

Bisimulations Up-to for the Linear Time Branching Time Spectrum*

David de Frutos Escrig and Carlos Gregorio Rodríguez

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

Abstract. Coinductive definitions of semantics based on bisimulations have rather pleasant properties and are simple to use. In order to get coinductive characterisations of those semantic equivalences that are weaker than strong bisimulation we use a variant of the bisimulation up-to technique in which we allow the use of a given preorder relation. We prove that under some technical conditions our bisimulations up-to characterise the kernel of the given preorder. It is remarkable that the adequate orientation of the ordering relation is crucial to get this result. As a corollary, we get nice coinductive characterisations of all the axiomatic semantic equivalences in Van Glabbeek's spectrum. Although we first prove our results for finite processes, reasoning by induction, then we see, by using continuity arguments, that they are also valid for infinite (finitary) processes.

1 Introduction

Along the years a great variety of concurrent 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 [Gla01] he presents the well known linear time-branching time spectrum for processes without internal transitions. There, fifteen different semantics are defined and ordered by their inclusion relations. Besides, for each equivalence a motivating testing scenario is provided, and for most of them, a complete axiomatisation for basic processes is given. Figure 1 shows these axiomatised semantics (but tree semantics) ordered by inclusion.

Not just because it is the strongest one of them, bisimulation [Par81, Mil89] merits a special attention. Bisimulation is a mathematically elegant concept that is recursively defined over the intensional description of processes. Its stability and elegance have been shown by several characterisations, for instance in terms

* Partially supported by the projects TERMAS TIC2003-07848-C02-01, MIDAS TIC2003-01000, PAC-03-001 and MRTN-CT-2003-505121/TAROT.

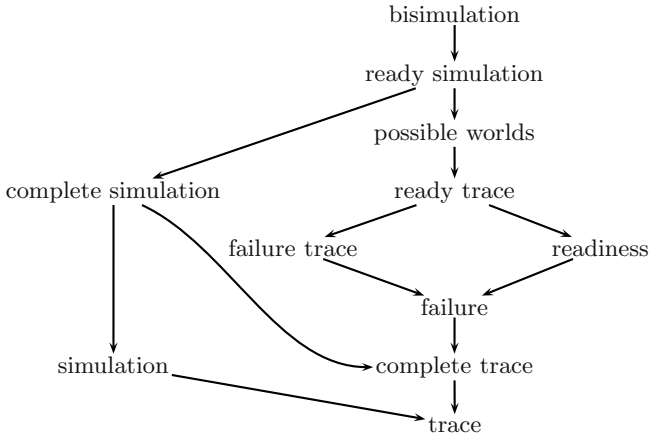


Fig. 1. Axiomatic Semantics in the Linear Time-Branching Time Spectrum I

of modal logic, final coalgebras, testing, etc. There also exist efficient algorithms to decide bisimulation equivalence and several tools that can effectively check process bisimilarity.

However, bisimulation is also too strong, and in many cases it is enough to take into account some weaker semantics. But, most of the semantics in Fig. 1 are extensional ([CS96]) and none of them has a symmetric, coinductive definition as bisimulation does. It is true that all 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?

In this paper we propose a way to weaken the definition of bisimulation by using a preorder relation, what we call *bisimulation up-to* the preorder. In this way we obtain a conductively and symmetrically defined equivalence, parameterised by 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 that 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.

With these results we have answered the questions we left open in [dFG04], where we studied ready simulation as a representative example. There, we defined our *global bisimulations* that are indeed closely related to bisimulations up-to. They were previously introduced in a different context in [dFLN99].

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 rather coalgebraic framework [JH03, KS03, Kli04, Jac04] and, in many cases, are based on relatively complex categorical concepts. These

works aim generality and their results are rather general, but just because of that the machinery to apply them in particular 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.

Rutten [Rut03] has also made a coalgebraic approach to the subject but based on the novel concept of *behavioural differential equations*. Boreale and Gadducci in [BG03] have applied this technique to define a fully abstract model for the failures semantics. However the extension to other semantics seems not easy.

A different approach is presented in [Gar03] 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.

The rest of the paper is structured as follows. In Sec. 2 definitions and notations on processes and preorders are presented. In Sec. 3 we define bisimulations up-to a preorder and present the main results of the paper, namely Theorems 1 and 2. As a corollary of Theorem 1 all the semantics in Fig. 1 can be expressed by a bisimulation-like definition. Some examples help to clarify the role of the conditions in the theorems. In Sec. 4 the results of the previous section are extended to infinite finitary tree-like processes. In Sec. 5 we discuss a simple application example. Finally, in Sec. 6 we present some conclusions and lines for future work.

