1992

J.R. Kennaway, J.W. Klop, M.R. Sleep, F.-J. de Vries

On the adequacy of graph rewriting for simulating term rewriting

Computer Science/Department of Software Technology Report CS-R9204 January

CWI is the research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a non-profit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands organization for scientific research (NWO).

On the Adequacy of Graph Rewriting for Simulating Term Rewriting

J.R. KENNAWAY [1], J.W. KLOP [2,3], M.R. SLEEP [4] & F.J. DE VRIES [5]

(1,4) *School of Information Systems, University of East Anglia*

(2,5) *CWI*

(3) *Department of Mathematics and Computer Science, Vrije Universiteit Amsterdam*

(1) jrk@sys.uea.ac.uk, (2) jwk@cwi.nl, (4) mrs@sys.uea.ac.uk, (5) ferjan@cwi.nl

ABSTRACT Several authors have investigated the correspondence between graph rewriting and term rewriting. Almost invariably they have considered only acyclic graphs. Yet cyclic graphs naturally arise from certain optimisations in implementing functional languages. They correspond to infinite terms, and their reductions correspond to transfinite term reduction sequences, which have recently received detailed attention. We first establish a close correspondence between finitary acyclic graph rewriting and finitary term rewriting, and between finitary cyclic graph rewriting and transfinite rational term rewriting.

Surprisingly, the correspondence breaks down for general transfinite rewriting. We present an example showing that transfinite term rewriting is strictly more powerful than transfinite graph rewriting.

1985 Mathematics Subject Classification: 68Q50.

1987 CR Categories: F4.1, F4.2.

Keywords and Phrases: Orthogonal term rewriting, graph rewriting, functional programming.

Note: All authors were partially sponsored by SEMAGRAPH, ESPRIT Basic Research Action 3074. The first author was also partially supported by a SERC Advanced Fellowship, and by SERC grant no. GR/F 91582.

Contents

Report CS-R9204
ISSN 0169-118X
CWI
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

1. Introduction

Several authors have written on the correctness of graph rewriting implementations of term rewriting(e.g. Sta80, Bar87, Far89]). Almost all of them restrict attention to acyclic graphs and orthogonal rule systems. Only Farmer and Watro study the cyclic Y-combinator.

In [Bar87] it is proved, for any orthogonal term rewrite system and its related graph rewrite system, that if an acyclic graph g unravels to a term t, then g has a normal form by acyclic graph rewriting iff t has a normal form by tree rewriting, and the normal form of g unravels to the normal form of t.

There are a number of things one might want to improve in this result. Firstly, it refers to the final targets of the computations only, whereas there is in fact a close correspondence between the reduction sequences themselves. Secondly, the possession of normal forms is in general undecidable. Thirdly it is restricted to acyclic graph rewriting. Cyclic graphs arise naturally as a result of certain optimisations in functional language implementation. Such graphs unravel to infinite terms, and their reduction sequences unravel to transfinite term reduction sequences. Infinitary term rewriting has recently received detailed attention [Far89, Der89a, Der89b, Der90a, Ken90, Ken91a].

In this paper we give a precise analysis of the relationship between cyclic term graph rewriting and infinitary term rewriting, for orthogonal rewrite systems. For non-orthogonal systems, graph rewriting and term rewriting differ significantly, although some of our results still hold.

Terms and computations of a finite acyclic graph rewriting system can be unravelled into terms and computations of a finitary term rewriting system. We will give an abstract definition of an "adequate mapping" between abstract reduction systems which captures the properties of the unravelling mapping. We will prove that

- *Finite acyclic graph rewriting is adequate for finite term rewriting.*

The quoted result of [Bar87] is a corollary.

To extend this to finitary cyclic graph rewriting, we have to consider infinitary term rewriting. Abstract reduction systems form a semantics only for finitary rewriting. By use of the weighted metric abstract reduction systems of [Ken91c] as a semantics for infinitary rewriting we strengthen the adequacy concept so that it applies to finitary cyclic graph rewriting. We will prove that:

- *Finite graph rewriting is adequate for rational term rewriting.*

Finally we will show by means of an counterexample that

- *Infinite graph rewriting is* not *adequate for infinite term rewriting.*

It should be noted that our present definition of an adequate mapping of one system to another adds to the abundance of concepts of simulation, in term rewriting (e.g. [Bar84, O'Do85]), complexity theory (for an overview see [vEB90]) or programming languages [Mit91].

2. Term rewriting

We briefly recall the definition of finitary and infinitary orthogonal term rewriting. The reader familiar with [Ken90a] or [Ken91a] can skip this section, and use it for later reference. General introductions to term rewriting are [Der90b] and [Klo91].

2.1. Finitary term rewriting systems

The theory of term rewriting has mainly been concerned with finitary term rewriting systems. In finitary term rewriting systems one considers finite terms only. Let us briefly recall definitions and notation. For more details the reader is referred to the extensive survey papers [Der90b] and [Klo91].

A *finitary term rewriting system* over a signature Σ is a pair $(Ter(\Sigma),R)$ consisting of the set $Ter(\Sigma)$ of finite terms over the signature Σ and a set of rewrite rules $R \subseteq Ter(\Sigma)\times Ter(\Sigma)$.

The *signature* Σ consists of a countably infinite set Var_Σ of variables (x,y,z,...) and a non-empty set of function symbols (A,B,C,...,F,G,...) of various finite arities ≥ 0. Constants are function symbols with arity 0. The set $Ter(\Sigma)$ of *finite* terms (t,s,...) over Σ can be defined as usual: the smallest set containing the variables and closed under function application.

The set O(t) of *occurrences* in t is defined by induction on the structure of t as follows: $O(t) = \{\langle\,\rangle\}$ if t is a variable and $O(t) = \{\langle\,\rangle\} \cup \{\langle i,u\rangle \mid 1\leq i\leq n \text{ and } \langle u\rangle \in O(t_i)\}$ if t is of the form $F(t_1,...,t_n)$. If $u\in O(t)$ then the subterm t/u at occurrence u is defined as follows: $t/\langle\,\rangle = t$ and $F(t_1,...,t_n)/\langle i,u\rangle = t_i/u$. The *depth* of a subterm of t at occurrence u is the length of u.

