CLASSICAL AND INCREMENTAL EVALUATORS
FOR ATTRIBUTE GRAMMARS

by Gilberto Filé

Université de Bordeaux 1
U.E.R. de Mathématiques et d'Informatique
351, Cours de la Libération
33405 Talence - Cedex - France

Introduction

The study of attribute grammars (AG's) has mainly concentrated on the problem of efficiently performing attribute evaluation. One class of AG's is particularly interesting in this respect: the class of the absolutely noncircular attribute grammars (ANC-AG's) [7]. This class is interesting because quite efficient evaluators can be (automatically) constructed for any ANC-AG and, moreover, testing whether any AG is ANC takes polynomial time. Because of these reasons ANC-AG's have been used in the construction of compiler writing systems [5] and studied in many theoretical papers [6,7,3,9-4].

In this article we present a unifying definition of ANC-AG's that includes as special cases the characterizations of this class given in [4,7,9]. Classically [7], an AG is ANC if for every nonterminal X there exists a graph D(X) on its attributes having some special properties (see Section 1). Based on these graphs in [3,6] it is shown that recursive evaluators can be constructed easily for any ANC-AG. The evaluators of [3,6] are especially simple. Unfortunately, they may recompute some attributes several times. A second defect of these evaluators is that they "compute one synthesized attribute at the time". In [4] an improved model of evaluation is proposed: the new evaluator is still composed of a set of recursive procedures, but each procedure does not necessarily compute only one synthesized attribute. From this point of view the graph D(X) is transformed into a graph T(X) having blocks of attributes of X as nodes and where edges run from blocks of inherited attributes to blocks of synthesized attributes. Clearly the evaluators based on these graphs are, in general, subject to recompute attributes, as those based on the D(X)'s.

In [9] it is shown that for each nonterminal X of an ANC-AG one can always find a set A(X) of graphs having blocks of inherited or synthesized attributes of X as nodes (as T(X)), but such that for any 2 nodes there is a directed path from one of them to the other one. Hence, these graphs have, in addition to the arcs
of $T(X)$, also arcs running from blocks of synthesized attributes to blocks of inherited ones. Such graphs will be called totally ordered. From the $A(X)$'s it is easy to develop an evaluator that does never recompute attributes [9]. Unfortunately, such an evaluator risks to be very big: the number of graphs for each nonterminal may be exponential in the number of its attributes!

In this paper we show that an AG is ANC if to each nonterminal $X$ one can associate a set $\Pi A(X)$ of partially ordered graphs (in contrast to the totally ordered of $A(X)$,[9]). $\Pi A(X)$ lies between $T(X)$ an $A(X)$ in the following sense. $T(X)$ and $A(X)$ represent, respectively a minimal and a maximal level of the knowledge of the dependencies among the attributes of $X$ that is necessary for constructing an evaluator. $\Pi A(X)$ is any intermediate level of this knowledge. One can show that from any $\Pi A(X)$ one can "throw away" knowledge for constructing $T(X)$ and that one can "add knowledge" in order to construct an $A(X)$. Clearly, the evaluators that are constructed from the $\Pi A(X)$'s occupy such an intermediate place too and, in general, one should try to have in $\Pi A(X)$ the amount of knowledge that minimizes the recomputation and the size of the evaluator.

The ideas discussed above are also used for characterizing a class of AG's less known than ANC: the class of OPC-AG's that was introduced in [1] and that we prefer to call doubly noncircular AG's (DNC-AG). Using our characterization of DNC-AG we show how to construct (in a way essentially similar to that used for ANC-AG's) incremental evaluators for DNC-AG's. Again our approach allows the "tuning" of the evaluator in the attempt of finding a best compromise between recomputation and size.

The remainder of the article consists of 4 parts. In the first section we describe our approach, starting from the necessary definitions and arriving to the announced characterization of ANC- and DNC-AG's. Several examples are also given for clarifying the definitions. The 2nd and 3rd parts are dedicated to the development of the evaluators for ANC-AG's and DNC-AG's, respectively, and have an informal character. Finally, in the fourth part we point out some interesting problems that we intend to study in the future.

1 - The characterization of ANC- and DNC-AG's

The notion of attribute grammar (AG) is assumed to be known. Let us present below an AG that will be our running example throughout the paper.

Example 1 : of an AG.

In this article AG's are always considered from a "schematic" viewpoint i.e., one is not interested in the interpretation of the functions used for computing the attributes, but only in the dependency relation