Complexity Analysis of Precedence Terminating Infinite Graph Rewrite Systems

Naohi Eguchi*

Faculty of Science, Chiba University, Japan

neguchi@g.math.s.chiba-u.ac.jp

The general form of *safe recursion* (or *ramified recurrence*) can be expressed by an infinite graph rewrite system including *unfolding graph rewrite rules* introduced by Dal Lago, Martini and Zorzi, in which the size of every normal form by innermost rewriting is polynomially bounded. Every unfolding graph rewrite rule is *precedence terminating* in the sense of Middeldorp, Ohsaki and Zantema. Although precedence terminating infinite rewrite systems cover all the primitive recursive functions, in this paper we consider graph rewrite systems *precedence terminating with argument separation*, which form a subclass of precedence terminating graph rewrite systems. We show that for any precedence terminating infinite graph rewrite system $\mathscr{G}$ with a specific argument separation, both the runtime complexity of $\mathscr{G}$ and the size of every normal form in $\mathscr{G}$ can be polynomially bounded. As a corollary, we obtain an alternative proof of the original result by Dal Lago et al.

1 Introduction

1.1 Backgrounds

In this paper we present a complexity analysis of a specific kind of infinite graph rewrite systems, *precedence terminating with argument separation*. The formulation of precedence termination with argument separation stems from a function-algebraic characterization of the polytime computable functions based on the principle known as *safe recursion* [6] or *tiered recursion* [13]. The schema of 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 formalizes the idea that recursive calls are restricted on normal arguments whereas substitutions of recursion terms are 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. Safe recursion is sound for polynomial runtime complexity over unary constructors, i.e., over numerals or lists, but it was not clear whether the general form of safe recursion over arbitrary constructors, which is called *general ramified recurrence* [10] or *general safe recursion*, could be related to polytime computability as well.

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

The authors of [10] answered this question positively (Theorem 1, Section 3) showing that the schema (**General Safe Recursion**) can be expressed by an infinite set of *unfolding graph rewrite rules*. To see a reason why graph rewriting was employed, consider a term rewrite system $\mathscr{R}$ over the constructors $\{\varepsilon, \mathsf{c}, 0, \mathsf{s}\}$ consisting of the following four rules with the argument separation indicated in the rules.

*The author is supported by Grants-in-Aid for JSPS Fellows (Grant No. $25 \cdot 726$).

A. Middeldorp and F. van Raamsdonk (Eds.): 8th International Workshop on Computing with Terms and Graphs (TERMGRAPH 2014)
EPTCS 183, 2015, pp. 33–47, doi:10.4204/EPTCS.183.3

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

Reduction of a term $\mathsf{f}(\mathsf{s}^m(0),t)$ generates a tree consisting of exponentially many copies of the tree t measured by m. Thus the computation should be performed over suitably shared graphs rather than terms. Moreover, the term $\mathsf{f}(\mathsf{s}(0),\mathsf{c}(\varepsilon,\varepsilon))$ leads to the term $\mathsf{c}(\mathsf{g}(\varepsilon,\varepsilon),\mathsf{g}(\varepsilon,\varepsilon))$ in three steps, where the subterm $\mathsf{g}(\varepsilon,\varepsilon)$ is duplicated, which means that costly recomputations potentially occur. Such duplications cannot be avoided by simple sharing but some essential memoization technique is necessary.

1.2 Outline

The most effort in [10] was devoted to show that unfolding graph rewrite rules expressing the schema (**General Safe Recursion**) only yield polynomial lengths of rewriting sequences and normal forms of polynomial sizes measured by the sizes of starting (term) graphs. The initial motivation of the present work was to deduce the complexity result by means of existing term rewriting techniques. In a technical report [11], rewriting sequences under unfolding graph rewrite rules are embedded into descending sequences under a termination order over lists of terms via a variant of the *predicative interpretation* [1, 3, 4]. In this paper, making the definition of unfolding graph rewrite rules more abstract, we define a class of graph rewrite systems precedence terminating with argument separation. Though the complexity analysis in the report above could be adopted, we avoid the use of intermediate termination orders but make use of numerical interpretation methods, which have been established as well as termination orders, e.g. [7]. The performed numerical interpretation is closely related to the predicative interpretation but also strongly motivated by polynomial *quasi*-interpretations presented in [8, 15, 9]. After preliminary sections, in Section 4, we show that every graph rewrite system precedence terminating with a specific argument separation reduces under the associated interpretation (Theorem 2), yielding an alternative proof of Theorem 1 (Corollary 3). In Section 5, to convince readers that the proposed method is indeed (potentially) more flexible than unfolding graph rewrite rules, we discuss two possibilities of application referring to related works.

2 Term graph rewriting

In this section, we present basics of term graph rewriting mainly following [5].

Definition 1 (Signatures, labeled graphs and paths)**.** Let $\mathscr{F}$ be a *signature*, a set of function symbols, and let $\mathsf{arity} : \mathscr{F} \rightarrow \mathbb{N}$ where $\mathsf{arity}(f)$ is called the *arity* of f. Throughout of the paper, we only consider finite signatures. We assume that $\mathscr{F}$ is partitioned into the set $\mathscr{C}$ of constructors and the set $\mathscr{D}$ of defined symbols.

Let $G = (V_G, E_G)$ be a directed graph consisting of a set V_G of vertices (or nodes) and a set E_G of directed edges. 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 $\mathsf{lab}_G : V_G \rightarrow \mathscr{F}$ and a (total) *successor* function $\mathsf{succ}_G : V_G \rightarrow V_G^*$, mapping a node $v \in V_G$ to a sequence of nodes of length $\mathsf{arity}(\mathsf{lab}_G)$, such that if $\mathsf{succ}_G(v) = v_1, \dots, v_k$, then $\{v_1, \dots, v_k\} = \{u \in V_G \mid (v,u) \in E_G\}$. In case $\mathsf{succ}_G(v) = v_1, \dots, v_k$, the node v_j is called the j^{th} *successor of* v for every $j \in \{1, \dots, k\}$. In particular, $\mathsf{succ}_G(v)$ is empty if $\mathsf{lab}_G(v)$ is not defined.

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 of *length* k if v_{j+1} is the m_j^{th} successor of v_j for each $j \in \{1, \dots, k-1\}$. In case $k = 0$, the list $\langle v \rangle$ consisting of a single node v is a trivial path of length 0. A labeled graph $(G, \mathsf{lab}_G, \mathsf{succ}_G)$ is *closed* if the labeling function lab_G is total.

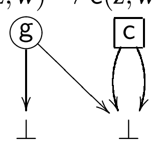
Definition 2 (Term graphs, sub-term graphs, basic term graphs, depths of term graphs and maximal sharing)**.** A quadruple $(G, \mathsf{lab}_G, \mathsf{succ}_G, \mathsf{root}_G)$ is a *term graph* if $(G, \mathsf{lab}_G, \mathsf{succ}_G)$ is a labeled graph and root_G is the *root* of G, i.e., a unique node in V_G from which every node is reachable. We write $\mathcal{TG}(\mathcal{F})$ to denote the set of term graphs over a signature $\mathcal{F}$. 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. We write $H \sqsubseteq G$ to express that H is a sub-term graph of G and $\sqsubset$ for the proper relation. A term graph $G \in \mathcal{TG}(\mathcal{F})$ is called *basic* if $\mathsf{lab}_G(\mathsf{root}_G) \in \mathcal{D}$ and $G{\restriction}v \in \mathcal{TG}(\mathcal{C})$ for every successor node v of root_G. For a term graph G, the length of the longest path(s) from root_G the *depth* of G, denoted as $\mathsf{dpth}(G)$.

Undefined nodes in a term graph G are intended to behave as free variables in a natural term representation of G. Let term_G be an injective mapping from undefined nodes in G to a (possibly infinite) set $\mathcal{V}$ of variables. The mapping term_G is canonically extended to a term representation (over $\mathcal{F} \cup \mathcal{V}$) of sub-term graphs of G as $\mathsf{term}_G(G{\restriction}v) = \mathsf{term}_G(v) \in \mathcal{V}$ in case $\mathsf{lab}_G(v)$ is not defined, and otherwise $\mathsf{term}_G(G{\restriction}v) = \mathsf{lab}_G(v)(\mathsf{term}_G(G{\restriction}v_1), \ldots, \mathsf{term}_G(G{\restriction}v_k))$ where $\mathsf{succ}_G(v) = v_1, \ldots, v_k$. A term graph G is *maximally shared* if, for any two nodes $u, v \in V_G$, $\mathsf{term}_G(G{\restriction}u) = \mathsf{term}_G(G{\restriction}v)$ implies $u = v$ (under an arbitrary choice of such a mapping term_G).

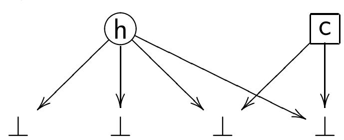
Definition 3 (Homomorphisms, redexes, graph rewrite rules and constructor graph rewrite rules)**.** Given two labeled graphs G and H, a *homomorphism* from G to H is a mapping $\varphi : V_G \to V_H$ such that $\mathsf{lab}_H(\varphi(v)) = \mathsf{lab}_G(v)$ for each $v \in \mathsf{dom}(\mathsf{lab}_G) \subseteq V_G$ and that, for each $v \in \mathsf{dom}(\mathsf{lab}_G)$, if $\mathsf{succ}_G(v) = v_1, \ldots, v_k$, then $\mathsf{succ}_H(\varphi(v)) = \varphi(v_1), \ldots, \varphi(v_k)$. By definition, these conditions are not required for a node $v \in V_G$ for which $\mathsf{lab}_G(v)$ is not defined. A homomorphism φ from a term graph G to another term graph H is a homomorphism $\varphi : (G, \mathsf{lab}_G, \mathsf{succ}_G) \to (H, \mathsf{lab}_H, \mathsf{succ}_H)$ 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 two distinct nodes l and r respectively called the *left* and *right* root. The term rewrite rule $\mathsf{g}(x, y) \to \mathsf{c}(y, y)$ is expressed by a graph rewrite rule (1) and $\mathsf{h}(x, y, z, w) \to \mathsf{c}(z, w)$ by (2) below.

(1)



(2)



In the examples, the left root is written in a circle while the right root is in a square, and undefined nodes are indicated as $\bot$. As in the usual term rewriting setting, we assume that every undefined node occurring in $G{\restriction}r$ occurs in $G{\restriction}l$ as well.

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 $\mathcal{G}$ 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 $G{\restriction}l$ is a basic term graph. A GRS $\mathcal{G}$ is called a constructor one if $\mathcal{G}$ consists only of constructor rewrite rules. The rewrite relation in a GRS $\mathcal{G}$ is defined by the *build*, *redirection* and *garbage collection* phase, denoted as $\to_{\mathcal{G}}$ (See, e.g., [10]). In case that $G \to_{\mathcal{G}} H$ is induced by a redex $((K, l, r), \varphi)$, one can find a homomorphism $\psi : K{\restriction}r \to H$ compatible with φ such that G results in H by replacing the sub-term graph $G{\restriction}\varphi(l)$ of G with $H{\restriction}\psi(r)$. A formal definition can be found in [5]. The m-fold iteration of $\to_{\mathcal{G}}$ is denoted as $\to_{\mathcal{G}}^{m}$ and the reflective and transitive closure as $\to_{\mathcal{G}}^{*}$. A rewriting $G \to_{\mathcal{G}} H$ induced by a redex $((K_0, l_0, r_0), \varphi_0)$ is *innermost* if there is no redex $((K, l, r), \varphi)$ such that $G{\restriction}\varphi(l)$ is a proper sub-term graph of $G{\restriction}\varphi_0(l_0)$. The innermost rewrite relation in $\mathcal{G}$ is denoted as $\xrightarrow{\mathsf{i}}_{\mathcal{G}}$, and $\xrightarrow{\mathsf{i}}_{\mathcal{G}}^{m}$, $\xrightarrow{\mathsf{i}}_{\mathcal{G}}^{*}$ are defined accordingly.

