



ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in
Theoretical Computer
Science

Electronic Notes in Theoretical Computer Science 192 (2007) 13–28

www.elsevier.com/locate/entcs

Simulations Up-to and Canonical Preorders (Extended Abstract)

David de Frutos Escrig^{1,3}

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

Carlos Gregorio Rodríguez^{2,4}

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

Abstract

In this paper we define simulations up-to a preorder and show how we can use them to provide a coinductive, simulation-like, characterization of semantic preorders for processes. The result applies to a wide class of preorders, in particular to all semantic preorders coarser than the ready simulation preorder in the linear time-branching time spectrum. An interesting but unexpected result is that, when built from an equivalence relation, the simulation up-to is a canonical preorder whose kernel is the given equivalence relation. These canonical preorders have several nice properties, the main being that since all of them are defined in a homogeneous way, their properties can be proved in a generic way. In particular, we present an axiomatic characterization of each of these canonical preorders, that is obtained just by adding a single axiom to the axiomatization of the original equivalence relation. This gives us an alternative axiomatization for every axiomatizable preorder in the linear time-branching time spectrum, whose correctness and completeness can be proved once and for all.

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

1 Introduction and Related Work

Whenever a semantic framework is introduced to define the meaning of some kind of formal language, an equivalence relation is also introduced that equates two terms if they have the same semantics. Reciprocally, an equivalence relation provides a way

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

² Partially supported by the Spanish MEC project WEST/FAST TIN2006-15578-C02.

³ Email: defrutos@sip.ucm.es

⁴ Email: cgr@sip.ucm.es

to define an abstract semantics by associating to each term the equivalence class to which it belongs.

Process algebras have been largely used to specify and study the behaviour of reactive systems and have given rise to well known languages such as CSP, CCS or ACP. Many different semantics and their respective equivalence relations have been proposed for reactive systems. Most of them were collected in the linear time-branching time spectrum (ltbt for short), which was thoroughly studied by Van Glabbeek [9]. There he presented a nearly exhaustive collection of semantics, each of which was characterized by a natural testing scenario, a modal logic to identify the set of equivalent processes, and a finite axiomatization to compare any pair of finite processes.

Bisimulation semantics is the strongest of all the equivalence semantics in the spectrum and also one of the most important. Bisimulation equivalence can be easily defined due to its coinductive flavour and thus coalgebraic techniques can be applied, which provides a fruitful alternative to the classic approach based on induction and continuity arguments. Besides, bisimilarity can be easily established by an efficient algorithm based on which several tools that can effectively check process bisimilarity [5] have been developed. Despite the fact that bisimulation has been thoroughly studied since it was proposed by David Park [21] (see [24] for a recent historic presentation on the subject), it is still the topic of quite a number of recent papers such as [17].

But sometimes bisimulation equivalence is too strong and many other interesting semantics weaker than bisimilarity have been proposed, most of them appearing in the ltbt spectrum. Traces, for instance, is the weakest reasonable semantics for processes. However, non-deterministic behaviours are not properly described by means of traces, since deadlock information is not accurately captured. Failure semantics were proposed in [12] to solve this problem. An even finer semantics is that defined by readiness. Failures and ready sets can be combined with traces, thus getting stronger semantics as described in [9].

Unfortunately, in general these equivalences cannot be so easily studied as the bisimulation semantics, mainly because of their lack of a direct coalgebraic definition. However, it is possible to relate bisimulation and the rest of the semantics so that these coalgebraic techniques can be of use for their study. In [6] we have proved that all the semantics in the ltbt spectrum can be characterized as quotients of bisimulation. This is done by relaxing the proof obligations of bisimulation, so that when playing the new game the defender can modify the following transitions of the process when he makes his move, thus becoming easier to prove the corresponding equivalence.

Preorders and equivalence relations are closely related, the latter being just a particular (symmetric) case of the former, while any preorder defines an induced equivalence relation by means of its kernel. Although semantics of processes are defined by equivalence relations, we need also order relations to compare non-equivalent processes that correspond to several order relations such as “is an implementation of” [3], “is faster than” [16], or “has less amortised cost than” [14].

Besides, an order relation is also needed to specify continuity requirements in semantic domains, by means of which we can define the semantics of recursive processes. These orders have been thoroughly studied, and in particular they also appear in [9], where they are introduced using a classical testing approach: “given two processes p and q , we have that p is better than q whenever p passes as many tests as q does”, following the ideas in [20,11].

In this paper we focus on the study of semantic preorders for processes and their coalgebraic characterization. We can find in [19,4] recursive definitions of testing semantics which can be considered a first step in the desired direction, but in both cases the authors used the *after* construction in their characterizations, which means a too global approach. Instead, we want a more local characterization where bisimulation steps solve the choices in the compared processes. One may think that this is not a difficult task given the results in [6] for process equivalences, but this is not the case: bisimulation is the strongest of the semantic equivalences and thus the idea of relaxing the bisimulation requirements allows, in principle, to get weaker equivalences. However, there exists no proper preorder whose kernel is the bisimulation equivalence; in fact, the simulation preorder (the most natural coinductively defined preorder) is not stronger than many of the semantic preorders in the ltbt spectrum and the equivalence relation it induces is much weaker than the bisimulation equivalence.

Fortunately, we can overcome this handicap by reinforcing simulations, that is, by imposing some additional condition to be satisfied by the pairs of processes being related. In particular ready simulations [15,2] are simulations constrained by the condition that the set of initial actions of the processes should be the same; the ready simulation preorder is finer than any other finitely axiomatizable preorder in the ltbt spectrum.

