Replace this file with `prentcsmacro.sty` for your meeting,
or with `entcsmacro.sty` for your meeting. Both can be
found at the ENTCS Macro Home Page.

On The Unification of Process Semantics: Operational Semantics

David de Frutos Escrig[1,3] Carlos Gregorio Rodríguez [2,4]
Miguel Palomino [1,5]

Departamento de Sistemas Informáticos y Computación
Universidad Complutense de Madrid
Madrid, Spain

Abstract

The complexity of parallel systems has produced a large collection of semantics for processes, a classification of which is provided by Van Glabbeek's linear time-branching time spectrum; however, no suitable unified definitions were available. After several years of studying these semantics, looking for homogeneous presentations like those we have provided using (bi)simulations up-to, we have discovered the way to unify all of them, first in an observational and in an equational framework, and now also in an operational way. We have shown that all the semantics in the spectrum are governed by a simulation part in which we make some additional identifications to obtain a linear semantics. As a consequence of these identifications, it is not clear how to obtain an operational semantics which characterizes any of those semantics in the usual coinductive manner. We have found a way to obtain these operational semantics by moderately enlarging the original semantics with the introduction of additional transitions which correspond to the application of the axiom that makes the extra identifications, but only at the roots of the corresponding processes. Thus we obtain a characterization of the orders defining any of the semantics in the spectrum by means of the corresponding constrained simulation order. This would be, for instance, ready simulation for failures or readiness semantics, plain simulation for traces, complete simulation for complete traces, or trace simulation for possible or impossible futures. It is interesting to observe that such a characterization by means of bisimulations would not be possible, which shows us that mutual similarity can be a more flexible coinductive way to capture equivalences. We conclude with a simple application to illustrate the interest of our operational characterizations.

Keywords: processes, semantic preorders, linear time-branching time spectrum.

1 Introduction

The classic linear semantics of processes (e.g. traces, failures, readiness, ...) are defined by means of decorated traces. Although these could be obtained from the computations defined by their ordinary operational semantics, the semantics of each process has to be *calculated* as a whole, by collecting together all its traces, so that no

[1] Partially supported by the MCyT project DESAFIOS TIN2006-15660-C02-02 and the project PROMESAS-CAM S-0505/TIC/0407.

[2] Partially supported by the MCyT project WEST TIN2006-15578-C02.

[3] Email:defrutos@sip.ucm.es

[4] Email:cgr@sip.ucm.es

[5] Email:miguelpt@sip.ucm.es

©2008 Published by Elsevier Science B. V.

simple coinductive characterization in terms of that original operational semantics is possible.

Instead, the classic way of defining an (abstract) operational semantics for processes, based on their transition systems, is by means of bisimulations. This immediately gives us a coinductive characterization of the corresponding semantic equivalence. An alternative way to obtain these coinductive characterizations is by means of simulations, or more in general constrained simulations [dFG08] that allow us to characterize the corresponding semantic preorder instead of the generated equivalence.

Some of the most recent work of our group has been devoted to looking for coinductive characterizations of semantics. In [dFG05] we characterised process equivalences by using bisimulations up-to and in [dFG07] we defined simulations up-to and we showed how to use them to characterize process preorders, including those in the linear time-branching time spectrum [Gla01]. In the second paper, simulations up-to were used to achieve a uniform treatment of different semantics. With their help we have proved some general results for a wide class of process semantics, such as the existence of a canonical preorder defining a given process equivalence, for which we have provided a correct and complete axiomatization obtained from that of the corresponding equivalence.

The existence of such an indirect coinductive characterization for, in particular, the linear semantics of processes begs the question of whether it is posible to obtain the preorders defining them directly as a constrained simulation order corresponding to an alternative operational description of processes.

That is the main goal of the present paper: to define for each of the linear semantics (traces, complete traces, failures, readines, failure traces, ready traces, and so on) an structured operational semantics over the syntactic terms of a simple process algebra such as BCCSP, generating a labelled transition system in such a way that the corresponding order can be characterized by the adequate simulation order that transition system. As a consequence of our unification of the axiomatic semantics of processes, those SOS exist and besides can be defined in a quite similar manner for all the linear semantics in the spectrum.