3 Unfolding graph rewrite rules for general safe recursion

To make the purpose of the present work precise, in this section we restate the main result in [10], formulating the *general safe recursive* functions. Let $\mathscr{C}$ be a set of constructors and $m \mapsto c_m$ ($1 \leq m \leq |\mathscr{C}|$) be an enumeration for $\mathscr{C}$. We assume that $\mathscr{C}$ contains at least one constant. We call a function $f : \mathscr{T}(\mathscr{C})^{k+l} \to \mathscr{T}(\mathscr{C})$ *general safe recursive* if, under a suitable argument separation $f(x_1,\dots,x_k;y_1,\dots,y_l)$, f can be defined from the initial functions by operating the schemata specified as follows.

- $O_j^{k,l}(x_1,\dots,x_k;y_1,\dots,y_l) = c_j$ if c_j is a constant. **(Constants)**
- $C_j(;x_1,\dots,x_{\mathsf{arity}(c_j)}) = c_j(x_1,\dots,x_{\mathsf{arity}(c_j)})$ if $\mathsf{arity}(c_j) \neq 0$. **(Constructors)**
- $I_j^{k,l}(x_1,\dots,x_k;x_{k+1},\dots,x_{k+l}) = x_j$ $(1 \leq j \leq k+l)$. **(Projections)**
- $P_{i,0}(;c_i) = c_i$ $(\mathsf{arity}(c_i) = 0)$, $P_{i,j}(;c_i(x_1,\dots,x_{\mathsf{arity}(c_i)})) = x_j$ $(1 \leq j \leq \mathsf{arity}(c_i))$. **(Predecessors)**
- $C(;c_j(x_1,\dots,x_{\mathsf{arity}(c_j)}),y_1,\dots,y_{|\mathscr{C}|}) = y_j$. **(Conditional)**
- $f(x_1,\dots,x_k;y_1,\dots,y_l) = h(x_{j_1},\dots,x_{j_m};g_1(\vec{x};\vec{y}),\dots,g_n(\vec{x};\vec{y}))$ $(\{j_1,\dots,j_m\} \subseteq \{1,\dots,k\})$, where h has m normal and n safe argument positions. **(Safe composition)**
- $f(c_j(x_1,\dots,x_{\mathsf{arity}(c_j)}),\vec{y};\vec{z}) = h_j(\vec{x},\vec{y};\vec{z}, f(x_1,\vec{y};\vec{z}),\dots, f(x_{\mathsf{arity}(c_j)},\vec{y};\vec{z}))$ $(j \in I)$. If c_j is a constant, the schema of denotes $f(c_j,\vec{y};\vec{z}) = h_j(\vec{y};\vec{z})$. **(General safe recursion)**

In [10] 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 $G \xrightarrow{\mathrm{i}}_{\mathscr{G}} H$ if such a term graph exists, or the value false if otherwise. In addition, a GRS $\mathscr{G}$ is *polynomially bounded* if there exists a polynomial $p : \mathbb{N} \to \mathbb{N}$ such that $\max\{m, |H|\} \leq p(|G|)$ holds whenever $G \xrightarrow{\mathrm{i}}{}^{m}_{\mathscr{G}} H$ holds.

Theorem 1 (Dal Lago, Martini and Zorzi [10])**.** *Every general safe recursive function can be computed by a polytime presentable and polynomially bounded constructor GRS.*

Remark 1. The schema (**General Safe Recursion**) is formulated based on safe recursion (on notation) following [6] whereas the schema of general ramified recurrence formulated in [10] is based on ramified recurrence following [13]. Due to the difference, the definition of general safe recursive functions above is slightly different from the original definition of *tiered recursive* functions in [10]. Notably, the schema (**Safe composition**) is a weaker form of the original one in [6], which was introduced in [12]. It is not clear whether there is a precise correspondence between general safe recursive functions in the current formulation and tiered recursive functions. However, it is known that the polytime functions (over binary words) can be covered with the weak form of safe composition, which means that the restriction of the general safe recursive functions to unary constructors still covers all the polytime functions.

Theorem 1 is shown by induction over a general safe recursive function f. The case that f is defined by (**General Safe Recursion**) is witnessed by an infinite set of *unfolding graph rewrite rules*.

Definition 4 (Unfolding graph rewrite rules)**.** Let Σ and Θ be two disjoint signatures in a 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 $f \not\in \Sigma \cup \Theta$ and $\mathsf{arity}(f) = 1 + k$. Given a natural $m \geq 1$, the m^{th} *set of unfolding graph rewrite rule over* Σ *and* Θ *defining* f consists of graph rewrite rules of the form (G,l,r) where $G = (V_G, E_G, \mathsf{succ}_G, \mathsf{lab}_G)$ is a labeled graph over a signature $\mathscr{F} \supseteq \Sigma \cup \Theta$ that fulfills the following conditions.

1. The set V_G of vertices consists of $1 + 2m + k$ elements y, $v_1,\dots,v_m$, $w_1,\dots$, w_m, $x_1,\dots,x_k$.
2. $l = y$ and $r = w_1$.
3. $\mathsf{lab}_G(y) = f$ and $\mathsf{succ}_G(y) = v_1, x_1,\dots,x_k$.

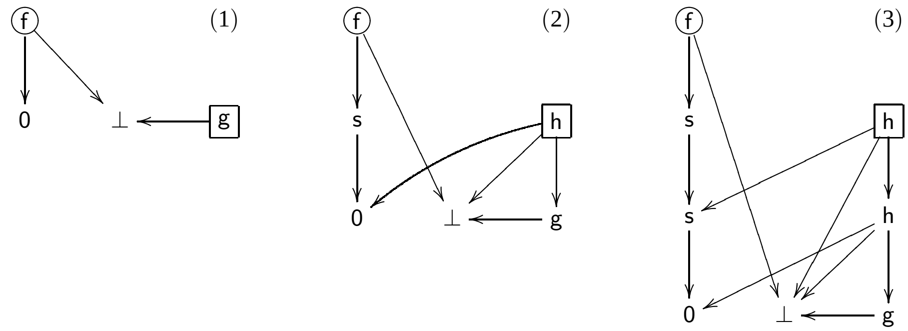


Figure 1: Examples of unfolding graph rewrite rules

4. $\mathsf{lab}_G(x_j)$ is not defined for all $j \in \{1,\dots,k\}$.
5. For each $j \in \{1,\dots,m\}$, $\mathsf{succ}_G(v_j) \in \{v_1,\dots,v_m\}^*$. Moreover, $V_{G\restriction v_1} = \{v_1,\dots,v_m\}$.
6. For each $j \in \{1,\dots,m\}$, $\mathsf{lab}_G(v_j) \in \Sigma$ and $\mathsf{lab}_G(w_j) = \varphi(\mathsf{lab}_G(v_j))$.
7. For each $j \in \{1,\dots,m\}$, $\mathsf{succ}_G(w_j) = v_{j_1},\dots,v_{j_n},x_1,\dots,x_k,w_{j_1},\dots,w_{j_n}$ if $\mathsf{succ}_G(v_j) = v_{j_1},\dots,v_{j_n}$.

Example 1. Let $\Sigma = \{\mathsf{0},\mathsf{s}\}$, $\Theta = \{\mathsf{g},\mathsf{h}\}$, $\varphi : \Sigma \to \Theta$ be a bijection defined as $\mathsf{0} \mapsto \mathsf{g}$ and $\mathsf{s} \mapsto \mathsf{h}$, and $\mathsf{f} \notin \Sigma \cup \Theta$, where the arities of $\mathsf{0},\mathsf{s},\mathsf{g},\mathsf{h},\mathsf{f}$ are respectively 0, 1, 1, 3 and 2. Namely we consider the case $k = 1$. The standard equations $\mathsf{f}(\mathsf{0},x) \to \mathsf{g}(x)$, $\mathsf{f}(\mathsf{s}(y),x) \to \mathsf{h}(y,x,\mathsf{f}(y,x))$ of primitive recursion can be expressed by the infinite GRS $\bigcup_{m\geq 1} \mathscr{G}_m$, where $\mathscr{G}_m$ is the m^{th} set of unfolding graph rewrite rules over $\mathscr{F} = \Sigma \cup \Theta \cup \{\mathsf{f}\}$ defining f. In this case, for each $m \geq 1$, $\mathscr{G}_m$ consists of a single rule. For example, in case $i = 1,2,3$, $\mathscr{G}_i$ consists of the rewrite rule (i) pictured in Figure 1. As seen from the pictures, the unfolding graph rewrite rules in Figure 1 express the infinite instances $\mathsf{f}(\mathsf{0},x) \to \mathsf{g}(x)$, $\mathsf{f}(\mathsf{s}(\mathsf{0}),x) \to \mathsf{h}(\mathsf{0},x,\mathsf{g}(x))$, $\mathsf{f}(\mathsf{s}(\mathsf{s}(\mathsf{0})),x) \to \mathsf{h}(\mathsf{s}(\mathsf{0}),x,\mathsf{h}(\mathsf{0},x,\mathsf{g}(x)))$, ..., representing terms as suitably shared term graphs.

To keep every term graph compatible with the associated argument separation, in [10], for any redex (R,φ), the homomorphism φ is limited to an *injective* one. In this paper, instead of assuming injectivity of homomorphisms, we rather indicate argument separations explicitly.

Definition 5 (Term graphs with argument separation)**.** In accordance with the idea of safe recursion, we assume that the argument positions of every function symbol are separated into the normal and safe ones, writing $f(x_1,\dots,x_k\,;x_{k+1},\dots,x_{k+l})$ to denote k normal arguments and l safe ones. We always assume that every constructor symbol in $\mathscr{C}$ has safe argument positions only. The argument separations of function symbols are taken into account in 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. For two distinct nodes v_0 and v_1, if $\mathsf{lab}_G(v_0) = \mathsf{lab}_G(v_1)$, then, for any $j \in \{1,\dots,\mathsf{arity}(\mathsf{lab}_G(v_0))\}$, $u_0 \in \mathsf{nrm}(v_0) \Leftrightarrow u_1 \in \mathsf{nrm}(v_1)$ for the j^{th} successor u_i of v_i $(i = 0,1)$. Notationally, we write $\mathsf{succ}_G(v) = v_1,\dots,v_k\,;v_{k+1},\dots,v_{k+l}$ to express the separation such that $v_1,\dots,v_k \in \mathsf{nrm}(v)$ and $v_{k+1},\dots,v_{k+l} \in \mathsf{safe}(v)$. We assume that any homomorphism $\varphi : G \to H$ preserves argument separations. Namely, for each $v \in \mathsf{dom}(\mathsf{lab}_G)$, if $\mathsf{succ}_G(v) = v_1,\dots,v_k;\ v_{k+1},\dots,v_{k+l}$, then $\mathsf{succ}_H(\varphi(v)) = \varphi(v_1),\dots,\varphi(v_k);\ \varphi(v_{k+1}),\dots,\varphi(v_{k+l})$.

