a weakly distinctive, bounded, positive smooth operation of a disjoint extension G of* FINTREE*, and let $\langle e_1, \ldots, e_l \rangle$ be a positive trigger of f. Let I be the set of arguments which are tested positively by rules for f of the form (7), and, for every $i \in I$, let y_i denote the target variable corresponding to x_i in rules for f. Finally, let $\vec{P}$ be the vector of terms given by:*

$$P_i \equiv \begin{cases} e_i.y_i & i \in I \\ x_i & \textit{otherwise} \end{cases}$$

Then:

$$\mathsf{BISIM}(G) \models f(\vec{P}) = \sum \left\{ c_\rho.C_\rho[\vec{P}, \vec{y}] \mid \rho \in R(f, \langle e_1, \ldots, e_l \rangle) \right\} \tag{24}$$

Proof: Let G' be a disjoint extension of G. Then, for every $\Sigma_{G'}$-substitution σ, action $a \in \mathsf{Act}$ and $Q \in \mathrm{T}(\Sigma_{G'})$, it can be shown that:

$$f(\vec{P})\sigma \xrightarrow{a}_{G'} Q \;\Leftrightarrow\; \sum \left\{ c_\rho.C_\rho[\vec{P}, \vec{y}] \mid \rho \in R(f, \langle e_1, \ldots, e_l \rangle) \right\}\sigma \xrightarrow{a}_{G'} Q$$

Essentially this holds because only the rules in $R(f, \langle e_1, \ldots, e_l \rangle)$ apply to closed instantiations of $f(\vec{P})$, and the targets of the resulting transitions are suitable instantiations of the $C_\rho[\vec{P}, \vec{y}]$'s.

The thesis follows immediately from this fact. □

It is easy to see that the above proposition gives the equations (23) when applied to the operation f given by the rules (3). For the internal choice operation $\oplus$, Proposition 7.3 gives the natural equation

$$x \oplus y \quad = \quad \tau.x + \tau.y$$

The reader will easily convince himself/herself that distributivity equations, inaction laws and instances of (24) give a strongly head normalizing theory for terms built only from positive weakly distinctive, bounded smooth operations. Using this observation, and following the approach of Proposition 17, we can now axiomatize positive smooth operations that are consistent by expressing them as a finite sum of weakly distinctive ones.

Proposition 7.4 *Let G be an infinitary GSOS system that disjointly extends* FINTREE*. Assume that f is an l-ary positive, consistent smooth operation of G. Then there exists a disjoint extension G' of G with l-ary positive, weakly distinctive smooth operations $f_1, \ldots, f_n$ such that:*

$$\mathsf{BISIM}(G') \models f(\vec{x}) = f_1(\vec{x}) + \cdots + f_n(\vec{x}) \tag{25}$$

Moreover, if f is bounded, so is each f_i.

Proof: Assume that f is an l-ary positive, consistent smooth operation of G. I shall show how to partition the set R of rules for f in R_G into sets $R_1, \ldots, R_n$ in such a way that that, for all $1 \leq i \leq n$, f is weakly distinctive in the infinitary GSOS system obtained from G by removing all the rules in $R - R_i$. This can be done by partitioning the set of rules for f according to the following equivalence relation on the rules for f:

$$\rho \equiv_f \rho' \;\Leftrightarrow\; \rho, \rho' \text{ are rules for } f \text{ that test the same arguments positively.}$$

Note that the cardinality of $R/\equiv_f$ is at most 2^l. Let $R_1, \ldots, R_n$ be the equivalence classes of rules for f determined by $\equiv_f$.

Define $\Sigma_{G'}$ to be the signature obtained by extending Σ_G with fresh l-ary operation symbols $f_1, \ldots, f_n$. Next define $R_{G'}$ to be the set of rules obtained by extending R_G, for each i, with rules derived from the rules of R_i by replacing the operation symbol in the source by f_i. It is immediate to see that each operation f_i so defined is weakly distinctive, and that (25) holds. Moreover, by construction, each f_i is bounded if f itself was bounded. □

The results presented so far in this section give strong head normalization for the terms in an infinitary GSOS system built from positive, consistent, bounded smooth operations only. In particular, they can be used to obtain strong head normalization for the terms in the recursion-free sublanguages of the bounded de Simone systems in the beautiful presentation given by Vaandrager in [39, Definition 3.10].

