Proc. Mathematical Foundations of Computer Science '97, vol. 1295 of Lecture Notes in Computer Science, 458–467 (1997)

Simplification Orders for Term Graph Rewriting

Detlef Plump*

CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands
E-mail: det@cwi.nl

Abstract. Term graph rewriting differs from term rewriting in that common subexpressions can be shared, improving the efficiency of rewriting in space and time. Moreover, computations by term graph rewriting terminate more often than computations by term rewriting. In this paper, *simplification orders* on term graphs are introduced as a means for proving termination of term graph rewriting. Simplification orders are based on an extension of the homeomorphic embedding relation from trees to term graphs. By generalizing Kruskal's Tree Theorem to term graphs, it is shown that simplification orders are well-founded. Then a *recursive path order* on term graphs is defined by analogy with the well-known order on terms, and is shown to be a simplification order. Examples of termination proofs with the recursive path order are given for rewrite systems that are non-terminating under term rewriting.

1 Introduction

When computations with term rewrite rules are implemented in, for example, interpreters of functional programming languages, symbolic computation systems, or theorem provers, terms are often represented by graph-like data structures. Graphs, in contrast to trees, allow to *share* common subterms. This improves the efficiency of rewriting not only in space but also in time since repeated computations can be avoided.

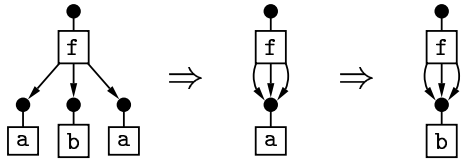
Term graph rewriting is a computational model in which term rewrite rules operate on graphs that represent terms. The technical setting of the present paper conforms to [8,15,16]. (See [1,2,9] and the collection [17] for some alternative approaches.) In this approach, term graphs can be transformed by both applications of term rewrite rules and so-called collapse steps which enhance the degree of sharing.

Compared with term rewriting, term graph rewriting is not only more efficient but also enjoys termination for a larger class of rewrite systems. For instance, the following non-terminating term rewriting system is given in [4]:

$$\begin{aligned} \mathtt{f}(\mathtt{a},\mathtt{b},\mathtt{x}) &\to \mathtt{f}(\mathtt{x},\mathtt{x},\mathtt{x}) \\ \mathtt{a} &\to \mathtt{b} \end{aligned}$$

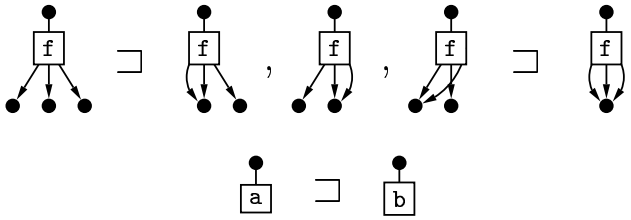
* On leave from Universität Bremen, Germany. Author's research is partially supported by the HCM Network EXPRESS, the ESPRIT Working Group APPLIGRAPH, and the TMR Network GETGRATS.

Non-termination is witnessed by the infinite rewrite sequence $\mathtt{f}(\mathtt{a},\mathtt{b},\mathtt{a}) \to \mathtt{f}(\mathtt{a},\mathtt{a},\mathtt{a}) \to \mathtt{f}(\mathtt{a},\mathtt{b},\mathtt{a}) \to \ldots$ In contrast, the same system does terminate under term graph rewriting. This is because graph rewrite steps with the first rule do not copy the argument `x` but create a shared subgraph. A terminating computation starting from the tree representing `f(a,b,a)` looks as follows:



The question arises how to prove termination for systems like the present one. Obviously, the techniques available for term rewriting (see [4] for a survey) are not directly applicable. In this paper, the well-known concept of a *simplification order* [3,12,18] is generalized from terms to term graphs. The main idea is to base simplification orders on precedences of so-called tops, which are graphs containing a single function symbol or variable. By ordering tops instead of function symbols, the homeomorphic embedding relation on trees can be extended to term graphs such that sharing as in the above derivation is reflected.

Consider, for instance, the following precedence (where the three tops in the middle of the first row are all smaller than the left top and greater than the right top):



Under this precedence the right term graph of the above derivation is embedded in the left term graph, but the left graph is *not* embedded in the middle graph. In contrast, the left graph (which is a tree) is homeomorphically embedded in the tree corresponding to the middle graph if `a` is greater than `b`.