Let us recall the idea of safe recursion that the number of recursive calls is measured only by a normal argument and recursion terms can be substituted only for safe arguments. This motivates us to introduce the following safe version of unfolding graph rewrite rules.

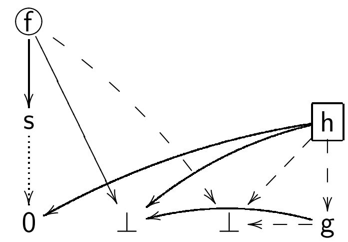


Figure 2: Example of a safe recursive unfolding graph rewrite rule

Definition 6 (Safe recursive unfolding graph rewrite rules)**.** We call an unfolding graph rewrite rule *safe recursive* if the following constraints imposed on the clause 3 and 7 in Definition 4 are satisfied.

1. In the clause 3, $v_1 \in \mathsf{nrm}(y)$, and in the clause 7, $v_{j_1}, \ldots, v_{j_n} \in \mathsf{nrm}(w_j)$ and $w_{j_1}, \ldots, w_{j_n} \in \mathsf{safe}(w_j)$.
2. In the clause 3 and 7, 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 $i \in \{1, \ldots, m\}$.

As a consequence of Definition 6, we have a basic property of safe recursive unfolding graph rewrite rules, which ensures that rewriting by any unfolding graph rewrite rule does not change the structures of subgraphs in normal argument positions.

Corollary 1. *Let (G, y, w_1) be a safe recursive unfolding graph rewrite rule with the set V_G of vertices consisting of $1+2m+k+l$ elements y, $v_1, \ldots, v_m$, $w_1, \ldots, w_m$, $x_1, \ldots, x_{k+l}$ specified as in Definition 4 and 6, where $\mathsf{succ}_G(y) = v_1, x_1, \ldots, x_k; x_{k+1}, \ldots, x_{k+l}$. Then $G\restriction u \sqsubset_{\mathsf{nrm}} G\restriction y$ holds for any $j \in \{1, \ldots, m\}$ and for any node $u \in \mathsf{nrm}(w_j)$.*

Corollary 1 follows from an observation that, for any $j \in \{1, \ldots, m\}$ and for any node $u \in \mathsf{nrm}(w_j)$, either $u \in \mathsf{dom}(\mathsf{lab}_G)$ and $u = v_i$ for some $i \in \{1, \ldots, m\}$, or $u \notin \mathsf{dom}(\mathsf{lab}_G)$ and $u = x_i$ for some $i \in \{1, \ldots, k\}$. This is exemplified by a safe recursive (constructor) unfolding graph rewrite rule in Figure 2 in case $m = 2$ and $l = k = 1$ that expresses the term rewrite rule $\mathsf{f}(\mathsf{s}(;0), x; y) \to \mathsf{h}(0, x; y, \mathsf{g}(x; y))$. To make a contrast, every edge $v \longrightarrow u$ is expressed as $v \dashrightarrow u$ if $u \in \mathsf{safe}(v)$ and $\mathsf{lab}_G(v) \in \mathscr{D}$ and as $v \cdots\!\!> u$ if $\mathsf{lab}_G(v) \in \mathscr{C}$.

4 Precedence termination with argument separation

Every unfolding graph rewrite rule is *precedence terminating* in the sense of [16]. In this section we propose a restriction of the standard precedence termination orders, *precedence termination with argument separation*. To show the polynomial runtime complexity of those GRSs, we also introduce a non-standard number-theoretic interpretation of GRSs precedence terminating with argument separation.

Let $\mathscr{F} = \mathscr{C} \cup \mathscr{D}$ be a signature. 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 $<$: $\mathsf{rk}(g) < \mathsf{rk}(f) \Leftrightarrow g < f$. We always assume that every constructor symbol is $<$-minimal.

Definition 7 (A restrictive sub-term graph relation $\sqsubset_{\mathsf{nrm}}$ and precedence termination with argument separation)**.** Let $<$ be a precedence on a signature $\mathscr{F}$ and $G, H \in \mathscr{TG}(\mathscr{F})$. We write $H \sqsubset_{\mathsf{nrm}} G$ if $H \sqsubseteq G\restriction v$ holds for some node $v \in \mathsf{nrm}(\mathsf{root}_G)$. The relation $H <_{\mathsf{pt+nrm}} G$ holds if $\mathsf{lab}_H(v) < \mathsf{lab}_G(\mathsf{root}_G)$ for any $v \in V_H$ whenever $\mathsf{lab}_H(v)$ is defined, and additionally one of the following two cases holds.

1. $H \leqslant_{\mathsf{pt+nrm}} G\restriction u$ for some successor node u of root_G.

2. $\mathsf{lab}_H(\mathsf{root}_H)$ is defined, i.e. $\mathsf{lab}_H(\mathsf{root}_H) < \mathsf{lab}_G(\mathsf{root}_G)$,
   - $H \sqsubset_{\mathsf{nrm}} G$ for each $v \in \mathsf{nrm}(\mathsf{root}_H)$, and
   - $H \restriction v <_{\mathsf{pt+nrm}} G$ for each $v \in \mathsf{safe}(\mathsf{root}_H)$.

We say that a GRS $\mathscr{G}$ over a signature $\mathscr{F}$ is *precedence terminating with argument separation* if for some separation of argument positions and for some precedence $<$ on $\mathscr{F}$, $G \restriction r <_{\mathsf{pt+nrm}} G \restriction l$ holds for each rule $(G,l,r) \in \mathscr{G}$ for the relation $<_{\mathsf{pt+nrm}}$ induced by the precedence $<$.

Let us recall we always assume that every constructor is minimal in any precedence. By the minimality, for any constructor rewrite rule $(G,l,r) \in \mathscr{G}$, if $G \restriction r <_{\mathsf{pt+nrm}} G \restriction l$ holds by Case 1 of Definition 7, $G \restriction r \sqsubseteq G \restriction v$ holds for some successor node v of l.

Definition 8 (Safe paths and a class $\mathscr{TG}_{\mathsf{nrm}}(\mathscr{F})$ of terms)**.**

1. A path $\langle v_1, m_1, \ldots, v_{k-1}, m_{k-1}, v_k \rangle$ in a term graph G is called a *safe* one if $v_{j+1} \in \mathsf{safe}(v_j)$ for all $j \in \{1, \ldots, k-1\}$. Notationally, for a term graph G and two nodes $u, v \in V_G$, we write $u \in \mathsf{safepath}_G(v)$ if u lies on a safe path from v in G. We will also use the notation $\mathsf{safepath}_G(v)$ to denote the set of such nodes u. The relation $u \in \mathsf{safepath}_G(\mathsf{root}_G)$ will be simply written as $u \in \mathsf{safepath}_G$.
2. Given a signature $\mathscr{F} = \mathscr{C} \cup \mathscr{D}$, we define a subset $\mathscr{TG}_{\mathsf{nrm}}(\mathscr{F}) \subseteq \mathscr{TG}(\mathscr{F})$. It holds $G \in \mathscr{TG}_{\mathsf{nrm}}(\mathscr{F})$ if $G \in \mathscr{TG}(\mathscr{C})$, or $G \restriction v \in \mathscr{TG}(\mathscr{C})$ for each $v \in \mathsf{nrm}(\mathsf{root}_G)$ and $G \restriction v \in \mathscr{TG}_{\mathsf{nrm}}(\mathscr{F})$ for each $v \in \mathsf{safe}(\mathsf{root}_G)$.

By definition, the root root_G of G lies on the trivial safe path from root_G in G. In the graph rewrite rule (G,l,r) in Figure 2, visually every safe path consists only of dashed edges $\cdot - - \succ \cdot$. Thus, for non-trivial examples, the right bottom $\bot$ lies on a safe path from l, and both the same $\bot$ and g lie on a safe path from r. The definition of the class $\mathscr{TG}_{\mathsf{nrm}}(\mathscr{F})$ yields $G \in \mathscr{TG}_{\mathsf{nrm}}(\mathscr{F})$ for any basic term graph $G \in \mathscr{TG}(\mathscr{F})$.

Lemma 1. *Let $\mathscr{G}$ be a constructor GRS over a signature $\mathscr{F}$ and $G \in \mathscr{TG}_{\mathsf{nrm}}(\mathscr{F})$.*

1. *Let $(H,l,r) \in \mathscr{G}$ be a rewrite rule and $\varphi : H \restriction l \to G$ a homomorphism. Then any path from root_G to $\varphi(l)$ is a safe path.*
2. *Suppose additionally that $\mathscr{G}$ is precedence terminating with argument separation. If $G \to_{\mathscr{G}} H$, then $H \in \mathscr{TG}_{\mathsf{nrm}}(\mathscr{F})$.*

Proof. PROPERTY 1. We show the property by contradiction. Assume that there exists a path from root_G to $\varphi(l)$ that is not safe. Then the path passes a normal argument position of an intermediate node v. Since constructors have only safe argument positions, $\mathsf{lab}_G(v)$ must be a defined symbol. Hence $G \restriction \varphi(l) \in \mathscr{TG}(\mathscr{C})$ by the definition of the class $\mathscr{TG}_{\mathsf{nrm}}(\mathscr{F})$. But $\mathsf{lab}_G(\varphi(l)) = \mathsf{lab}_H(l) \in \mathscr{D}$ since $\mathscr{G}$ is a constructor GRS, contradicting $G \restriction \varphi(l) \in \mathscr{TG}(\mathscr{C})$.

PROPERTY 2. Suppose that G results in H by applying a redex (R, φ) for a rule $R = (K,l,r) \in \mathscr{G}$. Since any path from root_G to $\varphi(l)$ is a safe one by Property 1, it suffices to show that $H \restriction r_H \in \mathscr{TG}_{\mathsf{nrm}}(\mathscr{F})$ for the node $r_H \in H$ corresponding to $r \in V_K$. We show that $H \restriction r_H \in \mathscr{TG}_{\mathsf{nrm}}(\mathscr{F})$ holds by induction according to the definition of the relation $<_{\mathsf{pt+nrm}}$.

CASE. $K \restriction r \leqslant_{\mathsf{pt+nrm}} K \restriction u$ for some successor node u of l: In this case, since $\mathscr{G}$ is a constructor GRS, $K \restriction r$ is a sub-term graph of $K \restriction l$ as noted after Definition 7. Hence $H \restriction r_H \in \mathscr{TG}_{\mathsf{nrm}}(\mathscr{F})$ follows from the assumption $G \restriction \varphi(u) \in \mathscr{TG}_{\mathsf{nrm}}(\mathscr{F})$.

CASE. Otherwise: For each $v \in \mathsf{nrm}(r)$, $K \restriction r$ is a sub-term graph of $K \restriction u$ for some $u \in \mathsf{nrm}(l)$. By assumption, $G \restriction \varphi(u) \in \mathscr{TG}(\mathscr{C})$ for each $u \in \mathsf{nrm}(l)$, and hence $H \restriction v \in \mathscr{TG}(\mathscr{C})$ also holds for each $v \in \mathsf{nrm}(r_H)$. On the other hand, $K \restriction v <_{\mathsf{pt+nrm}} K \restriction l$ for each $v \in \mathsf{safe}(r)$. The induction hypothesis yields $H \restriction v \in \mathscr{TG}_{\mathsf{nrm}}(\mathscr{F})$ for each $v \in \mathsf{safe}(r_H)$. These allow us to conclude $H \restriction r_H \in \mathscr{TG}_{\mathsf{nrm}}(\mathscr{F})$. ☐

For a finite set $N = \{m_i \in \mathbb{N} \mid i \in I\}$, let $\sum N$ denote the natural $\sum_{i\in I} m_i$ with the convention $\sum \emptyset = 0$.