To conclude this section, I shall now show how to axiomatize bounded positive GSOS operations with limited fan-in (cf. Definition 3.8). This can done following the spirit of Proposition 5.13.

First, we need a technical lemma, which is a slightly sharpened version of Lemma 5.12 on page 19.

Lemma 7.5 *Suppose G is an infinitary GSOS system and $P = f(\vec{z})$ and $Q = f'(\vec{v})$ are terms over Σ_G with variables that do not occur in R_G. Suppose that there exists a 1-1 correspondence between rules for f and rules for f' such that, whenever a rule ρ for f with source $f(\vec{x})$ is related to a rule ρ' for f' with source $f'(\vec{y})$, we have that there exists a bijective map $\xi_{\rho,\rho'}$ from the target variables of ρ' to those of ρ such that:*

1. $\mathsf{ante}(\rho)\{\vec{z}/\vec{x}\} = \mathsf{ante}(\rho')(\{\vec{v}/\vec{y}\} \circ \xi_{\rho,\rho'})$, *and*
2. $\mathsf{target}(\rho)\{\vec{z}/\vec{x}\} = \mathsf{target}(\rho')(\{\vec{v}/\vec{y}\} \circ \xi_{\rho,\rho'})$.

Then $\mathsf{BISIM}(G) \models P = Q$.

Proof: (Following the proof of Lemma 4.12 in [2]). Suppose that G' is a disjoint extension of G and σ is a closed $\Sigma_{G'}$-substitution. We have to prove $P\sigma \leftrightarrows_{G'} Q\sigma$. For this it suffices to show that, for all $a \in \mathsf{Act}$ and $S \in \mathrm{T}(\Sigma_{G'})$,

$$P\sigma \xrightarrow{a}_{G'} S \;\Leftrightarrow\; Q\sigma \xrightarrow{a}_{G'} S.$$

In fact, it is sufficient to prove the implication '⇒', since the reverse implication is symmetric. So suppose $P\sigma \xrightarrow{a}_{G'} S$. I will prove $Q\sigma \xrightarrow{a}_{G'} S$.

Since $P\sigma \xrightarrow{a}_{G'} S$, it must be the case that $R_{G'}$ contains a rule ρ of the form

$$\frac{H}{f(\vec{x}) \xrightarrow{a} T}$$

and there exists a $\Sigma_{G'}$-substitution τ such that

$$\tau(x_i) \xrightarrow{a_{ij}}_{G'} \tau(x'_{ij}) \quad \text{for every positive antecedent } x_i \xrightarrow{a_{ij}} x'_{ij} \in H \tag{26}$$

$$\tau(x_i) \stackrel{b_{ik}}{\not\rightarrow}_{G'} \quad \text{for every negative antecedent } x_i \stackrel{b_{ik}}{\not\rightarrow} \in H \tag{27}$$

$$f(\vec{x})\tau \equiv P\sigma \tag{28}$$

$$T\tau \equiv S \tag{29}$$

Since G' disjointly extends G, we know that ρ is a rule of G. Thus there exists a rule ρ' in R_G (and hence in $R_{G'}$) of the form

$$\frac{H'}{f'(\vec{y}) \xrightarrow{a} T'}$$

such that, by the proviso of the lemma,

$$H\{\vec{z}/\vec{x}\} = H'(\{\vec{v}/\vec{y}\} \circ \xi_{\rho,\rho'}) \quad (30)$$

$$T\{\vec{z}/\vec{x}\} \equiv T'(\{\vec{v}/\vec{y}\} \circ \xi_{\rho,\rho'}) \quad (31)$$

for some bijective map $\xi_{\rho,\rho'}$ from the target variables of ρ' to those of ρ.

