ABSTRACT

Processes are mathematical objects which are elements of domains in the sense of Scott and Plotkin. Process domains are obtained as solutions of equations solved by techniques from metric topology as advocated by Nivat. We discuss how such processes can be used to assign meanings to languages with concurrency, culminating in a definition of the ADA rendez-vous. An important intermediate step is a version of Hoare's CSP for which we describe a process semantics and which is used, following Gerth, as target for the translation of the ADA fragment. Furthermore, some ideas will be presented on a mathematically tractable treatment of fairness in the general framework of processes.

1. INTRODUCTION

This paper presents a case study in the area of the semantics of concurrency. In the initial years of the theory of concurrency, most of the attention was devoted to notions such as composition and synchronization of parallel processes – often established through suitably restricted interleaving of the elementary actions of the components, and mostly referring to a shared variable model. More recently there has been a considerable increase in the interest for communication between processes – often referring to a model where the individual processes have disjoint variables which interact only through the respective communication mechanisms. Instrumental in this development have been the studies of BRINCH HANSEN [6], HOARE [10] and MILNER [15], where a variety of forms of communication was proposed and embedded in a language design or studied with the tools of operational and denotational semantics. The incorporation of the notions of tasking and rendez-vous in the language ADA ([1]) provides additional motivation for the study of communication, and it is the latter notion in particular which we have chosen as the topic of our investigation.

The main purpose of our paper is firstly to provide a rigorous definition for the ADA rendez-vous with the tools of denotational semantics, and secondly to introduce a mathematically tractable approach to fairness which is applicable in general in various situations where choices have to be made on a fair basis, and in particular to the ADA rendez-vous definition.

The general framework we apply in our paper was first outlined in DE BAKKER & ZUCKER [3], and later described in detail in DE BAKKER & ZUCKER [4]. In order to keep the present paper self-contained, we shall provide a summary description of the main points of the latter paper, without going into much mathematical detail. Our approach to the ADA rendez-vous and to fairness owes much to two contributions to ICALP 82.
In GERTH [8] the idea of translating the ADA fragment to a version of CSP was proposed; the same approach will be applied by us in section 6. In PLOTKIN [19], the fundamental idea of specifying a fair merge through suitable use of - essentially - an appropriate succession of random choices was proposed and embedded in a category-theoretic setting. (The suggestion of applying a version of such random choice in the framework of processes arose in a discussion with Plotkin during an IFIP WG 2.2 meeting.)

The structure of the paper is the following. After this introduction we present in section 2 an outline of the underlying semantic framework, though without most of the mathematics. In denotational semantics, language constructs are provided with mathematical objects (functions, operators, etc.) as their meanings. In the present paper, these meanings are so-called processes (in our paper a technical term for certain mathematical objects rather than for -syntactic- components of a program). Processes are elements of domains in the general sense as introduced by SCOTT [21,22]. Technically, domains of processes are obtained as solutions of domain equations. The solution of such equations in a context with nondeterminacy and concurrency was first studied in detail by PLOTKIN [18] (see [4] for more recent references). We have based our solution techniques on completion methods in metric topology (as advocated recently by Nivat and his school, see e.g. [16]). Throughout our paper, we shall introduce a variety of processes, corresponding to a variety of programming concepts we encounter on the way to our understanding of the ADA rendez-vous. In section 2, processes are still simple. We call them uniform, and they bear a close resemblance to trees - though there are also a few crucial differences. Section 2 further introduces various operations upon processes - which will undergo successive refinements in later sections. We moreover illustrate uniform processes by using them in the semantics of a very simple language with parallel merge as its only concurrent notion. In section 3 we use uniform processes as a vehicle to explain the key idea of our approach to fairness, viz. suitable alternation of random choices. (Ultimately, this idea may be traced back to the use of oracles to handle fairness. Fundamental studies of the semantics of fairness were made by PARK [17]; proof-theoretic investigations are described, e.g., in [2,11,12,20].) Section 4 describes a number of ways of providing processes with additional structure. Firstly, we enrich them with a synchronization mechanism in the sense of MILNER's ports ([15]). We then obtain structures which are close to his synchronization trees. Next, we add a functional flavor to uniform processes, and obtain objects which have PLOTKIN's resumptions ([18]) as forerunners. Finally, we add a communication feature to processes yielding a counterpart for Milner's communication trees ([15]). Whereas in section 4 we introduce each extension independently, we need their combination in section 5 to define the semantics of a language with both parallel merge, (synchronization through) communication, and a version of Milner's restriction operator. This language is an abstraction of HOARE's CSP ([10]), and we use it to provide a translation of the ADA fragment featuring its rendez-vous concept ([1], chapter 9) in section 6. Section 7, finally, extends the