A general conservative extension theorem in process algebras with inequalities

Pedro R. D'Argenio
LIFIA, Dept. of Comp. Sci.
Universidad Nacional de La Plata
e-mail: pedro@info.unlp.edu.ar

Abstract

A general conservative extension theorem for process algebras with inequalities is stated. General results for proving operational conservative extension up to a semantic preorder and equational conservative extension of equational specifications with inequalities are proposed. The proof of these facts reduces to check some simple conditions in the term deduction system of the process theory. A general theorem for proving completeness in extended process algebras with inequalities is given as a corollary.

1 Introduction

Process theories such as CCS, CSP and ACP have been extended with new features such as real-time and probabilistics. Hence, it is desirable that any property which has been proved in the old theory remains valid (for the old part) in the extended theory. That is, an *extended* theory should be *conservative* somehow with respect to the original one.

Conservativity in transition system specifications (or term deduction systems) was studied in [GV92], [Gro93], [BG91], [Ver94b] and [FV95]. In this setting, *(operational) conservativity* means that the provable transitions for an original term are the same both in the original and in the extended term deduction systems.

Verhoef proposed in [Ver94b] a general conservative extension theorem for equational specifications. Here, *(equational) conservativity* means that exactly the same identities between closed terms in the original framework can be proved both in the original and in the extended equational specifications. That theorem solves several complications when an equational specification is extended. For instance, it avoids to deal with term rewriting analysis which are frequently used to prove equational conservativity. These term rewriting systems often have no nice properties. Thus, according to [Ver94b], the problem of proving conservative extension of transition system based equational specifications (namely process algebra) can be reduced to check operational conservative extension of the associated term deduction system.

This article extends that work to deal with semantic preorders defined on transition systems and equational specifications with inequalities or *inequational specifications*, as they will be called in this article.

The proposed method involves three steps. The first one is to state conservative extension of the term deduction systems. In order to do this, an operational conservative extension theorem is given. It is a simple variation of that introduced in [FV95] that considers term deduction systems which have a unique well supported model (see also [Gla95]). The second step states operational conservative extension up to a preorder defined exclusively in terms of transition relations and predicates. This is proved to be an immediate consequence of the first step. The last step is to prove conservative extension of the inequational specifications. With this purpose, a general conservative extension theorem is introduced for inequational specifications that axiomatize that kind of preorders.

Thus, the proof of inequational conservative extension and operational conservative extension up to certain preorder reduces to check operational conservative extension of the term deduction system, which can be done by verifying some simple conditions.

The paper is organized as follows. Section 2 introduces preliminary concepts. The first paragraph briefly explains the SOS theory. The second paragraph introduces basic notions of the algebraic treatment of inequational specifications. Section 3 states the results of this paper. In the first paragraph, operational conservativity results are stated. The second paragraph deals with inequational conservativity and proves the general conservative extension theorem. Finally, some examples of application are given.

Acknowledgements. I am grateful to Chris Verhoef for encouraging me to make this paper, and also for his valuable comments and suggestions. I also thank Twan Basten and the anonymous referees for their useful suggestions.

2 Preliminaries

This section briefly recalls some notions about SOS theory and inequational specifications. The first paragraph explains SOS theory following [Ver94a], [Gla95] and [FV95], since they seem to have the most general treatment. The second paragraph gives some basic notions about the algebraic treatment of inequational specifications.

Some concepts of SOS

Assume an infinite set V of variables. A *(single sorted) signature* Σ is a set of functions symbols together with their arity. The notion of *term* (over Σ) is defined as expected: $x \in V$ is a term; and, if $t_1, \ldots, t_n$ are terms and if $f \in \Sigma$ is n-ary then $f(t_1, \ldots, t_n)$ is a term. A term is also called an *open term* and the set of open terms is denoted by $O(\Sigma)$. A term containing no variables is called a *closed term* and the set of closed terms is denoted by $C(\Sigma)$. Let $t \in O(\Sigma)$ then $var(t) \subseteq V$ is the set of all variables occurring in t.

A *substitution* is a function $\sigma : V \to O(\Sigma)$. This map can easily be extended to the set of all terms by substituting for each variable occurring in an open term its σ-image.

Definition 2.1 (Term deduction systems) A *term deduction system* is a structure (Σ, D) where Σ is a signature and D is a set of *deduction rules*. The set $D = D(T_p, T_r)$ is parametrized with two sets which are called respectively the set of *predicate symbols* and the set of *relation symbols*. Let $s, t, u \in O(\Sigma)$, $P \in T_p$ and $R \in T_r$. Expressions Ps, $\neg Ps$, tRu and $t\neg R$ are called *formulas*. Formulas Ps and tRu are called *positive* and $\neg Ps$ and $t\neg R$ are called *negative*. Let F be a set of formulas. $PF(F)$ denotes the subset of positive formulas of F and $NF(F)$ denotes the subset of negative formulas of F.

