Kruskal's Tree Theorem for Acyclic Term Graphs*

Georg Moser
Universität Innsbruck, Austria
georg.moser@uibk.ac.at

Maria A. Schett
Universität Innsbruck, Austria
maria.schett@uibk.ac.at

In this paper we study termination of term graph rewriting, where we restrict our attention to *acyclic* term graphs. Motivated by earlier work by Plump we aim at a definition of the notion of *simplification order* for acyclic term graphs. For this we adapt the homeomorphic embedding relation to term graphs. In contrast to earlier extensions, our notion is inspired by morphisms. Based on this, we establish a variant of Kruskal's Tree Theorem formulated for acyclic term graphs. In proof, we rely on the new notion of embedding and follow Nash-Williams' minimal bad sequence argument. Finally, we propose a variant of the *lexicographic path order* for acyclic term graphs.

1 Introduction

It is well-known that term graph rewriting is *adequate* for term rewriting. However, this requires suitable care in the treatment of sharing, typically achieved by extending the term graph rewrite relation with *sharing* (aka *folding*) steps and *unsharing* (aka *unfolding*) steps, cf. [20, 7]. If one focuses on term graph rewriting alone, then it is well-known that termination of a given graph rewrite system does not imply termination of the corresponding term rewrite system [19]. This follows as the representation of a term as a graph enables us to share equal subterms. However, if we do not provide the possibility to unshare equal subterms, we change the potential rewrite steps. Then not every term rewrite step can be simulated by a graph rewrite step. This motivates our interest in termination techniques *directly* for term graph rewriting. More generally our motivation to study term graph rewriting stems from ongoing work on complexity or termination analysis of programs based on transformation to term rewrite systems (see e.g. [14, 9, 21, 2, 3]). In particular in work on termination of imperative programs (see e.g. [21]) these works require a term representation of the heap, which would be much more naturally be encoded as term dags (see the definition below). However, complexity and termination analysis of term graph rewrite systems have only recently be conceived attention in the literature [8, 11, 10, 4]. In particular, at the moment there are no automated tools, which would allow an application for program analysis and could be compared to existing approaches using either AProVE [13] or T$_{\mathsf{C}}$T [5].

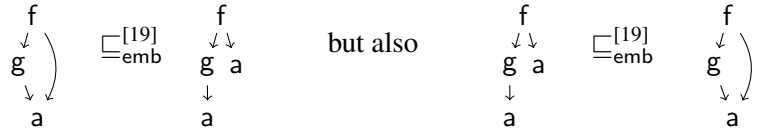
In our definition of term graph rewriting we essentially follow Barendsen [7], but also [6, 1], which are notationally closest to our presentation. We restrict our attention to term graphs, which represent such (finite) terms, that is in our context term graphs are directed, *rooted*, and *acyclic* graphs with node labels over a set of function symbols and variables. In term rewriting, termination is typically established via compatibility with a reduction order. Well-foundedness of such an order is more often than not a consequence of Kruskal's Tree Theorem [16] (e.g. in [12]). In particular, Kruskal's Tree Theorem underlies the concept of simple termination (see e.g. [17]). Indeed, Plump [19] defines a simplification order for acyclic term graphs. This order relies on the notion of *tops*. The top of a term graph is its root *and* its direct successors—thus keeping information on how these successors are shared.

*This work was partially supported by FWF (Austrian Science Fund) project P 25781-N15.

A. Corradini & H. Zantema (Eds.)
Computing with Terms and Graphs (TERMGRAPH 2016)
EPTCS 225, 2016, pp. 25–34, doi:10.4204/EPTCS.225.5

We recall briefly. Let $\leqslant$ be a partial order. If for any infinite sequence, we can find two elements a_i, a_j with $i < j$ where $a_i \leqslant a_j$, then $\leqslant$ is a well-quasi order. Now, Kruskal's Tree Theorem states, in a formulation suited to our needs, that given a well-quasi order $\sqsubseteq$ on the symbols in a term, the *homeomorphic embedding* relation $\sqsubseteq_{\mathsf{emb}}$ is a well-quasi order $\sqsubseteq_{\mathsf{emb}}$ on terms. We consider term graphs, not terms, and our symbols are tops. Usually, the relation $\sqsubseteq_{\mathsf{emb}}$ is simply called an *embedding*.

Plump [19] defines the embedding $\sqsubseteq^{[19]}_{\mathsf{emb}}$, but as he notes, for the following two term graphs, his definition of $\sqsubseteq^{[19]}_{\mathsf{emb}}$ holds in both directions.



In particular, [19] does not take sharing into account—except for direct successors through tops. This is a consequence of identifying each sub-graph independently. This is the inspiration and starting point for our work: We want to define an embedding relation, which also takes sharing into account. With this new embedding relation we re-prove Kruskal's Tree Theorem. Also here we take a slightly different approach to [19], which relies on an encoding of tops to function symbols with different arities. It is stated that there is a direct proof based on [18], which will be our direction.

As already mentioned, the context of this paper is the quest for termination techniques for term graph rewriting. Here *termination* refers to the well-foundedness of the graph rewrite relation $\to_{\mathcal{G}}$, induced by a graph rewrite system $\mathcal{G}$, cf. [7]. In particular, we seek a technique based on orders. This is in contrast to related work in the literature. There termination is typically obtained through interpretations or weights, cf. Bonfante et al. [8]. Also Bruggink et al. [11, 10] use an interpretation method, where they use type graphs to assign weights to graphs to prove termination. Finally, in [4] complexity of acyclic term graph rewriting is investigated, based on the use of interpretations and suitable adaptions of the dependency pair framework.

This paper is structured as follows. The next section provides basic definitions. In Section 3 we discuss potential adaptions of the homeomorphic embedding relation to term graphs and establish a suitable notion that extends the notion of *collapse* known from the literature. Section 4 establishes our generalisation of Kruskal's Tree Theorem to acyclic term graphs. In Section 5 we establish a new notion of simplification orders. Finally, in Section 6 we conclude and mention future work.

