Term Graph Rewriting and Strong Sequentiality

Sjaak Smetsers
University of Nijmegen*

Abstract

Abstract

This paper propagates the use of term graph rewriting systems as a formal computational model. The definition of both term graphs and the reduction relation on term graphs are more complex than their corresponding notions in term rewriting systems. Consequently, it is often believed that reasoning about term graph rewriting systems is inherently more difficult than reasoning about term rewriting systems. In this paper we will show that it is very well possible to investigate formal properties of term graph rewriting systems entirely within the framework of these systems. First, we will establish a basic theory based on *graph homomorphisms* and the *term graph reduction relation*. In this theory the concepts of *index* and *strong sequentiality* are incorporated. Lastly, we will prove that index reduction is normalising for the so-called *strongly sequential term graph rewriting systems*.

1. Introduction

Nowadays, it is a commonly accepted that graph rewriting is the most suitable technique to implement lazy functional languages efficiently. The explicit presence of sharing makes it possible to represent multiple occurrences of identical expressions by references to a single occurrence of one of these expressions. Together with a lazy evaluation scheme, fully lazy evaluation is obtained without any extra effort. However, graph rewriting is often considered to be an implementation issue only. But, elevating graph rewriting to the level theoretical models would enable not only formal reasoning about the implementation but also about properties of the model itself. This will give more insight in such a model as well as in the operational behaviour of functional programs. An example of such behaviour is the complexity in time and in use of memory during the execution of a functional program. If one does not have detailed knowledge about the implementation it is very difficult to predict how a functional program uses resources.

Many papers investigate a graph rewriting formalism called *term graph rewriting* in which terms and term rewrite rules are interpreted (lifted) as graphs and graph rewriting rules (Barendregt et al. (1987), Kennaway (1991), Raoult (1984)). In this

*Department of Computer Science, Toernooiveld 1, 6525 ED Nijmegen, The Netherlands, e-mail `sjakie@cs.kun.nl`.

formalism it is also possible to create cyclic graphs by assigning references to nodes to variables that can be used anywhere in the right-hand side graph of the rewrite rule. However, both graphs and graph rewriting are more complex than terms and term rewriting respectively. This makes reasoning about term substantially more difficult than reasoning about terms. Furthermore, many notions, properties and proofs do not easily carry over from the term to the graph rewriting world.

All these facts have caused that graph rewriting as a formal model of computation is not as popular as it should be. In this paper we want to illustrate that it is very well possible to reason entirely within the framework of term graph rewriting. Taking Barendregt et al. (1987) as a starting point we will develop a theoretical basis which is intended to serve as a first step towards a sound theoretical model (section 2 and 3). We will also show that it is sometimes even easier to derive properties of term graph rewriting systems than to derive similar properties of term rewriting systems.

In section 4, the concept of *strong sequentiality* is incorporated in TGRS's. This concept has been introduced in TRS's by Huet and Lévy (Huet and Lévy (1979)). Strong sequentiality is a decidable property which admits an efficiently normalizing reduction strategy for the corresponding class of strongly sequential TRS's. The key idea of the strongly sequential TRS's is that in these systems it always possible to indicate at least one *needed redex*. It should be noted that for ordinary orthogonal TRS's this in general not possible.

Strong sequentiality as defined by Huet and Lévy (1979) and Klop and Middeldorp (1991) is based on the notion of Ω-reduction. In this paper a similar notion for graph rewriting is presented, called $\perp$-*reduction* (section 4). The $\perp$-reduction of a graph leads to its $\perp$-*reduct*. $\perp$-reducts are used to introduce the concept of *index* with which the so called *strongly sequential TGRS's* are defined (again comparable with the approach used in Klop and Middeldorp (1991)). In Section 5 we will prove that index reduction is normalizing for strongly sequential TGRS's. The elegance of the proof is that it is short and straightforward.

Apart from the notions *extension*, *replacement* and *interference*, section 2 and 3 consist mainly of material that has been introduced by Barendregt et al. (1987). Sections 4 and 5 are basically new.

2. Graph rewriting

This section briefly recalls some basic notions that were introduced by Barendregt et al. (1987). The most important difference with Barendregt et al. (1987) is that from the original definition of graph rewriting three basic operations have been derived, to wit *redirection*, *extension* and *garbage collection*. These operations are used to define a new notion called *replacement*. Replacements, that can be compared with substitutions in TRS's, appear to be essential for specifying and proving properties in the rest of the paper. A more comprehensive treatement of graph rewriting (which has also been used to produce the description in this paper) can be found in Barendsen and Smetsers (1992).

Graphs

The objects of our interest are directed graphs in which each node is labeled with a specific symbol. The number of outgoing edges of a node is determined by its symbol. In the sequel we assume that $\mathcal{N}$ is some basic set of *nodes* (infinite; one often takes $\mathcal{N} = \mathbb{N}$), and Σ is a (possibly infinite) set of *symbols* with *arity* in $\mathbb{N}$.

2.1. Definition. (i) A *labeled graph* (over $\langle \mathcal{N}, \Sigma \rangle$) is a triple

$$g = \langle N, \mathit{symb}, \mathit{args} \rangle$$

such that

(1) $N \subseteq \mathcal{N}$; N is the set of *nodes* of g;
(2) $\mathit{symb} : N \to \Sigma$; $\mathit{symb}(n)$ is the *symbol* at node n;
(3) $\mathit{args} : N \to N^*$ such that

$$\text{length}(\mathit{args}(n)) = \text{arity}(\mathit{symb}(n)).$$

Thus $\mathit{args}(n)$ specifies the outgoing edges of n. The i-th component of $\mathit{args}(n)$ is denoted by $\mathit{args}(n)_i$.

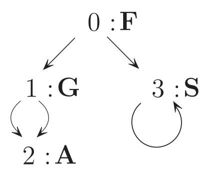
(ii) A *rooted graph* is a quadruple

$$g = \langle N, \mathit{symb}, \mathit{args}, r \rangle$$

such that $\langle N, \mathit{symb}, \mathit{args} \rangle$ is a labeled graph, and $r \in N$. The node r is called the *root* of the graph g.

(iii) The collection of all finite rooted labeled graphs over $\langle \mathcal{N}, \Sigma \rangle$ is indicated by $\mathbb{G}$.

2.2. Example. The following is a suggestive drawing of a rooted graph.



Here $N = \{0, 1, 2, 3\}$, $\mathit{symb}(0) = \mathbf{F}$, $\mathit{symb}(1) = \mathbf{G}$, etc., $\text{arity}(\mathbf{F}) = \text{arity}(\mathbf{G}) = 2$, $\text{arity}(\mathbf{S}) = 1$, $\text{arity}(\mathbf{A}) = 0$, $\mathit{args}(0) = (1, 3)$, etc., $r = 0$. The (names of) nodes are usually left implicit in drawings; they are only displayed when needed.

Convention. (i) $m, n, n', \ldots$ range over nodes; $g, g', h, \ldots$ range over (rooted) graphs.

(ii) If g is a (rooted) graph, then its components are referred to as N_g, symb_g, args_g (and r_g) respectively. Further abbreviations, such as writing N_2 instead of N_{g_2}, will be self-explanatory.

(iii) To simplify notation we usually write $n \in g$ instead of $n \in N_g$. Moreover we use $\mathbf{S} \in g$ to indicate that $\mathbf{S} = \mathit{symb}_g(n)$ for some $n \in g$.

