Universal Coinductive Characterisations of Process Semantics

David de Frutos Escrig[*] and Carlos Gregorio Rodríguez[**]

Department of Sistemas Informáticos y Computación
Universidad Complutense de Madrid
defrutos,cgr@sip.ucm.es

Abstract. We present a theoretical framework which allows to define in a uniform way coinductive characterisations of nearly any semantic preorder or equivalence between processes, by means of *simulations up-to* and *bisimulations up-to*. In particular, all the semantics in the linear time-branching time spectrum are covered. *Constrained simulations*, that generalise plain simulations by including a constraint that all the pairs of related processes must satisfy, are the key to obtain such a general framework. We provide a simple axiomatisation of any constrained simulation preorder and also for the corresponding equivalence. These axiomatizations allow us to prove in a uniform way that each constrained simulation preorder (equivalence) defines a class of process preorders (equivalences) which share commons properties, like the possibility of giving coinductive characterisations for all of them, or the existence of a canonical preorder inducing each of these equivalences.

1 Introduction and Related Work

One of the essential decisions that should be taken when defining a process algebra is to settle in the most adequate way its underlying semantics. Occasionally the semantics is directly determined by an equivalence relation but more often it is based on a preorder relation, although certainly every preorder induces an equivalence by means of its kernel; besides, the ordering relations can be used to compare non-equivalent processes or to define continuous domains in order to apply fix point arguments to define the behaviour of recursive processes.

Every semantics sets forth a level of abstraction that determines which aspects of the behaviour of processes are of importance and which are not. Mainly because of the generality and diversity of the applications of process algebras, there is no prevailing semantic notion, but rather a number of different proposals have arisen from diverse approaches, contexts and applications.

We consider that this variety of process semantics is a good sign of the applicability of process algebras just proving the healthiness of the formalism. However, this plurality of semantics becomes a hindrance when the goal is to

study general properties for all of them, to compare different semantics or to determine what semantics suits a given application better.

That is why it would be nice to have a unified model that could provide us with a general and uniform approach to the different semantics. Our work targets this goal and looks for a common framework in which to include the semantics for processes. Bisimulation semantics [Par81,Mil89] is one of the most elegant and powerful equivalences defined for processes and it was our starting point to achieve this goal of uniformity. In [dFG05] we showed how to weaken the notion of bisimulation defining our bisimulations up-to that characterise many other interesting equivalences. In [dFG07] we continued that work, extending our results by considering process preorders instead of equivalences, and we have found out that this approach is indeed even more general, giving rise to a richer and more elegant theory. In both cases ready simulation [BIM95] was our main support, and that meant that we could only apply our results to preorders that were coarser than ready simulation. This restriction also appear in other related works such as [AFI07].

However, it was not clear that only the semantics coarser than ready simulation would satisfy our results. In fact, we had already presented in [dFG05] a result (Theorem 2, there) proving that we could also get coinductive characterisations of some equivalences finer than the ready simulation equivalence. To prove that theorem we required a quite ad-hoc property, that we called Hoare-Equivalence, but in [dFG07] we did not find the way to transfer these results to the framework of semantic preorders.

This paper focuses on the generalisation of the simulations up-to, and provides a general coinductive characterisation of a great variety of semantics preorders, either coarser or finer than the ready simulation preorder, in particular, this characterisation can be applied to all the semantics in the linear time-branching time spectrum [Gla01]. This generalisation has been possible after the observation that ready simulation was just a significant example of what we have called *constrained simulations*. This kind of simulations preserve the properties we need in order to prove the generalisation of previous results.

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 our previous works [dFG05, dFG07]. In Section 3 we define the family of *constrained simulations*, where simulations are constrained by the obligation to relate processes that satisfy some adequate condition. We provide a sound and complete axiomatization for the preorders and the induced equivalence relations (see Theorem 4 and 5). These axiomatizations are one of the key points in the proofs of the main results of the paper that follow in the next sections.

The core of our results is collected in Sections 4 and 5, where we define the notion of *constrained simulation up-to a preorder*; we develop our theory through a number of results that provide characterisations of the semantic preorders and equivalences in terms of constrained simulations up-to (see Theorem 6, 7, 8 and 9). Some additional results that illustrate the applications of the theory are

also included (see Theorem 10). Finally in Section 6 we conclude by discussing some research lines for future work.

2 Preliminaries and Previous Work

The behaviour of processes is usually described using the well-established formalism of *labelled transition systems* [Plo81] or lts for short.

Definition 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 $p_0 \in \mathcal{P}$.

Act is the set of actions that processes can perform and the relation $\rightarrow$ describes the process transitions after the execution of actions. The triple $\langle p, a, q \rangle$ is represented by $p \xrightarrow{a} q$, indicating that process p performs action a evolving to process q. A rooted lts describes the semantics of a process: that corresponding to its initial state p_0.

Some usual notations on lts are used. 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 for finite processes are just directed graphs which become finite trees 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. Given a set of actions Act, the set of BCCSP processes is defined by the following BNF-expression:

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

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 lts, either finite or infinite. We are going to prove the results in this paper mainly by induction on the depth of BCCSP processes. Then, by using continuity arguments (in a similar way as we did in [dFG05]) these results can be extended to arbitrary finitely branching transition systems, since by unfolding any of them we can get an equivalent finitary tree process.