A deduction rule $d \in D$ has the form $\frac{H}{C}$; with H a set of formulas, and C a positive formula. Elements of H are called the *hypothesis* of d, and C is the *conclusion* of d. If the set of hypothesis of a deduction rule is empty, it is called an axiom, and it will be denoted only by its conclusion provided that no confusion arises. The notions of "substitution", "*var*" and "closed" extend to formulas and deduction rules as expected. ∎

Note that arbitrary many premises are allowed in the set of hypotheses of a deduction rule.

Example 2.1 As a running example, the operational semantics of the process algebra PA [BK84, BW90] is presented. The signature contains constants a of a set $\mathbf{A}$ of *atomic actions*, and four binary operators: the *alternative composition* $(+)$, the *sequential composition* $(\cdot)$, the *parallel composition* or *merge* $(||)$ and the *left merge* $(\lfloor\!\lfloor)$. It is easy to see that the above signature plus the deduction rules in Table 1 form a term deduction system. This term deduction system has relations $\xrightarrow{a}$ and predicates $\xrightarrow{a}\surd$ for all $a \in \mathbf{A}$. The intended interpretation of $x \xrightarrow{a} y$ is that a process x executes an action a and then behaves like y. The intended meaning of $x \xrightarrow{a} \surd$ is that x terminates successfully after the execution of a.

$$a \xrightarrow{a} \surd \qquad \frac{x \xrightarrow{a} x'}{x \cdot y \xrightarrow{a} x' \cdot y} \qquad \frac{x \xrightarrow{a} \surd}{x \cdot y \xrightarrow{a} y}$$

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

$$\frac{x \xrightarrow{a} x'}{x||y \xrightarrow{a} x'||y \quad y||x \xrightarrow{a} y||x'} \qquad \frac{x \xrightarrow{a} \surd}{x||y \xrightarrow{a} y \quad y||x \xrightarrow{a} y}$$

$$\frac{x \xrightarrow{a} x'}{x \lfloor\!\lfloor y \xrightarrow{a} x'||y} \qquad \frac{x \xrightarrow{a} \surd}{x \lfloor\!\lfloor y \xrightarrow{a} y}$$

Table 1: Operational rules for PA

In addition, notice that the set of operations $\Sigma_{\mathrm{BPA}} = \mathbf{A} \cup \{+, \cdot\}$ together with the deduction rules above the line in Table 1 form another term deduction system which will be called BPA [BW90]. Yet another term deduction system is formed by the signature $\Sigma_{\mathrm{MRG}} = \mathbf{A} \cup \{+, \cdot, ||, \lfloor\!\lfloor\}$ plus the deduction rules below the line in Table 1. It will be called MRG since it defines the operational semantics of the merge operators. ∎

The following definition tells when a formula is provable from a term deduction system.

Definition 2.2 (Proof of a rule) Let $T = (\Sigma, D)$ be a term deduction system. A *proof* of a rule $\frac{H}{C}$ from T is a well-founded, upwardly branching tree of which the nodes are labelled with formulas of T, such that:

1. the root is labelled with C, and

2. if ϕ is the label of a node q and F is the set of labels of the nodes directly above q, then

   - either $F = \emptyset$ and $\phi \in H$,
   - or $\frac{F}{\phi}$ is a substitution instance of a rule $d \in D$.

If a proof of $\frac{H}{C}$ from T exists, then $\frac{H}{C}$ is *provable* from T, notation $T \vdash \frac{H}{C}$. ∎

Definition 2.3 Let T be a term deduction system. Let $F(T)$ be the set of all closed formulas of T. Let $PF(T)$ be the set of all positive closed formulas over T. Let $X \subseteq PF(T)$ and $\phi \in F(T)$. Then $X \models \phi$ (read ϕ *hols in* X) is defined according to the form of ϕ by:

$$\begin{array}{lll} X \models sRt & \text{if} & sRt \in X, \\ X \models Ps & \text{if} & Ps \in X, \\ X \models s\neg R & \text{if} & \forall t \in C(\Sigma) : sRt \notin X, \\ X \models \neg Ps & \text{if} & Ps \notin X. \end{array}$$

∎

The purpose of a term deduction system is to define a set of positive formulas that can be deduced using the deduction rules. That is, one wants to talk about models for term deduction systems. Moreover, one would like to work with the most representative model. Meaning of transition system specifications was studied by *Bol & Groote* in [BG91] and more widely by *van Glabbeek* in [Gla95]. In this article, the definition of *well supported model* or *stability* is taken from [Gla95] although it was originally introduced in [BG91].