2.3. DEFINITION. (i) A *path* in a graph g is a sequence

$$p = (n_0, i_0, n_1, i_1, \ldots, n_m)$$

where $n_0, n_1, \ldots, n_m \in g$, and $i_0, i_1, \ldots, i_{m-1} \in \mathbb{N}$ are 'edge specifications' such that

$$n_{k+1} = \mathit{args}(n_k)_{i_k}.$$

In this case p is said to be a *path from* n_0 *to* n_m (notation $p : n_0 \rightsquigarrow n_m$).

(ii) Let p be as above. The *length* of p (notation $|p|$) is m. Let $n \in g$, p *contains* n (notation $n \in p$) if $n_k = n$ for some $k \leq m$. p is a *rooted path* if it starts with the root of the graph (i.e. $p : r_g \rightsquigarrow n$ for some $n \in g$). $m \in g$ is *reachable from* n (notation $n \rightsquigarrow m$) if $p : n \rightsquigarrow m$ for some path p in g. p is *cyclic* if $n_j = n_k$ for some $j \neq k$.

2.4. DEFINITION. Let g be a rooted graph.

(i) g is *coherent* if

$$\forall n \in g \;\; \exists p \quad p : r_g \rightsquigarrow n.$$

(ii) g is a *tree* if

$$\forall n \in g \;\; \exists! p \quad p : r_g \rightsquigarrow n.$$

2.5. DEFINITION. Let g be a graph and $n \in g$. The *subgraph of* g *at* n (notation $g \,|\, n$) is the rooted graph

$$\langle N, \mathit{symb}_g \restriction N, \mathit{args}_g \restriction N, n \rangle$$

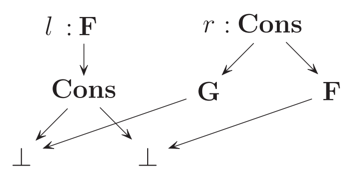
where $N = \{m \in g \mid n \rightsquigarrow m\}$. Here the notation $f \restriction N$ is used to denote the restriction of a function f to N.

Rewrite rules

Rewrite rules specify transformations of graphs. Each rewrite rule is represented by a special graph containing two roots. These roots determine the left-hand side and the right-hand side of the rule. Variables are represented by 'variable nodes'. E.g. the rule

$$\mathbf{F}(\mathbf{Cons}(x, y)) \rightarrow \mathbf{Cons}(\mathbf{G}(x), \mathbf{F}(y))$$

is represented by the following graph.



Instead of using names for variables, multiple occurrences of the same variable are indicated via sharing. E.g. all occurrences of the variable x in the above rule are represented by references to the same variable node.

A special symbol is used to indicate 'empty nodes' in a graph. One of its applications is the representation of variables in rewrite rules.

2.6. DEFINITION. Let $\perp$ be a special symbol in Σ with arity 0. Let g be a graph.

(i) The set of *empty nodes* of g (notation g°) is the collection

$$g^\circ = \{n \in g \mid symb_g(n) = \perp\}.$$

(ii) The set of *non-empty nodes* (or the *interior*) of g is denoted by $g^\bullet$. So $N_g = g^\circ \cup g^\bullet$.

(iii) g is *closed* if $g^\circ = \emptyset$.

The objects on which computations are performed are closed graphs; the others are used as auxiliary objects, e.g. for defining graph rewrite rules.

2.7. DEFINITION. A *term graph* is a closed rooted graph.

2.8. DEFINITION. (i) A *term graph rewrite rule* (or *rule* for short) is a triple

$$R = \langle g, l, r\rangle$$

where g is a (possibly open) graph, and $l, r \in g$ (called the *left root* and *right root* of R), such that

(1) $(g \mid l)^\bullet \neq \emptyset$;

(2) $(g \mid r)^\circ \subseteq (g \mid l)^\circ$.

(ii) R is a *selector rule* if $(g \mid r)^\bullet = \emptyset$.

(iii) If $symb_g(l) = \mathbf{F}$ then R is said to be a *rule for* $\mathbf{F}$.

Here condition (1) expresses that the left-hand side of the rewrite rule should not be just a variable. Moreover condition (2) states that all variables occurring on the right-hand side of the rule should also occur on the left-hand side.

NOTATION. We will write $R \mid l$, $R \mid r$ for $g_R \mid l_R$, $g_R \mid r_R$ respectively.

Graph homomorphisms

Let R be some rewrite rule. A graph g can be *rewritten* according to R if R is applicable to g, i.e. the left-hand side (the *pattern*) of R *matches* g. A *match* is a mapping from the pattern of R to a subgraph of g that preserves the node structure.

Below we define these structure preserving mappings called *graph homomorphisms*, and an ordering $\preceq$ on graphs, induced by these mappings.

2.9. DEFINITION. Let g, h be graphs, and $\varphi : N_g \to N_h$.

(i) φ is a *(graph) homomorphism* from g to h (notation $\varphi : g \to h$) if for all $n \in g^\bullet$

$$\begin{aligned} symb_h(\varphi(n)) &= symb_g(n), \\ args_h(\varphi(n))_i &= \varphi(args_g(n)_i). \end{aligned}$$

(ii) φ is a *rooted homomorphism* (notation $\varphi : g \underset{r}{\to} h$) if $\varphi : g \to h$ and $\varphi(r_g) = r_h$.

2.10. REMARK. (i) If $\varphi : g \to h$ then φ is completely determined by $\varphi(r_g)$. Consequently, if $\varphi : g \underset{r}{\to} h$ then φ is unique.

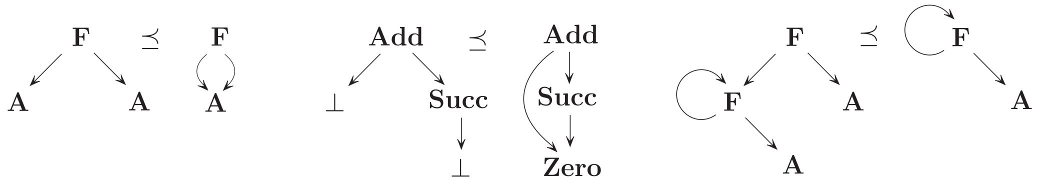
(ii) Let $\varphi : g \underset{r}{\to} h$ with h coherent. If g is closed then h is also closed.

2.11. Definition. The ordering $\preceq$ on graphs is defined as follows.

$$g \preceq h \quad \text{if} \quad \varphi : g \underset{\mathrm{r}}{\rightarrow} h \text{ for some } \varphi.$$

Sometimes we write $g \preceq_\varphi h$ to indicate the homomorphism explicitly.

2.12. Example.



2.13. Lemma. (i) *The class of rooted homomorphisms is closed under composition, i.e.*

$$\varphi : g \underset{\mathrm{r}}{\rightarrow} g',\ \psi : g' \underset{\mathrm{r}}{\rightarrow} g'' \ \Rightarrow\ \psi \circ \varphi : g \underset{\mathrm{r}}{\rightarrow} g''.$$

(ii) $g \preceq g',\ g' \preceq g'' \ \Rightarrow\ g \preceq g''$.

