On Relating Concurrency and Nondeterminism*

Luca Aceto
INRIA - Sophia Antipolis
F-06560 Valbonne Cedex, France

Abstract

We present an intuitive preorder for a simple CCS-like language whose semantic theory allows us to relate concurrency and nondeterminism without reducing the former to the latter. The preorder over processes is induced by using an equationally defined preorder over computations in a bisimulation-like protocol. The relationships of the proposed preorder with pomset bisimulation and standard strong bisimulation equivalence are studied in detail. Moreover, we give an axiomatization of the preorder over recursion-free processes.

1 Introduction

In recent years a lot of research has been devoted to the study of notions of equivalence for Process Algebras, such as CCS [Mil89], CSP [Hoare85] and ACP [BK85]. The interest in equivalences and preorders, which relate descriptions of concurrent systems in terms of these languages, stems from the fact that process algebras are used not only for describing actual systems, but also their specifications. Notions of equivalence between descriptions are thus an important component of these languages as they allow one to formally state when (the description of) a system is a correct implementation of a given specification. Roughly, the proposals presented in the literature may be divided into two broad classes:

1. the equivalences and preorders which semantically reduce parallelism to sequential nondeterminism, and

2. those whose semantic theories treat parallelism as a primitive notion.

The equivalences which semantically reduce parallelism to sequential nondeterminism are usually called *interleaving equivalences* as in their associated theories concurrency between events is interpreted as their arbitrary interleaving (i.e. their possibility to occur in any temporal order). Several different notions of equivalence, based upon the interleaving approach, have been proposed in the

*This work has been supported by a grant from the United Kingdom Science and Engineering Research Council and by the Esprit BRA project CEDISYS.
literature; the most widely used amongst them are trace equivalence [Hoare85], observational equivalence [Mil80], bisimulation equivalence [Pa81], testing equivalence [DH84], [H88a I and failure equivalence [BHR84]. Although their theories are very different from each other, all of these behavioural equivalences have much in common; the only form of abstraction supported by the above-mentioned equivalences is related to nondeterminism. Processes can only be equivalent if their behaviour is the same modulo nondeterminism.

Several researchers have recently argued that, although they allow a faithful description of the interactive behaviour of processes, the assumptions underlying the interleaving semantic models are inadequate in accounting for the nonsequential behaviour of distributed systems and that semantic theories which consider parallelism as a primitive notion are best suited for this purpose. Consequently, several equivalences for process algebras which distinguish concurrency from sequential nondeterminism have recently been proposed in the literature, e.g. distributed bisimulation [CH87], timed equivalence [H88b], pomset bisimulation [BC87,88], NMS-bisimulation [DDM86]. A variety of equivalences has been discussed in [GV87] in the context of Petri Nets [Rei85]. All of these equivalences are based on adaptations of the standard notion of bisimulation equivalence and draw a sharp line between concurrent execution of actions and their interleavings, between causal and temporal dependencies among computational events.

Although the above-mentioned equivalences differ in the degree in which they model the interplay between causality and the branching structure of processes, they identify descriptions of processes only if they can exhibit the same degree of parallelism. However, it may be argued that, whereas the interleaving equivalences are too coarse in that they forget too much of the structure of processes, the above-mentioned equivalences are perhaps too discriminating in that they do not allow one to relate “concurrent implementations” and “nondeterministic specifications” at all. As it is frequently more natural to specify the behaviour of a system in terms of a sequential nondeterministic process and more efficient to implement it in a parallel fashion, it would be helpful to have a semantic theory of processes which allows us to relate these two notions without semantically reducing parallelism to sequential nondeterminism. One possible use of this feature of the theory is in requiring that all the parallelism which is present in a specification be maintained in the implementation.

The main aim of this paper is to provide such a semantic theory for a simple CCS-like language and to show how standard tools used in defining interleaving and non-interleaving equivalences for this language may be adapted in order to reconcile both philosophies to the semantics of concurrency.

Intuitively, our proposal is based upon a preorder \( \leq \) over processes such that, for processes \( p \) and \( q \), \( p \leq q \) if, and only if, \( p \) and \( q \) have the same “interactive behaviour” and \( q \) is “at least as parallel as” \( p \). In order to formalize this idea, one needs to make precise the notion of “interactive behaviour” over processes and to give a formal way of measuring the degree of parallelism a process may exhibit during its evolution. In this paper we take the view that a reasonable notion of equivalence between the interactive behaviour of two concurrent, nondeterministic processes is captured by bisimulation equivalence, [Pa81]. Of course there is some arbitrariness in the choice of such an interleaving equiva-