Definition 2.4 (Well supported model) Let $T = (\Sigma, D)$ be a term deduction system and let $X \subseteq PF(T)$ be a set of positive closed formulas. *X is a well supported model* T if

$$\phi \in X \iff \begin{array}{l} \text{there exists a closed rule } \frac{H}{\phi} \text{ without positive hypotheses such that } T \vdash \frac{H}{\phi} \\ \text{and for all } h \in H,\ X \models h \end{array}$$

If T has a unique well supported model, it is denoted by $S(T)$. ∎

Definition 2.5 (Source dependence) [Gla93a, FV95] Let $d = \frac{H}{C}$ be a deduction rule where C has the form Pt or tRt'. The collection of *source dependent* variables $SV(d)$ is defined inductively as follows:

- $var(t) \subseteq SV(d)$; and
- if $sR's' \in H$ and $var(s) \subseteq SV(d)$, then $var(s') \subseteq SV(d)$.

d is called *source dependent* if $SV(d) = var(d)$. A term deduction system is called *source dependent* if all of its rules are. ∎

Example 2.2 The operational rules of PA are in *path* format [BV93] and they are source dependent. As PA has only positive rules, it has a unique well supported model (see [Gla95]). ∎

Many equivalences are definable in terms of relation and predicate symbols only. A complete survey of interleaving semantics equivalences was made by *van Glabbeek* in [Gla90] (for concrete processes) and [Gla93b] (for processes with abstraction). Equivalences for true concurrency were also defined in that way, for instance, step bisimulation [NP84, Pom86] and pomset bisimulation [BC88]. Besides, not only equivalences are defined in term of relation and predicate symbols but also preorders. Examples of preorders are: simulation, n-nested simulations [GV92], ready simulation [BIM88], the preorder for the degree of parallelism based on pomset bisimulation of [Ace91], the "more distributed than" preorders of [Cas93] and [Yan93], the preorder for unstable nondeterminism of [VB95] and the preorder of bisimulation with divergence of [Abr87] and the ones of [Wal90].

Example 2.3 Simulation is defined for the PA terms as an example of preorders defined in terms of relation and predicate symbols.

A binary relation S on the set of closed PA terms is a *simulation* if for all $(s,t) \in S$ and for all $a \in \mathbf{A}$, the two following *transfer properties* hold:

- $\forall s' : s \xrightarrow{a} s' \implies \exists t' : t \xrightarrow{a} t' \ \wedge \ (s', t') \in S$,
- $s \xrightarrow{a} \surd \implies t \xrightarrow{a} \surd$.

If there is a simulation S such that $(s,t) \in S$, then s *is simulated by* t, notation $s \underrightarrow{\subset} t$. ∎

Some concepts of inequational specifications

Definition 2.6 (Inequational specifications) An *inequational specification* is a structure (Σ, E) where Σ is a signature and E is a set of *inequalities* of the form $s \leq t$ where $s, t \in O(\Sigma)$. Sometimes, E also contains *rules* or *conditional inequalities* $G \Rightarrow s \leq t$ where G is a set of inequalities. $s = t$ is often written standing for $s \leq t$ and $t \leq s$. The notion of "substitution", "*var*" and "closed" extend to *inequalities* as expected. ∎

An inequational specification is indeed an equational specification. I chose such a name to make clear that inequalities are explicitly managed and to relate the idea that their models will be based in preorders instead of equivalences.

Definition 2.7 (Derivability) Let $L = (\Sigma, E)$ be an inequational specification. Let $s, t \in O(\Sigma)$. An inequality $s \leq t$ *can be derived* from E, notation $E \vdash s \leq t$, according to the following definition

1. $s \leq t \ \in \ E$ implies $E \vdash s \leq t$;
2. for all substitutions $\sigma : V \to O(\Sigma)$, $E \vdash s \leq t$ implies $E \vdash \sigma(s) \leq \sigma(t)$;
3. let $\sigma : V \to O(\Sigma)$ and let $G \Rightarrow s \leq t \ \in \ E$, if for all $u \leq v \ \in \ G$, $E \vdash \sigma(u) \leq \sigma(v)$ then $E \vdash \sigma(s) \leq \sigma(t)$;
4. for all $f \in \Sigma$ with arity n, $E \vdash s_i \leq t_i$ for all $i \in \{1, \ldots, n\}$ imply $E \vdash f(s_1, \ldots, s_n) \leq f(t_1, \ldots, t_n)$;
5. $E \vdash t \leq t$;
6. $E \vdash s \leq t$ and $E \vdash t \leq u$ implies $E \vdash s \leq u$.

■

Notice that the rule for symmetry is not included, and so, the equational specifications one uses to manage are a particular case of inequational specifications. Actually, 1, 2 and 3 are enough to define derivability. 4, 5 and 6 may be rules (and axioms) in E and, moreover, they could not be present.

Example 2.4 The signature of PA together with the axioms in Table 2 form an inequational specification. It will be called $\mathrm{PA}^{\leq}$. In addition, two inequational specifications more are considered: the signature of BPA together with axioms in the left column form the inequational specification $\mathrm{BPA}^{\leq}$, and the signature of MRG together with axioms in the right column form the inequational specification $\mathrm{MRG}^{\leq}$.

