Proving Termination of Unfolding Graph Rewriting for General Safe Recursion

Naohi Eguchi*
Institute of Computer Science, University of Innsbruck, Austria
naohi.eguchi@uibk.ac.at

We present a new termination proof and complexity analysis of *unfolding graph rewriting* which is a specific kind of infinite graph rewriting expressing the general form of *safe recursion*. We introduce a termination order over sequences of terms together with an interpretation of term graphs into sequences of terms. Unfolding graph rewrite rules expressing the general safe recursion can be embedded into the termination order by the interpretation, yielding the polynomial runtime complexity.

1 Introduction

In this paper we present a new termination proof and complexity analysis of a specific kind of infinite graph rewriting called *unfolding graph rewriting* [7]. The formulation of unfolding graph rewriting stems from a function-algebraic characterisation of the polytime computable functions based on the principle known as *safe recursion* [6] or *tiered recursion* [9]. Safe recursion is a syntactic restriction of the standard primitive recursion based on a specific separation of argument positions of functions into two kinds. Notationally, the separation is indicated by semicolon as $f(x_1,\dots,x_k\,;x_{k+1},\dots,x_{k+l})$, where $x_1,\dots,x_k$ are called *normal* arguments while $x_{k+1},\dots,x_{k+l}$ are called *safe* ones. The schema of safe recursion formalises the idea that recursive calls are restricted on normal arguments whereas substitution of recursion terms is restricted for safe arguments: $f(0,\vec{y};\vec{z}) = g(\vec{y};\vec{z})$, $f(c_i(x),\vec{y};\vec{z}) = h_i(x,\vec{y};\vec{z},f(x,\vec{y};\vec{z}))$ $(i \in I)$, where I is a finite set of indices. As discussed in [7], safe recursion is sound for polytime computability over unary constructor, i.e., over numerals or lists, but it was not clear whether general forms of safe recursion over arbitrary constructors, which is called *general ramified recurrence* [7] or (**General Safe Recursion**), could be related to polynomial complexity.

$$f(c_i(x_1,\dots,x_{\mathsf{arity}(c_i)}),\vec{y};\vec{z}) = h_i(\vec{x},\vec{y};\vec{z},f(x_1,\vec{y};\vec{z}),\dots,f(x_{\mathsf{arity}(c_i)},\vec{y};\vec{z}))\ (i \in I) \quad \textbf{(General Safe Recursion)}$$

To see the difficulty of this question, consider a term rewrite system (TRS for short) $\mathscr{R}$ over the constructors $\{\varepsilon,\mathsf{c},\mathsf{0},\mathsf{s}\}$ consisting of the following rules with the argument separation indicated in the rules.

$$\mathsf{g}(\varepsilon\,;z) \to z \quad \mathsf{g}(\mathsf{c}(\,;x,y)\,;z) \to \mathsf{c}(\,;\mathsf{g}(x;z),\mathsf{g}(y;z)) \quad \mathsf{f}(\mathsf{0},y\,;) \to \varepsilon \quad \mathsf{f}(\mathsf{s}(\,;x),y\,;) \to \mathsf{g}(y\,;\mathsf{f}(x,y\,;))$$

Under a natural interpretation, $\mathsf{g}(x,y)$ generates the binary tree appending the tree y to every leaf of the tree x, and $\mathsf{f}(\mathsf{s}^m(\mathsf{0}),x)$ generates a tree consisting of exponentially many copies of the tree x measured by m. Namely, rewriting in the TRS $\mathscr{R}$ results in normal forms of exponential size measured by the size of starting terms. This problem cannot be solved by simple sharing. The authors of [7] solved this problem, showing that the equation of general safe recursion can be expressed by an infinite set of unfolding graph rewrite rules. In the present work, we propose complexity analysis by means of termination orders over sequences of terms (Section 3) together with a successful embedding (Section 4), sharpening the complexity result obtained in [7] (Corollary 1). Missing details can be found in a technical report [8].