The operational semantics for BCCSP terms is defined in Fig. 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

As usual, trailing occurrences of the constant **0** are omitted: we write a instead of $a\mathbf{0}$. By using $\sum$ as a shorthand for multiple choice (which is commutative and associative) we can define any process as $\sum_i \sum_j a_i p_{ij}$. A process aq' is a summand of the process q if and only if $q \xrightarrow{a} q'$. Given $a \in Act$ we define $p|_a$ as the (sub)process we get by adding all the a-summands of p. That is, if $p = \sum_i \sum_j a_i p_{ij}$, then $p|_{a_i} = \sum_j a_i p_{ij}$.

Preorders, that we represent by $\sqsubseteq$, are reflexive and transitive relations. 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 $q \sqsubseteq p$. Finally, bisimulation equivalence is denoted by $=_B$.

Definition 3. A preorder relation $\sqsubseteq$ over processes is a *behaviour preorder* if it is coarser 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 $p + r \sqsubseteq q + r$. Besides, if the relation is symmetric, i.e. is an equivalence relation, we say that it is a *behaviour equivalence*.

In [dFG05] we introduced bisimulations up-to a preorder (that we denote by $\eqsim_\sqsubseteq$) in order to weaken the definition of bisimulations in such a way that weaker equivalences could be captured by a coinductive definition.

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

Two processes are *bisimilar up-to* $\sqsubseteq$, written $p \eqsim_\sqsubseteq q$, if there exists a bisimulation up-to $\sqsubseteq$, S, such that pSq.

The added capability introduced by the $\sqsupseteq$-reduction generalises the original definition of bisimulation, so that we have now more chances to prove the equivalence of two processes. When the behaviour preorder is just the identity relation we get the bisimulation equivalence, but, as we proved in [dFG05], we get other interesting semantics (traces, failures, ready simulation and so on) by considering other behaviour preorders. One of the main results in that paper (see Theorem 1 below) required the preorders to satisfy the axiom (RS), $ax \sqsubseteq ax + ay$ (that characterises the ready simulation preorder [BIM95]) so that it could only be applied to semantics coarser than the ready simulation.

Definition 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 $a \in I(p)$.

Theorem 1 ([dFG05]). *For every behaviour preorder $\sqsubseteq$ that is initials preserving, action factorised and satisfying the axiom (RS), we have that $p \eqsim_{\sqsubseteq} q$ if and only if $p \equiv q$.*

This theorem provides a symmetric, coinductive, bisimulation-like characterisation for any equivalence in the linear time-branching time spectrum from trace equivalence to ready simulation equivalence.

Once we had coinductive characterisations for many semantic equivalences we shifted the focus from equivalences to preorders. In [dFG07] we first achieved characterisations of some semantic preorders in terms of simulations up-to.

Definition 6. Let $\sqsubseteq$ be 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 $p'_a S q'_a$.

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

Theorem 2 ([dFG07]). *For every behaviour preorder $\sqsubseteq$ that satisfies the axiom (S) $x \sqsubseteq x + y$, we have $p \lesssim_{\sqsubseteq} q$ if and only if $p \sqsubseteq q$.*

This result only applies to preorders coarser than the simulation preorder and therefore it falls short of the generality we got in Theorem 1. In order to regain this generality we needed to strengthen the simulation relation to achieve a greater discriminating power. Ready simulation was again called to play an essential role.

Definition 7. Let I be the binary relation that captures the equivalence of initial actions and is defined over pairs of processes by $pIq \Leftrightarrow I(p) = I(q)$. Let $\sqsubseteq$ be 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 $q \sqsupseteq q' \xrightarrow{a} q'$ and $p'_a S q'_a$;
– pIq;

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

By using this definition we were able to characterise all the preorders finer than failures and coarser than ready simulation.

Theorem 3 ([dFG07]). *For every behaviour preorder $\sqsubseteq$ that satisfies the axiom (RS), and $\sqsubseteq \subseteq I$, we have that $p \sqsubseteq_{\sqsubseteq}^{I} q$ if and only if $p \sqsubseteq q$.*

Once again, we needed the axiom (RS) to prove the theorem above. In the rest of the paper we will generalise our results by taking as starting point a more general class of simulations that we have called *constrained simulations*.

3 Constrained Simulations

C-constrained simulations are just plain simulations to which we impose that their pairs should also be related by the constraint C.

Definition 8. Given a relation C over BCCSP processes, a relation S_C is a *C-constrained simulation*, if pS_Cq implies:

– For every a, if $p \xrightarrow{a} p'$ there exists q', $q \xrightarrow{a} q'$ and $p'S_Cq'$, and
– pCq.

We say that process p is C-simulated by process q, or that q C-simulates p, written $p \sqsubseteq_{\rightarrow}^{C} q$, whenever there exists a C-constrained simulation S_C, such that pS_Cq.

Since we want to characterise behaviour preorders by using C-simulations it is reasonable to impose on these simulations the condition of being themselves behaviour preorders; that is guaranteed whenever the constraints are also behaviour preorders. Given that the operators in our basic algebra BCCSP are those generating finite trees, this condition is quite natural and the results we will prove based on it are indeed rather general.

Example 1. Let us briefly present several examples of constrained simulations, all of them corresponding to relations being behaviour preorders.

– Ordinary simulation is a constrained simulation taking as C the universal relation, xCy for every x and y.
– Ready simulation is just the I-constrained simulation, where $pIq \Leftrightarrow I(p) = I(q)$.
– Ready simulation is perhaps the most important C-constrained simulation but we can also achieve a greater discriminating power. Let us consider, for instance, the simulation preorder $\sqsubseteq_S$ and $C = \sqsubseteq_S^{-1}$; then 2-nested simulations [GV92] are just the corresponding class of C-constrained simulations.