A1	$x + y = y + x$	M1	$x \| y = x \lfloor\!\lfloor y + y \lfloor\!\lfloor x$
A2	$x + (y + z) = (x + y) + z$	M2	$a \lfloor\!\lfloor x = a \cdot x$
A3	$x + x = x$	M3	$a \cdot x \lfloor\!\lfloor y = a \cdot (x \| y)$
A4	$(x + y) \cdot z = x \cdot z + y \cdot z$	M4	$(x + y) \lfloor\!\lfloor z = x \lfloor\!\lfloor z + y \lfloor\!\lfloor z$
A5	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$		
SM	$x \leq x + y$	MP	$x \cdot y \leq x \| y$

Table 2: Axioms of $\mathrm{PA}^{\leq}$

Axioms A1 to A5 and M1 to M4 are the well known axioms for the PA process algebra [BK84, BW90]. SM is the axiom of simulation; it is often used for theoretical reasons in process algebra theory (for instance it was used for proving completeness of BPA. See [BW90]). MP stands for "more parallel" and introduces the idea that $x\|y$ has a "more parallel behaviour" than $x \cdot y$. For closed terms, MP can be deduced by induction from the other axioms. ■

Definition 2.8 (Algebras and axiomatizations) An *algebra* is a set A of elements together with certain functions over A of arity $n \geq 0$.

Let Σ be a signature. A Σ-*algebra* A is an algebra within a function for each function symbol in Σ with the same arity. Such a correspondence is called an *interpretation*. The notion of "interpretation" extends to closed terms as expected and for $t \in C(\Sigma)$, $[\![t]\!]$ denotes the interpretation of t in A. Interpretation extends also to open terms by universally quantifying the variables.

Let $L = (\Sigma, E)$ be an inequational specification. Let A be a Σ-algebra with A being the set of elements. Let $\preceq$ be a preorder on A that preserves all functions in A, i.e., $\preceq$ is a precongruence on A. E is a *sound inequality axiomatization with respect to* $\preceq$ of A if for all $s, t \in O(\Sigma)$,

$$E \vdash s \leq t \implies [\![s]\!] \preceq [\![t]\!].$$

Moreover, if for all closed terms $s, t \in C(\Sigma)$

$$E \vdash s \leq t \iff [\![s]\!] \preceq [\![t]\!]$$

E is called a *complete inequality axiomatization with respect to* $\preceq$ for A. ■

Example 2.5 It is not difficult to see that $\subsetneq_{\to}$ is a precongruence for PA. In order to prove that PA$^{\leq}$ is a sound inequality axiomatization with respect to the $\subsetneq_{\to}$ model induced by the PA term deduction system, it is enough to prove that for every axiom $s \leq t$ of PA$^{\leq}$ with free variables in V, the relation

$$S = \{(\sigma(s), \sigma(t)) | \sigma \text{ substitutes variables in V to closed terms}\} \cup Id$$

is a simulation.

Moreover, BPA$^{\leq}$ is a complete inequality axiomatization with respect to the $\subsetneq_{\to}$ model induced by the BPA term deduction system. The proof of this is quite similar to the proof of completeness of BPA with respect to bisimulation (see [BW90]). ∎

3 The conservative extension theorems

This section is devoted to states several results of conservative extension. The general conservative extension theorem of [Ver94b] is extended to deal with preorders and inequational specifications.

Definition 3.1 Let Σ_0 and Σ_1 be two signatures. If for all $f \in \Sigma_0 \cap \Sigma_1$ the arity of f in Σ_0 is the same as the arity of f in Σ_1 then $\Sigma_0 \oplus \Sigma_1$, called the *sum* of Σ_0 and Σ_1, is defined as the signature $\Sigma_0 \cup \Sigma_1$. Note that $\oplus$ is not simply the union of two signatures since sometimes it is not defined. ∎

Definition 3.2 Let $T_i = (\Sigma_i, D_i)$ be term deduction systems with predicate and relation symbols in T_p^i and T_r^i respectively ($i = 0, 1$). Let $\Sigma_0 \oplus \Sigma_1$ be defined and let $T_p^0 \cap T_r^1 = T_r^0 \cap T_p^1 = \emptyset$. Then $T_0 \oplus T_1$, called the *sum* of T_0 and T_1, is the term deduction system $(\Sigma_0 \oplus \Sigma_1, D_0 \cup D_1)$ with predicate and relation symbols $T_p^0 \cup T_p^1$ and $T_r^0 \cup T_r^1$. ∎

Example 3.1 $\Sigma_{\mathrm{BPA}} \oplus \Sigma_{\mathrm{MRG}}$ is defined and equals to the signature of PA which is the same as Σ_{MRG}. Moreover, the term deduction system PA equals to BPA $\oplus$ MRG. ∎

Operational conservativity

This paragraph defines the notions of operational conservative extension and operational conservative extension up to some preorder which is defined in terms of predicate and relation symbols.

The following definition is adapted from [BG91] and based on [FV95] which is given below with the name of *weak operational conservative extension*.