Contexts are terms in $Ter(\Sigma\cup\{\square\})$, in which the special constant $\square$, denoting an empty place, occurs exactly once. Contexts are denoted by C[ ] and the result of substituting a term t in place of $\square$ is $C[t]\in Ter(\Sigma)$. A *proper* context is a context not equal to $\square$.

Substitutions are maps $\sigma: Var_\Sigma \to Ter(\Sigma)$ satisfying $\sigma(F(t_1,...,t_n)) = F(\sigma(t_1),...,\sigma(t_n))$.

The set R of *rewrite rules* contains pairs (l,r) of terms in $Ter(\Sigma)$, written as $l \to r$, such that the left-hand side l is not a variable and the variables of the right-hand side r are contained in l. The result l^σ of the application of the substitution of σ to the term l is called an instance of l. A *redex* (reducible expression) is an instance of a left-hand side of a rewrite rule. A *reduction step* $t \to s$ is a pair of terms of the form $C[l^\sigma] \to C[r^\sigma]$, where $l \to r$ is a rewrite rule in R. Concatenating reduction steps we get either a *finite reduction* $t_0 \to t_1 \to ... \to t_n$, which we also denote by $t_0 \to_n t_n$, or an infinite reduction $t_0 \to t_1 \to ...$. A *normal form* is a term without redexes.

Finally we can give the definition of an orthogonal term rewriting system.

2.1.1. DEFINITION. Let R be a finitary term rewriting system.

(i) R is *left-linear* if no variable occurs more than once in the left hand side of a rewrite rule of R.

(ii) R is *non-overlapping* if for any two (not necessarily distinct) rules of r, with left-hand sides s and t, any occurrence u in t, and any substitutions σ and τ it holds that if $(t/u)^\sigma = s^\tau$ then either t/u is a variable or t and s are left-hand sides of the same rewrite rule and u is the empty occurrence $\langle\,\rangle$, the position of the root. If for some pair of rules this condition fails to hold, the rules are said to *conflict*.

(iii) R is *orthogonal* if R is both left-linear and non-overlapping.

EXAMPLE. The rules $F(G(x)) \to H$ and $G(K(x)) \to K$ conflict with each other. The rule $F(F(x)) \to G$ conflicts with itself (take $u = \langle 1 \rangle$).

2.2. Infinitary term rewriting systems

An *infinitary term rewriting system* over a signature Σ is a pair $(Ter^\infty(\Sigma),R)$ consisting of the set $Ter^\infty(\Sigma)$ of finite and infinite terms over Σ and a set of rewrite rules $R \subseteq Ter(\Sigma) \times Ter^\infty(\Sigma)$. Note that we require that the left-hand side of a rule is a finite term.

It takes some elaboration to define the set $Ter^\infty(\Sigma)$ of *finite and infinite terms* precisely. Finite terms may be represented as finite trees, well-labelled with variables and function symbols. Well-labelled means that a node with $n \geq 1$ successors is labelled with a function symbol of arity n and that a node with no successors is labelled with either a constant or a variable. Now *infinite terms* are infinite well-labelled trees with nodes at finite distance to the root. Substitutions, contexts, reduction steps and normal forms generalize unchanged to the set of infinitary terms: e.g. a normal form is a (possibly infinite) term without redexes.

The definition of orthogonality for finitary term rewriting systems extends verbatim to infinitary systems.

2.3. Strongly converging reductions

Recall that the set of terms over a signature is a complete metric space (see for instance [Arn80]). The metric (in fact, an ultrametric) is given by $d(t,s) = 0$ if t and s are equal, and is otherwise 2^{-k}, where $k \in \mathbb{N}$ is the largest number such that the labels of all nodes of s and t at depth less than or equal to k are equally labelled. Now consider the following rule systems and reduction sequences.

(i) $A \to B \to A \to B \to \ldots$, in a TRS with rules $A \to B$ and $B \to A$.

(ii) $D(E) \to D(S(E)) \to D(S(S(E))) \to \ldots$, in a TRS with rule $D(x) \to D(S(x))$.

(iii) $C \to S(C) \to S(S(C)) \to \ldots$, in a TRS with rule $C \to S(C)$.

Example (i) is a diverging reduction sequence. Example (ii) is a *weakly converging* reduction with limit $D(S^\omega)$. Example (iii) is *strongly converging* with limit S^ω. The distinction between the two types of convergence is that a weakly converging reduction need only converge in the topological sense, while a strongly converging reduction must satisfy the additional requirement that the depth of reduced redexes tends to infinity.

In [Ken90a] we have shown that strongly converging transfinite reduction has a more well-behaved theory than weakly converging transfinite reduction. For this reason, we here consider only the former type of transfinite reduction.

We write $t \to^\omega s$ (resp. $t \to^{\leq\omega} s$) to denote a strongly converging reduction of length ω (resp. at most ω) from t to s. Concatenation of (a possibly infinite number of) strongly converging reductions gives reductions of any countable ordinal length. For such a reduction of length α to be strongly converging we require that, considered as a mapping from $\alpha+1$ to terms, it be continuous with respect to the usual topology on ordinals, that the depth of reduced redexes tends to infinity, and that every proper initial segment also be strongly converging. (The essential content of the last condition is that the depth of reduced redexes tends to infinity at every limit ordinal $\lambda \leq \alpha$.) We write $t \twoheadrightarrow^{\alpha} s$ for a strongly converging reduction of ordinal length α.

While the notion of a reduction sequence longer than ω may seem devoid of computational content, the Compressing Lemma of [Ken90a] shows that if t reduces to s by a strongly converging reduction of length greater than ω, it also reduces to s by a sequence of length at most ω.

Finally, $t \twoheadrightarrow^{\infty} s$ denotes a strongly converging reduction of any finite or infinite length.

3. Graph rewriting

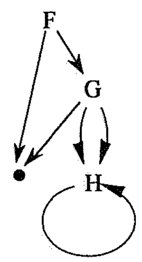
Graph rewriting is a common method of implementing term rewrite languages [Pey87]. It relies on the basic insight, that when a variable occurs many times on the right-hand side of a rule, one need only copy pointers to the corresponding parts of the term being evaluated, instead of making copies of the whole subterm. The reader familiar with graph rewriting may skip this section. Note however that we allow cyclic graphs; these correspond to certain infinite terms.