Proof. (i) Trivial.
(ii) By (i). □

2.14. Definition. (i) g is *isomorphic* to h (notation $g \cong h$) if there exists an isomorphism from g to h.
(ii) g is *equivalent* to h (notation $g \equiv h$) if there exists a rooted isomorphism from g to h. We will usually identify equivalent graphs and replace $\equiv$ by $=$.

2.15. Lemma. $g \preceq h,\ h \preceq g \ \Rightarrow\ g \equiv h$.

Proof. Suppose $g \preceq_\varphi h$, $h \preceq_\psi g$. Since φ and ψ are rooted one has $\varphi = \psi^{-1}$, so φ, ψ are in fact rooted isomorphisms. Hence $g \equiv h$. □

2.16. Lemma. *Let $g_1, g_2 \preceq h$ with g_1 closed.*

$$g_2 \textit{ is a tree } \ \Rightarrow\ g_2 \preceq g_1.$$

Proof. By using the results of Barendsen and Smetsers (1992). □

Graph rewriting

The combination of a rule and a match to a graph g is called a *redex*. If a redex has been determined, the graph can be rewritten according to the structure of the right-hand side of the rule involved. This is done in three steps. Firstly, the graph is *extended* with an instance of the right-hand side of the rule. Then the extension is embedded in the original graph by *redirecting* all references to the root of the redex to the root of the extension. Finally all unreachable nodes are removed by performing *garbage collection*.

2.17. Definition. Let g be a graph, and $\mathcal{R}$ a set of rewrite rules. An *$\mathcal{R}$-redex* in g (or just *redex*) is a tuple

$$\Delta = \langle R, \mu \rangle$$

where $R \in \mathcal{R}$, and $\mu : (R \mid l) \to g$. The map μ is called a *match*.

2.18. Definition. Let $n, m \in g$. The result of redirecting all references from n to m (notation $g[n := m]$) is the graph

$$g[n := m] = \langle N_g, symb_g, args, r \rangle$$

where $args$ are r are defined as follows.

$$\begin{array}{rcll} args(a)_i & = & m & \text{if } args_g(a)_i = n, \\ & = & args_g(a)_i & \text{otherwise;} \\ r & = & m & \text{if } r_g = n, \\ & = & r_g & \text{otherwise;.} \end{array}$$

Below a substitution-like operation on graphs will be described.

2.19. Definition. Let g be a graph.

(i) Let h be a graph, $H \subseteq N_h$, and $\mu : (N_h \backslash H) \to N_g$. The *extension* of g with $\langle h, H \rangle$ according to μ (notation $g +_\mu \langle h, H \rangle$) is defined as follows. Set $g +_\mu \langle h, H \rangle = k$, where

$$\begin{array}{rcll} N_k & = & N_g \uplus H; & \\ symb_k(n) & = & symb_g(n) & \text{if } n \in g, \\ & = & symb_h(n) & \text{otherwise;} \\ args_k(n)_i & = & args_g(n)_i & \text{if } n \in g, \\ & = & args_h(n)_i & \text{if } n, args_h(n) \in H, \\ & = & \mu(args_h(n)_i) & \text{otherwise;} \\ r_k & = & r_g. & \end{array}$$

Here $\uplus$ denotes the disjoint union.

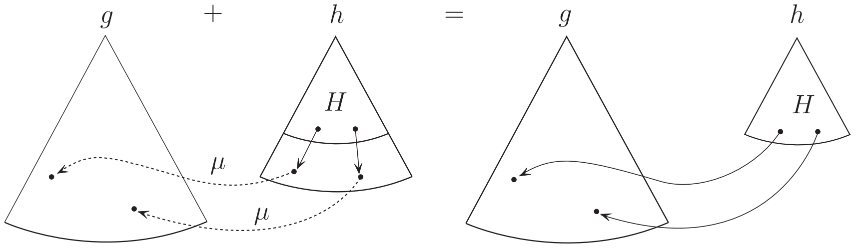
(ii) An *extension specification* (or *extension* for short) for g is a triple $E = \langle h, H, \mu \rangle$ with h, H and μ as above. Then the extension of g with E is

$$g + E = g +_\mu \langle h, H \rangle.$$

(iii) The *extension root* of E (notation $\mathrm{r}(E)$) is defined as follows.

$$\begin{array}{rcll} \mathrm{r}(E) & = & r_h & \text{if } r_h \in H, \\ & = & \mu(r_h) & \text{otherwise.} \end{array}$$

This construction can be visualized as follows.



2.20. DEFINITION. Let g be a rooted graph. The result of performing *garbage collection* on g (notation $\mathrm{GC}(g)$) is the graph obtained from g by deleting all nodes that are not reachable from its root, i.e.

$$\mathrm{GC}(g) = g \mid r_g.$$

The three basic operations extension, redirection, and garbage collection are combined into a single operation called *replacement*. This mechanism corresponds to *substitution* in lambda calculus and term rewriting.

2.21. DEFINITION. Let E be an extension of g, and let $n \in g$. The result of *replacing* n by E in g (notation $g[n := E]$) is defined by

$$g[n := E] = \mathrm{GC}((g + E)[n := \mathrm{r}(E)]).$$

NOTATION. We abbreviate the replacement $g[n := \langle h, N_h, \mu\rangle]$ by $g[n := h]$. Note that in such a replacement the map μ is not used.

Following Barendregt et al. (1987), one can distinguish the following steps in the construction of $g[n := E]$.

$$\begin{array}{lll} g & \mapsto & g + E & \text{(build phase)}, \\ & \mapsto & (g+E)[n := \mathrm{r}(E)] & \text{(redirection phase)}, \\ & \mapsto & \mathrm{GC}((g+E)[n := \mathrm{r}(E)]) & \text{(garbage collection phase)}. \end{array}$$

CONVENTION. (i) Note that for any replacement $g[n := E]$ if $\mathrm{r}(E) \notin g$ then $n \notin g[n := E]$. Now it appears to be useful if in $g[n := E]$ the node $r(E)$ is renamed to n. Obviously, this renaming cannot affect the graph structure. Without defining it formally we assume that in $g[n := E]$ the node n is the root of the extension.

(ii) Let $\mathbf{O} \in \Sigma$ with arity 0. To simplify notation we use $\mathbf{O}$ also as a denotation for a graph consisting of a single $\mathbf{O}$ node.

(iii) For $n \notin g$ we define $g \mid n = \perp$ (see also definition 2.5).

2.22. DEFINITION. Let g be a graph, and $\Delta = \langle R, \mu\rangle$ be a $\mathcal{R}$-redex of g

(i) The extension associated with Δ is

$$E_\Delta = \langle R \mid r,\ (R \mid r)^\bullet \backslash (R \mid l)^\bullet,\ \mu\rangle.$$

(ii) The result g' of *contracting* Δ in g is given by the replacement

$$g' = g[\mu(l) := E_\Delta].$$

In this case we say that g reduces to g' via Δ (notation $g \overset{\Delta}{\underset{\mathcal{R}}{\rightarrow}} g'$, or just $g \underset{\mathcal{R}}{\rightarrow} g'$). Moreover, $\mathrm{r}(\Delta)$ means $\mathrm{r}(E_\Delta)$.

