

# 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.

*Key words:* 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  $\rightarrow_{\mathcal{R}}^* \subseteq \rightarrow_{\mathcal{S}}^*$  and  $\text{NF}_{\mathcal{R}} = \text{NF}_{\mathcal{S}}$ . Here,  $\rightarrow_{\mathcal{R}}^*$  and  $\rightarrow_{\mathcal{S}}^*$  denote the transitive reflexive closures of the rewrite relations of  $\mathcal{R}$  and  $\mathcal{S}$ , and  $\text{NF}_{\mathcal{R}}$  and  $\text{NF}_{\mathcal{S}}$  denote the sets of normal forms of  $\mathcal{R}$  and  $\mathcal{S}$ .

*This is a preliminary version. The final version will be published in  
Electronic Notes in Theoretical Computer Science  
URL: [www.elsevier.nl/locate/entcs](http://www.elsevier.nl/locate/entcs)*

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. Moreover, by  $U \uplus V$  we denote the disjoint union of the sets  $U$  and  $V$ .

By  $\Gamma$  we denote an arbitrary alphabet. Here,  $\Gamma^*$  and  $\Gamma^+$  denote the sets of finite strings and finite non-empty strings over  $\Gamma$ ,  $\epsilon$  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  $\epsilon$  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  $\Sigma$  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  $f^0(x) = x$  and  $f^{n+1}(x) = f(f^n(x))$ .

We denote the set of positions of a term  $t \in \mathcal{T}er(\Sigma, X)$  by  $\mathcal{P}os(t) \subseteq \mathbb{N}^*$ . The *depth* of a subterm at  $p \in \mathcal{P}os(t)$  is  $|p|$ . If  $t$  is linear, then each  $x \in \mathcal{V}ar(t)$  has a unique depth, which we denote  $d_t(x)$ .

By  $\mathcal{R} = (\Sigma, R)$  we denote a term rewriting system (TRS) with the signature  $\Sigma$  and the set of rewrite rules  $R$ . The elements of  $R$  are denoted

$l \rightarrow r$ . As usual in the study of approximations we only require  $l \notin X$ , not  $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$ . The transitive reflexive closure of  $\rightarrow$  is denoted by  $\rightarrow^*$ . A rule  $l \rightarrow r$  is called *linear* if  $l$  and  $r$  are linear.

Let  $\mathcal{R} = (\Sigma, R)$  be a TRS and  $s, t \in \mathcal{T}er(\Sigma, X)$ . The term  $t$  is *reachable* from  $s$  if  $s \rightarrow^* t$ . Moreover,  $s$  *normalises* if  $s$  has a normal form. A redex in  $s$  is *needed* iff when  $s$  has a normal form, a descendant of the redex is eliminated by contraction of an overlapping redex in every reduction from  $s$  to normal form [10, Definition 9.2.1]. A *needed reduction strategy* eliminates in every term a needed redex.

### 3 Two Variants of Post's Correspondence Problem

In this section we introduce Post's Correspondence Problem (PCP) and two variants of that problem. In the definitions we use the following notation.

**Definition 3.1** Let  $s \in \Gamma^+$ . Given an  $n \in \mathbb{N}$ , define  $s^{[n]}$  by

- $s^{[n]} = a^n$  when  $s = a$  with  $a \in \Gamma$ , and
- $s^{[n]} = a^n \cdot t^{[n]}$  when  $s = a \cdot t$  with  $a \in \Gamma$  and  $t \in \Gamma^+$ .

Note that this is not exponentiation, which is defined as  $s^0 = \epsilon$  and  $s^{n+1} = s \cdot s^n$ . Assuming  $\Gamma = \{a, b\}$ , we have  $(ab)^{[2]} = aabb$ , while  $(ab)^2 = abab$ .

We next define three kinds of pairs, as used in the definition of PCP and its two variants.

**Definition 3.2** Let  $u, v \in \Gamma^+$  and  $n$  a natural number.

- The pair  $(u, v)$  is called a *PCP pair*.
- The pair  $(u^{[n]}, v^{[n]})$  is called an  *$n$ -PCP pair*.
- The pair  $(u^{[n]} \cdot e^{[kn]}, v^{[n]} \cdot e^{[ln]})$  is called a *padded  $n$ -PCP pair* when
  - $k = \max\{|u|, |v|\} - |u|$ ,
  - $l = \max\{|u|, |v|\} - |v|$ , and
  - $e \notin \Gamma$ .

