Information and Computation 207 (2009) 146–170

Contents lists available at ScienceDirect

Information and Computation

journal homepage: www.elsevier.com/locate/ic

(Bi)simulations up-to characterise process semantics☆

David de Frutos Escrig [1], Carlos Gregorio Rodríguez *,[2]

Departamento de Sistemas Informáticos y Computación, Facultad de Ciencias Matemáticas, Universidad Complutense. Plaza de Ciencias 3, 28040 Madrid, Spain

ARTICLE INFO

Article history:
Received 15 February 2007
Revised 26 December 2007
Available online 14 November 2008

Keywords:
Semantic equivalences
Semantic preorders
Bisimulations up-to
Simulations up-to
Canonical preorder
Linear time-branching time spectrum

ABSTRACT

We define (bi)simulations up-to a preorder and show how we can use them to provide a coinductive, (bi)simulation-like, characterisation of semantic (equivalences) preorders for processes. In particular, we can apply our results to all the semantics in the linear time-branching time spectrum that are defined by preorders coarser than the ready simulation preorder.

The relation between bisimulations up-to and simulations up-to allows us to find some new relations between the equivalences that define the semantics and the corresponding preorders. In particular, we have shown that the simulation up-to an equivalence relation is a canonical preorder whose kernel is the given equivalence relation. Since all of these canonical preorders are defined in an homogeneous way, we can prove properties for them in a generic way. As an illustrative example of this technique, we generate an axiomatic characterisation of each of these canonical preorders, that is obtained simply by adding a single axiom to the axiomatization of the original equivalence relation. Thus we provide an alternative axiomatization for any axiomatizable preorder in the linear time-branching time spectrum, whose correctness and completeness can be proved once and for all.

Although we first prove, by induction, our results for finite processes, then we see, by using continuity arguments, that they are also valid for infinite (finitary) processes.

1. Introduction and related work

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 [3], CCS [4] or ACP [5]. But, besides these classic ones, along the years a great variety of process semantics have been proposed under different settings and from quite dissimilar points of view. The comparative study of concurrency semantics tries to shed light on this heterogeneous field to bring up differences and similarities that will allow to order and classify the variety of semantics, in spite of the different ways they are defined.

Clearly, the thorough work of van Glabbeek is a cornerstone in the field of comparative concurrency semantics. In [6] he presents the well known linear time-branching time spectrum for processes without internal transitions. There he presented a quite extensive collection of semantics, each of which was characterised by a natural testing scenario, a modal logic to identify the set of equivalent processes, and a finite axiomatization (whenever that was possible) that allows to develop a pure algebraic study of the generated equivalence relation between pairs of finite processes. Fig. 1 shows these axiomatized semantics (but for the tree semantics) ordered by inclusion.

☆ Part of the contents of this paper appeared in [1,2].
* Corresponding author.
E-mail addresses: defrutos@sip.ucm.es (D. de Frutos Escrig), cgr@sip.ucm.es (C. Gregorio Rodríguez).
[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 MCyT Project WEST TIN2006-15578-C02.

0890-5401/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.ic.2007.12.003

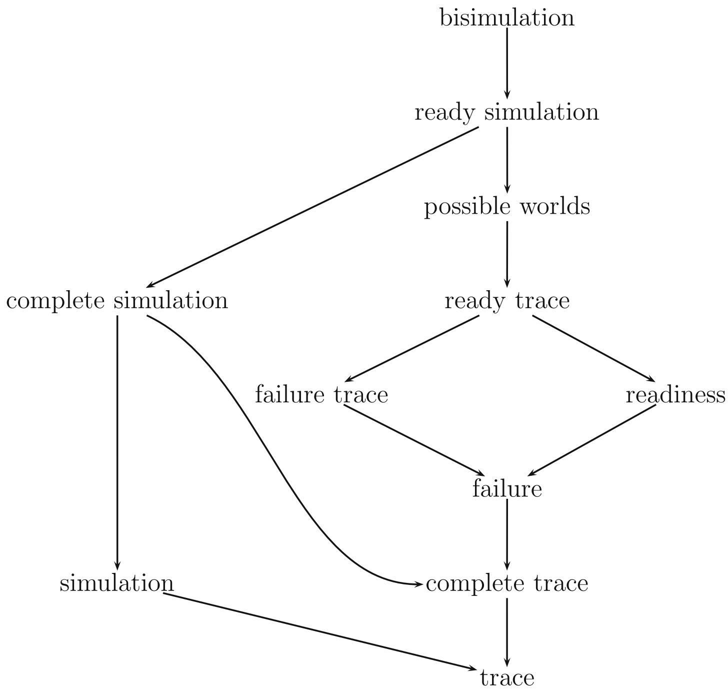


Fig. 1. Axiomatic semantics in the linear time-branching time spectrum I.

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

Moreover, a semantics can be also defined by a preorder which compares pairs of processes in a natural way. These can be easily generated whenever we have a testing scenario or a modal logic characterising the semantics, simply saying that a process is *better* than another when it passes more tests or, equivalently, when it satisfies more formulas of the logic. Certainly, 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.

These order relations between processes have also interesting applications by themselves when they correspond to relations such as “is an implementation of” [7] , “is faster than” [8], or “has less amortised cost than” [9]. 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.

In [6] both equivalences and preorders have been 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 [10,11]. Besides, the inclusion order between semantics corresponds to the different expressive power of the families of tests defining each of them, and as a consequence it is the same for both behaviour preorders and equivalences.

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.

Bisimulation can also be presented as a game [12,13], and this provides a fruitful metaphor: by playing the game of bisimulation an attacker can check that two processes are not bisimilar in a finite number of steps; however, if the attacker has no strategy to win the game, the two processes are bisimilar. It is also characterised by a simple and natural logic, the well known Hennessy–Milner Logic (HML) [14]. Finally, bisimilarity can be easily established either by means of explicit bisimulations described in a symbolic way or, in the case of finite state processes, by an efficient algorithm [15,16] based on which several tools that can effectively check process bisimilarity [17] have been developed.

Despite the fact that bisimulation has been thoroughly studied since it was proposed by David Park [18] (see [19] for a recent historic presentation on the subject), it is still the topic of quite a number of recent papers such as [20–22].

However, sometimes bisimulation equivalence is too strong, and many other interesting semantics weaker than bisimilarity have been proposed, most of them appearing in the linear time-branching time spectrum. Traces, for instance, is the weakest reasonable semantics for processes, it just collects the sequences of actions that can be executed by a process. However, non-deterministic behaviours are not properly described by means of traces, since deadlock information is not accurately captured. Failure semantics was proposed in [3] to solve this problem. An even finer semantics is that defined by readiness [23], where we keep count of the sets of offerings at each state of a process. Failures and ready sets can be combined with traces, thus getting stronger semantics as described in [6].

Most of the semantics in Fig. 1 are extensional [24] and none of them has a symmetric, coinductive definition as bisimulation does. It is true that all of the simulation semantics (simulation, ready simulation and so on) are intensional and quite close to bisimulation, but the induced equivalences are just the kernel of the corresponding preorder and do not admit a direct single symmetric definition. Could these semantics be somehow characterised by a symmetric definition? And for the other extensional semantics? Could they be expressed in a coinductive way? Could we also characterise the corresponding preorders?

As we will see in this paper, we can indeed do that. In order to characterise the equivalence relations, all of them coarser than bisimilarity, we weaken the definition of bisimulation by using a preorder relation, to obtain what we call *bisimulation up-to* that preorder. In this way we propose a family of coinductively and symmetrically defined equivalences, parameterised by the preorders. As main results we prove that, under quite sensible assumptions on the considered preorder, bisimulation up-to such a preorder defines exactly the same equivalence as the kernel of the preorder does. These results are quite general and can be applied to all the semantics in Fig. 1 (and beyond), so that we get symmetric, coinductive, bisimulation-like definitions for nearly any reasonable semantics.

One may think that the same could we done for the preorders arising from simulations instead of bisimulations, but that is far from true: bisimilarity is the strongest of the semantic equivalences and thus by relaxing the bisimulation requirements we 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 finer 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 [25,26] 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, and this is why it can be used to characterise all of them by means of *I-simulations up-to* a preorder, that weaken ready simulations in the adequate way.

Once we have got coinductive characterisations of both behaviour equivalences and preorders we have been able to find several interesting connections between bisimulations up-to and simulations up-to which provide us with new relations between behaviour equivalences and preorders. One of these results was quite unexpected for us, but also extremely nice: we have found that 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.

Certainly, given an equivalence relation there are many different preorders, including the equivalence itself, whose kernel is the given equivalence relation. It is nice to get a natural characterisation of a canonical one among them, particularly if, as we have proved for all of the equivalences in van Glabbeek's spectrum that are coarser than the ready simulation, our canonical preorders define the same order relations as the ones in the literature. This has been the origin of many pleasant properties. In particular, we can obtain a complete axiomatization for finite processes for any of these preorders from the corresponding axiomatization for the equivalences in a systematic way, so that the completeness and correctness of these axiomatizations can be proved once and for all, thus avoiding the necessity of repeated proofs as those presented in [6].

It has come as a nice surprise for us to know that in [27] it has been found the way to establish the opposite relation between the axiomatizations of the preorders that are weaker than the ready simulation and those of the corresponding equivalences, for the semantics in the van Glabbeek's spectrum. We agree with the authors of that paper on the fact that it 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. We comment more about this subject in the conclusion of the paper.

Concerning the related work, we can find in [28,29] recursive definitions of testing semantics which can be considered a first step in the direction we aim, but in both cases the authors used the *after* construction in their characterisations, which means a too global approach. Instead, we want a more local characterisation where bisimulation steps mainly will solve, as usual, the initial choices of each pair of related processes. This idea also inspired some of our former works on this subject [30,13], where we use our so called *global bisimulation*, in order to get a symmetric bisimulation-like characterisation of the ready similarity and other classical semantics. These global bisimulations were previously used, in a different context, in [31].

There have been indeed some other previous approaches to the problem of getting coinductive characterisations of extensional semantics. Most of them study the question in a pure coalgebraic framework [32,33,34,35] and, in many cases, are based on relatively complex categorical concepts. These works aim generality and their results are indeed rather general. This is why the machinery needed to apply them, even in some simple cases, can be rather complex. Instead, our results, at least as presented here, can only be applied to transition systems, but they are quite simple to state and to apply.

In [36] Boreale and Gadducci have defined a fully abstract model for the failures semantics based on the novel concept of *behavioural differential equations*, introduced by Rutten [37]. However, the extension of their results to cover other semantics seems not easy.

A different approach is presented in [38] where the author uses predicate transformers to get a variant of the bisimulation equivalence that gives rise to both trace and failure preorders. However, for each of these preorders an ad-hoc construction is needed and it is not clear how to extend it to cover other semantics. Certainly there are further connections between our own work and most of these quite recent papers that we plan to explore in the future.

To conclude this introductory section we outline the contents that appear in the rest of the paper. In Section 2 we collect the essential definitions and notations on processes and semantic preorders and equivalences that we are going to use along the paper.

Section 3 is devoted to the study of the coinductive characterisations for the semantic equivalences. We define bisimulations up-to a preorder and present the results that are illustrated with some examples in order to clarify the role of the conditions in the theorems.

In Section 4, we change the focus from the equivalences to the semantic preorders. We define simulations up-to and prove some results that characterise behaviour preorders with simulations up-to a preorder. In Section 4.3 we comment on some connections between the results developed in Sections 3 and 4.

In Section 5 we continue with the theory of simulations up-to but this time we want to study the semantic equivalences. We show how we can characterise equivalences as the kernel of simulations up-to. From that, we identify a canonical coinductive preorder whose kernel is a given equivalence relation.

Section 6 is rather technical. It is devoted to show how the results obtained in the previous sections can be extended to infinite finitary processes. For that, we define a proper notion of approximation and then, we use the Approximation Induction Principle [39] and standard continuity techniques to prove the desired results.

To illustrate the relevance of the theory of (bi)simulations up-to we show in Section 7 some examples of applications of the coinductive characterisations that we have proposed for semantic preorders and equivalences. In particular, we provide alternative axiomatic definitions of the preorders in the linear time-branching time spectrum. The proof of their completeness is easy and simple, and can be done once for all.

Finally, in Section 8 we present some conclusions and some lines for future work.

We would like to express our gratitude to Miguel Palomino for his comments, that helped us to improve the presentation of this paper.

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 [40] (reprinted in [41]).

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 $p_0 \in \mathcal{P}$.

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, $I(p) = \{a \mid a \in Act \text{ and } p \xrightarrow{a}\}$.

LTSs for finite processes are just directed graphs which become finite trees[3] if expanded. These finite trees can be syntactically described by the basic process algebra BCCSP, which was also used, for instance, in [6,1].

Definition 2. Given a set of actions Act, the set of BCCSP processes is defined by the following BNF-grammar:

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

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.

So we have that BCCSP is just the initial algebra for the signature $(\mathbf{0}, a \in Act, +)$. The set of rooted LTSs is another algebra for this signature, by defining prefix and choice operators in the natural way.

All the definitions we present in the paper are valid for arbitrary processes, that is, for arbitrary rooted LTSs, either finite or infinite. However, the proofs that we provide in Sections 3–5 make extensive use of inductive reasonings and therefore they are only valid for BCCSP processes, that is, for finite processes. However, as we will show in Section 6, by using the

[3] We obtain directly a tree if we generate the states on the fly introducing a new state for each transition generated by the application of the rules defining the operational semantics, see for instance [4].

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

Fig. 2. Operational semantics for BCCSP terms.

Approximation Induction Principle [39], we can extend all our results first to infinite depth finitely branching tree processes, and then to arbitrary finitely branching rooted 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 Fig. 2. The depth of a BCCSP process is the depth of the tree it denotes.

Trailing occurrences of the constant **0** are omitted: we write a instead of $a\mathbf{0}$. As usual (see for instance [6]), 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 $q \sqsubseteq p$.