2 Preliminaries

First, we introduce our flavour of term graphs based on term dags, define term graph rewriting in our context, and give the collapse relation. Then we investigate tops with respect to a function symbol but also with respect to a node in a term graph. Based on this, we will consider a precedence on tops.

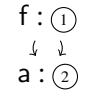
Definition 1. Let $\mathcal{N}$ be a set of nodes, $\mathcal{F}$ a set of function symbols, and $\mathcal{V}$ a set of variables. A *graph* is $G = (N, \mathsf{succ}, \mathsf{label})$, where $N \subseteq \mathcal{N}$, $\mathsf{succ} : N \to N^*$, and $\mathsf{label} : N \to \mathcal{F} \cup \mathcal{V}$. Here, succ maps a node n to an ordered list of *successors* $[n_1 \dots n_k]$. Further, label assigns labels, where *(i)* for every node $n \in G$ with $\mathsf{label}(n) = f \in \mathcal{F}$ we have $\mathsf{succ}(n) = [n_1, \dots, n_{\mathsf{arity}(f)}]$, and *(ii)* for every $n \in G$ with $\mathsf{label}(n) \in \mathcal{V}$, we have $\mathsf{succ}(n) = [\,]$. If G is acyclic, then G is a *term dag*.

The *size* of a graph $|G|$ is the number of its nodes N. We write $n \in G$ and mean $n \in N$, and call G *ground*, if $\mathsf{label} : N \to \mathcal{F}$. If $\mathsf{succ}(n) = [\dots, n_i, \dots]$, we write $n \xrightarrow{i} n_i$, or simply $n \rightharpoonup n_i$ for any i. Further, $\rightharpoonup^+$ is the transitive, and $\rightharpoonup^*$ the reflexive, transitive closure of $\rightharpoonup$. If $n \rightharpoonup^* n'$, then n' is

reachable from n. In the sub-graph $G\restriction[n_1,\ldots,n_k]$ all nodes reachable from $n_1,\ldots,n_k$ are collected, i.e. $N=\{n \mid n_i \rightharpoonup^* n, 1\leqslant i\leqslant k\}$, and the domains of succ and label are restricted accordingly.

Definition 2. Let T be a term dag. If all nodes are reachable from one node called $\mathsf{root}(T)$, that is, T is *rooted*, then T is a *term graph* with $\mathsf{inlets} := \mathsf{succ}(\mathsf{root}(T))$. For a term dag G with $\mathsf{inlets}=[t_1,\ldots,t_n]$, the *argument graph* is defined as $G\restriction$inlets', where inlets' $:= \mathsf{succ}(t_1)\cdots\mathsf{succ}(t_l)$.

Example 3. On the right we show the term graph $T=(\{①,②\},\mathsf{succ},\mathsf{label})$, with succ : $① \mapsto [②,②], ② \mapsto [\,]$, and $\mathsf{label} : ① \mapsto \mathsf{f}, ② \mapsto \mathsf{a}$. The term representation of T is $\mathsf{f}(\mathsf{a},\mathsf{a})$, $|T|=2$, and T is ground. The argument graph of T is $\mathsf{a} : ②$ with $\mathsf{inlets}=[②,②]$.



A *graph rewrite rule* is a term dag G with a root node l of the left hand side, and a root node r of right hand side. We denote a graph rewrite rule by $L\to R$, where $G\restriction[l]=L$ and $G\restriction[r]=R$. For a graph rewrite rule the following has to hold: *(i)* $\mathsf{label}(l)\notin\mathcal{V}$, *(ii)* if $n\in R$ with $\mathsf{label}(n)\in\mathcal{V}$ then $n\in L$, and *(iii)* for all nodes $n,n'\in G$, if $\mathsf{label}(n)=\mathsf{label}(n')\in\mathcal{V}$ then $n=n'$. A *graph rewrite system (GRS)* $\mathcal{G}$ is a set of graph rewrite rules. To define a *graph rewrite step*, we first need the auxiliary concepts of *redirection* of edges and *union* of two term dags. To *redirect* edges pointing from node u to node v, we write $G[v\leftarrow u]$, which is defined as $(N_G,\mathsf{succ}_{G[v\leftarrow u]},\mathsf{label}_G)$, where for all nodes $n\in G$, $\mathsf{succ}^i_{G[v\leftarrow u]}(n):=v$ if $n=u$, and $\mathsf{succ}^i_{G[v\leftarrow u]}(n):=n$ otherwise. Note, that for $G[v\leftarrow u]$ we still have $u\in G$. For two term dags G and H, their (left-biased) *union*, denoted by $G\oplus H$, is defined as $(N_G\cup N_H,\mathsf{succ}_G\oplus\mathsf{succ}_H,\mathsf{label}_G\oplus\mathsf{label}_H)$, where for $f\in\{\mathsf{succ},\mathsf{label}\}$ we define $f_G\oplus f_H(n):=f_G(n)$ if $n\in G$, and $f_H(n)$ if $n\notin G$ and $n\in H$. Note, that we do not require $N_G\cap N_H=\varnothing$. Next we investigate how to determine whether a graph rewrite rule matches a term graph. Therefore we first need to find a common structure between two graphs—through a *morphism*.

Definition 4. Let S,T be term graphs, and $\Delta\subseteq\mathcal{F}\cup\mathcal{V}$. A function $m:S\to T$ is *morphic* if for a node $n\in S$

(i) $\mathsf{label}_S(n)=\mathsf{label}_T(m(n))$ and

(ii) if $n\xrightarrow{i}_S n_i$ then $m(n)\xrightarrow{i}_T m(n_i)$ for all appropriate i.

A Δ-*morphism* from S to T is a mapping $m:S\to_\Delta T$, which is morphic in all nodes $n\in S$ with $\mathsf{label}(n)\notin\Delta$ and additionally $m(\mathsf{root}(S))=\mathsf{root}(T)$ holds.