Definition 3.3 (Operational conservative extension) Let $T_i = (\Sigma_i, D_i)$ be term deduction systems with $T = (\Sigma, D) = T_0 \oplus T_1$ defined and let $D = D(T_p, T_r)$. The term deduction system T is an *operational conservative extension* of T_0 if it has $S(T)$ as unique well supported model and $S(T_0) = \{Ps, sRt \in S(T) | s \in C(\Sigma_0), P \in T_p, R \in T_r\}$ is the unique well supported model of T_0. ∎

Definition 3.4 (Weak operational conservative extension) Let $T_i = (\Sigma_i, D_i)$ be term deduction systems with $T = (\Sigma, D) = T_0 \oplus T_1$ defined and let $D = D(T_p, T_r)$. The term deduction system T is a *weak operational conservative extension* of T_0 if for each well supported model X of T the set $\{Ps, sRt \in X | s \in C(\Sigma_0), P \in T_p, R \in T_r\}$ is a well supported model of T_0. ∎

Weak operational conservative extension cannot be used in the context of this study since it does not consider unique model. The problem of multiple models is that a preorder defined in terms of predicate and relation symbols may relate different closed terms in each model and this fact can introduce inconsistency of axiomatizations.

The following definition and the next theorem originate in this article. They are the generalization for preorders of the case of operational conservativity up to an equivalence given by *Verhoef* [Ver94b].

Definition 3.5 (Operational conservative extension up to a semantical preorder) Let $T_i = (\Sigma_i, D_i)$ be term deduction systems and let $T = (\Sigma, D) = T_0 \oplus T_1$ defined. Let ξ be a semantic preorder defined in terms of predicate and relation symbols only. T is an *operational conservative extension of* T_0 *up to* ξ *preorder* if for all $s, t \in C(\Sigma_0)$, $s \preceq_\xi^\oplus t \iff s \preceq_\xi^0 t$. ■

Theorem 3.1 (Conservation of operational conservativity) *Let $T_i = (\Sigma_i, D_i)$ be term deduction systems and let $T = (\Sigma, D) = T_0 \oplus T_1$ defined. If T is an operational conservative extension of T_0, then it is also an operational conservative extension up to ξ preorder, for any preorder ξ defined in terms of predicate and relation symbols only.*

Proof. *(Sketch)* Let $s, t \in C(\Sigma_0)$. Since T is an operational conservative extension of T_0, the state-transition diagrams (or better: the term-relation-predicate diagrams) of s in both T and T_0 are the same, and so are the term-relation-predicate diagrams of t. Let ξ be a preorder defined in terms of relation and predicate symbols. Because $\preceq_\xi^\oplus$ is defined in the same way for relation and predicate symbols in T_0 as $\preceq_\xi^0$, and the term-relation-predicate diagrams of s and t are the same in both term deduction systems, $s \preceq_\xi^0 t$ implies $s \preceq_\xi^\oplus t$. The counterpositive is analogously proved. ■

Groote & Vaandrager [GV92] gave a first theorem for operational conservative extension in positive transition system specifications. *Bol & Groote* introduced in [BG91] a set of conditions that ensure conservativity in transition system specifications with negative premises. *Verhoef* did the same for stratifiables term deduction systems in [Ver94b]. The next conservative extension theorem was introduced by *Fokkink & Verhoef* in [FV95].

Theorem 3.2 (Weak operational conservativity) *Let $T_0 = (\Sigma_0, D_0)$ and $T_1 = (\Sigma_1, D_1)$ be two term deduction system satisfying:*

1. *T_0 is source dependent, and*
2. *if there is a conclusion sRs' or Ps of a rule $d \in D_1$ with $s = x$ or $s = f(x_1, ..., x_n)$ for an $f \in \Sigma_0$, then, there is a hypotheses of d which has the form $P't$ or $tR'u$ where $P' \notin T_p^0$, $R' \notin T_r^0$ or $u \notin O(\Sigma_0)$, $t \in O(\Sigma_0)$ and $\mathrm{var}(t) \subseteq SV(d)$.*

If $T_0 \oplus T_1$ is defined then it is a weak operational conservative extension of T_0.

Proof. [FV95] ■

As a corollary of this theorem, yet another operational conservative extension theorem is introduced.

Theorem 3.3 (Operational conservativity) *Let $T_0 = (\Sigma_0, D_0)$ and $T_1 = (\Sigma_1, D_1)$ be two term deduction systems satisfying statements 1. and 2. of Theorem 3.2 and, in addition,*

3. T_0 has unique well supported model.

If $T_0 \oplus T_1$ is defined and it has unique well supported model, $T_0 \oplus T_1$ is an operational conservative extension of T_0.

