Formal Computations of Non Deterministic Recursive Program Schemes*

André Arnold and Maurice Nivat

Université de Lille I

Laboratoire d'Informatique Théorique et Programmation, Université Paris VII

Abstract. We extend to non deterministic recursive program schemes the methods and results which permit definition of the semantics of such schemes in the deterministic case. Under natural hypothesis the set of finite and infinite trees generated by a scheme is proved to be the greatest fixed point of the functional mapping usually attached to this scheme.

Introduction

Several authors have been recently interested in non deterministic recursive programs [1, 5, 11, 13, 14, 15, 16, 17].

These programs are obtained by introducing a connective or which has to be interpreted operationally as giving alternatives of a choice. As in the theory of deterministic recursive programs developed in [19], we shall consider non deterministic recursive program as interpretations of what we call non deterministic recursive programs schemes having the form \( \phi_i = \tau_i / 1 \leq i \leq k \) where each \( \tau_i \) is a term built up from constant function symbols, unknown function symbols \( \phi_i \) and the binary symbol or. It is clear that every non deterministic recursive program can be obtained from such schemes by interpreting the constant function symbols.

Our aim in this paper is to define the one-many function computed by a non deterministic recursive program scheme under the Herbrand interpretation \( H \), including all infinite computations of the program scheme (\( H \) is the interpretation which has as domain the set of finite and infinite trees and where the

* A first version of this paper was presented at the Foundations of Computer Science Symposium, in Poznań, in September 1977 [3].

1 Now at: Laboratoire d'Informatique, Université de Poitiers.
function $f_H$ interpreting the symbol $f$ of arity $n$ maps the $n$ trees $t_1, \ldots, t_n$ on the tree $f(t_1, \ldots, t_n)$. As shown in [21], the Herbrand interpretation has the same "initiality" property for non-deterministic as for deterministic recursive program schemes: the function computed under any interpretation $I$ is the image of the function computed under $H$ in the morphism induced by $I$.

It has been already noticed that non deterministic recursive program schemes look like context-free tree-grammars [1, 9, 11, 12], hence the results in this paper can be seen as generalizing those in [20] about context-free word-grammars.

Under the Herbrand interpretation a deterministic recursive program scheme naturally computes an infinite tree which is an element of the completion of the free magma considered as a partially ordered set. It is traditional to define the computed infinite tree as the least upper bound of some sequence or some directed set of finite approximations [8, 25].

In the present paper, to deal with nondeterminism we have to change the point of view by defining the computed infinite tree as the result of an infinite computation rather than the lub of the results of a collection of finite computations. This is equivalent to define the result of a successful infinite computation as an infinite intersection: it corresponds to the idea that the undefined element, usually denoted $\perp$ or $\Omega$, represents indeed the whole set of possible values. To be precise, when we replace an unknown function symbol in a non terminal expression by "undefined", this really means that the sub-expression headed by this function symbol can take all the values in the computation domain. Thus an expression is given a set of values: the larger this set the more undefined is the expression. An expression is defined (and the computation leading to it is successful) if and only if its set of values is a singleton.

This new point of view matches the intuitive idea that one has of a non deterministic program which computes a one-many function. At the beginning of the computation the value of this one-many function at a point can be any subset of the computation domain and in the course of the computation the set of possible values decreases.

Naturally enough, when one knows the relationship between infinite computations and greatest fixed points [6, 18, 24], the set of trees computed by a nondeterministic recursive program scheme is proved to be the greatest fixed point of some functional associated with the scheme, when the powerset of trees is simply ordered by inclusion. This is at least true for Greibach schemes, the condition of being Greibach playing a central role as soon as one deals with infinite computations [6, 20].

This paper contains four parts. The first one contains the main definitions and notations used in this paper. In the second part are defined the non deterministic recursive program schemes, the successful computations of them and their results. The third part establishes some relations between the set of results and the greatest fixed point of the functional $\hat{S}$ associated with the scheme $S$. The fourth one is devoted to Greibach schemes. For these Greibach schemes we prove not only equality between the set of results of $S$ and the greatest fixed point of $\hat{S}$ but also that this set is the least upper bound of a sequence of finite sets of finite trees in the Smyth preorder [26] which mixes inclusion and the usual order on the set of finite and infinite trees containing the "undefined" constant symbol $\Omega$. 