ABSTRACT: It is well known that confluence (which is equivalent to Church-Rosser property) is undecidable for arbitrary term rewriting systems. We prove here decidability of confluence for ground term rewriting systems. To obtain this result, we construct a special class of finite state tree transducers that we code in recognizable tree languages. Our work illustrates how tree language theory is useful in term rewriting systems study and we give easily some other results in the ground case (as decidability of uniform termination).

O. INTRODUCTION

Term (or tree) rewriting systems (t.r.s) are a very general and useful model of computation. Intuitively, a step of computation consists of rewriting in a context the right part of a rule in place of the left part.

The following two concepts are very important:

- a t.r.s is noetherian (or uniformly terminating) iff there is no infinite computation,

- a t.r.s is confluent iff, for any term, all sequences of rewrites emanating from it are extendable to a common term.

It is well known that, if a t.r.s is confluent and noetherian, it computes "normal form" of terms.

Unfortunately, neither confluence nor uniform termination are decidable. So, the motivation of many works is to obtain decision results for particular classes of t.r.s - as decidability of confluence for noetherian t.r.s (See for example Dershowitz [7], Guttag, Kapur, Musser [10], Huet & Oppen [12] for termination and Jouannaud [13], Huet [11] for confluence), Kozen [14], Raoult [17]).

In this paper, we study ground t.r.s. Ground terms are terms without variable (ground terms are also called words in algebra or abstract syntax trees in computer science). A t.r.s is a ground t.r.s iff each part of each rule is a ground term. Computations on stacks can be formalized by ground t.r.s (Buchi [5]). In expert sys-
tems, we can obtain a lot of ground inference rules.

Our main result is decidability of confluence for ground t.r.s Huet and Oppen stated this problem as open in their survey on equations and rewriting rules [12].

Furthermore, we think that the way we use here is as interesting as the goal. Our crucial tools are recognizable tree languages and finite state tree transducers (Doner [8], Engelfriet [9], Ogden & Rounds [16], Thatcher [20]). We think that this work illustrates an interesting connection between tree languages theory and t.r.s. (Brainers [4], Raoult [17]).

In the first part, we recall some definitions and notations. In part II, we introduce a new class of finite state tree transducers, that we call the class GTT of ground tree transducers. We give some properties of closure of the class and we remark the importance of recognizability for the description of these transformations.

In part III, we associate to every ground t.r.s. \( R \) a ground tree transducer \( T_R \) which computes the same transformation.

In part IV, we deduce decidability of confluence for ground t.r.s from decidability of inclusion in GTT. This last result comes from the ability of GTT to be coded in recognizable tree languages. Finally, we give some other simple results in part V.

Hint: Constructions are given without proof (see [6]), but the reader who is familiar with tree automata will be easily convinced. There is in Thatcher [19] a nice survey on tree automata and we use notation very near from Engelfriet's one [9] for tree automata and tree transducers.

I. DEFINITIONS AND NOTATIONS

In this paper, we always consider finite ranked alphabets.

Let from now on \( X \) be a fixed denumerable set of variables \( x_1, x_2, \ldots : X = \{ x_1, x_2, x_3, \ldots \} \).

\( T_\Sigma (X) \) denotes the set of trees on the ranked alphabet \( \Sigma \) indexed by variables of \( X \).

\( T_\Sigma^p (x_1, \ldots, x_q) \) (resp. \( T_\Sigma^q (x_1, \ldots, x_q) \)) denotes sequences of \( p \) trees indexed by \( \{ x_1, \ldots, x_q \} \). (resp. indexed by the ordered sequence \( x_1, \ldots, x_q \)).

We note sometimes \( \uparrow a \) uple of trees.