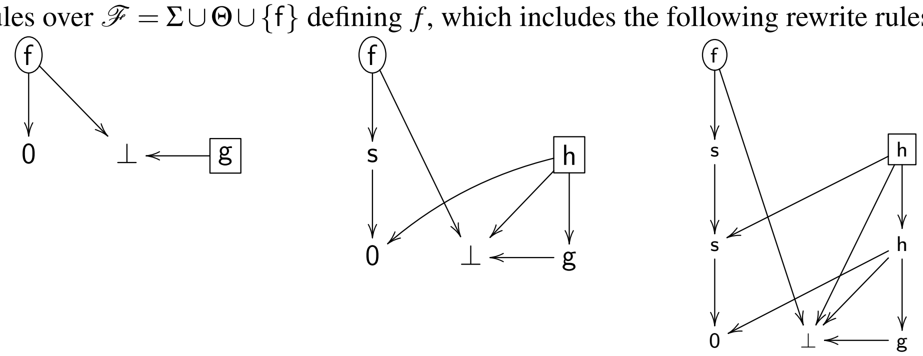
*The author is supported by Grant-in-Aid for JSPS Fellows (Grant No. $25 \cdot 726$) that is granted at Graduate School of Science, Chiba University, Japan.

2 Unfolding graph rewrite rules for general safe recursion

In this section we specify the shape of unfolding graph rewrite rules which is compatible with the schema of (**General Safe Recursion**). We present basics of term graph rewriting following [5]. Let $\mathscr{F}$ be a *signature*, a finite set of function symbols, and let $\mathsf{arity} : \mathscr{F} \to \mathbb{N}$ where $\mathsf{arity}(f)$ is called the *arity* of f. We assume that $\mathscr{F}$ is partitioned into the set $\mathscr{C}$ of constructors and the set $\mathscr{D}$ of defined symbols. A *labeled graph* is a triple $(G, \mathsf{lab}_G, \mathsf{succ}_G)$ of an acyclic directed graph $G = (V_G, E_G)$, a partial *labeling* function $\mathsf{lab}_G : V_G \to \mathscr{F}$ and a (total) *successor* function $\mathsf{succ}_G : V_G \to V_G^*$, mapping a node $v \in V_G$ to a sequence of nodes of length $\mathsf{arity}(\mathsf{lab}_G)$. In case $\mathsf{succ}_G(v) = v_1, \dots, v_k$, the node v_j is called the *jth successor of v* for every $j \in \{1, \dots, k\}$. A labeled graph $(G, \mathsf{lab}_G, \mathsf{succ}_G)$ is *closed* if the labeling function lab_G is total. Given two labeled graphs G and H, a *homomorphism* from G to h is a mapping $\varphi : V_G \to V_H$ such that $\mathsf{lab}_H(\varphi(v)) = \mathsf{lab}_G(v)$ for each $v \in \mathrm{dom}(\mathsf{lab}_G) \subseteq V_G$, and for each $v \in \mathrm{dom}(\mathsf{lab}_G)$, if $\mathsf{succ}_G(v) = v_1, \dots, v_k$, then $\mathsf{succ}_H(\varphi(v)) = \varphi(v_1), \dots, \varphi(v_k)$. A quadruple $(G, \mathsf{lab}_G, \mathsf{succ}_G, \mathsf{root}_G)$ is a *term graph* if $(G, \mathsf{lab}_G, \mathsf{succ}_G)$ is a labeled graph and root_G is the *root* of G. We write $\mathscr{TG}(\mathscr{F})$ to denote the set of term graphs over a signature $\mathscr{F}$. For a labeled graph $G = (G, \mathsf{succ}_G, \mathsf{lab}_G)$ and a node $v \in V_G$, $G \restriction v$ denotes the sub-term graph of G rooted at v. A homomorphism φ from a term graph G to another term graph H is a homomorphism φ such that $\mathsf{root}_H = \varphi(\mathsf{root}_G)$. A *graph rewrite rule* is a triple $\rho = (G, l, r)$ of a labeled graph G and distinct two nodes l and r respectively called the *left* and *right* root. A *redex* in a term graph G is a pair (R, φ) of a rewrite rule $R = (H, l, r)$ and a homomorphism $\varphi : H \restriction l \to G$. A set $\mathscr{G}$ of graph rewrite rules is called a *graph rewrite system* (GRS for short). A graph rewrite rule (G, l, r) is called a *constructor* one if $\mathsf{lab}_G(l) \in \mathscr{D}$ and $\mathsf{lab}_G(v) \in \mathscr{C}$ for any $v \in V_{G \restriction l} \setminus \{l\}$ whenever $\mathsf{lab}_G(v)$ is defined. A GRS is called a constructor one if it consists only of constructor rewrite rules. The rewrite relation defined by a GRS $\mathscr{G}$ is denoted as $\to_{\mathscr{G}}$ and its m-fold iteration is denoted as $\to_{\mathscr{G}}^m$. The corresponding *innermost* rewrite relations $\xrightarrow{\mathrm{i}}_{\mathscr{G}}$ and $\xrightarrow{\mathrm{i}}{}_{\mathscr{G}}^m$ are defined accordingly.