Along this paper we make use of the semantics in Fig. 1. Most of them can be considered *classical* and are well known, anyway we refer to [Gla01] for formal definition of each semantics and to Tables 2 and 3 in that paper for the complete axiomatisation for the equivalences and preorders, respectively, that we use in some of our examples and proofs.

2 Processes and Preorders

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

Definition 1. *A labelled transition system is a structure $\mathcal{T} = (\mathcal{P}, Act, \rightarrow)$ where*

- \mathcal{P} is a set of processes, agents or states,
- Act is a set of actions and
- $\rightarrow \subseteq \mathcal{P} \times Act \times \mathcal{P}$ is a transition relation.

A rooted lts is a pair (\mathcal{T}, p_0) with $p_0 \in \mathcal{P}$.

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

Some usual notations on lts are used. We write $p \xrightarrow{a}$ if there exists a process q such that $p \xrightarrow{a} q$ and, on the contrary, we write $p \not\xrightarrow{a}$ if there exists no process q such that $p \xrightarrow{a} q$. For a string of actions $\sigma = a_1 a_2 \dots a_n$, $a_i \in Act$, $p \xrightarrow{\sigma} q$ means that there exist processes $q_1 \dots q_{n-1}$, such that $p \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \xrightarrow{a_3} \dots q_{n-1} \xrightarrow{a_n} q$. The function I calculates the set of initial actions of a process, $I(p) = \{a \mid a \in Act \text{ and } p \xrightarrow{a}\}$.

Lts's for finite processes are just finite trees, which can be syntactically described by a basic process algebra BCCSP, which was also used in [Gla01].

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

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

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

Therefore, BCCSP is just the term algebra for the signature $(\mathbf{0}, a \in Act, +)$. The set of rooted lts's is also the support of such an algebra, by defining prefix and choice operators in the natural way. All the definitions in the paper are valid for arbitrary processes, that is, for arbitrary rooted lts's. However we are going to prove the main results in the paper in two steps. First, we reason by induction on the depth of processes, and therefore the results would only be valid, at the moment, for BCCSP processes. Second, we use continuity arguments to extend these results to a general class of infinite tree-like processes.

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.

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

Fig. 2. Operational Semantics for BCCSP Terms

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

Preorders, that we represent by \sqsubseteq , are reflexive and transitive relations. We use the symbol \sqsupseteq to represent the preorder relation \sqsubseteq^{-1} . Every preorder induces an equivalence relation that we denote by \equiv ; that is, $p \equiv q$ if and only if $p \sqsubseteq q$ and $q \sqsubseteq p$. We will denote by $=_B$ the bisimulation equivalence. We are interested on preorders that are weaker than it.

Definition 3. A preorder relation \sqsubseteq over processes is a behaviour preorder when it is weaker than the bisimulation equivalence, i.e. $p =_B q \Rightarrow p \sqsubseteq q$, and it is a precongruence with respect to the prefix and choice operators, i.e. if $p \sqsubseteq q$ then $ap \sqsubseteq aq$; and if $p \sqsubseteq q$ then $p + r \sqsubseteq q + r$.

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 $a \in I(p)$.

Initials preservation and factorisation are natural properties that are satisfied by any of the behaviour preorders corresponding to the semantics in Fig. 1, from trace preorder to ready simulation preorder (Table 3 in [Gla01] shows the axiomatisation of these preorders).

There are other properties that a behaviour preorder can satisfy and that are going to play an important role in the rest of the paper. We say that a behaviour preorder \sqsubseteq satisfies the property

$$\begin{aligned} (S) \quad & \text{if for all } p \text{ and } q, p \sqsubseteq p + q \\ (CS) \quad & \text{if for all } a, p \text{ and } q, ap \sqsubseteq ap + q \\ (RS) \quad & \text{if for all } a, p \text{ and } q, ap \sqsubseteq ap + aq \end{aligned}$$

These axioms characterise the simulation preorder, the complete simulation preorder and the ready simulation preorder, respectively.

We finish this section by introducing another interesting property.

