



ELSEVIER

Information Processing Letters 84 (2002) 207–214

Information
Processing
Letters

www.elsevier.com/locate/ipl

Hierarchical termination revisited

Enno Ohlebusch

University of Bielefeld, Faculty of Technology, P.O. Box 100131, 33501 Bielefeld, Germany

Received 9 November 2001; received in revised form 25 February 2002

Communicated by H. Ganzinger

Keywords: Program correctness; Theory of computation; Term rewriting; Termination; Hierarchical programs

1. Introduction

During the past two decades, term rewriting has gained enormous importance in various fields of computer science. Among other things, term rewriting constitutes a Turing-complete computational model which is closely related to functional programming. In other words, term rewriting systems (TRSs) can be viewed as programs. It is a well-known programming paradigm in computer science that programs should be developed incrementally. That is, a programmer defines some basic functions (collected in one or several base systems) and then proceeds by defining new functions using the given ones in an auxiliary manner. In an iteration of this process, the extended system will then act itself as a base system which is further extended by new functions and so on. A hierarchical structure of programs naturally emerges from this incremental way of programming.

Proving correctness of a program consists in showing partial correctness (that is, the program meets its specification) and termination (that is, the program cannot run forever). It is well known that termination is in general undecidable. However, powerful methods for automatically proving termination of TRSs exist. For instance, the innovative dependency pair technique provides such a method; see Arts and Giesl [1]. Here it will be shown that virtually all automatic termination proofs by the dependency pair method yield $\mathcal{C}_{\mathcal{E}}$ -termination. A TRS \mathcal{R} is $\mathcal{C}_{\mathcal{E}}$ -terminating if the TRS $\mathcal{R} \uplus \{Cons(x, y) \rightarrow x, Cons(x, y) \rightarrow y\}$ is terminating, where $Cons$ is some binary function symbol that does not occur in the signature of \mathcal{R} . Moreover, Urbain [11,12] showed how the dependency pair method can be used to prove ($\mathcal{C}_{\mathcal{E}}$ -)termination of TRSs incrementally. Of course, it would be even more desirable if the termination proofs of a base system \mathcal{R}_1 and its extension \mathcal{R}_2 could be done independently of each other and then (if some easily checked syntactic criteria are satisfied) one may conclude—without any extra proof—that the hierarchical combination of \mathcal{R}_1 and \mathcal{R}_2 is also terminating. In other words, for hierarchical combinations of TRSs, it would be desirable to know conditions under which termination is preserved. Dershowitz [2] coined the name “hierarchical termination” for this preservation, while other authors use the longer term “modularity of termination for hierarchical combinations”. By using Urbain’s [11,12] result about incremental termination proofs, we will prove the most powerful modularity result known so far. Furthermore, we will provide

E-mail address: enno@techfak.uni-bielefeld.de (E. Ohlebusch).

very simple proofs of known modularity results on simple termination. Finally, we will refute a conjecture of Krishna Rao [7].

Due to space limitations, we will review neither the dependency pair method nor basic concepts of modularity. The reader is referred to [1,4] instead. Alternatively, one may consult the full version of this article [10].

2. Proving \mathcal{C}_E -termination automatically

Many methods for proving termination automatically are based on simplification orderings, and thus they yield simple termination. Gramlich [5] showed the following theorem: If \mathcal{R} is simply terminating, then $\mathcal{R} \cup \text{Emb}(\mathcal{F})$ is \mathcal{C}_E -terminating.¹ To date, however, the TRSs for which automated termination proofs are (potentially) feasible are no longer just the simply terminating systems, but the *DP quasi-simply terminating* systems, i.e., those systems whose termination can be verified by quasi-simplification orderings in combination with dependency pairs.

As already mentioned, the definitions of the following notions can be found in [1,4]. A finite TRS \mathcal{R} is called *DP quasi-simply terminating* if for every cycle \mathcal{P} in the estimated dependency graph² of \mathcal{R} ($\text{EDG}(\mathcal{R})$) there exists an argument filtering system (AFS) \mathcal{A} and a quasi-simplification ordering (QSO) \succsim on $\mathcal{T}(\mathcal{F}', \mathcal{V})$ satisfying the constraints:

- (a) $l \downarrow_{\mathcal{A}} \geq r \downarrow_{\mathcal{A}}$ for every rule $l \in r$ in \mathcal{R} ,
- (b) $s \downarrow_{\mathcal{A}} \geq t \downarrow_{\mathcal{A}}$ for every dependency pair $\langle s, t \rangle$ in \mathcal{P} ,
- (c) $s \downarrow_{\mathcal{A}} \geq t \downarrow_{\mathcal{A}}$ for at least one dependency pair $\langle s, t \rangle$ in \mathcal{P} ,

where \mathcal{F}' consists of all function symbols occurring in the constraints.

Recall that a QSO is a quasi-rewrite ordering possessing the subterm property. Kruskal's theorem implies that the stable-strict relation corresponding to a QSO is well-founded provided the underlying signature is finite.

A natural question is whether DP quasi-simple termination also implies \mathcal{C}_E -termination. The next lemma shows that this is indeed the case.