Let τ' be the $\Sigma_{G'}$-substitution defined by

$$\tau'(w) \equiv \begin{cases} \sigma(w) & \text{if } w \text{ occurs in } \vec{z} \text{ or } \vec{v} \\ \tau(w) & \text{otherwise} \end{cases}$$

I claim that for all variables w that do not occur in $\vec{z}$ or $\vec{v}$, $\tau' \circ \langle \vec{z}/\vec{x} \rangle(w) \equiv \tau(w)$. In fact, either w does not occur in $\vec{x}$ and we have $\tau' \circ \langle \vec{z}/\vec{x} \rangle(w) \equiv \tau'(w) \equiv \tau(w)$, or w does occur in $\vec{x}$, in which case the claim follows since

$$\begin{aligned} f(\vec{x})\tau &\equiv f(\vec{z})\sigma && \text{(by (28))} \\ &\equiv f(\vec{z})\tau' && \text{(since } \sigma = \tau' \text{ on } \vec{z}) \\ &\equiv f(\vec{x})\{\vec{z}/\vec{x}\}\tau' \end{aligned}$$

Now it is not too difficult to show that $Q\sigma \xrightarrow{a}_{G'} S$ does indeed hold by applying the substitution $\tau' \circ \{\vec{v}/\vec{y}\} \circ \xi_{\rho,\rho'}$ to the rule ρ'. □

Proposition 7.6 *Suppose G is an infinitary GSOS system containing a positive operation f with arity l and limited fan-in that is not both smooth and consistent. Then there exists a disjoint extension G' of G with a positive smooth and consistent operation f' with arity l' (possibly different from l) and limited fan-in, and there exist vectors $\vec{z}$ of l distinct variables, and $\vec{v}$ of l' variables in $\vec{z}$ (possibly repeated), such that:*

$$\mathsf{BISIM}(G') \models f(\vec{z}) = f'(\vec{v}) \quad (32)$$

Moreover, if f is bounded, then so is f'.

Proof: Let f be a positive operation with arity l and limited fan-in that is not both smooth and consistent. First of all, we determine the arity of its smooth and consistent version f'. For ρ a GSOS rule for f of the form

$$\frac{\bigcup_{i=1}^{l} \left\{ x_i \xrightarrow{a_{ij}} y_{ij} | 1 \leq j \leq m_i \right\}}{f(x_1, \ldots, x_l) \xrightarrow{c} C[\vec{x}, \vec{y}]} \quad (33)$$

and $1 \leq i \leq l$, let $N(\rho, i)$ be given by:

$$\begin{array}{rl} m_i & \text{if } x_i \text{ does not occur in the target of } \rho \\ m_i + 1 & \text{otherwise} \end{array}$$

Note that, as f has limited fan-in, $N(\rho, i)$ is less than or equal to $m_{(f,i)} + 1$ for all ρ, where $m_{(f,i)}$ is the maximum number of positive antecedents for i in the rules for f. $N(f, i)$ is then defined as the maximum over all rules ρ for f of the $N(\rho, i)$s. Let $l' = \sum_{i=1}^{l} N(f, i)$ and let f' be a fresh operation symbol. Then $\Sigma_{G'}$ is defined as the signature that extends Σ_G with an l'-ary operation symbol f'. Let $\vec{w} = w_{11}, \ldots, w_{1N(f,1)}, \ldots, w_{l1}, \ldots, w_{lN(f,l)}$ and $\vec{u} = u_{11}, \ldots, u_{1N(f,1)}, \ldots, u_{l1}, \ldots, u_{lN(f,l)}$ be disjoint vectors of l' different variables. Suppose ρ is a rule for f as in (33) and consider the substitution τ_ρ given by:

$$\tau_\rho(w) = \begin{cases} x_i & \text{if } w = w_{ij} \text{ for some } 1 \leq i \leq l \text{ and } 1 \leq j \leq N(f,i) \\ y_{ij} & \text{if } w = u_{ij} \text{ for some } 1 \leq i \leq l \text{ and } 1 \leq j \leq m_i \\ w & \text{otherwise} \end{cases}$$

I now wish to construct a rule ρ' for f' such that $\rho'\tau_\rho$ and ρ are identical with the exception of their sources. This can be done as follows. Let ρ' be the positive smooth GSOS rule obtained from ρ by replacing each antecedent $x_i \stackrel{a_{ij}}{\rightarrow} y_{ij}$ with $w_{ij} \stackrel{a_{ij}}{\rightarrow} u_{ij}$, taking $f'(\vec{w})$ as the source of the rule, and replacing each occurrence of a variable x_i in the target with w_{im_i+1}. It is immediate to verify that the rule ρ' does meet the desired requirement.