As we will see in the rest of the paper, we have indeed obtained coalgebraic characterizations of the preorders, together with interesting results relating equivalences and preorders, and therefore, bisimulations up-to with simulations up-to. One of these results was quite unexpected, but also extremely nice: for any equivalence relation (under sensible assumptions) there exists a *canonical* preorder (non-trivial, that is, different from the equivalence itself) whose kernel is the original equivalence relation. This is the cause behind many pleasant properties, in particular, we can obtain a complete axiomatization for finite processes for any of these canonical preorders from the axiomatization for the corresponding equivalences in a systematic way, so that completeness and correctness of these axiomatizations can be proved once and for all. Besides, for all of the equivalences coarser than the ready simulation in Van Glabbeek’s spectrum, our canonical preorders define the same order relations as the ones in the literature for those semantics.

It has been a nice surprise to know that in [1] the authors⁵ have found the way to establish the opposite relation between the axiomatizations of the preorders that

⁵ We want to thank specially to Wan Fokkink for some useful indications, including those in the discussion on a previous version of this paper that is lengthily commented on in [1]. In fact, following the suggestions there we have slightly improved some of the results in this final version, so that you could find some inconsistencies between their comments and the printed text here.

are weaker than the ready simulation and those of the corresponding equivalences, for the semantics in the Van Glabbeek's spectrum. We agree with them on the fact that is more natural to look for the axiomatization of the induced equivalence starting from that of a preorder, than the other way around. Nevertheless, it is also nice to have a canonical way to obtain a non-trivial preorder whose kernel is a given equivalence relation, as we have done.

The rest of the paper is structured as follows. In Section 2 we introduce the basic definitions and notations on processes and preorders and we recall some results from [6]. In Section 3, we define simulations up-to and prove some results that characterize behaviour preorders with simulations up-to a preorder and simulations up-to an equivalence. These results are presented in two steps, first for preorders coarser than the simulation preorder and then for preorders coarser than the ready simulation, that require more elaborated proofs (although nice auxiliary results emerge). In Section 4 we shift the focus from preorders to equivalences and show some results that allow us to characterize equivalences as kernels of simulations up-to. Building on this, we characterize a canonical coinductive preorder whose kernel is a given equivalence relation. In Section 5, as an example of application of the theory developed in the paper, we provide alternative axiomatic definitions of the preorders in the linear time-branching time spectrum. The proof of their completeness is easy and simple, using the ideas of simulation up-to developed in the paper. Finally, in Section 6 we present some conclusions and lines for future work.

2 Preliminaries

The usual way to describe the behaviour of processes is by means of an operational description. As usual, we provide it by using the well-established formalism of *labelled transition systems*, or LTS for short, introduced by Plotkin (reprinted in [22]).

Definition 2.1 *A labelled transition system is a structure $\mathcal{T} = (\mathcal{P}, Act, \rightarrow)$ where \mathcal{P} is a set of processes, agents or states; Act is a set of actions; and $\rightarrow \subseteq \mathcal{P} \times Act \times \mathcal{P}$ 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 \rightarrow describes the process transitions after the execution of actions. Any triple $\langle p, a, q \rangle$ in the transition relation \rightarrow is represented by $p \xrightarrow{a} q$, 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 $p \xrightarrow{a}$ if there exists a process q such that $p \xrightarrow{a} q$. The function I calculates the set of initial actions of a process, $I(p) = \{a \mid a \in Act \text{ and } p \xrightarrow{a}\}$.

LTS's for finite processes are just directed graphs which become finite trees⁶ if expanded. These finite trees can be syntactically described by the basic process algebra BCCSP, which was also used, for instance, in [9,6].

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.

All the definitions we present in the paper are valid for arbitrary processes, that is, for arbitrary rooted LTSs, either finite or infinite. The proofs that we provide in Sections 3 and 4 make extensive use of inductive reasoning and therefore they are only valid for BCCSP processes, that is, for finite processes. However, as we did in [6], by using the standard Approximation Induction Principle [8], we can extend all our results first to infinite depth finitary branching tree processes, and then to arbitrary finitely branching transition systems, since by unfolding any of them we can get an equivalent finitary tree process.

The operational semantics for the BCCSP terms is defined in Figure 1. The depth of a BCCSP process is the depth of the tree it denotes.

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

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 [9]), 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 indexes 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 2.3 *A preorder relation \sqsubseteq over processes is a behaviour preorder if*

⁶ We obtain 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 [18].

	B	RS	PW	RT	FT	R	F	CS	CT	S	T
$(x + y) + z = x + (y + z)$	+	+	+	+	+	+	+	+	+	+	+
$x + y = y + x$	+	+	+	+	+	+	+	+	+	+	+
$x + 0 = x$	+	+	+	+	+	+	+	+	+	+	+
$x + x = x$	+	+	+	+	+	+	+	+	+	+	+
.....
$ax \sqsubseteq ax + ay$		+	+	+	+	+	+	v	v	v	v
$a(bx + by + z) = a(bx + z) + a(by + z)$			+	v	v	v	v		v		v
$I(x) = I(y) \Rightarrow ax + ay = a(x + y)$				+	v	v	v		v		v
$ax + ay \sqsupseteq a(x + y)$					+		v		v		v
$a(bx + u) + a(by + v) \sqsupseteq a(bx + by + u)$						+	v		v		v
$ax + a(y + z) \sqsupseteq a(x + y)$							+		v		v
.....
$ax \sqsubseteq ax + y$								+	+	v	v
$a(bx + u) + a(cy + v) = a(bx + cy + u + v)$									+		v
.....
$x \sqsubseteq x + y$										+	+
$ax + ay = a(x + y)$											+