Lemma 2.1. *A finite TRS \mathcal{R} is DP quasi-simply terminating if and only if $\mathcal{R} \uplus \mathcal{C}_E$, is DP quasi-simply terminating. In particular, DP quasi-simple termination implies \mathcal{C}_E -termination.*

Proof. The if-direction is obviously true. For the other direction, let \mathcal{R} be a DP quasi-simply terminating TRS. That is, for every cycle \mathcal{P} in $\text{EDG}(\mathcal{R})$ there exists an AFS \mathcal{A} and a QSO \prec on $\mathcal{T}(\mathcal{F}', \mathcal{V})$ such that \succsim satisfies the constraints (a)–(c). We have to show that $\mathcal{R} \uplus \mathcal{C}_E$ is DP quasi-simply terminating. First, note that every cycle \mathcal{P} in $\text{EDG}(\mathcal{R} \uplus \mathcal{C}_E)$ is also a cycle in $\text{EDG}(\mathcal{R})$.³

Let Ω be the function that replaces every subterm $\text{Cons}(u_1, u_2)$ of a term $u \in \mathcal{T}(\mathcal{F}' \cup \{\text{Cons}\}, \mathcal{V})$ with the distinguished variable z . Let the quasi-ordering \succsim' be defined on $\mathcal{T}(\mathcal{F}' \cup \{\text{Cons}\}, \mathcal{V})$ by $u \succsim' v$ if and only if $\Omega(u) \succsim \Omega(v)$. It is relatively simple to show that \succsim' is a quasi-rewrite ordering, i.e., \succsim' is closed under contexts and substitutions. The quasi-ordering \succsim'' defined by $\succsim'' = (\succsim' \cup \rightarrow_{\mathcal{C}_E})^*$ inherits closure under contexts and substitutions from \succsim' and $\rightarrow_{\mathcal{C}_E}$; hence it is a quasi-rewrite ordering. We claim that \succsim'' is a QSO on $\mathcal{T}(\mathcal{F}' \cup \{\text{Cons}\}, \mathcal{V})$. For every n -ary $f \in \mathcal{F}'$ and $1 \leq i \leq n$, we have

$$\Omega(f(x_1, \dots, x_n)) = f(x_1, \dots, x_n) \succsim x_i = \Omega(x_i)$$

¹ The TRS $\text{Emb}(\mathcal{F})$ contains, for every $f \in \mathcal{F}$ of arity n , the rewrite rules $f(x_1, \dots, x_n) \rightarrow x_i$ for all $i \in \{1, \dots, n\}$, in which the variables x_1, \dots, x_n are pairwise distinct.

² Here we use the definition of [1], but alternative definitions exist; see [9].

³ This is not true if we consider the dependency graph instead of the estimated dependency graph, as the system $f(0, 1, x) \rightarrow f(x, x, x)$ shows.

because \succsim is a QSO on $\mathcal{T}(\mathcal{F}', \mathcal{V})$. Therefore, $f(x_1, \dots, x_n) \succsim' x_i$ and hence also $f(x_1, \dots, x_n) \succsim'' x_i$. Moreover, we have $\text{Cons}(x_1, x_2) \succsim'' x_i$ because $\text{Cons}(x_1, x_2) \rightarrow_{\mathcal{C}_E} x_i$, where $i \in \{1, 2\}$. Thus \succsim'' has the subterm property. In other words, \succsim'' is a QSO on $\mathcal{T}(\mathcal{F}' \cup \{\text{Cons}\}, \mathcal{V})$. It is easy to verify that \succsim'' satisfies the constraints (a)–(c) with the same AFS as above and furthermore $\text{Cons}(x_1, x_2) \succsim'' x_i$ holds true. We conclude that $\mathcal{R} \uplus \mathcal{C}_E$ is DP quasi-simply terminating.

Lastly, DP quasi-simple termination of $\mathcal{R} \uplus \mathcal{C}_E$ implies that $\mathcal{R} \uplus \mathcal{C}_E$ is terminating, or equivalently, that \mathcal{R} is \mathcal{C}_E -terminating. \square

3. Proving \mathcal{C}_E -termination incrementally

In this section, we will briefly recapitulate some results that will be needed later.

Definition 3.1. Let $(\mathcal{F}_1, \mathcal{R}_1)$ and $(\mathcal{F}_2, \mathcal{R}_2)$ be TRSs. If there is an infinite sequence $\langle s_1, t_1 \rangle \langle s_2, t_2 \rangle \dots$ of dependency pairs from \mathcal{R}_1 and a substitution $\sigma : \mathcal{V} \rightarrow \mathcal{T}(\mathcal{F}_1 \cup \mathcal{F}_2, \mathcal{V})$ such that $t_j \sigma \rightarrow_{\mathcal{R}_1 \cup \mathcal{R}_2}^* s_{j+1} \sigma$ holds for all consecutive pairs $\langle s_j, t_j \rangle$ and $\langle s_{j+1}, t_{j+1} \rangle$ in the sequence, then the sequence $\langle s_1, t_1 \rangle \langle s_2, t_2 \rangle \dots$ is called an *infinite \mathcal{R}_1 -chain over $\mathcal{R}_1 \cup \mathcal{R}_2$* . Furthermore, if every subterm u of $s_j \sigma$ and $t_j \sigma$ is terminating with respect to $\mathcal{R}_1 \cup \mathcal{R}_2$, then the chain is called *minimal*.