Below it is shown that the embedding relation is a well-quasi-order on term graphs whenever the given precedence is a well-quasi-order on tops. This result extends Kruskal's Tree Theorem [11] to term graphs. Simplification orders are then defined as certain strict orders on term graphs such that "strictly embedded" is a special case of "simpler". These orders are shown to be well-founded whenever the underlying precedence is a well-quasi-order. Subsequently, a *recursive path order* on term graphs is introduced by analogy with the corresponding order on terms and is shown to be a simplification order. In the present example, the recursive path order over the given precedence allows to prove termination of term graph rewriting.

2 Term Graphs

A *signature* Σ is a set of function symbols such that each $f \in \Sigma$ comes with a natural number $\mathrm{arity}(f) \geq 0$. Function symbols of arity 0 are called *constants*. For simplicity, it is assumed that Σ contains at least one constant. A set X of *variables* for Σ must satisfy $X \cap \Sigma = \emptyset$. For each variable x, let $\mathrm{arity}(x) = 0$.

A *hypergraph* over $\Sigma \cup X$ is a system $G = \langle \mathrm{V}_G, \mathrm{E}_G, \mathrm{lab}_G, \mathrm{att}_G \rangle$ consisting of two finite sets V_G and E_G of *nodes* and *hyperedges*, a labelling function $\mathrm{lab}_G : \mathrm{E}_G \to \Sigma \cup X$, and an attachment function $\mathrm{att}_G : \mathrm{E}_G \to \mathrm{V}_G^*$ which assigns a string of nodes to a hyperedge e such that the length of $\mathrm{att}_G(e)$ is $1 + \mathrm{arity}(\mathrm{lab}_G(e))$. In the following, hypergraphs and hyperedges are simply called graphs and edges.

Given a graph G and an edge e with $\mathrm{att}_G(e) = v\, v_1 \ldots v_n$, node v is the *result node* of e while $v_1, \ldots, v_n$ are the *argument nodes*. The result node is denoted by $\mathrm{res}(e)$. For each node v, $G[v]$ is the subgraph consisting of all nodes that are reachable from v and all edges having these nodes as result nodes.

In pictures of graphs, edges are depicted as boxes with inscribed labels, and bullets represent nodes. A line connects each edge with its result node while arrows point to the argument nodes. The order among the argument nodes is given by the left-to-right order of the arrows leaving the box.

Definition 1 (Term graph). A graph G is a *term graph* if

(1) there is a node root_G from which each node is reachable,
(2) G is acyclic, and
(3) each node is the result node of a unique edge.

The set of all term graphs over $\Sigma \cup X$ is denoted by $\mathcal{TG}_{\Sigma,X}$, and $\mathcal{TG}_\Sigma$ stands for the subset of all term graphs without variables; the latter are called *ground term graphs*.

A *graph morphism* $f : G \to H$ between two graphs G and H consists of two functions $f_\mathrm{V} : \mathrm{V}_G \to \mathrm{V}_H$ and $f_\mathrm{E} : \mathrm{E}_G \to \mathrm{E}_H$ that preserve labels and attachment to nodes, that is, $\mathrm{lab}_H \circ f_\mathrm{E} = \mathrm{lab}_G$ and $\mathrm{att}_H \circ f_\mathrm{E} = f_\mathrm{V}^* \circ \mathrm{att}_G$ (where $f_\mathrm{V}^* : \mathrm{V}_G^* \to \mathrm{V}_H^*$ maps a string $v_1 \ldots v_n$ to $f_\mathrm{V}(v_1) \ldots f_\mathrm{V}(v_n)$). The morphism f is an *isomorphism* if f_V and f_E are bijective. In this case G and H are *isomorphic*, which is denoted by $G \cong H$.

3 A Well-quasi-order on Term Graphs

In this section, precedences are introduced as orders on certain small graphs. Every precedence induces an embedding relation on term graphs. Recall that a *preorder* (or *quasi-order*) is a reflexive and transitive relation, while a *strict order* is irreflexive and transitive. A strict order $\succ$ on a set A is *well-founded* (or *terminating*) if no infinite sequence $a_1 \succ a_2 \succ \ldots$ over A exists. A preorder $\succeq$ on A is a *well-quasi-order* (wqo for short) if for every infinite sequence $a_1, a_2, \ldots$ over A there are i and j such that $i < j$ and $a_i \preceq a_j$. Note that if A is finite, then every preorder on A is a well-quasi-order.

Definition 2 (Top). Let G be a term graph. The *top* of G, denoted by top_G, is the subgraph consisting of the unique edge e with $\mathrm{res}(e) = \mathrm{root}_G$ and all nodes in $\mathrm{att}_G(e)$. The unique edge label of a top t is denoted by $\mathrm{lab}(t)$, and Tops_Σ is the set of all tops with function symbols from Σ.