Definition 5. Let \sqsubseteq be a behaviour preorder and \equiv the induced equivalence. Then \sqsubseteq has the Hoare equivalence property¹ (HE for short) whenever

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

3 Bisimulation Up-to a Preorder

In Sec. 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 [Par81] and it has become one of the fundamental notions in the theory of concurrent processes. It is defined as follows.

Definition 6 ([Mil89]). A binary relation \mathcal{R} is called a (strong) bisimulation if for all p, q processes such as $p \mathcal{R} q$, 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 $p' \mathcal{R} q'$.

¹ The name comes from Hoare's powerdomain construction.

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 [Gla01] for details).

In [Mil89], 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 [Mil89], was wrong. Later, in [SM92] two new up-to (now correct, but more involved!) techniques were proposed. Sangiorgi continued with the study of up-to techniques in [San98], but focusing on reducing the size of the bisimulation relations to prove that two given processes are bisimilar.

In this paper we retake the concept of bisimulation up-to but we use it with a different goal. We are looking for the adequate way to weaken the definition of bisimulation in such a manner that weaker equivalences can be captured by a coinductive definition.

Definition 7. Let \sqsubseteq be a behaviour preorder. Then a binary relation S over processes is a bisimulation up-to \sqsubseteq , if pSq implies that:

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

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

The key point in the previous definition is that the process that has to mimic the movement of the other is allowed to: first, to transform itself according to the inverse of the considered preorder relation; second, to execute the corresponding action. The added capability generalises the original definition of bisimulation, so that we have now more chances to prove the equivalence between processes. When the behaviour preorder is just the identity relation we get the bisimulation equivalence, but, as we are going to prove below, considering other behaviour preorders we will be able to get other interesting semantics (traces, failures, ready simulation and so on).

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

Proposition 1. For every behaviour preorder \sqsubseteq , if $p \equiv q$ then $p \approx q$.

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 $p \sqsupseteq q \xrightarrow{a} q'_a$.

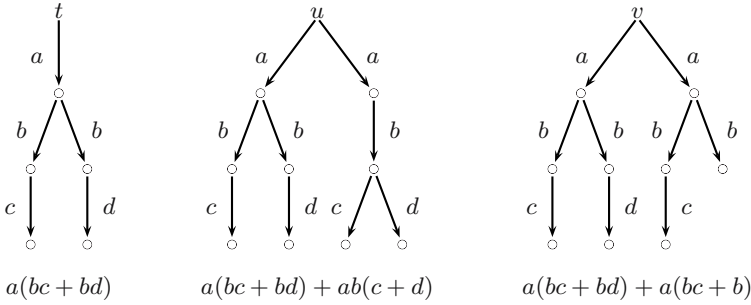


Fig. 3. Examples of Processes

Example 1. Let us consider processes t and v in Fig. 3. Let \sqsubseteq_S be the simulation preorder, $=_S$ the induced equivalence, and $=_B$ the (strong) bisimulation equivalence. Processes t and v are not (strongly) bisimilar, $t \neq_B v$, but they are bisimilar up-to the simulation preorder, $t \approx_{\sqsubseteq_S} v$. The only difficult point to find 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 conclude the proof.

Lemma 1. For every initials preserving behaviour preorder \sqsubseteq , if $p \approx q$ 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 1. For every behaviour preorder \sqsubseteq , that is initials preserving, action factorised and satisfying the axiom (RS), we have that $p \approx q$ if and only if $p \equiv q$.

Proof. If $p \equiv q$ then $p \approx q$ is proved in Proposition 1. We prove the reverse implication, if $p \approx q$ then $p \equiv q$. We proceed by induction on the depth of process p and prove that if $p \approx q$ then $p \sqsubseteq q$.

By definition of $p \approx q$, if $p \xrightarrow{a} p'_a$ then $q \sqsupseteq q' \xrightarrow{a} q'_a$ and $p'_a \approx q'_a$. 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, due to the factorised property, $q|_a \sqsupseteq q'|_a$.

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 have also $I(r) = \{a\}$. 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

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

and therefore

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

Finally, by Lemma 1, $I(p) = I(q)$ and we conclude that $\sum_i q |_{a_i} = q$ and therefore $p \sqsubseteq q$.

This result even if simple is rather general, all the preorders for the semantics in Fig. 1 below the ready simulation 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 any equivalence in the linear time-branching time spectrum from trace equivalence to ready simulation equivalence. Besides, as a corollary, we get that for any of the equivalence relations, defined by the semantics in Fig. 1, it is true also that \approx_{\equiv} is equal to \equiv .