Table 1, borrowed from [6], shows a complete axiomatization for some of the semantics in the ltbt 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 the upper left of Table 1 characterise bisimilarity, that is, the bisimulation equivalence, that we denote by $=_B$. They also belong to any other axiomatic characterisation 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 (see Definition 33) is characterised by the four axioms of the bisimulation equivalence plus the axiom $(RS)\ ax \sqsubseteq ax + ay$. Analogously, axioms for bisimulation together with axiom $(S)\ x \sqsubseteq x + y$, characterise the simulation preorder.

The next quite simple definitions introduce a couple of common properties that are fulfilled by most of the natural semantics preorders, and will be extensively used along the rest of the paper.

Definition 3. A preorder relation $\sqsubseteq$ over processes is a *behaviour preorder* if

- it is weaker than bisimilarity, 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
Axiomatization for the preorders in the linear time-branching time spectrum.

	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
....	...	...	...	...	...	...	...	...	...	...	...
$x \sqsubseteq x+y$										+	+
$ax+ay = a(x+y)$											+

Definition 4. 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 $a \in I(p)$.

To be exact, factorisation is satisfied by any of the behaviour preorders in the ltbt spectrum, while any preorder finer than the trace preorder is initials preserving.

3. Up-to characterisation of semantic equivalences

In Section 2 the behaviour of processes is described in terms of the actions they can perform, so it is natural to define the process equivalence in terms of these action transitions. That is precisely what bisimulations do: they inductively explore the intensional behaviour of processes. Bisimulation was introduced in [18] and it has become one of the fundamental notions in the theory of concurrent processes. It is defined as follows:

Definition 5 (*[4]*)**.** A binary relation $\mathcal{R}$ is called a (strong) *bisimulation* if for all p, q processes such that $p \mathcal{R} q$, and for all $a \in Act$, the following properties are satisfied:

- Whenever $p \xrightarrow{a} p'$, there exists some q' such that $q \xrightarrow{a} q'$ and $p' \mathcal{R} q'$.
- Whenever $q \xrightarrow{a} q'$, there exists some p' such that $p \xrightarrow{a} p'$ and $p' \mathcal{R} q'$.

Two processes p and q are *bisimilar*, notation $p =_B q$, if there exists a bisimulation containing the pair $\langle p,q \rangle$.

Let us recall that the definition imposes *simultaneous simulations* by means of a single symmetrical definition of bisimulations. If, instead, *separated simulations* are considered, the induced equivalence relation, that we call mutual simulation, is weaker than bisimulation equivalence (see [6] for details).

In [4], in order to make bisimilarity easier to decide, Milner introduced the notion of *bisimulation up-to* (strong) bisimilarity. This is a useful technique, but care must be taken when generalising it. It is well known that the original (simple and natural!) definition of weak bisimulation up-to weak bisimulation, that appeared in [4], was wrong. Later, in [42] two new up-to (now correct, but more involved!) techniques were proposed. Sangiorgi continued the study of up-to techniques in [43], but focusing on reducing the size of the bisimulation relations to prove that two given processes are bisimilar. Recent work continuing with the study of the subject can be found in [44,45].

However, instead of capturing bisimilarity we want to use the idea of bisimulations up-to to characterise coarser equivalences in a coinductive way. This is done by using the adequate preorder in the up-to part of the definition.

3.1. Bisimulations up-to a preorder

Using the game view of bisimulation, bisimulations up-to are defined by allowing the defending player to remove some of the future capabilities of the process where he makes his move to mimic the move of the other player. This is formalised by a reduction with respect to the considered behaviour preorder.

Definition 6. 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.

It is obvious that the introduction of the preorder generalises the original definition of bisimulation, so that we have now more chances to prove the equivalence between two processes by giving a bisimulation up-to that relates them. When the behaviour preorder is just the identity relation, or even the bisimilarity relation, we just get bisimilarity itself, but, as we are going to prove below, considering other behaviour preorders leads to other interesting semantics (traces, failures, ready simulation and so on).

For the sake of simplicity, we often drop the subscript and use $\eqsim$ instead of $\eqsim_{\sqsubseteq}$ when the behaviour preorder is clear from the context.

Proposition 7. *For every behaviour preorder* $\sqsubseteq$, *if* $p \equiv q$ *then* $p \eqsim q$.

Proof. If $p \equiv q$ then $p \sqsubseteq q$ and $q \sqsubseteq p$. For every transition $p \xrightarrow{a} p'_a$, then $q \sqsupseteq p \xrightarrow{a} p'_a$ and, symmetrically, for every transition $q \xrightarrow{a} q'_a$, then $p \sqsupseteq q \xrightarrow{a} q'_a$. □

Example 8. Let us consider the processes t and v in Fig. 3 and let $\sqsubseteq_S$ be the simulation preorder. Clearly, processes t and v are not bisimilar, $t \neq_B v$, but they are bisimilar up-to the simulation preorder, $t \precsim_{\sqsubseteq_S} v$. The only difficult point to get a bisimulation up-to between t and v corresponds to the case when v starts executing a and evolves into $v' = bc + b$. Then t can be reduced to abc, since $abc \sqsubseteq_S t$, and then performing the action a the process evolves into $t' = bc$. Now, by using the fact that $b \sqsubseteq_S bc$ one can check in a similar way that v' and t' are bisimilar up-to the simulation preorder, and finish the proof.

Lemma 9. *For every initials preserving behaviour preorder* $\sqsubseteq$ *, if* $p \precsim q$ *then* $I(p) = I(q)$.

Proof. It is enough to show that $I(p) \subseteq I(q)$. For any $a \in I(p)$, since $q \sqsupseteq q' \xrightarrow{a} q'_a$, $a \in I(q')$, and therefore $a \in I(q)$, due to the initials preservation property of $\sqsubseteq$. □

Theorem 10. *For every behaviour preorder* $\sqsubseteq$ *that is initials preserving, action factorised and satisfying the axiom* (RS)*, we have that* $p \precsim q$ *if and only if* $p \equiv q$.

Proof. If $p \equiv q$ then $p \precsim q$ is proved in Proposition 7. We prove the reverse implication, if $p \precsim q$ then $p \equiv q$. We proceed by induction on the depth of process p and prove that if $p \precsim q$ then $p \sqsubseteq q$.

By definition of $p \precsim q$, if $p \xrightarrow{a} p'_a$ then $q \sqsupseteq q' \xrightarrow{a} q'_a$ and $p'_a \precsim q'_a$. By induction hypothesis $p'_a \equiv q'_a$, in particular it is also true that $p'_a \sqsubseteq q'_a$, and, since $\sqsubseteq$ is a precongruence, $ap'_a \sqsubseteq aq'_a$. On the other hand, $q \sqsupseteq q'$ and, since the order $\sqsubseteq$ is action factorised we obtain $q|_a \sqsupseteq q'|_a$.

We would like to establish the order relation between $q'|_a$ and aq'_a. In fact, $q'|_a = aq'_a + r$, and given that $I(q'|_a) = \{a\}$ we also have $I(r) = \{a\}$. Then we can use the axiom (RS) $ax + ay \sqsupseteq ax$, to conclude that $q'|_a \sqsupseteq aq'_a$. All together:

$$ap'_a \sqsubseteq aq'_a \sqsubseteq q'|_a \sqsubseteq q|_a$$

Considering now the general definition of $p = \sum_i \sum_j a_i p_{ij}$, we can write for every i and j the following sequence of relations

$$a_i p_{ij} \sqsubseteq a_i q^{ij}_{a_i} \sqsubseteq q^{ij}|_{a_i} \sqsubseteq q|_{a_i}$$

and therefore

$$p = \sum_i \sum_j a_i p_{ij} \sqsubseteq \sum_i q|_{a_i}$$

Finally, by Lemma 9, $I(p) = I(q)$ and we conclude that $\sum_i q|_{a_i} = q$ and therefore $p \sqsubseteq q$. □

This result, if simple, is rather general: all the preorders for the semantics weaker than the ready simulation in Fig. 1 satisfy the axiom (RS) and therefore the corresponding bisimulations up-to characterise each equivalence. That is, this theorem provides a symmetric, bisimulation-like characterisation for every equivalence in the linear time-branching time spectrum from trace equivalence to ready simulation equivalence.

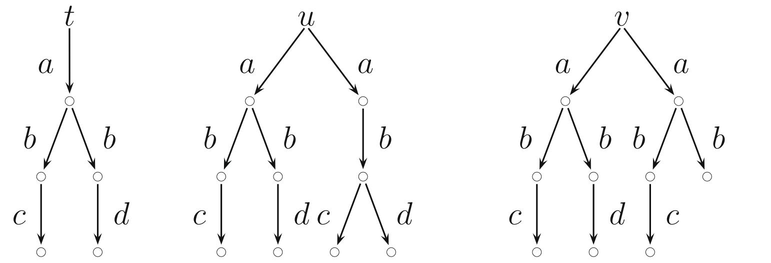
Example 11. Let us revisit our Example 8 and consider the processes t and v in Fig. 3. Since the simulation preorder satisfies the conditions of Theorem 10, the fact that $t \precsim v$ is enough to conclude that t and v are simulation equivalent. By applying Theorem 10 we have been able to prove it by exhibiting a single bisimulation up-to instead of two simulations, one for each of $t \sqsubseteq_S v$ and $v \sqsubseteq_S t$.

The conditions imposed to the behaviour preorders in Theorem 10 suggest that not every preorder is adequate to get the induced equivalence by means of a bisimulation up-to. This is indeed the case. Let us first consider an example that shows that the condition of being initials preserving is necessary.

Example 12. Let us consider the behaviour preorder defined by the following axiom: $p + q \sqsubseteq p$. This preorder relation is the inverse of the simulation preorder $(\sqsubseteq_S)$ and therefore its kernel is also the simulation equivalence. However, bisimulation up-to $\sqsubseteq$ is far from being equal to the simulation equivalence. In fact it relates any two processes: for every p and q whenever $p \xrightarrow{a} p'$, $q \sqsupseteq q + p \xrightarrow{a} p'$ and conversely, whenever $q \xrightarrow{a} q'$, $p \sqsupseteq p + q \xrightarrow{a} q'$.

We have not contradicted Theorem 10 because the preorder $\sqsubseteq$ is not initials preserving. Now we see that it is also necessary that the preorders satisfy the axiom (RS).

Example 13. Let us consider the behaviour preorder induced by the axiom $a(p + q) \sqsubseteq ap + aq$. Obviously, by definition, this relation is action factorised and initials preserving. Let us consider the processes t and u in Fig. 3. Let us take $t' = bc + bd$ and $u' = b(c + d)$. It is true that $u \sqsubseteq t$ $(t = a(bc + bd) \sqsubseteq a(bc + bd) + a(bc + bd) \sqsubseteq a(bc + bd) + ab(c + d) = u)$, but $t \not\sqsubseteq u$, because the application of the axiom only allows to take choices earlier, but never to delay them as in the right subprocess of u. However, t and u are bisimilar up-to $\sqsubseteq$:



$$a(bc + bd) \quad a(bc + bd) + ab(c + d) \quad a(bc + bd) + a(bc + b)$$

Fig. 3. Examples of processes.

- Any action transition of t can be trivially simulated by u because t is a subprocess of u.
- If u performs action a and evolves into t', then t can trivially simulate that move.
- If u performs action a and evolves into u', then t can delay its choice and reduce to $ab(c + d)$, then performing action a, evolving also into u'.

As in the previous example, we have neither contradicted Theorem 10 because in this case the preorder $\sqsubseteq$ does not satisfy the axiom (*RS*).

The following example shows us that we can neither weaken the axiom (RS) by considering particular cases like the axiom $ap + aq \sqsubseteq ap + aq + a(p + q)$.

Example 14. Let us consider the preorder $\sqsubseteq$ induced by the axiom $ap + aq \sqsubseteq ap + aq + a(p + q)$. If we consider the processes $z = a(bc + bd + b(c + d))$ and $w = a(bc + bd) + z$ and we take $z' = bc + bd + b(c + d)$ and $w' = bc + bd$, we have that $z \not\sqsubseteq w$ but z and w are bisimilar up-to $\sqsubseteq$:

- Any action transition of z can be trivially simulated by w because z is a subprocess of w.
- If w performs action a and evolves into z', then z can trivially simulate that move.
- If w performs action a and evolves into w', then z transforms itself according to the axiom that defines the preorder and becomes $a(bc + bd)$, then performing action a, evolves into z'.

3.2. Bisimulation up-to an equivalence

As a straightforward corollary of our main result in Section 3.1, we get that for any of the equivalence relations defined by the semantics in Fig. 1, it is also true that $\precsim_{\equiv}$ is equal to $\equiv$.

Corollary 15. *For every behaviour preorder $\sqsubseteq$ that is initials preserving, action factorised and satisfies the axiom (RS), and the induced equivalence relation $\equiv$, we have $\equiv \,=\, \precsim_{\sqsubseteq} \,=\, \precsim_{\equiv}$.*

Proof. For any processes p and q we have that $p \equiv q \Rightarrow p \precsim_{\equiv} q \Rightarrow p \precsim_{\sqsubseteq} q$. Finally, Theorem 10 proves that $p \precsim_{\sqsubseteq} q \Rightarrow p \equiv q$. □

One could ask why we introduced the characterisations of behaviour equivalences by means of bisimulation up-to behaviour preorders instead of just proving those up-to the equivalences themselves. There are two reasons for that: the first and most important was that bisimulations up-to preorders are more general, and therefore more powerful, when trying to establish the equivalence between two processes by presenting a bisimulation up-to that relates them. Besides, the fact that behaviour preorders satisfy the axiom (*RS*) is used in the proof of Theorem 10, and a direct proof of Corollary 15 without using that more general result seemed hard to find. However, encouraged by a referee of a previous version of this paper we looked for such a direct proof and finally got the result stated in Theorem 17. But first we are going to extend, in a natural way, the definition of behaviour preorders to equivalence relations.

Definition 16. 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 $p + r \equiv q + r$.