It is true that already fifteen years ago Cleaveland and Hennessy [CH93] presented their characterization of testing semantics as a bisimulation semantics. This was indeed an important first step in the same direction. Even if we will present later a more detailed comparison between their work and ours, let us advance here that the main virtue of our approach is its genericity, covering all the linear semantics in an uniform way. This has been obtained as a final result of our unification work [dFGP08a,dFGP08b], covering all the semantics for processes in van Glabbeek's spectrum [Gla01]. Another important property of the operational semantics that we have obtained is that they can be presented as SOS, what is related to the fact that they can be defined in a "local" way from the original operational semantics of processes. Instead, the transition system in [CH93] has to be defined as a whole since their states and transitions are derived from the acceptation trees defining the (denotational model of the) testing semantics, far removed from the original operational semantics.

The rest of the paper is structures as follows. In Section 2 we introduce some simple technical preliminaries. In Section 3 we recall our previous results on the coinductive characterization of semantics by means of simulations up-to and the generic axiomatizations of all the semantics in the spectrum that, combined, allow us to obtain a new characterization in terms of so-called local simulations up-to. Next we present in Section 4 our operational characterization of the semantics coarser than ready simulation, that is, of the diamond formed by failures, readiness, failure traces, and ready traces semantics. These are probably the most important linear semantics in the spectrum and by focusing on the case of ready simulation, which corresponds to our I-constrained simulations, we expect to facilitate the presentation and understanding of our main results here. Section 5 contains the extension to other constraints and a brief discussion on trace semantics. Finally, we present an application in Section 6 and some conclusions.

2 Preliminaries

Labelled transition systems, or LTS for short, introduced by Plotkin twenty five years ago (reprinted in [Plo04]), are the usual way to describe the behaviour of processes in an operational way.

Definition 2.1 *A labelled transition system is an structure $\mathcal{T} = (\mathcal{P}, Act, \rightsquigarrow)$ where $\mathcal{P}$ is a set of processes, agents or states; Act is a set of actions; and $\rightsquigarrow \subseteq \mathcal{P} \times Act \times \mathcal{P}$ is a transition relation. A rooted LTS is a pair $(\mathcal{T}, p_0)$ with $p_0 \in \mathcal{P}$.*

The set Act denotes the alphabet of actions that processes can perform and the relation $\rightsquigarrow$ describes the process transitions after the execution of actions. Any triple $\langle p, a, q\rangle$ in the transition relation $\rightsquigarrow$ is represented by $p \overset{a}{\rightsquigarrow} q$, indicating that process p performs action a and evolves into process q. A rooted LTS describes the semantics of a concrete process: that corresponding to its initial state p_0.

Some usual notations on LTSs are used along the paper. We write $p \overset{a}{\rightsquigarrow}$ if there exists a process q such that $p \overset{a}{\rightsquigarrow} q$. The function $I^{\rightsquigarrow}$ calculates the set of initial actions of a process with respect to a given LTS, $I^{\rightsquigarrow}(p) = \{a \mid a \in Act \text{ and } p \overset{a}{\rightsquigarrow}\}$; we omit the superscript $\rightsquigarrow$ if no confusion arises.

LTS's for finite processes are directed graphs which become finite trees[6] if expanded. These finite trees can be syntactically described by the basic process algebra BCCSP, which was also used, for instance, in [Gla01,dFG05].

Definition 2.2 *Given a set of actions Act, the set of BCCSP processes is defined by the following BNF-grammar:*

$$p ::= \mathbf{0} \mid ap \mid p + q$$

where $a \in Act$. $\mathbf{0}$ represents the process that performs no action; for every action in Act, there is a prefix operator; and $+$ is a choice operator.

In the rest of the paper we mix finite processes, corresponding to BCCSP terms, and arbitrary transition systems. Thanks to the fact that all the semantics that we consider and the characterizations we use are continuous, as proved for instance in

[6] We obtain directly a tree if we generate the states on the fly introducing a new state for each transition generated by the application of the rules defining the operational semantics, see for instance [Mil89].

[dFG05], we can indeed prove all our results only for BCCSP processes and then apply them to arbitrary processes.

Definition 2.3 *The operational semantics for the BCCSP terms is given by the LTS* $(\mathcal{P}, Act, \longrightarrow)$ *where the transition relation is defined by the rules in Figure 1.*

$$ap \xrightarrow{a} p \qquad \frac{p \xrightarrow{a} p'}{p+q \xrightarrow{a} p'} \qquad \frac{q \xrightarrow{a} q'}{p+q \xrightarrow{a} q'}$$

Fig. 1. Operational Semantics for BCCSP Terms

Trailing occurrences of the constant $\mathbf{0}$ are omitted: we write a instead of $a\mathbf{0}$. As usual (see for instance [Gla01]), since the operational semantics of choice defines it as a commutative and associative operator, and any other semantics in which we are interested is based on that, we can use the n-ary choice operator $\sum$ to write any process as $\sum_a \sum_i ap_a^i$. This corresponds to the transition tree of each process, and the fact that we use sets as indices makes that operator commutative and associative by definition.