One could argue that, whenever we admit a nearly arbitrary constraint when defining the constrained simulations, we are vitiating the local character of the notion of simulation, thus spoiling the coinductive nature of the generalisation. We do not agree with such an opinion for several reasons. First, we can still

consider local constraints that provide interesting results, such as I, as we are going to see. More in general, this family of constrained simulations will allow us to set a benchmark to compare and classify the complexity of process semantics, most of these semantics can be characterised with a local constraint, but others, such as 2-nested simulation, are intrinsically non-local (see [AFGI04]). To have a common framework to unify all these simulation semantics is very useful because we can prove general results for all of then as we present in the rest of this section.

C-constrained similarity, $\sqsubseteq_{\rightarrow}^{C}$, can be conditionally axiomatized in a simple way. For any constraint C we just need to consider the axiom

$$(P_C) \qquad xCy \Rightarrow x \sqsubseteq x + y$$

We define the axiomatization $\mathcal{P}_C$ as the set of axioms obtained by adding the axiom P_C to the set of axioms that characterises bisimulation equivalence (Figure 2), $\mathcal{P}_C = \{B_1, B_2, B_3, B_4, P_C\}$. As usual, we write $\mathcal{P}_C \vdash p \sqsubseteq q$ when the relation $p \sqsubseteq q$ is provable from $\mathcal{P}_C$ using the rules of inequational logic. $\mathcal{P}_C$ is sound and complete with respect to $\sqsubseteq_{\rightarrow}^{C}$.

(B_1)	$x + y = y + x$	(B_3)	$x + x = x$
(B_2)	$(x + y) + z = x + (y + z)$	(B_4)	$x + \mathbf{0} = x$

Fig. 2 Axiomatisation for the (Strong) Bisimulation Equivalence

Theorem 4. *For every constraint C being a behaviour preorder, we have that*

$$\mathcal{P}_C \vdash p \sqsubseteq q \Longleftrightarrow p \sqsubseteq_{\rightarrow}^{C} q$$

Proof. Soundness. Bisimilarity axioms are sound for both the relation C and for the C-constrained simulation preorder. Therefore, we only need to prove that the axiom (P_C) is also sound. Process $p + q$ can obviously simulate p and since we have pCq and C is a congruence with respect to choice, we also have $pC(q + p)$ and we conclude that $p \sqsubseteq_{\rightarrow}^{C} p + q$.

Completeness. By induction on the depth of processes. If $p = \mathbf{0}$ then $\mathbf{0}Cq$ and, applying $(P_C), (B_1)$ and (B_4), $\mathcal{P}_C \vdash \mathbf{0} \sqsubseteq q$. Consider now the general case $p = \sum a_i p_i$. On the one hand, if $p \sqsubseteq_{\rightarrow}^{C} q$ then pCq and we can use (P_C) to prove that $\mathcal{P}_C \vdash p \sqsubseteq p + q$. On the other hand, whenever $p \xrightarrow{a_i} p_i$ then $q \xrightarrow{a_i} q_{j_i}$ with $p_i \sqsubseteq_{\rightarrow}^{C} q_{j_i}$; by induction hypothesis we have $\mathcal{P}_C \vdash p_i \sqsubseteq q_{j_i}$, therefore we have $\mathcal{P}_C \vdash q + \sum a_i p_i \sqsubseteq q + \sum a_i q_{j_i}$, equivalently $\mathcal{P}_C \vdash q + p \sqsubseteq q$. Combining both cases, $\mathcal{P}_C \vdash p \sqsubseteq q + p \sqsubseteq q$.

We next study the axiomatization of the equivalence relation associated to the C-constrained simulation, $\rightleftarrows^{C} = \sqsubseteq_{\rightarrow}^{C} \cap \sqsupseteq_{\rightleftarrows}^{C}$. We propose the following axiom

for each constraint C:

$$(E_C) \qquad xCy \Rightarrow a(x+y) = a(x+y) + ay$$

We define the set $\mathcal{E}_C = \{B_1, B_2, B_3, B_4, E_C\}$, containing the axioms that characterise bisimulation equivalence (Figure 2) and the axiom E_C. We write $\mathcal{E}_C \vdash p = q$ when the equation $p = q$ is provable from $\mathcal{E}_C$.

$\mathcal{E}_C$ is sound and complete with respect to $\rightleftarrows^C$. However, in this case, to prove this result the constraint has to be symmetric, that is, it has to be a behaviour equivalence; below we will comment more on this subject.

Theorem 5. *For every constraint C being a behaviour equivalence, we have that*

$$\mathcal{E}_C \vdash p = q \Leftrightarrow p \rightleftarrows^C q$$

Proof. Soundness. Let us just prove that (E_C) is sound. Whenever pCq we also have $\mathcal{E}_C \vdash a(p+q) = a(p+q)+aq$. We have to prove that $a(p+q) \sqsubseteq^C_{\rightarrow} a(p+q)+aq$ and $a(p+q)+aq \sqsubseteq^C_{\rightarrow} a(p+q)$. Let us start by proving $a(p+q) \sqsubseteq^C_{\rightarrow} a(p+q)+aq$, processes in both sides of the relation can trivially simulate each other and taking into account that C is a behaviour preorder, from pCq we derive $a(p+q)\ C\ a(p+q) + aq$ and, we immediately conclude that we have a C-simulation.