Proof. Immediate from Definition 3.3 and Theorem 3.2 ∎

The theorem is somehow more general than the one of [BG91]. [BG91] requires that for every rule $\frac{H}{tRu} \in D_1$, t is not in $O(\Sigma_0)$, and moreover, $T_0 \oplus T_1$ should be positive after reduction. Theorem 3.3 is more relaxed about the form of the new rules, and, in addition, a term deduction system may be not reducible to a positive one but may have a unique well supported model (see [Gla95]). However, the statement *3* cannot be omitted while no analogous one is required in [BG91]. I decided to pay this cost for two reasons: first, conditions *1* and *2* are quite more general than that proposed by [BG91], and second, in the context proposed by this article, one already knows whether condition *3* holds.

Example 3.2 It is easy to see that BPA and MRG satisfy the conditions of theorem 3.3. Thus PA is an operational conservative extension of BPA. Moreover, because of theorem 3.1, PA is an operational conservative extension up to simulation preorder. ∎

Inequational conservativity

This last paragraph states the general conservative extension for inequational specifications.

Definition 3.6 Let $L_i = (\Sigma_i, E_i)$ be inequational specifications $(i = 0, 1)$. Let $\Sigma_0 \oplus \Sigma_1$ be defined. $L_0 \oplus L_1$ is the *sum* of L_0 and L_1 defined as the inequational specification $(\Sigma_0 \oplus \Sigma_1, E_0 \cup E_1)$. ∎

Example 3.3 Notice that $\mathrm{BPA}^{\leq} \oplus \mathrm{MRG}^{\leq}$ forms $\mathrm{PA}^{\leq}$. ∎

Definition 3.7 (Inequational conservative extension) Let $L_i = (\Sigma_i, E_i)$ be inequational specifications $(i = 0, 1)$ and let $L = (\Sigma, E) = L_0 \oplus L_1$ be defined. L is an *inequational conservative extension* of L_0 if for all $s, t \in C(\Sigma_0)$

$$E \vdash s \leq t \iff E_0 \vdash s \leq t.$$

If for all $s \in C(\Sigma)$ there is a $t \in C(\Sigma_0)$ such that $E \vdash s = t$, then L has the *elimination property*. ∎

Theorem 3.4 (The general conservative extension theorem) *Let $L_i = (\Sigma_i, E_i)$ be inequational specifications and let $L = (\Sigma, E) = L_0 \oplus L_1$ be defined. Let $T_i = (\Sigma_i, D_i)$ be term deduction systems and let $T = (\Sigma, D) = T_0 \oplus T_1$ be defined. Let ξ be a preorder definable exclusively in terms of predicate and relation symbols. Let E_0 be a complete inequality axiomatization with respect to the ξ preorder model induced by T_0 and let E be a sound inequality axiomatization with respect to the ξ preorder model induced by T. If T is an operational conservative extension up to ξ preorder of T_0, then L is an inequational conservative extension of L_0.*

Moreover, if L has the elimination property, E is a complete inequality axiomatization with respect to the ξ preorder model induced by T.

Proof. The proof that for all $s, t \in C(\Sigma_0)$, $E_0 \vdash s \leq t \implies E \vdash s \leq t$ is trivial. Now, let $s, t \in C(\Sigma_0)$ and suppose $E \vdash s \leq t$. Since E is sound, $s \preceq^{\oplus}_{\xi} t$. Because T is an operational conservative extension up to ξ preorder of T_0, $s \preceq^{0}_{\xi} t$. Finally, $E_0 \vdash s \leq t$ since E is complete with respect to ξ. So E is an inequational conservative extension of E_0

Now, suppose moreover that L has the elimination property. Let $s, t \in C(\Sigma)$ such that $s \preceq^{\oplus}_{\xi} t$. Then, there are $s', t' \in C(\Sigma_0)$ such that $E \vdash s = s'$ and $E \vdash t = t'$. Since E is sound, $s' \asymp^{\oplus}_{\xi} s \preceq^{\oplus}_{\xi} t \asymp^{\oplus}_{\xi} t'$, where $\asymp^{\oplus}_{\xi}$ stands for $\preceq^{\oplus}_{\xi} \cap \succeq^{\oplus}_{\xi}$. So, it is enough to prove that $E \vdash s' \leq t'$, but T is an operational conservative extension of T_0 up to ξ preorder, so $s' \preceq^{0}_{\xi} t'$ and, because E_0 is complete, $E_0 \vdash s' \leq t'$ which trivially implies $E \vdash s' \leq t'$. ■

Example 3.4 Example 2.5 states that $\mathrm{BPA}^{\leq}$ is complete with respect to $\subsetneq$ for BPA and $\mathrm{PA}^{\leq}$ is sound with respect to $\subsetneq$ for PA. Since, in addition, PA is an operational conservative extension up to $\subsetneq$ of BPA, $\mathrm{PA}^{\leq}$ is an inequational conservative extension of $\mathrm{BPA}^{\leq}$. Moreover, because $\mathrm{PA}^{\leq}$ has the elimination property (see [BW90]), it is a complete inequality axiomatization with respect to $\subsetneq$ of PA. ■