A process aq' is an a-summand of the process q if and only if $q \xrightarrow{a} q'$. We define $p|_a$ as the (sub)process we get by adding all the a-summands of p. That is, if $p = \sum_a \sum_i ap_a^i$, then $p|_a = \sum_i ap_a^i$.

Preorders are reflexive and transitive relations that we represent by $\sqsubseteq$. For the sake of simplicity, we use the symbol $\sqsupseteq$ to represent the preorder relation $\sqsubseteq^{-1}$. Every preorder induces an equivalence relation that we denote by $\equiv$, that is $p \equiv q$ if and only if $p \sqsubseteq q$ and $q \sqsubseteq p$.

Definition 2.4 *A preorder relation* $\sqsubseteq$ *over processes is a* behaviour preorder *if*

- *it is weaker than the bisimulation equivalence, i.e.* $p \equiv_B q \Rightarrow p \sqsubseteq q$,
- *and it is a precongruence with respect to the prefix and choice operators, i.e. if* $p \sqsubseteq q$ *then* $ap \sqsubseteq aq$ *and* $p + r \sqsubseteq q + r$.

3 Local simulations up-to

In [dFG07,dFG08] starting from I-simulations, which are just the classic ready simulations, we have generalized them to obtain arbitrary constrained N-simulations. In order to characterize all the reasonable behavior preorders in a coinductive way we have introduced N-simulations up-to an order $\sqsubseteq$, which are defined as follows:

Definition 3.1 *Let* $\sqsubseteq$ *be a behaviour preorder, and* N *a relation over processes. We say that a binary relation* S *over processes is an* N-simulation up-to $\sqsubseteq$ *if* $S \subseteq N$ *(that is,* $pSq \Rightarrow pNq$*) and* S *is a simulation up-to* $\sqsubseteq$*. Or equivalently, in a coinductive way, whenever we have* pSq *we also have:*

- *For every* a*, if* $p \xrightarrow{a} p'_a$ *there exist* q', q'_a *such that* $q \sqsupseteq q' \xrightarrow{a} q'_a$ *and* $p'_a S q'_a$*;*
- pNq.

We say that process p *is* N*-simulated up-to* $\sqsubseteq$ *by process* q*, or that process* q N*-simulates process* p *up-to* $\sqsubseteq$*, written* $p \precsim_{\sqsubseteq}^{N} q$*, if there exists an* N*-simulation up-to*

$\sqsubseteq$, S, such that pSq.

We often just write $\sqsubseteq^N_\sim$, instead of $\sqsubseteq^N_{\sim\sqsubseteq}$, when the behaviour preorder is clear from the context.

We proved in [dFG07] that all the preorders defining the semantics in the linear time-branching time spectrum can be characterised as N-simulations up-to the corresponding equivalence relation $\equiv$, where N is the constraint defining the coarsest simulation semantics finer than the given semantics. For instance, the result for the semantics between failures semantics and ready simulation was the following:

Theorem 3.2 ([dFG07]) *For every behaviour preorder $\sqsubseteq$ verifying the axiom (RS) and $\sqsubseteq \subseteq I$, we have $p \sqsubseteq q$ if and only if $p \sqsubseteq^I_{\sim\sqsubseteq} q$.*

Table 1 shows the constraints defining the adequate constrained simulation order finer than each of the semantics in the linear time-branching time spectrum.

	T	S	CT	CS	F	R	FT	RT	PW	RS	PF	$2N$
$C_{\mathcal{O}}$	U	U	C	C	I	I	I	I	I	I	W	Z

$$pUq \iff true \qquad\qquad pWq \iff p \equiv_T q$$
$$pCq \iff (p = \mathbf{0} \Leftrightarrow q = \mathbf{0}) \qquad\qquad pZq \iff p \equiv_S q$$
$$pIq \iff I(p) = I(q)$$

Table 1
Constraints for the Semantics in the ltbt Spectrum

$$
\begin{array}{ll}
(B_1) & x + y \simeq y + x \\
(B_2) & (x + y) + z \simeq x + (y + z) \\
(B_3) & x + x \simeq x \\
(B_4) & x + \mathbf{0} \simeq x
\end{array}
$$