Let us prove now that $a(p+q)+aq \sqsubseteq^C_{\rightarrow} a(p+q)$, as before, processes in both sides can simulate each other, to have a C-simulation we just need to prove $a(p+q)+aq\ C\ a(p+q)$. As before, from pCq we derive $a(p+q)\ C\ a(p+q)+aq$ and, since C is symmetric, we conclude $a(p+q)+aq\ C\ a(p+q)$.

Completeness. The proof of the completeness of the axiomatization of the simulation equivalence in [Gla01] (Section 17.2) can be transferred without any changes just checking the additional proof obligations imposed by the condition in the axiom (E_C).

It is interesting to note that the cases in which C is not symmetric are not completely excluded from the result above. This can be concluded from the following results.

Definition 9. We say that two constraints C_1 and C_2 are *cs-equivalent*, which we denote by $C_1 \sim C_2$, iff they define the same C-constrained similarity relation, that is $\sqsubseteq^{C_1}_{\rightarrow} = \sqsubseteq^{C_2}_{\rightarrow}$.

Next proposition is just a snapshot of a nice algebraic theory that can be developed around constrained simulations and cs-equivalence.

Proposition 1. *For any behaviour preorders C, C_1 and C_2 we have:*

1. $C_1 \sim C_2 \Rightarrow (C_1 \cap C_2) \sim C_1$.
2. $C \sim \sqsubseteq^C_{\rightarrow}$ and $\sqsubseteq^C_{\rightarrow}$ is the smallest C-simulation that is cs-equivalent to C.
3. If $C_1 \sim C_2$ and $C_1 \supseteq C \supseteq C_2$, then $C \sim C_1$.

4. For the simulation preorder $\sqsubseteq_S$ we have that $C \sim (C \cap \sqsubseteq_S)$.

Example 2. Next we show some illustrative examples of cs-equivalent constraints.

– Let us consider the classical simulation preorder, $\sqsubseteq_S$. $\sqsubseteq_S = \overset{U}{\underset{\rightarrow}{\sqsubseteq}}$ where U is the universal relation, xUy for every x and y. On the other hand, if we use $\sqsubseteq_S$ as constraint, it is immediate to see that $\sqsubseteq_S = \overset{\sqsubseteq_S}{\underset{\rightarrow}{\sqsubseteq}}$ and therefore $U \sim \sqsubseteq_S$, but while U is symmetric, $\sqsubseteq_S$ is not.
– Taking the constraint $I_\supseteq$ given by $pI_\supseteq q \Leftrightarrow I(p) \supseteq I(q)$, we have that the ready simulation preorder is $\overset{I_\supseteq}{\underset{\rightarrow}{\sqsubseteq}}$, as it was originally defined in [BIM95]. It is well known that the ready simulation preorder also coincides with $\overset{I}{\underset{\rightarrow}{\sqsubseteq}}$, see for instance [Gla01]. Again, I is a symmetric relation, $I_\supseteq$ is not, and $I \sim I_\supseteq$.
– In a similar way, we can define the 2-nested simulation as a constrained simulation using either $\sqsubseteq_S^{-1}$ or the equivalence relation $\sqsubseteq_S^{-1} \cap \sqsubseteq_S$.

From the examples above, one could guess that any constraint might be cs-equivalent to some symmetric one. This is indeed the case for any "interesting" constraint we have found, but in general it is not true, as the following counterexample shows.

Example 3. If we consider the behaviour preorder $\sqsubseteq$ defined by the axioms of bisimulation equivalence (Figure 2) together with the axiom $x \sqsubseteq x + aa$, where a represents any arbitrary action in *Act*, it can be checked that there is no symmetric constraint cs-equivalent to $\sqsubseteq$.

4 Constrained Simulations Up-to a Preorder

Starting from constrained simulations we define a general notion of constrained simulation up-to a preorder that will allow us to provide simulation-like characterisations for behaviour preorders.

Definition 10. Let $\sqsubseteq$ be a behaviour preorder, and C a relation over processes. We say that a binary relation S over processes is a *C-simulation up-to* $\sqsubseteq$, if $S \subseteq C$ (that is, $pSq \Rightarrow pCq$), 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 $p \overset{a}{\longrightarrow} p'_a$ there exist q', q'_a such that $q \sqsupseteq q' \overset{a}{\longrightarrow} q'_a$ and $p'_a S q'_a$;
– pCq.

We say that process p is C-simulated up-to $\sqsubseteq$ by process q, or that process q C-simulates process p up-to $\sqsubseteq$, written $p \overset{C}{\underset{\sqsubseteq}{\lesssim}} q$, if there exists a C-simulation up-to $\sqsubseteq$, S, such that pSq.

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

The following proposition highlights the tight relation between a behaviour preorder and its kernel equivalence.

Proposition 2. *Given a behaviour preorder $\sqsubseteq$ and a behaviour equivalence C such that $\sqsubseteq_{\rightarrow}^{C} \subseteq \sqsubseteq \subseteq C$, we have $p \sqsubseteq q \iff q \equiv q + p \land pCq$.*

Proof. First we prove the right to left implication. Given that processes p and q satisfy the constraint C, we can apply the axiomatic characterisation for $\sqsubseteq_{\rightarrow}^{C}$ in Theorem 4, and using the axiom (P_C) we obtain $p \sqsubseteq_{\rightarrow}^{C} p + q$. Now, since $\sqsubseteq_{\rightarrow}^{C} \subseteq \sqsubseteq$, we also have $p \sqsubseteq p + q \equiv q$ and therefore $p \sqsubseteq q$.