Define $R_{G'}$ to be a set of rules that extends R_G with a rule ρ', defined as above, for each rule ρ for f. It is easy to see that, by construction, f' is a positive, consistent smooth operation. Moreover, again by construction, f' is bounded if so is f.

Let now $\vec{z} = z_1, \ldots, z_l$ be a vector of different variables, all of them not occurring in R_G, and let $\vec{v} = v_{11}, \ldots, v_{1N(f,1)}, \ldots, v_{l1}, \ldots, v_{lN(f,l)}$ be the vector of length l' given by $v_{ij} = z_i$. It is easy to see that, for each pair ρ, ρ' of corresponding rules:

1. $\mathsf{ante}(\rho)\{\vec{z}/\vec{x}\} = \mathsf{ante}(\rho')(\{\vec{v}/\vec{w}\} \circ \xi_{\rho,\rho'})$, and
2. $\mathsf{target}(\rho)\{\vec{z}/\vec{x}\} = \mathsf{target}(\rho')(\{\vec{v}/\vec{w}\} \circ \xi_{\rho,\rho'})$

where $\xi_{\rho,\rho'}$ denotes the restriction of τ_ρ to the target variables of ρ'. Thus we can apply Lemma 7.5 to obtain that $\mathsf{BISIM}(G') \models f(\vec{z}) = f'(\vec{v})$, as required. □

As an example of application of the methods used in the proof of the above proposition, let us consider the (useless) positive GSOS operation f given by the rules:

$$\frac{x \stackrel{a}{\rightarrow} x_1, \; x \stackrel{b}{\rightarrow} x_2}{f(x,y) \stackrel{a}{\rightarrow} x} \qquad \frac{x \stackrel{a}{\rightarrow} z_1, \; x \stackrel{c}{\rightarrow} z_2}{f(x,y) \stackrel{c}{\rightarrow} \mathbf{0}}$$

This operation is not smooth as it has more than one positive hypothesis for its first argument. The smooth and consistent version of f given by the above proposition is the ternary operation f' given by the rules:

$$\frac{x \stackrel{a}{\rightarrow} x', \; y \stackrel{b}{\rightarrow} y'}{f'(x,y,z) \stackrel{a}{\rightarrow} z} \qquad \frac{x \stackrel{a}{\rightarrow} x', \; y \stackrel{c}{\rightarrow} y'}{f'(x,y,z) \stackrel{c}{\rightarrow} \mathbf{0}}$$

The corresponding instance of equation (32) relating f and its smooth and consistent version f' is the following law:

$$f(x,y) \quad = \quad f'(x,x,x)$$

Note that the fact that f does not use its second argument is reflected in the above equation. In fact f' makes three copies of x, but no copy of y.

So far, I have been unable to find a strategy for axiomatizing general GSOS operations (with negative premises) that uses the weaker notion of distinctiveness from Definition 7.1, and does not contain equation schemas.

7.2 Further Work

The developments of this paper suggest several interesting topics for further research, some of which are already being investigated by the author. Below I list some directions for further work that I plan to explore.

The class of regular operations that has been axiomatized in this paper is quite large, and includes most of the standard operations found in the literature on process algebras. A notable exception is the desynchronizing operation Δ present in the early versions of Milner's SCCS [27, 21]. This operation is given by the rules (one such rule for each $a \in \mathsf{Act}$):

$$\frac{x \stackrel{a}{\rightarrow} x'}{\Delta x \stackrel{a}{\rightarrow} \delta \Delta x'}$$

which are not simple. It is a challenging open problem to extend the class of regular GSOS operations considered in this paper to include operations like Milner's Δ.