Definition 1 (Unfolding graph rewrite rules [7]) Let Σ and Θ be two disjoint signatures in bijective correspondence by $\varphi : \Sigma \to \Theta$. For a fixed $k \in \mathbb{N}$, suppose that $\mathsf{arity}(\varphi(g)) = 2\mathsf{arity}(g) + k$ for each $g \in \Sigma$. Let $f \notin \Sigma \cup \Theta$ be a fresh function symbol such that $\mathsf{arity}(f) = 1 + k$. An *unfolding graph rewrite rule over* Σ *and* Θ *defining* f is a graph rewrite rule $\rho = (G, l, r)$ where $G = (V_G, E_G, \mathsf{succ}_G, \mathsf{lab}_G)$ is a labeled graph over a signature $\mathscr{F} \supseteq \Sigma \cup \Theta$, for the set V_G of vertices consists of $1 + 2m + k$ elements y, $v_1, \dots, v_m$, $w_1, \dots$, w_m, $x_1, \dots, x_k$, that fulfills the following conditions: (i) $l = y$, $r = w_1$, $\mathsf{lab}_G(y) = f$, $\mathsf{succ}_G(y) = v_1, x_1, \dots, x_k$, and $\mathsf{lab}_G(x_j)$ is undefined for all $j \in \{1, \dots, k\}$. (ii) $V_{G \restriction v_1} = \{v_1, \dots, v_m\}$, and, for each $j \in \{1, \dots, m\}$, $\mathsf{lab}_G(v_j) \in \Sigma$ and $\mathsf{succ}_G(v_j) \in \{v_1, \dots, v_m\}^*$. (iii) For each $j \in \{1, \dots, m\}$, $\mathsf{lab}_G(w_j) = \varphi(\mathsf{lab}_G(v_j))$, and $\mathsf{succ}_G(w_j) = v_{j_1}, \dots, v_{j_n}, x_1, \dots, x_k, w_{j_1}, \dots, w_{j_n}$ if $\mathsf{succ}_G(v_j) = v_{j_1}, \dots, v_{j_n}$.