Definition 9 (Number-theoretic interpretation of term graphs)**.** Let $G \in \mathscr{TG}_{\mathsf{nrm}}(\mathscr{F})$ be a closed term graph over a finite signature $\mathscr{F} = \mathscr{C} \cup \mathscr{D}$, $f = \mathsf{lab}_G(\mathsf{root}_G)$, and $<$ be a precedence on $\mathscr{F}$. Then, given a positive natural ℓ, we define an interpretation $\mathsf{I}_\ell : \mathscr{TG}_{\mathsf{nrm}}(\mathscr{F}) \to \mathbb{N}$ by

$$\mathsf{I}_\ell(G) = \sum \left\{ (1+\ell)^{2\cdot \mathsf{rk}(f)} \cdot \left(1 + \textstyle\sum_{u\in \mathsf{nrm}(v)} \mathsf{dpth}(G{\restriction}u)\right) \mid v \in \mathsf{safepath}_G \text{ and } G{\restriction}v \notin \mathscr{TG}(\mathscr{C}) \right\}.$$

By definition, $\mathsf{I}_\ell(G) = 0$ if $G \in \mathscr{TG}(\mathscr{C})$. We write $\mathsf{J}_\ell(G{\restriction}v)$ to abbreviate the component $(1+\ell)^{2\cdot\mathsf{rk}(f)} \cdot \left(1 + \sum_{u\in\mathsf{nrm}(v)} \mathsf{dpth}(G{\restriction}u)\right)$.

Lemma 2 (Main lemma)**.** *Let (G,l,r) be a constructor rewrite rule such that $G{\restriction}r <_{\mathsf{pt+nrm}} G{\restriction}l$ holds for the relation $<_{\mathsf{pt+nrm}}$ induced by a precedence $<$ on a finite signature $\mathscr{F}$. Also let $G_{\mathsf{L}}, G_{\mathsf{R}} \in \mathscr{TG}_{\mathsf{nrm}}(\mathscr{F})$ respectively denote closed instances of $G{\restriction}l$ and $G{\restriction}r$. If $|G{\restriction}r| \leq \ell$, then $\mathsf{I}_\ell(G_{\mathsf{R}}) < \mathsf{I}_\ell(G_{\mathsf{L}})$ holds for the interpretation I_ℓ induced by the precedence $<$.*

Proof. We estimate an upper bound for $\mathsf{I}_\ell(G_{\mathsf{R}}) = \sum\{\mathsf{J}_\ell(G{\restriction}v) \mid v \in \mathsf{safepath}_{G_{\mathsf{R}}} \text{ and } G{\restriction}v \notin \mathscr{TG}(\mathscr{C})\}$ dividing the domain $\{v \in V_{G_{\mathsf{R}}} \mid \mathsf{safepath}_{G_{\mathsf{R}}} \text{ and } G{\restriction}v \notin \mathscr{TG}(\mathscr{C})\}$ into two parts. Let $V_{\mathsf{l}} \subseteq V_{G_{\mathsf{L}}}$ denote the set of labeled nodes that already occur in $G{\restriction}l$. More precisely, if G_{L} is the result of instantiation by a homomorphism φ from $G{\restriction}l$ to an underlying term graph, $V_{\mathsf{l}} = \{v \in V_{G_{\mathsf{L}}} \mid \exists u \in V_{G{\restriction}l} \cap \mathsf{dom}(\mathsf{lab}_G) \text{ s.t. } v = \varphi(u)\}$. In other words, $V_{G_{\mathsf{L}}} \setminus V_{\mathsf{l}}$ is the set of nodes that are newly added by the instantiation. Let V_{r} denote the corresponding subset of $V_{G_{\mathsf{R}}}$. Since every undefined node in $G{\restriction}r$ occurs in $G{\restriction}l$ as a general assumption, $V_{G_{\mathsf{R}}} \setminus V_{\mathsf{r}} \subseteq V_{G_{\mathsf{L}}} \setminus V_{\mathsf{l}}$ holds. Write f to denote $\mathsf{lab}_G(l)$.

Claim 1. $\mathsf{J}_\ell(G_{\mathsf{R}}{\restriction}v) \leq (1+\ell)^{2\cdot\mathsf{rk}(f)-1} \cdot \left(1 + \sum_{u\in\mathsf{nrm}(\mathsf{root}_{G_{\mathsf{L}}})} \mathsf{dpth}(G_{\mathsf{L}}{\restriction}u)\right)$ *holds for any* $v \in \mathsf{safepath}_{G_{\mathsf{R}}} \cap V_{\mathsf{r}}$.

Write g to denote $\mathsf{lab}_{G_{\mathsf{R}}}(v)$, which is defined by definition of V_{r}. By the assumption $G{\restriction}r <_{\mathsf{pt+nrm}} G{\restriction}l$, $g < f$ for the given precedence $<$, and hence $\mathsf{rk}(g) < \mathsf{rk}(f)$ holds. By Definition 7, for each $v' \in \mathsf{nrm}(v)$, $G_{\mathsf{R}}{\restriction}v'$ is a sub-term graph of $G_{\mathsf{L}}{\restriction}u$ for some $u \in \mathsf{nrm}(\mathsf{root}_{G_{\mathsf{L}}})$. Hence, for each $v' \in \mathsf{nrm}(v)$, $\mathsf{dpth}(G_{\mathsf{R}}{\restriction}v') \leq \sum_{u\in\mathsf{nrm}(\mathsf{root}_{G_{\mathsf{L}}})} \mathsf{dpth}(G_{\mathsf{L}}{\restriction}u)$, i.e., $1 + \sum_{u\in\mathsf{nrm}(v)} \mathsf{dpth}(G_{\mathsf{R}}{\restriction}u) \leq 1 + \mathsf{arity}(g) \cdot \left(\sum_{u\in\mathsf{nrm}(\mathsf{root}_{G_{\mathsf{L}}})} \mathsf{dpth}(G_{\mathsf{L}}{\restriction}u)\right) \leq (1+\ell) \cdot \left(1 + \sum_{u\in\mathsf{nrm}(\mathsf{root}_{G_{\mathsf{L}}})} \mathsf{dpth}(G_{\mathsf{L}}{\restriction}u)\right)$. Letting $v \in \mathsf{safepath}_{G_{\mathsf{R}}} \cap V_{\mathsf{r}}$, this allow us to reason as follows.

$$\begin{array}{rcll}
\mathsf{J}_\ell(G_{\mathsf{R}}{\restriction}v) & \leq & (1+\ell)^{2\cdot\mathsf{rk}(f)-2} \cdot \left(1 + \sum_{u\in\mathsf{nrm}(v)} \mathsf{dpth}(G_{\mathsf{R}}{\restriction}u)\right) & (\text{since } \mathsf{rk}(g) \leq \mathsf{rk}(f) - 1) \\
& \leq & (1+\ell)^{2\cdot\mathsf{rk}(f)-1} \cdot \left(1 + \sum_{u\in\mathsf{nrm}(\mathsf{root}_{G_{\mathsf{L}}})} \mathsf{dpth}(G_{\mathsf{L}}{\restriction}u)\right) &
\end{array}$$

Since $|\mathsf{safepath}_{G_{\mathsf{R}}} \cap V_{\mathsf{r}}| \leq |G{\restriction}r| \leq \ell$, Claim 1 allows us to reason as follows.

$$\begin{array}{rll}
& \sum \left\{ \mathsf{J}_\ell(G_{\mathsf{R}}{\restriction}v) \mid v \in \mathsf{safepath}_{G_{\mathsf{R}}} \cap V_{\mathsf{r}} \text{ and } G_{\mathsf{R}}{\restriction}v \notin \mathscr{TG}(\mathscr{C}) \right\} & \\
\leq & \ell \cdot (1+\ell)^{2\cdot\mathsf{rk}(f)-1} \cdot \left(1 + \sum_{u\in\mathsf{nrm}(\mathsf{root}_{G_{\mathsf{L}}})} \mathsf{dpth}(G_{\mathsf{L}}{\restriction}u)\right) & (\text{by Claim 1}) \\
< & (1+\ell)^{2\cdot\mathsf{rk}(f)} \cdot \left(1 + \sum_{u\in\mathsf{nrm}(\mathsf{root}_{G_{\mathsf{L}}})} \mathsf{dpth}(G_{\mathsf{L}}{\restriction}u)\right) = \mathsf{J}_\ell(G_{\mathsf{L}}) &
\end{array} \qquad (1)$$

Claim 2. *If $v \in \mathsf{safepath}_{G_{\mathsf{R}}} \setminus V_{\mathsf{r}}$ and $G_{\mathsf{R}}{\restriction}v \notin \mathscr{TG}(\mathscr{C})$, then $v \in \mathsf{safepath}_{G_{\mathsf{L}}} \setminus V_{\mathsf{l}}$ and $G_{\mathsf{L}}{\restriction}v \notin \mathscr{TG}(\mathscr{C})$.*

Suppose $v \in \mathsf{safepath}_{G_{\mathsf{R}}} \setminus V_{\mathsf{r}}$ and, $G_{\mathsf{R}}{\restriction}v \notin \mathscr{TG}(\mathscr{C})$. Then $G_{\mathsf{L}}{\restriction}v \notin \mathscr{TG}(\mathscr{C})$ since $G_{\mathsf{R}}{\restriction}v \sqsubseteq G_{\mathsf{L}}{\restriction}v$ holds as mentioned above. We show $v \in \mathsf{safepath}_{G_{\mathsf{L}}} \setminus V_{\mathsf{l}}$ by contradiction. So assume $v \notin \mathsf{safepath}_{G_{\mathsf{L}}} \setminus V_{\mathsf{l}}$. Then, since $G_{\mathsf{L}} \in \mathscr{TG}_{\mathsf{nrm}}(\mathscr{F})$, $G_{\mathsf{L}}{\restriction}v \in \mathscr{TG}(\mathscr{C})$ holds as observed in the proof of Lemma 1.1. Claim 2 allows us to reason as follows.

$$\begin{aligned} & \sum\left\{\mathsf{J}_\ell(G_{\mathsf{R}}{\restriction}v) \mid v \in \mathsf{safepath}_{G_{\mathsf{R}}} \setminus V_{\mathsf{r}} \text{ and } G_{\mathsf{R}}{\restriction}v \notin \mathscr{TG}(\mathscr{C})\right\} \\ \leq\ & \sum\left\{\mathsf{J}_\ell(G_{\mathsf{L}}{\restriction}v) \mid v \in \mathsf{safepath}_{G_{\mathsf{L}}} \setminus V_{\mathsf{l}} \text{ and } G_{\mathsf{L}}{\restriction}v \notin \mathscr{TG}(\mathscr{C})\right\} \qquad \text{(by Claim 2)} \\ \leq\ & \sum\left\{\mathsf{J}_\ell(G_{\mathsf{L}}{\restriction}v) \mid v \in \mathsf{safepath}_{G_{\mathsf{L}}} \setminus \{\mathsf{root}_{G_{\mathsf{L}}}\} \text{ and } G_{\mathsf{L}}{\restriction}v \notin \mathscr{TG}(\mathscr{C})\right\} \qquad (2) \end{aligned}$$

Combining the inequalities (1) and (2), we conclude $\mathsf{I}_\ell(G_{\mathsf{R}}) < \mathsf{I}_\ell(G_{\mathsf{L}})$. □

The next lemma states that the *normal part* of a starting basic term graph does not change under precedence termination with argument separation.

