Title	消去法による項書換え系の停止性
Author(s)	中村，正樹
Citation	
Issue Date	1999-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1249
Rights	
Description	Supervisor:外山 芳人，情報科学研究科，修士

On Proving Termination by Eliminations

Masaki NAKAMURA

School of Information Science,
Japan Advanced Institute of Science and Technology

February 15, 1999

Keywords: Term Rewriting System, Termination, Transformation, Elimination, Dependency Pair .

This thesis is about termination of Term Rewriting Systems (TRSs). The purpose of this study is to improve the elimination methods, which are transformation methods for simplifying the task of proving termination of TRSs. We have known that the elimination methods are sound with respect to termination, i.e., if the transformed TRS terminates, so does the original one. The elimination methods can be quite useful in the field of automatic termination proofs, since they can easily be implemented and used to pre-process the TRSs to be proved terminating. In this thesis, we extend the elimination methods by removing rules that are not necessary for soundness from transformed TRSs. Using the improved elimination methods, we can transform TRSs that can not be applied by original methods into terminating ones.

1 background

Term Rewriting systems provide a simple formalism useful for the study of computational procedures. For equational reasoning, TRSs are applicable in various fields, for example, algebraic specification, functional programming languages and automated theorem proving. A TRS is a set of oriented equations that we call rules describing some relation between terms. In order to obtain a reduction from a term, we have to identify a part of it that matches the left-hand side of some rewrite rule. Then we can replace the matched part of the term with the right-hand side of the rule matched.

A TRS is said to be terminating if no term admits an infinite reduction sequence. Termination is an important property of TRSs. For a terminating TRS, a normal form of a given term can be found by a simple depth-first search. Unfortunately, termination is an undecidable property of TRSs; nevertheless, there are some techniques that are successful in particular cases. Those techniques can be divided into two main groups: syntactical methods and semantical methods. In the first class, only the syntactical structure of the terms is used to prove termination. The well-known syntactical method is recursive path order (rpo) by Dershowiz(1982). Given a finite TRS, it is decidable if termination of the TRS can be proved with rpo. However, these TRSs are restricted to simplification orderings in which any term is greater than its strict subterms. In the second class, terms are interpreted in some algebra in order to prove termination. For any terminated TRS, some algebra guarantees its termination.

There are many TRSs whose termination is difficult to prove with current methods like rpo. One of the approaches to proving termination of such TRSs is a method for transforming TRSs such that the transformed systems are easier to deal with than the original ones. The elimination methods are the transformation methods in which function symbols considered 'useless' are eliminated to simplify the rewrite rules.

Suppose we want to prove termination of the following system R_1 of which we can not prove termination with rpo.

$$R_1 = \{f(f(x)) \to f(g(f(x)))\}$$

Intuitively termination of this system is not difficult: at every step, the number of nested operation symbols f decreases. By dummy elimination proposed by Ferreira and Zantema(1995), we can eliminate a function symbol g and transform the TRS R_1 into

$$\mathcal{E}(R_1) = \left\{ \begin{array}{lcl} f(f(x)) & \to & f(\diamond) \\ f(f(x)) & \to & f(x) \end{array} \right.$$

where $\diamond$ is a flesh constant. Termination of the TRS $\mathcal{E}(R_1)$ is proved with rpo. In the definition of dummy elimination, terms whose root symbol is the one to be eliminated are replaced by a fresh constant $\diamond$ and its subterms are treated as separate entities that add the right-hand side of rules. The TRS R_1 terminates since dummy elimination is sound transformation with respect to termination. Using a pre-process of the TRSs to be proved terminating, we can get the procedure for automatic termination proofs, which can apply to TRSs of which termination can not be proved directly.

Another approach is the notion of dependency pairs by Arts and Giesl(1997), which gives an effective method for analyzing an infinite reduction sequence. Given a TRS, defined symbols of the TRS are defined as symbols in the root positions of the left-hand side of its rules. Since every left-hand side has of course a defined symbol as its root symbol, no rule matches a term without defined symbols. The subterms of the right-hand sides of which the root symbol is a defined symbol are of importance for the analysis of infinite reduction sequence. The notion of dependency pairs focuses on the subterms of the right-hand sides that have a defined symbol as the root symbol. By regarding a sequence of these dependency pairs, the occurrences of defined symbols can be traced.

2 our results

In this thesis, we extend the sphere of the application of dummy elimination.

To put it concretely, we direct our attention to the occurrences of defined symbols in transformed rules and remove rules that do not affect soundness. The following is an example of the TRS that can not be transformed into terminated one by dummy elimination.

$$R_2 = \left\{ \begin{array}{rcl} f(a) & \rightarrow & f(b) \\ b & \rightarrow & g(a) \end{array} \right.$$

We can eliminate the function symbol g and transform the TRS R_1 into

$$\mathcal{E}(R_2) = \left\{ \begin{array}{rcl} f(a) & \rightarrow & f(b) \\ b & \rightarrow & \diamond \\ b & \rightarrow & a. \end{array} \right.$$

Clearly, the TRS $\mathcal{E}(R_2)$ does not terminate because of the rule $b \rightarrow a$ added to the transformed TRS. The rule $b \rightarrow a$ is what to preserve the subterm a under the eliminated function symbol g. In the dummy elimination, we have to preserve the subterms under the eliminated function symbols as the right-hand sides of transformed rules. We propose that it is not necessary to add the subterms in which the defined symbols do not occur. By our extended dummy elimination, we transform the TRS R_2 into the following TRS $\mathcal{E}'(R)$ without the rule $b \rightarrow a$.

$$\mathcal{E}'(R_2) = \left\{ \begin{array}{rcl} f(a) & \rightarrow & f(b) \\ b & \rightarrow & \diamond \end{array} \right.$$

It is easy to prove termination of TRS $\mathcal{E}'(R)$ with rpo. Hence, we can get the procedure that can apply to more TRSs, including R_2, than the current procedures induced from original dummy elimination for automatic termination proofs.

In the proof of soundness of the extended dummy elimination, a weak reduction order plays an important role. We introduce the argument filtering method to make a weak reduction order and generally show the inclusion relation between a given TRS and a transformed TRS in order to ensure soundness.

Furthermore, we introduce the other elimination methods, the distribution elimination by Zantema(1994), and the general dummy elimination by Ferreira(1995). We can improve their methods in a similar way and prove their soundness with argument filtering method.