Example 1 Let $\Sigma = \{0, \mathsf{s}\}$, $\Theta = \{\mathsf{g}, \mathsf{h}\}$, $\varphi : \Sigma \to \Theta$ be a bijection defined as $0 \mapsto g$ and $\mathsf{s} \mapsto \mathsf{h}$, and $\mathsf{f} \notin \Sigma \cup \Theta$, where the arities of $0, \mathsf{s}, \mathsf{g}, \mathsf{h}, \mathsf{f}$ are respectively 0, 1, 1, 3 and 2. The equations $\mathsf{f}(0, x) \to \mathsf{g}(x)$, $\mathsf{f}(\mathsf{s}(y), x) \to \mathsf{h}(y, x, \mathsf{f}(y, x))$ for primitive recursion can be expressed by the infinite set of unfolding graph rewrite rules over $\mathscr{F} = \Sigma \cup \Theta \cup \{\mathsf{f}\}$ defining f, which includes the following rewrite rules.



In the examples, the left root is written in a circle while the right root is in a square. Undefined nodes are indicated as $\perp$. As seen from the pictures, the unfolding graph rewrite rules express the infinite instances $\mathsf{f}(0,x) \to \mathsf{g}(x)$, $\mathsf{f}(\mathsf{s}(0),x) \to \mathsf{h}(0,x,\mathsf{g}(x))$, $\mathsf{f}(\mathsf{s}(\mathsf{s}(0)),x) \to \mathsf{h}(\mathsf{s}(0),x,\mathsf{h}(0,x,\mathsf{g}(x)))$, $\ldots$.

In [7] a GRS $\mathscr{G}$ is called *polytime presentable* if there exists a deterministic polytime algorithm which, given a term graph G, returns a term graph H such that $G \xrightarrow{\mathsf{i}}_{\mathscr{G}} H$ if such a term graph exists, or the value false if otherwise. A GRS $\mathscr{G}$ is *polynomially bounded* if there exists a polynomial p such that $\max\{m, |H|\} \leq p(|G|)$ holds whenever $G \xrightarrow{\mathsf{i}}{}^{m}_{\mathscr{G}} H$ holds. The main result in [7] is restated as follows.

Theorem 1 (Dal Lago, Martini and Zorzi [7]) *Every general safe recursive function can be represented by a polytime presentable and polynomially bounded constructor GRS.*

In the proof, the case that the function is defined by (**General Safe Recursion**) is witnessed by an infinite set of unfolding graph rewrite rules in a specific shape compatible with the argument separation as indicated in the schema (**General Safe Recursion**). According to the idea of safe recursion, we assume that the argument positions of every function symbol are separated into the normal and safe ones, writing $f(x_1,\ldots,x_k\,;x_{k+1},\ldots,x_{k+l})$ to denote k normal and l safe arguments. We take the argument separation into labeled graphs in such a way that for every successor u of a node v we write $u \in \mathsf{nrm}(v)$ if u is connected to a normal argument position of $\mathsf{lab}_G(v)$, and $u \in \mathsf{safe}(v)$ if otherwise. Notationally, we write $\mathsf{succ}_G(v) = v_1,\ldots,v_k\,;v_{k+1},\ldots,v_{k+l}$ to express the separation that $v_1,\ldots,v_k \in \mathsf{nrm}(v)$ and $v_{k+1},\ldots,v_{k+l} \in \mathsf{safe}(v)$. We assume that any homomorphism preserves the argument separation.

Definition 2 (Safe recursive unfolding graph rewrite rules) We call an unfolding graph rewrite rule *safe recursive* if the following constraints imposed on the clause (i) and (iii) in Definition 1 are satisfied.
(a) In the clause (i), $v_1 \in \mathsf{nrm}(y)$, and in the clause (iii), $v_{j_1},\ldots,v_{j_n} \in \mathsf{nrm}(w_j)$ and $w_{j_1},\ldots,w_{j_n} \in \mathsf{safe}(w_j)$.
(b) For each $j \in \{1,\ldots,k\}$, $x_j \in \mathsf{nrm}(y)$ if and only if $x_j \in \mathsf{nrm}(w_i)$ for all $i \in \{1,\ldots,m\}$.