The intuition behind a padded  $n$ -PCP pair is that it is an  $n$ -PCP pair in which the shortest string is padded with  $e$  symbols. This gives both strings in the pair the same length. For any padded  $n$ -PCP pair  $(u^{[n]} \cdot e^{[kn]}, v^{[n]} \cdot e^{[ln]})$

$$n(|u| + k) = |u^{[n]} \cdot e^{[kn]}| = |v^{[n]} \cdot e^{[ln]}| = n(|v| + l)$$

and, denoting the substitution of  $\epsilon$  for  $e$  by  $[e := \epsilon]$ ,

$$(u^{[n]} \cdot e^{[kn]}, v^{[n]} \cdot e^{[ln]})[e := \epsilon] = (u^{[n]}, v^{[n]}),$$

which is an  $n$ -PCP pair. Thus, we can consider  $e$  to be a placeholder for the empty string.

We now define PCP and its two variants.

**Problem 3.3 (PCP)** *Let  $P$  be a finite set of PCP pairs. Does there exist an  $m \geq 1$  such that  $u_1 \cdot \dots \cdot u_m = v_1 \cdot \dots \cdot v_m$  with  $(u_i, v_i) \in P$  for all  $1 \leq i \leq m$ ?*

**Problem 3.4 (*n*-PCP)** *Let  $P$  be a finite set of  $n$ -PCP pairs. Does there exist an  $m \geq 1$  such that  $u_1 \cdot \dots \cdot u_m = v_1 \cdot \dots \cdot v_m$  with  $(u_i, v_i) \in P$  for all  $1 \leq i \leq m$ ?*

**Problem 3.5 (Padded *n*-PCP)** *Let  $e \notin \Gamma$  and let  $P$  be a finite set of padded  $n$ -PCP pairs. Does there exist an  $m \geq 1$  such that  $(u_1 \cdot \dots \cdot u_m)[e := \epsilon] = (v_1 \cdot \dots \cdot v_m)[e := \epsilon]$  with  $(u_i, v_i) \in P$  for all  $1 \leq i \leq m$ ?*

PCP and its two variants can be transformed into each other, as there exists for each kind of pair a “related” pair of each of the other kinds. For example, assuming again  $\Gamma = \{a, b\}$ , we have for the PCP pair  $(a, ab)$  that  $(a^{[n]}, (ab)^{[n]})$  is the “related”  $n$ -PCP pair and that  $((ae)^{[n]}, (ab)^{[n]})$  is the “related” padded  $n$ -PCP pair. This leads to the following theorem.

**Theorem 3.6** *PCP,  $n$ -PCP, and padded  $n$ -PCP are reducible to each other for  $n \geq 1$ .*

**Proof** Using the previously described “related” pairs, this follows directly from the definitions of PCP,  $n$ -PCP and padded  $n$ -PCP.  $\square$

By the previous theorem and the undecidability of PCP [8], we have the following.

**Corollary 3.7** *The  $n$ -PCP and padded  $n$ -PCP questions are undecidable for  $n \geq 1$ .*

For padded  $n$ -PCP note that if  $(u_1 \cdot \dots \cdot u_m)[e := \epsilon] = (v_1 \cdot \dots \cdot v_m)[e := \epsilon]$ , then the number of  $e$  occurrences must be equal in  $u_1 \cdot \dots \cdot u_m$  and  $v_1 \cdot \dots \cdot v_m$ . If not, the lengths of  $(u_1 \cdot \dots \cdot u_m)[e := \epsilon]$  and  $(v_1 \cdot \dots \cdot v_m)[e := \epsilon]$  are different which contradicts equality.

Using the previous fact and assuming a string rewrite system (SRS) with for all  $a \in \Gamma$  and  $e \notin \Gamma$  the rewrite rules  $a \cdot e \rightarrow e \cdot a$  and  $e \cdot a \rightarrow a \cdot e$ , we can rephrase padded  $n$ -PCP.

**Problem 3.8 (Padded *n*-PCP)** *Let  $\Delta = \Gamma \uplus \{e\}$  and let  $P$  be a finite set of padded  $n$ -PCP pairs. Does there exist an  $m \geq 1$  and an  $s \in \Delta^+$  such that  $u_1 \cdot \dots \cdot u_m \rightarrow^* s$  and  $v_1 \cdot \dots \cdot v_m \rightarrow^* s$  with  $(u_i, v_i) \in P$  for all  $1 \leq i \leq m$ ?*

## 4 Undecidable $n$ -Growing Term Rewriting Systems

In this section we describe our first class of TRSs for which reachability, normalisation, and neededness are undecidable. The class is defined as follows.