3.1. DEFINITION. A *graph* g over a signature $\Sigma = (\mathcal{F}, \mathcal{V})$ is a quadruple (*nodes*(g), *lab*(g), *succ*(g), *roots*(g)), where *nodes*(g) is a finite or infinite set of nodes, *lab*(g) is a function from a subset of the nodes of g to $\mathcal{F}$, *succ*(g) is a function from the same subset to tuples of nodes of g, and *roots*(g) is a tuple of (not necessarily distinct) nodes of g. Furthermore, every node of g must be *accessible* (defined below) from at least one root. Nodes of g outside the common domain of *lab*(g) and *succ*(g) are called *empty*. □

3.2. DEFINITION. A *path* in a graph g is a finite or infinite sequence a,i,b,j,... of alternating nodes and integers, beginning and (if finite) ending with a node of g, such that for each m,i,n in the reduction, where m and n are nodes, n is the i'th successor of m. The length of the path is the number of integers in it. If the path starts from a node m and ends at a node n, it is said to be a path from m to n. If there is a path from m to n, then n is said to be *accessible from* m. When this is so, the *distance* of n from m is the length of a shortest path from m to n. □

We may write $n:F(n_1,\ldots,n_k)$ to indicate that *lab*(g)(n) = F and *succ*(g)(n) = $(n_1,\ldots,n_k)$. A finite graph may then be presented as a list of such *node definitions*.

For example, the graph x:F(y,z), z:G(y,w,w), w:H(w) represents the following graph:



In such pictures, we may omit the names x, y, z, etc., as their only function in the textual representation is to identify the nodes. In particular, x, y, z, etc. do not represent variables: variables are represented by empty nodes. Different empty nodes need only be distinguished by the fact that they are different nodes; we do not need any separate alphabet of variable names. Multiple references to the same variable in a term are represented in a graph by multiple references to the same empty node.

The tabular description demonstrated above may conveniently be condensed, by nesting the definitions; for example, another way of writing the same graph is F(y,z:G(y,w,w:H(w))).

In general a graph may have more than one root. We will only use graphs with either one root (which represent terms) and graphs with two roots (which represent term rewrite rules).

3.3. DEFINITION. A *graph homomorphism* from a graph g to a graph h is a function f from the nodes of g to the nodes of h, such that for all nodes n in the domain of *lab*(g), *lab*(h)(f(n)) = *lab*(g)(n), and *succ*(h)(f(n)) = *succ*(g)(n). □

Note that a graph homomorphism is not required to map the roots of its domain to the roots of its codomain.

On graphs one can define many general graph rewrite mechanisms. We are concerned with one particular form: term graph rewriting.

3.4. DEFINITION. A *term graph* is a graph with one root. □

3.5. DEFINITION. A *term graph rewrite rule* is a graph with two, not necessarily distinct, roots (called the *left and right* roots), in which every empty node is accessible from the left root, and the subgraph containing those nodes accessible from the left root is a finite tree. The *left* (resp. *right*) *hand side* of a term graph rewrite rule r is the subgraph consisting of all nodes and edges accessible from the left (resp. right) root: notation *left*(r) (resp. *right*(r)). □

3.6. DEFINITION. A *redex* of a term graph rewrite rule r in a graph g is a homomorphism from the left-hand side of r to g. The *occurrence* of the redex is the minimal occurrence of the node of g to which the left root is mapped. The *depth* of a redex is the length of the occurrence. □

The result of *reducing* a redex of the rule r in a graph g at occurrence u is the graph obtained by the following construction.

3.7. CONSTRUCTION. (i) Construct a graph h by adding to g a copy of all nodes and edges of r not in *left*(r). Where such an edge has one endpoint in *left*(r), the copy of that edge in h is connected to the image of that endpoint by the homomorphism.

(ii) Let n_l be the node of h corresponding to the left root of r, and n_r the node corresponding to the right root of r. (These are not necessarily distinct.) In h, replace every edge whose target is n_l by an edge with the same source and target n_r, obtaining a graph k. The root of k is the root of h, unless this is n_l, otherwise it is n_r.

(iii) Remove all nodes which are not accessible from the root of k. The resulting graph is the result of the rewrite. □

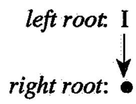
We have now the ingredients to give the general definition of a Term Graph Rewrite System.

3.8. DEFINITION. Let Σ be a signature. A Term Graph Rewrite System (GRS for short) is a pair (G(Σ),**R**) where G(Σ) is the set of graphs for the signature Σ, and **R** a set of term graph rewrite rules for the signature Σ. □

Having defined term graph rewriting and the notion of depth on term graphs, the concepts of normal form, infinitary rewriting, orthogonality, etc. carry over to term graphs.

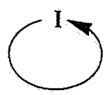
As an example consider the following rule, given in both textual and pictorial forms:

$$I(x) \rightarrow x$$



and the graph

$$a{:}I(a)$$



It is clear that the graph is a redex of the rule. In [Ken90a] we have called this redex "circular I". It reduces to itself. Circular I is one instance of a class of redexes having the same behaviour, the circular redexes.

3.9. DEFINITION. (i) A redex of a rule r is *circular* if the roots of r are distinct and the homomorphism from *left*(r) to g maps both roots of r to the same node. (This can only happen if the right root of r is accessible from the left root.)

(ii) A rule is a *collapse* rule if its right root is a variable. □

An example of a collapsing rule is x:Head(Cons(y,z)) → y. An example of a non-collapsing rule which admits circular redexes is x:F(y:F(z)) → y. Note that this rule conflicts with itself: it has two overlapping redexes in the graph F(F(F(G))). A circular redex of this rule is x:F(x).

3.10. PROPOSITION. [Ken90a] *In an orthogonal term graph rewrite system, a rule has a circular redex iff it is a collapse rule.* □

For a different treatment of circular redexes see [Ari92].

From now on we will consider term graphs and term graph rewriting only, and often we will simply call them graphs and graph rewriting.

4. Unravelling

Unravelling transforms (term) graphs to terms. Both graphs and computations can be unravelled. In this section we will prove that for any term graph rewrite system, if g reduces to g' by strongly convergent reduction, then U(g) similarly reduces to U(g') in the unravelled system. This is so even for non-orthogonal systems.