Lemma 3. *Let $\mathscr{G}$ be a constructor GRS over a signature $\mathscr{F}$ that is precedence-terminating with argument separation and $G_0 \in \mathscr{TG}(\mathscr{F})$ a closed basic term graph. If $G_0 \to^*_{\mathscr{G}} G$, then $G{\restriction}u \sqsubset_{\mathsf{nrm}} G_0$ holds for any nodes $v \in \mathsf{safepath}_G$ and $u \in \mathsf{nrm}(v)$.*

Proof. Suppose $G_0 \to^n_{\mathscr{G}} G$. We show the assertion by induction on $n \geq 0$. In the base case $n = 0$, $G = G_0$ and $\mathsf{nrm}(v) = \emptyset$ for any $v \in \mathsf{safepath}_G \setminus \{\mathsf{root}_G\}$ since G_0 is basic. Hence the assertion trivially holds.

For the induction step, suppose that $G_0 \to^n_{\mathscr{G}} G$ holds and that $G \to_{\mathscr{G}} H$ is induced by a redex (R, φ) in H for a rewrite rule $R = (K, l, r) \in \mathscr{G}$ and a homomorphism $\varphi : K{\restriction}l \to G$. Then $G, H \in \mathscr{TG}_{\mathsf{nrm}}(\mathscr{F})$ by Lemma 1.2. First let us consider the case $\varphi(l) = \mathsf{root}_G$. By induction hypothesis, it suffices to show that for any nodes $v_H \in \mathsf{safepath}_H$ and $u_H \in \mathsf{nrm}(v_H)$ there exists a node $v_G \in \mathsf{safepath}_G$ such that $H{\restriction}u_H \sqsubset_{\mathsf{nrm}} G{\restriction}v_G$ holds. Let $v_H \in \mathsf{safepath}_H$ and $u_H \in \mathsf{nrm}(v_H)$.

CASE. $K{\restriction}r \leqslant_{\mathsf{pt+nrm}} K{\restriction}u$ for some successor node u of l: Since $\mathscr{G}$ is a constructor GRS, $K{\restriction}r \sqsubseteq K{\restriction}u$, and hence $H \sqsubseteq G$ holds. If $v_H \in \mathsf{safepath}_G$, then we can let $v_G = v_H$. If $v_H \notin \mathsf{safepath}_G$, then any path from root_G to v_H passes an normal argument position of a node $v_G \in \mathsf{safepath}_G$. This means $H{\restriction}v_H \sqsubset_{\mathsf{nrm}} G{\restriction}v_G$, and thus $H{\restriction}u_G \sqsubset_{\mathsf{nrm}} G{\restriction}v_G$.

CASE. Otherwise: If $v \in V_G \setminus \{\varphi(u) \mid u \in V_{G{\restriction}l}\}$, then, as in the previous case, one can find a node $v_G \in \mathsf{safepath}_G$ such that $H{\restriction}v_H \sqsubset_{\mathsf{nrm}} G{\restriction}v_G$. Thus we assume that v_H is mapped from $V_{K{\restriction}r}$ by φ. Then $K{\restriction}r <_{\mathsf{pt+nrm}} K{\restriction}l$ yields $H{\restriction}v_H <_{\mathsf{pt+nrm}} G$. By the definition of the relation $<_{\mathsf{pt+nrm}}$, $H{\restriction}u_H \sqsubset_{\mathsf{nrm}} G{\restriction}\varphi(l)$ holds. Since $\varphi(l) \in \mathsf{safepath}_G$ by Lemma 1.1, we can let $v_G = \varphi(l)$.

Now consider the case $\varphi(l) \neq \mathsf{root}_H$. Let $r_H \in H$ denote the node corresponding to $r \in K$. Let us consider the subcase $v_H \in V_{H{\restriction}r_H}$. In this subcase, since $v_H \in \mathsf{safepath}_H(r_H)$, as in the case $\varphi(l) = \mathsf{root}_G$, there exists a node $v_G \in \mathsf{safepath}_G$ such that $H{\restriction}u_H \sqsubset_{\mathsf{nrm}} G \restriction v_G$ holds. Since $\varphi(l) \in \mathsf{safepath}_G$ by Lemma 1.1, the induction hypothesis yields $H{\restriction}u_G \sqsubset_{\mathsf{nrm}} G_0$. Consider the subcase $v_H \notin V_{H{\restriction}r_H}$. In this subcase, $v_H \in \mathsf{safepath}_G$. As in the previous subcase, $V_{H{\restriction}u_H} \cap V_{H{\restriction}r_H} \subseteq V_{G{\restriction}u_H} \cap V_{G{\restriction}\varphi(l)}$ holds. On the other side $V_{H{\restriction}u_H} \setminus V_{H{\restriction}r_H} \subseteq V_{G{\restriction}u_H} \setminus V_{G{\restriction}\varphi(l)}$ holds. Combining the two inclusions, we reason as

$$V_{H{\restriction}u_H} = (V_{H{\restriction}u_H} \cap V_{H{\restriction}r_H}) \cup (V_{H{\restriction}u_H} \setminus V_{H{\restriction}r_H}) \subseteq \left(V_{G{\restriction}u_H} \cap V_{G{\restriction}\varphi(l)}\right) \cup \left(V_{G{\restriction}u_H} \setminus V_{G{\restriction}\varphi(l)}\right) \subseteq V_{G{\restriction}u_H}.$$

This implies $H{\restriction}u_H \sqsubseteq G{\restriction}u_H$. Since $v_H \in \mathsf{safepath}_G$ and $u_H \in \mathsf{nrm}(v_H)$, the induction hypothesis yields $G{\restriction}u_H \sqsubset_{\mathsf{nrm}} G_0$, and thus $H \restriction u_H \sqsubset_{\mathsf{nrm}} G_0$. □

To express that a term graph G is maximally shared *with respect to normal argument positions* of the root root_G, we define a term graph $G \cap \mathsf{nrm}$ consisting only of sub-term graphs connecting to normal argument positions of root_G. If G represents a term $f(t_1, \ldots, t_k; t_{k+1}, \ldots, t_{k+l})$, then $G \cap \mathsf{nrm}$ represents the term $f(t_1, \ldots, t_k; x_1, \ldots, x_l)$ with l fresh variables $x_1, \ldots, x_l$.

Definition 10. Let $G \in \mathscr{TG}(\mathscr{F})$ be a term graph with $\mathsf{succ}_G(\mathsf{root}_G) = v_1,\dots,v_k\,;v_{k+1},\dots,v_{k+l}$. If $\mathsf{lab}_G(v)$ is not defined for any $v \in \mathsf{safe}(\mathsf{root}_G)$, then $G \cap \mathsf{nrm}$ simply denotes G. Otherwise, $G \cap \mathsf{nrm}$ is defined from l distinct nodes $u_1,\dots,u_l$ not contained in V_G as follows.

$$
\begin{aligned}
V_{G\cap\mathsf{nrm}} &= \{\mathsf{root}_G\} \cup \left(\textstyle\bigcup_{v\in\mathsf{nrm}(\mathsf{root}_G)} V_{G\restriction v}\right) \cup \{u_1,\dots,u_l\} \\
E_{G\cap\mathsf{nrm}} &= \left\{(u,v) \in E_G \mid u,v \in \{\mathsf{root}_G\} \cup \left(\textstyle\bigcup_{v\in\mathsf{nrm}(\mathsf{root}_G)} V_{G\restriction v}\right)\right\} \cup \{(\mathsf{root}_G, u_j) \mid j = 1,\dots,l\} \\
\mathsf{lab}_{G\cap\mathsf{nrm}}(v) &= \begin{cases} \mathsf{lab}_G(v) & \text{if } v \in V_G, \\ \text{not defined} & \text{otherwise.} \end{cases} \\
\mathsf{root}_{G\cap\mathsf{nrm}} &= \mathsf{root}_G
\end{aligned}
$$

By definition, $\mathsf{succ}_{G\cap\mathsf{nrm}}(\mathsf{root}_{G\cap\mathsf{nrm}}) = v_1,\dots,v_k\,;u_1,\dots,u_l$. A choice of nodes $u_1,\dots,u_l$ is not important and hence will be always omitted in later discussions.

Since an underlying signature $\mathscr{F} = \mathscr{C} \cup \mathscr{D}$ is finite, for any (infinite) constructor GRS $\mathscr{G}$ over $\mathscr{F}$, the defined symbols $\mathscr{D}$ can be partitioned into two sets $\mathscr{D}_{\mathsf{inf}}$ and $\mathscr{D}_{\mathsf{fin}}$ so that every symbol $f \in \mathscr{D}_{\mathsf{inf}}$ is defined by an infinite number of rules whereas every symbol $f \in \mathscr{D}_{\mathsf{fin}}$ is defined by a finite number of rules. Accordingly, we define a partition of every constructor GRS $\mathscr{G}$ into two sets $\mathscr{G}_{\mathsf{inf}}$ and $\mathscr{G}_{\mathsf{fin}}$ by $\mathscr{G}_{\mathsf{inf}} = \{(G,l,r) \in \mathscr{G} \mid \mathsf{lab}_G(l) \in \mathscr{D}_{\mathsf{inf}}\}$ and $\mathscr{G}_{\mathsf{fin}} = \{(G,l,r) \in \mathscr{G} \mid \mathsf{lab}_G(l) \in \mathscr{D}_{\mathsf{fin}}\}$.

Theorem 2. *Let $\mathscr{G}$ be a constructor GRS over a finite signature $\mathscr{F}$ that is precedence terminating with argument separation and let $\max(\{\mathsf{arity}(f) \mid f \in \mathscr{F}\} \cup \{|K\restriction r| \mid \exists l\ (K,l,r) \in \mathscr{G}_{\mathsf{fin}}\}) \le d$. Suppose that, for any rule $(K,l,r) \in \mathscr{G}_{\mathsf{inf}}$,* (i) *$(K\restriction l) \cap \mathsf{nrm}$ is maximally shared,* (ii) *$K\restriction v$ is closed for every $v \in \mathsf{nrm}(l) \cap \mathsf{dom}(\mathsf{lab}_K)$,* (iii) *$|\{v \in \mathsf{nrm}(l) \mid v \notin \mathsf{dom}(\mathsf{lab}_K)\} \cup (\bigcup_{v\in\mathsf{safe}(l)} V_{K\restriction v})| \le d$, and* (iv) *$|K\restriction r| \le |K\restriction l| + |\bigcup_{v\in\mathsf{nrm}(r)} V_{K\restriction v}|$. Then, for any closed basic term graph $G_0 \in \mathscr{TG}(\mathscr{F})$, if $G_0 \to_{\mathscr{G}}^* G$, $G \to_{\mathscr{G}} H$ and $2\cdot(|\bigcup_{v\in\mathsf{nrm}(\mathsf{root}_{G_0})} V_{G_0\restriction v}| + d) \le \ell$, then $\mathsf{I}_\ell(H) < \mathsf{I}_\ell(G)$ holds.*

We write $(V_G)_{\mathsf{nrm}}$ to abbreviate the set $\bigcup_{u\in\mathsf{nrm}(\mathsf{root}_G)} V_{G\restriction u}$. The conditions (i) and (ii) ensure that one step of rewriting can only reduce a constant number of nodes in $(V_{K\restriction l})_{\mathsf{nrm}}$ by sharing. The condition (iii) ensures the same for nodes in $V_{K\restriction l} \setminus (V_{K\restriction l})_{\mathsf{nrm}}$. Since the condition (iv) implies $|K\restriction r| \le 2\cdot|K\restriction l|$, the condition expresses that not only $K\restriction l$ but $K\restriction r$ is also suitably shared.