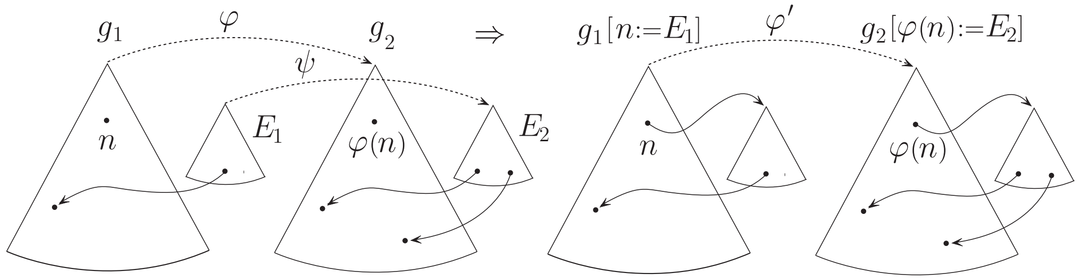
For some graph rewriting examples we refer to Barendregt et al. (1987) and Barendsen and Smetsers (1992).

Intuitively, the next proposition states that homomorphisms are preserved by replacements.

2.23. PROPOSITION. *Let $g_1 \preceq_\varphi g_2$. Furthermore, let E_1, E_2 be extensions of g_1 and g_2 respectively, such that $(g_1+E_1)|\mathrm{r}(E_1) \preceq_\psi (g_2+E_2)|\mathrm{r}(E_2)$. Take $O = g_1 \cap (g_1+E_1)|\mathrm{r}(E_1)$ then for any node $n \in g_1$*

$$\varphi \restriction O = \psi \restriction O \;\Rightarrow\; g_1[n := E_1] \preceq g_2[\varphi(n) := E_2].$$

This proposition can be visualized as follows.



The restriction that $\varphi \restriction O = \psi \restriction O$ is needed to provide that each node in O corresponds to exactly one node in $g_2[\varphi(n) := E_2]$.

PROOF. Define the function φ' as follows.

$$\begin{array}{rcll} \varphi'(m) & = & \psi(m) & \text{if } m \in g_1[n := E_1] \mid n, \\ \varphi'(m) & = & \varphi(m) & \text{otherwise.} \end{array}$$

Then proving that $g_1[n := E_1] \preceq_{\varphi'} g_2[\varphi(n) := E_2]$ is a matter of rather easy casuistics. □

We end this section with some technical results concerning replacements.

2.24. LEMMA. *Let g be a graph.*

(i) *Let $n \notin g$ then for any extenstion E one has $g[n := E] = g$.*
(ii) *Let $n \in g$, and $\mathbf{O} \in \Sigma$ with arity 0. If $g \mid n = \mathbf{O}$ then $g[n := \mathbf{O}] = g$.*
(iii) *Let $n \in g$, and $E = \langle h, N_h, \mu \rangle$ with $h = \perp$ and $\mu = r_h \mapsto n$. Then $g[n := E] = g$.*
(iv) *Let $n \in g$, and h be a graph. Then $(g[n := h]) \mid n = h$.*
(v) *Let $n \in g$, and h be a graph. Then $(g[n := h]) \mid n \cap g = \emptyset$.*
(vi) *Let $n, m \in g$, and h be a graph. Then $(g \mid m)[n := h] = g[n := h] \mid m$.*

(vii) *Let $n, m \in g$, and h_1, h_2 be graphs. Then* $g[m := h_1][n := h_2] = g[n := h_2][m := h_1]$.

Here the notation $a \mapsto b$ is used to denote a function that takes a to b and is the identity elsewhere.

PROOF. (i)-(vi) Straightforward.
(vii) See Barendsen and Smetsers (1992). $\square$

3. Graph rewrite systems

A collection of graphs and a set of rewrite rules can be combined into a graph rewrite system. The special class of so called orthogonal graph rewrite systems will be studied in more detail.

3.1. DEFINITION. A *term graph rewrite system* (TGRS) is a tuple

$$\mathcal{T} = \langle \mathcal{G}, \mathcal{R} \rangle$$

where $\mathcal{R}$ is a set of rewrite rules, and $\mathcal{G} \subseteq \mathbb{G}$ is a set of term graphs which is closed under $\mathcal{R}$-reduction.

3.2. DEFINITION. Let $\mathcal{T} = \langle \mathcal{G}, \mathcal{R} \rangle$ be a TGRS.
(i) A rule $R \in \mathcal{R}$ is *left-linear* if $R \mid l$ is a tree.
(ii) $\mathcal{T}$ is *left-linear* if each $R \in \mathcal{R}$ is left-linear.

The following definition distinguishes redexes on the basis of their relative position in a graph.

3.3. DEFINITION. Let $\Delta_1 = \langle R_1, \mu_1 \rangle$ and $\Delta_2 = \langle R_2, \mu_2 \rangle$ be redexes in g.
(i) Δ_1 and Δ_2 are *disjoint* if

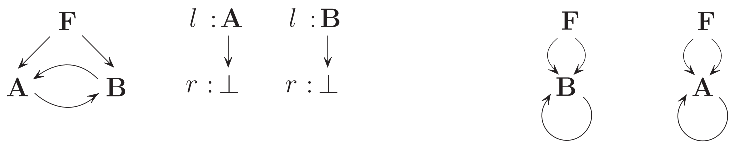
$$\mu_1(l_1) \notin \mu_2(R_2 \mid l_2)^\bullet \text{ and } \mu_2(l_2) \notin \mu_1(R_1 \mid l_1)^\bullet.$$

(ii) Δ_1 and Δ_2 *interfere* if

$$\mu_1(l_1) = \mathrm{r}(\Delta_2) \text{ and } \mu_2(l_2) = \mathrm{r}(\Delta_1).$$

If a graph contains redexes that are not disjoint, it is conceivable that different reduction orders may yield different results. This is also the case if a graph contains interfering redexes, as is illustrated by the following example.

3.4. EXAMPLE. Below a graph, rewrite rules for **A** and **B**, and the respective results after contracting the **A**-redex and the **B**-redex in the graph have been drawn.



3.5. Definition. Let $\mathcal{T} = \langle \mathcal{G}, \mathcal{R} \rangle$ be a TGRS.

(i) $\mathcal{T}$ is *regular* if for each $g \in \mathcal{G}$ the $\mathcal{R}$-redexes in g are pairwise disjoint.

(ii) $\mathcal{T}$ is *orthogonal* if $\mathcal{T}$ is both left-linear and regular.

(iii) $\mathcal{T}$ is *interference-free* if $\mathcal{G}$ does not contain any graph with interfering redexes.

In Barendregt et al. (1987) it is shown that for orthogonal interference-free TGRS's the reduction order can be changed [1]. The most important results of Barendregt et al. (1987) concerning this subject are given below.

3.6. Lemma. *Let $\Delta_1 = \langle R_1, \mu_1 \rangle$ and $\Delta_2 = \langle R_2, \mu_2 \rangle$ be disjoint redexes in g. Suppose $g \overset{\Delta_1}{\underset{\mathcal{R}}{\rightarrow}} g'$. If $\mu_2(l_2) \in g'$, then*

$$\Delta_2' = \langle R_2, [\mu_1(l_1) \mapsto \mathrm{r}(\Delta_1)] \circ \mu_2 \rangle$$

is a redex in g'. Moreover $[\mu(l_1) \mapsto \mathrm{r}(\Delta_1)] \circ \mu_2(l_2) = \mu_2(l_2)$.