Table 1
Axiomatization for the Preorders in the Linear Time-Branching Time Spectrum I [9]

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

Table 1, borrowed from [9], shows a complete axiomatization for some of the semantics in the lbtbt spectrum, with the corresponding axioms for each preorder (column) marked with “+”. Axioms marked with “v” are satisfied but not required. The shorthands on top of the columns refer to the different semantics, B stands for bisimulation equivalence, and similarly for the rest of the preorders that appear on the linear time-branching time spectrum.

The first four axioms on top left of Table 1 characterize bisimulation equivalence. They also belong to any other axiomatic characterization and therefore are assumed and usually omitted when talking about other semantics with less discriminatory power. We can see, for instance, that the ready simulation preorder is characterized by the four axioms of the bisimulation equivalence plus the axiom (RS) $ax \sqsubseteq ax + ay$. Analogously, axioms for bisimulation together with axiom (S) $x \sqsubseteq x + y$, characterize the simulation preorder.

Let us conclude this preliminary section with some results about semantic equivalences. In [6] we introduced bisimulations up-to a preorder in order to weaken the definition of bisimulations in such a way that weaker equivalences could be captured by a coinductive definition.

Definition 2.4 *Let \sqsubseteq be a behaviour preorder. Then a binary relation S over processes is a bisimulation up-to \sqsubseteq if pSq implies that:*

- For every a , if $p \xrightarrow{a} p'_a$, then there exist q' and q'_a , $q \sqsupseteq q' \xrightarrow{a} q'_a$ and $p'_a Sq'_a$;
- For every a , if $q \xrightarrow{a} q'_a$, then there exist p' and p'_a , $p \sqsupseteq p' \xrightarrow{a} p'_a$ and $p'_a Sq'_a$.

Two processes are bisimilar up-to \sqsubseteq , written $p \approx_{\sqsubseteq} q$, if there exists a bisimulation

up-to \sqsubseteq , S , such that pSq .

The added capability introduced by the \sqsubseteq -reduction generalizes the original definition of bisimulation, so that we have now more chances of proving equivalence between processes.

To understand the conditions in Theorem 2.6 below we need the following definitions.

Definition 2.5 *A behaviour preorder \sqsubseteq is initials preserving when $p \sqsubseteq q$ implies $I(p) \subseteq I(q)$. It is action factorised (or just factorised) when $p \sqsubseteq q$ implies $p|_a \sqsubseteq q|_a$, for all $a \in I(p)$.*

Initials preservation and factorisation are properties that are satisfied by any “natural” process semantics, in particular by any of the behaviour preorders in the ltbt spectrum, from trace preorder to ready simulation preorder.

Theorem 2.6 ([6]) *Let \sqsubseteq a behaviour preorder that is initials preserving, action factorised and satisfies the axiom (RS), and \equiv the induced equivalence, we have that \approx_{\sqsubseteq} and \equiv are the same relation.*

This result is rather general and applies to a wide class of process semantics. In particular, any of the preorders in Table 1 satisfies the conditions of the theorem. Therefore, for any of these semantic preorders the corresponding bisimulation up-to characterizes the induced equivalence.

Corollary 2.7 ([6]) *For every behaviour preorder \sqsubseteq that is initials preserving, action factorised and satisfies the axiom (RS), we have $\equiv = \approx_{\sqsubseteq} = \approx_{\equiv}$.*

Although our results in [6] are quite promising, they do not provide a complete answer to the problem of finding coinductive characterizations of semantics for processes. Why? As we mentioned above, preorders are even more important than equivalences for defining semantics of process algebras, and therefore it is also interesting to get coinductive characterizations for them.

3 Simulations Up-to a Preorder

When we first addressed the problem of finding coinductive characterizations for process equivalences we had a clear starting point: bisimulation equivalence. Bisimulation is the strongest equivalence and therefore by weakening its definition (Definition 2.4) we could obtain weaker semantics (Theorem 2.6). Finding out how to define coinductive characterizations for the semantic preorders is not such an easy task. We start by modifying the classical definition of simulation to get the following definition of simulations up-to a preorder.

Definition 3.1 *For \sqsubseteq a behaviour preorder, we say that a binary relation S over processes is a simulation up-to \sqsubseteq , if pSq implies that:*

- *For every a , if $p \xrightarrow{a} p'_a$ there exist q' and q'_a , $q \sqsupseteq q' \xrightarrow{a} q'_a$ and $p'_aSq'_a$.*

We say that process p is simulated up-to \sqsubseteq by process q , or that q simulates p up-to

\sqsubseteq , written $p \sqsubseteq_{\sqsubseteq} q$, if there exists a simulation up-to \sqsubseteq , S , such that pSq .

For the sake of simplicity, we often just write \sqsubseteq instead of $\sqsubseteq_{\sqsubseteq}$ when the behaviour preorder is clear from the context.

Example 3.2 Let us consider the processes $s = a(b(d + e) + cd)$ and $t = abf + a(be + bd + cd)$. It is clear that for the simulation preorder we have $s \not\sqsubseteq_S t$ because after executing ab in s we arrive to a state in which the choice $d + e$ is possible, but after executing ab in t it is not.

