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

regystems [illegible] ensurThd(for)Tj18(systems-44Tj40Td(regular 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 $x \stackrel{a_i}{\to} y$ 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:

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

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

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]*:*

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

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

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

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

Definition 3.4 *A GSOS rule of the form (2) is* simple *iff* $C[$

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
$$T_{\mathsf{RCCS}} \vdash \; P = \sum \left\{ a.Q \mid P \xrightarrow{a}_{\mathsf{RCCS}} Q \right\}$$
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 preser2999.7(equational,)-12999.4(apart)-12000.6(from)-11000.7(

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

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

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*

$$\Big\{ x_i \xrightarrow{a_i} y_i \Big|$$

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

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

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, $1 \leq i \leq l$, 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 $1 \leq h \leq l$ such that $h \neq i$, 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 $1 \leq k \leq n_h$.

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 $i \in an$

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

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

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. □

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:

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:

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

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

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

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

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, $m \leq 2$

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 $f(P) \Leftrightarrow_{G'} f_1(P) + \cdots + f_n(P)$, it must be the case that, for some $1 \leq i \leq n$ and $1 \leq j < h \leq n$

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 $\vec{w} = w$ $\qquad$ *is*

Proposition 5.14 *Let* 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 $g(P) \leftrightarrow_{G'} g_1(P^{l_1}) + \cdots + g_n(P^{l_n})$, there exists $1 \leq j \leq n$ and $Q \in \mathrm{T}(\Sigma_{G'})$ such that $g_j(P^{l_j}) \xrightarrow{a_{l+1}}_{G'} Q \leftrightarrow_{G'} \mathbf{0}$. 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 $b \in$ 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 $g_1(R^{l_1}) + \cdots + g$

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

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

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

Next define

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

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

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

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

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 $P, Q \in \mathrm{T}(\Sigma_{G'})$, $\mathsf{Bisim}(G$

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:

$$f(\mathbf{0}$$

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 thbget.*

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 '$\Rightarrow$', 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,

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

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

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

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 $a \in \mathsf{Act}$):

$$\frac{x \xrightarrow{a} x'}{\Delta x \xrightarrow{a} \delta \Delta x'}$$

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

[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.