4.1. DEFINITION. The *unravelling* U(g) of a graph g is the term representation of the following forest. The nodes of U(g) are the paths of g which start from any of its roots. Given a node a,i,b,j,...y of U(g), if y is a nonempty node of g, then this node of U(g) is labelled with the function symbol *lab*(g)(y), and its successors are all paths of the form a,i,b,j,...y,n,z, where z is the n'th successor of y in g. If y is empty, then it is labelled with a variable symbol, a different symbol being chosen for every empty node of g.

Note that a cyclic graph will have an infinite unravelling. For example, the unravelling of the graph shown in the previous picture is the term $F(y,G(y,H^{\omega},H^{\omega}))$.

It is easy to see that for a term graph g, U(g) is a term, and for a graph rewrite rule r, U(r) is a term rewrite rule. We can also apply the notion of unravelling to a whole rewrite system.

4.2. DEFINITION. The *unravelling* of a GRS (G(Σ),**R**) is the TRS $(\mathrm{Ter}^{\infty}(\Sigma),U(\mathbf{R}))$whose rules U(**R**) are the unravellings of the rules in **R**. This TRS is also denoted by U(G(Σ),**R**); its set of terms is U(G).

So, given a signature Σ the operator U transforms GRS's over Σ into TRS's over Σ. Note that a GRS is orthogonal if and only if its unravelling is orthogonal.

4.3. PROPOSITION. *There is a homomorphism from* U(g) *to* g *which takes the root of* U(g) *to the root of* g. □

The homomorphism is obtained by mapping each node of U(g) (which is a finite path of g) to its final element. If g is acyclic, this is clearly the only homomorphism from U(g) to g, but if g is cyclic there can be more than one: for example, if g = x:A(A(x)), there are two.

The following proposition is not hard to prove:

4.4. PROPOSITION. *A graph g in the GRS* (G(Σ),**R**) *is a normal form iff its unravelling* U(g) *is a normal form in* $(\mathrm{Ter}^{\infty}(\Sigma),U(\mathbf{R}))$. □

4.5. THEOREM. *Let* g → g' *in a GRS. Then* $U(g) \to_{\leq\omega} U(g')$ *in the corresponding TRS. Moreover, the depth of every redex reduced in the term sequence is at least equal to the depth of the redex reduced in* g.

PROOF. Let r be the rule that was applied to reduce g to g', and u the occurrence at which it was applied. We need to distinguish two cases.

If the redex is circular, then it reduces to itself, and g'=g. Clearly, U(g) reduces to U(g') by the empty reduction sequence. The condition on depths is trivially satisfied.

Otherwise, we shall show that there is a redex of U(r) at every occurrence in U(g,u), that all these redexes can be reduced by a strongly convergent reduction, and that the limit of this reduction is U(g').

It is clear that there is a redex of U(r) at every occurrence in U(g,u). If U(g,u) is finite, then the theorem holds as shown in [Bar87]. Otherwise, suppose U(g,u) is infinite.

Let the members of U(g,u), ordered by depth, be $u_1, u_2, \ldots$, with depths $d_1, d_2, \ldots$ Consider the effect of reducing the redex at u_1.

Those redexes at occurrences incomparable with u_1 will still be present afterwards, and at the same occurrences. In particular, all redexes previously at the same depth as u_1 will be at the same depth afterwards.

Redexes at occurrences which extend u_1 must be at greater depth. We shall show that after reducing the redex at u_1, the depths of the residuals of such redexes are still greater than d_1.

Suppose this were not the case. If a redex is at $u_i > u_1$, then after reduction at u_1 its residuals must still be within the subterm at u_1. If the redex formerly at u_i has a residual at depth d_1, that residual must therefore be at u_1. But this is only possible if the right-hand side of the rule is a variable, and the subterm matched to that variable by the redex at u_1 is the subterm at u_i. But both redexes originate from the same redex of the original graph. Therefore the graph redex was a cyclic collapsing redex, a case we have already eliminated.

Therefore, after reduction at u_1, for every redex at depth greater than d_1, all its residuals by u_1 are still at depth greater than d_1. Since there can only be finitely many redexes at depth d_1, reduction of all of them leaves redexes only at depths greater than d_1. Repeating the argument for the newly shallowest redexes constructs a strongly convergent reduction.

When all the remaining redexes are at depths greater than some depth d, then the term t_d at that point agrees with U(g') down to depth d. Thus the distance between t_d and U(g') is less than 2^{-d}. Therefore the limit of the reduction of terms t_d as d tends to infinity is U(g').

Finally, the condition on the depths of the term reduction steps is immediate. □

Note that we do not need to suppose that the system is orthogonal. If the graph is cyclic, it is possible for U(r) to contain conflicting redexes. But so long as they are reduced in outermost-first order, as is done in the above proof, the tree U(g') will always be obtained. An example is the rule F(F(x)) → G and the graph y:F(y). This reduces to G by a single graph rewrite. The unravelled term F^ω contains infinitely many redexes. If the second-outermost redex is reduced first, one obtains the normal form F(G); but if one instead first reduces the outermost redex one obtains G, as described by the theorem.

4.6. COROLLARY. *Let* $g \twoheadrightarrow_\alpha g'$ *in a GRS for some infinite ordinal* α. *Then* $U(g) \twoheadrightarrow_{\le\alpha} U(g')$ *in the corresponding TRS.*

PROOF. By applying theorem 4.5 to each step in the reduction $g \twoheadrightarrow_\alpha g'$. This gives a reduction sequence composed of subsequences of the form $U(g_\gamma) \twoheadrightarrow_{\le\omega} U(g_{\gamma+1})$, each of which is strongly convergent. That the concatenation of these is strongly convergent also follows from theorem 4.5. □

5. Adequate mappings between abstract reduction systems

When do we say that a graph rewrite system $(G,\rightarrow)$ is an implementation of term rewrite system $(T,\rightarrow)$? The intuitive answer is (cf. [vEB90]): everything which T can do to any term can be performed by G as well, modulo the unravelling of graphs to terms.

Before studying this particular example we give an abstract definition of implementation in the context of abstract reduction systems.