We now prove the left to right implication. On the one hand, since $\sqsubseteq$ is a behaviour preorder we have $p \sqsubseteq q \Rightarrow p + q \sqsubseteq q$. On the other hand, if $p \sqsubseteq q$ then pCq and also qCp, since C is symmetric. As before, we use the axiom for C-similarity to obtain $q \sqsubseteq_{\rightarrow}^{C} q + p$. Since $\sqsubseteq_{\rightarrow}^{C} \subseteq \sqsubseteq$, we conclude $q \sqsubseteq q + p$.

Next we show that whenever p is C-simulated up-to $\sqsubseteq$ by q we also have $q \equiv q + p$. Using this lemma and the previous proposition we will prove later our Theorem 6.

Lemma 1. *For every behaviour preorder $\sqsubseteq$ and every behaviour equivalence C, such that $\sqsubseteq_{\rightarrow}^{C} \subseteq \sqsubseteq \subseteq C$, we have $p \sqsubseteq_{\sim\sqsubseteq}^{C} q \Rightarrow q \equiv q + p$*

Proof. We prove it by induction on the depth of process p. Since $\equiv$ is a congruence with respect to the choice operator, it is enough to show that $p \sqsubseteq_{\sim\sqsubseteq}^{C} q \Rightarrow q \equiv q + p|_a$ for every $a \in I(p)$. Whenever $p \xrightarrow{a} p'_a$ then there exist q_a and q'_a such that $q \sqsupseteq q_a \xrightarrow{a} q'_a$ with $p'_a \sqsubseteq_{\sim\sqsubseteq}^{C} q'_a$, so that $p'_a C q'_a$ and, applying the induction hypothesis, $q'_a \equiv q'_a + p'_a$. Since $\sqsubseteq_{\rightarrow}^{C} \subseteq \sqsubseteq$, we can use the axiomatization given in Theorem 5 and apply the axiom (E_C) to obtain $p'_a C q'_a \Rightarrow a q'_a \equiv a(q'_a + p'_a) + a p'_a$; therefore $a q'_a \equiv a q'_a + a p'_a$. Hence we get $\sum a q'_a \equiv \sum a q'_a + p|_a$. On the other hand, $q_a = a q'_a + r_a$ and since $\equiv$ is a congruence with respect to choice $\sum a q'_a + \sum r_a \equiv \sum a q'_a + \sum r_a + p|_a$, that is, $\sum q_a \equiv \sum q_a + p|_a$. We can also add q in both sides, getting $\sum q_a + q \equiv \sum q_a + q + p|_a$. Now, applying Proposition 2 to $\sqsubseteq$, since for every q_a we have $q_a \sqsubseteq q$, we conclude $q \equiv q + q_a$, and therefore $q \equiv q + p|_a$, as we wanted to prove.

Theorem 6. *For every behaviour preorder $\sqsubseteq$ and every behaviour equivalence C such that $\sqsubseteq_{\rightarrow}^{C} \subseteq \sqsubseteq \subseteq C$, we have $p \sqsubseteq_{\sim\sqsubseteq}^{C} q \Leftrightarrow p \sqsubseteq q$.*

Proof. The right to left implication is obvious: we have pCq and if $p \xrightarrow{a} p'_a$ then we can take $q \sqsupseteq p \xrightarrow{a} p'_a$. To prove the left to right implication we use Lemma 1, $p \sqsubseteq_{\sim\sqsubseteq}^{C} q \Rightarrow q \equiv q + p$ and, since pCq is also satisfied, we can now apply Proposition 2, to conclude $p \sqsubseteq_{\sim\sqsubseteq}^{C} q \Rightarrow p \sqsubseteq q$.

The condition of symmetry imposed to C is necessary, as the following counterexample shows.

Example 4. Let us consider the non-symmetric constraint $I_{\supseteq}$ defined in Example 2. Let us consider the preorder $\sqsubseteq$ defined by the axioms that define the ready simulation preorder (that is, the axioms that characterise bisimulation equivalence plus the axiom $ax \sqsubseteq ax + ay$), together with the axiom $x + bb \sqsubseteq x$. The constrained simulation $\overset{I_{\supseteq}}{\underset{\rightarrow}{\sqsubseteq}}$ is the ready simulation preorder and thus it is immediate to check both $\overset{I_{\supseteq}}{\underset{\rightarrow}{\sqsubseteq}} \subseteq \sqsubseteq$ and $\sqsubseteq \subseteq I_{\supseteq}$. However this preorder does not satisfy the thesis of Theorem 6, because if we consider the processes $p = a + bb$ and $q = a + b$, we have $p \not\sqsubseteq q$ but $p \overset{I_{\supseteq}}{\underset{\sqsubseteq}{\sim}} q$, because $I(p) = I(q)$ and if $p \overset{a}{\longrightarrow} \mathbf{0}$ then $q \overset{a}{\longrightarrow} \mathbf{0}$, while for $p \overset{b}{\longrightarrow} b$ we can take $q \sqsupseteq a + bb + b \overset{b}{\longrightarrow} b$.

Theorem 6 generalises Theorem 2 and 3 (and also Theorem 44 in [dFG08]). And what is even more important, it provides a uniform framework that allows to better understand the role of the premises in the previous results we had on simulations up-to. Thanks to the use of constrained simulations up-to, now we can clearly see that a great deal of semantic preorders both coarser and finer than the ready simulation preorder can be characterised by using simulations up-to. In particular, we can characterise not only all the preorders coarser than the ready simulation that appear in Van Glabbeek's linear time-branching time spectrum but also the rest of semantics there. Next example illustrates the case of possible-futures semantics that is not coarser than the ready simulation and therefore falls outside the scope of Theorem 3.

