Goal-Directed and Relative Dependency Pairs for Proving the Termination of Narrowing*

José Iborra¹, Naoki Nishida², and Germán Vidal¹

¹ DSIC, Universidad Politécnica de Valencia, Spain
{jiborra,gvidal}@dsic.upv.es
² Graduate School of Information Science, Nagoya University, Nagoya, Japan
nishida@is.nagoya-u.ac.jp

Abstract. In this work, we first consider a goal-oriented extension of the dependency pair framework for proving termination w.r.t. a given set of initial terms. Then, we introduce a new result for proving relative termination in terms of a dependency pair problem. Both contributions put together allow us to define a simple and powerful approach to analyzing the termination of narrowing, an extension of rewriting that replaces matching with unification in order to deal with logic variables. Our approach could also be useful in other contexts where considering termination w.r.t. a given set of terms is also natural (e.g., proving the termination of functional programs).

1 Introduction

Proving that a program terminates is a fundamental problem that has been extensively studied in almost all programming paradigms. In term rewriting, where termination analysis has attracted considerable attention (see, e.g., the surveys of Dershowitz [8] and Steinbach [23]), the termination of a rewrite system is usually proved for all possible reduction sequences.

In some cases, however, one is only interested in those sequences that start from a distinguished set of terms. This case has been already considered in some previous works, e.g., for proving the termination of logic programs [20], for proving the termination of Haskell programs [10], and for proving the termination of narrowing [25], an extension of rewriting to deal with logic variables. Unfortunately, these works do not focus on proving termination from an initial set of terms—only consider this problem to some extent—and are difficult to generalize.

In this paper, we first extend the well-known dependency pair framework [3,11] for proving the termination of rewriting in order to only consider derivations.

* This work has been partially supported by the Spanish Ministerio de Ciencia e Innovación under grant TIN2008-06622-C03-02, by Generalitat Valenciana under grants ACOMP/2009/017 and GV/2009/024, and by UPV (programs PAID-05-08 and PAID-06-08). The second author has been partially supported by a grant from Nagoya Industrial Science Research Institute.
from a given initial set of terms. The fundamental improvements are twofold: firstly, we introduce a notion of chain which considers only reachable loops, thus reducing the number of pairs to consider; secondly, we also present a notion of usable rules that regards as usable only those rules which occur in the derivation from an initial term, allowing us to reduce the number of rules.

As a second contribution of this paper, we study a direct application of the dependency pair approach to the solution of relative termination problems when the involved TRSs form a hierarchical combination. Roughly speaking, we can study whether $R$ terminates relative to $B$ (i.e., whether all $\rightarrow_R \cup \rightarrow_B$ reductions contain only finitely many $\rightarrow_R$ steps) in those cases where $B$ does not make calls to functions defined in $R$. Although this application is arguably folklore in the literature (see, e.g., the work of \cite{24}), to our knowledge this is the first time that the necessary conditions have been ascertained and proved in a formal publication.

Finally, we illustrate the usefulness of our developments by applying them to proving the termination of narrowing starting from initial terms. Our results are more general and potentially more accurate than previous approaches (e.g., \cite{25}). Moreover, our approach could also be useful in other contexts where considering termination w.r.t. a given set of terms is also a natural requirement, like the approach to proving the termination of Haskell programs of \cite{10} or that to proving the termination of logic programs by translating them to rewrite systems of \cite{20}.

The paper is organized as follows. After introducing some preliminaries in the next section, we present the goal-directed dependency pair framework in Section 3. Then, Section 4 first states a useful result for proving the relative termination of a rewrite system and, then, presents a new approach for proving the termination of narrowing. Finally, Section 5 reports on the implementation of a termination prover based on the ideas of this paper and concludes. An extended version including proofs of technical results can be found in \cite{15}.

2 Preliminaries

We assume familiarity with basic concepts of term rewriting and narrowing. We refer the reader to, e.g., \cite{4} and \cite{13} for further details.

**Terms and Substitutions.** A *signature* $\mathcal{F}$ is a set of function symbols. We often write $f/n \in \mathcal{F}$ to denote that the arity of function $f$ is $n$. Given a set of variables $\mathcal{V}$ with $\mathcal{F} \cap \mathcal{V} = \emptyset$, we denote the domain of terms by $\mathcal{T}(\mathcal{F}, \mathcal{V})$. We assume that $\mathcal{F}$ always contains at least one constant $f/0$. We use $f, g, \ldots$ to denote functions and $x, y, \ldots$ to denote variables. A *position* $p$ in a term $t$ is represented by a finite sequence of natural numbers, where $\epsilon$ denotes the root position. Positions are used to address the nodes of a term viewed as a tree. The root symbol of a term $t$ is denoted by root$(t)$. We let $t|_p$ denote the *subterm* of $t$ at position $p$ and $t[s]_p$ the result of replacing the subterm $t|_p$ by the term $s$. $\text{Var}(t)$ denotes the set of variables appearing in $t$. A term $t$ is *ground* if $\text{Var}(t) = \emptyset$. We write $\mathcal{T}(\mathcal{F})$ as a shorthand for the set of ground terms $\mathcal{T}(\mathcal{F}, \emptyset)$. 