A Δ-morphism only enforces Conditions *(i)* and *(ii)* on nodes with labels which are not in Δ. With $\Delta=\mathcal{V}$ we can determine whether a left-hand side of a graph rewrite rule *matches* a term graph, i.e., L matches S if there is a morphism $m:L\to_\mathcal{V} S$. Here, a node representing a variable in L can be mapped to a node with any label and successors. The morphism m is *applied* to R, denoted by $m(R)$, by redirecting all variable nodes in R to their image. That is, for all $n_1,\ldots,n_k\in R$, where $\mathsf{label}(n_i)\in\mathcal{V}$, we define $m(R)=((R\oplus S)[m(n_1)\leftarrow v_1])\ldots[m(n_k)\leftarrow n_k]$. Finally, for two term graphs S,T, n a node in S, and $N_S\cap N_T=\varnothing$, the replacement of the subgraph $S\restriction n$ by T, denoted $S[T]_n$, is defined as T, if $n=\mathsf{root}(S)$, and as $(S\oplus T)[\mathsf{root}(T)\leftarrow n]\restriction\mathsf{root}(S)$ otherwise.

Definition 5. Let $\mathcal{G}$ be a GRS. A term graph S *rewrites* to a term graph T, denoted by $S\to_\mathcal{G} T$, if there is a graph rewrite rule $L\to R\in\mathcal{G}$ with $N_R\cap N_S=\varnothing$, and a morphism $m:L\to S\restriction n$ such that $S[m(R)]_n=T$.

Finally, we can introduce the notion of termination.

Definition 6. If $\to_\mathcal{G}$ is well-founded, we say that the GRS $\mathcal{G}$ is *terminating*.

So far, we have not taken sharing into account—which we will investigate next. For term graphs S and T, we may ask: Is S a "more shared" version of T? Are S and T "equal"? To answer this, we rely again on a morphism as in Definition 4, where we require Condition *(i)* and *(ii)* for every node, i.e. we set $\Delta=\varnothing$.

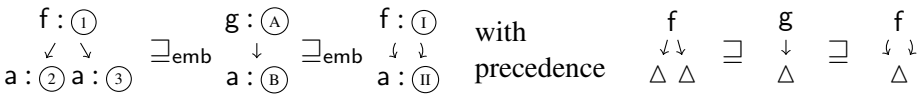


Figure 1: Intuitive Embeddings

Definition 7. If there is a morphism $m : S \to_{\varnothing} T$, then S *collapses* to T, denoted by $S \trianglerighteq T$. If $S \trianglerighteq T$ and $T \trianglerighteq S$, then S is *isomorphic* to T, denoted by $S \cong T$.

Reconsidering Example 3, let S be a tree representation of $\mathsf{f}(\mathsf{a},\mathsf{a})$, then $S \trianglerighteq T$. Now recall, that we aim to give a notion of Top, which takes the sharing of successor nodes into account, formalised via the collapse relation. Thus—with collapsing—we can give a definition of Tops for a function symbol f.

Definition 8. Let $f \in \mathcal{F}$, $\triangle$ a fresh symbol wrt. $\mathcal{F}$, and S a tree representation of $f(\triangle,\ldots,\triangle)$. Then $\mathsf{Tops}(f) = \{T \mid T \text{ is a termgraph, and } S \trianglerighteq T\}$ and $\mathsf{Tops}(\mathcal{F}) = \bigcup_{f\in\mathcal{F}} \mathsf{Tops}(f)$.

Now, similar to a precedence on function symbols, we define a precedence $\sqsubseteq$ on $\mathsf{Tops}(\mathcal{F})$.

Definition 9. A *precedence* on $\mathcal{F}$ is a transitive relation $\sqsubseteq$ on $\mathsf{Tops}(\mathcal{F})$, where for $S,T \in \mathsf{Tops}(\mathcal{F})$ we have *(i)* $S \cong T$ implies $S \sqsubseteq T$ and $T \sqsubseteq S$, and *(ii)* $T \sqsubseteq S$ implies $|T| \leqslant |S|$.

Condition *(i)* implies reflexivity, but also includes isomorphic copies. Condition *(ii)* hints at a major distinction to the term rewriting setting: We can distinguish the same function symbol with different degrees of sharing—and even embed nodes, which are labelled with function symbols with a smaller arity, in nodes labelled with function symbols with a larger arity. But, to ensure that such an embedding is indeed possible, enough nodes have to present—which is guaranteed by Condition *(ii)*. With Definition 8 we can compute the Tops for a function symbol—but we also want to compute the Top from some node in a term dag.

Definition 10. For a term dag $G = (N, \mathsf{succ}, \mathsf{label})$ and a node $n \in G$, we define $\mathsf{Top}_G(n) := (\{n\} \cup \mathsf{succ}(n), \mathsf{label'}, \mathsf{succ'})$, where *(i)* $\mathsf{label'}(n) = \mathsf{label}(n)$, $\mathsf{succ'}(n) = \mathsf{succ}(n)$, and *(ii)* for $n_i \in \mathsf{succ}(n)$, $\mathsf{label'}(n_i) = \triangle$, and $\mathsf{succ'}(n_i) = [\,]$.

For $\mathsf{Top}_G(n)$, where $\mathsf{label}_G(n) = f$, we find an isomorphic copy G' of $\mathsf{Top}_G(n)$ in $\mathsf{Tops}(f)$, i.e. $\mathsf{Top}_G(n) \cong G' \in \mathsf{Tops}(f)$.

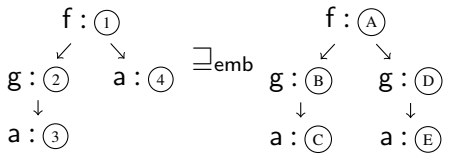
In the context of this work we focus on the graph rewrite relation $\to_{\mathcal{G}}$ and not on a relation combined with any explicit collapsing relation $\trianglerighteq$, as e.g., in [19]. In passing, we note that for the below established notion of homeomorphic embedding a similar relation to the collapse relation $\trianglerighteq$ is possible as in Plump's work, cf. [19, Lemma 24].