5.1. DEFINITION. An abstract reduction system, or ARS $(A,\rightarrow)$ is a set A with a binary relation $\rightarrow$ on A. □

One can make the following observation. An abstract reduction system becomes a *category* by taking the elements of A as objects and the finite reductions (including reductions of length 0) as morphisms. The observation allows us to borrow names from category theory for some concepts. For instance, a *functor* from $(A,\rightarrow)$ to $(B,\rightarrow)$ is a function $f:A\rightarrow B$ such that for any a,a' in A if $a \rightarrow^* a'$ in $(A,\rightarrow)$ then $f(a) \rightarrow^* f(a')$ in $(B,\rightarrow)$.

The existence of a functor f: $(A,\rightarrow) \rightarrow (B,\rightarrow)$ expresses that $(B,\rightarrow)$ can mimic all reductions that $(A,\rightarrow)$ can make. We shall see that the unravelling operation has this property. However, the property is rather weaker than the relationship which we wish to demonstrate between graph and term rewriting. That relationship is captured by the following abstract definition.

5.2. DEFINITION. A functor U: $(G,\rightarrow) \rightarrow (T,\rightarrow)$ is an *adequate mapping* if:

(i) U is surjective.

(ii) $g\in G$ is a normal form iff U(g) is a normal form.

(iii) For $g\in G$ and $t\in T$, if $U(g) \rightarrow^* t$ then there is a $g'\in G$ such that $g \rightarrow^* g'$ and $t \rightarrow^* U(g')$.

□

We can formulate the last condition differently:

5.3. DEFINITION. (i) Given a set X and a binary relation $\rightarrow$ on X, Y is a *cofinal* subset of X if for all x in X there exists y in Y such that $x \rightarrow^* y$.

(ii) Given a rewrite order $(A,\rightarrow)$, and a member a of A, the rewrite order $a\downarrow A$ (the *restriction* of A to a) is the rewrite order whose objects are the objects of A which a is reducible to, together with the restriction of $\rightarrow$ to this subset. □

Condition (iii) of definition 5.2 then says that for all $g\in G$, U maps $g\downarrow G$ to a cofinal subset of $U(g)\downarrow T$.

We refer to the three conditions of definition 5.2 respectively as surjectivity, preservation of normal forms, and cofinality.

We shall briefly mention some other relationships between rewrite systems which have been defined in the literature. There is implicit in theorem 7.3.10 of [Bar84] (concerning the relationship between lambda calculus and combinatory logic) a notion of simulation which involves a pair of functions f:A→B and g:B→A, such that $A\mathcal{f}a=g(f(a))$, $B\mathcal{f}b=f(g(b))$, $A\mathcal{f}a=a' \Rightarrow B\mathcal{f}f(a)=f(a')$, and $B\mathcal{f}b=b' \Rightarrow A\mathcal{f}f(b)=f(b')$. O'Donnell [O'Do85] gives a rather complicated definition of simulation of B by A, in which he is concerned to place bounds on the amount of work which must be done in A to simulate a fixed amount of work in B. [Bar87] defines concepts called the graph-reducibility and tree-reducibility of a mapping from one system to another; it is easy to show that an adequate mapping as defined here implies both these properties.

All of the concepts of simulation, implementation, etc. of one rewrite system in another, which we have encountered in the literature, however abstractly defined, were motivated by the properties of some particular mappings between systems, and actually have little general applicability. Our notion of adequate mapping is no exception: it is designed to describe the relationship between term and graph rewriting. It remains to be seen whether the same concept is useful in other situations.

6. Adequacy of acyclic graph rewriting for finite term rewriting

We now prove a result which is well-known, but our proof will later be used as a basis for our more general result on cyclic graph rewriting and rational term rewriting. From here on we restrict attention to orthogonal rewrite systems. Finitary orthogonal term rewrite systems are known to satisfy the Church-Rosser property; it is also satisfied by finitary orthogonal term graph rewrite systems (cf. [Ken91b]).

6.1. DEFINITION. A *development* of a set of redexes R of a term t of some given TRS is a reduction sequence in which each step reduces a descendant (or residual) of a member of R by the preceding steps. It is a *complete* development if R has no residuals by the whole sequence. By GK(t) we denote the unique term—if there is one—resulting from a complete development of all redexes in t (i.e. *Gross-Knuth* reduction). Similarly we define GK(g) for a graph g. □

The only reason GK(t) may not exist is if the term is infinite and its redexes cannot all be reduced in a strongly convergent manner.

6.2. THEOREM. *In an orthogonal rule system,* GK(U(g)) *exists iff* GK(g) *exists and* g *contains no circular redexes. They always exist if* g *is finite and acyclic. When they exist,* U(GK(g)) = GK(U(g)).

PROOF. Immediate from theorem 4.6. □

6.3. THEOREM. *Finite orthogonal acyclic graph rewriting is adequate for finite orthogonal term rewriting, via the unravelling mapping.*

PROOF.

(i) Surjectivity is trivially satisfied, since every finite term is a finite acyclic graph.

(ii) Preservation of normal forms is proposition 4.4.

(iii) For cofinality, let $t_0 = U(g_0)$, and let $t_0 \rightarrow^n t_n$ be a reduction sequence. We shall construct a reduction diagram of the form of Figure 1, in which the given reduction sequence forms the top line, and each t'_i is equal to $U(g_i)$, demonstrating cofinality.

We are given $t_0 = U(g_0)$, and define $t'_0 = t''_0 = t_0$. Suppose the diagram has been constructed up to terms t_{i-1}, t'_{i-1}, t''_{i-1}, g_{i-1} and the reduction sequences connecting them. Suppose also that $t'_{i-1} = U(g_{i-1})$. We extend the diagram thusly:

• $g_{i-1} \rightarrow^* g_i$ is a complete development of all the redexes of g_{i-1} (i.e. a Gross-Knuth reduction).

• $t'_{i-1} \rightarrow^* t''_i$ is a complete development of the residuals of $t_{i-1} \rightarrow t_i$ over $t_{i-1} \rightarrow^* t''_{i-1} \rightarrow^* t'_{i-1}$.

• $t''_i \rightarrow^* t'_i$ is an extension of this complete development to a complete development of all the redexes of t'_{i-1}; such an extension exists by orthogonality.

• $t_i \rightarrow^* t''_i$ is the projection of the sequence $t_{i-1} \rightarrow^* t''_{i-1} \rightarrow^* t'_{i-1}$ over $t_{i-1} \rightarrow t_i$ (this also depends on orthogonality).