**Definition 4.1** Let  $l \rightarrow r$  be a rewrite rule. The rule is *n-growing* when it is linear and when for all  $x \in \mathcal{V}ar(l) \cap \mathcal{V}ar(r)$  it holds that  $d_l(x) = n$  and  $d_r(x) > n$ . A TRS is *n-growing* when all its rewrite rules are *n-growing*.

Observe that in  $n$ -growing TRSs we restrict the forms of the rewrite rules. Moreover, observe that  $n$ -growing rewrite rules and TRSs are closely related to

the following rewrite rules and TRSs, as defined by Jacquemard [4, Definition 4].

**Definition 4.2** Let  $l \rightarrow r$  be a rewrite rule. The rule is *growing* when it is linear and when for all  $x \in \mathcal{V}ar(l) \cap \mathcal{V}ar(r)$  it holds that  $d_l(x) = 1$  and  $d_r(x) \in \mathbb{N}$ . A TRS is *growing* when all its rewrite rules are growing.

Obviously, for  $n = 1$  the  $n$ -growing TRSs form a sub-class of the growing TRSs. However, for  $n > 1$  this is not the case. We have, for example, the  $n$ -growing rewrite rule

$$f^n(x) \rightarrow f^{n+1}(x).$$

For  $n > 1$  this rewrite rule is not growing, as  $d_{f^n(x)}(x) = n > 1$ .

The growing TRSs do not form a sub-class of the  $n$ -growing TRSs for any  $n$ . We have, for example, the growing rewrite rule

$$f(x) \rightarrow x.$$

This rewrite rule is not  $n$ -growing, as  $d_x(x) = 0$ .

Using tree automata techniques, Jacquemard [4] proves that reachability and normalisation are decidable for growing TRSs. Durand and Middeldorp [2] prove that neededness is decidable for orthogonal growing TRSs. As each 1-growing TRS is growing, we also have decidability of reachability, normalisation, and (in the orthogonal case) neededness for 1-growing TRSs. However, as we show next, these results do not generalise to  $n$ -growing TRSs with  $n > 1$ .

**Theorem 4.3** *Let  $n \geq 1$ . Reachability is undecidable for  $n+1$ -growing TRSs.*

**Proof** We reduce  $n$ -PCP to reachability in an  $n+1$ -growing TRS. Suppose we have a finite set  $P$  of  $n$ -PCP pairs. Define the signature  $\Sigma = \Gamma \uplus \{c, d, f, g, h\}$ . The arities are as follows

- $c$  and  $d$  are constants,
- $g, h$  and the elements of  $\Gamma$  have arity 1, and
- $f$  has arity 2.

Define for all  $(u, v) \in P$  and  $a \in \Gamma$  the following rewrite rules

- (1)  $c \rightarrow f(g^n(u(d)), g^n(v(d)))$
- (2)  $f(g^n(x), g^n(y)) \rightarrow f(g^n(u(x)), g^n(v(y)))$
- (3)  $f(g^n(x), g^n(y)) \rightarrow h^{n+1}(f(x, y))$
- (4)  $f(a^n(x), a^n(y)) \rightarrow h^{n+1}(f(x, y))$
- (5)  $f(d, d) \rightarrow d$
- (6)  $h^{n+1}(d) \rightarrow d$

As is easy to see, we have a finite number of  $n + 1$ -growing rewrite rules. Hence, the rewrite rules form an  $n + 1$ -growing TRS. By  $n \geq 1$ , we have for the TRS that  $d$  is reachable from  $c$  if and only if  $n$ -PCP has a solution for  $P$ .