Example 2. Let us retake again our Example 1. As the simulation preorder is one of those in the conditions of Theorem 1, the fact that $t \approx v$ is enough to conclude that t and v are simulation equivalent, that is, we have proved it by constructing a single bisimulation instead of two simulations $t \sqsubseteq_S v$ and $v \sqsubseteq_S t$.

The conditions imposed to the behaviour preorders in Theorem 1 suggest that not every preorder is adequate to get the induced equivalence by means of a bisimulation up-to. Next we comment some examples.

Example 3. 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 $q \xrightarrow{a} q'$, $p \sqsupseteq p + q \xrightarrow{a} q'$. Note that we have not contradicted Theorem 1 because the preorder \sqsubseteq is not initials preserving.

There exist also other preorders which do not allow bisimulation up-to characterization via Theorem 1 because they do not fulfil the axiom (RS) as shown by the next example.

Example 4. Let us consider the behaviour preorder relation that is induced by the axiom $a(p + q) \sqsubseteq ap + aq$. Obviously, by definition, this relation is action factorised and initials preserving, but does not satisfy the axiom (RS). 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 :

- 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 movement;
- If u performs action a and evolves into u' , then t can delay its choice and reduces to $ab(c + d)$, then performing action a , evolves also into u' .

Although the range from trace equivalence to ready simulation equivalence is quite wide and most of the classic semantics fall into it, we have studied whether the use of the bisimulations up-to is also possible outside these margins. We have found that there is a family of semantic preorders for which the bisimulations up-to work properly. Any preorder in this family is a simulation (see, for instance, [Gla01]).

Lemma 2. *For every behaviour preorder \sqsubseteq being a simulation, whenever $p \sqsubseteq p' \xrightarrow{a} p'_a$, there exists p_a such that $p \xrightarrow{a} p_a \sqsubseteq p'_a$.*

Proof. By definition of simulation.

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

Theorem 2. *For every behaviour preorder \sqsubseteq , being a simulation and satisfying the Hoare equivalence property, $p \approx q$ if and only if $p \equiv q$.*

Proof. If $p \equiv q$ then $p \approx q$ is proved by Proposition 1. The reverse implication, if $p \approx q$ then $p \equiv q$, is proved by induction on the depth of the first process.

Let us consider $p \approx q$. Then whenever $p \xrightarrow{a} p'_a$ there exist q' and q'_a such that $q \sqsubseteq q' \xrightarrow{a} q'_a$ and $p'_a \approx q'_a$ and, by induction hypothesis, $p'_a \equiv q'_a$. As the behaviour preorder is a simulation, by Lemma 2, there exists q_a such that $q \xrightarrow{a} q_a \sqsubseteq q'_a$. Therefore, for some process r , it is true that $q = aq_a + r \sqsubseteq aq'_a + r \equiv ap'_a + r$. That is, for every $p \xrightarrow{a} p'_a$ there exists q_a such that $q_a \sqsubseteq p'_a$.

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

Both the simulation preorder and the ready simulation preorder are simulations and satisfy the HE property, so for these preorders Theorem 2 provides an alternative proof to that of Theorem 1. But there are other interesting preorders that induce equivalences between strong bisimulation and ready simulation equivalence for which Theorem 2 provide a characterisation in terms of bisimulation up-to.

Example 5. Let us consider the preorder \sqsubseteq_{FS} defined as $p \sqsubseteq_{FS} q$ if there exist 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 $F(p) = \{(a, X) \mid a \in I(p), X \subset Act \text{ } p \xrightarrow{a} p' \text{ and } X \cap I(p') = \emptyset\}$

That is, \sqsubseteq_{FS} is much alike 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 2: obviously it is a simulation and it can be easily checked that satisfies the HE property. Therefore, bisimulation up-to \sqsubseteq_{FS} defines the same equivalence relation as $\sqsubseteq_{FS} \cap \sqsubseteq_{FS}^{-1}$. To check that the induced equivalence is finer than the ready simulation equivalence let us consider, for instance, the processes $p = a(bc + bd)$ and $q = abc + a(bc + bd)$, that are ready simulation equivalent but $q \not\sqsubseteq_{FS} p$.

Following the ideas in the previous example it is quite easy to find other *constrained* simulations in the conditions of Theorem 2 that define equivalences between the ready simulation and strong bisimilarity. Some of them can be defined axiomatically in an easy way, as that in the following example.