Theorem 17. *For every behaviour equivalence* $\equiv$, *we have* $\equiv = \precsim_{\equiv}$.

Proof. The fact that whenever we have $p \equiv q$ we also have $p \precsim_{\equiv} q$ is a direct application of Proposition 7 just using the fact that any behaviour equivalence is a behaviour preorder whose kernel is again itself. Let us now take $p = \sum_i a_i p_i$ and $q = \sum_j b_j q_j$. By definition of $p \precsim q$, if $p \xrightarrow{a_i} p_i$ then $q \equiv q^i \xrightarrow{a_i} q_i$ and $p_i \precsim q_i$. This means that there exists some process r_i such that $q^i = a_i q_i + r_i$. Moreover, by induction hypothesis, we have $p_i \equiv q_i$. Reasoning in a symmetric way starting from $q \xrightarrow{b_j} q_j$, we get $p \sqsupseteq p^j \xrightarrow{b_j} p_j$ and $q_j \precsim p_j$, with some process s_j such that $p^j = b_j p_j + s_j$, and by induction hypothesis $q_j \equiv p_j$. Then we have $p \equiv \sum_j p^j = \sum_j b_j p_j + s_j \equiv \sum_j b_j q_j + s_j$ and $q \equiv \sum_i q^i = \sum_i a_i q_i + r_i \equiv \sum_i a_i p_i + r_i$. Now replacing each one of these equivalences into the other, and using the idempotence of $\equiv$, we get $p \equiv \sum_j s_j + \sum_i r_i + p + q$ and $q \equiv \sum_i r_i + \sum_j s_j + p + q$, thus obtaining $p \equiv q$. □

3.3. Characterising equivalences finer than ready simulation

The range from trace equivalence to ready simulation equivalence is quite wide and most of the classic semantics fall into it. However, there are still some interesting process semantics out of it. For instance, those constrained simulations as the one in Example 22, where the defining constraint is finer than condition I, or the nested simulation semantics.

We have studied whether the use of bisimulations up-to a preorder is also possible fot these semantics. We have found that there is another family of semantic preorders for which bisimulations up-to work properly. Any preorder in this family is a simulation, so that we recall right now its definition. Simulations will be also the main topic of the rest of the sections of the paper.

Definition 18. A binary relation S over processes is a *simulation*, if pSq implies that:

- For every a, if $p \xrightarrow{a} p'$ there exists q', $q \xrightarrow{a} q'$ and $p'Sq'$.

We say that process p is simulated by process q, or that q simulates p, written $p \sqsubseteq_S q$, whenever there exists a simulation S such that pSq.

Lemma 19. *For every behaviour preorder* $\sqsubseteq$ *being a simulation, whenever* $p \sqsupseteq p' \xrightarrow{a} p'_a$, *there exists* p_a *such that* $p \xrightarrow{a} p_a \sqsupseteq p'_a$.

Proof. By definition of simulation. □

Next, in order to obtain the results that follow, we introduce a property that one could consider quite technical and a bit ad hoc, but in fact it is quite natural, and therefore satisfied by most of the semantics for concurrent processes in the literature.

Definition 20. We say that a behaviour preorder $\sqsubseteq$ has the *Hoare equivalence property*[4] (HE for short) whenever it satisfies:

$$\left.\begin{array}{l}\text{If for all } p \xrightarrow{a} p' \text{ there exists } q', \text{ such that } q \xrightarrow{a} q' \text{ and } p' \sqsubseteq q' \\ \text{and for all } q \xrightarrow{a} q' \text{ there exists } p', \text{ such that } p \xrightarrow{a} p' \text{ and } q' \sqsubseteq p'\end{array}\right\} \text{ then } p \equiv q.$$

For behaviour preorders that are simulations and satisfy the Hoare Equivalence property, we have the following result:

Theorem 21. *For every behaviour preorder* $\sqsubseteq$, *being a simulation and satisfying the Hoare equivalence property, we have that* $p \precsim q$ *if and only if* $p \equiv q$.

Proof. If $p \equiv q$ then $p \precsim q$ is proved by Proposition 7. The reverse implication, if $p \precsim q$ then $p \equiv q$, is proved by induction on the depth of the first process.

Let us consider $p \precsim q$. Then, whenever $p \xrightarrow{a} p'_a$ there exist q' and q'_a such that $q \sqsupseteq q' \xrightarrow{a} q'_a$ and $p'_a \precsim q'_a$ and, by induction hypothesis, $p'_a \equiv q'_a$. As the behaviour preorder is a simulation, by Lemma 19 there exists q_a such that $q \xrightarrow{a} q_a \sqsupseteq q'_a$. Therefore, for some process r it is true that $q = aq_a + r \sqsupseteq aq'_a + r \equiv ap'_a + r$. That is, for every $p \xrightarrow{a} p'_a$ there exists q_a such that $q_a \sqsupseteq p'_a$.

Symmetrically, we can prove that for every $q \xrightarrow{a} q'_a$ there exists p_a such that $p_a \sqsupseteq q'_a$. These are the premises for the HE property that our behaviour preorder satisfies, and so we conclude that $p \equiv q$. □

[4] The name comes from Hoare's powerdomain construction.

Both the simulation preorder and the ready simulation preorder are simulations and satisfy the HE property, so for these preorders Theorem 21 provides an alternative proof to that of Theorem 10. But there are other interesting preorders that induce equivalences between strong bisimulation and ready simulation equivalence for which Theorem 21 provides a characterisation in terms of bisimulation up-to.

Example 22. Let us consider the preorder $\sqsubseteq_{FS}$ defined as $p \sqsubseteq_{FS} q$ if there exists a binary relation S over processes such that pSq implies

- for every $a, p \xrightarrow{a} p'$, there exists $q', q \xrightarrow{a} q'$ and $p'Sq'$,
- $F(p) = F(q)$,

where $F(p) = \{(a,X) \mid a \in I(p), X \subset Act, p \xrightarrow{a} p' \text{ and } X \cap I(p') = \emptyset\}$.

That is, $\sqsubseteq_{FS}$ is much like the ready simulation preorder, but instead of checking the equality of initial actions, we check the equality of the *failures* immediately below the root of the processes.

The preorder $\sqsubseteq_{FS}$ satisfies the conditions to apply Theorem 21: obviously it is a simulation and it can be easily checked that it satisfies the HE property. Therefore, bisimulation up-to $\sqsubseteq_{FS}$ defines the same equivalence relation as $\sqsubseteq_{FS} \cap \sqsubseteq_{FS}^{-1}$.

The equivalence induced by the preorder $\sqsubseteq_{FS}$ is strictly finer than ready simulation equivalence. To see that, let us consider, for instance, the processes $p = a(bc + bd)$ and $q = abc + a(bc + bd)$, that clearly are ready simulation equivalent, but $q \not\sqsubseteq_{FS} p$, since we also have $bc \not\sqsubseteq_{FS} bc + bd$, because $(b,\{c\}) \in F(bc + bd) - F(bc)$.

Following the ideas in the previous example it is quite easy to find other *constrained* simulations in the conditions of Theorem 21 that define equivalences between the ready simulation and strong bisimilarity. Some of them can be defined axiomatically in an easy way, as in the following example:

Example 23. Let us consider the behaviour preorder defined by the axiom $a(p + q) \sqsubseteq a(p + q) + ap$. This axiom refines the axiom (RS) and therefore defines a simulation. Besides it satisfies the HE property: Let $p = \sum_i a_i p_i$ and $q = \sum_j b_j q_j$ that verify the provisos in the definition of the property. Then for each index i we have some summand of q, $a_i q_i$, with $p_i \sqsubseteq q_i$; and symmetrically, for each index j we have some summand of p, $b_j q_j$, with $q_j \sqsubseteq p_j$.

By iterating this reasoning in a ping-pong way using the finiteness of the terms, we finally get for each index i some maximal summands of p, $a_i' p_i'$, and q, $a_i' q_i'$, with respect to the order $\sqsubseteq$, with $a_i = a_i'$, $p_i' \equiv q_i'$ and $p_i \sqsubseteq p_i'$. Symmetrically, for each index j we get some maximal summands of p $b_j' p_j'$ and q $b_j' q_j'$, with respect to the order $\sqsubseteq$, with $b_j = b_j'$, $q_j' \equiv p_j'$ and $q_j \sqsubseteq q_j'$.

If we take $q' = \sum_i a_i q_i'$ we obviously have $p \sqsubseteq q'$. It is also easy to check that taking $q'' = \sum_j b_j q_j'$ we have $q' = q''$. Let us finally check that $q'' \sqsubseteq q$. We only need to see that for each index j we have $b_j q_j' \sqsubseteq b_j q_j' + b_j q_j$. This is a consequence of the following general result: whenever we have $q \sqsubseteq p$ we have also $ap \sqsubseteq ap + aq$. This is because $\sqsubseteq$ verifies that $p \sqsubseteq q \Rightarrow q \equiv p + q$, what can be proved by proof induction, using the form of the axiom defining $\sqsubseteq$. Then we have $ap \sqsubseteq a(p + q) \sqsubseteq a(p + q) + aq \equiv ap + aq$, and thus we conclude $p \sqsubseteq q$, and symmetrically we would also get $q \sqsubseteq p$, and finally $p \equiv q$, proving that $\sqsubseteq$ verifies the property HE. This means that we can apply our Theorem 21, thus getting $p \precsim q$ if and only if $p \equiv q$.

The next example points out the necessity of the HE property in the conditions of Theorem 21. It is interesting to see that the considered order $\sqsubseteq$ is also a simulation order, defined by an axiom quite similar to that defining Example 23 above. However, in this case the order does not verify the property HE, and the corresponding bisimulation up-to $\precsim$ is strictly coarser than the induced equivalence $\equiv$.

Example 24. Let us consider the axiom $ap \sqsubseteq ap + a(p + q)$ and the induced behaviour preorder. This preorder refines the axiom of the simulation preorder but it does not satisfies the HE property. We will see that there exist some pairs of processes which are not related by the induced equivalence relation, but however are bisimilar up-to that preorder. For instance, let us consider $m = a(bc + b(c + d)) + abc$ and $n = a(bc + b(c + d))$, we have that $m \sqsubseteq n$ because $bc \sqsubseteq bc + b(c + d)$ and therefore $abc \sqsubseteq n$, so that we conclude $abc + n \sqsubseteq n$, and thus $m \sqsubseteq n$. However, we have $n \not\sqsubseteq m$ because the summand abc of m cannot be generated by applying the axiom $ap \sqsubseteq ap + a(p + q)$, since it only allows to introduce new summands that expand those in the original process, n in this case. But m and n are bisimilar up-to $\sqsubseteq$:

- m can trivially simulate n.
- If m performs action a and evolves into $bc + b(c + d)$ then n can trivially simulate that move.
- If m performs action a and evolves into bc then n can be reduced by the preorder to abc, and then performing a, it evolves into bc.

The processes m and n also illustrate that the preorder $\sqsubseteq$ does not satisfy the property HE, since as seen before m can trivially simulate n, and if m performs action a and evolves into bc then n can perform a evolving into $a(bc + b(c + d))$, that satisfies $abc \sqsubseteq a(bc + b(c + d))$.

4. Up-to characterisations of semantic preorders

Section 3 provides quite general and interesting results about the coinductive characterisation of a semantic equivalence, both in terms of the equivalence itself, and in terms of the behaviour preorder that generates the equivalence. But in order to complete the study of the subject, in the following sections we will look for coinductive characterisations of these behaviour preorders. As we mentioned in the introduction, preorders are even more important than equivalences to provide semantics of process algebras, and therefore finding coinductive characterisations for them we would have a new tool for the study of these semantics.

When we first addressed the problem of finding a coinductive characterisation for process equivalences we had a clear starting point: bisimulation equivalence. Bisimulation is the strongest equivalence and therefore by weakening its definition (Definition 6) we could obtain weaker semantics (Theorem 10). To find out coinductive characterisations for the semantic preorders is not such an easy task. Certainly the first idea is to start from the simulation preorder, even if it is not finer than most of the semantic preorders in the spectrum, and therefore we could never characterise these by weakening the definition of simulation in any way, as we made for the equivalences in Section 3. However, it is reasonable to start from plain simulations to characterise the behaviour preorders coarser than it, and then to look for the adequate way to characterise the rest of finitely axiomatizable preorders in the linear time-branching time spectrum.

4.1. Simulations up-to

By modifying the definition of simulation (Definition 18) in the same way as we did for bisimulations in Definition 6 we define simulations up-to a preorder.

Definition 25. 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 such that $q \sqsupseteq q' \xrightarrow{a} q'_a$ 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.

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

Example 26. Let us consider the processes $s = a(b(d + e) + cd)$ and $t = abf + a(be + bd + cd)$ in Fig. 4. It is clear that for the simulation preorder $\sqsubseteq_S$ 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 $s \lesssim_{\sqsubseteq_T} t$, 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 evolving into $t' = be + bd + cd$. Now we have to check that $s' \lesssim_{\sqsubseteq_T} t'$: 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 (see Table 1), $t' \sqsupseteq_T b(e + d) + cd$, and then action b is executed to arrive to $d + e$ as well, thus completing the verification of the simulation up-to obligations.

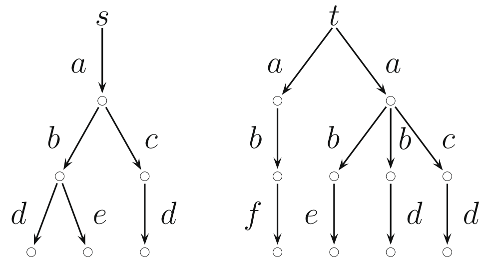
Certainly, if we know in advance that $s \sqsubseteq_T t$, we could directly reduce t into s when checking $s \lesssim_{\sqsubseteq_T} t$, but what we want to illustrate here is how we would use in practice our coalgebraic characterisation: we do not want to use any complicate information about the corresponding 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 27. *For every behaviour preorder* $\sqsubseteq$, *if* $p \sqsubseteq q$ *then* $p \lesssim_{\sqsubseteq} q$.