3 Termination orders on sequences of terms

In this section we consider a termination order $>_\ell$ indexed by a positive natural ℓ over sequences of terms, which is essentially the same as *small polynomial path orders on sequences* [3] but without recursive comparison. It can be shown that, for any fixed ℓ, the length of any $>_\ell$-reduction sequence can be linearly bounded measured by the size of a starting term but polynomially bounded if measured by ℓ. Let $\mathscr{F} = \mathscr{C} \cup \mathscr{D}$ be a signature. The set of terms over $\mathscr{F}$ (and the set $\mathscr{V}$ of variables) is denoted as $\mathscr{T}(\mathscr{F},\mathscr{V})$, and the set of closed terms is denoted as $\mathscr{T}(\mathscr{F})$. We write $s \rhd t$ to express that s is a *proper super-term* of t. A *precedence* $>$ is a well founded partial binary relation on $\mathscr{F}$. The *rank* $\mathsf{rk} : \mathscr{F} \to \mathbb{N}$ is defined to be compatible with $>$: $\mathsf{rk}(f) > \mathsf{rk}(g) \Leftrightarrow f > g$. To form sequences of terms, assume an auxiliary function symbol $\circ$ whose arity is finite but arbitrary. A term of the form $\circ(t_1,\ldots,t_k)$ will be called a sequence if $t_1,\ldots,t_k \in \mathscr{T}(\mathscr{F},\mathscr{V})$, denoted as $[t_1\ \cdots\ t_k]$. We will write $a,b,c,\ldots$ for both terms and sequences. We also write $[s_1\ \cdots\ s_k]^\frown[t_1\ \cdots\ t_l]$ to denote the concatenation $[s_1\ \cdots\ s_k\ t_1\ \cdots\ t_l]$.

Definition 3 Let $>$ be a precedence on a signature $\mathscr{F}$. Suppose that $\ell \in \mathbb{N}$ and $1 \leq \ell$. Then $a >_\ell b$ holds if one of the following three cases holds:
(i) $a = f(s_1,\ldots,s_k)$, $b = g(t_1,\ldots,t_l)$, $f,g \in \mathscr{F}$, $f > g$, $f(s_1,\ldots,s_k) \rhd t_j$ for all $j \in \{1,\ldots,k\}$, and $l \leq \ell$.
(ii) $a = f(s_1,\ldots,s_k)$, $f \in \mathscr{F}$, $b = [t_1 \cdots t_l]$, $f(s_1,\ldots,s_k) >_\ell t_j$ for all $j \in \{1,\ldots,l\}$, and $l \leq \ell$.
(iii) $a = [s_1 \cdots s_k]$, $b = [t_1 \cdots t_l]$ and there exists a permutation $\pi : \{1,\ldots,l\} \to \{1,\ldots,l\}$, and there exist terms or sequences b_j $(j = 1,\ldots,k)$ such that $b_1{}^\frown \cdots {}^\frown b_k = [t_{\pi(1)} \cdots t_{\pi(l)}]$, $s_j \geqslant_\ell b_j$ for all $j \in \{1,\ldots,k\}$, and $s_i >_\ell b_i$ for some $i \in \{1,\ldots,k\}$. In case some b_i is a term t, the concatenation $\cdots{}^\frown b_i{}^\frown \cdots$ should be understood as $\cdots{}^\frown [t]^\frown \cdots$.

Definition 4 $\mathsf{G}_\ell(a) := \max\{k \in \mathbb{N} \mid \exists a_1,\dots,a_k$ *such that* $a >_\ell a_1 >_\ell \cdots >_\ell a_k\}$.