Example 6. Let us consider the axiom $a(p + q) \sqsubseteq a(p + q) + ap$ and the induced behaviour preorder. This preorder refines the axiom of the simulation preorder, and satisfies the HE property.

Next example points out the necessity of the HE property in the conditions of Theorem 2.

Example 7. 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 $n \sqsubseteq m$ and $m \not\sqsubseteq n$ 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 movement;
- If m performs action a and evolves into bc then n can be reduced by the preorder to $a(bc + b(c + d)) \sqsubseteq abc$, and then, performing a , it evolves into bc .

4 Bisimulations 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. We will use the same notation as for finite trees (prefix, choice, multiple choice...) 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 give way to level continuous bisimulations up-to. Once this result is stated, Theorems 1 and 2 can also be proved for level continuous behaviour preorders, using simple continuity reasonings. The definition of level continuity is rather natural, so that every behaviour preorder for the semantics in Fig. 1 is indeed level continuous.

Definition 8. A behaviour preorder is level continuous if $p \sqsubseteq q$ if and only if for all n $p \downarrow_n \sqsubseteq q \downarrow_n$ where $p \downarrow_n$ is the result of pruning process p below level n , that is:

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

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 cardinal of a set A and $[p]_{=B}$ to denote the equivalence class of p with respect to bisimulation equivalence, $=_B$.

Lemma 3. If the alphabet of actions Act is finite, for any 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 $p = \sum_i a p_a^i$ and $q = \sum_j a q_a^j$, 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 a action, $\{[p_a^i]_{=B}\} = \{[q_a^j]_{=B}\}$, therefore, the elements of $\{[p]_{=B} \mid depth(p) \leq n + 1\}$ are in one to one correspondence with functions in $Act \rightarrow \mathcal{P}(\{[p]_{=B} \mid depth(p) \leq n\})$. 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 4. If a behaviour preorder \sqsubseteq is stronger than the trace preorder ($\sqsubseteq \Rightarrow \sqsubseteq_T$), for any finite process q , then the set of bisimilarity classes $\{[p]_{=B} \mid p \sqsubseteq q\}$ is finite.

Proof. Since $\sqsubseteq \Rightarrow \sqsubseteq_T$ we have that $p \sqsubseteq q \Rightarrow depth(p) \leq depth(q)$ 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 3.

Proposition 2. For every behaviour preorder \sqsubseteq , and the corresponding bisimulation up-to \sqsubseteq, \approx , if \sqsubseteq is level continuous then \approx is level continuous too.

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

Let S be a bisimulation up-to \sqsubseteq , then $S_f = \{(p \downarrow_n, q \downarrow_n) \mid p S q\}$ is also a bisimulation up-to \sqsubseteq . Whenever $p \downarrow_n \xrightarrow{a} p'_a \downarrow_{n-1}$, because of the level continuity of \sqsubseteq , $q \downarrow_n \sqsupseteq q' \downarrow_n \xrightarrow{a} q'_a \downarrow_{n-1}$, 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 $R = \{(p, q) \mid \text{for all } n \ p \downarrow_n \approx q \downarrow_n\}$. 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 $p'_a \downarrow_{n-1} \approx q'_{n,a}$.

It is easy to check that for all $m > n$, $p'_a \downarrow_{n-1} \approx q'_{m,a} \downarrow_{n-1}$. Then, we define $Q_n^m = \{q'_m \downarrow_n \mid q \downarrow_m \sqsupseteq q'_m \xrightarrow{a} q'_{m,a} \text{ and } p'_a \downarrow_{n-1} \approx q'_{m,a}\}$ 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 4, $Q_n^{m'} /_{=B} \subseteq Q_n^m /_{=B}$ 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 $p'_a \downarrow_{n-1} \approx q'_{n,a}$, so that we have both $q \sqsupseteq q'$ and $q' \xrightarrow{a} q'_a$ with $p'_a \downarrow_{n-1} \approx q'_a \downarrow_{n-1}$, thus proving that the pair $(p'_a, q'_a) \in R$, so that R is indeed a bisimulation up-to \sqsubseteq .

All the preorders for the semantics in Fig. 1 are level continuous. We give the proof for two representative examples.