In the following, we deal with a hierarchical combination $(\mathcal{F}, \mathcal{R})$ of the base system $(\mathcal{F}_1, \mathcal{R}_1)$ and the extension $(\mathcal{F}_2, \mathcal{R}_2)$. That is, \mathcal{R} is the TRS $\mathcal{R}_1 \cup \mathcal{R}_2$ over the signature $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2$ and $\mathcal{F}_1 \cap \mathcal{D}_2 = \emptyset$, where \mathcal{D}_2 denotes the set of defined symbols in \mathcal{R}_2 . We assume that the signatures contain already the tuple symbols used in the dependency pair approach. Because $\mathcal{D}_1 \cap \mathcal{D}_2 = \emptyset$ and every tuple symbol is a unique fresh constructor, this entails no loss of generality. Finally, we assume that both \mathcal{R}_1 and \mathcal{R}_2 are *finitely branching*.

Let $s \in \mathcal{T}(\mathcal{F})$ with $\text{root}(s) \in \mathcal{F}_1$. We use the notation $s = C[s_1, \dots, s_n]$ if $C[\dots] \in \mathcal{T}(\mathcal{F}_1)$ and $\text{root}(s_1), \dots, \text{root}(s_n) \in \mathcal{F}_2 \setminus \mathcal{F}_1$. Furthermore, $\mathcal{T}_{\text{SN}}(\mathcal{F})$ stands for the set $\{t \in \mathcal{T}(\mathcal{F}) \mid \text{every subterm } t' \text{ of } t \text{ with } \text{root}(t') \in \mathcal{F}_2 \setminus \mathcal{F}_1 \text{ is terminating with respect to } \mathcal{R}\}$.

The transformation Φ extends Gramlich's [5] transformation to hierarchical combinations.

Definition 3.2. Define $\Phi : \mathcal{T}_{\text{SN}}(\mathcal{F}) \rightarrow \mathcal{T}(\mathcal{F}_1 \uplus \{\text{Cons}, \text{Nil}\})$ by

- $\Phi(t) = t$, if $t \in \mathcal{T}(\mathcal{F}_1)$.
- $\Phi(t) = C[\Phi(s_1), \dots, \Phi(s_n)]$, if $\text{root}(t) \in \mathcal{F}_1$ and $t = C[t_1, \dots, t_n]$.
- $\Phi(t) = \text{Sort}(\Phi^*(\Delta_1^*(t)))$, if $\text{root}(t) \in \mathcal{F}_2 \setminus \mathcal{F}_1$

$$\text{with } \Delta_1^*(t) = \{u \in \mathcal{T}(\mathcal{F}) \mid t \rightarrow_{\mathcal{R}}^* u, \text{root}(u) \in \mathcal{F}_1\},$$

$$\Phi^*(M) = \{\Phi(u) \mid u \in M\} \quad \text{for } M \subseteq \mathcal{T}_{\text{SN}}(\mathcal{F}),$$

$$\text{Sort}(\{t_1, \dots, t_m\}) = \langle t_{\pi(1)}, \dots, t_{\pi(m)} \rangle \text{ such that } t_{\pi(j)} \succ t_{\pi(j+1)} \quad \text{for } 1 \leq j < m,$$

$$\langle t_{\pi(1)}, \dots, t_{\pi(m)} \rangle = \text{Cons}(t_{\pi(1)}, \dots, \text{Cons}(t_{\pi(m)}, \text{Nil}) \dots),$$

where \succ is an arbitrary but fixed total ordering on $\mathcal{T}(\mathcal{F}_1 \uplus \{\text{Cons}, \text{Nil}\})$.

Note that if $t \in \mathcal{T}_{\text{SN}}(\mathcal{F})$, then $\Delta_1^*(t) \subseteq \mathcal{T}_{\text{SN}}(\mathcal{F})$. Thus, Φ^* is always applied to a subset of $\mathcal{T}_{\text{SN}}(\mathcal{F})$.

Lemma 3.3. If $s \in \mathcal{T}_{\text{SN}}(\mathcal{F})$ and $s \rightarrow_{\mathcal{R}}^* t$, then $\Phi(s) \rightarrow_{\mathcal{R}_1 \uplus \mathcal{C}_E}^* \Phi(t)$.

With the preceding lemma (a proof of which can be found in [10]), it is not difficult to prove the following result of Urbain [11]; see [10] for an alternative proof.

Theorem 3.4. *Let \mathcal{R} be the hierarchical combination of two finitely branching TRSs \mathcal{R}_1 and \mathcal{R}_2 . \mathcal{R} is terminating ($\mathcal{C}_\mathcal{E}$ -terminating, respectively) if base system \mathcal{R}_1 is $\mathcal{C}_\mathcal{E}$ -terminating and there is no minimal infinite \mathcal{R}_2 -chain over \mathcal{R} (over $\mathcal{R} \uplus \mathcal{C}_\mathcal{E}$, respectively).*

4. Hierarchical $\mathcal{C}_\mathcal{E}$ -termination

In this section, we provide a new modularity result about hierarchical combinations of TRSs. We start with a result due to Gramlich [5].