4 Further remarks

Applications

Voorhoeve & Basten introduced in [VB95] a preorder for unstable nondeterminism. They dealt with a set of *autonomous actions* which can be regarded as observable actions that somehow behaves as the silent step. Several algebras were defined there. $\mathrm{BPA}_{\delta}aa^{\leq}$ is the basic process algebra with deadlock and autonomous actions. They used results in this article to extend $\mathrm{BPA}_{\delta}aa^{\leq}$ with the parallel operator, obtaining thus $\mathrm{ACP}aa^{\leq}$. Moreover, since $\mathrm{ACP}aa^{\leq}$ has the elimination property, completeness was proved using results introduced below. In addition, they added the binary Kleene star [BBP94] to both theories. Since $\mathrm{BPA}^{\star}_{\delta}aa^{\leq}$ and $\mathrm{ACP}^{\star}aa^{\leq}$ are sound, and the respective term deduction systems satisfy the conditions of Theorem 3.3, operational and inequational conservative extension can be also proved. Figure 1 shows this overview. There, an arrow A $\longrightarrow$ B means that A is both an operational conservative extension and an inequational conservative extension of B, and that it can be shown by using results in this article.

$$\begin{tikzcd}
\mathrm{BPA}^{\star}_{\delta}aa^{\leq} \arrow[d] & \mathrm{ACP}^{\star}aa^{\leq} \arrow[dl] \arrow[d] \\
\mathrm{BPA}_{\delta}aa^{\leq} & \mathrm{ACP}aa^{\leq} \arrow[l]
\end{tikzcd}$$

Figure 1: Conservative extension in algebras for Voorhoeve & Basten's preorder

Perhaps, the reader expected the arrow $\mathrm{BPA}^{\star}_{\delta}aa^{\leq} \longrightarrow \mathrm{ACP}^{\star}aa^{\leq}$. In this case only operational conservative extension can be proved using results in this articles (and so operational conservative extension up to the preorder). Since $\mathrm{BPA}^{\star}_{\delta}aa^{\leq}$ is not complete (see [Sew93, VB95]) Theorem 3.4 cannot be used.

Walker introduced in [Wal90] a complete (but non finite) axiomatization for a preorder that extends τ bisimulation with divergence. Now, I sketch some proofs of conservativity and completeness using the results in this article. Let ST be the algebra of synchronization trees with Milner's τ laws [Mil89]. The signature of ST has prefixing operators, the alternative composition and the *nil* process. Let CCS be the well known calculus of Milner [Mil89] that extends ST with renaming, restriction and parallel composition, and the expansion laws. Let $\mathrm{ST}_\perp$ and $\mathrm{CCS}_\perp$ the respective extensions of ST and CCS including the divergence operator with the laws for divergence given in [Wal90]. It is worthy to remark that for all CCS term Walker's preorder agrees with rooted τ bisimulation [Wal90].

Again, by looking at the term deduction systems and knowing that all the theories are sound and particularly ST and $\mathrm{ST}_\perp$ are complete, theorems of this article can be applied and, by interpreting arrows as before, Figure 2 is obtained as a result. In addition, since ST is

$$\begin{tikzcd}
\mathrm{ST}_\perp \arrow[d, "\bullet"'] & \mathrm{CCS}_\perp \arrow[l] \arrow[dl, "\bullet"'] \arrow[d, "\bullet"'] \\
\mathrm{ST} & \mathrm{CCS} \arrow[l]
\end{tikzcd}$$

Figure 2: Conservative extension in algebras for Walker's preorder

complete for the preorder, CCS is also complete because the new operators can be eliminated. Similarly, $\mathrm{CCS}_\perp$ is complete since $\mathrm{ST}_\perp$ is complete and $\mathrm{CCS}_\perp$ has the elimination property. Nevertheless, nor $\mathrm{ST}_\perp$ neither $\mathrm{CCS}_\perp$ have the elimination property with respect to ST or CCS.

Moreover, it deserves to notice that the results labelled with a $\bullet$ are new in this article.

Conclusions

This article extended the general conservative results of [Ver94b] with respect to preorders for transition system based process theories with inequalities. It only required reasonable and easy-to-check conditions. As a simple corollary, a general completeness theorem for inequational specifications was proved.

As it was explained above, the results of this article were already applied in [VB95]. Besides, an example was taken from the literature and results of conservativity and completeness were recreated. In addition, some new results on these examples were quickly proved by means of the techniques introduced here.

The use of preorders and inequational specifications is not so widely diffused as equivalences and equational specifications; perhaps it is due to the fact that they are more difficult to manage. However, results presented here above seem to make it easier.

References

[Abr87] Samson Abramsky. Observation equivalence as a testing equivalence. *Theoretical Computer Science*, 53:225–241, 1987.

[Ace91] Luca Aceto. On relating concurrency and nondeterminism. In S. Brookes, M. Main, A. Melton, M. Mislove, and D. Schmidt, editors, *Proceedings of Mathematical Foundations of Programming Semantics, 7th. International Conference*, pages 376–402, Pittsburgh, 1991. LNCS 598, Springer-Verlag.