Proof. If $p \sqsubseteq q$ then for every $p \xrightarrow{a} p'_a$ we have $q \sqsupseteq p \xrightarrow{a} p'_a$. □

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, $\sqsubseteq_S$.



$$a(b(d+e)+cd) \quad abf + a(be+bd+cd)$$

Fig. 4. A pair of processes.

Theorem 28. *For every behaviour preorder* $\sqsubseteq$ *that satisfies the axiom* (S), *we have* $p \precsim_{\sqsubseteq} q$ *if and only if* $p \sqsubseteq q$.

Proof. The right to left implication was proved in Proposition 27. To prove the left to right implication we proceed by induction on the depth of process p.

For $p = \mathbf{0}$ and any q we immediately have $\mathbf{0} \sqsubseteq q$, since $\sqsubseteq$ satisfies (S).

In the inductive case, by definition of $\precsim$, if $p \xrightarrow{a} p'_a$ then $q \sqsupseteq q' \xrightarrow{a} q'_a$ and $p'_a \precsim q'_a$. By induction hypothesis $p'_a \sqsubseteq q'_a$, and since $\sqsubseteq$ is a precongruence, $ap'_a \sqsubseteq aq'_a$.

Now, as $q' \xrightarrow{a} q'_a$ then there exists some process r such that $q' = aq'_a + r$, and applying the axiom (S) we obtain $aq'_a \sqsubseteq q'$, and therefore $ap'_a \sqsubseteq q$. Hence, for every $p \xrightarrow{a} p'_a$, it holds that $ap'_a \sqsubseteq q$ and by adding all the summands of p we conclude $p \sqsubseteq q$. □

As said before, the given preorder must satisfy the axiom (S) since we have the following simple proposition.

Proposition 29. *Any behaviour preorder* $\sqsubseteq$ *that satisfies* $\sqsubseteq = \precsim_{\sqsubseteq}$ *is coarser than the simulation preorder*.

Proof. By applying Definition 25 we have that any simulation is also a simulation up-to any behaviour preorder $\sqsubseteq$, and therefore $\sqsubseteq \subseteq \precsim_{\sqsubseteq}$. □

Theorem 28 characterises semantic preorders in the same way that semantic equivalences were characterised in Theorem 10, though in both cases we use preorders for the up-to relation. It would be nice to have a dual characterisation where the equivalences were used to characterise the semantic preorder. That is indeed possible, as stated in the following results.

Proposition 30. *For every behaviour preorder* $\sqsubseteq$ *that satisfies the axiom* (S), *we have that* $p \sqsubseteq q \Rightarrow p \precsim_{\equiv} q$.

Proof. Let us first prove that if $p \sqsubseteq q$ then $q \equiv q + p$. We have $p \sqsubseteq q \Rightarrow p + q \sqsubseteq q$, and since $\sqsubseteq$ satisfies (S) we also have $q \sqsubseteq q + p$.

Now we can use this equivalence, and whenever we have $p \sqsubseteq q$ and $p \xrightarrow{a} p'_a$, we have $q \equiv p + q \xrightarrow{a} p'_a$, and the simulation up-to condition is established. □

We can now state a result similar to that in Corollary 15, relating simulation up-to a preorder and simulation up-to an equivalence.

Corollary 31. *For every behaviour preorder* $\sqsubseteq$ *that satisfies the axiom* (S), *and the induced equivalence relation* $\equiv$, *we have that the relations* $\sqsubseteq$, $\precsim_{\sqsubseteq}$ *and* $\precsim_{\equiv}$ *are the same*.

Proof. Direct consequence of the application of Theorem 28, Proposition 30 and the fact that $\precsim_{\equiv} \subseteq \precsim_{\sqsubseteq}$. □

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

Corollary 32. *For every behaviour preorder $\sqsubseteq$ that satisfies the axiom (S), and the induced equivalence relation $\equiv$, the following correspondences hold*:

$$
\begin{array}{ccc}
p \equiv q & \Leftrightarrow & p \sqsubseteq q \wedge p \sqsupseteq q \\
\Updownarrow & & \Updownarrow \\
p \eqsim_{\sqsubseteq} q & \Leftrightarrow & p \underset{\sim}{\sqsubset}_{\sqsubseteq} q \wedge p \underset{\sim}{\sqsupset}_{\sqsubseteq} q \\
\Updownarrow & & \Updownarrow \\
p \eqsim_{\equiv} q & \Leftrightarrow & p \underset{\sim}{\sqsubset}_{\equiv} q \wedge p \underset{\sim}{\sqsupset}_{\equiv} q
\end{array}
$$

Proof. The equivalences in the second column are direct consequences of Corollary 31. The remainder equivalences follow from these inclusions: $\equiv \subseteq \eqsim_{\sqsubseteq}$, $\equiv \subseteq \eqsim_{\equiv}$, $\eqsim_{\sqsubseteq} \subseteq (\underset{\sim}{\sqsubset}_{\sqsubseteq} \cap \underset{\sim}{\sqsupset}_{\sqsubseteq})$ and $\eqsim_{\equiv} \subseteq (\underset{\sim}{\sqsubset}_{\equiv} \cap \underset{\sim}{\sqsupset}_{\equiv})$. □

Considering the semantics in the ltbt spectrum, (only) trace and simulation preorders (see Table 1) satisfy the axiom (S) and thus fulfil the hypothesis of Corollary 32. 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 characterisations of each of these preorders and four alternative characterisations of the induced equivalences.

4.2. I-simulations up-to

The results in Section 4.1 for semantic preorders, are quite interesting, but fall short of the generality that we achieved in Section 3 when dealing with semantic equivalences.

This limitation comes from the fact that the definition of $\underset{\sim}{\sqsubset}$ 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 28, 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 [6]. 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 $pIq \Leftrightarrow I(p) = I(q)$. Then we recall the definition of ready simulations in [26].

Definition 33 (*[26]*)**.** A binary relation $\mathcal{R}$ on processes is called a *ready simulation* if for all p, q such as $p \mathrel{\mathcal{R}} q$, and for all $a \in Act$, the following properties are satisfied:

- Whenever $p \xrightarrow{a} p'$ there exists some q' such that $q \xrightarrow{a} q'$ and $p' \mathrel{\mathcal{R}} q'$.
- pIq.

Two processes are ready similar, what we denote by $p =_{RS} q$, if there exists a ready simulation $\mathcal{R}$ with $p \mathrel{\mathcal{R}} q$ ($p \sqsubseteq_{RS} q$) and also a ready simulation $\mathcal{S}$ such that $q \mathrel{\mathcal{S}} p$ ($q \sqsubseteq_{RS} p$).

Starting now from ready simulations instead of plain simulations we define next our I-simulations up-to that we will study in the rest of this section.

Definition 34. 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 $p'_a S q'_a$.
- 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 $p \underset{\sim}{\sqsubset}^{I}_{\sqsubseteq} q$, if there exists an I-simulation up-to $\sqsubseteq$, S, such that pSq.

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

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

Proposition 35. *For every preorder $\sqsubseteq$ such that $\sqsubseteq \subseteq I$, if $p \sqsubseteq q$ then $p \precsim^I_{\sqsubseteq} q$.*

Proof. Let us see that $\sqsubseteq$ is an I-simulation. Since $\sqsubseteq \subseteq I$ we only have to check the other I-simulation condition, but if $p \xrightarrow{a} p'_a$ we have $q \sqsupseteq p \xrightarrow{a} p'_a$. □

Now we can use I-simulations up-to to prove a similar result to that in Theorem 28, for semantic preorders with more discriminating power than the simulation. Note that all the pairs of processes related by any preorder relation ranging from failure preorder to ready simulation preorder (see Table 1) in the linear time-branching time spectrum satisfy the I condition.

Theorem 36. *For every behaviour preorder $\sqsubseteq$ that satisfies the axiom (RS) and $\sqsubseteq \subseteq I$, we have $p \precsim^I_{\sqsubseteq} q$ if and only if $p \sqsubseteq q$.*

Proof. The idea behind the proof is that whenever $p \xrightarrow{a} p'_a$ then $q \sqsupseteq q' \xrightarrow{a} q'_a$ and by induction hypothesis $p'_a \sqsubseteq q'_a$; therefore, roughly speaking, $p = \sum ap'_a \sqsubseteq \sum aq'_a$. The process $\sum aq'_a$ is not q, as we would wish, but fortunately it has the necessary form to apply the (RS) axiom and then we can add all the missing subterms to conclude that $p \sqsubseteq \sum aq'_a \sqsubseteq q$.

The right to left implication, if $p \sqsubseteq q$ then $p \precsim^I q$, is proved in Proposition 35. For the left to right side we proceed by induction on the depth of process p.

If $p = \mathbf{0}$ then $I(p) = \emptyset = I(q)$ and thus $q = \mathbf{0}$. If $p = \sum_a \sum_i ap^i_a$, by definition of $\precsim^I$, if $p \xrightarrow{a} p^i_a$ then $q \sqsupseteq q'^i_a \xrightarrow{a} q^i_a$ and $p^i_a \precsim^I q^i_a$. By applying the induction hypothesis, $p^i_a \sqsubseteq q^i_a$ and, since $\sqsubseteq$ is a precongruence, $ap^i_a \sqsubseteq aq^i_a$ and therefore $p = \sum_a \sum_i ap^i_a \sqsubseteq \sum_a \sum_i aq^i_a$. Moreover, $q'^i_a = aq^i_a + r^i_a$ and since $\sqsubseteq \subseteq I$ we have $I(p) = I(q) = I(\sum_a \sum_i q'^i_a) = I(\sum_a \sum_i aq^i_a + r^i_a)$ and therefore we can use (RS) to add all the subterms r^i_a: $\sum_a \sum_i aq^i_a \sqsubseteq \sum_a \sum_i aq^i_a + r^i_a = \sum_a \sum_i q'^i_a$. We can now conclude that $p \sqsubseteq \sum_a \sum_i aq^i_a \sqsubseteq \sum_a \sum_i q'^i_a \sqsubseteq q$. □

As happened for plain simulations up-to, I-simulations up-to can only characterise behaviour preorders that are coarser than the ready simulation preorder. Besides, by definition, we must also have $\sqsubseteq \subseteq I$.

Proposition 37. *Any behaviour preorder $\sqsubseteq$ that satisfies $\sqsubseteq = \precsim^I_{\sqsubseteq}$ is coarser than the ready simulation preorder and must be included in the relation I.*

Proof. By applying Definition 34 we have that any ready simulation is also an I-simulation up-to any behaviour preorder $\sqsubseteq$, and therefore $\sqsubseteq_{RS} \subseteq \precsim^I_{\sqsubseteq} = \sqsubseteq$. Besides, I-simulations only relate pairs of processes in I, so that we have $p \precsim^I_{\sqsubseteq} q \Rightarrow pIq$. □

As in the characterisations of behaviour equivalences in Section 3.2, we have just characterised the behaviour preorders that satisfy the hypothesis of Theorem 36 in terms of themselves, but this is just the same idea used in the original works by Milner and Sangiorgi [4,43] on classical bisimulations up-to. Therefore, our (bi)simulations up-to can be used exactly in the same way: by using a known part of the relation $\sqsubseteq$ we could generate, via $\precsim^I_{\sqsubseteq}$, other pairs in the relation.

In particular, we can also characterise a preorder in terms of a simulation up-to its kernel equivalence, which is indeed a way of avoiding the circularity in the characterisation in Theorem 36. It is interesting to observe that the situation is in some way dual to that we had for behaviour equivalences, which could be characterised by using the behaviour preorders that generated them, which obviously are coarser relations. Now we have the opposite situation, and the (finer) behaviour equivalences will be enough to generate the corresponding behaviour preorders, by means of I-simulations up-to them.

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 38. *For every behaviour preorder $\sqsubseteq$ that satisfies the axiom (RS) and is initials preserving, we have that $p \sqsubseteq q \Rightarrow q \equiv q + p$.*

Proof. If $p \sqsubseteq q$ then, since $\sqsubseteq$ is a precongruence with respect to the choice operator, $p + q \sqsubseteq q + q$, and therefore $p + q \sqsubseteq q$.

To prove $q \sqsubseteq p + q$ it is enough to show that $q|_a \sqsubseteq q|_a + p|_a$, since $\sqsubseteq$ is a precongruence wrt the choice operator, and if $p \sqsubseteq q$ then $I(p) \subseteq I(q)$. But for all $a \in I(q)$ we have $q|_a \sqsubseteq q|_a + p|_a$, because $\sqsubseteq$ satisfies the (RS) property. □

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 initials preserving.

It is also interesting to note that the converse of the previous result is not true in general. In order to have it, we need to reinforce our hypothesis by considering only preorders that are finer than the relation I.

Proposition 39. *For every behaviour preorder $\sqsubseteq$ that satisfies the axiom (RS) and $\sqsubseteq\ \subseteq I$, we have that $p \sqsubseteq q \Leftrightarrow q \equiv q + p \ \wedge\ pIq$.*

Proof. The left to right implication is essentially proved in Lemma 38.

For the right to left implication we have that $pIq \Rightarrow p \sqsubseteq_{RS} p + q$ and therefore $p \sqsubseteq p + q$. By hypothesis we also have $q \equiv q + p$, so $p \sqsubseteq p + q \equiv q$, and therefore $p \sqsubseteq q$. □

We will not use the previous result in this section, however it is a clear inspiration for one of our main results in Section 5, namely Corollary 52.

By using Lemma 38 we can now easily prove the following result.

Proposition 40. *For every behaviour preorder $\sqsubseteq$ that satisfies the axiom (RS) and $\sqsubseteq\ \subseteq I$, we have $p \sqsubseteq q \Rightarrow p \sqsubseteq^I_{\precsim_\equiv} q$.*

Proof. Applying Lemma 38, whenever $p \xrightarrow{a} p'_a$ and given that $p \sqsubseteq q$, we have $q \equiv p + q \xrightarrow{a} p'_a$, and then the condition imposed by simulations up-to is satisfied. □

The following corollaries summarise the previous results.