Proposition 4.1. *$\mathcal{C}_\mathcal{E}$ -termination is a modular property of finitely branching constructor-sharing TRSs.*

Next, we will generalize this modularity result to certain hierarchical combinations, viz. proper extensions. In order to review the definition of proper extensions, due to Krishna Rao [7,8], we have to introduce the dependency relation \succeq_d .

For a TRS $(\mathcal{F}, \mathcal{R})$, the dependency relation \succeq_d is the smallest quasi-ordering satisfying the condition $f \succeq_d g$ whenever there is a rewrite rule $f(\dots) \rightarrow C[g(\dots)] \in \mathcal{R}$ with $g \in \mathcal{D}$. So $f \succeq_d g$ holds if the function f depends on the definition of the function g .

Let $(\mathcal{F}, \mathcal{R})$, be the hierarchical combination of base system $(\mathcal{F}_1, \mathcal{R}_1)$ and extension $(\mathcal{F}_2, \mathcal{R}_2)$. The defined symbols \mathcal{D}_2 of \mathcal{R}_2 are split into two sets \mathcal{D}_2^1 and \mathcal{D}_2^2 , where \mathcal{D}_2^1 contains all defined symbols of \mathcal{D}_2 that depend on a defined symbol of \mathcal{R}_1 , i.e., $\mathcal{D}_2^1 = \{f \mid f \in \mathcal{D}_2, f \succeq_d g \text{ for some } g \in \mathcal{D}_1\}$ and $\mathcal{D}_2^2 = \mathcal{D}_2 \setminus \mathcal{D}_2^1$.

\mathcal{R}_2 is a *proper extension* of \mathcal{R}_1 if every rewrite rule $l \rightarrow r \in \mathcal{R}_2$ satisfies the following condition: If t is a subterm of r such that $\text{root}(t) \in \mathcal{D}_2^1$ and $\text{root}(t) \succeq_d \text{root}(l)$, then t does not contain a function symbol from $\mathcal{D}_1 \cup \mathcal{D}_2^1$ strictly below its root.

\mathcal{R}_2 is a *restricted proper extension* of \mathcal{R}_1 if it is a proper extension of \mathcal{R}_1 such that no *left-hand side* of the rewrite rules of \mathcal{R}_2 contains a function symbol from $\mathcal{D}_1 \cup \mathcal{D}_2^1$ strictly below its root.

Example 4.2. \mathcal{R}_* is a restricted proper extension of \mathcal{R}_+ .

$$\mathcal{R}_+ = \begin{cases} 0 + y & \rightarrow y, \\ s(x) + y & \rightarrow s(x + y), \end{cases}$$

$$\mathcal{R}_* = \begin{cases} 0 * y & \rightarrow 0, \\ s(x) * y & \rightarrow (x * y) + y. \end{cases}$$

Now we are in a position to state our new modularity result.

Theorem 4.3. *Let \mathcal{R}_1 and \mathcal{R}_2 be finite $\mathcal{C}_\mathcal{E}$ -terminating TRSs. If \mathcal{R}_2 is a restricted proper extension of \mathcal{R}_1 , then $\mathcal{R}_1 \cup \mathcal{R}_2$ is $\mathcal{C}_\mathcal{E}$ -terminating.*

Proof. Let $\mathcal{R}_2^1 = \{l \rightarrow r \in \mathcal{R}_2 \mid \text{root}(l) \in \mathcal{D}_2^1\}$ and $\mathcal{R}_2^2 = \{l \rightarrow r \in \mathcal{R}_2 \mid \text{root}(l) \in \mathcal{D}_2^2\}$. The TRS \mathcal{R}_2^2 does not contain function symbols from $\mathcal{D}_1 \cup \mathcal{D}_2^1$. This is because for all $l \rightarrow r \in \mathcal{R}_2^2$ neither l (because \mathcal{R}_2 is a restricted proper extension of \mathcal{R}_1) nor r contains function symbols from $\mathcal{D}_1 \cup \mathcal{D}_2^1$ (if r would contain such a function symbol, then $\text{root}(l)$ would be an element of \mathcal{D}_2^1). In other words, \mathcal{R}_1 and \mathcal{R}_2^2 are constructor-sharing. Because both are $\mathcal{C}_\mathcal{E}$ -terminating, so is their union by Proposition 4.1. In order to show $\mathcal{C}_\mathcal{E}$ -termination of $\mathcal{R}_1 \cup \mathcal{R}_2 = (\mathcal{R}_1 \cup \mathcal{R}_2^2) \cup \mathcal{R}_2^1$, it suffices to show that there is no minimal infinite \mathcal{R}_2^1 -chain over $(\mathcal{R}_1 \cup \mathcal{R}_2) \uplus \mathcal{C}_\mathcal{E}$; see Theorem 3.4. Suppose, on the contrary, that there is a minimal infinite \mathcal{R}_2^1 -chain $\langle s_1, t_1 \rangle \langle s_2, t_2 \rangle \dots$ over $(\mathcal{R}_1 \cup \mathcal{R}_2) \uplus \mathcal{C}_\mathcal{E}$, i.e., there exists a substitution $\sigma : \mathcal{V} \rightarrow \mathcal{T}(\mathcal{F} \uplus \{\text{Cons}\})$ such that $t_j \sigma \rightarrow_{(\mathcal{R}_1 \cup \mathcal{R}_2) \uplus \mathcal{C}_\mathcal{E}}^* s_{j+1} \sigma$ holds for all consecutive pairs $\langle s_j, t_j \rangle$ and $\langle s_{j+1}, t_{j+1} \rangle$ in the sequence. Because we consider finite TRSs, we may assume that every