3 On Embedding

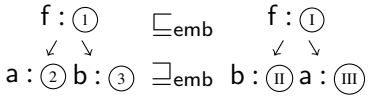
Next we continually develop a suitable definition of *homeomorphic embedding* for term dags. To get an intuition for embedding of term graphs consider the following example.

Example 11. In Figure 1, we find three term graphs, which are intuitively embedded from left to right under the given precedence.

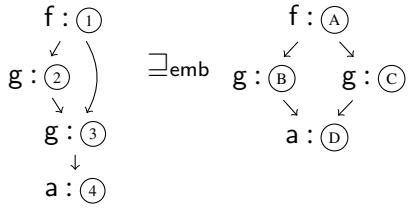
We base our definition of embedding on morphisms. We evolve this definition to highlight difficulties and pitfalls. In the first attempt we try mapping nodes from the embedded to the embedding graph.



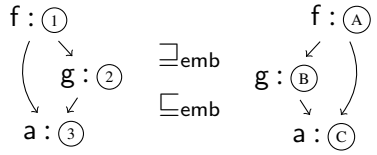
(a) Valid after Definition 12



(b) Valid after Definition 12 and Definition 15



(c) Demonstration of Definition 19



(d) Valid after Definition 19

Figure 2: Variants of Embedding

Definition 12 (first attempt)**.** Let $\sqsubseteq$ be a precedence. We say that S is *embedded* in T, denoted as $S \sqsubseteq_{\mathsf{emb}} T$, if there exists a function $m\colon S \to T$, such that for all nodes $s \in S$, we have

(i) $\mathsf{Top}_S(s) \sqsubseteq \mathsf{Top}_T(m(s))$, and

(ii) if $s \rightharpoonup_S s'$ for some $s' \in S$, then $m(s) \rightharpoonup_T^+ m(s')$ holds.

Example 13. We illustrate this definition with Figure 1. From the first to the second term dag we have a function m, with $m(\text{A}) = 1$, and either $m(\text{B}) = 2$ or $m(\text{B}) = 3$. Here m is not unique. From the second to the third term dag the morphism m' maps $m'(\text{I}) = \text{A}$ and $m'(\text{II}) = \text{B}$.

In Definition 12 the morphism maps nodes from the embedded graph S to nodes in the embedding graph T. The following example shows a problem arising from this.

Example 14. The embedding given in Figure 2(a) is valid after Definition 12. Here a morphism that satisfies both conditions is $m(\text{A}) = 1$, $m(\text{B}) = 2$, $m(\text{C}) = 3$, and also $m(\text{D}) = 2$ as well as $m(\text{E}) = 3$. This embedding could be prohibited by demanding m to be injective.

Demanding injectivity in Definition 12 prohibits the embedding $S \sqsubseteq_{\mathsf{emb}} T$ if $S \trianglelefteq T$ (in general). Thus we attempt to expand our definition such that a term dag also embeds a collapsed version of itself, i.e. embedding takes sharing into account. To achieve this the embedding relation has to contain the collapse relation of Definition 7. Then the embedding relation relies on a *partial* mapping from the embedding term graph S to T.

Definition 15 (second attempt)**.** Let $\sqsubseteq$ be a precedence. We say that S *embeds* T, denoted as $S \sqsupseteq_{\mathsf{emb}} T$, if there exists a partial, surjective function $m\colon S \to T$, such that for all nodes s in the domain of m, holds

(i) $\mathsf{Top}_T(m(s)) \sqsubseteq \mathsf{Top}_S(s)$, and

(ii) $m(s) \rightharpoonup_T m(s')$ implies $s \rightharpoonup_S^+ n'$ for some $n' \in \{n \mid (n) = m(s')\}$.

Example 16. Again consider Figure 1. From the first to the second term dag we have a function m, with $m(1) = \text{A}$, $m(2) = \text{B}$, and/or $m(3) = \text{B}$. Here m is not unique. From the second to the third term dag the morphism m' maps $m'(\text{A}) = 1$ and $m'(\text{B}) = \text{II}$.

One may observe, that both definitions of embedding so far are very permissive: it does not regard the order of the arguments. This is best illustrated by an example.

Example 17. The two term graph shown in Figure 2(b) representing the terms $\mathsf{f}(\mathsf{a},\mathsf{b})$ and $\mathsf{f}(\mathsf{b},\mathsf{a})$ are mutually embedded: from left to right we have the morphism m with $m(①) = ①$, $m(②) = ③$, and $m(③) = ②$. But—the inverse morphism m^{-1} also fulfils both conditions in Definition 12 and Definition 15.

To remedy this, we need to take the order of the arguments into account. Informally speaking, we want the preserve the relative order between the nodes: if a node n is "left of" a node n', $m(n)$ should be "left of" $m(n')$ in the embedded graph. For a formal description of "left of", we employ *positions*. Positions are sequences of natural numbers with $\cdot$ as delimiter. The set of positions of a node n in a term graph S is defined as follows: $\mathsf{Pos}_S(n) := \{\varepsilon\}$ if $n = \mathsf{root}(S)$, and $\mathsf{Pos}_S(n) := \{p \cdot i \mid \exists n' \in S \text{ with } n' \xrightarrow{i}_S n \text{ and } p \in \mathsf{Pos}_S(n')\}$ otherwise. For a term dag G with inlets_G, the base case is adapted slightly: $\mathsf{Pos}_G(n) := \{i\}$ if n is on ith position in inlets_G. We can now compare two positions p and q: p is left—or above—of q, if $p = p_1 \cdots p_k <_{\mathsf{lex}} q_1 \cdots q_l = q$, i.e. $p_i = q_i$ for $1 \leqslant i \leqslant j$ and $j = k < l$ or $p_j < q_j$.