To see that  $d$  is reachable from  $c$  when  $n$ -PCP has a solution for  $P$ , suppose that  $u_1 \dots u_m = v_1 \dots v_m$  is a solution. We can now construct the reduction

sequence

$$\begin{aligned} c &\rightarrow_{(1)} f(g^n(u_m(d)), g^n(v_m(d))) \\ &\rightarrow_{(2)}^* f(g^n(u_1 \cdot \dots \cdot u_m(d)), g^n(v_1 \cdot \dots \cdot v_m(d))) \\ &\rightarrow_{(3)} h^{n+1}(f(u_1 \cdot \dots \cdot u_m(d), v_1 \cdot \dots \cdot v_m(d))) \end{aligned}$$

As  $u_1 \cdot \dots \cdot u_m = v_1 \cdot \dots \cdot v_m$ , we can, for some  $k$ , extend the reduction sequence to

$$\begin{aligned} c &\rightarrow^* h^{n+1}(f(u_1 \cdot \dots \cdot u_m(d), v_1 \cdot \dots \cdot v_m(d))) \\ &\rightarrow_{(4)}^* h^{k(n+1)}(f(d, d)) \\ &\rightarrow_{(5)} h^{k(n+1)}(d) \\ &\rightarrow_{(6)}^* d \end{aligned}$$

Hence,  $d$  is reachable from  $c$ .

To see that  $n$ -PCP has a solution for  $P$  when  $d$  is reachable from  $c$ , note that the only way to reduce  $c$  to  $d$  is to first perform an (1)-step and a number of (2)-steps, then to perform a (3)-step and a number of (4)-steps, and to finally perform a (5)-step and a number of (6)-steps. Also note that the reduction sequence only ends in  $d$  when the (1)-step and the (2)-steps give us a term  $f(g^n(u_1 \cdot \dots \cdot u_m(d)), g^n(v_1 \cdot \dots \cdot v_m(d)))$  with  $u_1 \cdot \dots \cdot u_m = v_1 \cdot \dots \cdot v_m$ . That is, when  $n$ -PCP has a solution for  $P$ .

Thus,  $n$ -PCP is reducible to a reachability problem in an  $n + 1$ -growing TRS. As  $n$ -PCP is undecidable for  $n \geq 1$ , so is reachability for  $n + 1$ -growing TRSs with  $n \geq 1$ .  $\square$

Observe in the previous proof that if we assume  $n = 0$ , then the rewrite rule

$$f(a^n(x), a^n(y)) \rightarrow h^{n+1}(f(x, y))$$

collapses to

$$f(x, y) \rightarrow h^{n+1}(f(x, y)).$$

As a consequence,  $d$  is no longer reachable from  $c$ . Something like this was to be expected, as reachability is decidable for 1-growing TRSs.

We now extend the above result to normalisation and neededness.

**Theorem 4.4** *Let  $n \geq 1$ . Normalisation is undecidable for  $n + 1$ -growing TRSs.*

**Proof** We reduce  $n$ -PCP to normalisation in an  $n + 1$ -growing TRS employing the proof showing that  $n$ -PCP is reducible to reachability.

Note that adding the  $n + 1$ -growing rewrite rule

$$(7) \quad h^{n+1}(x) \rightarrow h^{n+1}(h^{n+1}(x))$$

to the TRS from the proof of Theorem 4.3 does not change the fact that  $d$  is reachable from  $c$  if and only if  $n$ -PCP has a solution for  $P$ . However, by adding the rule, if  $d$  is reachable from  $c$ , then  $d$  becomes the only possible

normal form of  $c$ . By this fact, and the fact that  $d$  is reachable from  $c$  if and only if  $n$ -PCP has a solution for  $P$ , we have that  $c$  has a normal form if and only if  $n$ -PCP has a solution for  $P$ .

Thus,  $n$ -PCP is reducible to normalisation in an  $n + 1$ -growing TRS. As  $n$ -PCP is undecidable for  $n \geq 1$ , so is normalisation for  $n + 1$ -growing TRSs with  $n \geq 1$ .  $\square$

**Theorem 4.5** *Let  $n \geq 1$ . Neededness is undecidable for  $n + 1$ -growing TRSs.*

**Proof** Suppose we have an  $n + 1$ -growing TRS with constants  $c_1, c_2, c_3$ , constants  $d_1, d_2$  and some rewrite rules with as their left-hand side  $c_1, c_2, c_3$ . We can add to this  $n + 1$ -growing TRS the  $n + 1$ -growing rewrite rules of Gustave's TRS [10, Example 9.2.35]

$$f(d_1, d_2, x) \rightarrow e$$

$$f(x, d_1, d_2) \rightarrow e$$

$$f(d_2, x, d_1) \rightarrow e$$

Here,  $f$  is a fresh function symbol and  $e$  is fresh constant.