Proposition 3. *The trace preorder \sqsubseteq_T is level continuous.*

Proof. $p \sqsubseteq_T q$ 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 , $p \downarrow_n \sqsubseteq_T q \downarrow_n$.

Proposition 4. *The ready simulation preorder \sqsubseteq_R is level continuous.*

Proof. $p \sqsubseteq_R q$ 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. Firstly, $I(p) = I(q)$, so that, whenever pRq we have $I(p) = I(q)$. Then, whenever $p \xrightarrow{a} p'$, we know $p \downarrow_n \xrightarrow{a} p' \downarrow_{n-1}$ 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 descendent 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 as $q'_{i(n)}$ this q' for any n . Then, $p' \downarrow_n \sqsubseteq_R q' \downarrow_n$ for all n and then $p'Rq'$, proving that R is a ready simulation containing the pair (p, q) .

Thus for any level continuous preorder verifying the hypothesis of any of the theorems in Sec. 3 the results of these theorems are also valid for infinite processes.

5 A Simple Application Example

As a simple application we present the same example used by Klin in [Kli04]. 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 care when coping with infinite processes. As Klin, we use here a coalgebraic reasoning to do it, but our proof is simpler than that in [Kli04], although it is true that Klin develops his approach in a broader framework than ours.

Definition 9. For any process $p = \sum_a \sum_i ap_{a,i}$ the deterministic form of p is defined as $Det(p) = \sum_a aDet(\sum_i p_{a,i})$.

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

Lemma 5. For any processes p and q we have that $Det(p) \sqsubseteq_T Det(p + q)$.

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

Proposition 5. For any process p , $p \approx_{\sqsubseteq_T} Det(p)$.

Proof. We will prove that the relation $R = \{(p, Det(p)) \mid p \text{ is a process}\}$ is a bisimulation up-to \sqsubseteq_T . Whenever $p \xrightarrow{a} p_{a,i}$, then, by using Lemma 5, $Det(p) = \sum_a aDet(\sum_i p_{a,i}) \sqsupseteq_T aDet(p_{a,i}) \xrightarrow{a} Det(p_{a,i})$. Besides, if $Det(p) \xrightarrow{a} Det(\sum_i p_{a,i})$, 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 $p \sqsupseteq_T a \sum p_{a,i} \xrightarrow{a} \sum p_{a,i}$.

It is important to note that even if in the definition of bisimulation up-to we have the full power of the trace preorder, we just use a simple part of it, namely that corresponding to the result of Lemma 5. As a matter of fact, we are just transferring the way *bisimulation up-to bisimulation* is used to prove bisimilarity between processes. Therefore, we are just proving the initial part of the property in which we are interested and coinduction makes the rest, by means of bisimulation up-to.

6 Conclusions and Future Work

We have defined the notion of bisimulation up-to a preorder. This settles a framework in which to define, in a coalgebraic flavour, many of the classical equivalences of process semantics, and therefore, the possibility of reasoning about them by using coinduction.

We have also transferred the up-to preorder technique to the simulation framework. Using simulations up-to preorders we have obtained coinductive characterisations of the considered preorders. Besides, simulations up-to and bisimulations up-to can be related concluding that, under similar conditions than those

in the results in the paper, two processes are bisimilar up-to a preorder if and only if they are mutually similar up-to it. Due to lack of space it has not been possible to reproduce here these results.

Although, obviously, it is not possible to avoid the high complexity of the equivalence problem with respect to most of the classical semantics, our results open the door for using the tools to check bisimilarity to decide other equivalences. In fact, some results already exist in that direction. A seminal paper relating testing semantics and bisimulation is [CH92]. There the authors change the transition system defining the operational semantics of processes, to get a more complex and (bigger) transition system where bisimulation corresponds to the original testing semantics. More recently, Kucera and Mayr have related simulation and bisimulation. First, in [KM02a] they prove that bisimulation can be easily translated into simulation, so proving that to decide the latter is at least as expensive as the former. In [KM02b] the opposite reduction is studied, and the results are similar to those in [CH92], but for the simulation semantics. They use an ad-hoc technique to transform the original transition system into a suitable transition system that, in this case, is smaller than the original one, but much more difficult to obtain, although they also prove that for a class of Petri Nets with at most one unbounded place the transformation can be effectively done.

As work in progress, we are studying the other semantics in the linear time-branching time not discussed in this paper, namely, the nested simulation semantics. They are the only ones for which Van Glabbeek provides no axiomatisation.