Corollary 41. *For every behaviour preorder $\sqsubseteq$ that satisfies the axiom (RS) and $\sqsubseteq\ \subseteq I$, we have that the relations $\sqsubseteq$, $\sqsubseteq^I_{\precsim_\sqsubseteq}$ and $\sqsubseteq^I_{\precsim_\equiv}$ are the same.*

Proof. From Theorem 36, Proposition 40 and the fact that $\sqsubseteq^I_{\precsim_\equiv}\ \subseteq\ \sqsubseteq^I_{\precsim_\sqsubseteq}$. □

Corollary 42. *For every behaviour preorder $\sqsubseteq$ that satisfies the axiom (RS) and $\sqsubseteq\ \subseteq I$, the following correspondences hold:*

$$\begin{array}{ccc} p \equiv q & \Leftrightarrow & p \sqsubseteq q \wedge p \sqsupseteq q \\ \Updownarrow & & \Updownarrow \\ p \precsim_\sqsubseteq q & \Leftrightarrow & p \sqsubseteq^I_{\precsim_\sqsubseteq} q \wedge p \sqsupseteq^I_{\precsim_\sqsubseteq} q \\ \Updownarrow & & \Updownarrow \\ p \precsim_\equiv q & \Leftrightarrow & p \sqsubseteq^I_{\precsim_\equiv} q \wedge p \sqsupseteq^I_{\precsim_\equiv} q \end{array}$$

Proof. The equivalences in the right column are direct consequences of Corollary 41. The rest of the equivalences in the diagram follow from the next inclusions: $\equiv\ \subseteq\ \precsim_\sqsubseteq$, $\equiv\ \subseteq\ \precsim_\equiv$, $\precsim_\sqsubseteq\ \subseteq (\sqsubseteq^I_{\precsim_\sqsubseteq} \cap \sqsupseteq^I_{\precsim_\sqsubseteq})$ and $\precsim_\equiv\ \subseteq (\sqsubseteq^I_{\precsim_\equiv} \cap \sqsupseteq^I_{\precsim_\equiv})$. The first two inclusions are obvious. For the third one we have that $p \precsim_\sqsubseteq q$ means that there is a bisimulation up-to $\sqsubseteq$ containing the pair (p,q). Any such bisimulation is contained in the binary relation I, and therefore it is also an I-simulation up-to $\sqsubseteq$, thus proving that $\precsim_\sqsubseteq\ \subseteq\ \sqsubseteq^I_{\precsim_\sqsubseteq}$. The rest of the inclusions are analogous. □

Corollaries 41 and 42 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 42 provides a characterisation both in terms of bisimulation-like relations and in terms of mutual simulation-like relations, for any of the preorders between failure and ready simulation and the corresponding equivalences.

4.3. Back to bisimulations up-to and CI-simulations up-to

Our results about simulations up-to (Section 4.1) and I-simulations up-to (Section 4.2) are quite similar to those for bisimulations up-to in Section 3, but they were obtained in an independent way. In particular, once we got the characterisations of behaviour preorders by means of simulations up-to we realized that it was very easy to relate them with the characterisations of the corresponding behaviour equivalences by means of bisimulations up-to.

However, to obtain these characterisations we had to consider two separate cases: first we studied the semantics that are coarser than simulation but finer than the trace semantics, which includes only these two semantics: simulation and trace semantics. Then we considered the semantics coarser than ready simulation whose defining behaviour preorders are finer than the relation I. This includes all the semantics in the ltbt spectrum between failures and ready simulation.

But there are still two semantics in Table 1 that are not included in the cases considered above: completed trace and completed simulation semantics. In fact, there are three simulation semantics in Table 1: plain simulations, completed simulations and ready simulations, and each one of them defines a slice in the ltbt spectrum. This justifies the use of dotted lines in that table to separate the three slices. We have already studied two of these slices. In this section we will see that we can adequate the techniques used before, to cover also the third slice.

Once all the semantics in the spectrum can be characterised by means of simulations up-to, the main result in Section 3, namely Theorem 10, can be now obtained as an immediate corollary.

To study the semantics coarser than ready simulation we introduced the condition I. In a similar way, we define the relation CI that relates those pairs of processes (p,q) such that $I(p) = \emptyset \Leftrightarrow I(q) = \emptyset$. This relation allows us to define CI-simulations up-to a preorder $\sqsubseteq$. The CI relation will be used together with axiom (CS) $ax \sqsubseteq ax + y$ that characterises the complete simulation preorder.

Definition 43. For $\sqsubseteq$ a behaviour preorder, we say that a binary relation S over processes is a *CI-simulation up-to* $\sqsubseteq$, if $S \subseteq CI$ (that is, $pSq \Rightarrow p\ CI\ q$), 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' and q'_a such that $q \sqsupseteq q' \xrightarrow{a} q'_a$ and $p'_a S q'_a$.
- $p\ CI\ q$.

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

Let us now state without proof, since they are quite similar to those for I-simulations, the main results we can prove for CI-simulations.

Theorem 44. *For every behaviour preorder $\sqsubseteq$ that satisfies the axiom (CS) and $\sqsubseteq\, \subseteq CI$, we have $p \sqsubset\!\!\!\!\sim^{CI}_{\sqsubseteq} q$ if and only if $p \sqsubseteq q$.*

Corollary 45. *For every behaviour preorder $\sqsubseteq$ that satisfies the axiom (CS) and $\sqsubseteq\, \subseteq CI$, the following correspondence holds:*

$$\begin{array}{ccc}
p \equiv q & \Leftrightarrow & p \sqsubseteq q \wedge p \sqsupseteq q \\
\Updownarrow & & \Updownarrow \\
p \eqsim_{\sqsubseteq} q & \Leftrightarrow & p \sqsubset\!\!\!\!\sim^{CI}_{\sqsubseteq} q \wedge p \sqsupset\!\!\!\!\sim^{CI}_{\sqsubseteq} q \\
\Updownarrow & & \Updownarrow \\
p \eqsim_{\equiv} q & \Leftrightarrow & p \sqsubset\!\!\!\!\sim^{CI}_{\equiv} q \wedge p \sqsupset\!\!\!\!\sim^{CI}_{\equiv} q
\end{array}$$

Then we restate our Theorem 10 as a simple corollary.

Corollary 46. *For all the behaviour preorders $\sqsubseteq$ defining the semantics in the linear time-branching time spectrum coarser than the ready simulation (see Table 1), we have that $p \eqsim q$ if and only if $p \equiv q$.*

Proof. We obtain the proof of this result as an immediate corollary of our results on the characterisation of behaviour preorders by means of (adequate) simulations up-to, simply by combining the results in our Corollaries 32, 42 and 45. ☐

Note that although the corollaries used in the proof of Corollary 46 were proved once we already had our characterisation of behaviour equivalences by means of bisimulations up-to, namely Theorem 10, we did not use this result at all in their proofs: we only used our results on the characterisation of the behaviour preorders by means of simulations up-to.

As a matter of fact, we could obtain this indirect proof of our main result of Section 3 not only for the semantics in the ltbt spectrum, but for any semantics that satisfies the semantic conditions in Corollaries 32, 42 and 45. It is also interesting to observe that we needed to restrict ourselves to action factorised preorders when investigating bisimulations up-to, but we did not need this hypothesis when working with simulations up-to. Nevertheless, requiring a preorder to be action factorised is not a big constraint at all, since this property is satisfied by almost every reasonable semantic preorder in which one could be interested.

5. Simulations up-to an equivalence and canonical preorders

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 from.

As we have discussed at the end of the Section 4, the results for simulations up-to come in slices determined by the base simulation we consider in each case. In order to avoid repetitions, in this section we just state and prove the most difficult case, that corresponding to the slice determined by the ready simulation semantics.

Therefore, we will study behaviour equivalences (see Definition 16) coarser than the ready simulation equivalence. They are those that satisfy the following axiom:

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

We recalled in Table 2 a complete axiomatization for the semantic equivalences in the linear time-branching time spectrum, as presented in [6].

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

Lemma 47. *For every behaviour equivalence $\equiv$ satisfying $(RS_{\equiv})$ and $\equiv \subseteq I$, we have that $p \sqsubseteq^{I}_{\sim\equiv} q \Rightarrow q \equiv q + p$.*

Proof. We proceed by induction on the depth of process p. If $p = \mathbf{0}$ then $I(p) = \emptyset = I(q)$ and thus $q = \mathbf{0}$.

If $p = \sum_a \sum_i ap^i_a$, whenever $p \xrightarrow{a} p^i_a$ then $q \equiv q'^i_a \xrightarrow{a} q^i_a$ and $p^i_a \sqsubseteq^{I}_{\sim} q^i_a$. By applying the induction hypothesis, $q^i_a \equiv q^i_a + p^i_a$ and then $aq^i_a \equiv a(q^i_a + p^i_a)$, since $\equiv$ is a congruence wrt the prefix operator.

On the other hand, given that $I(q^i_a) = I(p^i_a)$ we can use $(RS_{\equiv})$ to obtain $aq^i_a \equiv a(q^i_a + p^i_a) + ap^i_a$; therefore $aq^i_a \equiv aq^i_a + ap^i_a$, and, adding all the subterms, $\sum_a \sum_i aq^i_a \equiv \sum_a \sum_i aq^i_a + \sum_a \sum_i ap^i_a$, that is $\sum_a \sum_i aq^i_a \equiv \sum_a \sum_i aq^i_a + p$.

Certainly, the process $\sum_a \sum_i aq^i_a$ may be not completely equal to q, but we can use similar arguments to those in the proof of Theorem 36 to add the missing subterms. We have that $q'^i_a = aq^i_a + r^i_a$ and, since $\equiv$ is a congruence wrt choice, $\sum_a \sum_i (aq^i_a + r^i_a) \equiv \sum_a \sum_i (aq^i_a + r^i_a) + p$ and therefore $\sum_a \sum_i q'^i_a \equiv \sum_a \sum_i q'^i_a + p$. Hence, as for every index i we have $q'^i_a \equiv q$, we conclude that $q \equiv q + p$. □

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

Theorem 48. *For every behaviour equivalence $\equiv$ satisfying $(RS_{\equiv})$ and $\equiv \subseteq I$, we have $p \equiv q \Leftrightarrow p \sqsubseteq^{I}_{\sim\equiv} q \wedge p \sqsupseteq^{I}_{\sim\equiv} q$.*

Proof. The left to right implication is obvious. To prove the converse we use Lemma 47 twice, getting $p \sqsubseteq^{I}_{\sim\equiv} q \Rightarrow q \equiv q + p$ and $q \sqsubseteq^{I}_{\sim\equiv} p \Rightarrow p \equiv q + p$, thus concluding $p \equiv q$. □

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

Corollary 49. *For every behaviour equivalence $\equiv$ satisfying $(RS_{\equiv})$ and $\equiv \subseteq I$, we have $p \equiv q \Leftrightarrow p \sqsubseteq^{I}_{\sim\equiv} q \wedge p \sqsupseteq^{I}_{\sim\equiv} q \Leftrightarrow p \eqsim_{\equiv} q$.*

Proof. By using Theorem 48 and the set inclusions $\equiv \subseteq\ \eqsim_{\equiv}\ \subseteq\ (\sqsubseteq^{I}_{\sim\equiv} \cap \sqsupseteq^{I}_{\sim\equiv})$. □

The characterisation in Theorem 48 tells us that any behaviour equivalence can be defined by means of simulations up-to. Besides, and this is even more important, in this way we define a particular preorder whose kernel is the original equivalence. This preorder satisfies some interesting properties.

Proposition 50. *For every behaviour equivalence $\equiv$ that satisfies $(RS_{\equiv})$ and $\equiv \subseteq I$, we have that*

- $\sqsubseteq^{I}_{\sim\equiv}$ *is a behaviour preorder that satisfies* (RS),
- $\sqsubseteq^{I}_{\sim\equiv} \subseteq I$,
- *the kernel of* $\sqsubseteq^{I}_{\sim\equiv}$ *is* $\equiv$.

Proof. $\sqsubseteq^{I}_{\sim\equiv}$ is a precongruence with respect to the choice operator because $\equiv$ is so. It is quite easy to check the other two properties. □

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

Theorem 51. *For every behaviour equivalence* $\equiv$ *satisfying* $(RS_{\equiv})$ *and* $\equiv\ \subseteq I$, *the preorder* $\underset{\sim}{\sqsubset}{}^{I}_{\equiv}$ *is the only behaviour preorder that satisfies* (RS) *and is contained in I whose kernel is* $\equiv$.

Proof. If $\sqsubseteq$ is a preorder satisfying the conditions above then we know from Corollary 41 that $\sqsubseteq$, $\underset{\sim}{\sqsubset}{}^{I}_{\sqsubseteq}$ and $\underset{\sim}{\sqsubset}{}^{I}_{\equiv}$ are equal. □

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

Corollary 52. *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 characterisation of the canonical preorder generated by* $\equiv$.

Proof. In order to apply Theorem 51, we need to check that $\sqsubseteq$ satisfies (RS), is contained in I and its kernel is $\equiv$. We have the second condition by definition. To prove the first one it is enough to see that for all processes p and q we have $ap \sqsubseteq ap + aq$, that is, $ap + aq \equiv ap + aq + ap$, what is true because $\equiv$ is a behaviour equivalence. Finally, $p \sqsubseteq q \Rightarrow q \equiv q + p$ and $q \sqsubseteq p \Rightarrow p \equiv p + q$ and therefore $p \equiv q$. Besides, $p \equiv q \Rightarrow p + q \equiv q \Rightarrow p \sqsubseteq q$. □

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 53. *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* $\equiv$.