3.7. Definition. Let $g, g', \Delta_1, \Delta_2, \Delta_2'$ be as in the previous lemma.

(i) If $\mu_1(l_2) \in g'$, then Δ_2' is called the *residual* of Δ_2 in g' (relative to the reduction $g \overset{\Delta_1}{\underset{\mathcal{R}}{\rightarrow}} g'$), notation $\Delta_2 /\!\!/ \Delta_1$. If $\mu_1(l_2) \notin g'$ then Δ_2 has no residual in g'.

(ii) The reduction (consisting of zero or one step) contracting $\Delta_2 /\!\!/ \Delta_1$ in g' (if it exists) is denoted by $\overset{\Delta_2 /\!\!/ \Delta_1}{\underset{\mathcal{R}}{\rightarrow}}$.

3.8. Proposition. *Let $\mathcal{T} = \langle \mathcal{G}, \mathcal{R} \rangle$ be interference-free. Suppose $g \overset{\Delta_1}{\underset{\mathcal{R}}{\rightarrow}} g_1$ and $g \overset{\Delta_2}{\underset{\mathcal{R}}{\rightarrow}} g_2$. Say $g_1 \overset{\Delta_2 /\!\!/ \Delta_1}{\underset{\mathcal{R}}{\rightarrow}} h_1$ and $g_2 \overset{\Delta_1 /\!\!/ \Delta_2}{\underset{\mathcal{R}}{\rightarrow}} h_2$. Then $h_1 = h_2$.*

3.9. Remark. If an orthogonal TGRS $\mathcal{T}$ is cycle-free then $\mathcal{T}$ is also interference-free.

Assumption. From now on we assume that all TGRS's in question are both orthogonal an cycle-free.

Convention. Regularity of TGRS's enables us to identify a redex Δ by the node n that corresponds to the root of Δ. Sometimes we will write $\overset{n}{\rightarrow}$ instead of $\overset{\Delta}{\rightarrow}$.

During the reduction of a graph g there may at each step be several rewrites possible, since g may contain several redexes. This raises the question of whether it possible to indicate a reduction order that will always lead to the normal form of g, provided that g has a normal form. Neededness is a concept for term rewriting systems that has been introduced by Huet and Lévy (1979). They prove that if a term has a normal but is not in normal form then it contains a needed redex. Moreover, repeated contraction of needed redexes of such a term leads to its normal. Unfortunately, neededness is an undecidable property. In the next section a class of term graph rewriting systems is defined, called *strongly sequential term graph rewriting systems*, in which some of the needed redexes can be identified by looking at left-hand sides only.

[1] Actually the proof was given for orthogonal TGRS's, but then the property does not hold as can be seen in the previous example. In Kennaway et al. (1990) an extension is proposed which handles arbitrary, possibly cyclic graphs correctly.

4. Strong Sequentiality

Strong sequentiality is based on two notions, to wit $\perp$-*reduction* and *index* which are based on the $\preceq$ ordering for graphs (definition 2.11).

4.1. DEFINITION. g and g' are *compatible* (notation $g \uparrow g'$), if for some h one has $g \preceq h$ and $g' \preceq h$. Otherwise g and g' are *incompatible* (notation $g \# g'$).

CONVENTION. (i) Let $H \subseteq \mathbb{G}$. We write $g \succeq H$ if for some $h \in H$ $g \succeq h$ for some $h \in H$. Otherwise $g \not\succeq H$. Analogously we write $g \uparrow H$ and $g \# H$.

(ii) The homomorphism φ in $\varphi : g \underset{\mathrm{r}}{\to} h$ relates nodes of g to nodes of h. In the rest of this paper it will not cause any confusion when this homomorphism is used implicitly. This means that if we have a node $n \in g$ then the node in h that corresponds to n is also indicated by n and not by $\varphi(n)$.

4.2. LEMMA. *Let $g_1 \preceq g_2$.*

(i) *Let $n \in g_2$, and E be an extension of g_2. Then for any graph h such that $h \preceq (g_2 + E_2) \mid \mathrm{r}(E_2)$ one has*

$$g_1[n := h] \preceq g_2[n := E].$$

(ii) *Let $h_1 \preceq h_2$, then*

$$g_1[n := h_1] \preceq g_2[n := h_2].$$

PROOF. (i) and (ii): Use lemma 2.24(iv & v) and proposition 2.23. $\square$

4.3. LEMMA. $\forall n \in g \quad g[n := g \mid n] \preceq g$.

PROOF. By lemma 2.24(iii) and lemma 4.2(i). $\square$

One might expect that also $g \preceq g[n := g \mid n]$ (and hence $g = g[n := g \mid n]$). This is not true, as illustrated by the next example in which two graphs g and $g[n := g \mid n]$ are given.

$$\begin{tikzcd}
 & \mathbf{F} \arrow[dl] \arrow[dd] & \succeq & & \mathbf{F} \arrow[dl] \arrow[dr] & \\
n : \mathbf{G} \arrow[dr] & & & \mathbf{G} \arrow[d] & & \mathbf{H} \\
 & \mathbf{H} & & \mathbf{H} & &
\end{tikzcd}$$

4.4. LEMMA. (i) *Let $g \uparrow h$. If g is closed and h be a tree then $g \succeq h$.*

(ii) *Let g, h be graphs, and $n, m \in g$. Then*

$$g[n := g \mid n] \mid m \uparrow h \;\Rightarrow\; g \mid m \uparrow h.$$

PROOF. (i) By lemma 2.16.

(ii) Using the results of Barendsen and Smetsers (1992). $\square$

4.5. LEMMA. *Let* $g_1 \stackrel{\Delta}{\to} g_2$, *where* $\Delta = \langle R, \mu \rangle$. *Then*

$$g_1[\mu(l_R) := \bot] \preceq g_2.$$

PROOF. Note that $g_2 = g_1[\mu(l_R) := \bot]$ (definition 2.22). Now apply lemma 4.2(i). $\square$

4.6. DEFINITION. Let $\langle \mathcal{G}, \mathcal{R} \rangle$ be a term graph rewrite system

(i) The set of *patterns* of $\mathcal{R}$ is

$$\mathcal{P} = \{(R \mid l) \mid R \in \mathcal{R}\}.$$

(ii) A graph $g \in \mathbb{G}$ is in $\bot$-*normal form* if

$$\forall n \in g \quad [g \mid n \neq \bot \;\Rightarrow\; g \mid n \# \mathcal{P}].$$

The set $\mathrm{NF}_\bot$ denotes the collection of all $\bot$-normal forms in $\mathbb{G}$.

4.7. REMARK. Let $g_1 \preceq g_2$. Then for all n one has $g_1 \mid n \preceq g_2 \mid n$.

4.8. DEFINITION. Let g be a graph, and $\mathcal{P}$ a set of patterns.

(i) $\bot$-*reduction*, denoted by $\underset{\bot}{\to}$, is defined on $\mathbb{G}$ by the following reduction relation.

$$g \underset{\bot}{\to} g[n := \bot], \quad \text{if } g \mid n \uparrow \mathcal{P} \text{ and } g \mid n \neq \bot.$$

(ii) A $\bot$-*reduct* of g is a normal form of g with respect to $\underset{\bot}{\to}$.