[BBP94] J.A. Bergstra, I. Bethke, and A. Ponse. Process algebra with iteration and nesting. *The Computer Journal*, 37(4):241–258, 1994.

[BC88] G. Boudol and I. Castellani. Concurrency and atomicity. *Theoretical Computer Science*, 59:25–84, 1988.

[BG91] R.N. Bol and J.F. Groote. The meaning of negative premises in transition system specifications. In J. Leach Albert, B. Monien, and M. Rodríguez Artalejo, editors, *Proceedings 18th. ICALP*, pages 481–494, Madrid, 1991. LNCS 510, Springer-Verlag.

[BIM88] B. Bloom, S. Istrail, and A.R. Meyer. Bisimulation can't be traced: preliminary report. In *Conference Record of the 15th ACM Symposium on Principles of Programming Languages*, pages 229–239, California, San Diego, 1988.

[BK84] J.A. Bergstra and J.W. Klop. Process algebra for synchronous communication. *Information and Computation*, 60:109–137, 1984.

[BV93] J.C.M. Baeten and C. Verhoef. A congruence theorem for structured operational semantics with predicates. In E. Best, editor, *Proceedings of CONCUR'93*, pages 477–492, Hildesheim, 1993. LNCS 715, Springer-Verlag.

[BW90] J.C.M. Baeten and W.P. Weijland. *Process algebra*. Cambridge University Press, 1990.

[Cas93] I. Castellani. Observing distribution in processes. In *Proceedings of MFCS 93*. LNCS 711, Springer-Verlag, 1993.

[FV95] Wan Fokkink and Chris Verhoef. Conservative extension in operational semantics with substitutions, 1995. In preparation.

[Gla90] R.J. van Glabbeek. The linear time - branching time spectrum. In J.C.M. Baeten and J.W. Klop, editors, *Proceedings of CONCUR'90*, pages 278–297, Amsterdam, 1990. LNCS 458, Springer-Verlag.

[Gla93a] R.J. van Glabbeek. Full abstraction and expressiveness in structural operational semantics. In M. Nivat, C. Rattray, T. Rus, and G. Scollo, editors, *Proceedings of the 3rd. AMAST Conference*, pages 77–84, Twente, 1993. Workshops in Computing, Springer-Verlag.

[Gla93b] R.J. van Glabbeek. The linear time - branching time spectrum II. (extended abstract). In E. Best, editor, *Proceedings of CONCUR'93*, pages 308–323, Hildesheim, 1993. LNCS 715, Springer-Verlag.

[Gla95] R.J. van Glabbeek. The meaning of negative premises in transition system specifications II. Technical Note 95-16, Computer Science Department, Stanford University, 1995.

[Gro93] J.F. Groote. Transition system specifications with negative premises. *Theoretical Computer Science*, 118(2):263–299, 1993.

[GV92] J.F. Groote and F.W. Vaandrager. Structured operational semantics and bisimulation as a congruence. *Information and Computation*, 100(2):202–260, 1992.

[Mil89] Robin Milner. *Communication and concurrency*. Prentice Hall, 1989.

[NP84] P. Nielsen and Thiagarajan P.S. Degrees of non-determinism and concurrency: a petri net view. In M. Joseph and R. Shyamasundar, editors, *Proceedings FST & TCS 5*, pages 89–118. LNCS 181, Springer-Verlag, 1984.

[Pom86] L. Pomello. Some equivalences notions for concurrent systems. An overview. In G. Rozemberg, editor, *Proceedings of Advances in Petri Nets 1985*, pages 381–400. LNCS 222, Springer-Verlag, 1986.

[Sew93] P. Sewell. Bisimulation is not (first order) equationally axiomatisable. Technical report, Department of Computer Science, Edinburgh University, 1993. To appear in LICS'94.

[VB95] M. Voorhoeve and T. Basten. Process algebra with autonomous actions. In A. Ponse, C. Verhoef, and S.F.M. van Vlijmen, editors, *Proceedings of the 2nd. Workshop ACP95*, Eindhoven, 1995. (These proceedings).

[Ver94a] C. Verhoef. A congruence theorem for structured operational semantics with predicates and negative premises. In B. Jhonsson and J. Parrow, editors, *Proceedings of CONCUR'94*, pages 433–448, Uppsala, 1994. LNCS 836, Springer-Verlag.

[Ver94b] C. Verhoef. A general conservative extension theorem in process algebra. In E.-R. Olderog, editor, *Proceedings of PROCOMET'94, IFIP 2 Working Conference*, pages 149–168, San Miniato, 1994. North-Holland.

[Wal90] D.J. Walker. Bisimulation and divergence. *Information and Computation*, 85:202–241, 1990.

[Yan93] Daniel Yankelevich. *Parametric views of process description languages.* PhD thesis, University of Pisa, 1993.