Available online at www.sciencedirect.com
SCIENCE DIRECT®
Information and Computation 199 (2005) 172–199
Information and Computation
www.elsevier.com/locate/ic

Automating the dependency pair method

Nao Hirokawa, Aart Middeldorp*

Institute of Computer Science, University of Innsbruck, 6020 Innsbruck, Austria

Received 28 November 2003; revised 16 July 2004

Abstract

Developing automatable methods for proving termination of term rewrite systems that resist traditional techniques based on simplification orders has become an active research area in the past few years. The dependency pair method of Arts and Giesl is one of the most popular such methods. However, there are several obstacles that hamper its automation. In this paper we present new ideas to overcome these obstacles. We provide ample numerical data supporting our ideas.

Keywords: Term rewriting; Termination; Dependency pair method

1. Introduction

Proving termination of term rewrite systems has been an active research area for several decades. In recent years the emphasis has shifted towards the development of powerful methods for automatically proving termination. The traditional methods for automated termination proofs of rewrite systems are simplification orders like the recursive path order, the Knuth–Bendix order, and (most) polynomial orders. The termination proving power of these methods has been significantly extended by the *dependency pair method* of Arts and Giesl [3]. In this method, depicted in

* Corresponding author.
E-mail addresses: nao.hirokawa@uibk.ac.at (N. Hirokawa), aart.middeldorp@uibk.ac.at (A. Middeldorp).

0890-5401/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.ic.2004.10.004

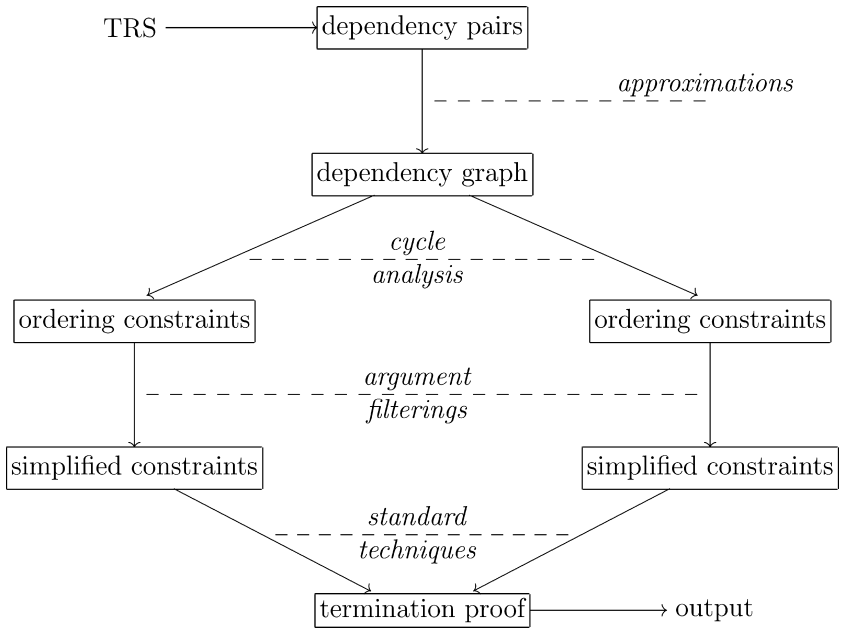

Fig. 1. The dependency pair method.

Fig. 1, a rewrite system is transformed into groups of ordering constraints such that termination of the system is equivalent to the (separate) solvability of these groups. The number and size of these groups is determined by the approximation used to estimate the dependency graph and, more importantly, by the cycle analysis algorithm that is used to extract the groups from the approximated dependency graph. Typically, the ordering constraints in the obtained groups must be simplified before traditional simplification orders like the recursive path order or the Knuth–Bendix order are applicable. Such simplifications are performed by so-called argument filterings. It is fair to say that the dependency pair method derives much of its power from the ability to use argument filterings to simplify constraints. The finiteness of the argument filtering search space has been stressed in many papers on the dependency pair method, but we do not hesitate to label the enormous size of this search space as the main obstacle for the successful automation of the dependency pair method when using strongly monotone simplification orders.

The dependency pair method can also be used for automatically proving *innermost* termination. Innermost termination is easier to prove than termination for the following two reasons: (1) the innermost dependency graph is typically much smaller than the dependency graph and (2) each group of ordering constraints for innermost termination is (often strictly) contained in a group of ordering constraints for termination.

We present several new ideas which help to tackle the argument filtering problem in Section 5. In Section 4, we present a new algorithm for cycle analysis and in Section 3 we present new approximations of the (innermost) dependency graph. A brief introduction to the dependency pair method is given in the next section. In Section 6, we report on the numerous experiments that we performed to assess the viability of our ideas.