Available online at www.sciencedirect.com
SCIENCE DIRECT®

Electronic Notes in Theoretical Computer Science

ELSEVIER Electronic Notes in Theoretical Computer Science 124 (2005) 51–63
www.elsevier.com/locate/entcs

Some Undecidable Approximations of TRSs

Jeroen Ketema

Department of Computer Science
Faculty of Sciences, Vrije Universiteit Amsterdam
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

Abstract

In this paper we study the decidability of reachability, normalisation, and neededness in n-shallow and n-growing TRSs. In an n-growing TRS, a variable that occurs both on the left- and right-hand side of a rewrite rule must be at depth n on the left-hand side and at depth greater than n on the right-hand side. In an n-shallow TRS, a variable that occurs both on the left- and right-hand side of a rewrite rule must be at depth n on both sides.
The n-growing and n-shallow TRSs are generalisations of the growing and shallow TRSs as introduced by Jacquemard and Comon. For both shallow and growing TRSs reachability, normalisation, and (in the orthogonal case) neededness are decidable. However, as we show, these results do not generalise to n-growing and n-shallow TRSs. Consequently, no algorithm exists that performs a needed reduction strategy in n-growing or n-shallow TRSs.

Keywords: Approximations, undecidability, reachability, normalisation, neededness.

1 Introduction

As is well-known, given an arbitrary term rewriting system (TRS), the following questions are undecidable [10].

- Reachability: is a term reachable from another term?
- Normalisation: does a term have a normal form?
- Neededness: is a redex in a term needed?

However, for some classes of TRSs these properties are decidable. These classes are often used as approximations. That is, let $\mathcal{R}$ and $\mathcal{S}$ be TRSs over the same signature, then $\mathcal{S}$ is an *approximation* of $\mathcal{R}$ if $\to_{\mathcal{R}}^{*} \subseteq \to_{\mathcal{S}}^{*}$ and $\mathsf{NF}_{\mathcal{R}} = \mathsf{NF}_{\mathcal{S}}$. Here, $\to_{\mathcal{R}}^{*}$ and $\to_{\mathcal{S}}^{*}$ denote the transitive reflexive closures of the

doi:10.1016/j.entcs.2004.11.024

rewrite relations of $\mathcal{R}$ and $\mathcal{S}$, and $\mathsf{NF}_{\mathcal{R}}$ and $\mathsf{NF}_{\mathcal{S}}$ denote the sets of normal forms of $\mathcal{R}$ and $\mathcal{S}$.

For most of the classes in which reachability, normalisation, and neededness are decidable, the forms of the rewrite rules are restricted [3,1,4,9]. Moreover, given the decidability of neededness there exists for orthogonal TRSs an algorithm that performs a needed reduction strategy [3,2,1,4].

To explore the boundaries of the decidability of reachability, normalisation, and neededness we introduce in this paper n-growing and n-shallow TRSs. These TRSs are generalisations of the growing and shallow TRSs as introduced respectively by Jacquemard [4] and Comon [1]. Although reachability, normalisation, and (in the orthogonal case) neededness are decidable for growing and shallow TRSs we show that this does not hold for our generalisations.

The n-growing and n-shallow TRSs are closely related to four other classes of TRSs for which it is known that reachability, normalisation, and neededness are undecidable [4,5,6]. We show that n-growing and n-shallow TRSs are different from those classes except in one instance.

We proceed as follows. In Sect. 2 we give some preliminary definitions. Then, in Sect. 3 we define two variants of Post's Correspondence Problem (PCP). We encode these variants as TRSs in Sect. 4 and Sect. 5 to show that reachability, normalisation, and neededness are undecidable for n-growing and n-shallow TRSs. In Sect. 6 we compare the n-growing and n-shallow TRSs to the other four classes of TRSs for which reachability, normalisation, and neededness are undecidable. Finally, in Sect. 7 we give some directions for further research.

2 Preliminaries

Throughout this paper we assume $\mathbb{N}$ is the set of non-negative integers. we denote the disjoint union of the sets U and V by $U \uplus V$.

By Γ we denote an arbitrary alphabet. Here, Γ^* and Γ^+ denote the sets of finite strings and finite non-empty strings over Γ, ϵ denotes the empty string, and if $s \in \Gamma^*$, then $|s|$ denotes the length of s.

If $s, t \in \Gamma^*$, then $s \cdot t$ denotes the concatenation of s and t. The empty string ϵ is the neutral element with respect to concatenation. If $a \in \Gamma$ and $n \in \mathbb{N}$, then $a^0 = \epsilon$ and $a^{n+1} = a \cdot a^n$.

By $\mathcal{T}er(\Sigma, X)$ we denote the set of terms over the signature Σ and the set of variables X. If $t \in \mathcal{T}er(\Sigma, X)$, then $\mathcal{V}ar(t)$ denotes the set of variables that occur in t. We call t *linear* when each variable occurs at most once in t. Moreover, we confuse signatures consisting only of unary function symbols and alphabets. Hence, given a unary function symbol f and an $n \in \mathbb{N}$ we have