We now have to extend this comparison from positions to nodes. This entails on the one hand an intra-node comparison which finds the smallest position within a node. Then an inter-node comparison comparing the smallest positions of two nodes. Two nodes are called *parallel* in a term graph G, if they are mutually unreachable.

Definition 18. Let G be a term dag. We define a partial order $\ll_G$ on the parallel nodes in G. Let $n, n' \in G$ and suppose n and n' are parallel. Further, suppose $p \in \mathsf{Pos}(n)$ is minimal wrt. $<_{\mathsf{lex}}$ and $q \in \mathsf{Pos}(n')$ is minimal wrt. $<_{\mathsf{lex}}$. Then $n \ll_G n'$ if $p <_{\mathsf{lex}} q$.

Based on the above definition, we develop Definition 15 further to the final version of embedding.

Definition 19 (final)**.** Let $\sqsubseteq$ be a precedence. We say that S *embeds* T, denoted as $S \sqsupseteq_{\mathsf{emb}} T$, if there exists a partial, surjective function $m\colon S \to T$, such that for all nodes s in the domain of m, holds

(i) $\mathsf{Top}_T(m(s)) \sqsubseteq \mathsf{Top}_S(s)$, and

(ii) $m(s) \rightharpoonup_T m(s')$ implies $s \rightharpoonup_S^+ n'$ for some $n' \in \{n \mid m(n) = m(s')\}$, and

(iii) $m(s) \ll_T m(s')$ implies either that none of the nodes in the preimage of $m(s')$ is parallel to s, or there exists $n' \in \{n \mid m(n) = m(s')\}$ such that $s \ll_S n'$.

Example 20. Recall Example 17. With the final definition of embedding, the two term graphs are not mutually embedded per se—embedding now depends on $\sqsubseteq$. As a further example for the embedding of the term graphs consider Figure 2(c). We have the following morphism: $m(①) = Ⓐ$, $m(②) = Ⓑ$, $m(③) = Ⓒ$, and $m(④) = Ⓓ$. Here we have $Ⓑ \ll Ⓒ$ but ② and ③ are not parallel. However, even with $\ll$ the two graphs in Figure 2(d) are mutually embedded. Here we have neither $② \ll ③$ nor $Ⓑ \ll Ⓒ$, so Condition *(iii)* holds trivially in both directions.

The relation $\sqsupseteq_{\mathsf{emb}}$ is transitive, i.e. $S \sqsupseteq_{\mathsf{emb}} T$ and $T \sqsupseteq_{\mathsf{emb}} U$ implies $S \sqsupseteq_{\mathsf{emb}} U$. The proof is straight forward: We construct the embedding $m_3 : S \to U$, based on the implied embeddings $m_2 : S \to T$ and $m_1 : T \to U$, by setting $m_3(n) = m_1(m_2(n))$ and show that m_3 fulfils the conditions in Definition 19.

4 Kruskal's Tree Theorem for Acyclic Term Graphs

Our proof follows [17] for the term rewrite setting, which in turn follows the minimal bad sequence argument of Nash-Williams [18]: we assume a minimal "bad" infinite sequence of term graphs and

construct an even smaller "bad" infinite sequence of their arguments. By minimality we contradict that this sequence of arguments is "bad", and conclude that it is "good". So we start by defining the notions of "good" and "bad".

Definition 21. Assume a reflexive and transitive order $\leqslant$, and an infinite sequence $\mathbf{a}$ with a_i, a_j in $\mathbf{a}$. If for some $i < j$ we have $a_i \leqslant a_j$, then $\mathbf{a}$ is *good*. Otherwise, $\mathbf{a}$ is *bad*. If every infinite sequence is good, then $\leqslant$ is a *well-quasi order* (wqo).

After we determined the sequence of arguments to be good, we want to— roughly speaking—plug the Top back on its argument. For this, we need a wqo on $\mathsf{Tops}(\mathcal{F})$ and the following, well established, lemma.

Lemma 22. *If $\leqslant$ is a wqo then every infinite sequence contains a subsequence—a* chain*—with $a_i \leqslant a_{i+1}$ for all i.*

With this lemma, we can construct witnesses that our original minimal bad sequence of term graphs is good, contradicting its badness and concluding the following theorem.

Theorem 23. *If $\sqsubseteq$ is a wqo on $\mathsf{Tops}(\mathcal{F})$, then $\sqsubseteq_{\mathsf{emb}}$ is a wqo on ground, acyclic term graphs.*

Proof. By definition, $\sqsubseteq_{\mathsf{emb}}$ is a wqo, if every infinite sequence is good, i.e. for every infinite sequence of term graphs, there are two term graphs T_i, T_j, such that $T_i \sqsubseteq_{\mathsf{emb}} T_j$ with $1 \leqslant i < j$. We construct a minimal bad sequence of term graphs $\mathbf{T}$: Assume we picked $T_1, \ldots, T_{n-1}$. We next pick T_n—minimal with respect to $|T_n|$—such that there are bad sequences that start with $T_1, \ldots, T_n$.

Let G_i be the argument graph of the ith term graph T_i. We collect in G the arguments of all term graphs of $\mathbf{T}$, i.e. $G = \bigcup_{i \geqslant 1} G_i$ and show that $\sqsubseteq_{\mathsf{emb}}$ is a wqo on G. For a contradiction, we assume G admits a bad sequence $\mathbf{H}$. We pick $G_k \in G$ with $k \geqslant 1$ such that $H_1 = G_k$. In G' we collect all argument graphs up to G_k, i.e. $G' = \bigcup_{i \geqslant 1}^{k} G_i$. The set G' is finite, hence there exists an index $l > 1$, such that for all H_i with $i \geqslant l$ we have that $H_i \in G$ but $H_i \notin G$'. We write $\mathbf{H}_{\geqslant l}$ for the sequence $\mathbf{H}$ starting at index l. Now consider the sequence $T_1, \ldots, T_{k-1}, G_k, \mathbf{H}_{\geqslant l}$. By minimality of $\mathbf{T}$ this is a good sequence. So we try to find a witness and distinguish on i, j:

$\underbrace{T_1, \ldots, T_{k-1}}_{i,j}, G_k, \mathbf{H}_{\geqslant l}$ For $1 \leqslant i < j \leqslant k-1$, we have $T_i \sqsubseteq_{\mathsf{emb}} T_j$, which contradicts the badness of $\mathbf{T}$.