Lemma 1 *Let* $\ell \geq 1$ *and* $\max\{\mathsf{arity}(f) \mid f \in \mathscr{F}\} \leq d$. *Then, for any function symbol* $f \in \mathscr{F}$ *with arity* $k \leq \ell$ *and for any closed terms* $s_1,\dots,s_k \in \mathscr{T}(\mathscr{C})$, $\mathsf{G}_\ell(f(s_1,\dots,s_k)) \leq d^{\mathsf{rk}(f)} \cdot (1+\ell)^{\mathsf{rk}(f)} \cdot \left(1+\sum_{j=1}^{k}\mathsf{dp}(s_j)\right)$ *holds, where* $\mathsf{dp}(t)$ *denotes the* depth *of a term t in the standard tree representation.*

4 Predicative embedding of safe recursive unfolding graph rewriting

In this section we present the *predicative* interpretation of term graphs into sequences of terms, showing that, by the interpretation, rewriting sequences by safe recursive unfolding graph rewrite rules can be embedded into the termination order $>_\ell$ presented in the previous section. This yields that the length of any rewriting sequence by safe recursive unfolding graph rewrite rules starting with a term graphs whose arguments are already normalised can be bounded by a polynomial in the sizes of the normal argument subgraphs only, sharpening the complexity result obtained in [7]. The predicative interpretation is defined modifying the predicative interpretations for terms employed in [1, 4, 2, 3].

Definition 5 (Interpretation of term graphs into unlabeled graphs) A list $\langle v_1,m_1,\dots,v_{k-1},m_{k-1},v_k\rangle$ consisting of nodes $v_1,\dots,v_m$ of a term graph G and naturals $m_1,\dots,m_{k-1}$ is called a *path* from v_1 to v_k if v_{j+1} is the m_jth successor of v_j for each $j \in \{1,\dots,k-1\}$. We call a path $\langle v_1,m_1,\dots,v_{k-1},m_{k-1},v_k\rangle$ in a term graph G a *safe* one if $v_{j+1} \in \mathsf{safe}(v_j)$ for all $j \in \{1,\dots,k-1\}$.

To define the predicative interpretation, we define an auxiliary interpretation $\mathscr{J}$ of term graphs into *unlabeled graphs*. For a term graph G, $\mathscr{J}(G)$ denotes the directed graph $(V_{\mathscr{J}(G)}, E_{\mathscr{J}(G)})$ with the root $\mathsf{root}_{\mathscr{J}(G)} = \mathsf{root}_G$ consisting of the set $V_{\mathscr{J}(G)} = V_G$ of vertices, and the set $E_{\mathscr{J}(G)}$ of edges defined as follows. For an edge $(u,v) \in E_G$, $(u,v) \in E_{\mathscr{J}(G)}$ holds if either (i) or (ii) holds.

(i) There are no distinct two safe paths from root_G to v.

(ii) The edge (u,v) lies on a safe path $\langle u_1,m_1,\dots,u_{k-1},m_{k-1},v\rangle$ from root_G to v, i.e., $u_1 = \mathsf{root}_G$ and $u_{k-1} = u$, and, for any distinct safe path $\langle v_1,n_1,\dots,v_{l-1},n_{l-1},v\rangle$ from root_G to v, $m_i < n_j$ holds whenever $u_i = v_j$ and $m_i \neq n_j$. Namely, a safe path is kept by the interpretation $\mathscr{J}$ only if it is the leftmost one.

For each symbol $f \in \mathscr{F}$ with k normal argument positions, let f_{n} denote a fresh function symbol with $\mathsf{arity}(f_{\mathsf{n}}) = k$. For a term graph G, we write $\mathsf{term}(G)$ to denote the standard term representation of G. For two successors v_0, v_1 of a node v, we write $v_0 < v_1$ if v_j is the k_jth successor for each $j \in \{0,1\}$ and $k_0 < k_1$. We extend the notation $G \restriction v$ to unlabeled (acyclic) directed graphs in the most natural way.