By theorem 6.2, it follows that $t'_i = U(g_i)$, as required. □

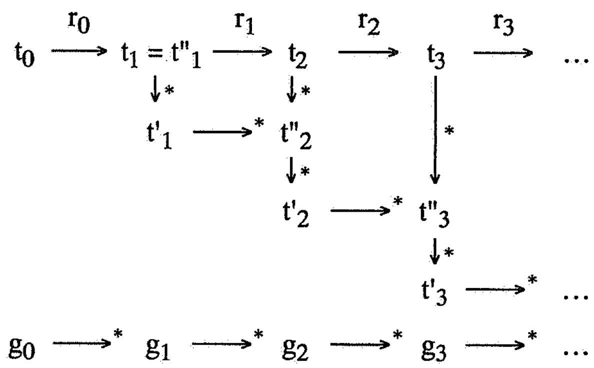


Figure 1

Here is an example of how the theorem fails for non-orthogonal rule systems.

Rules: $F(x) \rightarrow A(x,x)$, $B \rightarrow C$, $B \rightarrow D$.

Term reduction sequence: $F(B) \rightarrow A(B,B) \rightarrow A(C,B) \rightarrow A(C,D)$.

By graph reduction, F(B) can be reduced only to A(x:B,x), A(x:C,x), or A(x:D,x). However, the term A(C,D) cannot be reduced to the unravellings of any of these graphs.

7. Infinite graph rewriting—a counterexample to adequacy

The last theorem fails for infinite graph reduction, even for orthogonal systems. Firstly, Gross-Knuth reduction is not necessarily strongly convergent, so the method of proof of the above theorem cannot be duplicated. One might try to rectify this by choosing some smaller set of redexes to reduce in each g_i. However, they must include at least all those redexes, any of whose unravellings has a residual which is reduced in the term sequence. But even this restricted subset may not be strongly reducible. Such considerations lead to the construction of an actual counterexample.

7.1. COUNTEREXAMPLE.

Symbols: The natural numbers

Rules: For each natural number **n**: $\mathbf{n}(x,y) \to y$

Graph: $g_0 = a_0{:}0(a_0,a_1),\ a_1{:}1(a_1,a_2),\ \ldots$

g_0 is an infinite chain of nodes, each labelled with a different integer, pointing to itself with its first argument, and pointing to the next node with its second argument. $t_0 = U(g_0)$ is an infinite binary tree, with root labelled 0, and where each node labelled **n** has left and right descendants labelled **n** and **n+1** respectively. t_0 is illustrated in Figure 2. The circled nodes in the figure indicate where we are going to perform reductions: at every node which is the left child of its parent, whose right sibling is labelled with a greater integer, and which does not have any ancestor satisfying these conditions.

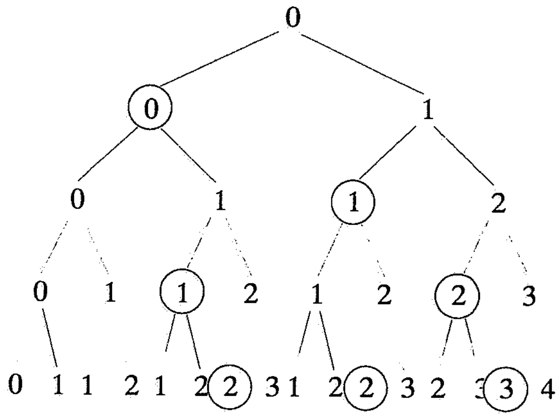


Figure 2

If we reduce all the marked redexes, in increasing order of depth, it is easy to see that the resulting sequence strongly converges to Figure 3 (for example, by observing that reduction of any outermost member of the marked set of nodes has the same effect as incrementing every integer at or below that node).

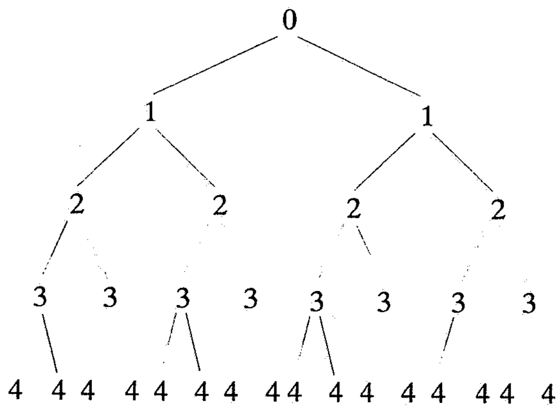


Figure 3

In Figure 3, for every **n**, there are only finitely many nodes of the form **n**(–,–). However, every graph which g_0 can be reduced to, by finite or infinite reduction, contains cycles which unravel to give an infinite number of such nodes, for some **n**. Therefore there is no graph g to which g_0 can be reduced such that $U(g) = t_\omega$. □

We thus see that transfinite orthogonal graph rewriting is not adequate for transfinite orthogonal term rewriting, as the cofinality condition fails.

8. Adequate mappings between metric abstract reduction systems

We will be discussing transfinite term rewriting in the sequel, and we must therefore consider a notion of ARS in which we can model the transfinite strongly converging reductions. We consider a slight modification of the Transfinite ARSs of [Ken91c].

We start by considering weighted abstract reduction systems.

8.1. DEFINITION. A *weighted abstract reduction system* (WARS) is a family of ARSs on the same underlying set with reductions labeled by positive reals: $(A,\rightarrow_s)_{s\in S}$ for some $S \subseteq \{r\in\mathbb{R} \mid r \geq 0\}$. □

Picturesque explanation: imagine A is a plane on which there is a grasshopper. If the grasshopper jumps from a to b, and reaches a height of r above the plane, then we have denoted this by $a \rightarrow_r b$.

We can consider limits of infinite sequences of jumps when we introduce a metric on the underlying space.

8.2. DEFINITION. A *metric abstract reduction system* (MARS) is a WARS $(A,\rightarrow_s)_{s\in S}$ where the underlying set A is a metric space. We denote its distance measure by d. □

8.3. DEFINITION. A *reduction step* of an ARS A is an instance of one of its relations $a \rightarrow_s b$.

□

The next definition uses the analogy of the jumping grasshopper.

8.4. DEFINITION. Let $a \rightarrow_s b$ be a reduction step of the MARS $\langle(A,\rightarrow_s)_{s\in S},d\rangle$. The *height* of the step $a \rightarrow_s b$ is s. The *length* of the step is d(a,b). □

We now fix some terminology for sequences of arbitrary ordinal length.