Fig. 2. Axiomatization of the Bisimulation Equivalence

In addition, we have also recently proved in [dFGP08b] that the linear semantics in the spectrum can be axiomatized by means of the set of axioms characterizing bisimulation equivalence (see Figure 2) plus the adequated instances of the axioms

$$
\begin{array}{lll}
(NS) & xNy & \Longrightarrow x \preceq x + y \\
(ND_{\equiv}) & M(x, y, w) & \Longrightarrow ax + a(x + y) + a(y + w) \simeq ax + a(y + w)
\end{array}
$$

For instance, for the four semantics in the diamond under ready simulations semantics (failures, readiness, failure traces and ready traces semantics) we have:

$$\begin{aligned}
M_F(x,y,w) &\iff \text{BCCSP}^3\\
M_R(x,y,w) &\iff I(x) \supseteq I(y)\\
M_{FT}(x,y,w) &\iff I(w) \subseteq I(y)\\
M_{RT}(x,y,w) &\iff I(x) = I(y) \text{ and } I(w) \subseteq I(y)
\end{aligned}$$

The proof of this result uses the adequate notion of head normal form which, roughly, are defined by applying repeatedly to any term p the axiom $(ND_{\equiv})$ from right to left, for as long as possible, thus adding to the set of summands of p some new summands. By definition of these hnf's we clearly have that from $\{B_1–B_4, (NS), (ND^X_{\equiv})\}$ we can infer $hnf^X(p) \simeq p$. Besides, whenever we have $p = \sum_a \sum_i ap^i_a$ and $p \sqsubseteq_x q$ for $hnf^X(q) = \sum_a \sum_j ah^j_a$, we have that for all i there exists j such that $p^i_a \sqsubseteq_x q^j_a$. This was the key result to complete the proof of completeness of our new axiomatizations, and also for introducing now the new notion of *local I-simulation up-to*.

Definition 3.3 *For $X \in \{F, R, FT, RT\}$ and $p = \sum_a \sum_i ap^i_a$, whenever we have a pair of indices i, j and a decomposition $p^j_a = r^j_a + s^j_a$ with $M_X(p^j_a, r^j_a, s^j_a)$ we say that p is 1-locally X-equivalent to $p + a(p^i_a + r^j_a)$, and we write $p \equiv^{l1}_X q$. We say simply that p and q are locally X-equivalent when they are related by the reflexive and transitive closure of $\equiv^{l1}_X$, and then we write $p \equiv^l_X q$.*

For $X \in \{F, R, FT, RT\}$ we call local I-simulations up-to $\equiv_X$ to the I-simulations up-to $\equiv^l_X$. We say that process p is locally I-simulated up-to $\equiv_X$ by process q, or that process q locally I-simulates process p up-to $\equiv_X$, written $p \sqsubseteq^I_{\sim\equiv^l_X} q$, if there exists a local I-simulation up-to $\equiv_X$ S, such that pSq.

Local I-simulations up-to are enough to characterize the four linear semantics in $\{F, R, FT, RT\}$.

Proposition 3.4 *For $X \in \{F, R, FT, RT\}$ we have $p \sqsubseteq_X q$ if and only if $p \sqsubseteq^I_{\sim\equiv^l_X} q$.*

Proof. The implication from right to left is an immediate consequence of Theorem 3.2. For the other, we observe that $\{(p, q) \mid p \sqsubseteq_X q\}$ is a local I-simulation up-to $\equiv_X$. Indeed, for any $p \xrightarrow{a} p^i_a$ we have $q \equiv^l_X hnf^X(q)$ and taking $hnf^X(q) = \sum_a \sum_i ah^j_a$ there exists some j such that $hnf^X(q) \xrightarrow{a} h^j_a$ and $p^i_a \sqsubseteq_X h^j_a$. □

Example 3.5 Let us consider the two processes $p = abc + abd$ and $q = a(bc + bd)$. We have $p \equiv_F q$ and we can check that $p \sqsubseteq^I_{\sim\equiv^l_F} q$ since $p \sqsubseteq_{RS} q$. In order to prove that also $q \sqsubseteq^I_{\sim\equiv^l_F} p$, we have to apply $\equiv^l_F$ to p to obtain $p \equiv^l_F p + q$ and then we have $q \sqsubseteq_{RS} p$.