In this paper, I have not considered issues related to the effectiveness of regular infinitary GSOS languages, and of the resulting axiomatizations. Standard GSOS languages à là Bloom, Istrail and Meyer enjoy pleasant recursion-theoretic properties, and any proper extension of their work to infinitary languages ought to possess at least some of them. In future work I shall investigate a class of infinitary, recursive GSOS languages — that is infinitary GSOS languages that could conceivably have interpreters — and study the resulting axiomatizations produced by the methods of [2].

Finally, it would be interesting to find alternative ways of axiomatizing general GSOS operations that, like the one presented in Section 7.1, do not use the full power of the technical notion of distinctiveness used in [2] and in this study.

Acknowledgements: Many thanks to Bard Bloom and Frits Vaandrager for the joint work [2] which formed the main inspiration for this paper. Y.S. Ramakrishna provided much needed terminological assistance.

References

[1] L. Aceto. GSOS and finite labelled transition systems. Report 6/93, Computer Science, University of Sussex, Brighton, 1993. To appear in *Theoretical Computer Science*.

[2] L. Aceto, B. Bloom, and F.W. Vaandrager. Turning SOS rules into equations. Report CS-R9218, CWI, Amsterdam, June 1992. To appear in the LICS 92 Special Issue of *Information and Computation*.

[3] D. Austry and G. Boudol. Algèbre de processus et synchronisations. *Theoretical Computer Science*, 30(1):91–131, 1984.

[4] E. Badouel and P. Darondeau. Structural operational specifications and trace automata. In W.R. Cleaveland, editor, *Proceedings CONCUR 92,* Stony Brook, NY, USA, volume 630 of *Lecture Notes in Computer Science*, pages 302–316. Springer-Verlag, 1992.

[5] J.C.M. Baeten and C. Verhoef. A congruence theorem for structured operational semantics. In E. Best, editor, *Proceedings CONCUR 91,* Hildesheim, volume 715 of *Lecture Notes in Computer Science*, pages 477–492. Springer-Verlag, 1993.

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

[7] J.A. Bergstra and J.W. Klop. Verification of an alternating bit protocol by means of process algebra. In W. Bibel and K.P. Jantke, editors, *Math. Methods of Spec. and Synthesis of Software Systems '85, Math. Research 31*, pages 9–23, Berlin, 1986. Akademie-Verlag. First appeared as: Report CS-R8404, CWI, Amsterdam, 1984.

[8] J.A. Bergstra and J.W. Klop. A complete inference system for regular processes with silent moves. In F.R. Drake and J.K. Truss, editors, *Proceedings Logic Colloquium 1986*, pages 21–81, Hull, 1988. North-Holland. First appeared as: Report CS-R8420, CWI, Amsterdam, 1984.

[9] B. Bloom. *Ready Simulation, Bisimulation, and the Semantics of CCS-like Languages.* PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, August 1989.

[10] B. Bloom. CHOCOLATE: Calculi of Higher Order COmmunication and LAmbda TErms (preliminary report), 1993. To appear in the Proceedings of POPL '94.

[11] B. Bloom. Ready, set, go: Structural operational semantics for linear-time process algebras. Technical Report TR 93-1372, Cornell, August 1993.

[12] B. Bloom. Structural operational semantics for weak bisimulations. Technical Report TR 93-1373, Cornell, August 1993.

[13] B. Bloom, S. Istrail, and A.R. Meyer. Bisimulation can't be traced: preliminary report. In *Conference Record of the Fifteenth Annual ACM Symposium on Principles of Programming Languages*, pages 229–239, 1988. Full version available as Technical Report 90-1150, Department of Computer Science, Cornell University, Ithaca, New York, August 1990. To appear in the *Journal of the ACM*.

[14] R.N. Bol and J.F. Groote. The meaning of negative premises in transition system specifications (extended abstract). In J. Leach Albert, B. Monien, and M. Rodríguez, editors, *Proceedings* 18^{th} *ICALP*, Madrid, volume 510 of *Lecture Notes in Computer Science*, pages 481–494. Springer-Verlag, 1991. Full version available as Report CS-R9054, CWI, Amsterdam, 1990.