Proof. All the preorders for these semantics (that appear in Table 1) satisfy the conditions in Theorem 51, that is, they satisfy (RS) and $\sqsubseteq\ \subseteq\ I$, and of course their kernels are the corresponding equivalences $\equiv$. □

Quite a number of results follow from the previous propositions and point to a rich underlying algebraic theory. Just to give a flavour, we present the following ones:

Corollary 54. *For every behaviour equivalence* $\equiv$ *that satisfies the property* $(RS_{\equiv})$ *and* $\equiv\ \subseteq I$, *we have* $\equiv\ =\ \eqsim_{\underset{\sim}{\sqsubset}{}^{I}_{\equiv}}$ *and* $\underset{\sim}{\sqsubset}{}^{I}_{\equiv} = \underset{\sim}{\sqsubset}{}^{I}_{\underset{\sim}{\sqsubset}{}^{I}_{\equiv}}$.

Proof. The proof of both equalities follows immediately from Proposition 50 and Corollary 42. □

To conclude this section we would like to comment the results in our Corollaries 52 and 53. 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 51. 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 52 we have that $p \sqsubseteq q \Leftrightarrow p \sqsubseteq' q \wedge I(p) = I(q)$. As a matter of fact, the only difference between $\sqsubseteq$ and $\sqsubseteq'$ lies in the set of initial actions of the processes, but this is crucial to get the characterisation of the corresponding preorders in Corollary 53.

Instead, in Theorem 28 and Corollary 31, we needed no condition on the simulations up-to that we used to characterise the behaviour preorders that satisfy the axiom (S). As a consequence, in this case our canonical preorder and the lattice canonical one are the same. In particular, for trace preorder, $\sqsubseteq_T$, and the simulation preorder, $\sqsubseteq_S$, we have both $p \sqsubseteq_T q \Leftrightarrow q \equiv_T q + p$ and $p \sqsubseteq_S q \Leftrightarrow q \equiv_S q + p$.

6. Bisimulations and simulations up-to for infinite processes

The results in the previous sections were proved for BCCSP processes. In this section we extend these results, considering processes to be (possibly) infinite finitary trees and using the Approximation Induction Principle [39]. We will use the same notation as for finite trees (prefix, choice, multiple choice, etc.) extended in the natural way.

To reduce infinite trees to (collections of) finite trees, we define an adequate notion of approximation, that we call *level continuity*, and prove how level continuous behaviour preorders lead to level continuous (bi)simulations up-to. Once this result is stated, all our characterisation results can also be proved for level continuous behaviour preorders, using standard continuity reasonings. The definition of level continuity is rather natural, so that in particular every behaviour preorder for the semantics in Fig. 1 is indeed level continuous.

Definition 55. A behaviour preorder is *level continuous if $p \sqsubseteq q$ if and only if $p\downarrow_n \sqsubseteq q\downarrow_n$ for all n, where $p\downarrow_n$ is the result of pruning process p below level n, that is:*

- $\mathbf{0}\downarrow_n = \mathbf{0}$,
- $p\downarrow_0 = \mathbf{0}$,
- $(\sum ap_a)\downarrow_{n+1} = \sum a(p_a\downarrow_n)$.

Note that $p\downarrow_n$ is always a finite process having depth at most n. Next we prove a technical lemma stating that the number of equivalence classes, with respect to the bisimulation equivalence, of processes having bounded depth is finite. We use $|A|$ to denote the cardinality of a set A and $[p]_{=_B}$ to denote the equivalence class of p with respect to bisimulation equivalence, $=_B$.

Lemma 56. *If the alphabet of actions Act is finite, for every natural number n we have*

$$|\{[p]_{=_B} \mid depth(p) \leq n\}| < \infty.$$

Proof. By induction on n. For $n = 0$, $p = \mathbf{0}$. For $n > 0$, if $p = \sum_i ap_a^i$ and $q = \sum_j aq_a^j$, then $p =_B q$ iff

- for all a and i there exists j such that $p_a^i =_B q_a^j$,
- for all a and j there exists i such that $p_a^i =_B q_a^j$.

Thus, $p =_B q$ iff for any action a, $\{[p_a^i]_{=_B}\} = \{[q_a^j]_{=_B}\}$, therefore, the elements of $\{[p]_{=_B} \mid \text{depth}(p) \leq n+1\}$ are in one to one correspondence with functions in $Act \longrightarrow \mathcal{P}(\{[p]_{=_B} \mid \text{depth}(p) \leq n\})$. And thus we conclude the proof by applying the induction hypothesis. □

Then, for every behaviour preorder stronger than the trace preorder we have the following finiteness result:

Lemma 57. *If a behaviour preorder $\sqsubseteq$ is finer than the trace preorder $(\sqsubseteq\, \subseteq\, \sqsubseteq_T)$, for any finite process q the set of bisimilarity classes $\{[p]_{=_B} \mid p \sqsubseteq q\}$ is finite.*

Proof. Since $\sqsubseteq\, \subseteq\, \sqsubseteq_T$ we have that $p \sqsubseteq q \Rightarrow \text{depth}(p) \leq \text{depth}(q)$ and that any action in the alphabet of process p is also in that of process q. We are then in the hypothesis of Lemma 56. □

Proposition 58. *For every level continuous behaviour preorder $\sqsubseteq$, the equivalence defined by the corresponding bisimulation up-to $\sqsubseteq$, $\eqsim$, is level continuous too.*

Proof. According to the definition, we have to prove that $p \eqsim q$ iff for all n, $p\downarrow_n \eqsim q\downarrow_n$. First we prove the left to right implication.

Let S be a bisimulation up-to $\sqsubseteq$, we will see that the relation $S_f = \{(p\downarrow_n, q\downarrow_n) \mid pSq\}$ is also a bisimulation up-to $\sqsubseteq$. Indeed, whenever $p\downarrow_n \xrightarrow{a} p'_a\downarrow_{n-1}$, we have also $q\downarrow_n \sqsupseteq q'\downarrow_n \xrightarrow{a} q'_a\downarrow_{n-1}$, because of level continuity of $\sqsubseteq$, and since $p'_a S q'_a$, we finally obtain $p'_a\downarrow_{n-1} S_f\, q'_a\downarrow_{n-1}$.

Now we prove the right to left implication. Let us define the relation $R = \{(p,q) \mid \text{ for all } n\;\; p\downarrow_n \eqsim q\downarrow_n\}$. We will see that it is a bisimulation up-to $\sqsubseteq$. We have that $p \xrightarrow{a} p'_a$ iff $p\downarrow_n \xrightarrow{a} p'_a\downarrow_{n-1}$, and then there exists $q\downarrow_n \sqsubseteq q'_n \xrightarrow{a} q'_{n,a}$ with $p'_a\downarrow_{n-1} \eqsim q'_{n,a}$.

It is easy to check that for all $m > n$, $p'_a\downarrow_{n-1} \eqsim q'_{m,a}\downarrow_{n-1}$. Then, we define $Q_n^m = \{q'_m\downarrow_n \mid q\downarrow_m \sqsupseteq q'_m \xrightarrow{a} q'_{m,a} \text{ and } p'_a\downarrow_{n-1} \eqsim q'_{m,a}\}$ and because $\sqsubseteq$ is weaker than bisimulation equivalence, we have that Q_n^m is closed under $=_B$. We can now check that for all $m' > m$, $Q_n^{m'} \subseteq Q_n^m$ since if $q'_{m'}\downarrow_n \in Q_n^{m'}$ then $(q'_{m'}\downarrow_m)\downarrow_n = q'_{m'}\downarrow_n$ and $(q'_{m'}\downarrow_m)\downarrow_n \in Q_n^m$. Now, applying Lemma 57, $Q_n^{m'}/_{=_B} \subseteq Q_n^m/_{=_B}$ and therefore $0 < |Q_n^m/_{=_B}| < \infty$.

We conclude that there exists a natural number m such that for any other natural number m', $Q_n^{m'} = Q_n^m$. Defining $Q_n = Q_n^m$ for such an m, we also have $Q_n = Q_{n'}\downarrow_n$ for all $n' \geq n$. Then it is clear that there exists some process q' such that for all n $q'\downarrow_n \in Q_n$ and therefore for all n $q\downarrow_n \sqsubseteq q'\downarrow_n$ and $q'\downarrow_n \xrightarrow{a} q'_{n,a}$ with $p'_a\downarrow_{n-1} \eqsim q'_{n,a}$, so that we have both $q \sqsupseteq q'$ and $q' \xrightarrow{a} q'_a$ with $p'_a\downarrow_{n-1} \eqsim q'_a\downarrow_{n-1}$, thus proving that the pair $(p'_a,q'_a) \in R$, so that R is indeed a bisimulation up-to $\sqsubseteq$. □

Exactly in the same way we can prove that whenever $\sqsubseteq$ is level continuous then $\sqsubset^I_{\sim_\sqsubseteq}$ is also level continuous, and the same is true for both $\sqsubset_{\sim_\sqsubseteq}$ and $\sqsubset^{CI}_{\sim_\sqsubseteq}$.

Proposition 59. *For every behaviour preorder $\sqsubseteq$, and the corresponding I-simulation up-to $\sqsubseteq$, $\sqsubset^I_{\sim_\sqsubseteq}$, if $\sqsubseteq$ is level continuous then $\sqsubset^I_{\sim_\sqsubseteq}$ is level continuous too.*

Proof. According to the definition of level continuity, we have to prove that $p \precsim^I_{\sqsubseteq} q$ iff for all n, $p\downarrow_n \precsim^I_{\sqsubseteq} q\downarrow_n$. First we prove the left to right implication.

Let S be an I-simulation up-to $\sqsubseteq$, then $S_f = \{(p\downarrow_n, q\downarrow_n) \mid pSq\}$ is also an I-simulation up-to $\sqsubseteq$. We have that $S_f \subseteq I$ because $S \subseteq I$, for $n = 0$ we have $p\downarrow_0 = \mathbf{0} = q\downarrow_0$, and for $n > 0$ we have $I(p) = I(p\downarrow_n)$. Moreover, whenever $p\downarrow_n \xrightarrow{a} p'_a\downarrow_{n-1}$, we have $q\downarrow_n \sqsupseteq q'\downarrow_n \xrightarrow{a} q'_a\downarrow_{n-1}$, because of level continuity of $\sqsubseteq$, and, since $p'_a S q'_a$, then $p'_a\downarrow_{n-1} S_f\, q'_a\downarrow_{n-1}$.

Now we prove the right to left implication. Let us define the relation $R = \{(p,q) \mid \text{for all } n \;\; p\downarrow_n \precsim^I_{\sqsubseteq} q\downarrow_n\}$. Obviously we have $R \subseteq I$, since $I(p) = I(p\downarrow_1)$. We will see that it is an I-simulation up-to $\sqsubseteq$. We have that $p \xrightarrow{a} p'_a$ iff $p\downarrow_n \xrightarrow{a} p'_a\downarrow_{n-1}$, and then there exists $q\downarrow_n \sqsupseteq q'_n \xrightarrow{a} q'_{n,a}$ with $p'_a\downarrow_{n-1} \precsim^I_{\sqsubseteq} q'_{n,a}$.

It is easy to check that for all $m > n$, $p'_a\downarrow_{n-1} \precsim^I_{\sqsubseteq} q'_{m,a}\downarrow_{n-1}$. Then, we define $Q^m_n = \{q'_m\downarrow_n \mid q\downarrow_m \sqsupseteq q'_m \xrightarrow{a} q'_{m,a} \text{ and } p'_a\downarrow_{n-1} \precsim^I_{\sqsubseteq} q'_{m,a}\}$ and because $\sqsubseteq$ is weaker than bisimulation equivalence, we have that Q^m_n is closed under $=_B$. We can now check that for all $m' > m$, $Q^{m'}_n \subseteq Q^m_n$ since if $q'_{m'}\downarrow_n \in Q^{m'}_n$ then $(q'_{m'}\downarrow_m)\downarrow_n = q'_{m'}\downarrow_n$ and $(q'_{m'}\downarrow_m)\downarrow_n \in Q^m_n$. Now, applying Lemma 57, $Q^{m'}_n/_{=_B} \subseteq Q^m_n/_{=_B}$ and therefore $0 < |Q^m_n/_{=_B}| < \infty$.

We conclude that there exists a natural number m such that for any other natural number m', $Q^{m'}_n = Q^m_n$. Defining $Q_n = Q^m_n$ for such an m, we also have $Q_n = Q_{n'}\downarrow_n$ for all $n' \geq n$. Then it is clear that there exists some process q' such that for all n $q'\downarrow_n \in Q_n$ and therefore for all n $q\downarrow_n \sqsubseteq q'\downarrow_n$ and $q'\downarrow_n \xrightarrow{a} q'_{n,a}$ with $p'_a\downarrow_{n-1} \precsim^I_{\sqsubseteq} q'_{n,a}$, so that we have both $q \sqsupseteq q'$ and $q' \xrightarrow{a} q'_a$ with $p'_a\downarrow_{n-1} \precsim^I_{\sqsubseteq} q'_a\downarrow_{n-1}$, thus proving that the pair $(p'_a, q'_a) \in R$, so that R is indeed an I-simulation up-to $\sqsubseteq$. □

Thus for any level continuous preorder verifying the hypothesis of any of the theorems in this paper the results of these theorems are also valid for infinite processes. In particular, all the preorders for the semantics in Fig. 1 are level continuous and therefore all the results that we have for BCCSP processes are also valid for all the processes defined by a finitary LTS.

Proposition 60. *All the behaviour preorders defining the semantics in Fig. 1 are level continuous.*

Proof. We give the proof for two (extreme) representative examples:

The trace preorder $\sqsubseteq_T$ is level continuous: $p \sqsubseteq_T q$ iff whenever $p \xrightarrow{\sigma}$ then $q \xrightarrow{\sigma}$ iff for all n, $p\downarrow_n \xrightarrow{\sigma}$ then $q\downarrow_n \xrightarrow{\sigma}$, iff for all n, $p\downarrow_n \sqsubseteq_T q\downarrow_n$.