By contrast, for the trace preorder we clearly have $s \sqsubseteq_T t$, since the set of traces of s , $\{abd, abe, acd\}$, is included in the set of traces of t , $\{abf, abe, abd, acd\}$. Let us see how we could check that $s \sqsubseteq_{\sqsubseteq_T} t$, by constructing the corresponding simulation up-to \sqsubseteq_T .

If process s performs action a and arrives to $s' = b(d + e) + cd$, then process t does not need to apply any preorder reduction, it just simulates the move by performing action a and evolving into $t' = be + bd + cd$. Now we have to check that $s' \sqsubseteq_{\sqsubseteq_T} t'$: if s' performs action c then t' can trivially emulate that move arriving to the same state. The only non trivial case to check happens when s' performs action b and evolves into $d + e$. In that case, t' should take advantage of the possibility of trace reduction, $t' \sqsupseteq_T b(e + d) + cd$, and then action b is executed to arrive to $d + e$ as well, thus completing the verification of the simulation up-to obligations.

Certainly, if we know in advance that $s \sqsubseteq_T t$, we could directly reduce t into s when checking $s \sqsubseteq_{\sqsubseteq_T} t$, but what we want to illustrate here is how we would use in practice our coalgebraic characterization: we do not want to use any complicate information about the corresponding order, \sqsubseteq_T in this case, but only some easier to obtain pairs of the relation, as we have done when reducing t' above.

The next result shows that simulations up-to are correct with respect to the corresponding base preorder.

Proposition 3.3 For every preorder \sqsubseteq , if $p \sqsubseteq q$ then $p \sqsubseteq_{\sqsubseteq} q$.

The next theorem states the completeness of the definition of simulations up-to a preorder with respect to any preorder satisfying the axiom (S), i.e., for any preorder that is weaker than the simulation preorder, \sqsubseteq_S .

Theorem 3.4 For every behaviour preorder \sqsubseteq that satisfies the axiom (S), we have $p \sqsubseteq_{\sqsubseteq} q$ if and only if $p \sqsubseteq q$.

Note that the given preorder must satisfy the axiom (S), since we have $\sqsubseteq_S = \sqsubseteq_{\emptyset} \Rightarrow \sqsubseteq_{\sqsubseteq}$.

Theorem 3.4 characterizes semantic preorders in the same way that semantic equivalences were characterized in Theorem 2.6, though in both cases we use preorders for the up-to relation. It would be nice to have a dual characterization where the equivalences were used to characterize the semantic preorder. That is indeed possible, as stated in the following proposition.

Proposition 3.5 For every behaviour preorder \sqsubseteq that satisfies the axiom (S), we have that $p \sqsubseteq q \Rightarrow p \sqsubseteq_{\sqsubseteq} q$.

Corollary 3.6 *For every behaviour preorder \sqsubseteq that satisfies the axiom (S), we have that the relations \sqsubseteq , $\sqsubseteq_{\sqsubseteq}$ and \sqsubseteq_{\equiv} are the same.*

Considering both bisimulations and simulations up-to we can draw the diagram of equivalences in the following corollary.

Corollary 3.7 *For every behaviour preorder \sqsubseteq that satisfies the axiom (S) the following equivalences hold:*

$$\begin{array}{ccccc}
 p \equiv q & \Leftrightarrow & p \sim_{\sqsubseteq} q & \Leftrightarrow & p \sim_{\equiv} q \\
 \Downarrow & & \Downarrow & & \Downarrow \\
 p \sqsubseteq q \wedge p \sqsupseteq q & \Leftrightarrow & p \sqsubseteq_{\sqsubseteq} q \wedge p \sqsupseteq_{\sqsubseteq} q & \Leftrightarrow & p \sqsubseteq_{\equiv} q \wedge p \sqsupseteq_{\equiv} q
 \end{array}$$

Considering the semantics in the lbt spectrum, (only) trace and simulation preorders (see Table 1) satisfy the axiom (S) and thus fulfil the hypothesis of Corollary 3.7. Therefore, in both cases mutual simulation up-to and bisimulation up-to define the same equivalence relation as the kernel of the preorder. Thus we provide two alternative characterizations of each of these preorders and four alternative characterizations of the induced equivalences.

These results, although interesting, fall short of the generality that we achieved in [6]. This limitation comes from the fact that the definition of \sqsubseteq is based on the simulation semantics, that has a rather weak discriminatory power. In order to get more general results, similar to those in Theorem 3.4, for other stronger semantics such as failures or readiness, we need to add more discriminating power to the simulations we start from. The ready simulation semantics is stronger than any other of the axiomatized semantics in [9]. It will serve as the basis to define an stronger notion of simulation up-to. From now on, we will consider the binary relation I defined over pairs of processes by $pIq \Leftrightarrow I(p) = I(q)$.

Definition 3.8 *For \sqsubseteq a behaviour preorder, we say that a binary relation S over processes is an I -simulation up-to \sqsubseteq , if $S \subseteq I$ (that is, $pSq \Rightarrow pIq$), 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 $p'_a Sq'_a$;
- pIq .

We say that process p is I -simulated up-to \sqsubseteq by process q , or that process q I -simulates process p up-to \sqsubseteq , written $p \sqsubseteq_{\sqsubseteq}^I q$, if there exists an I -simulation up-to \sqsubseteq , S , such that pSq .