[15] D.J.B. Bosscher. Term rewriting properties of SOS axiomatizations, 1993. To appear as CWI technical report. An extended abstract will appear in the Proceedings of TACS '94.

[16] S.D. Brookes, C.A.R. Hoare, and A.W. Roscoe. A theory of communicating sequential processes. *Journal of the ACM*, 31(3):560–599, 1984.

[17] R. De Nicola and M. Hennessy. CCS without τ's. In H. Ehrig, R. Kowalski, G. Levi, and U. Montanari, editors, *Proceedings TAPSOFT 87, Vol. I*, volume 249 of *Lecture Notes in Computer Science*. Springer-Verlag, 1987.

[18] R.J. van Glabbeek and F.W. Vaandrager. Modular specification of process algebras. *Theoretical Computer Science*, 113(2):293–348, 1993.

[19] J.F. Groote. Transition system specifications with negative premises. Report CS-R8950, CWI, Amsterdam, 1989. An extended abstract appeared in J.C.M. Baeten and J.W. Klop, editors, *Proceedings CONCUR 90*, Amsterdam, LNCS 458, pages 332–341. Springer-Verlag, 1990.

[20] J.F. Groote and F.W. Vaandrager. Structured operational semantics and bisimulation as a congruence. *Information and Computation*, 100(2):202–260, October 1992.

[21] M. Hennessy. A term model for synchronous processes. *Information and Control*, 51(1):58–75, 1981.

[22] M. Hennessy. Synchronous and asynchronous experiments on processes. *Information and Control*, 59:36–83, 1983.

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

[24] M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM*, 32(1):137–161, 1985.

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

[26] E. Madelaine and D. Vergamini. Finiteness conditions and structural construction of automata for all process algebras. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 3:275–292, 1991.

[27] R. Milner. On relating synchrony and asynchrony. Technical Report CSR–75–80, Department of Computer Science, University of Edinburgh, 1981.

[28] R. Milner. Calculi for synchrony and asynchrony. *Theoretical Computer Science*, 25:267–310, 1983.

[29] R. Milner. A complete inference system for a class of regular behaviours. *Journal of Computer and System Sciences*, 28:439–466, 1984.

[30] R. Milner. *Communication and Concurrency.* Prentice-Hall International, Englewood Cliffs, 1989.

[31] R. Milner. A complete axiomatisation for observational congruence of finite-state behaviors. *Information and Computation*, 81(2):227–247, May 1989.

[32] D.M.R. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, 5^{th} *GI Conference*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer-Verlag, 1981.

[33] J. Parrow. The expressive power of parallelism. *Future Generations Computer Systems*, 6:271–285, 1990.

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

[35] R. de Simone. *Calculabilité et Expressivité dans l'Algebra de Processus Parallèles* MEIJE. Thèse de 3^e cycle, Univ. Paris 7, 1984.

[36] R. de Simone. Higher-level synchronising devices in MEIJE–SCCS. *Theoretical Computer Science*, 37:245–267, 1985.

[37] I. Ulidowski. Equivalences on observable processes. In *Proceedings* 7^{th} *Annual Symposium on Logic in Computer Science,* Santa Cruz, California, pages 148–159. IEEE Computer Society Press, 1992.

[38] F.W. Vaandrager. On the relationship between process algebra and input/output automata (extended abstract). In *Proceedings* 6^{th} *Annual Symposium on Logic in Computer Science,* Amsterdam, pages 387–398. IEEE Computer Society Press, 1991.

[39] F.W. Vaandrager. Expressiveness results for process algebras. In J.W. de Bakker, W.P. de Roever, and G. Rozenberg, editors, *Proceedings REX Workshop on Semantics: Foundations and Applications,* Beekbergen, The Netherlands, volume 666 of *Lecture Notes in Computer Science*, pages 609–638. Springer-Verlag, 1993.

[40] C. Verhoef. A congruence theorem for structured operational semantics with predicates and negative premises. Report CSN 93/18, Eindhoven University of Technology, 1993. To appear in *Proceedings CSL'93*, Swansea 1993.