8.5. DEFINITION. A *transfinite reduction* of elements of a MARS $\langle(A,\rightarrow_s)_{s\in S},d\rangle$ is a function a: $\alpha \rightarrow A$ for some ordinal α, such that for any $\beta<\alpha$ there is an r in S such that $a_\beta \rightarrow_r a_{\beta+1}$. □

8.6. DEFINITION. A transfinite reduction sequence a: $\alpha \rightarrow A$ is *strongly converging* to a_α (notation $a_0 \rightarrow_\alpha a_\alpha$) if

(i) the sequence $\{ a_\beta \mid \beta \leq \alpha \}$ of elements of A is continuous,

(ii) the sequence $\{ s_\beta \mid \beta < \alpha \}$ of positive reals converges to zero at each limit ordinal $\leq \alpha$,

(iii) if $\alpha = \beta+1$ then $a_\beta \rightarrow_s a_{\beta+1}$ for some s. □

After these preliminaries we update a number of concepts to deal with strongly converging reductions in MARSs, instead of merely finite reductions in ARSs.

8.7. DEFINITION. (i) Given sets X and Y, with Y a subset of X, and a binary relation $\rightarrow$ on X, Y is a cofinal subset of X if for all x in X there exists y in Y such that x strongly converges to y.

(ii) Given a MARS $(A,\rightarrow_s)_{s\in S}$, and a member a of A, the restriction of A to a (denoted by $a{\downarrow}A$) is the ARS whose objects are the objects of A to which a may strongly converge, together with the restrictions of $(\rightarrow_s)_{s\in S}$ to this subset.

(iii) Given two MARSs $(G,\rightarrow_s)_{s\in S}$ and $(T,\rightarrow_r)_{r\in R}$, a function U: G $\rightarrow$ T is a functor from $(G,\rightarrow_s)_{s\in S}$ to $(T,\rightarrow_r)_{r\in R}$ if for all g, g' in G whenever g converges strongly to g' then U(g) converges strongly to U(g'). □

The old definition of adequacy applies almost verbatim to the present context.

8.8. DEFINITION. Given two MARSs $(G,\rightarrow_s)_{s\in S}$ and $(T,\rightarrow_r)_{r\in R}$, a functor U: $(G,\rightarrow_s)_{s\in S} \rightarrow (T,\rightarrow_r)_{r\in R}$ is an *adequate mapping* of $(G,\rightarrow_s)_{s\in S}$ to $(T,\rightarrow_r)_{r\in R}$ if:

(i) U is surjective.

(ii) For any g, U(g) is a normal form if and only if g is a normal form.

(iii) For each g in G, U maps $g{\downarrow}G$ to a cofinal subset of $U(g){\downarrow}T$, or in plain words, whenever U(g) converges strongly to t in $(T,\rightarrow)$, then there exists a g' in G such that g converges strongly to g' in G and t converges strongly to U(g') in T. □

Clearly the following holds:

8.9. PROPOSITION.

(i) *A composition of adequate mappings on MARSs is adequate.*

(ii) *The identity on an MARS is adequate.*

(iii) *Metric abstract reduction systems and adequate mappings form a category.* □

9. Adequacy of finite graph rewriting for rational term rewriting

Let us consider what sort of graph reduction a real machine can perform. The graphs must be finite, and the reduction sequences must be finitely long. The graphs can, however, be cyclic. As we have seen, such graphs unravel to infinite terms, and their reduction sequences unravel to transfinite term reduction sequences. We wish to establish an adequacy theorem relating finitary cyclic graph rewriting to some form of infinitary term rewriting.

Unravelling finite cyclic graphs gives but a subset of the possible infinite terms. So in order to keep unraveling surjective we have to restrict to what we shall call rational term rewriting. The concept of a rational term is well-known:

9.1. DEFINITION. A *rational term* is a term containing only finitely many non-isomorphic subterms. □

The following equivalent characterisation is also well-known.

9.2. THEOREM. *A term is rational iff it is the unravelling of a finite graph.*

PROOF. Let t be a rational term. Define a graph whose nodes correspond with isomorphism classes of subterms of t. Given a node n, let t' be a member of the isomorphism class corresponding to n. Attach to n the function symbol at the root of t. The successors of n are the nodes corresponding to the isomorphism classes of the successors of the root of t'. The root of the graph is the node corresponding to the isomorphism class of t itself. It is obvious that the resulting graph unravels to t. □

9.3. DEFINITION. (i) A *rational set* of nodes of a rational term is a set of nodes such that, if each of the nodes in the set is marked, the resulting term is still rational, taking the marks into account when testing isomorphism.

(ii) A *rational set of redexes* of a rational term is a set of redexes whose roots are a rational set of nodes. □

9.4. THEOREM. *A set of nodes of a rational term* t *is rational iff there is a graph* g *unravelling to* t, *and a set of nodes of* g *which map by the unravelling to the given set of nodes of* t.

PROOF. Similar to theorem 9.2. □

9.5. DEFINITION. The *rational term reduction sequences* are defined by the following axioms:

(i) A strongly convergent complete development, of length ω, of a rational set of redexes, is rational.

(ii) A concatenation of finitely many rational reduction sequences is rational.

(iii) A subsequence of a rational reduction sequence is rational.

(iv) There are no other rational reduction sequences. □

9.6. THEOREM. *Finitary graph rewriting is adequate for rational infinitary term rewriting.*

PROOF. Surjectivity of unravelling is immediate from Theorem 9.2. Preservation of normal forms is trivial. For cofinality our proof is based on that of theorem 6.3. Significant modifications are required, which incidentally are applicable to the earlier theorem, giving a more refined account of the correspondence between the two systems.

Firstly, in a cyclic graph, a single redex can correspond to infinitely many redexes in the unravelling; furthermore, for certain redexes, it may not be possible to reduce them all in a strongly convergent manner. However, a strongly convergent development of the unravelling of a graph redex fails to exist only when the redex is circular. But as we noted in the proof of theorem 4.5, a circular redex reduces to itself by graph reduction, the effect of which can be obtained in term reduction by the empty sequence. Therefore, whenever the construction calls on us to reduce a term redex which reduces to itself, we omit that reduction step.