Definition 3 (Precedence). Given a signature Σ, a *precedence* is a transitive relation $\sqsupseteq$ on Tops_Σ such that for all $s, t \in \mathrm{Tops}_\Sigma$, $s \cong t$ implies $s \sqsupseteq t$.

Thus, precedences are preorders satisfying a stronger property than reflexivity. The containment of isomorphism guarantees that precedences are well-quasi-orders whenever Σ is finite. (Reflexivity is not sufficient for this as there are infinitely many isomorphic copies of every top.)

Definition 4 (String embedding). Let $\sqsupseteq$ be a preorder on a set A. The *string embedding* relation $\sqsupseteq^{\mathrm{str}}$ on A^* is defined as follows: $a_1 \ldots a_m \sqsupseteq^{\mathrm{str}} b_1 \ldots b_n$ if $b_1 \ldots b_n$ is empty or if there are $j_1, \ldots, j_n$ such that $1 \leq j_1 < j_2 \ldots < j_n \leq m$ and $a_{j_1} \sqsupseteq b_1, \ldots, a_{j_n} \sqsupseteq b_n$.

Hence, $a \sqsupseteq^{\mathrm{str}} b$ means that b is embedded in a. By Higman's Lemma [7], $\sqsupseteq^{\mathrm{str}}$ is a well-quasi-order on A^* if $\sqsupseteq$ is a well-quasi-order on A.

Definition 5 (Immediate subgraphs). Let G be a term graph and e be the unique edge such that $\mathrm{att}_G(e) = \mathrm{root}_G v_1 \ldots v_n$ for some nodes $v_1, \ldots, v_n$ ($n \geq 0$). Then $G[v_1], \ldots, G[v_n]$ are the *immediate subgraphs* of G and sub_G is the string $G[v_1] \ldots G[v_n]$.

The next definition extends homeomorphic embedding from trees to term graphs (see [4] for a definition of tree embedding).

Definition 6 (Embedding). Let $\sqsupseteq$ be a precedence. The *embedding* relation $\trianglerighteq$ on $\mathcal{TG}_\Sigma$ is defined inductively as follows: $G \trianglerighteq H$ if

(1) $S \trianglerighteq H$ for some immediate subgraph S of G, or
(2) $\mathrm{top}_G \sqsupseteq \mathrm{top}_H$ and $\mathrm{sub}_G \trianglerighteq^{\mathrm{str}} \mathrm{sub}_H$.

It is easy to show that $\trianglerighteq$ is a preorder containing isomorphism of ground term graphs. In order to state Kruskal's Tree Theorem in terms of $\trianglerighteq$, call a term graph G a *tree* if $\mathrm{indegree}(v) = 1$ for each non-root node v.[1]

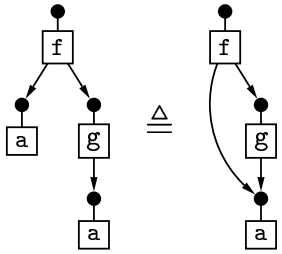
Theorem 7 (Tree Theorem [11]). *Let $\geq$ be a well-quasi-order on Σ and $\sqsupseteq$ be the precedence $\{\langle s, t\rangle \in \mathrm{Tops}^2_\Sigma \mid \mathrm{lab}(s) \geq \mathrm{lab}(t)\}$. Then $\trianglerighteq$ is a well-quasi-order on the set of all trees over Σ.*

Note that the above precedence in general contains pairs with tops that are not in tree form. But the restriction of $\trianglerighteq$ to trees is clearly independent of this part of the precedence.

[1] Given a node v in a term graph G, $\mathrm{indegree}(v)$ is the total number of occurrences of v in the attachment strings of all edges e with $\mathrm{res}(e) \neq v$.

Definition 8. The relations $\triangleq$ and $\rhd$ on $\mathcal{TG}_\Sigma$ are defined as follows:
(1) $G \triangleq H$ if $G \trianglerighteq H$ and $H \trianglerighteq G$.
(2) $G \rhd H$ if $G \trianglerighteq H$ and $H \not\trianglerighteq G$.

Observe that $G \triangleq H$ need not imply that G and H are isomorphic, even with isomorphism as precedence. For example, the following equivalence holds over every precedence:



Now the Tree Theorem is extended to term graphs.