By contrast, if we wanted to apply our bisimulation up-to characterization to prove directly $p \equiv_F$ we would have to turn q into $q + p$ to adequately simulate the transition $p \xrightarrow{a} bc$. This would correspond to the local application of $(ND^F_{\equiv})$

combined with that of

$$(RS_{\equiv}) \quad I(x) = I(y) \Longrightarrow a(x+y) \simeq a(x+y) + ax\,.$$

But if we replaced the action a by a larger prefix $a_1 \ldots a_n$ then we should also modify the process $q' = a_1 \ldots a_n(bc+bd)$ in a non-local way in order to obtain $q'' = q' + p'$, so that we could suitably simulate the transition $p' = a_1 \ldots a_n bc + a_1 \ldots a_n bd \xrightarrow{a_1} a_2 \ldots a_n bc$. Certainly, this is not necessary when checking $p' \equiv_F q'$ by means of local simulations up-to.

Note then that we cannot get a local notion of bisimulation up-to equivalent to our unrestricted notion of bisimulation up-to.

Therefore, the coinductive characterization of the semantics by means of simulations up-to has at least two important advantages over that using bisimulations up-to. First, we can characterize the orders defining the semantics and not just the induced equivalences; and second, we can use a local variant of the notion so that we only need to rely on the equivalence relation $\equiv^l_X$ for the up-to part.

4 SOS for the linear semantics of processes

In Section 3 we have introduced and proved some results that establish the framework from where to tackle our goal: to define an SOS over BCCSP terms in such a way that we can use constrained simulations to characterize the classic linear semantics. Let us consider for instance the failures preorder $\sqsubseteq_F$. We are going to define a new operational semantics for BCCSP terms $(\mathcal{P}, Act, \Rightarrow_F)$ such that $p \sqsubseteq_F q$ if and only if q ready simulates p in $(\mathcal{P}, Act, \Rightarrow_F)$.

As we said in the introduction, we will concentrate first on the diamond of linear semantics coarser than ready simulation. All these semantics are based on the observation of the initial set of actions of each process, that can be obtained by application of the SOS rules in Figure 3.

$$\frac{}{\mathbf{0} \longrightarrow_I \emptyset} \qquad \frac{}{ap \longrightarrow_I \{a\}} \qquad \frac{p \longrightarrow_I A, \ q \longrightarrow_I B}{p+q \longrightarrow_I A \cup B}$$

Fig. 3. Rules that compute the set of initial actions of a process

We want to stress the fact that although the rule for the sum of processes is a compositional one that has to look at the initials of its arguments, this does not mean that the observation of the initials is not local, since the sum of processes is an static operation. By contrast, for the prefix operation (that is dynamic) we have no premise to apply the rule. As a matter of fact, if we used the transition system style grammar with the n-ary sum of prefixed processes $\sum ap_a$, we would directly obtain $I(p)$ by collecting all the prefixes in the arguments of the sum.

The rules in Figure 4 define the transition relation $\Longrightarrow$ that will define the operational semantics to characterize each of the X-semantics. The transition relation $\longleftrightarrow_X$ is an auxiliary relation that captures the reiterated application of the axiom $(ND^X_{\equiv})$. Rules (RF) and (TR) define reflexivity and transitivity of the relation $\longleftrightarrow_X$. Finally, the rule (CL) combines the auxiliar relation $\longleftrightarrow_x$ and the original operational transition relation $\longrightarrow$ (see Definition 2.3), to define the new labelled transitions $\Longrightarrow_X$.

$$\text{(ND)}\ \frac{p \longrightarrow_I A_p \quad q \longrightarrow_I A_q \quad r \longrightarrow_I A_r \quad M_X(A_p, A_q, A_r)}{ap + a(q+r) + s \longleftrightarrow_X ap + a(q+r) + a(p+q) + s}$$

$$\text{(RF)}\ \frac{}{p \longleftrightarrow_X p} \qquad \text{(TR)}\ \frac{p \longleftrightarrow_X q \quad q \longleftrightarrow_X r}{p \longleftrightarrow_X r}$$

$$\text{(CL)}\ \frac{p \longleftrightarrow_X p' \quad p' \xrightarrow{a} q}{p \overset{a}{\Longrightarrow}_X q}$$

Fig. 4. Operational Semantics Characterizing the Linear Semantics

Definition 4.1 *For $X \in \{F, R, FT, RT\}$, the operational semantics for BCCSP terms is given by the LTS $(\mathcal{P}, Act, \Longrightarrow_X)$ where the transition relation $\Longrightarrow_X$ is defined by the rules in Figure 4.*

By abuse of notation, we have written $M_X(A_p, A_q, A_r)$ to express that we check $M_X(p, q, r)$ using the initials computed by $\longrightarrow_I$.