For the sake of simplicity, we sometimes just write \sqsubseteq^I instead of $\sqsubseteq_{\sqsubseteq}^I$ when the behaviour preorder is clear from the context.

The following proposition relates a behaviour preorder with the corresponding I -simulation up-to.

Proposition 3.9 *For every preorder \sqsubseteq such that $\sqsubseteq \subseteq I$, if $p \sqsubseteq q$ then $p \sqsubseteq_{\sqsubseteq}^I q$.*

Now we can use I -simulations up-to to prove a similar result to that in Theorem 3.4 for semantic preorders with more discriminating power. Note that a pair of processes related by any preorder relation ranging from failure preorder to ready simulation preorder (see Table 1) in the ltbt spectrum satisfies the I condition.

Theorem 3.10 *For every behaviour preorder \sqsubseteq that satisfies the axiom (RS) and $\sqsubseteq \subseteq I$, we have $p \sqsubseteq_{\sqsubseteq}^I q$ if and only if $p \sqsubseteq q$.*

Note that the given preorder must satisfy the axiom (RS), since $\sqsubseteq_{RS} = \sqsubseteq_{\emptyset}^I \Rightarrow \sqsubseteq_{\sqsubseteq}^I$, where \sqsubseteq_{RS} is the ready simulation preorder. We also need the second proviso $\sqsubseteq \subseteq I$, because we always have $\sqsubseteq_{\sqsubseteq}^I \subseteq I$.

The original technique of bisimulation up-to in the works by Milner and Sangiorgi [18,23] aims to decrease the size of the relations that prove that two processes are bisimilar. Without going into detail, it is possible to use a small (known) part of the bisimilarity relation to generate other less obvious bisimilar pairs. Our (bi)simulations up-to can be used exactly in the same way: by using a part of the relation \sqsubseteq we can generate, via $\sqsubseteq_{\sqsubseteq}^I$, other pairs in the relation.

Next we will see that we can also characterize a preorder in terms of simulation up-to its kernel equivalence, so that the characterized preorder disappears from the definition of the characterizing one. We first present an auxiliary result relating a preorder with the induced equivalence relation. In our opinion this result, even if rather simple, is quite interesting by itself.

Lemma 3.11 *For every behaviour preorder \sqsubseteq that satisfies the axiom (RS) and that is either initials preserving or weaker than I , we have that $p \sqsubseteq q \Rightarrow q \equiv q + p$.*

All the preorders defining the semantics in the ltbt spectrum that are coarser than the ready simulation satisfy the hypothesis of this lemma, since all of them are action factorised. We have also considered the other alternative proviso in order to get a more general result. Besides, it is interesting to note that the converse of the previous result is not true in general. In order to have it we need to impose an additional condition.

Proposition 3.12 *For every behaviour preorder \sqsubseteq that satisfies the axiom (RS) and $\sqsubseteq \subseteq I$, we have that $p \sqsubseteq q \Leftrightarrow q \equiv q + p \wedge pIq$.*

Although we will not use the previous result in this section, it is a clear inspiration for one of the main results in Section 4, namely Corollary 4.7.

By using Lemma 3.11 we can now easily prove the following result.

Proposition 3.13 *For every behaviour preorder \sqsubseteq that satisfies the axiom (RS) and $\sqsubseteq \subseteq I$, we have $p \sqsubseteq q \Rightarrow p \sqsubseteq_{\sqsubseteq}^I q$.*

The following corollaries summarize the previous results.

Corollary 3.14 *For every behaviour preorder \sqsubseteq that satisfies the axiom (RS) and $\sqsubseteq \subseteq I$, we have that the relations \sqsubseteq , $\sqsubseteq_{\sqsubseteq}^I$ and \sqsubseteq_{\equiv}^I are the same.*

Corollary 3.15 *For every behaviour preorder \sqsubseteq that satisfies the axiom (RS) and*

$\sqsubseteq \subseteq I$, the following equivalences hold:

$$\begin{array}{ccccc}
 p \equiv q & \Leftrightarrow & p \sim_{\sqsubseteq} q & \Leftrightarrow & p \sim_{\equiv} q \\
 \Downarrow & & \Downarrow & & \Downarrow \\
 p \sqsubseteq q \wedge p \sqsupseteq q & \Leftrightarrow & p \sqsubseteq^I q \wedge p \sqsupseteq^I q & \Leftrightarrow & p \sqsubseteq^I q \wedge p \sqsupseteq^I q
 \end{array}$$

Corollaries 3.14 and 3.15 apply to a wide class of process preorders. Considering the ltbt spectrum, any behaviour preorder between failure and ready simulation satisfies the conditions and therefore we can apply these results to them. Therefore, Corollary 3.15 provides for any of the preorders between failure and ready simulation, and the corresponding equivalences, a characterization both in terms of bisimulation-like relations and in terms of mutual simulation-like relations.

In fact, from our results about simulations up-to we can now prove in an indirect way Theorem 2.6, that was our main result in [6].