Theorem 9. *Let $\sqsupseteq$ be a precedence that is a well-quasi-order on* Tops_Σ. *Then $\trianglerighteq$ is a well-quasi-order on $\mathcal{TG}_\Sigma$.*

The Tree Theorem is a corollary of this result. For if $\geq$ is a well-quasi-order on Σ, the precedence $\{\langle s,t\rangle \in \mathrm{Tops}^2_\Sigma \mid \mathrm{lab}(s) \geq \mathrm{lab}(t)\}$ is clearly a well-quasi-order on Tops_Σ. With Theorem 9 follows that $\trianglerighteq$ is a well-quasi-order on $\mathcal{TG}_\Sigma$, and hence, in particular, on the set of all trees over Σ.

Theorem 9 can be proved—without difficulties—by the "minimal bad sequence" method used by Nash-Williams for proving the Tree Theorem [13]. Alternatively, Theorem 9 can be proved by the Tree Theorem via an encoding of term graphs as trees. This proof is given below.

Proof of Theorem 9. First, Σ is enlarged to a signature Σ_θ such that there is a bijection between function symbols in Σ_θ and isomorphism classes of tops over Σ. To this end, introduce for every $f \in \Sigma$ and every equivalence relation $\sim$ on $\{1,\ldots,\mathrm{arity}(f)\}$ a function symbol $f_\sim$ with $\mathrm{arity}(f_\sim) = \mathrm{arity}(f)$. Let $\Sigma_\theta = \{f_\sim \mid f \in \Sigma\}$. Now consider any $t \in \mathrm{Tops}_\Sigma$ with $\mathrm{lab}(t) = f$ and string of argument nodes $v_1 \ldots v_n$ $(n \geq 0)$. Define $\theta(t) = f_\sim$, where $\sim$ is the equivalence relation $\{\langle i,j\rangle \mid v_i = v_j\}$ on $\{1,\ldots,n\}$.

Claim: The relation $\geq_\theta = \{\langle \theta(s), \theta(t)\rangle \mid \langle s,t\rangle \in \sqsupseteq\}$ is a wqo on Σ_θ.

Observe first that reflexivity of $\geq_\theta$ follows from reflexivity of $\sqsupseteq$ and surjectivity of the mapping θ. To see that θ is transitive, suppose that $\theta(t_1) \geq_\theta \theta(t_2) = \theta(t_2') \geq_\theta \theta(t_3)$. Then $t_1 \sqsupseteq t_2 \cong t_2' \sqsupseteq t_3$ because θ identifies only isomorphic tops. Hence $t_1 \sqsupseteq t_3$ and $\theta(t_1) \geq_\theta \theta(t_3)$. Finally, since $\sqsupseteq$ is a wqo, surjectivity of θ implies that $\geq_\theta$ is a wqo, too.

Next, θ is extended to a mapping Θ from $\mathcal{TG}_\Sigma$ to the set of trees over Σ_θ as follows: If G is a term graph with $\mathrm{sub}_G = S_1 \ldots S_n$ $(n \geq 0)$, then $\Theta(G)$ is a tree with $\mathrm{lab}(\mathrm{top}_{\Theta(G)}) = \theta(\mathrm{top}_G)$ and $\mathrm{sub}_{\Theta(G)} = \Theta(S_1)\ldots\Theta(S_n)$. (This

defines $\Theta(G)$ uniquely up to isomorphism.) Now consider the precedence $\sqsupseteq_\theta = \{\langle s,t\rangle \in \mathrm{Tops}^2_{\Sigma_\theta} \mid \mathrm{lab}(s) \geq_\theta \mathrm{lab}(t)\}$ and its induced embedding relation $\trianglerighteq_\theta$. By the above claim and the Tree Theorem, $\trianglerighteq_\theta$ is a wqo on the set of all trees over Σ_θ. Moreover, an easy induction on the size of (combined) term graphs shows that for all $G, H \in \mathcal{TG}_\Sigma$, $G \trianglerighteq H$ if and only if $\Theta(G) \trianglerighteq_\theta \Theta(H)$. It follows that $\trianglerighteq$ is a wqo, too. □

The next two lemmas characterize the equivalence $\triangleq$ and the strict part $\triangleright$ of $\trianglerighteq$. Given a string $a = a_1 \ldots a_n$, $|a|$ denotes its length n while, for $i = 1, \ldots, n$, $a[i]$ refers to the element a_i. The relations $\equiv$, $\sqsupset$ and $\triangleright^{\mathrm{str}}$ are defined as follows: $\equiv = (\sqsupseteq \cap \sqsubseteq)$, $\sqsupset = (\sqsupseteq - \sqsubseteq)$ and $\triangleright^{\mathrm{str}} = (\trianglerighteq^{\mathrm{str}} - \trianglelefteq^{\mathrm{str}})$.