The relation $\Longrightarrow_X$ has some interesting properties. First, it is an extension of the original transition system.

Proposition 4.2 *For $X \in \{F, R, FT, RT\}$, p and q BCCSP processes, and α a sequence of actions in Act, we have that $p \xrightarrow{\alpha} q$ implies $p \overset{\alpha}{\Longrightarrow}_X q$.*

Although usually some new transitions appear, the set of initial actions of any process always remains the same.

Corollary 4.3 *For $X \in \{F, R, FT, RT\}$ and for any BCCSP process p, we have $I^{\rightarrow}(p) = I^{\Rightarrow_X}(p)$.*

It is also clear that, for any $X \in \{F, R, FT, RT\}$, the auxiliary relation $\longleftrightarrow_X$ preserves the equivalence $\equiv_X$ since the rule (ND) corresponds to the application of axiom (I-$ND^X_{\equiv}$), which is correct with respect to $\equiv^I_x$.

Proposition 4.4 *For $X \in \{F, R, FT, RT\}$ and BCCSP processes p and q, we have $p \longleftrightarrow_X q$ implies $p \equiv_X q$.*

Next we will prove the main theorem of this paper, that asserts that for each of the semantics in the diamond the corresponding operational semantics is defined as in Figure 4.

Theorem 4.5 *For $X \in \{F, R, FT, RT\}$ and BCCSP processes p and q, we have that*

$$p \sqsubseteq_X q \iff p \sqsubseteq^{\Rightarrow_X}_{RS} q\,.$$

Proof. We will apply our characterization of the orders $\sqsubseteq_X$ by means of local I-simulations up-to of Proposition 3.4; we will show that $p \sqsubseteq^{\Rightarrow_X}_{RS} q$ implies $p \sqsubseteq^I_{\sim_{\equiv^l_X}} q$. This is because any ready simulation over the transition system $\Longrightarrow_X$ is also a local I-simulation up-to $\equiv_X$. Indeed, if R is a ready simulation over the transition system

$\Longrightarrow_X$, if pRq, whenever we have $p \xrightarrow{a} p'$ we also have $p \overset{a}{\Longrightarrow}_X p'$, and therefore there is some $q \overset{a}{\Longrightarrow}_X q'$ with $p'Rq'$. By definition of the transition system $\Longrightarrow_X$, there is some process q'' such that $q \longleftrightarrow_X q''$ and $q'' \xrightarrow{a} q'$. Then we also have $q \equiv^l_X q''$, and thus R is indeed a local I-simulation up-to $\equiv_X$.

To prove that $p \sqsubseteq^I_{\sim_{\equiv^l_X}} q$ implies $p \sqsubseteq^{\Rightarrow_X}_{RS} q$, we will check that the relation $\sqsubseteq^N_{\sim_{\equiv^l_X}}$ is a ready simulation over the transition relation $\Longrightarrow_X$. If we have $p \sqsubseteq^N_{\sim_{\equiv^l_X}} q$, whenever $p \overset{a}{\Longrightarrow}_X p'$ we have some process p'' such that $p \longleftrightarrow_X p''$ and $p'' \xrightarrow{a} p'$. Then we also have $p \equiv_X p''$, and so $p'' \sqsubseteq^N_{\sim_{\equiv^l_X}} q$. From $p'' \xrightarrow{a} p'$ we now obtain that there are some $q \equiv^l_X q''$, $q'' \xrightarrow{a} q'$, and therefore we also have $q \longleftrightarrow_X q''$ concluding the proof. □

As a consequence of our negative results of Section 3, it is not possible to obtain an operational semantics locally defined from that which characterizes the linear semantics by means of bisimilarity. However, this has been done using mutual similarity.

Certainly, the fact that the characterizations in terms of bisimilarity cannot be defined in a local way is related to the fact that the transition systems generated by application of the algorithm in [CH93] will be larger than those generated by our local transformation here. Unfortunately, it is true that this does not magically lead (at least at the theoretical level) to more efficient algorithms to decide the equivalences with respect to the linear semantics, that are known to be quite hard to decide. Obviously, this is related to the fact that simulation is harder than bisimulation [KM02]. Even so, these are mainly theoretical bounds and it is nice to know that in practice we can apply a local transformation to generate the transition systems characterizing those semantics by means of the simulation orders, that in many real cases appearing in practice will be also not so difficult to decide.