dependency pair in $\langle s_1, t_1 \rangle \langle s_2, t_2 \rangle \dots$ belongs to a cycle \mathcal{P} that consists solely of dependency pairs from \mathcal{R}_2^1 . Let $\langle F(u_1, \dots, u_n), G(t_1, \dots, t_m) \rangle$ be one of the dependency pairs in \mathcal{P} . Suppose $\langle F(u_1, \dots, u_n), G(v_1, \dots, v_m) \rangle$ originates from the rewrite rule $f(u_1, \dots, u_n) \rightarrow C[g(v_1, \dots, v_m)]$ of \mathcal{R}_2^1 . Clearly, we have $f \succeq_d g$ and $g \succeq_d f$ because \mathcal{P} is a cycle. Because \mathcal{R}_2 is a proper extension of \mathcal{R}_1 , the terms v_1, \dots, v_m do not contain symbols from $\mathcal{D}_1 \cup \mathcal{D}_2^1$. The same is true for u_1, \dots, u_n because \mathcal{R}_2 is a restricted proper extension of \mathcal{R}_1 . All in all, none of the subterms of the dependency pairs in \mathcal{P} contains a symbol from $\mathcal{D}_1 \cup \mathcal{D}_2^1$.

Since \mathcal{R}_2^2 does not contain function symbols from $\mathcal{D}_1 \cup \mathcal{D}_2^1$ the TRS $\mathcal{R}_1 \cup \mathcal{R}_2^1 \cup \mathcal{C}_{\mathcal{E}}$ can be viewed as an extension of \mathcal{R}_2^2 . Because the infinite \mathcal{R}_2^1 -chain is minimal (i.e., u is terminating with respect to $\mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{C}_{\mathcal{E}}$ for every subterm u of some $s_j\sigma$ and $t_j\sigma$), we may apply Lemma 3.3 to every $t_j\sigma \rightarrow_{\mathcal{R}_2^2 \cup (\mathcal{R}_1 \cup \mathcal{R}_2^1 \cup \mathcal{C}_{\mathcal{E}})}^* s_{j+1}\sigma$. This yields $\Phi(t_j\sigma) \rightarrow_{\mathcal{R}_2^2 \cup \mathcal{C}'_{\mathcal{E}}}^* \Phi(s_{j+1}\sigma)$, where $\mathcal{C}'_{\mathcal{E}} = \{Cons'(x, y) \rightarrow x, Cons'(x, y) \rightarrow y\}$ (because we have to use a fresh symbol $Cons'$ instead of the already used symbol $Cons$). In other words, the cycle \mathcal{P} admits an infinite \mathcal{R}_2^1 -chain over $\mathcal{R}_2^2 \cup \mathcal{C}'_{\mathcal{E}}$. This contradicts $\mathcal{C}_{\mathcal{E}}$ -termination of \mathcal{R}_2 . \square

Theorem 4.3 can be extended to restricted proper extension with a common subsystem; see [10]. Using Theorem 4.3, we can infer $\mathcal{C}_{\mathcal{E}}$ -termination of the following hierarchical combination originating from [4].

Example 4.4. Consider the TRS \mathcal{R}_1

$$\begin{array}{ll} 0 + y \rightarrow y & s(x) + y \rightarrow s(x + y) \\ nil ++ ys \rightarrow ys & (x : xs) ++ ys \rightarrow x : (xs ++ ys) \\ sum(x : nil) \rightarrow x : nil & sum(x : y : xs) \rightarrow sum((x + y) : xs) \\ sum(xs ++ (x : y : ys)) \rightarrow sum(xs ++ sum(x : y : ys)) & \end{array}$$