Lemma 10. *Let $\sqsupseteq$ be a precedence. Then for all term graphs G and H, $G \triangleq H$ if and only if (1) $\mathrm{top}_G \equiv \mathrm{top}_H$, (2) $|\mathrm{sub}_G| = |\mathrm{sub}_H|$, and (3) $\mathrm{sub}_G[i] \triangleq \mathrm{sub}_H[i]$ for $i = 1, \ldots, |\mathrm{sub}_G|$.*

Lemma 11. *Let $\sqsupseteq$ be a precedence. Then for all term graphs G and H, $G \triangleright H$ if and only if (1) $S \trianglerighteq H$ for some immediate subgraph S of G, or (2) $\mathrm{top}_G \sqsupset \mathrm{top}_H$ and $\mathrm{sub}_G \trianglerighteq^{\mathrm{str}} \mathrm{sub}_H$, or (3) $\mathrm{top}_G \equiv \mathrm{top}_H$ and $\mathrm{sub}_G \triangleright^{\mathrm{str}} \mathrm{sub}_H$.*

4 Simplification Orders

Simplification orders are certain strict orders that contain the strict embedding relation. Theorem 9 guarantees that such orders are well-founded whenever the given precedence is a well-quasi-order.

Definition 12 (Simplification order). Let $\trianglerighteq$ be the embedding relation induced by a precedence that is a well-quasi-order. A transitive relation $\succ$ on $\mathcal{TG}_\Sigma$ is a *simplification order* if it contains $\triangleright$ and if for all $G, H \in \mathcal{TG}_\Sigma$, $G \triangleq H$ implies $G \not\succ H$.

Note that simplification orders are irreflexive, in particular.

Theorem 13. *Every simplification order is well-founded.*

Proof. Let $\succ$ be a simplification order. Then, by Theorem 9, the underlying embedding relation $\trianglerighteq$ is a well-quasi-order. Now suppose that there is an infinite sequence $G_1 \succ G_2 \succ \ldots$ Then there are i and j such that $G_i \trianglelefteq G_j$. On the other hand, $G_i \succ G_{i+1} \succ \ldots \succ G_j$ implies $G_i \succ G_j$ by transitivity of $\succ$. Hence, by the definition of simplification orders, $G_i \triangleq G_j$ is impossible. But then $G_i \triangleleft G_j$ and therefore $G_i \prec G_j$. It follows $G_i \succ G_i$, contradicting the irreflexivity of simplification orders. Thus, $\succ$ is well-founded. □

In order to introduce a recursive path order on term graphs, the lifting of an order to a multiset order is recalled.

Definition 14 (Multiset extension). Let $\succ$ be a strict order on a set A. The *multiset extension* $\succ^{\mathrm{mul}}$ on the set of finite multisets over A is defined as follows: $M \succ^{\mathrm{mul}} N$ if there are multisets X and Y such that

(1) $\emptyset \neq X \subseteq M$,
(2) $N = (M - X) \cup Y$, and
(3) for all $y \in Y$ there is some $x \in X$ with $x \succ y$.

Lemma 15 (Dershowitz and Manna [6]). *If $\succ$ is a strict order on a set A, then $\succ^{\mathrm{mul}}$ is a strict order on the set of finite multisets over A. If $\succ$ is moreover well-founded, then $\succ^{\mathrm{mul}}$ is well-founded, too.*

The equivalence relation $\approx_{\mathrm{rpo}}$ defined next will be used in the definition of the recursive path order.

Definition 16. Let $\sqsupseteq$ be a precedence. The relation $\approx_{\mathrm{rpo}}$ on $\mathcal{TG}_\Sigma$ is defined inductively as follows: $G \approx_{\mathrm{rpo}} H$ if (1) $\mathrm{top}_G \equiv \mathrm{top}_H$, (2) $|\mathrm{sub}_G| = |\mathrm{sub}_H|$, and (3) there is a bijection π on $\{1, \ldots, |\mathrm{sub}_G|\}$ such that $\mathrm{sub}_G[i] \approx_{\mathrm{rpo}} \mathrm{sub}_H[\pi(i)]$ for $i = 1, \ldots, |\mathrm{sub}_G|$.