- Corollary 3.7 is true for any preorder satisfying axiom (S), in particular, for trace and simulation preorders in the ltbt spectrum.
- Corollary 3.15 can be applied to any behaviour preorder satisfying (RS) and such that $p \sqsubseteq q \Rightarrow I(p) = I(q)$, in particular, this is the case for the preorders in the ltbt spectrum between failure preorder and ready simulation preorder, as one can immediately conclude looking at the defining axioms in Table 1.
- There are only two other semantics in the linear time-branching time spectrum, completed trace and completed simulation, whose preorders satisfy neither the conditions of Corollary 3.7 nor those of Corollary 3.15. However, for these preorders the corresponding results can be easily proved by restricting the simulation by means of the relation CI given by $I(p) = \emptyset \Leftrightarrow I(q) = \emptyset$, and defining CI -simulations up-to \sqsubseteq , that induce the order relation $\sqsubseteq_{\sqsubseteq}^{CI}$. As a matter of fact, in this case the results would be true for any behaviour preorder satisfying the (CS) axiom, $ap \sqsubseteq ap + q$.

Theorem 2.6 imposes to the preorders the condition of being initials preserving and action factorised and the result is valid for any behaviour preorder satisfying the axiom (RS). By contrast, when using simulations up-to we do not need to ask for action factorisation, while initials preservation was only assumed when we imposed $\sqsubseteq \subseteq I$, in Corollary 3.15.

Therefore to obtain our results we have classified the semantic preorders in the ltbt spectrum in three different slices. These slices are separated in Table 1 by dotted lines. Each slice corresponds, respectively, to any of the axioms (S), (CS) and (RS).

4 Simulations Up-to an Equivalence

All the results we have presented in the previous sections are based on the existence of a semantic preorder which satisfies certain properties. In many cases, these results

relate a given preorder and its induced equivalence. However, as we will show in this section, the technique of simulations up-to produces some interesting results even if we do not have such a preorder to start with.

As we have discussed at the end of Section 3 the results for simulations up-to come in slices determined by the basic class of simulations we consider in each case. In this section we just state and prove the most difficult case, that corresponding to the slice determined by the ready simulation semantics.

To start with, we extend the definition of behaviour preorder to equivalence relations in a natural way.

Definition 4.1 *An equivalence relation \equiv over processes is a behaviour equivalence when it is weaker than bisimulation equivalence, i.e. $p =_B q \Rightarrow p \equiv q$, and it is a congruence with respect to the prefix and choice operators, i.e. if $p \equiv q$ then $ap \equiv aq$ and $p + r \equiv q + r$.*

We will be specially interested in all the behaviour equivalences coarser than the ready simulation equivalence. They are those that satisfy the following axiom:

$$(RS_{\equiv}) \quad I(x) = I(y) \Rightarrow a(x + y) \equiv a(x + y) + ay$$

In Table 2 a complete axiomatization for the semantic equivalences in the ltbt spectrum can be found.

The first result that we present relates I -simulations up-to and equivalences, and the application of choice to the processes related by it.

Lemma 4.2 *For every behaviour equivalence \equiv satisfying (RS_{\equiv}) and $\equiv \subseteq I$, we have that $p \sqsubset_{\equiv}^I q \Rightarrow q \equiv q + p$.*

Now we can state and prove the characterization of a given equivalence relation by means of the corresponding simulations up-to.

Theorem 4.3 *For every behaviour equivalence \equiv satisfying (RS_{\equiv}) and $\equiv \subseteq I$, we have $p \equiv q \Leftrightarrow p \sqsubset_{\equiv}^I q \wedge p \sqsupset_{\equiv}^I q$.*

As a consequence, we also get a characterization of the equivalences in terms of bisimulations up-to.

Corollary 4.4 *For every behaviour equivalence \equiv satisfying (RS_{\equiv}) and $\equiv \subseteq I$, we have $p \equiv q \Leftrightarrow p \sqsubset_{\equiv}^I q \wedge p \sqsupset_{\equiv}^I q \Leftrightarrow p \approx_{\equiv} q$.*

The characterization in Theorem 4.3 tells us that any behavior equivalence can be defined by means of simulations up-to. Besides, and this is even more important, in this way a preorder is defined whose kernel is the original equivalence. Moreover, this preorder satisfies some interesting properties.

Proposition 4.5 *For every behaviour equivalence \equiv satisfying (RS_{\equiv}) and $\equiv \subseteq I$, we have that \sqsubset_{\equiv}^I is a behaviour preorder that satisfies (RS) , $\sqsubset_{\equiv}^I \subseteq I$, and whose kernel is \equiv .*

As a consequence, given an equivalence, we have a way to characterize a particular preorder whose kernel is that equivalence.

Theorem 4.6 *For every behaviour equivalence \equiv satisfying (RS_{\equiv}) and $\equiv \subseteq I$, the preorder \sqsubseteq_{\equiv}^I is the only behaviour preorder that satisfies (RS) and is contained in I , whose kernel is \equiv .*

This means that \sqsubseteq_{\equiv}^I is the *canonical* preorder generated by \equiv fulfilling all the conditions above. This canonical preorder can be characterized in a simple way in terms of the corresponding equivalence and the condition I that all of them satisfy.

Corollary 4.7 *For every behaviour equivalence \equiv satisfying (RS_{\equiv}) and $\equiv \subseteq I$, the preorder defined as $p \sqsubseteq q \Leftrightarrow q \equiv q + p \wedge I(p) = I(q)$ is another characterization of the canonical preorder generated by \equiv .*

It is nice to find out that the “classical” preorders that appear in the literature for the different semantics in the linear time-branching time spectrum coincide with our canonical preorders.