Definition 6 (Predicative interpretation) For a closed term graph G over a signature $\mathscr{F} = \mathscr{C} \cup \mathscr{D}$, let $f = \mathsf{lab}_G(\mathsf{root}_G)$ and $\mathsf{succ}_G(\mathsf{root}_G) = v_1,\dots,v_k;v_{k+1},\dots,v_{\mathsf{arity}(f)}$. Suppose that $\{u_1,\dots,u_n\} = \{v \in V_G \mid v \in \mathsf{safe}(\mathsf{root}_G)$ and $(\mathsf{root}_G, v) \in E_{\mathscr{J}(G)}\}$ and $u_1 < \cdots < u_n$. Then, $\mathscr{I}(G) := [\,]$ (the empty sequence) if $G \in \mathscr{TG}(\mathscr{C})$, or otherwise $\mathscr{I}(G) := [f_{\mathsf{n}}(\mathsf{term}(G \restriction v_1),\dots,\mathsf{term}(G \restriction v_k))]^\frown \mathscr{I}(G \restriction u_1)^\frown \cdots {}^\frown \mathscr{I}(G \restriction u_n)$.

For a signature $\mathscr{F} = \mathscr{C} \cup \mathscr{D}$, we define a subset $\mathscr{TG}_{\mathsf{nrm}}(\mathscr{F}) \subseteq \mathscr{TG}(\mathscr{F})$. Let $G \in \mathscr{TG}(\mathscr{F})$ with $\mathsf{succ}_G(\mathsf{root}_G) = v_1,\dots,v_k;v_{k+1},\dots,v_{k+l}$. Then $G \in \mathscr{TG}_{\mathsf{nrm}}(\mathscr{F})$ if either $G \in \mathscr{TG}(\mathscr{C})$, or $G \restriction v_j \in \mathscr{TG}(\mathscr{C})$ for each $j \in \{1,\dots,k\}$ and $G \restriction v_j \in \mathscr{TG}_{\mathsf{nrm}}(\mathscr{F})$ for each $j \in \{k+1,\dots,k+l\}$. In addition, G is called *basic* if $\mathsf{lab}_G(\mathsf{root}_G) \in \mathscr{D}$ and $G \restriction v_j \in \mathscr{TG}(\mathscr{C})$ for all $j \in \{1,\dots,k+l\}$.

Lemma 2 *Let* $\mathscr{G}$ *be a set of constructor safe recursive unfolding graph rewrite rules over a signature* $\mathscr{F}$. *Suppose that* $G \to_{\mathscr{G}} H$ *is induced by a redex* (R,φ) *in a closed basic term graph* $G \in \mathscr{TG}_{\mathsf{nrm}}(\mathscr{F})$ *for a rule* $R = (G',l,r) \in \mathscr{G}$ *and a homomorphism* $\varphi : G' \restriction l \to G$. *Let* $r' \in V_H$ *denote the node corresponding to* $r \in V_{G'}$. *Then,* $\mathscr{I}(G \restriction \varphi(l)) >_\ell \mathscr{I}(H \restriction r')$ *holds for* $\ell = \max(\{|G' \restriction r|\} \cup \{\mathsf{arity}(f) \mid f \in \mathscr{F}\})$.

Theorem 2 *Let $\mathscr{G}$ be a set of constructor safe recursive unfolding graph rewrite rules over a signature $\mathscr{F}$. Suppose that $\max\{\mathsf{arity}(\mathsf{f}) \mid \mathsf{f} \in \mathscr{F}\} \leq d$ and that $G_0 \in \mathscr{TG}(\mathscr{F})$ is a closed basic term graph such that $\mathsf{succ}_{G_0}(\mathsf{root}_{G_0}) = v_1, \ldots, v_k ; v_{k+1}, \ldots, v_{k+l}$. Then, in any $\mathscr{G}$ rewriting starting with G_0, if $G \to_{\mathscr{G}} H$, then $\mathscr{I}(G) >_\ell \mathscr{I}(H)$ holds for $\ell = 2|\bigcup_{j=1}^{k} V_{G_0 \restriction v_j}| + d$.*

The following corollary is a consequence of Lemma 1 and Theorem 2.