Proof. Given a closed basic term graph $G_0 \in \mathscr{TG}(\mathscr{F})$, suppose that $G_0 \to_{\mathscr{G}}^* G$ and that $G \to_{\mathscr{G}} H$ is induced by a redex (R,φ) in G for a rule $R = (K,l,r)$ and a homomorphism $\varphi : K\restriction l \to G$. Then $G,H \in \mathscr{TG}_{\mathsf{nrm}}(\mathscr{F})$ holds by Lemma 1.2. Let $2\cdot\left(|(V_{G_0})_{\mathsf{nrm}}| + d\right) \le \ell$. We show that $|K\restriction r| \le 2\cdot\left(|(V_{G\restriction\varphi(l)})_{\mathsf{nrm}}| + d\right)$ holds. In case $(K,l,r) \in \mathscr{G}_{\mathsf{fin}}$, $|K\restriction r| \le d$ holds by assumption. In case $(K,l,r) \in \mathscr{G}_{\mathsf{inf}}$ we deduce the following two inequalities.

$$|K\restriction l| \le |(V_{G\restriction\varphi(l)})_{\mathsf{nrm}}| + d \tag{3}$$

$$|(V_{K\restriction r})_{\mathsf{nrm}}| \le |(V_{G\restriction\varphi(l)})_{\mathsf{nrm}}| + d \tag{4}$$

The homomorphism φ is injective over $(V_{K\restriction l})_{\mathsf{nrm}} \cap \mathsf{dom}(\mathsf{lab}_K)$ by maximal sharing of $(K\restriction l) \cap \mathsf{nrm}$. Hence $|K\restriction l| \le |(V_{G\restriction\varphi(l)})_{\mathsf{nrm}}| + d$ holds by the assumptions (ii) and (iii).

We deduce the inequality (4) by case analysis. In case that $K\restriction r \leqslant_{\mathsf{pt+nrm}} K\restriction u$ for some successor node u of l, $K\restriction r \in \mathscr{TG}(\mathscr{C})$, and hence $|(V_{K\restriction r})_{\mathsf{nrm}}| = |\emptyset| = 0$ since constructors only have safe argument positions. Otherwise, for every $v \in \mathsf{nrm}(r)$, $K\restriction v$ is a sub-term graph of $K\restriction u$ for some $u \in \mathsf{nrm}(l)$. Thus $|(V_{K\restriction r})_{\mathsf{nrm}}| \le |(V_{K\restriction l})_{\mathsf{nrm}}|$ holds, and hence the inequality (4) follows from $|(V_{K\restriction l})_{\mathsf{nrm}}| \le |(V_{G\restriction\varphi(l)})_{\mathsf{nrm}}| + d$.

Combining the assumption (iv) with the inequalities (3) and (4) yields $|K\restriction r| \le 2\cdot\left(|(V_{G\restriction\varphi(l)})_{\mathsf{nrm}}| + d\right)$. On the other hand, since $\varphi(l) \in \mathsf{safepath}_G$ by Lemma 1.1, Lemma 3 yields $|(V_{G\restriction\varphi(l)})_{\mathsf{nrm}}| \le |(V_{G_0})_{\mathsf{nrm}}|$.

Therefore $|K{\restriction}r| \leq 2 \cdot \left(|(V_{G_0})_{\mathsf{nrm}}| + d\right) \leq \ell$ holds. Now, letting $r_H \in V_H$ denote the node corresponding to $r \in V_K$, we deduce $\mathsf{I}_\ell(H) < \mathsf{I}_\ell(G)$ as follows.

$$
\begin{array}{ll}
& \mathsf{I}_\ell(H) \\
= & \mathsf{I}_\ell(H{\restriction}r_H) + \sum\{\mathsf{J}_\ell(H{\restriction}v) \mid v \in \mathsf{safepath}_H \setminus \mathsf{safepath}_H(r_H) \text{ and } H{\restriction}v \notin \mathscr{TG}(\mathscr{C})\} \\
< & \mathsf{I}_\ell(G{\restriction}\varphi(l)) + \sum\{\mathsf{J}_\ell(H{\restriction}v) \mid v \in \mathsf{safepath}_H \setminus \mathsf{safepath}_H(r_H) \text{ and } H{\restriction}v \notin \mathscr{TG}(\mathscr{C})\} \quad \text{(by Lemma 2)} \\
\leq & \mathsf{I}_\ell(G{\restriction}\varphi(l)) + \sum\{\mathsf{J}_\ell(G{\restriction}v) \mid v \in \mathsf{safepath}_G \setminus \mathsf{safepath}_G(\varphi(l)) \text{ and } H{\restriction}v \notin \mathscr{TG}(\mathscr{C})\} = \mathsf{I}_\ell(G)
\end{array}
$$

The second inequality follows from $\mathsf{safepath}_H \setminus \mathsf{safepath}_H(r_H) \subseteq \mathsf{safepath}_G \setminus \mathsf{safepath}_G(\varphi(l))$. ◻

Lemma 4. *Let $\mathscr{G}$ be a constructor GRS over a finite signature $\mathscr{F}$ that is precedence-terminating with argument separation and let* $\max(\{\mathsf{arity}(f) \mid f \in \mathscr{F}\} \cup \{|G{\restriction}r| \mid \exists l\ (G,l,r) \in \mathscr{G}_{\mathsf{fin}}\}) \leq d$. *Suppose the assumptions* (i)–(iv) *in Theorem 2 are fulfilled. Then, for any closed basic term graph $G_0 \in \mathscr{TG}(\mathscr{F})$, if $G_0 \to^n_{\mathscr{G}} G$, then $|G| \leq |G_0| + n \cdot \left(|\bigcup_{v \in \mathsf{nrm}(\mathsf{root}_{G_0})} V_{G_0{\restriction}v}| + 2d\right)$ holds.*

Proof. By induction on n. In the base case $n = 0$, $G = G_0$ and hence the assertion trivially holds. For the induction step, suppose that $G_0 \to^n_{\mathscr{G}} G$ holds and that $G \to_{\mathscr{G}} H$ is induced by a redex (R,φ) in G for a rule $R = (K,l,r) \in \mathscr{G}$ and a homomorphism $\varphi : K{\restriction}l \to G$. In case $R \in \mathscr{G}_{\mathsf{fin}}$, $|H| \leq |G| + d$ by the choice of the constant d. Suppose $R \in \mathscr{G}_{\mathsf{inf}}$. As in the proof of Theorem 2, the homomorphism φ is injective over $(V_{K{\restriction}l})_{\mathsf{nrm}} \cap \mathsf{dom}(\mathsf{lab}_K)$ by maximal sharing of $(K{\restriction}l) \cap \mathsf{nrm}$. Thus, by the assumptions (ii) and (iii), at most d nodes in $V_{K{\restriction}l}$ can be shared by the homomorphism φ. These observations imply $|H| \leq |G| + |H{\restriction}r_H| - |G{\restriction}\varphi(l)| \leq |G| + |K{\restriction}r| + d - |K{\restriction}l|$ for the node $r_H \in V_H$ corresponding to $r \in V_K$. Therefore $|H| \leq |G| + |(V_{K{\restriction}l})_{\mathsf{nrm}}| + d$ holds by the assumption (iv) $|K{\restriction}r| \leq |K{\restriction}l| + |(V_{K{\restriction}l})_{\mathsf{nrm}}|$. For the same reason as above $|(V_{K{\restriction}l})_{\mathsf{nrm}}| \leq |(V_{G{\restriction}\varphi(l)})_{\mathsf{nrm}}| + d$ holds, and thus $|H| \leq |G| + |(V_{G{\restriction}\varphi(l)})_{\mathsf{nrm}}| + 2d$ holds. On the other hand, since $\varphi(l) \in \mathsf{safepath}_G$ by Lemma 1.1, $|(V_{G{\restriction}\varphi(l)})_{\mathsf{nrm}}| \leq |(V_{G_0{\restriction}\mathsf{root}_{G_0}})_{\mathsf{nrm}}|$ holds by Lemma 3. Therefore $|H| \leq |G| + |(V_{G{\restriction}\mathsf{root}_{G_0}})_{\mathsf{nrm}}| + 2d$ holds. Combining this inequality with the induction hypothesis $|G| \leq |G_0| + n \cdot \left(|(V_{G_0{\restriction}\mathsf{root}_{G_0}})_{\mathsf{nrm}}| + 2d\right)$ allows us to conclude $|H| \leq |G_0| + (n+1) \cdot \left(|(V_{G_0{\restriction}\mathsf{root}_{G_0}})_{\mathsf{nrm}}| + 2d\right)$. ◻

Corollary 2. *Suppose that $\mathscr{G}$ is a constructor GRS over a finite signature $\mathscr{F}$ precedence-terminating with argument separation that enjoys the assumptions* (i)–(iv) *in Theorem 2. Then there exists a polynomial $p : \mathbb{N} \to \mathbb{N}$ such that, for any closed basic term graph $G_0 \in \mathscr{TG}(\mathscr{F})$ and for any term graph $G \in \mathscr{TG}(\mathscr{F})$, if $G_0 \to^m_{\mathscr{G}} G$, then the following two conditions hold.*

1. $m \leq p\left(|\bigcup_{v \in \mathsf{nrm}(\mathsf{root}_{G_0})} V_{G_0{\restriction}v}|\right)$.
2. $|G| \leq p\left(|\bigcup_{v \in \mathsf{nrm}(\mathsf{root}_{G_0})} V_{G_0{\restriction}v}|\right) + |V_{G_0} \setminus \bigcup_{v \in \mathsf{nrm}(\mathsf{root}_{G_0})} V_{G_0{\restriction}v}|$.