Moreover, the study of bisimulations up-to has showed us that all the semantics in Van Glabbeek's spectrum have always a simulation part, corresponding to axioms such as $p \sqsubseteq p + q$, that characterises the intensional behaviour and, possibly, another component that characterises the extensional behaviour, for instance, $a(p + q) = ap + aq$ for the trace semantics. In particular, we are interested in the axiomatisations and we are looking for a systematic way to relate the axioms of the preorders with those of the corresponding equivalences.

References

- [BG03] Michele Boreale and Fabio Gadducci. Denotational testing semantics in coinductive form. In Branislav Rován and Peter Vojtás, editors, *28th International Symposium, MFCS 2003*, volume 2747 of *Lecture Notes in Computer Science*, pages 279–289. Springer, 2003.
- [CH92] Rance Cleaveland and Matthew Hennessy. Testing equivalence as a bisimulation equivalence. *Formal Aspects of Computing*, 3:1–21, 1992.
- [CS96] Rance Cleaveland and Scott A. Smolka. Strategic directions in concurrency research. *ACM Computing Surveys.*, 28(4):607–625, 1996.
- [dFG04] David de Frutos-Escrig and Carlos Gregorio-Rodríguez. Semantics equivalences defined with global bisimulations. Annual meeting of the IFIP Working Group 2.2, Bertinoro, Italy, September 2004.
- [dFLN99] David de Frutos-Escrig, Natalia López, and Manuel Núñez. Global timed bisimulation: An introduction. In *Formal Methods for Protocol Engineering*

and Distributed Systems, FORTE XII / PSTV XIX, pages 401–416. Kluwer Academic Publishers, 1999.

- [Gar03] Paul Gardiner. Power simulation and its relation to traces and failures refinement. *Theoretical Computer Science*, 309:157–176, 2003.
- [Gla01] Rob J. van Glabbeek. *Handbook of Process Algebra*, chapter The Linear Time – Branching Time Spectrum I: The Semantics of Concrete, Sequential Processes, pages 3–99. Elsevier, 2001.
- [Jac04] Bart Jacobs. Trace semantics for coalgebras. In *CMCS'04: 7th International Workshop on Coalgebraic Methods in Computer Science*, volume 106 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2004.
- [JH03] Bart Jacobs and Jesse Hughes. Simulations in coalgebra. In *CMCS'03: 6th International Workshop on Coalgebraic Methods in Computer Science*, volume 82 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2003.
- [Kli04] Bartek Klin. A coalgebraic approach to process equivalence and a coinductive principle for traces. In *CMCS'04: 7th International Workshop on Coalgebraic Methods in Computer Science*, volume 106 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2004.
- [KM02a] Antonín Kucera and Richard Mayr. Why is simulation harder than bisimulation? In *CONCUR 2002 - Concurrency Theory, 13th International Conference, Proceedings*, volume 2421 of *Lecture Notes in Computer Science*, pages 594–610. Springer, 2002.
- [KM02b] Antonín Kucera and Richard Mayr. Simulation preorder over simple process algebra. *Information and Computation*, 173(2):184–198, 2002.
- [KS03] Bartek Klin and Pawel Sobocinski. Syntactic formats for free. In *14th International Conference, CONCUR 2003 - Concurrency Theory, Proceedings*, volume 2761 of *Lecture Notes in Computer Science*, pages 72–86. Springer, 2003.
- [Mil89] Robin Milner. *Communication and Concurrency*. Prentice Hall, 1989.
- [Par81] David M.R. Park. Concurrency and automata on infinite sequences. In *Theoretical Computer Science, 5th GI-Conference*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer, 1981.
- [Plo81] Gordon D. Plotkin. A structural approach to operational semantics. Technical Report DAIMI FN-19, Computer Science Department, Aarhus University, 1981.
- [Rut03] Jan J. M. M. Rutten. Behavioural differential equations: a coinductive calculus of streams, automata, and power series. *Theoretical Computer Science*, 308(1-3):1–53, 2003.
- [San98] Davide Sangiorgi. On the bisimulation proof method. *Journal of Mathematical Structures in Computer Science*, 8(5):447–479, 1998.
- [SM92] D. Sangiorgi and R. Milner. The problem of “Weak Bisimulation up to”. In W.R. Cleveland, editor, *Proc. CONCUR '92*, volume 630 of *Lecture Notes in Computer Science*, pages 32–46. Springer, 1992.