If we now consider the term  $f(c_1, c_2, c_3)$ , then we cannot decide which of the three present redexes is needed. It requires us to know for all  $1 \leq i \leq 3$  if  $d_1$  and  $d_2$  are reachable from  $c_i$  and reachability of  $n + 1$ -growing is undecidable by Theorem 4.3. Hence, neededness is undecidable.  $\square$

As a consequence of the previous theorem we have that no algorithm exists that performs a needed reduction strategy in  $n + 1$ -growing TRSs with  $n \geq 1$ .

## 5 Undecidable $n$ -Shallow Term Rewriting Systems

In this section we describe our second class of TRSs for which reachability, normalisation, and neededness are undecidable. The class is defined as follows.

**Definition 5.1** Let  $l \rightarrow r$  be a rewrite rule. The rule is  *$n$ -shallow* when it is linear and when for all  $x \in \mathcal{V}ar(l) \cap \mathcal{V}ar(r)$  it holds that  $d_l(x) = d_r(x) = n$ . A TRS is  *$n$ -shallow* when all its rewrite rules are  $n$ -shallow.

Observe that, like  $n$ -growing TRSs,  $n$ -shallow TRSs have restrictions on the forms of their rewrite rules. The  $n$ -shallow TRSs form neither a sub-class nor a super-class of the  $n$ -growing TRSs. Consider, for example, the  $n$ -shallow rewrite rule

$$f^n(x) \rightarrow f^n(x).$$

This rewrite rule is not  $n$ -growing, as  $d_{f^n(x)}(x) = d_{f^n(x)}(x)$ . We also have the  $n$ -growing rewrite rule

$$f^n(x) \rightarrow f^{n+1}(x).$$

This rewrite rule is not  $n$ -shallow, as  $d_{f^n(x)}(x) < d_{f^{n+1}(x)}(x)$ .

The  $n$ -shallow rewrite rules and TRSs are closely related to the following rewrite rules and TRSs, as defined by Comon [1, Sect. 4].

**Definition 5.2** Let  $l \rightarrow r$  be a rewrite rule. The rule is *shallow* when it is linear and when for all  $x \in \mathcal{V}ar(l) \cap \mathcal{V}ar(r)$  it holds that  $d_l(x) = 1$  and  $d_r(x) \leq 1$ . A TRS is *shallow* when all its rewrite rules are shallow.

Obviously, for  $n = 1$  the  $n$ -shallow TRSs form a sub-class of the shallow TRSs. For  $n > 1$  this is not the case. For example, consider again the  $n$ -shallow rewrite rule

$$f^n(x) \rightarrow f^n(x).$$

For  $n > 1$  this rewrite rule is not shallow, as  $d_{f^n(x)}(x) = d_{f^n(x)}(x) = n > 1$ .

The shallow TRSs do not form a sub-class of the  $n$ -shallow TRSs for any  $n$ . This follows by the same example that shows that the growing TRSs do not form a sub-class of the  $n$ -growing TRSs for any  $n$ .

Using tree automata techniques, Comon [1] proves that reachability and normalisation are decidable for shallow TRSs. Durand and Middeldorp [2] prove that neededness is decidable for orthogonal shallow TRSs. As each 1-shallow TRS is shallow, we also have decidability of reachability, normalisation, and (in the orthogonal case) neededness for 1-shallow TRSs. However, as we show next, these results do not generalise to  $n$ -shallow TRSs with  $n > 1$ .

In the proofs below, we denote by  $[a, b]$  a unary function symbol with  $a, b \in \Delta = \Gamma \uplus \{e\}$ . We also use the following definition.

**Definition 5.3** Let  $u, v \in \Delta^+$  with  $|u| = |v|$ . For  $a, b \in \Delta$  and  $u', v' \in \Delta^+$ , define  $[u, v](x)$  by

- $[u, v](x) = [a, b](x)$  if  $u = a$  and  $v = b$ , and
- $[u, v](x) = [a, b]([u', v'](x))$  if  $u = a \cdot u'$  and  $v = b \cdot v'$ .