Proof. We only show the existence of a witnessing polynomial $q : \mathbb{N} \to \mathbb{N}$ for Property 1. The construction of a polynomial p witnessing both Property 1 and 2 will be clear from the polynomial q and Lemma 4. Given a GRS $\mathscr{G}$ over a finite signature $\mathscr{F}$, let $\max\{\mathsf{arity}(f) \mid f \in \mathscr{F}\} \leq d$. In addition, given a closed basic term graph $G_0 \in \mathscr{TG}(\mathscr{F})$, let $2 \cdot \left(|(V_{G_0{\restriction}\mathsf{root}_{G_0}})_{\mathsf{nrm}}| + d\right) \leq \ell$. Suppose that $G_0 \to^m_{\mathscr{G}} G$ holds for some term graph $G \in \mathscr{TG}(\mathscr{F})$. By Theorem 2, m can be bounded by $\mathsf{I}_\ell(G_0)$. Since $G_0{\restriction}v \in \mathscr{TG}(\mathscr{C})$ for any node $v \in \mathsf{safepath}_{G_0} \setminus \{\mathsf{root}_{G_0}\}$, $\mathsf{I}_\ell(G_0) = \mathsf{J}_\ell(G_0)$ holds. Now let q denote a polynomial such that $(2x + 2d + 1)^{2 \cdot \max\{\mathsf{rk}(f) \mid f \in \mathscr{F}\}} \cdot (1 + dx) \leq q(x)$. Since $\sum_{v \in \mathsf{nrm}(\mathsf{root}_{G_0})} \mathsf{dpth}(G_0{\restriction}v) \leq d \cdot |(V_{G_0{\restriction}\mathsf{root}_{G_0}})_{\mathsf{nrm}}|$, the inequality $2 \cdot \left(|(V_{G_0{\restriction}\mathsf{root}_{G_0}})_{\mathsf{nrm}}| + d\right) \leq \ell$ allows us to conclude $m \leq \mathsf{I}_\ell(G_0) \leq q\left(|(V_{G_0{\restriction}\mathsf{root}_{G_0}})_{\mathsf{nrm}}|\right)$. ◻

Remark 2. The assumption (iv) in Theorem 2 can be relaxed as $|K{\restriction}r| \leq |K{\restriction}l| + p\left((V_{K\restriction l})_{\mathsf{nrm}}\right)$ for some polynomial p if ℓ is sufficiently large so that a certain polynomial in $|(V_{G_0})_{\mathsf{nrm}}|$ determined by p can be bounded by ℓ. Since such a relaxed form of the condition (iv) likely holds under a suitable term rewriting adoption of unfolding graph rewrite rules, it turns out that just unfolding a recursion schema seems not crucial to deduce the polynomial complexity. But, more importantly, as implied from the assumption (iii), the number of variables occurring in the right-hand side of every rule can be constantly bounded, which clearly fails in any reasonable term rewriting formulation of unfolding rewrite rules.

The next lemma ensures that the assumption (i) in Theorem 2 is not too restrictive.

Lemma 5. *Let $\mathscr{G}$ be a constructor GRS over a finite signature $\mathscr{F}$ precedence-terminating with argument separation. For any maximally shared, closed basic term graph $G_0 \in \mathscr{TG}(\mathscr{F})$, if $G_0 \to^*_{\mathscr{G}} G$, then $(G{\restriction}v) \cap \mathsf{nrm}$ is maximally shared for any $v \in \mathsf{safepath}_G$.*

Proof. Let $v \in \mathsf{safepath}_G$ and $u_0, u_1 \in V_{G\restriction v}$. Assume $\mathsf{term}_G(G{\restriction}u_0) = \mathsf{term}_G(G{\restriction}u_1)$. By the definition of the term graph $(G{\restriction}v) \cap \mathsf{nrm}$, it suffices to consider the case $u_0, u_1 \in (V_{G\restriction v})_{\mathsf{nrm}}$. In this case, by Lemma 3, $G{\restriction}u_j \sqsubset_{\mathsf{nrm}} G_0$ holds for each $j = 0, 1$. This means that $G{\restriction}u_j = G_0{\restriction}u_j$ holds for each $j = 0, 1$, and thus $\mathsf{term}_{G_0}(G_0{\restriction}u_0) = \mathsf{term}_{G_0}(G_0{\restriction}u_1)$ holds by the assumption. Maximal sharing of G_0 implies $u_0 = u_1$. ☐

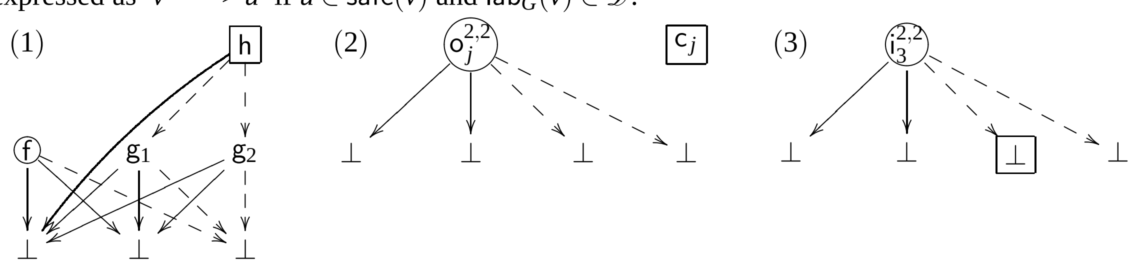
As a consequence of Lemma 1.1 and Lemma 5, for any (completely defined) constructor GRS $\mathscr{G}$ over a finite signature that is precedence terminating with argument separation, if there exists a constant d such that the assumptions (i)–(iv) in Theorem 2 hold for any rule $(K, l, r) \in \mathscr{G}_{\mathsf{inf}}$, then any rewriting sequence $G_0 \to_{\mathscr{G}} G_1 \to_{\mathscr{G}} \cdots$ starting with a maximally shared, closed basic term graph G_0 leads to a constructor term graph in normal form.

Theorem 3. *Every general safe recursive function can be computed by a constructor GRS that precedence terminating with an argument separation fulfilling the conditions* (i)–(iv) *in Theorem 2.*

Proof. By induction over the definition of f. In the base case, every initial function can be defined by a single constructor rewrite rule (G, l, r) in one of the following shapes 1 and 2.

1. $G{\restriction}r = G{\restriction}v$ for some successor node v of l.
2. V_G consists of $2 + k + l + n$ elements u, v, $x_1, \ldots, x_{k+l}$, $w_1, \ldots, w_n$ such that $l = u$, $r = v$,
   - $\{\mathsf{lab}_G(u), \mathsf{lab}_G(v), \mathsf{lab}_G(w_1), \ldots, \mathsf{lab}_G(w_n)\} \subseteq \mathscr{F}$,
   - $\mathsf{lab}_G(x_j)$ is undefined for all $j \in \{1, \ldots, k + l\}$,
   - $\mathsf{succ}_G(u) = x_1, \ldots, x_k; x_{k+1}, \ldots, x_{k+l}$,
   - $\mathsf{succ}_G(v) = x_{j_1}, \ldots, x_{j_m}; w_1, \ldots, w_n$ for some $\{j_1, \ldots, j_m\} \subseteq \{1, \ldots, k\}$, and
   - $\mathsf{succ}_G(w_j) = x_1, \ldots, x_k; x_{k+l}, \ldots, x_{k+l}$ for all $j \in \{1, \ldots, n\}$.

The graph rewrite rule (1) below is an instance of Case 2 with $k = 2$, $l = 1$ and $n = 2$, which expresses the term rewrite rule $\mathsf{f}(x_1, x_2; x_3) \to \mathsf{h}(x_1; \mathsf{g}_1(x_1, x_2; x_3), \mathsf{g}_2(x_1, x_2; x_3))$. As in Figure 2, every edge $v \longrightarrow u$ is expressed as $v \dashrightarrow u$ if $u \in \mathsf{safe}(v)$ and $\mathsf{lab}_G(v) \in \mathscr{D}$.



Every instance of **(Constants)** can be defined by a single graph rewrite rule as (2) above in a special shape of Case 2, and each of **(Projections)**, **(Predecessors)** and **(Conditional)** can be defined by a single graph rewrite rule as (3) in the form of Case 1. The induction step splits into two cases. In case that f is defined by **(Safe composition)**, f is defined by a constructor graph rewrite rule in the form of Case 2 together with the constructor GRSs obtained from induction hypothesis. In case that f is defined by **(General safe recursion)**, f is defined by an infinite set of constructor safe recursive unfolding graph rewrite rules together with the constructor GRSs obtained from induction hypothesis. For instance, suppose that f is defined by $f(\varepsilon\,;z) = g(\,;z)$ and $f(c(\,;x,y)\,;z) = h(x,y;z, f(x;z), f(y;z))$. By induction hypothesis, g and h can be respectively computed by some constructor GRSs $\mathscr{G}_{\mathsf{g}}$ and $\mathscr{G}_{\mathsf{h}}$ defining the corresponding function symbols $\mathsf{g},\mathsf{h} \in \mathscr{D}$. Let $\mathsf{e},\mathsf{c} \in \mathscr{C}$ respectively correspond to ε, c and $\mathsf{f} \in \mathscr{D}$ to f Also let $\Sigma = \{\mathsf{e},\mathsf{c}\}$ and $\Theta = \{\mathsf{g},\mathsf{h}\}$ with a bijective correspondence $\mathsf{e} \mapsto \mathsf{g}, \mathsf{c} \mapsto \mathsf{h}$. Then, for each $m \geq 1$, one can define the m^{th} set $\mathscr{G}_m$ of safe recursive unfolding graph rewrite rules over $\Sigma \cup \Theta$ defining f. Since $\Sigma \subseteq \mathscr{C}$, $\mathscr{G}_m$ is a constructor GRS for every $m \geq 1$. Since elements of $\bigcup_{m\geq 1} \mathscr{G}_m$ express $\mathsf{f}(\mathsf{e};z) \to \mathsf{g}(\,;z)$, $\mathsf{f}(\mathsf{c}(\,;\mathsf{e},\mathsf{e})\,;z) \to \mathsf{h}(\mathsf{e},\mathsf{e};z,\mathsf{g}(\,;z),\mathsf{g}(\,;z))$, ..., f is computed by the infinite GRS $\mathscr{G}_{\mathsf{g}} \cup \mathscr{G}_{\mathsf{h}} \cup \left(\bigcup_{m\geq 1} \mathscr{G}_m\right)$.

The precedence $<$ is defined so that, letting every constructor be $<$-minimal, for every rule (G,l,r), $\mathsf{lab}_G(v) < \mathsf{lab}_G(l)$ for any $v \in V_{G\restriction r}$ whenever $\mathsf{lab}_G(v)$ is defined. Then $\mathsf{g} < \mathsf{f}$ means that f is defined from g for the functions f,g respectively corresponding to f,g. Hence the well-foundedness of $<$ follows from the observation that the relation "is defined from" is well-founded by the definition of general safe recursive functions. Precedence termination of so obtained GRSs is obvious.