The equivalence class of a ground term graph G with respect to $\approx_{\mathrm{rpo}}$ is written $[G]$. Given a strict order $\succ$ on $\mathcal{TG}_\Sigma$ such that $G' \approx_{\mathrm{rpo}} G \succ H \approx_{\mathrm{rpo}} H'$ implies $G' \succ H'$, $\succ$ is lifted to an order on equivalence classes as follows: $[G] \succ [H]$ if $G \succ H$. (See [12] for a similar lifting of preorders.) For $G \in \mathcal{TG}_\Sigma$ with $\mathrm{sub}_G = S_1 \ldots S_n$, the multiset $\{[S_1], \ldots, [S_n]\}$ of equivalence classes of immediate subgraphs is denoted by SUB_G.

Definition 17 (Recursive path order). Let $\sqsupseteq$ be a precedence. The *recursive path order* $\succ_{\mathrm{rpo}}$ on $\mathcal{TG}_\Sigma$ is defined inductively as follows: $G \succ_{\mathrm{rpo}} H$ if

(1) $S \succ_{\mathrm{rpo}} H$ or $S \approx_{\mathrm{rpo}} H$ for some immediate subgraph S of G, or
(2) $\mathrm{top}_G \sqsupset \mathrm{top}_H$ and $G \succ_{\mathrm{rpo}} T$ for all immediate subgraphs T of H, or
(3) $\mathrm{top}_G \equiv \mathrm{top}_H$ and $\mathrm{SUB}_G \succ^{\mathrm{mul}}_{\mathrm{rpo}} \mathrm{SUB}_H$.

Lemma 18. *For all $G', G, H, H' \in \mathcal{TG}_\Sigma$, $G' \approx_{\mathrm{rpo}} G \succ_{\mathrm{rpo}} H \approx_{\mathrm{rpo}} H'$ implies $G' \succ_{\mathrm{rpo}} H'$.*

Theorem 19. *The recursive path order is a simplification order whenever the underlying precedence is a well-quasi-order.*

The proof of this result requires to show the three conditions of Definition 12: (1) transitivity of $\succ_{\mathrm{rpo}}$, (2) $\rhd \subseteq \succ_{\mathrm{rpo}}$, and (3) for all $G, H \in \mathcal{TG}_\Sigma$, $G \triangleq H$ implies $G \not\succ_{\mathrm{rpo}} H$. These properties are shown by induction on the size of term graphs, where the induction steps use case distinctions according to the three cases of Definition 17.

As a corollary of Theorem 19, $\succ_{\mathrm{rpo}}$ is well-founded if the given precedence is a well-quasi-order. This can also be shown by using the corresponding result for the recursive path order on terms [4], exploiting the encoding Θ of term graphs as trees given in the proof of Theorem 9. One has to show that for all $G, H \in \mathcal{TG}_\Sigma$, $G \succ_{\mathrm{rpo}} H$ if and only if $\Theta(G) \succ^{\theta}_{\mathrm{rpo}} \Theta(H)$, where $\succ^{\theta}_{\mathrm{rpo}}$ is the recursive path order over the enlarged signature Σ_θ.

5 Termination of Term Graph Rewriting

This section starts with a brief review of the term graph rewriting model investigated in [8,15,16]. In this approach, rewriting includes not only applications of term rewrite rules but also steps for compressing term graphs.

Definition 20 (Collapsing). Given two term graphs G and H, G *collapses* to H if there is a graph morphism $f{:}G \to H$ mapping root_G to root_H. This is denoted by $G \succeq_{\mathcal{C}} H$. The collapsing is *proper*, denoted by $G \succ_{\mathcal{C}} H$, if f is non-injective.

A *term rewrite rule* $l \to r$ consists of two terms l and r such that l is not a variable and all variables in r occur also in l. A set $\mathcal{R}$ of term rewrite rules is a *term rewriting system*. (See [5,10,14] for surveys of term rewriting.)

For every term t, let $\Diamond t$ be a term graph representing t such that only variables are shared.[2] The graph resulting from a term graph G after removing all edges labelled with variables is denoted by $\underline{G}$.

Definition 21 (Instance). A term graph H is an *instance* of a term graph G if there is graph morphism $\underline{G} \to H$ sending root_G to root_H. An instance that is a ground term graph is a *ground instance*.

Definition 22 (Term graph rewriting). Let G and H be term graphs, $l \to r$ be a rewrite rule and v be a node in G such that $G[v]$ is an instance of $\Diamond l$. Then there is a *proper rewrite step* $G \Rightarrow_{v,\, l\to r} H$ if H is isomorphic to the term graph G_3 constructed as follows:

(1) $G_1 = G - \{e\}$ is the graph obtained from G by removing the unique edge e satisfying $\mathrm{res}(e) = v$.
(2) G_2 is the graph obtained from the disjoint union $G_1 + \underline{\Diamond r}$ by
   - identifying v with $\mathrm{root}_{\Diamond r}$,
   - identifying the image of $\mathrm{res}(e_1)$ with $\mathrm{res}(e_2)$, for each pair $\langle e_1, e_2\rangle \in \mathrm{E}_{\Diamond l} \times \mathrm{E}_{\Diamond r}$ with $\mathrm{lab}_{\Diamond l}(e_1) = \mathrm{lab}_{\Diamond r}(e_2) \in X$.
(3) $G_3 = G_2[\mathrm{root}_G]$ is the term graph obtained from G_2 by removing all nodes and edges not reachable from root_G ("garbage collection").