$\underbrace{T_1, \ldots, T_{k-1}}_{i}, \underbrace{G_k}_{j}, \mathbf{H}_{\geqslant l}$ For $1 \leqslant i \leqslant k-1$ and $j = k$, we have $T_i \sqsubseteq_{\mathsf{emb}} G_k$ and $G_k \sqsubseteq_{\mathsf{emb}} T_k$, where the latter is a direct consequence of the definitions. Hence, by transitivity, $T_i \sqsubseteq_{\mathsf{emb}} T_j$, which contradicts the badness of $\mathbf{T}$.

$\underbrace{T_1, \ldots, T_{k-1}}_{i}, G_k, \underbrace{\mathbf{H}_{\geqslant l}}_{j}$ For $1 \leqslant i \leqslant k-1$ and $j \geqslant l$, we have $H_j \notin G$' by construction, but then $H_j = G_m$ for some $m > k$ and thus $H_j \sqsubseteq_{\mathsf{emb}} T_m$. Together with $T_i \sqsubseteq_{\mathsf{emb}} H_j$, we obtain $T_i \sqsubseteq_{\mathsf{emb}} T_m$ by transitivity, which contradicts the badness of $\mathbf{T}$.

$T_1, \ldots, T_{k-1}, \underbrace{G_k, \mathbf{H}_{\geqslant l}}_{i,j}$ Hence for some $1 \leqslant i < j$, where $i, j \notin \{2, \ldots, l-1\}$, we have some $H_i \sqsubseteq_{\mathsf{emb}} H_j$, which contradicts the badness of $\mathbf{H}$.

We conclude $\mathbf{H}$ is a good sequence and $\sqsubseteq_{\mathsf{emb}}$ is wqo on G.

Next we consider the Tops of $\mathbf{T}$. Let these Tops be $\mathbf{f}$. By assumption, $\sqsubseteq$ is a wqo on $\mathsf{Tops}(\mathcal{F})$, and by Lemma 22, $\mathbf{f}$ contains a chain $\mathbf{f}_\phi$, i.e. $f_{\phi(i)} \sqsubseteq f_{\phi(i+1)}$ for all $i \geqslant 1$. We proved $\sqsubseteq_{\mathsf{emb}}$ to be a wqo on G. Hence we have $G_{\phi(i)} \sqsubseteq_{\mathsf{emb}} G_{\phi(j)}$ for some $1 \leqslant i < j$. It remains to be shown, that $f_{\phi(i)} \sqsubseteq f_{\phi(j)}$ and $G_{\phi(i)} \sqsubseteq_{\mathsf{emb}} G_{\phi(j)}$ implies $T_{\phi(i)} \sqsubseteq_{\mathsf{emb}} T_{\phi(j)}$. We construct $T_{\phi(i)}$, and analogous $T_{\phi(j)}$, from $f_{\phi(i)} = (n_i, \mathsf{label}_{f\phi(i)}, \mathsf{succ}_{f\phi(i)})$ and $G_{\phi(i)} = (N_{G\phi(i)}, \mathsf{label}_{G\phi(i)}, \mathsf{succ}_{G\phi(i)})$ with $\mathsf{inlets}_{G\phi(i)}$. We have $N_{G\phi(i)} \cap \{n_i\} = \varnothing$. Then $T_{\phi(i)} = (N_{T\phi(i)}, \mathsf{label}_{T\phi(i)}, \mathsf{succ}_{T\phi(i)})$ where

(i) the nodes $N_{T\phi(i)} := N_{G\phi(i)} \cup \{n_i\}$,

(ii) $\mathsf{label}_{T\phi(i)} := \mathsf{label}_{G\phi(i)}$ extended by $\mathsf{label}_{T\phi(i)}(n_i) = \mathsf{label}_{f\phi(i)}(n_i)$, and

(iii) $\mathsf{succ}_{T\phi(i)} := \mathsf{succ}_{G\phi(i)}$ extended by $\mathsf{succ}_{T\phi(i)}(n_i) = \mathsf{inlets}_{G\phi(i)}$.

We aim for $T_{\phi(i)} \sqsubseteq_{\mathsf{emb}} T_{\phi(j)}$ and therefore construct the morphism $m : T_{\phi(j)} \to T_{\phi(i)}$. From $G_{\phi(i)} \sqsubseteq_{\mathsf{emb}} G_{\phi(j)}$, we obtain a morphism $m_G : G_{\phi(j)} \to G_{\phi(i)}$. We set $m(n) = m_G(n)$ for $n \in G_{\phi(j)}$, and $m(n_j) = n_i$. It remains to be shown that m fulfils Definition 19. Surjectivity of m follows directly from the surjectivity of m_G. Condition *(i)* holds for all nodes in m_G, and by $f_{\phi(i)} \sqsubseteq f_{\phi(j)}$ also for $\mathsf{root}(T_{\phi(j)}) = n_j$. For Condition *(ii)* we have to show: If $m(n_j) \rightharpoonup_{T_{\phi i}} n'_i = m(n'_j)$ then $n_j \rightharpoonup^+ n'_j$. By definition $n'_i \in \mathsf{inlets}_{G\phi(i)}$ and hence also $n'_i \in G_{\phi(i)}$. By surjectivity of m_G exist $m_G(n'_j) = n'_i$. It remains to be shown that $n_j \rightharpoonup^+ n'_j$. By definition $n_j \rightharpoonup u_j$, where $u_j \in \mathsf{inlets}_{G\phi(j)}$. By definition of argument graph, all nodes in $G_{\phi(j)}$ are reachable from nodes in $\mathsf{inlets}_{G\phi(j)}$, and in particular $n_j \rightharpoonup u_j \rightharpoonup^* n'_j$. Finally, Condition *(iii)* holds trivially for n and by $G_{\phi(i)} \sqsubseteq_{\mathsf{emb}} G_{\phi(j)}$. Hence we found a $T_{\phi(i)} \sqsubseteq_{\mathsf{emb}} T_{\phi(j)}$, which contradicts the badness of $\mathbf{T}$. Therefore $\mathbf{T}$ is good and $\sqsubseteq_{\mathsf{emb}}$ is a wqo. □

