WEIGHTED GRAPHS:
A TOOL FOR LOGIC PROGRAMMING

Philippe DEVRIENNE - EUDIL and LIFL (UA CNRS 369)
Patrick LEBEGUE - IUT A and LIFL (UA CNRS 369)

UNIVERSITY OF LILLE-I - Èt M3
59655 VILLENEUVE D'ASCQ Cédex FRANCE

ABSTRACT

Unfoldings of oriented graphs generate infinite trees that we generalize by weighting arrows of these graphs. Indexes along a branch are added during unfoldings and the result indexes variables. We study formal properties of these graphs (substitution, equivalence, unification,...). We use them to solve the halting problem of a recursive head-rewriting rule (as in PROLOG-like languages).

1. - INTRODUCTION

Usually, unification is recognized to act an important role in rewriting systems: for efficient unification algorithms, [4],[7],[12],[13] used tools such as directed acyclic graphs (dags), or oriented graphs (gos).

But the main problem is linked to the strategy of resolution, mainly in the case of recursive rules which may give rise to infinite rewritings, i.e. an infinite resolution.

In general term rewriting systems, this problem has been often studied [2][10] by introducing termination orderings. But these results are limited to particular rules (left-side linear, recursive path ordering,...). Our approach is quite different, in the case of head-rewriting systems (as in Prolog-like languages): We solve them for any single recursive rule (with or without occur-check), which can be written in Prolog in the following form:

\[ P(\beta) \rightarrow P(\tau) \]

where \( \beta \) and \( \tau \) are rational trees built on \( \Sigma \) and \( V \); and \( P \) is a predicate symbol, \( P \notin \Sigma \).

The resolution in forward or backward strategy can be infinite for some goals or facts \( P(\alpha) \) if the recursive rule can be infinitely applied.
To characterize this infinite resolution, we introduce a new kind of graphs.

2 - WEIGHTED GRAPHS

Unformally, we consider graphs and their unfoldings in infinite rational trees [7] (see fig. 1). A weighted graph (WG) is a graph with a top (as a tree) and paths which are weighted with relative integers (see fig. 2). In all the paper, we suppose that nodes are occurrences of letters of a finite graduate alphabet Σ in the usual sens [7] and, as in trees, we consider some leaves in a set V of variables; P, Q, ... can be interpreted in part 4 as predicate symbols.

A(G) denotes the infinite tree deduced from G by:
- infinite unfolding of loops
- attribution to each occurrence of x of its index i, which is the algebraic sum of the weights along the branch leading from the top down to x.

Then we note x_i (see fig. 3) and all the variable leaves of A(G) are of this type.

Furthermore, we associate to x a period j, j is a positive integer or is infinite. We note x(j) and x_i(j). If J = +∞, we omit it. x_i(j) can be interpreted as x_i modulo j.

graphs :

\[
\begin{array}{c}
\text{fig. 1} \\
\text{fig. 2} \\
\text{fig. 3}
\end{array}
\]