5 Characterizing the semantics corresponding to other constraints

Let us start by considering the case of the universal constraint U. As discussed in [dFGP08b], if we also use U in the condition M_X it is clear that all the semantics in the corresponding diamond collapse into a single one: trace semantics.

We can immediately realize that the transition system to characterize it in terms of plain simulations is the same transition system $\Longrightarrow_F$ that we can use to characterize the failures semantics by means of ready simulations.

Theorem 5.1 *The trace preorder $\sqsubseteq_T$ coincides with the simulation order on the transition system $\Longrightarrow_F$, that is, $p \sqsubseteq_T q$ iff $p \sqsubseteq^{\Rightarrow_X}_S q$.*

Even if this coincidence is a simple fact that reflects the relation between traces and failures semantics, we think that such clear presentation will contribute to clarify it. We can now say in plain words that failures semantics is just traces semantics enriched by the observation of initials, so that the plain simulation order that implies the trace order becomes the ready simulation order.

For other finer observers such as T we can also characterize the corresponding semantic orders, such as possible and impossible futures, in terms of local simulations up-to. Therefore we can also use that result to justify that the corresponding transition systems $\Longrightarrow_X^T$ would characterize the semantics orders $\sqsubseteq_X^T$ in terms of T-simulations that preserve the set of traces of the simulated process. In this case the corresponding SOS definition has to include rules for the computation of the set of traces $T(p)$ and this cannot be done for infinite processes. But except for the computation of these sets, the rest of the rules for the generation of the corresponding transition systems $\Longrightarrow_X^T$ are also valid, their local character is still present, and the resemblance with the rule defining $\Longrightarrow_X$ (we just turn the conditions M_X into those using the observer T in place of I) will contribute once more to our unification work, continuing that it [dFGP08a,dFGP08b].

6 Applications: trace deterministic normal forms

As a simple application we present the example used by Klin in [Kli04], that we already used in [dFG05] to illustrate our coinductive characterization of the behavior preorders by means of our bisimulations up-to.

Definition 6.1 For any process $p = \sum_a \sum_i ap_a^i$ the *deterministic form* of p is defined as $Det(p) = \sum_a aDet(\sum_i p_a^i)$.

We wish to prove that p and $Det(p)$ are trace equivalent. We will do it by proving that they are simulation equivalent over the transition system $\Longrightarrow_F$.

Proposition 6.2 *For any process p we have $p \sqsubseteq_F Det(p)$.*

Proof. We will prove that $p \sqsubseteq_S^{\Rightarrow X} Det(p)$ by showing that $R = \{(p, Det(p+q)) \mid p, q \text{ processes}\}$ is a simulation for the transition system $\Longrightarrow_F$. For $q = \sum_a \sum_j aq_a^j$ we have $Det(p+q) = \sum_a Det(\sum_i p_a^i + \sum_j q_a^j)$. Then, for any $p \stackrel{a}{\Longrightarrow}_F p'$ we have $p = p_a^i + \sum_k r_a^k$, for some index i and $p_a^k = r_a^k + s_a^k$ a decomposition of any of the rest of the summands of p. We have $Det(p+q) \stackrel{a}{\longrightarrow} Det(\sum_i ap_a^i + \sum_j aq_a^j) = Det((p_a^i + \sum_k r_a^k) + (\sum_k r_a^k + \sum_j q_a^j))$, so that we also have $Det(p+q) \stackrel{a}{\Longrightarrow}_F Det((p_a^i + \sum_k r_a^k) + (\sum_k r_a^k + \sum_j q_a^j))$, with $(p_a^i + \sum_k r_a^k, Det((p_a^i + \sum_k r_a^k) + (\sum_k r_a^k + \sum_j q_a^j))) \in R$. □

Proposition 6.3 *For any process p we have $Det(p) \sqsubseteq_F p$.*

Proof. We will prove that $Det(p) \sqsubseteq_S^{\Rightarrow X} p$ by showing that $R = \{(Det(p), p)\}$ is a simulation for the transition system $\Longrightarrow_F$. Since $Det(p)$ is deterministic for each $a \in Act$ there is a unique transition $Det(p) \stackrel{a}{\Longrightarrow}_F Det(\sum_i p_a^i)$. By applying the definition of $\stackrel{a}{\Longrightarrow}_F$ we have $p \stackrel{a}{\Longrightarrow}_F \sum_i p_a^i$, and clearly we have $(Det(\sum_i p_a^i), \sum_i p_a^i) \in R$. □