Now the term graph rewrite relation $\Rightarrow_{\mathcal{R}}$ on $\mathcal{TG}_{\Sigma,X}$ is defined by adding proper collapse steps: $G \Rightarrow_{\mathcal{R}} H$ if $G \succ_{\mathcal{C}} H$ or $G \Rightarrow_{l\to r} H$ for some rule $l \to r$ in $\mathcal{R}$. The relation $\Rightarrow_{\mathcal{R}}$ is *terminating* if no infinite sequence $G_1 \Rightarrow_{\mathcal{R}} G_2 \Rightarrow_{\mathcal{R}} \ldots$ exists.

Definition 23. A precedence $\sqsupseteq$ is *collapse compatible* if whenever there is a graph morphism $t \to u$ between two tops $t, u \in \mathrm{Tops}_{\Sigma}$, then $t \sqsupseteq u$. A collapse compatible precedence that is a well-quasi-order is a *well-precedence*.

Lemma 24. *Let $\sqsupseteq$ be a precedence. The embedding relation $\trianglerighteq$ contains the collapse relation $\succeq_{\mathcal{C}}$ if and only if $\sqsupseteq$ is collapse compatible.*

[2] That is, $\mathrm{indegree}(\mathrm{res}(e)) \leq 1$ for each edge e with $\mathrm{lab}_{\Diamond t}(e) \notin X$, and $e_1 = e_2$ for all edges e_1,e_2 with $\mathrm{lab}_{\Diamond t}(e_1) = \mathrm{lab}_{\Diamond t}(e_2) \in X$.

Theorem 25. *Let $\succ_{\mathrm{rpo}}$ be induced by a well-precedence. Then $\Rightarrow_{\mathcal{R}}$ is terminating if $G \Rightarrow_{l \to r} H$ implies $G \succ_{\mathrm{rpo}} H$, for every rule $l \to r$ in $\mathcal{R}$ and all ground term graphs G and H.*

Proof. It suffices to show the absence of infinite derivations over $\mathcal{TG}_{\Sigma}$, since all occuring variables can be replaced by a constant. Suppose that there is an infinite sequence $G_1 \Rightarrow_{\mathcal{R}} G_2 \Rightarrow_{\mathcal{R}} \ldots$ over $\mathcal{TG}_{\Sigma}$. As proper collapsing is terminating, there are $i_1, i_2, \ldots$ such that $1 = i_1 \leq i_2 < i_3 \leq i_4 < \ldots$ and $G_{i_1} \succeq_{\mathcal{C}} G_{i_2} \Rightarrow_{\mathcal{R}} G_{i_3} \succeq_{\mathcal{C}} G_{i_4} \Rightarrow_{\mathcal{R}} \ldots$, where all $\Rightarrow_{\mathcal{R}}$-steps are proper rewrite steps. By the assumption and Lemma 24, this implies $G_{i_1} \trianglerighteq G_{i_2} \succ_{\mathrm{rpo}} G_{i_3} \trianglerighteq G_{i_4} \succ_{\mathrm{rpo}} \ldots$ As $\succ_{\mathrm{rpo}}$ is a simplification order, $\trianglerighteq$ is contained in $\succ_{\mathrm{rpo}} \cup \approx_{\mathrm{rpo}}$. With Lemma 18 follows that there is an infinite subsequence $G_{j_1} \succ_{\mathrm{rpo}} G_{j_2} \succ_{\mathrm{rpo}} \ldots$ of $G_1 \Rightarrow_{\mathcal{R}} G_2 \Rightarrow_{\mathcal{R}} \ldots$ But $\succ_{\mathrm{rpo}}$ is well-founded by Theorems 19 and 13, a contradiction. Thus $\Rightarrow_{\mathcal{R}}$ is terminating. □

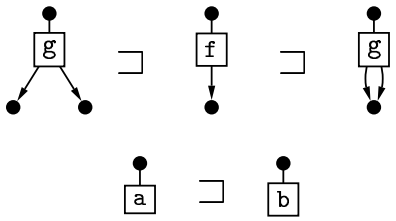
Due to a monotonicity property of $\succ_{\mathrm{rpo}}$, the premise of Theorem 25 can be weakened.