**Theorem 5.4** Let  $n \geq 1$ . *Reachability is undecidable for  $n+1$ -shallow TRSs.*

**Proof** We reduce padded  $n+1$ -PCP to reachability in an  $n+1$ -shallow TRS. Suppose we have a finite set  $P$  of padded  $n+1$ -PCP pairs and an  $e \notin \Gamma$ . Let  $\Delta = \Gamma \uplus \{e\}$ . Define the signature  $\Sigma = \{[a, b] \mid a, b \in \Delta\} \uplus \{c, d\}$ , with each  $[a, b]$  a unary function symbol and  $c, d$  constants. Also define for all  $(u, v) \in P$  and  $a \in \Delta$  the following rewrite rules

- (1)  $c \rightarrow [u, v](c)$
- (2)  $[a, a](c) \rightarrow d$
- (3)  $[a, a](d) \rightarrow d$

Finally, define for all  $u, u', v, v' \in \Delta^+$  with  $u[e := \epsilon] = u'[e := \epsilon]$ ,  $v[e := \epsilon] = v'[e := \epsilon]$ , and  $|u| = |u'| = |v| = |v'| = n+1$  the following rewrite rule

- (4)  $[u, v](x) \rightarrow [u', v'](x)$

This last rewrite rule can be considered as a series of applications to the strings  $u$  and  $v$  of the rewrite rules given just before Problem 3.8.

As is easy to see, we have a finite number of  $n+1$ -shallow rewrite rules. Hence, the rewrite rules form an  $n+1$ -shallow TRS. By  $n \geq 1$ , we have for

the TRS that  $d$  is reachable from  $c$  if and only if padded  $n + 1$ -PCP has a solution for  $P$ .

To see that  $d$  is reachable from  $c$  when padded  $n + 1$ -PCP has a solution for  $P$ , suppose that for some  $s \in \Delta^+$  we have  $u_1 \cdot \dots \cdot u_m \rightarrow^* s$  and  $v_1 \cdot \dots \cdot v_m \rightarrow^* s$ . That is, padded  $n + 1$ -PCP has a solution for  $P$ . We can construct the following reduction sequence

$$\begin{aligned} c &\rightarrow_{(1)} [u_1, v_1](c) \\ &\rightarrow_{(1)}^* [u_1 \cdot \dots \cdot u_m, v_1 \cdot \dots \cdot v_m](c) \end{aligned}$$

As  $u_1 \cdot \dots \cdot u_m \rightarrow^* s$  and  $v_1 \cdot \dots \cdot v_m \rightarrow^* s$ , we can extend the reduction sequence to

$$\begin{aligned} c &\rightarrow^* [u_1 \cdot \dots \cdot u_m, v_1 \cdot \dots \cdot v_m](c) \\ &\rightarrow_{(4)}^* [s, s](c) \\ &\rightarrow_{(2)} \dots \\ &\rightarrow_{(3)}^* d \end{aligned}$$

Hence,  $d$  is reachable from  $c$ .

To see that  $n$ -PCP has a solution for  $P$  when  $d$  is reachable from  $c$ , note that the only way to reduce  $c$  to  $d$  is to first perform a number of (1)-steps, then to perform a (2)-step, and to finally perform a number of (3)-steps and to interleave these steps with (4)-steps. Also note that the reduction sequence only ends in  $d$  if the (1)-steps together with a number of (4)-steps give us a term  $[s, s](c)$  for some  $u_1 \cdot \dots \cdot u_m \rightarrow^* s$  and  $v_1 \cdot \dots \cdot v_m \rightarrow^* s$ . That is, as padded  $n + 1$ -PCP has a solution for  $P$ .

Thus, padded  $n + 1$ -PCP is reducible to a reachability problem in an  $n + 1$ -shallow TRS. As padded  $n + 1$ -PCP is undecidable for  $n \geq 1$ , so is reachability for  $n + 1$ -shallow TRSs with  $n \geq 1$ .  $\square$

Observe in the previous proof that if we assume  $n = 0$ , then the last rewrite rule collapses to

$$[a, b](x) \rightarrow [a, b](x)$$

with  $a, b \in \Delta$ . As a consequence,  $d$  is no longer reachable from  $c$ . Something like this was to be expected, as reachability is decidable for 1-shallow TRSs.

Note that by the TRS specified in the previous proof an even stronger property holds.

**Theorem 5.5** *Let  $n \geq 1$ . Reachability is undecidable for  $n + 1$ -shallow TRSs in which every rewrite rule has at most one variable which occurs both at the left-hand and right-hand side of the rewrite rule.*

We now extend the above result to normalisation and neededness.

**Theorem 5.6** *Let  $n \geq 1$ . Normalisation is undecidable for  $n + 1$ -shallow TRSs.*

**Proof** We reduce padded  $n + 1$ -PCP to normalisation in an  $n + 1$ -shallow TRS employing the proof showing that  $n + 1$ -PCP is reducible to reachability.