5 Simplification Orders

In the term rewriting setting simplification orders are defined through the embedding relation. That is, a rewrite order $\prec$ is a *simplification order* if $\sqsubset_{\mathsf{emb}} \subseteq \prec$ [17]. Then, if we can orient the rules in a rewrite system with $\prec$, there are no infinite rewrite sequences. We try to directly transfer this idea to the term graph rewriting setting—but this is not sufficient, as the following example shows.

Example 24. We can orient the rule on the left with $\sqsupset_{\mathsf{emb}}$, but still may get an infinite rewrite sequence, as shown on the right.

Note, that this infinite rewrite sequence is not bad wrt. $\sqsubseteq_{\mathsf{emb}}$.

This problem is *not* caused by our definition of embedding, and also occurs in [19]. Rather, the reason is that from orientation of the rules, we cannot conclude orientation of all rewrite steps. However, it should be noted, that the definition of simplification order in [19] is indeed transferable to our presentation.

Definition 25 ([19])**.** Let $\sqsubseteq_{\mathsf{emb}}$ be the embedding relation induced by a precedence $\sqsubseteq$ that is a wqo. A transitive relation $\prec$ is a *simplification order*, if

(i) $\sqsubset_{\mathsf{emb}} \subset \prec$, and

(ii) for all S and T, if $S \sqsubseteq_{\mathsf{emb}} T$ and $T \sqsubseteq_{\mathsf{emb}} S$ then $S \not\prec T$.

A direct consequence of the second condition is that simplification orders are irreflexive. We obtain the following theorem.

Theorem 26. *Every simplification order is well-founded.*

Proof. Let $\succ$ denote a simplification order. Thus there exists a well-quasi ordered precedence and an induced embedding relation, such that its strict part $\sqsubset_{\mathsf{emb}}$ is contained in $\succ$. Due to Theorem 23, $\sqsubset_{\mathsf{emb}}$ is a well-quasi order. Further, by definition $\succ$ is an irreflexive and transitive extension of $\sqsubseteq_{\mathsf{emb}}$. Thus $\succ$ is well-founded. □

Based on this observation, we adapt the definition of a *lexicographic path order* (*LPO* for short) from term rewriting to term graph rewriting and thus have a technique to show termination directly for acyclic term graph rewriting. Based on the above definition of embedding, it is natural to define LPO on term dags. Thus, we obtain the following definition of $<_{\mathsf{lpo}}$ induced by a well-quasi ordered precedence.

Definition 27. Let $\sqsubseteq$ be a well-quasi ordered precedence. We write $\sqsubset_{\mathsf{lex}}$ for the lexicographic extension of $\sqsubset$. Let S,T be term dags with $\mathsf{inlets}_S = [s_1,\ldots,s_k]$ and $\mathsf{inlets}_T = [t_1,\ldots,t_k]$, where s_i, s_j and t_i, t_j are parallel. Then $T <_{\mathsf{lpo}} S$ if one of the following holds

(i) $T \leqslant_{\mathsf{lpo}} S{\restriction}[s_{i_1},\ldots,s_{i_{k'}}]$ for some $1 \leqslant i_1 < \ldots < i_{k'} \leqslant k$, or

(ii) $[\mathsf{Top}(t_1),\ldots,\mathsf{Top}(t_l)] \sqsubset_{\mathsf{lex}} [\mathsf{Top}(s_1),\ldots,\mathsf{Top}(s_k)]$ and $\mathsf{arg}(T) <_{\mathsf{lpo}} S$, or

(iii) $[\mathsf{Top}(t_1),\ldots,\mathsf{Top}(t_l)] = [\mathsf{Top}(s_1),\ldots,\mathsf{Top}(s_k)]$ and $\mathsf{arg}(T) <_{\mathsf{lpo}} \mathsf{arg}(S)$.

Example 28. Recall Example 17. Given the precedence $\mathsf{a} \sqsubset \mathsf{b}$ we can orient the two term graphs: from right to left. To orient the term graphs wit $<_{\mathsf{lpo}}$ we first use *(iii)* and compare the argument graphs. Then we compare their respective inlets lexicographically, i.e., $[\mathsf{Top}(②),\mathsf{Top}(③)] \sqsubset_{\mathsf{lex}} [\mathsf{Top}(Ⅱ),\mathsf{Top}(Ⅲ)]$ using *(ii)*.

To prove that $<_{\mathsf{lpo}}$ contains $\sqsubseteq_{\mathsf{emb}}$ for term graphs, it important to note that $<_{\mathsf{lpo}}$ requires that nodes are parallel within inlets. That means, we can inductively step through a term graph, with inlets forming a level in the term graph. With *(i)* we can project the largest term dag to the dag that is actually used in the embedding.

6 Conclusion and Discussion

Inspired by [19] we defined an embedding relation for the term graph rewriting flavour of [1, 4] and re-proved Kruskal's Tree Theorem. Furthermore, based on Plump's work [19] we establish a new notion of simplification order for acyclic term graphs and provide a suitable adaption of the lexicographic path order to acyclic term graphs.