The function $sum(xs)$ computes the sum of all numbers in the list xs (e.g., sum applied to the list $[1, 2, 3]$ returns $[6]$). The polynomial interpretation

$$\begin{array}{ll} 0_{\mathbb{N}} = 0 & nil_{\mathbb{N}} = 0 \\ s_{\mathbb{N}}(x) = x + 1 & x :_{\mathbb{N}} xs = xs + 1 \\ x +_{\mathbb{N}} y = x + y & xs ++_{\mathbb{N}} ys = xs + ys + 1 \\ sum_{\mathbb{N}}(xs) = 1 & PLUS_{\mathbb{N}}(x, y) = x \\ APP_{\mathbb{N}}(xs, ys) = xs & SUM_{\mathbb{N}}(xs) = xs \\ Cons_{\mathbb{N}}(x, y) = x + y & \end{array}$$

shows $\mathcal{C}_{\mathcal{E}}$ -termination of \mathcal{R}_1 . (\mathcal{R}_1 is actually DP quasi-simply terminating.) Now we want to extend this system such that one can also compute the average of the numbers in a list. The TRS \mathcal{R}_2 does this:

$$\begin{array}{ll} 0 - s(y) \rightarrow 0 & x - 0 \rightarrow x \\ s(x) - s(y) \rightarrow x - y & quot(0, s(y)) \rightarrow 0 \\ quot(s(x), s(y)) \rightarrow s(quot(x - y, s(y))) & length(nil) \rightarrow 0 \\ length(x : xs) \rightarrow s(length(xs)) & hd(x : xs) \rightarrow x \\ avg(xs) \rightarrow quot(hd(sum(xs)), length(xs)) & \end{array}$$

By using an appropriate AFS and rpo , it is not difficult to show that \mathcal{R}_2 is DP quasi-simply terminating; see [1, Example 9]. Therefore, \mathcal{R}_2 is $\mathcal{C}_{\mathcal{E}}$ -terminating as well. Note that neither \mathcal{R}_1 nor \mathcal{R}_2 is simply terminating. We have $\mathcal{D}_2^1 = \{avg\}$, whereas all other symbols of \mathcal{D}_2 belong to \mathcal{D}_2^2 . Because avg does not occur on a right-hand side of a rewrite rule, \mathcal{R}_2 is a proper extension of \mathcal{R}_1 . Furthermore, since none of the symbols $+$, $++$, sum , and avg appears in a proper subterm of a left-hand side of the rewrite rules of \mathcal{R}_2 , \mathcal{R}_2 is even a restricted proper extension of \mathcal{R}_1 . Because \mathcal{R}_2 is a restricted proper extension of \mathcal{R}_1 , it follows from Theorem 4.3 that $\mathcal{R}_1 \cup \mathcal{R}_2$ is $\mathcal{C}_{\mathcal{E}}$ -terminating.

We stress that in the preceding example none of the known modularity results is applicable; see Section 6.

The following example taken from [5] demonstrates why a symbol from $\mathcal{D}_1 \cup \mathcal{D}_2^1$ may not occur below a \mathcal{D}_2^1 -symbol on the right-hand side of a rewrite rule from \mathcal{R}_2 : The TRSs $\mathcal{R}_1 = \{a \rightarrow b\}$ and $\mathcal{R}_2 = \{h(x, x) \rightarrow h(a, b)\}$ are simply terminating (hence \mathcal{C}_E -terminating), but their union is not terminating, as the derivation $h(a, b) \rightarrow_{\mathcal{R}_1} h(b, b) \rightarrow_{\mathcal{R}_2} h(a, b)$ shows.

The requirement that function symbols from $\mathcal{D}_1 \cup \mathcal{D}_2^1$ may not occur at non-root positions on the left-hand sides of rewrite rules from \mathcal{R}_2 is also necessary. This is shown by the following example.

Example 4.5. The term rewriting system $\mathcal{R}_1 = \{g(x, y) \rightarrow x\}$ is obviously \mathcal{C}_E -terminating. It can be shown that the same is true for the TRS (see [10]):

$$\mathcal{R}_2 = \begin{cases} f(0, 1, x) \rightarrow f(h(x), h(x), x) \\ h(0) \rightarrow 0 \\ h(g(x, y)) \rightarrow y \end{cases}$$

However, the hierarchical combination of \mathcal{R}_1 and \mathcal{R}_2 is not terminating:

$$\begin{aligned} f(0, 1, g(0, 1)) &\rightarrow_{\mathcal{R}_2} f(h(g(0, 1)), h(g(0, 1)), g(0, 1)) \\ &\rightarrow_{\mathcal{R}_1} f(h(0), h(g(0, 1)), g(0, 1)) \\ &\rightarrow_{\mathcal{R}_1}^+ f(0, 1, g(0, 1)). \end{aligned}$$

Note that the modularity of simple termination for restricted proper extensions does not directly follow from Theorem 4.3. This is because the fact that \mathcal{R}_2 is a proper extension of \mathcal{R}_1 does not imply that $\mathcal{R}_2 \cup \text{Emb}(\mathcal{F}_2)$ is a proper extension of $\mathcal{R}_1 \cup \text{Emb}(\mathcal{F}_1)$ (the systems $\text{Emb}(\mathcal{F}_1)$ and $\text{Emb}(\mathcal{F}_2)$ usually share defined symbols which is forbidden in hierarchical combinations). Nevertheless, it will be shown in the next section that simple termination is indeed modular for restricted proper extensions.

5. Hierarchical simple termination

In this section we will provide simple proofs of known results about the modularity of simple termination. Furthermore, we will disprove a conjecture of Krishna Rao [7]. Again, we start with a lemma due to Gramlich [5].