Note that adding for all  $u, v \in \Delta^+$  with  $|u| = |v| \leq n + 1$  the  $n + 1$ -shallow rewrite rule

$$(5) \quad [u, v](d) \rightarrow [u, v](d)$$

to the TRS from the proof of Theorem 5.4 does not change the fact that  $d$  is reachable from  $c$  if and only if padded  $n+1$ -PCP has a solution for  $P$ . However, by adding these rules, if  $d$  is reachable for  $c$ , then the term  $d$  becomes the only possible normal form of  $c$ . By this fact and the fact that  $d$  is reachable from  $c$  if and only if padded  $n + 1$ -PCP has a solution for  $P$ , we have that  $c$  has a normal form if and only if padded  $n + 1$ -PCP has a solution for  $P$ .

Thus, padded  $n + 1$ -PCP is reducible to normalisation in an  $n + 1$ -shallow TRS. As padded  $n + 1$ -PCP is undecidable for  $n \geq 1$ , so is normalisation for  $n + 1$ -shallow TRSs with  $n \geq 1$ .  $\square$

**Theorem 5.7** *Let  $n \geq 1$ . Neededness is undecidable for  $n + 1$ -shallow TRSs.*

**Proof** This proof is equal to the proof of Theorem 4.5, but with  $n + 1$ -shallow substituted for  $n + 1$ -growing, and with Theorem 5.4 substituted for Theorem 4.3.  $\square$

As a consequence of the previous theorem we have that no algorithm exists that performs a needed reduction strategy in a  $n + 1$ -shallow TRSs with  $n \geq 1$ .

## 6 Related Work

In this section, we compare  $n$ -growing and  $n$ -shallow TRSs to four other classes of TRSs for which reachability, normalisation, and neededness are undecidable [4,5,6]. We use the following four rewrite rules in the comparison.

- (1)  $f(x, x) \rightarrow g(x)$
- (2)  $f(f(x)) \rightarrow x$
- (3)  $f^n(x, g(y)) \rightarrow f^{n+1}(x)$
- (4)  $f^n(x, g(y)) \rightarrow f^n(x)$

Note that (1) and (2) are neither  $n$ -growing nor  $n$ -shallow and that (3) and (4) are respectively  $n$ -growing and  $n$ -shallow.

The first class, defined by Jacquemard [4, Theorem 18], requires for all  $l \rightarrow r$  and  $x \in \mathcal{V}ar(l) \cap \mathcal{V}ar(r)$  that  $d_l(x) \leq 1$ . This definition is equal to the definition of growing, except that rewrite rules no longer need to be linear. Hence 1-growing and 1-shallow TRSs form sub-classes. For  $n > 1$  the  $n$ -growing and  $n$ -shallow TRSs do not form sub-classes by (3) and (4). Moreover, for all  $n$  the  $n$ -growing and  $n$ -shallow TRSs do not form super-classes by (1), which is neither  $n$ -growing nor  $n$ -shallow as the left-hand side is not linear.

The second class, also defined by Jacquemard [4, Theorem 19], requires for all  $l \rightarrow r$  and  $x \in \mathcal{V}ar(l) \cap \mathcal{V}ar(r)$  that either  $d_l(x) \leq 1$  or  $d_r(x) \leq 1$  and that  $l \rightarrow r$  is linear. Again, this definition is equal to the definition of growing,

except that in this case  $d_r(x) \leq 1$  is also allowed. Consequently, 1-growing and 1-shallow TRSs form again sub-classes. For  $n > 1$  the  $n$ -growing and  $n$ -shallow TRSs do not form sub-classes by (3) and (4). Moreover, for all  $n$  the  $n$ -growing and  $n$ -shallow TRSs do not form super-classes by (2), which is neither  $n$ -growing nor  $n$ -shallow as  $d_x(x) = 0$ .

The third class, again defined by Jacquemard [5, Definition 1], requires for all  $l \rightarrow r$  that  $l$  and  $r$  are of the form  $f(t_1, \dots, t_k)$  with  $f \in \Sigma \cup X$  and with  $t_i$  either a variable or a ground term for all  $1 \leq i \leq k$ . For all  $n$  the  $n$ -growing and  $n$ -shallow TRSs do not form sub-classes by (3) and (4). Moreover, for all  $n$  the  $n$ -growing and  $n$ -shallow TRSs do not form super-classes by (1), which is neither  $n$ -growing nor  $n$ -shallow as the left-hand side is not linear.