Corollary 4.8 *For every semantic equivalence \equiv in the ltbt spectrum between failure equivalence and the ready simulation equivalence, the corresponding preorder \sqsubseteq is the canonical preorder generated by the given equivalence \equiv .*

Quite a number of results follow from the previous propositions and announce a rich underlying algebraic theory. We would like to point out the following ones:

Corollary 4.9 *For every behaviour equivalence \equiv that satisfies the property (RS_{\equiv}) and $\equiv \subseteq I$, we have $\equiv = \approx_{\sqsubseteq_{\equiv}^I}$, and $\sqsubseteq_{\equiv}^I = \approx_{\sqsubseteq_{\equiv}^I}^I$.*

To conclude this section we would like to comment on the results in our Corollaries 4.7 and 4.8. There are several preorders whose kernels are a given behaviour equivalence. Amongst them we have the canonical preorder, as defined above, the equivalence itself, or the so called canonical preorder in terms of lattice theory, that is defined as $p \sqsubseteq' q \Leftrightarrow q \equiv q + p$. It can be seen that \sqsubseteq' is not the same as our canonical preorder, which here we will just denote by \sqsubseteq , for all the behaviour equivalences satisfying the hypothesis of Theorem 4.6.

For instance, for the preorders induced by the ready simulation equivalence we have $\mathbf{0} \sqsubseteq' p$ for any process p , but if $p \neq \mathbf{0}$ then $\mathbf{0} \not\sqsubseteq p$. Applying Corollary 4.7 we have that $p \sqsubseteq q \Leftrightarrow p \sqsubseteq' q \wedge I(p) = I(q)$, therefore the only difference between \sqsubseteq and \sqsubseteq' is in the set of initial actions of the processes, but this is crucial to get the characterization of the corresponding preorders in Corollary 4.8.

5 A Few Results on Axiomatic Characterizations

As an example of the possibilities that the up-to technique offers, in this section we prove some results on the axiomatic characterization of behaviour preorders.

Corollary 3.14 states that a behaviour preorder (under some conditions) can be characterized by the I -simulation up-to the kernel of that preorder, $\sqsubseteq = \sqsubseteq_{\equiv}^I$. This result suggested to us the possibility of finding an axiomatization for the preorder from that of the equivalence. More concretely, if A_E is a set of axioms that characterizes a given equivalence \equiv , we can easily define an axiomatization for the

canonical preorder \sqsubseteq_{\equiv}^I . The new set of axioms can be obtained by just adding the (RS) axiom to the axioms for the equivalence: $A_P = A_E \cup \{ax \sqsubseteq ax + ay\}$. We formalise this in the following theorem.

Theorem 5.1 *For every behaviour equivalence \equiv satisfying (RS $_{\equiv}$) and $\equiv \subseteq I$, for which we have an axiomatization A_E , we have that $A_P = A_E \cup \{ax \sqsubseteq ax + ay\}$ is an axiomatization of the relation \sqsubseteq_{\equiv}^I .*

	B	RS	PW	RT	FT	R	F	CS	CT	S	T
$(x + y) + z = x + (y + z)$	+	+	+	+	+	+	+	+	+	+	+
$x + y = y + x$	+	+	+	+	+	+	+	+	+	+	+
$x + 0 = x$	+	+	+	+	+	+	+	+	+	+	+
$x + x = x$	+	+	+	+	+	+	+	+	+	+	+
$I(x) = I(y) \Rightarrow a(x + y) = a(x + y) + ay$		+	v	v	v	v	v	v	v	v	v
$a(bx + by + z) = a(bx + z) + a(by + z)$			+	v	v	v	v		v		v
$I(x) = I(y) \Rightarrow ax + ay = a(x + y)$				+	+	v	v		v		v
$ax + ay = ax + ay + a(x + y)$					+		v		v		v
$a(bx + u) + a(by + v) = a(bx + by + u) + a(by + v)$						+	+		v		v
$ax + a(y + z) = ax + a(x + y) + a(y + z)$							+		v		v
$a(x + by + z) = a(x + by + z) + a(by + z)$								+	v	v	v
$a(bx + u) + a(cy + v) = a(bx + cy + u + v)$									+		v
$a(x + y) = a(x + y) + ay$										+	v
$ax + ay = a(x + y)$											+

Table 2
Axiomatization for the Equivalences in the Linear Time-Branching Time Spectrum I [9]

We can directly apply Theorem 5.1 to those equivalences in the ltbt spectrum that satisfy the right conditions. In Table 2 appears an axiomatization for the equivalences of the ltbt spectrum. From these axioms we can define an alternative axiomatization of the preorders in Table 1.

Corollary 5.2 *Let us consider $\mathcal{O} \in \{F, R, FT, RT, PW, RS\}$, we have a finite axiomatization for the preorders $\sqsubseteq_{\mathcal{O}}$ just by adding the axiom (RS) to the axioms for $\equiv_{\mathcal{O}}$.*

It is interesting to note that in [9], to prove the completeness of the axiomatizations in Table 1 elaborated proofs were needed, whereas here we get all these completeness results, once for all, based on the completeness of the axiomatization of the corresponding equivalence.

6 Conclusions and Future Work

In this paper we have introduced the notion of simulation up-to by means of which we have got coalgebraic characterizations of semantic preorders. In particular we have characterized all the preorders associated to the semantics in the linear time-branching time spectrum, in a similar way to the corresponding characterizations of the semantic equivalences using bisimulations up-to [6].

Besides, we have obtained several new results connecting semantic preorders with the corresponding equivalences, and also some others relating bisimulations

up-to with mutual simulations up-to. In fact, for large families of semantics, including those in the ltbt spectrum coarser than ready simulation, the results for bisimulations up-to arose as a corollary for the corresponding ones for simulations up-to (see Corollaries 3.7 and 3.15).

