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
doi:10.1016/j.ic.2007.12.003

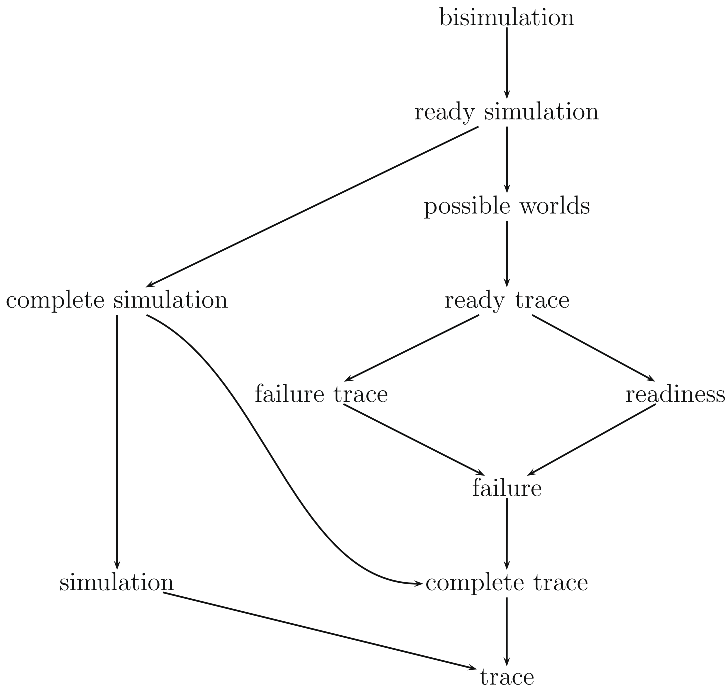

Fig. 1. Axiomatic semantics in the linear time-branching time spectrum I.

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