Example 5. If we denote by $\sqsubseteq_{PF}$ the possible-futures preorder [RB81, Gla01], then, taking the constraint $pTq \Leftrightarrow \mathrm{traces}(p) = \mathrm{traces}(q)$, we have that T is a behaviour equivalence and that $\overset{T}{\underset{\rightarrow}{\sqsubseteq}} \subseteq \sqsubseteq_{PF} \subseteq T$. Thus we are under the hypothesis of Theorem 6 and therefore $\overset{T}{\underset{\sqsubseteq_{PF}}{\sim}}$ and $\sqsubseteq_{PF}$ are the same relation.

Once again, the next result shows the interplay between preorders and the induced equivalences. For a given preorder, we could use the induced equivalence relation $\equiv$ to characterise the preorder by means of a C-simulation up-to $\equiv$.

Theorem 7. *For every behaviour preorder $\sqsubseteq$ and its induced equivalence relation $\equiv$, for every behaviour equivalence C such that $\overset{C}{\underset{\rightarrow}{\sqsubseteq}} \subseteq \sqsubseteq \subseteq C$, we have*

$$p \overset{C}{\underset{\sqsubseteq}{\sim}} q \Leftrightarrow p \overset{C}{\underset{\equiv}{\sim}} q.$$

Proof. The right to left implication is obvious. The left to right implication is a consequence of Proposition 2; whenever q would reduce into q' by applying $q \sqsupseteq q'$ we could also reduce it by $\equiv$, by applying $q \equiv q + q'$, and then we could execute all the transitions of q'.

5 Constrained Simulations up-to an Equivalence

The previous section was devoted to the study of constrained simulations up-to
a preorder. The starting point there was a given preorder, instead in this sec-
tion we show that the theory of constrained simulations up-to can be developed
even if we do not have such a preorder to start from. It is true that equiva-
lence relations are particular cases of preorders but equivalences are symmetric
relations and cannot be characterised by means of proper simulations that are
intrinsically non-symmetric. An interesting result that we present in this section
is how to use our up-to technique to build up an adequate preorder for a given
equivalence relation.

Lemma 2. *For every behaviour equivalence $\equiv$, and for every constraint C that
is a behaviour equivalence such that $\overrightarrow{\rightleftarrows}^C \subseteq \equiv$, we have $p \sqsubseteq_{\equiv}^C q \Rightarrow q \equiv q + p$.*

*Proof. The proof uses the same notations and follows similar arguments to those
in the proof of Lemma 1. We use induction on the depth of process p. Since $\equiv$
is a congruence with respect to the choice operator, it is enough to show that
$p \sqsubseteq_{\equiv}^C q \Rightarrow q \equiv q + p|_a$ for every $a \in I(p)$. Whenever $p \xrightarrow{a} p_a'$ then $q \equiv q_a \xrightarrow{a} q_a'$
and $p_a' \sqsubseteq_{\equiv}^C q_a'$, and by applying the induction hypothesis we obtain $q_a' \equiv q_a' + p_a'$.
On the other hand, since $\overrightarrow{\rightleftarrows}^C \subseteq \equiv$, we can use the axiomatic characterisation
given in Theorem 5 and apply the axiom (E_C) to obtain $aq_a' \equiv a(q_a' + p_a') + ap_a'$;
therefore $aq_a' \equiv a(q_a' + p_a') + ap_a' \equiv aq_a' + ap_a'$. Adding all up $\sum aq_a' \equiv \sum aq_a' + p|_a$.
Since $\equiv$ is a congruence with respect to the choice operator we can add subterms
in both sides of the equivalence, in particular, every $q_a = aq_a' + r_a$ and therefore
$\sum aq_a' + \sum r_a \equiv \sum aq_a' + \sum r_a + p|_a$, that is, $\sum q_a \equiv \sum q_a + p|_a$. Since for all
q_a we have $q \equiv q_a$, then we conclude that $q \equiv q + p|_a$.*

Theorem 8. *For every behaviour equivalence $\equiv$, and every constraint C that
is a behaviour equivalence such that $\overrightarrow{\rightleftarrows}^C \subseteq \equiv \subseteq C$, we have $p \sqsubseteq_{\equiv}^C q \wedge p \sqsupseteq_{\equiv}^C q \Leftrightarrow
p \equiv q$.*

*Proof. The right to left implication is obvious, just considering that $\equiv \subseteq C$.
We prove the left to right implication. Since $p \sqsubseteq_{\equiv}^C q$ then, by Lemma 2, we have
that $q \equiv q + p$; symmetrically, from $q \sqsubseteq_{\equiv}^C p$ we get $p \equiv q + p$ and therefore $p \equiv q$.*

As desired, for any equivalence relation fulfilling the hypothesis of the the-
orem above, we get a preorder such that its kernel is the original equivalence.
Moreover, this preorder satisfies some interesting properties.

Proposition 3. *For every behaviour equivalence $\equiv$, and for every constraint C
that is a behaviour equivalence such that $\overrightarrow{\rightleftarrows}^C \subseteq \equiv \subseteq C$, we have that $\sqsubseteq_{\equiv}^C$ is a
behaviour preorder and $\sqsubseteq_{\rightarrow}^C \subseteq \sqsubseteq_{\equiv}^C \subseteq C$.*

Proof. That $\sqsubseteq^C_{\sim\equiv}$ is a precongruence with respect to the choice operator follows from the congruence with respect to the choice of $\equiv$. The rest of the properties are immediate.