Certainly this is a very simple example, but even so it is interesting to compare the proof above with that in [dFG05]. This proof is simpler and more natural, mainly because the proof obligations to check bisimulations forced us to remove the subterms that were not in the chosen transition when we had to simulate it. Instead, this is not necessary for any of the two simulations that are needed to check mutual simulation as above. Obviously this is also related with the impossibility to obtain a notion of local bisimulation up-to characterizing the equivalence under any of the linear semantics, as commented at the end of Section 3.

7 Conclusions and future work

We have presented an operational characterization of all the linear semantics in the spectrum. It uses the SOS approach to generate, for each such semantics, a transition system over which the adequate constrained simulation order characterizes the order defining the given semantics.

The main virtues of these characterizations are, on the one hand, their local character, so that the corresponding transition systems are easily defined and seem to be not too large for real applications. On the other hand, and most important, the genericity of the definition, that is the same for all the semantics.

The genericity is a consequence of our unification work both in [dFGP08a] but especially in [dFGP08b]. where we have presented a generic axiomatization of all the semantics in the spectrum. From there we have obtained our characterization in terms of local simulations up-to that have allowed us to prove the correctness of the operational characterizations presented in this paper.

Therefore, it seems that our unification work extends to all the fields related to the definition of process semantics, and by applying it we hope to be able to improve our understanding of both the relations between the different semantics and those between the different ways of defining them: observational, axiomatic, operational, and logical. As a matter of fact, the logical framework in the only one we have not explored yet and we plan to do it shortly, once we have found the basic pieces in which to base such last unified study.

References

[CH93] Rance Cleaveland and Matthew Hennessy. Testing equivalence as a bisimulation equivalence. *Formal Aspects of Computing*, 3:1–21, 1993.

[dFG05] David de Frutos-Escrig and Carlos Gregorio-Rodríguez. Bisimulations up-to for the linear time-branching time spectrum. In *CONCUR 2005 - Concurrency Theory, 16th International Conference*, volume 3653 of *Lecture Notes in Computer Science*, pages 278–292. Springer, 2005.

[dFG07] David de Frutos-Escrig and Carlos Gregorio-Rodríguez. Simulations up-to and canonical preorders (extended abstract). In *Structural Operational Semantics SOS 2007*, volume 192 of *Electronic Notes in Theoretical Computer Science*, pages 13–28. Elsevier, 2007.

[dFG08] David de Frutos-Escrig and Carlos Gregorio-Rodríguez. Universal coinductive characterizations of process semantics. In *5th IFIP International Conference on Theoretical Computer Science*, to appear in *Springer Science and Business Media*. Springer, 2008. http://maude.sip.ucm.es/~miguelpt/papers/universal.pdf.

[dFGP08a] David de Frutos Escrig, Carlos Gregorio-Rodríguez, and Miguel Palomino. On the unification of semantics for processes: observational semantics. Submitted. Available at http://maude.sip.ucm.es/~miguept/, 2008.

[dFGP08b] David de Frutos Escrig, Carlos Gregorio-Rodríguez, and Miguel Palomino. On the unification of process semantics: axiomatic semantics. Submitted. Available at http://maude.sip.ucm.es/~miguept/, 2008.

[Kli04] Bartek Klin. A coalgebraic approach to process equivalence and a coinductive principle for traces. In *CMCS'04: 7th International Workshop on Coalgebraic Methods in Computer Science*, volume 106 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2004.

[KM02] Antonín Kucera and Richard Mayr. Why Is Simulation Harder than Bisimulation? *CONCUR'02: Proceedings of the 13th International Conference on Concurrency Theory*, volume 2421 of *Lecture Notes in Computer Science*, pages 594–610. Springer, 2002.

[Gla01] Rob J. van Glabbeek. *Handbook of Process Algebra*, chapter The Linear Time – Branching Time Spectrum I: The Semantics of Concrete, Sequential Processes, pages 3–99. Elsevier, 2001.

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

[Plo04] Gordon D. Plotkin. A structural approach to operational semantics. *Journal of Logic and Algebraic Programming*, 60-61:17–139, 2004.