Simple Termination is Difficult

Aart Middeldorp
Institute of Information Sciences and Electronics
University of Tsukuba, Tsukuba 305, Japan
ami@softlab.is.tsukuba.ac.jp

Bernhard Gramlich
Fachbereich Informatik, Universität Kaiserslautern
Postfach 3049, D-6750 Kaiserslautern, Germany
gramlich@informatik.uni-kl.de

ABSTRACT

A terminating term rewriting system is called simply terminating if its termination can be shown by means of a simplification ordering, an ordering with the property that a term is always bigger than its proper subterms. Almost all methods for proving termination yield, when applicable, simple termination. We show that simple termination is an undecidable property, even for one-rule systems. This contradicts a result by Jouannaud and Kirchner. The proof is based on the ingenious construction of Dauchet who showed the undecidability of termination for one-rule systems.

1. Introduction

It is well-known that termination is an undecidable property of term rewriting systems. This result was obtained by Iluet and Lankford [8] in 1978. They showed that every Turing machine can be coded as a string rewriting system—a term rewriting system with only unary function symbols—such that termination of the resulting string rewriting system is equivalent to the uniform halting problem for the originating Turing machine. The number of rules in their construction depends on the number of Turing machine instructions. Later, Dershowitz [3] showed that every Turing machine can be simulated by means of a two-rule term rewriting system. This result was improved by Dauchet [2], who showed that termination remains undecidable even if we restrict our attention to one-rule term rewriting systems that are orthogonal and variable preserving. His skillful construction will be explained in detail later in this paper. On the other
hand, Caron [1] recently showed that termination is an undecidable property of length-preserving string rewriting systems—systems in which the left-hand side and the right-hand side of each rule have the same length—by a reduction to the uniform halting problem for linear bounded automata—a restricted kind of Turing machines.

From this last result one easily obtains the undecidability of simple termination for the same class of term rewriting systems. Simple termination is a stronger notion than termination. A term rewriting system is simply terminating if the addition of all rewrite rules of the form \( f(x_1, \ldots, x_n) \rightarrow x_i \) results in a terminating system. Virtually all methods for proving termination yield, when applicable, simple termination. Simple termination is closely related to the non-self-embedding property, since every simply terminating term rewriting system is non-self-embedding. Plaisted [15] showed that the non-self-embedding property is undecidable. From this result we cannot infer the undecidability of simple termination, however. As a matter of fact, it is known that negative results for the class of non-self-embedding systems do not always carry over to the class of simply terminating systems, see [6]. In this paper we show the undecidability of simple termination for one-rule term rewriting systems. This contradicts a result of Jouannaud and Kirchner [9]. The undecidability proof is based on the ingenious construction of Dauchet. He showed in [2] that with every Turing machine \( M \) one can associate a term rewriting system \( R_M \) consisting of a single rewrite rule such that

\[
M \text{ halts for all configurations } \iff R_M \text{ is terminating.}
\]

From this we cannot immediately infer the undecidability of simple termination for one-rule systems, since the implication “\( R_M \) is terminating \( \Rightarrow \) \( R_M \) is simply terminating” does not hold for every Turing machine \( M \). However, we will show that if we start the construction of Dauchet from a linear bounded automaton \( M \) instead of a Turing machine, termination and simple termination of \( R_M \) coincide.

The paper is organized as follows. The next section contains a brief introduction to term rewriting, including a discussion of the property simple termination. In Section 3 we define linear bounded automata. Section 4 describes Dauchet’s construction. Actually, we present a somewhat simpler construction. We show that the equivalence

\[
M \text{ halts for all configurations } \iff R_M \text{ is terminating}
\]

is easily obtained for all linear bounded automata \( M \) by using a recent result of Zantema [16] on type removal. In Section 5 we show that the equivalence