Now we can say that for any behaviour equivalence fulfilling the hypothesis of Theorem 8, the preorder $\sqsubseteq^C_{\sim\equiv}$ is canonical in the sense specified in the following result.

Theorem 9. *For every behaviour equivalence $\equiv$, and for every constraint C that is a behaviour equivalence, such that $\rightleftarrows^C \subseteq \equiv \subseteq C$, the preorder $\sqsubseteq^C_{\sim\equiv}$ is the only behaviour preorder that satisfies $\sqsubseteq^C_{\rightarrow} \subseteq \sqsubseteq^C_{\sim\equiv} \subseteq C$, and whose kernel is $\equiv$. Therefore, it can be said to be the* canonical preorder under the constraint C *that induces the equivalence $\equiv$.*

Proof. Proposition 3 says that $\sqsubseteq^C_{\sim\equiv}$ satisfies the hypothesis of the results in Section 4. On the other hand, if there is any other behaviour preorder $\sqsubseteq$ such that $\equiv = \sqsubseteq \cap \sqsupseteq$, then by applying Theorem 6 and 7 we conclude $\sqsubseteq \Leftrightarrow \sqsubseteq^C_{\sim\sqsubseteq} \Leftrightarrow \sqsubseteq^C_{\sim\equiv}$.

The canonicity of the preorder $\sqsubseteq^C_{\sim\equiv}$ is bounded by the constraint C appearing in its definition. For instance, trace equivalence $\equiv_T$ satisfies $\rightleftarrows^U \subseteq \equiv_T$, and therefore we can obtain the U-canonical preorder $\sqsubseteq^U_{\sim\equiv_T}$, which is just the classic preorder $\sqsubseteq_T$. But we also have $\rightleftarrows^I \subseteq \equiv_T \subseteq I$, and then we also consider the I-canonical preorder $\sqsubseteq^I_{\sim\equiv_T}$ whose kernel is also trace equivalence, but it is strictly finer than $\sqsubseteq_T$. Then we could even conclude that we should not call canonical to these generated preorders. We have decided to maintain this term because it is true that given both the equivalence $\equiv$ and the constraint C, then the generated preorder $\sqsubseteq^C_{\sim\equiv}$ is unique indeed.

Anyway, if we want to associate to an equivalence a unique canonical preorder we can define it as the C-canonical preorder with the coarsest constraint C. That is, the coarsest C such that $\rightleftarrows^C \subseteq \equiv \subseteq C$ that generates the preorder $\sqsubseteq^C_{\sim\equiv}$. We have not been able to prove the existence of such a coarser constraint for any arbitrary behaviour equivalence. However, if we restrict ourselves to the semantics in the linear time-branching time spectrum, it is easy to see that such a coarsest constraint exists.

Proposition 4. *Let $\mathcal{O} \in \{T, S, CT, CS, F, R, FT, RT, PW, RS, 2N, PF\}$ be any of the semantics in the linear time-branching time spectrum [Gla01]. Then, there exists a coarsest constrained $C_{\mathcal{O}}$ such that $\rightleftarrows^{C_{\mathcal{O}}} \subseteq \equiv_{\mathcal{O}} \subseteq C_{\mathcal{O}}$ and that is defined in Table 1.*

Proof. We prove some of the results, the rest can be proved in a similar way.

T *Since $\rightleftarrows^U \subseteq \equiv_T$, we have immediately $C_T = U$.*

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

$$pUq \;\forall p,q \qquad\qquad pWq \iff p \equiv_T q$$
$$pVq \iff (p = \mathbf{0} \Leftrightarrow q = \mathbf{0}) \qquad pXq \iff p \equiv_S q$$
$$pIq \iff I(p) = I(q)$$

Table 1 Coarsest Constraints for the Semantics in the ltbt Spectrum

F We have $\rightleftarrows^I \subseteq\; \equiv_F \;\subseteq I$. We know that $\rightleftarrows^I$ can be axiomatized by $pIq \Rightarrow a(p + q) \equiv a(p + q) + ap$. If we consider any other $\rightleftarrows^C \subseteq\; \equiv_F \;\subseteq C$ with $\rightleftarrows^I \not\subseteq \rightleftarrows^C$ we should have $I \not\subseteq C$ and therefore there should be some processes such that $p'Cq'$ and some action $b \in I(q') - I(p')$. From the axiomatization of $\rightleftarrows^C$ we would obtain $a(p' + q') \equiv a(p' + q') + ap'$ but these two processes are not failure equivalent, since the right one can reject $\{b\}$ after executing a, and the leftone cannot.

$2N$ As for the other simulation semantics, $\mathcal{O}_{2N}$ is just the constraint used in the definition.

PF We have $\rightleftarrows^T \subseteq\; \equiv_{PF} \;\subseteq T$. As above, if we have another constraint $\rightleftarrows^T \not\subseteq \rightleftarrows^C$ we should have some processes such that $p'Cq'$, and therefore $a(p' + q') \equiv a(p' + q') + ap'$, but $T(p) \neq T(q)$. This is not possible because a possible future for the process in the right side after executing a is $T(p')$, that is not a possible future for the process on the left side.

As a byproduct, we have detected a new simulation semantics that does not appear in the linear time-branching time spectrum, the T-constrained simulation, which we could call trace equivalence simulation semantics. This new semantics should be added in the spectrum between ready simulation and 2-nested simulation, and above possible futures.