The ready simulation preorder $\sqsubseteq_{RS}$ is level continuous: We have to check that $p \sqsubseteq_R q$ iff for all n, $p\downarrow_n \sqsubseteq_R q\downarrow_n$. For the left to right implication we define the relation $R = \{(p\downarrow_n, q\downarrow_n) \mid p \sqsubseteq_R q\}$ that is a ready simulation since $I(p) = I(q)$ implies that $I(p\downarrow_n) = I(q\downarrow_n)$ and if $p \xrightarrow{a} p'$ then $p\downarrow_n \xrightarrow{a} p'\downarrow_{n-1}$.

For the other implication we define $R = \{(p,q) \mid \text{ for all } n,\, p\downarrow_n \sqsubseteq_R q\downarrow_n\}$, and show that it is a ready simulation. First, $I(p) = I(p\downarrow_1)$, so that, whenever pRq we have $I(p) = I(q)$. Then, whenever $p \xrightarrow{a} p'$, we know that $p\downarrow_n \xrightarrow{a} p'\downarrow_{n-1}$, for all $n \geq 1$, and therefore there exists q''_n such that $q\downarrow_n \xrightarrow{a} q''_n$ with $p'\downarrow_{n-1} \sqsubseteq_R q''_n$. Obviously, there exists some succesor of q that extends q''_n, that is there exists $q'_{i(n)}$ such that $q \xrightarrow{a} q'_{i(n)}$ and $q'_{i(n)}\downarrow_{n-1} = q''_n$.

Since q is finitely branching there exists some q' such that $q' = q'_{i(n)}$ for infinitely many n and, therefore, we can take q' as $q'_{i(n)}$ for any n. Then $p'\downarrow_n \sqsubseteq_R q'\downarrow_n$, for all n and therefore $p'Rq'$, proving that R is a ready simulation containing the pair (p,q). □

Let us just give an example of how the generalised characterisation results are stated and how they are easily proved as immediate corollaries of the corresponding results for (finite) BCCSP processes, using the fact that the involved behaviour preorders are level continuous. We consider, for instance, our main result in Section 4, namely Theorem 36.

Theorem 61. *For every behaviour preorder $\sqsubseteq$ that satisfies the axiom (RS) and $\sqsubseteq\, \subseteq I$, and for any two processes p and q defined by a rooted finitary LTS, we have $p \precsim^I_{\sqsubseteq} q$ if and only if $p \sqsubseteq q$.*

Proof. Since $\sqsubseteq$ is level continuous we have $p \sqsubseteq q$ iff for all n $p\downarrow_n \sqsubseteq q\downarrow_n$. But for any n both $p\downarrow_n$ and $q\downarrow_n$ are finite processes so that we can apply Theorem 36 to obtain $p\downarrow_n \sqsubseteq q\downarrow_n$ iff $p\downarrow_n \precsim^I_{\sqsubseteq} q\downarrow_n$, and since $\precsim^I_{\sqsubseteq}$ is also level continuous we finally obtain $p \precsim^I_{\sqsubseteq} q$. □

7. Applications of the coinductive characterisations

As we already mentioned, a first application of our coinductive characterisations of the behaviour preorders and equivalences would be their direct use to infer that some pairs of processes are related by the corresponding relations. Thisis done by constructing a (bi)simulation up-to that contains these pairs, based on a part already known of, either the corresponding equivalence, or the preorder that generates it. Our first application below uses these ideas to prove a simple property of trace semantics.

But at the moment we are more interested in those applications that are related with our main motivation when undertaking this work, that was to study the general properties of the different semantics for concurrent processes and the relationships between them.

If we consider the way classical bisimulations are defined we observe that there is first a (symmetric) local condition that relates the sets of initial actions of any two related processes, which is mainly our condition I, and then the coinductive hypothesis imposing that the derived processes p' and q' must be related. Our (bi)simulations up-to proceed in a similar way, and although the reduction by means of the reversed order $\sqsupseteq$ could change the initial set of the reduced process, the fact that we only consider (bi)simulations up-to behaviour preorders guarantees that no new initial actions can appear when reducing a process. This property, even if apparently trivial, becomes quite powerful when preserved by the coinductive definition of (bi)simulations.

Our second application is much more interesting than the first and shows how a general property of semantics can be proved once for all, without needing to repeat the reasonings for each particular semantics. We are sure that some other interesting applications will be soon discovered, and in fact you can find in the conclusions of the paper the announcement of a forthcoming paper where some quite recent results presented in [27] will be proved in a more general and simple way, using our coinductive characterisations of the semantics.

7.1. Coinductive proofs of properties of the semantics

Next we consider the same example used by Klin to illustrate his results in [34]. We prove that any process has the same traces as its deterministic form. This result can be easily proved, by induction, for finite processes. But we need to be careful when copying with infinite processes. Like Klin, we will use coinductive reasoning to do it, but certainly our proof is simpler than that in [34], although it is also true that Klin develops his approach in a framework quite broader than ours.

Definition 62. For any process $p = \sum_a \sum_i ap_{a,i}$ the *deterministic form* of p is defined as $\mathrm{Det}(p) = \sum_a a\ \mathrm{Det}(\sum_i p_{a,i})$.

We wish to prove that p and $\mathrm{Det}(p)$ are trace equivalent. We will do it by using our bisimulation up-to technique. First we prove the following lemma.

Lemma 63. *For any processes p and q we have that* $\mathrm{Det}(p) \sqsubseteq_T \mathrm{Det}(p+q)$.

Proof. We prove something stronger, in fact $\mathrm{Det}(p)$ is simulated by $\mathrm{Det}(p+q)$. Since $\mathrm{Det}(p) = \sum_a a\ \mathrm{Det}(\sum_i p_{a,i})$, whenever $\mathrm{Det}(p) \xrightarrow{a} \mathrm{Det}(\sum_i p_{a,i})$ we also have $\mathrm{Det}(p+q) \xrightarrow{a} \mathrm{Det}(\sum_i p_{a,i} + \sum_j q_{a,j})$. □

Proposition 64. *For any process p*, $p \eqsim_{\sqsubseteq_T} \mathrm{Det}(p)$.

Proof. We will prove that the relation $R = \{(p, \mathrm{Det}(p)) \mid p \text{ is a process }\}$ is a bisimulation up-to $\sqsubseteq_T$. Whenever $p \xrightarrow{a} p_{a,i}$, then, by using Lemma 63, $\mathrm{Det}(p) = \sum_a a\ \mathrm{Det}(\sum_i p_{a,i}) \sqsupseteq_T a\ \mathrm{Det}(p_{a,i}) \xrightarrow{a} \mathrm{Det}(p_{a,i})$. Besides, if $\mathrm{Det}(p) \xrightarrow{a} \mathrm{Det}(\sum p_{a,i})$, applying the axioms that characterise the trace preorder ($x \sqsubseteq_T x+y$, $a(x+y) =_T ax+ay$) we have that $p \sqsupseteq_T \sum_i ap_{a,i} \sqsupseteq_T a \sum_i p_i$ and, therefore $p \sqsupseteq_T a \sum p_{a,i} \xrightarrow{a} \sum p_{a,i}$. □

If we examine in detail the proof above we see how we have only used the new information about the trace semantics provided by Lemma 63 when constructing the bisimulation up-to that shows that p and $\mathrm{Det}(p)$ are trace equivalent. It is also interesting to observe that although we are proving the trace equivalence between these two processes, we need the full power of the preorder $\sqsubseteq_T$, since in general $\mathrm{Det}(p)$ and $\mathrm{Det}(p+q)$ do not have the same traces. Besides, we can see how we have used the general ideas discussed above, since in order to prove the trace equivalence between these two processes, what we have mainly done is to prove that their sets of initials actions are the same, and then we apply coinduction to get the rest.

7.2. Some results on axiomatic characterisations

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

Corollary 41 states that a behaviour preorder (under some adequate conditions) can be characterised by the I-simulations up-to the kernel of that preorder, $\sqsubseteq = \sqsubseteq^I_{\sim_\equiv}$. This result suggested to us the possibility of finding an axiomatization for this

Table 2
Axiomatization for the equivalences in the linear time-branching time spectrum.

	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
$ax+ay=a(x+y)$											+

preorder from that of the equivalence. We have found that, if A_E is a set of axioms that characterises a given equivalence $\equiv$, we can easily define an axiomatization for the canonical preorder $\sqsubseteq^I_{\equiv}$. The new set of axioms can be defined by just adding the (RS) axiom to the axioms of the equivalence: $A_P = A_E \cup \{ax \sqsubseteq ax + ay\}$. We formalise this in the following theorem.

Theorem 65. *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 $\sqsubseteq^I_{\equiv}$.*

Proof. We write $A_P \vdash p \sqsubseteq q$ when the inequality $p \sqsubseteq q$ is provable from A_P. We prove that $A_P \vdash p \sqsubseteq q$ iff $p \sqsubseteq^I_{\equiv} q$.

Soundness is straightforward: we have $\equiv\, \subseteq\, \sqsubseteq^I_{\equiv}$ and $\sqsubseteq^I_{\equiv}$ satisfies (RS).

The proof of completeness is similar to that of Theorem 36. We proceed by induction on the depth of process p.

The base case is trivial: if $p = \mathbf{0}$ then $I(p) = \emptyset = I(q)$ and thus $q = \mathbf{0}$.

The inductive case: If $p = \sum_a \sum_i ap^i_a$, whenever $p \xrightarrow{a} p^i_a$ then $q \equiv q'^i_a \xrightarrow{a} q^i_a$ and $p^i_a \sqsubseteq^I_{\sim} q^i_a$. By applying the induction hypothesis, $A_P \vdash p^i_a \sqsubseteq q^i_a$ and also $A_P \vdash ap^i_a \sqsubseteq aq^i_a$ and therefore $A_P \vdash \sum_a \sum_i ap^i_a \sqsubseteq \sum_a \sum_i aq^i_a$. That is, $A_P \vdash p \sqsubseteq \sum_a \sum_i aq^i_a$. We also have that $q'^i_a = aq^i_a + r^i_a$, and given that $I(q) = I(\sum_a \sum_i q'^i_a) = I(\sum_a \sum_i aq^i_a + r^i_a)$ we can use the (RS) axiom in A_P to add all the missing subterms r^i_a: $A_P \vdash \sum_a \sum_i aq^i_a \sqsubseteq \sum_a \sum_i aq^i_a + r^i_a$, and since $\sum_a \sum_i aq^i_a + r^i_a = \sum_a \sum_i q'^i_a$, we can now conclude that $A_P \vdash p \sqsubseteq \sum_a \sum_i aq^i_a \sqsubseteq \sum_a \sum_i q'^i_a \equiv q$. □

We can directly apply Theorem 65 to those equivalences in the ltbt spectrum that satisfy the right conditions. In Table 2 appears an axiomatization for the (finitely) axiomatizable equivalences in the ltbt spectrum. From these axioms we can get an alternative axiomatization of the preorders in Table 1.

Corollary 66. *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 $\equiv_{\mathcal{O}}$.*

Proof. The equivalences $\equiv_{\mathcal{O}}$ satisfy in the conditions of Theorem 65, and the preorders $\sqsubseteq_{\mathcal{O}}$ are in fact the canonical preorders, as proved in Corollary 53. □

It is interesting to note that to prove the completeness of the axiomatizations in Table 1 elaborated proofs were needed in [6], whereas here we get all these completeness results, once for all, based on the completeness of the axiomatization of the corresponding equivalence.

Theorem 65 can also be applied to extend some other recent results on axiomatizations of semantics.

In [6] the axiomatizations of failure trace equivalence (FT) and ready trace equivalences (RT) are both of them conditional (see Table 2). The existence of a non-conditional axiomatization was an open question. In [46] non-conditional axiomatizations for these equivalences were studied. In particular, it was proved that if the alphabet of actions is finite then there exists a finite equational axiomatization for the process algebra BCCSP modulo ready trace equivalence. Nothing is said about the preorder for this semantics. In fact, the axiom that makes conditional the axiomatization for the ready trace equivalence conditional is also used when defining the axiomatization of the ready trace preorder, and therefore that one is also a conditional axiomatization (see Table 1). Probably the authors were not interested in the ready trace preorder, because, in the past no connections were established between the axiomatizations of the equivalence relations and those of the corresponding preorders.

Fortunately, the study of the theory of simulations up-to reveals that preorders and induced equivalences are closely related, and from now on we can take advantage of the characterisations we provide in this paper. In particular, given that the ready trace equivalence satisfies the hypothesis of Theorem 65, we can apply it to any axiomatization of it, and thus we can state the following extension of the commented result in [46].

Corollary 67. *If the alphabet of actions is finite there exists a finite equational axiomatization for BCCSP modulo the ready trace preorder.*

Proof. If we add the non-conditional axiom (*RS*), $ax \sqsubseteq ax + ay$, to the axiomatization given in [46] for the ready trace equivalence we get a non-conditional axiomatization for the ready trace preorder, as an immediate consequence of our Theorem 65. □

8. Conclusions and future work

In this paper we have studied in detail the notions of bisimulation up-to and simulation up-to a preorder, by means of which we have got coinductive characterisations of semantics, both for equivalences and preorders (for instance, Theorems 10 and 36). In particular, we have characterised all the equivalences and preorders associated to the semantics in the linear time-branching time spectrum.

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 (for instance, Corollary 42). In fact, for large families of semantics, including those in the linear time-branching time spectrum coarser than ready simulation, the results for bisimulations up-to may be obtained as a corollary of the corresponding ones for simulations up-to.

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 simulation up-to it (Theorem 51). 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 is 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 characterisation we have discovered new properties that open the door to new 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 (Theorem 65).

Once we have canonical characterisations of the preorders defining each of the semantics in the spectrum, and since these preorders can be also defined by the testing and the logical characterisations of the corresponding semantics, one could claim that those characterisations are also canonical. However, it would be nice to get a more direct justification of this canonicity, so that we could talk about characteristic tests or logical formulae for each semantics, defining them by means of a predicate on the full universe of tests or formulae. We plan to continue working on this subject.