4.9. LEMMA. (i) $\underset{\bot}{\to}$ *is terminating, that is to say, for any graph g every $\bot$-reduction sequence starting with g is finite.*

(ii) *If* $g \underset{\bot}{\to} h$ *then* $h \preceq g$.

4.10. LEMMA. *Let g be a graph, and g' a $\bot$-reduct of g.*

(i) $g' \in \mathrm{NF}_\bot$.

(ii) $\forall h \in \mathrm{NF}_\bot[h \preceq g \;\Rightarrow\; h \preceq g']$.

(iii) g' *is unique.*

PROOF. (i) Easy.

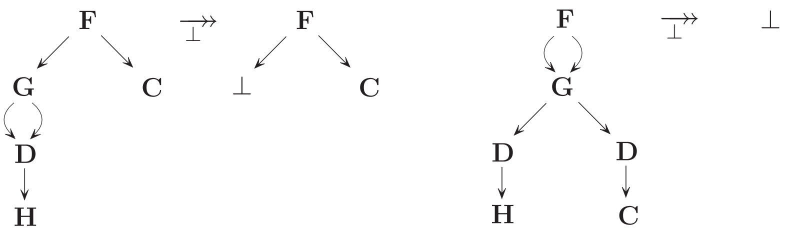
(ii) By induction on the length of the $\bot$-reduction sequence from g to g'. Let g^n be the result after the n^{th} reduction step. We will prove the following.

$$\forall n\, [\forall h \in \mathrm{NF}_\bot\, [h \preceq g \;\Rightarrow\; h \preceq g^n\,]\,].$$

The basis step is trivial since $g^0 = g$. g^{n+1} can be written as $g^n[m := \bot]$, for some $m \in g^n$ with $g^n \mid m \uparrow P$ and $g^n \mid m \neq \bot$. I.H. applied to g^n (i.e. $h \preceq g^n$) implies that if $m \in h$ then $h \mid m \preceq g^n \mid m$. So $h \mid m \uparrow P$ and therefore $h \mid m = \bot$. Applying lemmas 4.2(ii) and 2.24(ii), $h \preceq g^{n+1}$.

(iii) By (i) and (ii). $\square$

4.11. Example. Let $\mathcal{P} = \{\mathbf{F}(\mathbf{D}(\perp), \mathbf{D}(\perp)), \mathbf{G}(\mathbf{D}(\mathbf{C}), \mathbf{D}(\perp)), \mathbf{H}\}$. Below two graphs and their $\perp$-reducts are given.



$\perp$-reducts in the graph world can be compared with the ω-normal forms in the term world. The lemmas and propositions in the rest of this section correspond to the properties for ω-normal forms formalized in Klop and Middeldorp (1991).

4.12. Lemma. *Let $g, h \in \mathrm{NF}_\perp$, $h \neq \perp$, and $n \in g$. Then*

$$g \mid n \preceq h \quad \Rightarrow \quad g[n := h] \in \mathrm{NF}_\perp.$$

Proof. Suppose $g[n := h] \notin \mathrm{NF}_\perp$. Then there exists a node $m \in g[n := h]$ and a $p \in \mathcal{P}$ such that $g[n := h] \mid m \neq \perp$ and $g[n := h] \mid m \uparrow p$. Since $h \in \mathrm{NF}_\perp$, $m \notin h$ so $m \in g$. Note that $g[n := g \mid n] \mid m \neq \perp$. From $g[n := g \mid n] \preceq g[m := h]$ it follows that $g[n := g \mid n] \mid m \uparrow p$. Using lemma 4.4(ii) one has $g \mid m \uparrow p$ which contradicts to $g \in \mathrm{NF}_\perp$. □

4.13. Lemma. (i) $g \preceq h \;\Rightarrow\; g_\perp \preceq h_\perp$.
 (ii) $\forall n \in g_\perp \quad g_\perp \mid n = (g \mid n)_\perp$
 (iii) $g[n := h]_\perp \preceq g[n := h_\perp]$.

Proof. (i) $g_\perp \preceq g \preceq h$. By lemma 4.10 and $g_\perp \in \mathrm{NF}_\perp$ one has $g_\perp \preceq h_\perp$.
 (ii) Let $n \in g_\perp$. Since $g_\perp \in \mathrm{NF}_\perp$, $g_\perp \mid n \in \mathrm{NF}_\perp$. Also $g_\perp \mid n \preceq g \mid n$ so $g_\perp \mid n \preceq (g \mid n)_\perp$. Now consider the graph $g_\perp[n := (g \mid n)_\perp]$. Using lemma 4.12 it follows that $g_\perp[n := (g \mid n)_\perp] \in \mathrm{NF}_\perp$. By lemma 4.2 (ii), $g_\perp[n := (g \mid n)_\perp] \preceq g[n := g \mid n]$, and since $g[n := g \mid n] \preceq g$ (lemma 4.3), $g_\perp[n := (g \mid n)_\perp] \preceq g_\perp$ (lemma 4.10 (ii)). Therefore $(g_\perp[n := (g \mid n)_\perp]) \mid n = (g \mid n)_\perp \preceq g_\perp \mid n$.
 (iii) Notice that $g[n := h] \underset{\perp}{\twoheadrightarrow} g[n := h_\perp]$. By the uniqueness of $\perp$-normal forms (lemma 4.10(iii)) $g[n := h_\perp] \underset{\perp}{\twoheadrightarrow} g[n := h]_\perp$. Applying lemma 4.5 one has $g[n := h_\perp] \preceq g[n := h]_\perp$. □

4.14. Proposition. *Let g be a closed graph (i.e. $g^\circ = \emptyset$) such that $g \neq g_\perp$. Then g contains a redex.*

Proof. First notice that $g \neq g_\perp$ implies that there exist an $m \in g_\perp$ such that $g_\perp \mid m = \perp$. Now we will show that there exist an $n \in g \mid m$ such that $g \mid n \uparrow \mathcal{P}$. By lemma 4.4 (i) one has that $g \mid n \succeq P$ so n is a redex. Assume that this n does not exist which means that for all $l \in g \mid m$, $g \mid l \# P$. Hence $g \mid m = (g \mid m)_\perp$. Applying lemma 4.13 (ii) contradicts the assumption that $g_\perp \mid m = \perp$. □

4.15. Lemma. (i) $(g \mid n)_\perp = \perp \;\Rightarrow\; g_\perp \mid n = \perp$.

(ii) $g_\perp \mid n = \perp \;\Rightarrow\; g_\perp \preceq g[n := \perp]$.

Proof. (i) If $n \notin g_\perp$ then it is trivial. Otherwise, we can apply lemma 4.13 (ii) $g_\perp \mid n = (g \mid n)_\perp = \perp$.

(ii) Use $g_\perp = g_\perp[n := \perp]$ (lemma 2.24(ii)) and lemma 4.2(ii).

4.16. Definition. Let g be a graph, $n \in g$ such that $(g \mid n)_\perp = \perp$. Furthermore, let $\bullet \in \Sigma$ be a new 0-ary symbol (i.e. $\bullet$ does not appear in any of the rewrite rules) .

(i) n is called an *index* of g if $g[n := \bullet]_\perp \neq g_\perp$. The set of indexes of g is indicated by $\mathrm{I}(g)$ and $\delta_I(g) = \delta(g) \cap \mathrm{I}(g)$ (i.e. the set of redexes of g that are also indexes).