The fourth class, defined by Jacquemard et al. [6, Proposition 6.1], but not presented in conference version [7] of this technical report, requires the rewrite rules to be of the following forms.

$$\begin{aligned} f_1(f_2(x)) &\rightarrow g_1(g_2(x)) \\ f_3(x) &\rightarrow t_1 \\ t_2 &\rightarrow g_3(x) \\ t_3 &\rightarrow t_4 \end{aligned}$$

where  $f_i$  and  $g_i$  are unary function symbols and  $t_i$  is a ground term for all  $1 \leq i \leq 4$ . For all  $n$  the classes of  $n$ -growing and  $n$ -shallow TRSs do not form sub-classes by (3) and (4). Moreover, for all  $n \neq 2$  the  $n$ -growing and  $n$ -shallow TRSs do not form super-classes by  $f_1(f_2(x)) \rightarrow g_1(g_2(x))$ . For  $n = 2$  the  $n$ -growing do not form super-classes again by  $f_1(f_2(x)) \rightarrow g_1(g_2(x))$ , but the 2-shallow TRSs do obviously form a super-class. Hence, Jacquemard et al. already proved that reachability, normalisation, and neededness are undecidable for 2-shallow TRSs. However, as the rewrite rules in proof of Theorem 5.4 show, it is even possible dispose of the rules  $f_3(x) \rightarrow t_1$  and  $t_2 \rightarrow g_3(x)$  and still end up with a class for which reachability is undecidable.

## 7 Further Directions

At least two questions remain. First of all, are reachability, normalisation, and neededness undecidable for TRSs in which for each rewrite rule  $l \rightarrow r$  we have for all  $x \in \mathcal{Var}(l) \cap \mathcal{Var}(r)$  that  $d_l(x) = n$  and  $d_r(x) < n$ ? Second, are reachability, normalisation, and neededness also undecidable for  $n$ -shallow and  $n$ -growing in case we require the TRSs to be orthogonal? Note that our proofs are no longer applicable, as they make heavy use TRSs which are not orthogonal.

## Acknowledgement

I would like to thank Jan Willem Klop, Aart Middeldorp, Femke van Raamsdonk, and Roel de Vrijer and the anonymous referees for their helpful com-

ments and remarks.

## References

- [1] Comon, H., *Sequentiality, monadic second-order logic and tree automata*, Information and Computation **157** (2000), pp. 25–51.
- [2] Durand, I. and A. Middeldorp, *Decidable call by need computations in term rewriting*, in: W. McCune, editor, *Proc. of the 14th Int. Conf. on Automated Deduction (CADE-14)*, LNCS **1249** (1997), pp. 4–18.
- [3] Huet, G. and J.-J. Lévy, *Computations in orthogonal rewriting systems*, in: J.-L. Lassez and G. Plotkin, editors, *Computational Logic*, MIT Press, 1991 pp. 395–443.
- [4] Jacquemard, F., *Decidable approximations of term rewriting systems*, in: H. Ganzinger, editor, *Proc. of the 7th Int. Conf. on Rewriting Techniques and Applications (RTA '96)*, LNCS **1103** (1996), pp. 362–376.
- [5] Jacquemard, F., *Reachability and confluence are undecidable for flat term rewriting systems*, Information Processing Letters **87** (2003), pp. 265–270.
- [6] Jacquemard, F., C. Meyer and C. Weidenbach, *Unification in extensions of shallow equational theories*, Technical Report MPI-I-98-2-002, Max-Planck-Institut für Informatik (1998).
- [7] Jacquemard, F., C. Meyer and C. Weidenbach, *Unification in extensions of shallow equational theories*, in: T. Nipkow, editor, *Proc. of the 9th Int. Conf. on Rewriting Techniques and Applications (RTA '98)*, LNCS **1379** (1998), pp. 76–90.
- [8] Post, E. L., *A variant of a recursively unsolvable problem*, Bulletin of the American Mathematical Society **52** (1946), pp. 264–268.
- [9] Seki, H., T. Takai, Y. Fujinaka and Y. Kaji, *Layered transducing term rewriting system and its recognizability preserving property*, in: S. Tison, editor, *Proc. of the 13th Int. Conf. on Rewriting Techniques and Applications (RTA 2002)*, LNCS **2378** (2002), pp. 98–113.
- [10] Terese, “Term Rewriting Systems,” Cambridge University Press, 2003.