Theorem 26. *Let $\succ_{\mathrm{rpo}}$ be induced by a well-precedence. Then $\Rightarrow_{\mathcal{R}}$ is terminating if $L \Rightarrow_{\mathrm{root}_L,\, l \to r} R$ implies $L \succ_{\mathrm{rpo}} R$, for every rule $l \to r$ in $\mathcal{R}$ and every ground instance L of $\Diamond l$.*

Example 27. Consider the following rewrite system $\mathcal{R}$:

$$\begin{aligned} \mathtt{f(x)} &\to \mathtt{g(x,x)} \\ \mathtt{a} &\to \mathtt{b} \\ \mathtt{g(a,b)} &\to \mathtt{f(a)} \end{aligned}$$

This system is non-terminating under term rewriting because there is an infinite rewrite sequence $\mathtt{f(a)} \to \mathtt{g(a,a)} \to \mathtt{g(a,b)} \to \mathtt{f(a)} \to \ldots$ Termination of term graph rewriting can easily be checked by means of Theorem 26, using the following well-precedence:



Acknowledgement. The author is grateful to Annegret Habel and Andreas Weiermann, who gave valuable comments on a previous version of this paper.

References

1. Zena M. Ariola and Jan Willem Klop. Equational term graph rewriting. *Fundamenta Informaticae*, 26:207–240, 1996.

2. Andrea Corradini and Francesca Rossi. Hyperedge replacement jungle rewriting for term rewriting systems and logic programming. *Theoretical Computer Science*, 109:7–48, 1993.
3. Nachum Dershowitz. Orderings for term rewriting systems. *Theoretical Computer Science*, 17:279–301, 1982.
4. Nachum Dershowitz. Termination of rewriting. *Journal of Symbolic Computation*, 3:69–116, 1987.
5. Nachum Dershowitz and Jean-Pierre Jouannaud. Rewrite systems. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science,* volume B, chapter 6. Elsevier, 1990.
6. Nachum Dershowitz and Zohar Manna. Proving termination with multiset orderings. *Communications of the ACM*, 22(8):465–476, 1979.
7. Graham Higman. Ordering by divisibility in abstract algebras. *Proceedings of the London Mathematical Society*, 3(2):326–336, 1952.
8. Berthold Hoffmann and Detlef Plump. Implementing term rewriting by jungle evaluation. *RAIRO Theoretical Informatics and Applications*, 25(5):445–472, 1991.
9. Richard Kennaway, Jan Willem Klop, Ronan Sleep, and Fer-Jan de Vries. On the adequacy of term graph rewriting for simulating term rewriting. *ACM Transactions on Programming Languages and Systems*, 16(3):493–523, 1994.
10. Jan Willem Klop. Term rewriting systems. In S. Abramsky, Dov M. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science,* volume 2, pages 1–116. Oxford University Press, 1992.
11. Joseph B. Kruskal. Well-quasi-ordering, the Tree Theorem, and Vazsonyi's conjecture. *Transactions of the American Mathematical Society*, 95:210–225, 1960.
12. Aart Middeldorp and Hans Zantema. Simple termination revisited. In *Proc. 12th International Conference on Automated Deduction*, volume 814 of *Lecture Notes in Artificial Intelligence*, pages 451–465. Springer-Verlag, 1994.
13. C. St. J. A. Nash-Williams. On well-quasi-ordering finite trees. *Proceedings of the Cambridge Philosophical Society*, 59:833–835, 1963.
14. David A. Plaisted. Equational reasoning and term rewriting systems. In Dov M. Gabbay, C.J. Hogger, and J.A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming,* volume 1, pages 273–364. Clarendon Press, 1993.
15. Detlef Plump. Collapsed tree rewriting: Completeness, confluence, and modularity. In *Proc. Conditional Term Rewriting Systems*, volume 656 of *Lecture Notes in Computer Science*, pages 97–112. Springer-Verlag, 1993.
16. Detlef Plump. Evaluation of functional expressions by hypergraph rewriting. Dissertation, Universität Bremen, Fachbereich Mathematik und Informatik, 1993.
17. Ronan Sleep, Rinus Plasmeijer, and Marko van Eekelen, editors. *Term Graph Rewriting: Theory and Practice.* John Wiley, 1993.
18. Joachim Steinbach. Simplification orderings — history of results. *Fundamenta Informaticae*, 24:47–87, 1995.