Lemma 5.1. *Simple termination is a modular property of finitely branching constructor-sharing TRSs.*

The next theorem constitutes the main result from [7]. The reader is invited to compare our proof with the long and involved original one.

Theorem 5.2. *Let $(\mathcal{F}_1, \mathcal{R}_1)$ and $(\mathcal{F}_2, \mathcal{R}_2)$ be finite simply terminating TRSs. If \mathcal{R}_2 is a restricted proper extension of \mathcal{R}_1 , then their hierarchical combination $(\mathcal{F}, \mathcal{R}) = (\mathcal{F}_1 \cup \mathcal{F}_2, \mathcal{R}_1 \cup \mathcal{R}_2)$ is simply terminating.*

Proof. Let $\mathcal{R}_2^1 = \{l \rightarrow r \in \mathcal{R}_2 \mid \text{root}(l) \in \mathcal{D}_2^1\}$ and $\mathcal{R}_2^2 = \{l \rightarrow r \in \mathcal{R}_2 \mid \text{root}(l) \in \mathcal{D}_2^2\}$. Furthermore, let $\mathcal{F}_2^2 = \{f \mid f \text{ occurs in } \mathcal{R}_2^2\}$. The TRS \mathcal{R}_2^2 does not contain function symbols from $\mathcal{D}_1 \cup \mathcal{D}_2^1$; cf. proof of Theorem 4.3. In other words, \mathcal{R}_1 and \mathcal{R}_2^2 are constructor-sharing. Because both are simply terminating, so is their union according to Lemma 5.1. In order to show termination of $\mathcal{R} \cup \text{Emb}(\mathcal{F})$, it is sufficient to show that there is no minimal infinite \mathcal{R}_2^1 -chain over $\mathcal{R} \cup \text{Emb}(\mathcal{F})$; see Theorem 3.4. Suppose, on the contrary, that there is a minimal infinite \mathcal{R}_2^1 -chain $\langle s_1, t_1 \rangle \langle s_2, t_2 \rangle \dots$ over $\mathcal{R} \cup \text{Emb}(\mathcal{F})$, i.e., there exists a substitution $\sigma : \mathcal{V} \rightarrow \mathcal{T}(\mathcal{F})$ such that $t_j \sigma \rightarrow_{\mathcal{R} \cup \text{Emb}(\mathcal{F})}^* s_{j+1} \sigma$ holds for all consecutive pairs $\langle s_j, t_j \rangle$ and $\langle s_{j+1}, t_{j+1} \rangle$ in the sequence. Because we consider finite TRSs, we may assume that every dependency pair in $\langle s_1, t_1 \rangle \langle s_2, t_2 \rangle \dots$ belongs to a cycle \mathcal{P} that consists solely of dependency

pairs from \mathcal{R}_2^1 . As in the proof of Theorem 4.3, one can show that none of the subterms of the dependency pairs in \mathcal{P} contains a symbol from $\mathcal{D}_1 \cup \mathcal{D}_2^1$.

Let $\mathcal{F}' = (\mathcal{F}_1 \cup \mathcal{F}_2) \setminus \mathcal{F}_2^2$. Because \mathcal{R}_2^2 does not contain function symbols from $\mathcal{D}_1 \cup \mathcal{D}_2^1$, the TRS $\mathcal{R}_1 \cup \mathcal{R}_2^1 \cup \text{Emb}(\mathcal{F}')$ can be viewed as an extension of $\mathcal{R}_2^2 \cup \text{Emb}(\mathcal{F}_2^2)$. Because the infinite \mathcal{R}_2^1 -chain is minimal, we may apply Lemma 3.3 to every

$$t_j \sigma \xrightarrow{*}_{(\mathcal{R}_2^2 \cup \text{Emb}(\mathcal{F}_2^2)) \cup (\mathcal{R}_1 \cup \mathcal{R}_2^1 \cup \text{Emb}(\mathcal{F}'))} s_{j+1} \sigma$$

and obtain $\Phi(t_j \sigma) \xrightarrow{*}_{(\mathcal{R}_2^2 \cup \text{Emb}(\mathcal{F}_2^2)) \uplus \mathcal{C}_\mathcal{E}} \Phi(s_{j+1} \sigma)$. In other words, the cycle \mathcal{P} admits an infinite \mathcal{R}_2^1 -chain over $(\mathcal{R}_2^2 \cup \text{Emb}(\mathcal{F}_2^2)) \uplus \mathcal{C}_\mathcal{E}$. Therefore, $\mathcal{R}_2 \cup \text{Emb}(\mathcal{F}_2)$ is not $\mathcal{C}_\mathcal{E}$ -terminating. This, however, contradicts the simple termination of \mathcal{R}_2 . \square

Theorem 5.2 can also be extended to restricted proper extension with a common subsystem; see [10] for details. It should be pointed out that Theorem 5.2 is not applicable to Example 4.4 because the TRSs are not simply terminating. Krishna Rao conjectured in [7] that Theorem 5.2 can be extended to finitely branching TRSs, but Example 5.3 disproves his conjecture.