A rather unexpected result was that given an equivalence relation we can obtain a canonical preorder whose kernel is precisely the equivalence relation, by means of the simulation up-to it. It is clear that we can obtain the same equivalence as the kernel of many different preorders, but now we can distinguish among them a canonical preorder which can be defined in a systematic way and has some interesting properties that come from the homogeneous way in which it was defined. It is nice to find that for all the semantics in the ltbt spectrum the so obtained canonical preorders are the same as the ones we already knew from the literature.

As a consequence of our characterization we have discovered new properties that provide us with new proof techniques to produce generic proofs valid for all these canonical preorders. In particular, we have obtained an axiomatization of the canonical preorder from the axiomatization of the corresponding equivalence.

Besides, we plan to continue with our work relating bisimulations and simulations, and we are particularly interested in translating our results to the pure coalgebraic world, comparing them to those presented by Hughes and Jacobs in [13] and Hasuo in [10]. Some related steps in this direction already appeared in [7], where we present several bisimulation-like semantics for distributed systems that can be formalised as categorical simulations, and therefore inherit all their nice coalgebraic properties.

Acknowledgements

We would like to express our gratitude to Miguel Palomino for all his comments on a previous version of this paper.

References

- [1] Luca Aceto, Wan Fokkink, and Anna Ingólfssdóttir. Ready to preorder: get your BCCSP axiomatization for free! In *Proc. 2nd Conference on Algebra and Coalgebra in Computer Science - CALCO'07*, volume to appear of *Lecture Notes in Computer Science*. Springer, 2007.
- [2] Bard Bloom, Sorin Istrail, and Albert R. Meyer. Bisimulation can't be traced. *Journal of the ACM*, 42(1):232–268, 1995.
- [3] E. Brinksma. A theory for the derivation of tests. In *Protocol Specification, Testing and Verification VIII*, pages 63–74. North Holland, 1988.
- [4] Rance Cleaveland and Matthew Hennessy. Testing equivalence as a bisimulation equivalence. *Formal Aspects of Computing*, 3:1–21, 1992.
- [5] Rance Cleaveland, Joachim Parrow, and Bernhard Steffen. The concurrency workbench: a semantics-based tool for the verification of concurrent systems. *ACM Trans. Program. Lang. Syst.*, 15(1):36–72, 1993.
- [6] 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.

- [7] David de Frutos-Escrig, Fernando Rosa-Velardo, and Carlos Gregorio-Rodríguez. New bisimulation semantics for distributed systems. In *FORTE 2007*, volume to appear of *Lecture Notes in Computer Science*. Springer, 2007.
- [8] Rob J. van Glabbeek. Bounded nondeterminism and the approximation induction principle in process algebra. In *STACS 87*, volume 247 of *Lecture Notes in Computer Science*, pages 336–347. Springer, 1987.
- [9] 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.
- [10] Ichiro Hasuo. Generic forward and backward simulations. In *International Conference on Concurrency Theory (CONCUR 2006)*, volume 4137 of *Lecture Notes in Computer Science*, pages 406–420. Springer, Berlin, 2006.
- [11] Matthew Hennessy. *Algebraic Theory of Processes*. MIT Press, 1988.
- [12] C.A.R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.
- [13] Jesse Hughes and Bart Jacobs. Simulations in coalgebra. *Theoretical Computer Science*, 327(1–2):71–108, 2004.
- [14] Astrid Kiehn and A. Arun-Kumar. Amortised bisimulations. In *Formal Techniques for Networked and Distributed Systems-FORTE 2005*, volume 3235 of *Lecture Notes in Computer Science*, pages 320–334. Springer, 2005.
- [15] Kim G. Larsen and Arne Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94(1):1–28, 1991.
- [16] Gerald Lüttgen and Walter Vogler. Bisimulation on speed: A unified approach. In *FOSSACS 2005*, volume 3341 of *Lecture Notes in Computer Science*, pages 79–94. Springer, 2005.
- [17] Massimo Merro and Matthew Hennessy. A bisimulation-based semantic theory of safe ambients. *ACM Trans. Program. Lang. Syst.*, 28(2):290–330, 2006.
- [18] Robin Milner. *Communication and Concurrency*. Prentice Hall, 1989.
- [19] Rocco De Nicola. Extensional equivalences for transition systems. *Acta Informatica*, 24(2):211–237, 1987.
- [20] Rocco De Nicola and Matthew Hennessy. Testing equivalence for processes. In *ICALP’83 – 10th International Colloquium on Automata, Languages and Programming*, volume 154 of *Lecture Notes in Computer Science*, pages 548–560. Springer, 1983.
- [21] David M.R. Park. Concurrency and automata on infinite sequences. In *Theoretical Computer Science, 5th GI-Conference*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer, 1981.
- [22] Gordon D. Plotkin. A structural approach to operational semantics. *Journal of Logic and Algebraic Programming*, 60-61:17–139, 2004.
- [23] Davide Sangiorgi. On the bisimulation proof method. *Journal of Mathematical Structures in Computer Science*, 8(5):447–479, 1998.
- [24] Davide Sangiorgi. Bisimulation and coinduction: from the origins to today. In *Algebraic Process Calculi: The first 25 years and beyond*, BRICS NS-05-3, pages 229–233, 2005.