Corollary 1 *For any set $\mathscr{G}$ of constructor safe recursive unfolding graph rewrite rules over a signature $\mathscr{F}$, there exists a polynomial p such that, for any closed basic term graph $G \in \mathscr{TG}(\mathscr{F})$ such that $\mathsf{succ}_{G}(\mathsf{root}_{G}) = v_1, \ldots, v_k ; v_{k+1}, \ldots, v_{k+l}$, if $G \to_{\mathscr{G}}^{m} H$ for some H, then $m \leq p(|\bigcup_{j=1}^{k} V_{G \restriction v_j}|)$ holds.*

In contrast to Theorem 1, the upper bound $p(|\bigcup_{j=1}^{k} V_{G \restriction v_j}|)$ depends only on the size $|\bigcup_{j=1}^{k} V_{G \restriction v_j}|$ (of the union) of the subgraphs connected to the normal argument positions. Moreover, innermost rewriting is not assumed as long as rewriting starts with a (closed) basic term graph. In this paper, every GRS $\mathscr{G}$ is restricted to a set of unfolding graph rewrite rules, but the restriction can be relaxed so that $\mathscr{G}$ contains a finite number of additional graph rewrite rules in certain shapes ([8, Section 6]).

5 Conclusion

Motivated by former works [1, 4, 2, 3], we introduced a termination order over sequences of terms together with an interpretation of term graphs. Unfolding graph rewrite rules expressing the equation of (**General Safe Recursion**) can be embedded into the termination order by the interpretation, which enables us to sharpen the result obtained in [7] about the runtime complexity of those unfolding graph rewrite rules. All the results presented in this paper have been obtained in the technical report [8] by the author. For further investigation, it would be natural to look into the possibility of new criteria for the polynomial runtime complexity of infinite graph rewriting based on the current approach.

References

[1] T. Arai & G. Moser (2005): *Proofs of Termination of Rewrite Systems for Polytime Functions*. In: *Proceedings of the 25th FSTTCS*, *LNCS* 3821, pp. 529–540.

[2] M. Avanzini, N. Eguchi & G. Moser (2011): *A Path Order for Rewrite Systems that Compute Exponential Time Functions*. In: *Proceedings of the 22nd RTA*, *LIPIcs* 10, pp. 123–138.

[3] M. Avanzini, N. Eguchi & G. Moser (2012): *A New Order-theoretic Characterisation of the Polytime Computable Functions*. In: *Proceedings of the 10th APLAS*, *LNCS* 7705, pp. 280–295.

[4] M. Avanzini & G. Moser (2008): *Complexity Analysis by Rewriting*. In: *Proceedings of the 9th FLOPS*, *LNCS* 4989, pp. 130–146.

[5] H. P. Barendregt, M. C. J. D. van Eekelen, J. R. W. Glauert, R. Kennaway, M. J. Plasmeijer & M. R. Sleep (1987): *Term Graph Rewriting*. In: *Proceedings of PARLE 1987, Volume II*, *LNCS* 259, pp. 141–158.

[6] S. Bellantoni & S. A. Cook (1992): *A New Recursion-theoretic Characterization of the Polytime Functions*. *Computational Complexity* 2(2), pp. 97–110.

[7] U. Dal Lago, S. Martini & M. Zorzi (2010): *General Ramified Recurrence is Sound for Polynomial Time*. In Patrick Baillot, editor: *Proceedings of DICE 2010*, pp. 47–62.

[8] N. Eguchi (2014): *Proving Termination of Unfolding Graph Rewriting for General Safe Recursion*. Available at `http://arxiv.org/abs/1404.6196v3`. Technical report, arXiv:1404.6196v3.

[9] D. Leivant (1995): *Ramified Recurrence and Computational Complexity I: Word Recurrence and Poly-time*. In Peter Clote & Jeffrey B. Remmel, editors: *Feasible Mathematics II, Progress in Computer Science and Applied Logic*, 13, Birkhäuser Boston, pp. 320–343.