Example 5.3. The term rewriting systems $\mathcal{R}_1 = \{a \rightarrow b\}$ and $\mathcal{R}_2 = \{h_j(x, x) \rightarrow h_{j+1}(a, b) \mid j \in \mathbb{N}\}$ are finitely branching and simply terminating (hence $\mathcal{C}_\mathcal{E}$ -terminating). \mathcal{R}_2 is a restricted proper extension of \mathcal{R}_1 because $h_{j+1} \not\leq_d h_j$ (so $a \in \mathcal{D}_1$ is allowed below $h_{j+1} \in \mathcal{D}_2^1$). However, the hierarchical combination of \mathcal{R}_1 and \mathcal{R}_2 is not terminating:

$$h_1(a, b) \rightarrow_{\mathcal{R}_1} h_1(b, b) \rightarrow_{\mathcal{R}_2} h_2(a, b) \rightarrow_{\mathcal{R}_1} h_2(b, b) \rightarrow_{\mathcal{R}_2} h_3(a, b) \rightarrow_{\mathcal{R}_1} \dots$$

6. Related work

Krishna Rao [8] and Giesl et al. [4] proved strong results concerning hierarchical innermost termination. Using these results, it is possible to infer innermost termination of the combined system \mathcal{R} in Example 4.4. If \mathcal{R} were an overlay system, one could further conclude that it is terminating; see Gramlich [6]. Note, however, that \mathcal{R} is not an overlay system. Thus, these results cannot show hierarchical termination of systems like that of Example 4.4.

Apart from the results on hierarchical innermost termination, Giesl et al. [4] also proved that DP quasi-simple termination is modular under disjoint union, but this and related results in [4] are confined to disjoint unions, constructor-sharing systems, or composable systems. Thus, these results cannot deal with hierarchical termination.

Other modularity criteria for hierarchical combinations are due to Dershowitz [2]. In his theorems, the system \mathcal{R}_2 must be *flat*, i.e., in any left- and right-hand side of a rule from \mathcal{R}_2 , no path from the root symbol has more than one D_2 symbol along it. Clearly, the system \mathcal{R}_2 of Example 4.4 is not flat because the right-hand side of the last rule contains nested D_2 symbols.

Another modularity result for hierarchical combinations was presented by Fernández and Jouannaud [3]. Their result is restricted to systems where the arguments of recursive calls in \mathcal{R}_2 decrease with respect to the subterm relation (compared as multisets or lexicographically). Hence, their result is also not applicable to Example 4.4.

References

- [1] T. Arts, J. Giesl, Termination of term rewriting using dependency pairs, *Theoret. Comput. Sci.* 236 (2000) 133–178.
- [2] N. Dershowitz, Hierarchical termination, in: *Proceedings of the 4th International Workshop on Conditional and Typed Rewriting Systems*, in: *Lecture Notes in Computer Science*, Vol. 968, Springer, Berlin, 1995, pp. 89–105.

- [3] M. Fernández, J.-P. Jouannaud, Modular termination of term rewriting systems revisited, in: *Recent Trends in Data Type Specification (Proceedings of the 10th Workshop on Specification of Abstract Data Types)*, in: *Lecture Notes in Computer Science*, Vol. 906, Springer, Berlin, 1995, pp. 255–272.
- [4] J. Giesl, T. Arts, E. Ohlebusch, Modular termination proofs for rewriting using dependency pairs, *J. Symbolic Comput.* (2002), to appear.
- [5] B. Gramlich, Generalized sufficient conditions for modular termination of rewriting, *Appl. Algebra Engrg. Comm. Comput.* 5 (1994) 131–158.
- [6] B. Gramlich, Abstract relations between restricted termination and confluence properties of rewrite systems, *Fund. Inform.* 24 (1995) 3–23.
- [7] M.R.K. Krishna Rao, Simple termination of hierarchical combinations of term rewriting systems, in: *Proceedings of the 2nd International Symposium on Theoretical Aspects of Computer Software*, in: *Lecture Notes in Computer Science*, Vol. 789, Springer, Berlin, 1994, pp. 203–223.
- [8] M.R.K. Krishna Rao, Modular proofs for completeness of hierarchical term rewriting systems, *Theoret. Comput. Sci.* 151 (1995) 487–512.
- [9] A. Middeldorp, Approximating dependency graphs using tree automata techniques, in: *Proceedings of the International Joint Conference on Automated Reasoning*, in: *Lecture Notes in Artificial Intelligence*, Vol. 2083, Springer, Berlin, 2001, pp. 593–610.
- [10] E. Ohlebusch, Hierarchical termination revisited. Report 01-05, *Forschungsberichte der Technischen Fakultät, Abteilung Informationstechnik*, Universität Bielefeld, 2001.
- [11] X. Urbain, Automated incremental termination proofs for hierarchically defined term rewriting systems, in: *Proceedings of the International Joint Conference on Automated Reasoning*, in: *Lecture Notes in Artificial Intelligence*, Vol. 2083, Springer, Berlin, 2001, pp. 485–499.
- [12] X. Urbain, *Approche incrémentale des preuves automatiques de terminaison*, PhD thesis, Université Paris-Sud, 2001.