Let $<_{\mathsf{pt+nrm}}$ be the relation induced by the precedence $<$. By definition, the subset $\mathscr{G}_{\mathsf{inf}}$ of $\mathscr{G}$ consists of safe recursive unfolding graph rewrite rules whereas $\mathscr{G}_{\mathsf{fin}}$ contains no unfolding graph rewrite rule. It follows from the definition of safe recursive unfolding graph rewrite rules that $G\restriction l <_{\mathsf{pt+nrm}} G\restriction r$ for each $(G,l,r) \in \mathscr{G}_{\mathsf{inf}}$ (See also Corollary 1). Consider a rewrite rule $(G,l,r) \in \mathscr{G}_{\mathsf{fin}}$. It is obvious that $G\restriction l <_{\mathsf{pt+nrm}} G\restriction r$ holds if (G,l,r) is an instance of Case 1. Suppose that V_G consists of $2+k+l+n$ elements l, r, $x_1,\ldots,x_{k+l}$, $w_1,\ldots,w_n$ as specified in Case 2. Let $v \in V_{G\restriction r} = \{r,x_1,\ldots,x_{k+l},w_1,\ldots,w_n\}$. Consider the case that $\mathsf{lab}_G(v)$ is not defined, i.e., $v \in \{x_1,\ldots,x_{k+l}\}$. In this case, v is a successor node of l. Namely $G\restriction v = G\restriction u$ for some successor node u of l, and hence $G\restriction l <_{\mathsf{pt+nrm}} G\restriction v$ holds. Assume that $\mathsf{lab}_G(v) \in \mathscr{F}$. Then $v \in \{r,w_1,\ldots,w_n\}$. Since $\mathsf{succ}_G(w_j) = x_1,\ldots,x_k;x_{k+1},\ldots,x_{k+l}$ for every $j \in \{1,\ldots,n\}$, $G\restriction l <_{\mathsf{pt+nrm}} G\restriction w_j$ for every $j \in \{1,\ldots,n\}$. This yields $G\restriction l <_{\mathsf{pt+nrm}} G\restriction v$ since $\mathsf{succ}_G(v) = x_{j_1},\ldots,x_{j_m};w_1,\ldots,w_n$ for some $\{j_1,\ldots,j_m\} \subseteq \{1,\ldots,k\}$. The conditions (ii)–(iv) follow from the definition of unfolding graph rewrite rules. Choosing every rewrite rule $(G,l,r) \in \mathscr{G}$ so that $(G\restriction l) \cap \mathsf{nrm}$ is maximally shared allows one to conclude. □

Corollary 3. *For every general safe recursive function f, there exist a constructor GRS $\mathscr{G}$ that computes f and a polynomial $p : \mathbb{N} \to \mathbb{N}$ such that, for any maximally shared, closed basic term graph G, if $G \to_{\mathscr{G}}^m H$, then $m \leq p(n)$ and $|H| \leq p(n) + |G|$ hold, where n denotes the size $|\bigcup_{v\in\mathsf{nrm}(\mathsf{root}_G)} V_{G\restriction v}|$ (of the union) of the subgraphs connected to the normal argument positions of root_G only.*

The corollary says that every general safe recursive function can be computed by a polynomially bounded constructor GRS. Since such a witnessing GRS is polytime presentable in particular, Corollary 3 yields an alternative proof of Theorem 1.

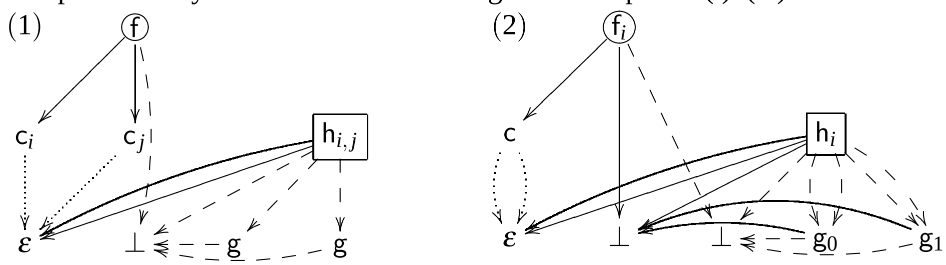
5 Related works and further application

In this section we discuss two related works to see some potential applicability of the method presented in the previous section and one more work to see a limit of the computational power of the method.

In [15] a term rewrite system $\mathscr{R}_{\mathsf{lcs}}$, which computes the length of the *longest common subs-sequence* of two strings, is discussed. The rewrite system $\mathscr{R}_{\mathsf{lcs}}$ contains instances of

$$\mathsf{f}(\varepsilon,y,z;w) \rightarrow \mathsf{g}(y,z;w) \qquad \mathsf{f}(x,\varepsilon,z;w) \rightarrow \mathsf{g}(x,z;w)$$
$$\mathsf{f}(\mathsf{c}_i(x),\mathsf{c}_j(y),z;w) \rightarrow \mathsf{h}_{i,j}(x,y,z;w,\mathsf{f}(x,\mathsf{c}_j(y),z;w),\mathsf{f}(\mathsf{c}_i(x),y,z;w)),$$

i.e., rewrite rules expressing safe recursion with multiple recursion arguments. For exactly the same reason as in case of general safe recursion, $\mathscr{R}_{\mathsf{lcs}}$ only admits a polynomial *quasi*-interpretation which says nothing about polynomial runtime complexity. Due to the restriction to single recursion arguments, it is not possible to represent these rules as instances of (safe recursive) unfolding graph rewrite rules. However, as seen from an instance (1) below (where the variable z is ignored to ease the presentation), $\mathscr{R}_{\mathsf{lcs}}$ could be represented by an infinite GRS fulfilling the assumptions (i)–(iv) in Theorem 2.



In a very recent work [2], Theorem 1 is expanded for *simultaneous* general safe recursion, e.g.,

$$\left.\begin{array}{rcl}\mathsf{f}_i(\varepsilon,z;w) & \rightarrow & \mathsf{g}_i(z;w)\\ \mathsf{f}_i(\mathsf{c}(x,y),z;w) & \rightarrow & \mathsf{h}_i(x,y,z;w,\mathsf{f}_0(x,z;w),\mathsf{f}_0(y,z;w),\mathsf{f}_1(x,z;w),\mathsf{f}_1(y,z;w))\end{array}\right. \quad (i=0,1).$$

In contrast to the current approach, instead of taking an advantage of sharing in term graph rewriting, the notion of *cache* is employed in [2] to avoid costly recomputations. A similar notion, called *minimal function graphs*, can be found in [15], yielding that the rewrite system $\mathscr{R}_{\mathsf{lcs}}$ can be executed in polynomial time. As mentioned in Remark 2, the condition (iv) in Theorem 2 can be relaxed as (iv)' $|K\restriction r| \leq |K\restriction l| + O(|(V_{K\restriction l})_{\mathsf{nrm}}|)$. Thus, as seen from an instance (2) above, such the schema of simultaneous recursion could be also represented by an infinite GRS enjoying the assumptions (i)–(iii) and (iv)'.

As shown in [14], it is known that the polynomial-space computable functions can be captured with safe recursion (on notation) *with parameter substitutions*. To see an explicit boundary of the proposed method, consider the term rewrite system below that expresses an instance of the schema.

$$\mathsf{f}(\varepsilon;y) \rightarrow \mathsf{g}(;y)$$
$$\mathsf{f}(\mathsf{c}(x);y) \rightarrow \mathsf{h}(x;y,\mathsf{f}(x;\mathsf{p}(x;y)),\mathsf{f}(x;\mathsf{q}(x;y)))$$

The rules below are the first three instances of unfolding the above rules.

$$\begin{array}{llcl}(0) & \mathsf{f}(\varepsilon;y) & \rightarrow & \mathsf{g}(;y)\\ (1) & \mathsf{f}(\mathsf{c}(\varepsilon);y) & \rightarrow & \mathsf{h}(\varepsilon;y,\mathsf{g}(;\mathsf{p}(\varepsilon;y)),\mathsf{g}(;\mathsf{q}(\varepsilon;y)))\\ (2) & \mathsf{f}(\mathsf{c}(\mathsf{c}(\varepsilon));y) & \rightarrow & \mathsf{h}(\mathsf{c}(\varepsilon);y,\mathsf{h}(\varepsilon;y,\mathsf{g}(;\mathsf{p}(\varepsilon;\mathsf{p}(\mathsf{c}(\varepsilon),y))),\mathsf{g}(;\mathsf{q}(\varepsilon;\mathsf{p}(\mathsf{c}(\varepsilon),y)))),\\ & & & \quad \mathsf{h}(\varepsilon;y,\mathsf{g}(;\mathsf{p}(\varepsilon;\mathsf{q}(\mathsf{c}(\varepsilon),y))),\mathsf{g}(;\mathsf{q}(\varepsilon;\mathsf{q}(\mathsf{c}(\varepsilon),y)))))\end{array}$$

One will see that g occurs 2^i times in the i^{th} instance (i) and none of the occurrences can be shared since their arguments are different. For this reason, even if they are represented as maximally shared GRS $\mathscr{G}$, $2^{|(V_{K\restriction l})_{\mathsf{nrm}}|} \leq |K\restriction r|$ for every $(K,l,r) \in \mathscr{G}_{\mathsf{inf}}$, and thus (even a relaxed form of) the condition (iv) fails.

6 Conclusion

Generalizing unfolding graph rewrite rules that express the schema (**General Safe Recursion**), we proposed restrictive precedence termination orders, precedence termination with argument separation. The restrictive notion together with suitable assumptions yields a new criterion for the polynomial runtime complexity of infinite GRSs and for the polynomial-size normal forms in infinite GRSs. As discussed in the last section, the proposed method can be potentially expanded for safe recursion with multiple recursion arguments or simultaneous general safe recursion, and thus is indeed more flexible than unfolding graph rules at least in a limited sense. It should be stressed, however, that it is unclear how to express infinite instances of those recursion schemata with infinite graph rewrite rules in a uniform way.

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, doi:10.1007/11590156_43.

[2] M. Avanzini & U. Dal Lago (2015): *On Sharing, Memoization, and Polynomial Time*. In: *Proceedings of the 32nd STACS*, pp. 62–75, doi:10.4230/LIPIcs.STACS.2015.62.

[3] 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, doi:10.4230/LIPIcs.RTA.2011.123.

[4] 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, doi:10.1007/978-3-642-35182-2_20.

[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: *Parallel Architectures and Languages Europe, Volume II*, 259, pp. 141–158, doi:10.1007/3-540-17945-3_8.

[6] S. Bellantoni & S. A. Cook (1992): *A New Recursion-theoretic Characterization of the Polytime Functions*. *Computational Complexity* 2(2), pp. 97–110, doi:10.1007/BF01201998.

[7] G. Bonfante, A. Cichon, J.-Y. Marion & H. Touzet (2001): *Algorithms with Polynomial Interpretation Termination Proof*. *J. Funct. Program.* 11(1), pp. 33–53, doi:10.1017/S0956796800003877.

[8] G. Bonfante, J.-Y. Marion & J.-Y. Moyen (2001): *On Lexicographic Termination Ordering with Space Bound Certifications*. In: *The 4th Andrei Ershov Memorial Conference, Revised Papers*, pp. 482–493, doi:10.1007/3-540-45575-2_46.

[9] G. Bonfante, J.-Y. Marion & J.-Y. Moyen (2011): *Quasi-interpretations A Way to Control Resources*. *Theor. Comput. Sci.* 412(25), pp. 2776–2796, doi:10.1016/j.tcs.2011.02.007.

[10] U. Dal Lago, S. Martini & M. Zorzi (2010): *General Ramified Recurrence is Sound for Polynomial Time*. In Patrick Baillot, editor: *Proceedings DICE 2010*, pp. 47–62, doi:10.4204/EPTCS.23.4.

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

[12] W. G. Handley & S. S. Wainer (1999): *Complexity of Primitive Recursion*. In U. Berger & H. Schwichtenberg, editors: *Computational Logic*, *NATO ASI Series F: Computer and Systems Science* 165, Springer, pp. 273–300, doi:10.1007/978-3-642-58622-4_8.

[13] 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, doi:10.1007/978-1-4612-2566-9_11.

[14] D. Leivant & J.-Y. Marion (1994): *Ramified Recurrence and Computational Complexity II: Substitution and Poly-Space*. In: *The 8th CSL, Selected Papers*, pp. 486–500, doi:10.1007/BFb0022277.

[15] J.-Y. Marion (2003): *Analysing the Implicit Complexity of Programs*. *Information and Computation* 183(1), pp. 2–18, doi:10.1016/S0890-5401(03)00011-7.

[16] A. Middeldorp, H. Ohsaki & H. Zantema (1996): *Transforming Termination by Self-Labeling*. In: *Proceedings of the 13th CADE*, *LNCS* 1104, pp. 373–387, doi:10.1007/3-540-61511-3_101.