(ii) The index reduction of an index $\Delta \in \delta_I(g)$ is indicated by $\overset{\Delta}{\underset{I}{\to}}$ or by $\overset{m}{\underset{I}{\to}}$ where m is the root of Δ. Furthermore, we use $\delta_{\mathrm{NI}}(g)$ to denote the collection $\delta(g) \backslash \delta_I(g)$.

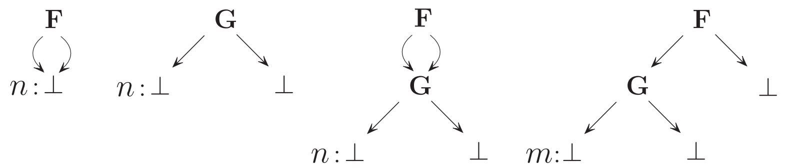
4.17. Lemma. *Let g be a graph. Then*

$$n \in I(g) \;\Leftrightarrow\; g[n := \bullet]_\perp \mid n \neq \perp.$$

Proof. ($\Leftarrow$) Trivial.

($\Rightarrow$) Suppose $g[n := \bullet]_\perp \mid n = \perp$. By lemma 4.15 (ii) one has that $g[n := \bullet]_\perp \mid n \preceq g[n := \bullet][n := \perp] = g[n := \perp] \preceq g$. So $g[n := \bullet]_\perp \preceq g_\perp$. Furthermore, notice that $g_\perp \preceq g[n := \perp] \preceq g[n := \bullet]$. Hence also $g_\perp \preceq g[n := \bullet]_\perp$ so $g_\perp = g[n := \bullet]_\perp$. But then $n \notin \mathrm{I}(g)$ which is a contradiction. $\square$

4.18. Example. Let $\mathcal{P} = \{\mathbf{F}(\mathbf{G}(\perp, \mathbf{A}), \mathbf{G}(\mathbf{B}, \perp)), \mathbf{G}(\mathbf{A}, \mathbf{B})\}$. In the following graphs all occurrences of the node n are indexes; the node m is not an index.



4.19. Definition. Given an orthogonal term graph rewriting system $\mathcal{T}$. $\mathcal{T}$ is *strongly sequential* if each (closed) graph g not in normal form, has a redex that is also an index (i.e. $g \neq g_\perp \;\Rightarrow\; \delta_I(g) \neq \emptyset$).

The main part of Klop and Middeldorp (1991) is concerned with proving the decidability of strong sequentiality for TRS's. In this paper, we wil investigate another important property of strongly sequential TGRS's, namely the fact that index reduction is normalizing for these systems. This will be proved in the next section.

5. Index reduction is normalizing

5.1. Definition. Let g be a rooted graph, and $n, m \in g$.

(i) n is a *descendant* of m (notation $m \Downarrow n$) if for every path $p : r_g \rightsquigarrow n$ one has $m \in p$.

(ii) n and m are independent (notation $n \nleftrightarrow m$) if neither $m \Downarrow n$ nor $n \Downarrow m$.

The intuitive idea behind this definition is that if there exist two redexes in a graph that are independent then the reduction of one of these redexes cannot influence the presence of the other one in the resulting graph.

5.2. LEMMA. *Let g be a graph, and $n, m \in g, n \neq m$*

(i) *Let h_1, h_2 be graphs. Then*

$$m \Downarrow n \;\Rightarrow\; g[n := h_1][m := h_2] = g[m := h_2].$$

(ii) *Let E be an extension. Then*

$$n \nleftrightarrow m \;\Rightarrow\; m \in g[n := E].$$

PROOF. (i) By lemma 2.24 (i).

(ii) There exist a path p in g such that $n \notin p$. Obviously p is also a path in $g[n := E]$. □

5.3. REMARK. Let $n, m \in \delta(g)$. Then from regularity it follows that if $m \in g[n := \bullet]$ one has that $m \in \delta(g[n := \bullet])$.

5.4. LEMMA. *Let g be a graph, and $n, m \in \delta_I(g)$, $n \neq m$. Then $n \nleftrightarrow m$.*

PROOF. Suppose $n \leftrightarrow m$. Without loss of generality we can assume that $m \Downarrow n$. By lemma 2.24 (vi) one has $(g \mid m)[n := \bullet] = g[n := \bullet] \mid m$. From remark 5.3 it follows that $(g \mid m)[n := \bullet] \succeq \mathcal{P}$ and hence $g[n := \bullet] \mid m \succeq \mathcal{P}$. By lemmas 4.13(ii) and 4.15(ii), $(g[n := \bullet])_\perp \preceq g[n := \bullet][m := \perp]$. Note that $g[n := \bullet][m := \perp] = g[m := \perp]$. Since $n \notin g[m := \perp]$ also $n \notin (g[n := \bullet])_\perp$ (remark 4.7). But then n is not an index of g which is a contradiction. □

The next lemma shows that indexes do not disappear when another redex is reduced.

5.5. LEMMA. *Let g be a graph, $n \in \delta_I(g)$, and $m \in \delta(g)$ with $n \neq m$. Let $g \xrightarrow{m} h$, then $n \in \delta_I(h)$.*

PROOF. First notice that if $n \in h$ then (by lemma 3.6) $n \in \delta(h)$ and that $g[m := \perp] \preceq h$ (lemma 4.5). If $n \leftrightarrow m$ then by lemma 5.4, m is not an index. Furthermore notice that $n \Downarrow m$ so (lemma 5.2) $g[m := \perp][n := \bullet] = g[n := \bullet] \preceq h[n := \bullet]$ which implies that n is an index of h. Now assume that $m \nleftrightarrow n$. Suppose n is not an index of h. Then by lemma 4.17, $h[n := \bullet]_\perp \mid n = \perp$. By lemma 2.24(vii) it follows that $g[n := \bullet][m := \perp] \preceq h[n := \bullet]$. Using lemma 4.15(i & ii) and remark 5.3, $g[n := \bullet]_\perp \preceq g[n := \bullet][m := \perp]$ so $g[n := \bullet]_\perp \preceq h[n := \bullet]$. But then $(g[n := \bullet]_\perp) \mid n = \perp$ so n is not an index of g. This is a contradiction. □

5.6. LEMMA. *Let g be a graph and $n, m \in \delta_I(g)$. Then*

$$g \xrightarrow[I]{n} g' \xrightarrow[I]{m} h \quad \textit{and} \quad g \xrightarrow[I]{m} g'' \xrightarrow[I]{n} h.$$

PROOF. By lemma 5.5 and proposition 3.8. □

Expressed in a diagram:

$$\begin{tikzcd} & \bullet \arrow[dl, "n"', "I"] \arrow[dr, "m", "I"'] & \\ \bullet \arrow[dr, dotted, "m", "I"'] & & \bullet \arrow[dl, dotted, "n"', "I"] \\ & \bullet & \end{tikzcd}$$

5.7. LEMMA. *Let g be a graph, $m \in \delta_{NI}(g)$, and $g \xrightarrow{m} h$. Then h is not in normal form.*

PROOF. Since g is not in normal form, from sequentiality it follows that g must contain an index, say n. Apply lemma 5.5. $\square$

5.8. LEMMA. *Let g be a graph, and $n \in \delta_I(g)$. Furthermore, let $g \xrightarrow{n_1} g_1 \xrightarrow{n_2} \ldots \xrightarrow{n_k} g_k$ (hence $n_i \in \delta_I(g_{i-1}), i \leq k$) such that g_k is in normal form. Then $n = n_j$ for some $j \leq k$.*