Besides, we will continue our work relating bisimulations and simulations. In particular we are interested in translating our results to the pure coalgebraic world, comparing them to those presented by Hughes and Jacobs in [32] and Hasuo in [47,48]. In more detail, the categorical notion of simulation studied in [32] uses functorial orders that are introduced in the definition of bisimulation in a way that resembles a lot the way bisimulations up-to were defined. However, since these preorders are used in an asymmetric way, it is possible that preorders that are not equivalences will be characterised in this way. Recently, we have presented in [49] a large collection of proposals for distributed bisimulations that we want to study in a common framework, using this notion of categorical simulation.

Certainly, a natural extension of our work concerns the characterisation of weak semantics that consider the existence of internal transitions. We have already obtained some quite promising results in this direction that show that things in the weak case are quite similar to those for strong semantics. Probably the greatest difficulty to present our results in a systematic way, as we have done in this paper for the strong case, is that the number of possibilities to define these weak semantics is enormous, as one can see in [50]. In particular, an elegant algebraic presentation of a representative large family of them, as that for the strong semantics presented in [6], is still missing.

Another interesting subject is the structure of the classification of semantics for processes. The three slices that we distinguish in Section 4 and that appear as dotted lines in Table 1 suggest to us that every behaviour preorder will be bounded by a finer, univocally defined, simulation-like preorder that has a coinductive definition, and can be axiomatized by means of axioms that are refinements of the axiom (*S*) that defines plain simulations. Any behaviour preorder that is not a simulation-like preorder will have other non-simulation axioms that define what we call its “static” part. We have found no references in the literature to this “decomposition” of semantic preorders into a “simulation” part and a “static” part, and we think that its study would contribute to the clarification of the above cited structure. Based on our results for semantics that are coarser than the ready simulation we have found that this simulation part, that is ready simulation in this case, and their defining axioms, just (*RS*) in this case, play an important role in the study of the properties of these preorders. Trying to generalise our results we have introduced arbitrary *constrained simulations* that are defined just as ready simulations, but changing the condition *I* in Definition 33 by any other adequate constraint, as we made with *CI* (Definition 43) when defining completed simulations. During the publication procedure of this paper we have indeed obtained a satisfactory extension of our results in this direction [51].

Another recent publication is [27], where it is also established a relation between the axiomatizations of the preorders that are weaker than the ready simulation and those of the corresponding equivalences, for the semantics in van Glabbeek's spectrum. In this case, it is explained how to obtain the axiomatization of the induced equivalence starting from that of a preorder. One can find in their paper an extensive discussion relating their results and ours.

But even if the results in [27] are quite interesting, we have seen [52] that using algebraic techniques we are able to obtain a unique general proof of these results, avoiding more than 20 pages of detailed proofs, that where needed in [27] to prove the main technical lemma there. It was the use of the general properties of the semantics, as we have also done in this paper, that allow us to avoid to consider all the semantics in the ltbt-spectrum separately, one by one. Still another more recent discovering has completed in a very nice way our work on the subject, by relating the algebraic and coalgebraic approaches [53].

References

[1] D. de Frutos-Escrig, C. Gregorio-Rodrguez, Bisimulations up-to for the linear time-branching time spectrum, in: CONCUR 2005 – Concurrency Theory, 16th International Conference, Vol. 3653 of Lecture Notes in Computer Science, Springer, 2005, pp. 278–292.
[2] D. de Frutos-Escrig, C. Gregorio-Rodrguez, Simulations up-to and canonical preorders (extended abstract), in: Structural Operational Semantics SOS 2007, Vol. 192 of Electronic Notes in Theoretical Computer Science, Elsevier, 2007, pp. 13–28.
[3] C. Hoare, Communicating Sequential Processes, Prentice Hall, 1985.
[4] R. Milner, Communication and Concurrency, Prentice Hall, 1989.
[5] J. Baeten, W. Weijland, Process Algebra, Cambridge Tracts in Computer Science, Cambridge University Press, 1990.
[6] R.J. v. Glabbeek, The linear time–branching time spectrum I: the semantics of concrete, sequential processes, Handbook of Process Algebra, Elsevier, 2001, pp. 3–99.
[7] E. Brinksma, A theory for the derivation of tests, in: Protocol Specification, Testing and Verification VIII, North Holland, 1988, pp. 63–74.
[8] G. Lttgen, W. Vogler, Bisimulation on speed: A unified approach, in: Foundations of Software Science and Computation Structures, 7th International Conference – FOSSACS 2005, Vol. 3441 of Lecture Notes in Computer Science, Springer, 2005, pp. 79–94.
[9] A. Kiehn, A. Arun-Kumar, Amortised bisimulations, in: Formal Techniques for Networked and Distributed Systems, 25th IFIP WG 6.1 International Conference – FORTE 2005, Vol. 3235 of Lecture Notes in Computer Science, Springer, 2005, pp. 320–334.
[10] R. De Nicola, M. Hennessy, Testing equivalence for processes, in: ICALP'83 – 10th International Colloquium on Automata, Languages and Programming, Vol. 154 of Lecture Notes in Computer Science, Springer, 1983, pp. 548–560.
[11] M. Hennessy, Algebraic Theory of Processes, MIT Press, 1988.
[12] C. Stirling, The joys of bisimulation, in: Mathematical Foundations of Computer Science, 23rd International Symposium, MFCS'98, Vol. 1450 of Lecture Notes in Computer Science, Springer, 1998, pp. 142–151.
[13] D. de Frutos-Escrig, C. Gregorio-Rodrguez, Process equivalences as global bisimulations, Journal of Universal Computer Science 12 (11) (2006) 1521–1550.
[14] M. Hennessy, R. Milner, Algebraic laws for nondeterminism and concurrency, Journal of the ACM 32 (1985) 137–161.
[15] R. Paige, R.E. Tarjan, Three partition refinement algorithms, SIAM Journal of Computing 16 (6) (1987) 973–989.
[16] P.C. Kanellakis, S.A. Smolka, CCS expressions, finite state processes, and three problems of equivalence, Information and Computation 86 (1) (1990) 43–68.
[17] R. Cleaveland, J. Parrow, B. Steffen, The concurrency workbench: a semantics-based tool for the verification of concurrent systems, ACM Transactions on Programming Languages and Systems 15 (1) (1993) 36–72.
[18] D.M. Park, Concurrency and automata on infinite sequences, in: Theoretical Computer Science, 5th GI-Conference, Vol. 104 of Lecture Notes in Computer Science, Springer, 1981, pp. 167–183.
[19] D. Sangiorgi, Bisimulation: From the origins to today, in: Proceedings of the Nineteenth Annual IEEE Symposium on Logic in Computer Science (LICS 2004), IEEE Computer Society Press, 2004, pp. 298–302.
[20] E. Sumii, B.C. Pierce, A bisimulation for type abstraction and recursion, in: Principles of Programming Languages, 32nd ACM SIGACT-SIGPLAN Symposium – POPL 2005, ACM Press, 2005, pp. 63–74.
[21] V. Koutavas, M. Wand, Small bisimulations for reasoning about higher-order imperative programs, in: Principles of Programming Languages, 33rd ACM SIGACT-SIGPLAN Symposium – POPL 2006, ACM Press, 2006, pp. 141–152.
[22] M. Merro, M. Hennessy, A bisimulation-based semantic theory of safe ambients, ACM Transactions on Programming Languages and Systems 28 (2) (2006) 290–330.
[23] E.R. Olderog, C. Hoare, Specification-oriented semantics for communicating processes, Acta Informatica 23 (1) (1986) 9–66.
[24] R. Cleaveland, S.A. Smolka, Strategic directions in concurrency research, ACM Computing Surveys. 28 (4) (1996) 607–625.
[25] K.G. Larsen, A. Skou, Bisimulation through probabilistic testing, Information and Computation 94 (1) (1991) 1–28.
[26] B. Bloom, S. Istrail, A.R. Meyer, Bisimulation can't be traced, Journal of the ACM 42 (1) (1995) 232–268.
[27] L. Aceto, W. Fokkink, A. Ingólfsdóttir, Ready to preorder: get your BCCSP axiomatization for free!, in: Algebra and Coalgebra in Computer Science, Second International Conference, CALCO 2007, Vol. 4624 of Lecture Notes in Computer Science, Springer, 2007, pp. 65–79.
[28] R. De Nicola, Extensional equivalences for transition systems, Acta Informatica 24 (2) (1987) 211–237.
[29] R. Cleaveland, M. Hennessy, Testing equivalence as a bisimulation equivalence, Formal Aspects of Computing 3 (1992) 1–21.
[30] D. de Frutos-Escrig, C. Gregorio-Rodrguez, Semantics equivalences defined with global bisimulations, Annual meeting of the IFIP Working Group 2.2, Bertinoro, Italy (September 2004).
[31] D. de Frutos-Escrig, N. López, M. Núñez, Global timed bisimulation: An introduction, in: Formal Methods for Protocol Engineering and Distributed Systems, FORTE XII / PSTV XIX, Vol. 156 of IFIP Conference Proceedings, Kluwer, 1999, pp. 401–416.
[32] J. Hughes, B. Jacobs, Simulations in coalgebra, Theoretical Computer Science 327 (1–2) (2004) 71–108.
[33] B. Klin, P. Sobocinski, Syntactic formats for free, in: CONCUR 2003 – Concurrency Theory, 14th International Conference, Vol. 2761 of Lecture Notes in Computer Science, Springer, 2003, pp. 72–86.
[34] B. Klin, A coalgebraic approach to process equivalence and a coinductive principle for traces, in: CMCS'04: 7th International Workshop on Coalgebraic Methods in Computer Science, Vol. 106 of Electronic Notes in Theoretical Computer Science, Elsevier, 2004, pp. 201–218.

[35] B. Jacobs, Trace semantics for coalgebras, in: CMCS'04: 7th International Workshop on Coalgebraic Methods in Computer Science, Vol. 106 of Electronic Notes in Theoretical Computer Science, Elsevier, 2004, pp. 167–184.
[36] M. Boreale, F. Gadducci, Denotational testing semantics in coinductive form, in: B. Rovan, P. Vojtás (Eds.), Mathematical Foundations of Computer Science 2003, 28th International Symposium, MFCS 2003, Vol. 2747 of Lecture Notes in Computer Science, Springer, 2003, pp. 279–289.
[37] J.J.M.M. Rutten, Behavioural differential equations: a coinductive calculus of streams, automata, and power series, Theoretical Computer Science 308 (1-3) (2003) 1–53.
[38] P. Gardiner, Power simulation and its relation to traces and failures refinement, Theoretical Computer Science 309 (2003) 157–176.
[39] R.J.v. Glabbeek, Bounded nondeterminism and the approximation induction principle in process algebra, in: STACS 87 – 4th Annual Symposium on Theoretical Aspects of Computer Science, Vol. 247 of Lecture Notes in Computer Science, Springer, 1987, pp. 336–347.
[40] G.D. Plotkin, A structural approach to operational semantics, Tech. Rep. DAIMI FN-19, Computer Science Department, Aarhus University (1981).
[41] G.D. Plotkin, A structural approach to operational semantics, Journal of Logic and Algebraic Programming 60-61 (2004) 17–139.
[42] D. Sangiorgi, R. Milner, The problem of "Weak Bisimulation up to", in: CONCUR '92 – Concurrency Theory, Third International Conference, Vol. 630 of Lecture Notes in Computer Science, Springer, 1992, pp. 32–46.
[43] D. Sangiorgi, On the bisimulation proof method, Journal of Mathematical Structures in Computer Science 8 (5) (1998) 447–479.
[44] D. Pous, Weak bisimulation up to elaboration, in: CONCUR 2006 – Concurrency Theory, 17th International Conference, Vol. 4137 of Lecture Notes in Computer Science, Springer, 2006, pp. 390–405.
[45] D. Pous, New up-to techniques for weak bisimulation, Theoretical Computer Science 380 (1–2) (2007) 164–180.
[46] S. Blom, W. Fokkink, S. Nain, On the axiomatizability of ready traces, ready simulation, and failure traces, in: Automata, Languages and Programming, 30th International Colloquium, ICALP 2003, 2003, pp. 109–118.
[47] I. Hasuo, Generic forward and backward simulations, in: CONCUR 2006 – Concurrency Theory, 17th International Conference, Vol. 4137 of Lecture Notes in Computer Science, Springer, Berlin, 2006, pp. 406–420.
[48] I. Hasuo, B. Jacobs, A. Sokolova, Generic trace theory, in: CMCS'06 – 8th International Workshop on Coalgebraic Methods in Computer Science, Vol. 164 of Electronic Notes in Theoretical Computer Science, Elsevier, 2006, pp. 47–65.
[49] D. de Frutos-Escrig, F. Rosa-Velardo, C. Gregorio-Rodrguez, New bisimulation semantics for distributed systems, in: Formal Techniques for Networked and Distributed Systems, 27th IFIP WG 6.1 International Conference – FORTE 2007, Vol. 4574 of Lecture Notes in Computer Science, Springer, 2007, pp. 143–159.
[50] R.J.v. Glabbeek, The linear time - branching time spectrum II, in: CONCUR '93 – Concurrency Theory, 5th International Conference, Vol. 715 of Lecture Notes in Computer Science, Springer, 1993, pp. 66–81.
[51] D. de Frutos-Escrig, C. Gregorio-Rodrguez, Universal coinductive characterizations of process semantics, in: 5th IFIP International Conference on Theoretical Computer Science, Vol. 273 of IFIP, Springer, 2008, pp. 397–412.
[52] D. de Frutos-Escrig, C. Gregorio-Rodrguez, M. Palomino, Ready to preorder: an algebraic and general proof, Journal of Logic and Algebraic Programming, to appear, doi:10.1016/j.jlap.2008.09.001.
[53] D. de Frutos-Escrig, C. Gregorio-Rodríguez, M. Palomino, Coinductive characterisations reveal nice relations between preorders and equivalences, in: First International Conference on Foundations of Informatics, Computing and Software – FICS 2008, Vol. 212 of Electronic Notes in Theoretical Computer Science, Elsevier, 2008, pp. 149–162.