In contrast to [19], where the proof uses an encoding of Top to function symbols with different arities, our proof operates on term graphs. With a new definition of the embedding relation, based on the notion of morphism and taking sharing into account, and a new definition of arguments we finally showed Kruskal's Tree Theorem for term graphs: A well-quasi order on Tops, i.e. $\sqsubseteq$, induces a well-quasi order $\sqsubseteq_{\mathsf{emb}}$ on ground term graphs. One insight from our proof concerns the arguments of a term graph—or rather *the* argument. For a term structure we have several subterms as arguments. For a term graph structure it is beneficial to regard the arguments as only one single argument graph. This preserves sharing. Moreover a single argument simplifies the proof as extending the order to sequences, Higman's Lemma [15], can be omitted.

In future work, we will focus on the establishment of genuinely novel notions of simplification orders for term graph rewriting and investigate suitable adaptions of reduction orders for complexity analysis.

References

[1] M. Avanzini (2013): *Verifying Polytime Computability Automatically*. Ph.D. thesis, Universität Innsbruck, Austria.

[2] M. Avanzini, U. Dal Lago & G. Moser (2015): *Analysing the Complexity of Functional Programs: Higher-Order Meets First-Order*. In: *Proc. 20th ICFP*, ACM, pp. 152–164, doi:10.1145/2784731.2784753.

[3] M. Avanzini & G. Moser (2016): *A Combination Framework for Complexity*. *IC* 248, pp. 22–55, doi:10.1016/j.ic.2015.12.007.

[4] M. Avanzini & G. Moser (2016): *Complexity of Acyclic Term Graph Rewriting*. In: *Proc. 1st FSCD*, LIPIcs. To appear.

[5] M. Avanzini, G. Moser & M. Schaper (2016): *TcT: Tyrolean Complexity Tool*. In: *Proc. of 22nd TACAS*, LNCS, pp. 407–423, doi:10.1007/978-3-662-49674-9_24.

[6] H. P. Barendregt, M. v. Eekelen, J. R. W. Glauert, J. R. Kennaway, M. J. Plasmeijer & M. R. Sleep (1987): *Term Graph Rewriting*. In: *PARLE (2)*, *LNCS* 259, pp. 141–158, doi:10.1007/3-540-17945-3_8.

[7] E. Barendsen (2003): *Term Graph Rewriting*. In: *Term Rewriting Systems*, chapter 13, *CTTCS* 55, Cambridge University Press, pp. 712–743.

[8] G. Bonfante & B. Guillaume (2013): *Non-simplifying Graph Rewriting Termination*. In: *Proc. 7th TERMGRAPH*, EPTCS, pp. 4–16, doi:10.4204/EPTCS.110.3.

[9] M. Brockschmidt, R. Musiol, C. Otto & J. Giesl (2012): *Automated Termination Proofs for Java Programs with Cyclic Data*. In: *Proc. 24th CAV*, *LNCS* 7358, pp. 105–122, doi:10.1007/978-3-642-31424-7_13.

[10] H. J. S. Bruggink, B. König, D. Nolte & H. Zantema (2015): *Proving Termination of Graph Transformation Systems Using Weighted Type Graphs over Semirings*. In: *Proc. 8th ICGT*, *LNCS* 9151, pp. 52–68, doi:10.1007/978-3-319-21145-9_4.

[11] H. J. S. Bruggink, B. König & H. Zantema (2014): *Termination Analysis for Graph Transformation Systems*. In: *Proc. of 8th IFIP TC 1/WG 2.2*, *LNCS* 8705, pp. 179–194, doi:10.1007/978-3-662-44602-7_15.

[12] N. Dershowitz (1982): *Orderings for Term-Rewriting Systems*. *TCS* 17, pp. 279–301, doi:10.1016/0304-3975(82)90026-3.

[13] J. Giesl, M. Brockschmidt, F. Emmes, F. Frohn, C. Fuhs, C. Otto, M. Plücker, P. Schneider-Kamp, T. Ströder, S. Swiderski & R. Thiemann (2014): *Proving Termination of Programs Automatically with AProVE*. In: *Proc. 7th IJCAR*, *LNCS* 8562, pp. 184–191, doi:10.1007/978-3-319-08587-6_13.

[14] J. Giesl, M. Raffelsieper, P. Schneider-Kamp, S. Swiderski & R. Thiemann (2011): *Automated Termination Proofs for Haskell by Term Rewriting*. *TOPLAS* 33(2), pp. 7:1–7:39, doi:10.1145/1890028.1890030.

[15] G. Higman (1952): *Ordering by Divisibility in Abstract Algebras*. *Proc. London Mathematical Society* 3(2), pp. 326–336, doi:10.1112/plms/s3-2.1.326.

[16] J. B. Kruskal (1960): *Well-Quasi-Ordering, The Tree Theorem, and Vazsonyi's Conjecture*. *Trans. of the AMS* 95(2), pp. 210–225, doi:10.2307/1993287.

[17] A. Middeldorp & H. Zantema (1997): *Simple Termination of Rewrite Systems*. *TCS* 175, pp. 127–158, doi:10.1016/S0304-3975(96)00172-7.

[18] C. St. J. A. Nash-Williams (1963): *On Well-Quasi-Ordering Finite Trees*. *Proc. Cambridge Philosophical Society* 59, pp. 833–835, doi:10.1017/S0305004100003844.

[19] D. Plump (1997): *Simplification Orders for Term Graph Rewriting*. In: *MFCS*, pp. 458–467, doi:10.1007/BFb0029989.

[20] D. Plump (1999): *Term Graph Rewriting*. In: *Handbook of Graph Grammars and Computing by Graph Transformation*, chapter 1, 2, World Scientific, pp. 3–61, doi:10.1142/9789812815149_0001.

[21] T. Ströder, J. Giesl, M. Brockschmidt, F. Frohn, C. Fuhs, J. Hensel & P. Schneider-Kamp (2014): *Proving Termination and Memory Safety for Programs with Pointer Arithmetic*. In: *Proc. 7th IJCAR*, *LNCS* 8562, pp. 208–223, doi:10.1007/978-3-662-46681-0_32.