We can take advantage of the close relation between any behaviour equivalence and the corresponding canonical preorder by turning an axiomatic characterisation of the former into an axiomatization of the latter.

Theorem 10. Let $\equiv$ be a behaviour equivalence and C a constraint that is a behaviour equivalence such that $\rightleftarrows^C \subseteq\; \equiv\; \subseteq C$. If A_E is an axiomatization of the equivalence $\equiv$, taking the axiom (P_C) to be $xCy \Rightarrow x \sqsubseteq x + y$, we have that $A_P = A_E \cup \{P_C\}$ is an axiomatization of the relation $\sqsubseteq^C_{\sim_\equiv}$.

Proof. As usual we write $A_P \vdash p \sqsubseteq q$ when the inequality $p \sqsubseteq q$ is provable from the set of axioms A_P. We prove that $A_P \vdash p \sqsubseteq q$ iff $p \sqsubseteq^C_{\sim_\equiv} q$.

Soundness. On the one hand, $\equiv\;\subseteq\sqsubseteq^C_{\sim_\equiv}$; on the other hand, from Proposition 3 we know that $\sqsubseteq^C_{\rightarrow} \subseteq \sqsubseteq^C_{\sim_\equiv} \subseteq C$; besides, from Theorem 4 we have that P_C is one of the axioms that characterise $\sqsubseteq^C_{\rightarrow}$.

Completeness. By Lemma 2 we know that $p \sqsubseteq^C_{\sim_\equiv} q \Rightarrow q \equiv q + p$ and therefore $A_P \vdash q \equiv q + p$. On the other hand, we have also pCq and then, using P_C, we get $A_P \vdash p \sqsubseteq p + q$. All together, $A_P \vdash p \sqsubseteq p + q \equiv q$.

With this result, the correctness and completeness of the axiomatization for the preorders are proved once and for all for a great variety of semantics.

6 Conclusions and Future Work

In this paper we have universalised the presentation of our theory of simulation up-to by means of which we provide coinductive characterisations for a great variety of semantics either coarser or finer than the ready simulation, including those in the linear time-branching time spectrum. Constrained simulations have played an essential role in our development; we have provided an axiomatization of the preorders defined by them and also of the induced equivalences.

An interesting result was that any behaviour equivalence induces a canonical preorder whose kernel is the given equivalence relation. It is nice to find that for all the semantics in the linear time-branching time spectrum the so obtained canonical preorder coincides with the one we already knew from the literature. As a consequence of the canonicity, some properties can be proved in general, once and for all. We have illustrated this fact by giving a general axiomatization of the preorders in terms of the axiomatization of the equivalences and the axiomatization of the constrained simulations.

There are several directions in which we plan to continue the study of the relations between preorders and equivalences. For instance, the axiomatization of the preorders that we have obtained is conditional. It would be interesting to know in which cases the axiom (P_C) can be turned into an equivalent finite collection of equational axioms. This is in fact the case for the semantics in the linear time-branching time spectrum coarser than the ready simulation, for which there exists an equational axiomatization equivalent to our conditional axiomatization.

In [AFIL05] you can find a review of (in)axiomatizability results, that its authors have recently completed giving other new results on the subject. We think that our characterisation of the semantics using constrained simulations up-to will be useful to get other new (in)axiomatizability general results. For instance, we conjecture that any interesting preorder stronger than the ready simulation is not finitely axiomatizable (see [AFGI04] for the seed results supporting this conjecture).

References

[AFGI04] Luca Aceto, Wan Fokkink, Rob van Glabbeek, and Anna Ingólfsdóttir. Nested semantics over finite tree are equationally hard. *Information and Computation*, 191(2):203–232, 2004.

[AFI07]   Luca Aceto, Wan Fokkink, and Anna Ingólfsdóttir. Ready to preorder: get your BCCSP axiomatization for free! In *CALCO'07*, volume 4624 of *Lecture Notes in Computer Science*, pages 65–79. Springer, 2007.

[AFIL05] Luca Aceto, Wan Fokkink, Anna Ingólfsdóttir, and Bas Luttik. Finite equational bases in process algebra: Results and open questions. In *Processes, Terms and Cycles*, volume 3838 of *LNCS*, pages 338–367. Springer, 2005.

[BIM95]   Bard Bloom, Sorin Istrail, and Albert R. Meyer. Bisimulation can't be traced. *Journal of the ACM*, 42(1):232–268, 1995.

[dFG05]   David de Frutos-Escrig and Carlos Gregorio-Rodríguez. Bisimulations up-to for the linear time-branching time spectrum. In *CONCUR 2005*, 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 *SOS 2007*, volume 192 of *ENTCS*, pages 13–28. Elsevier, 2007.

[dFG08]   David de Frutos-Escrig and Carlos Gregorio-Rodríguez. (Bi)simulations up-to characterise process semantics. *Information and Computation*, (to appear), 2008.

[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.

[GV92]   Jan Friso Groote and Frits Willem Vaandrager. Structured operational semantics and bisimulations as a congruence. *Information and Computation*, 100(2):202–260, 1992.

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

[Par81]   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.

[Plo81]   Gordon D. Plotkin. A structural approach to operational semantics. Tech. Report DAIMI FN-19, Comp. Sci. Dept., Aarhus University, 1981.

[RB81]   William Rounds and Stephen Brooks. Possible futures, acceptances, refusals, and communicating processes. In *22nd Foundations of Computer Science Annual Symposium*, pages 140–149. IEEE, 1981.