PROOF. Apply lemma 5.5. $\square$

5.9. LEMMA. *Let $g \underset{I}{\twoheadrightarrow} h$ in k steps. Then any index reduction sequence starting with g ends with h in k steps.*

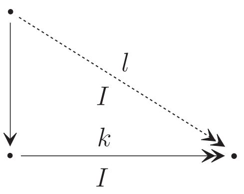
PROOF. By induction onto k: The case $k = 0$ is trivial. Let $g \xrightarrow[I]{n_1} g_1 \underset{I}{\to}^{k} h$. Moreover, let $g \xrightarrow[I]{n_2} g_2$. This situation can be visualized as follows.

$$\begin{tikzcd} & g \arrow[dl, "I"'] \arrow[dr, "I"] & & \\ g_2 \arrow[dr, dotted, "I"'] & & g_1 \arrow[dl, dotted, "I"'] \arrow[dr, two heads, "k-1", "I"'] & \\ & g_3 \arrow[rr, dotted, two heads, "k-2", "I"'] & & h \end{tikzcd}$$

Applying I.H. to g_1 one has that any reduction sequence starting with g_1 ends with h in k steps. The case $n_1 = n_2$ which means that $g_1 = g_2$ is trivial. Assume $n_1 \neq n_2$. By lemma 5.6 it follows that $g_1 \xrightarrow[I]{n_2} g_3$ and $g_2 \xrightarrow[I]{n_1} g_3$. From I.H. it follows that $g_3 \underset{I}{\to}^{k-1} h$. Thus $g_2 \xrightarrow[I]{n_1} g_3 \underset{I}{\to}^{k-1} h$ which implies that g_2 reduces to h in k steps. Applying the I.H. to g_2 one has that any reduction sequence starting with g_2 ends with h in k steps. Hence g reduces to h in $k+1$ steps. $\square$

5.10. LEMMA. *Let g be a graph, $n \in \delta(g)$, and $g \xrightarrow{n} g'$ such that $g' \underset{I}{\twoheadrightarrow} h$ in k steps where h is in normal form. Then $g \underset{I}{\twoheadrightarrow} h$ in either k or $k+1$ steps.*

The next diagram illustrates this lemma.

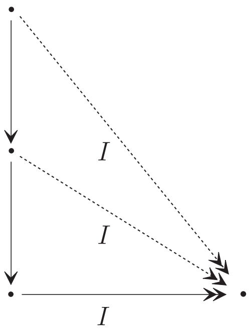


PROOF. Note that if n then the lemma holds trivially. Assume n is not an index. By sequentiality one has an $m \in \delta_I(g)$. Use induction on k. The case $k = 1$ implies that the index reduced in second step from g' to h has to be m. Then , $g \overset{m}{\underset{I}{\to}} g'' \overset{n/\!/m}{\underset{I}{\to}} h$ by lemma 3.6. The reduction sequence from g'' to h has either length 1 (in case $n/\!/m \in \delta_I(g')$) or length 0. So g reduces to h in 1 or 2 steps. Let $g \overset{n}{\to} g' \underset{I}{\twoheadrightarrow}^{k+1} h$. From lemma 5.8 it follows that $g' \underset{I}{\twoheadrightarrow}^{j-1} g_{j-1} \overset{m}{\underset{I}{\to}} g_j \underset{I}{\twoheadrightarrow}^{k-j} h$ for some j. By lemma 5.6, $g' \overset{m}{\underset{I}{\to}} g_1 \underset{I}{\twoheadrightarrow}^{k} h$. Now consider the reduction sequence $g \overset{m}{\to} g_m \overset{n/\!/m}{\to} g_1$. If $\overset{n/\!/m}{\to}$ has length 0 then g reduces to h in $k+1$ steps. Otherwise, we can apply the I.H. to $g_m \overset{n/\!/m}{\to} g_1 \underset{I}{\to}^{k} h$ providing that g reduces to h in either $k+1$ or $k+2$ steps. $\square$

This proof does not carry over to the term rewriting world. The reason for this is when exchanging the above redexes n and m, in the graph world this results in a reduction sequence that is either empty or it consists of a single reduction step. However, in the term world where redexes can get duplicated, this reduction sequence may have length greater than one. Even worse, such a reduction sequence may contain indexes as well as non-indexes.

5.11. LEMMA. *Let g be a graph, $g \twoheadrightarrow^k g' \underset{I}{\to}^+ h$, where h is in normal form. Then $g \underset{I}{\to}^+ h$*

PROOF. By using lemma 5.10 and induction on k or by the following diagram.



$\square$

5.12. PROPOSITION. *Let g be a graph, $g \twoheadrightarrow h$, and h be in normal form. Then $g \underset{I}{\twoheadrightarrow} h$.*

PROOF. If $g = h$ then it is trivial. Assume g is not in normal form. From lemma 5.7 it follows that the last step used in the reduction from g to h must be the reduction of an index. Apply lemma 5.11. $\square$

5.13. THEOREM. *Let g be a graph having a normal form h. Then, any index reduction starting with g terminates with h. Moreover, the index reduction sequences from g to h are all of the same length.*

PROOF. Follows directly from lemma 5.9 and proposition 5.12. $\square$

6. Conclusion

The proof of the Church-Rosser property for a-cyclic graphs presented in Barendregt et al. (1987) and the proof that index reduction is normalising for strongly sequential TGRS's illustrate that reasoning within the framework of term graph rewriting is very well possible. Sections 2 and 3 provide a bases with which all the results and necessary notions can be given in an elegant way. A nice example of such a notion is $\perp$-reduction as defined in section 4 used to prove the normalizing property of index reduction. Hence, we have the strong feeling that this paper will encourage readers not only to use graph rewriting as an implementation issue but also to consider and to use it as a fully fledged formalism.

References

Barendregt, H.P., M.C.J.D. van Eekelen, J.R.W. Glauert, J.R. Kennaway, M.J. Plasmeijer and M.R. Sleep (1987). Term graph reduction, *Proc. of Parallel Architectures and Languages Europe (PARLE)*, Eindhoven, The Netherlands, Springer Verlag, LNCS 259 II, pp. 141–158.

Barendsen, Erik and Sjaak Smetsers (1992). Graph rewriting and copying, *Technical Report 92-20*, University of Nijmegen.

Huet, G. and J.J. Lévy (1979). Call by need computations in non-ambiguous linear term rewriting systems, *Technical Report 359*, INRIA.

Kennaway, J.R. (1991). Graph rewriting in some categories of partial homomorphisms, *in:* Kreowski Ehrig and Rozenberg (eds.), *Proc. of 4th International Workshop on Graph-Grammars and Their Application to Computer Science*, Springer Verlag, LNCS 532, pp. 490–504.

Kennaway, J.R., J.W. Klop, M.R. Sleep and F.J. de Vries (1990). Transfinite reductions in orthogonal term rewriting systems, *Technical Report CS9041*, CWI.

Klop, J.W. and A. Middeldorp (1991). Sequentiality in orthogonal term rewriting systems, *Journal of symbolic computation.*

Raoult, J.C. (1984). On graph rewritings, *Theoretical Computer Science.*