Secondly, the construction of the graph reduction sequence is too profligate with reductions. When cyclic graphs are allowed, the Church-Rosser property in general fails, even for complete developments in finite orthogonal graph rewriting. We exhibit an example of this after the end of the proof. Instead, we shall choose for $g_i \rightarrow^{\infty} g_{i+1}$ the complete development of all redexes r of

g_i for which U(r) contains a residual of $t_i \to t_{i+1}$ by $t_i \twoheadrightarrow^\infty t''_i \twoheadrightarrow^\infty t'_i$. To demonstrate that the result of this development is independent of the order in which the redexes are reduced, we will show that in fact it consists of a single redex. This is done by the next two lemmas.

9.7. LEMMA. *Let* s: $t \twoheadrightarrow^\infty U(g)$ *be a strongly convergent term reduction sequence. Say that* s *is* graph-compatible *if for every node* n *of* t, *there exists a node* n' *of* g, *such that* n/s ⊆ U(n'). *If* s *is graph-compatible, then for every redex* r *of* t, *there exists a redex* r'' *of* g *such that* r/s ⊆ U(r').

PROOF. Let n be the root of r. From graph-compatibility, it follows that there is a node n' of g such that n/s ⊆ U(n'). The roots of r/s are a subset of n/s. Let n_i be the i'th immediate successor of n, and suppose it is pattern-matched by r. Then the i'th immediate successors of the roots of r/s are a subset of n_i/s, which are a subset of U(n'_i), where n'_i is the i'th immediate succesor of n' in g. Proceeding in this way, we find that if r/s is nonempty, then there is a redex r' of g rooted at n', such that r/s ⊆ U(r'). □

9.8. LEMMA. *Let* s: $t \twoheadrightarrow^\infty U(g)$ *be a graph-compatible strongly convergent term reduction sequence. Let* t *reduce to* t' *by reduction of a redex* r, *and let* r' *be the redex of* g *such that* r/s ⊆ U(r'). *Let* g *reduce to* g' *by* r'. *Then there is a graph-compatible strongly convergent reduction of* t' *to* U(g').

PROOF. By the parallel moves lemma for strongly convergent transfinite reduction [Ken90a], strongly convergent sequences s/r and r/s exist and converge to the same term t''. r/s consists of the complete development of all residuals of r by s (which is necessarily strongly convergent, as these residuals are all disjoint). By hypothesis, these residuals are all members of U(r') for some redex r' of g. If U(r') has a strongly convergent complete development, then U(g) reduces to U(g') by complete development of U(r') = (r/s)·(U(r')/(r/s)), and t' reduces to U(g') by complete development of U(r')/(r/s). Otherwise, every member of U(r') reduces to itself, and U(g) = t'' = U(g'). In either case, t' may be reduced to U(g').

Now let n be a node of t'. Either there is a node n' of t such that n∈n'/r, or n is created by r. In the first case, n'/s ⊆ U(n'') for some node n'' of g. But then U(n''/r') = U(n'')/U(r') = U(n'')/(r/s)/s' = (n'/s)/(r/s)/s'' = n'/(s·(r/s))/s'' = n'/(r·(s/r))/s'' ⊇ n/(s/r)/s''. In the second case, let u be the address of n relative to the root of r. Then each member of n/(s/r)/s'' is at address u relative to the root of some member of U(r'), and conversely; briefly, n/(s/r) = U(root(r'))/u = U(root(r')/u). So in both cases graph-compatibility is satisfied. □

Continuing the proof of theorem 9.6: it remains to show that the possibly infinite reduction sequences appearing in the proof are all rational. But this is routine. □

Here is the promised counterexample.

Rules: K(x,y) → x.

Graph: x:K(y:K(x,A),B).

There are two redexes in the graph, at x and y. Reducing x and then y yields z:K(z,A); reducing y and then x yields z:K(z,B). We leave it as an exercise for the reader to consider the unravelling of the given graph to the infinite term K(K(K(K(...,A),B),A),B), and to apply the construction of the above theorem to term reduction sequences starting from this term.

10. References

[Ari92] Z.M. ARIOLA and ARVIND, *Graph Rewriting Systems*, to appear as report, MIT (1992).

[Arn80] A. ARNOLD and M. NIVAT, *The metric space of infinite trees*. Algebraic and topological properties, Fundamenta Informatica, **4** (1980) 445-76.

[Bar87] H.P. BARENDREGT, M.C.J.D. VAN EEKELEN, J.R.W. GLAUERT, J.R KENNAWAY, M.J. PLASMEIJER and M.R. SLEEP, Term graph rewriting, Proc. PARLE'87 Conference II, LNCS **259**, 141-158, (Springer-Verlag 1987).

[Der89a] N. DERSHOWITZ and S. KAPLAN, *Rewrite, rewrite, rewrite, rewrite, rewrite*, Proc. ACM Conference on Principles of Programming Languages, Austin, Texas, pp. 250–259, (1989).

[Der89b] N. DERSHOWITZ, S. KAPLAN and D.A. PLAISTED, *Infinite Normal Forms* (plus corrigendum), Proc. ICALP 89, LNCS **372**, pp. 249–262, (Springer-Verlag 1989).

[Der90a] N. DERSHOWITZ, S. KAPLAN and D.A. PLAISTED, *Rewrite, rewrite, rewrite, rewrite, rewrite*, Th. Comp. Sci., **83**, 71–96, (1991).

[Der90b] N. DERSHOWITZ and J.P. JOUANNAUD, Rewrite systems, in *Handbook of Theoretical Computer Science, volume B*, ed. J. van Leeuwen, (Elsevier, Amsterdam, 1990).

[vEB90] P. VAN EMDE BOAS, Machine models and simulations, in Algorithms and Complexity, in *Handbook of Theoretical Computer Science, volume A*, ed. J. van Leeuwen, (Elsevier, Amsterdam, 1990).

[Far89] W.M. FARMER and R.J. WATRO, Redex capturing in term graph rewriting, in *Computing with the Curry Chip*, Report M89-59, The MITRE Corporation, Massachusetts, (1989). A concise version has appeared in Proc. Conference on Rewriting Techniques and Applications, ed. R.V. Book, LNCS **488**, (Springer-Verlag 1991).

[Ken90a] KENNAWAY, J.R., KLOP, J.W., SLEEP, M.R., and DE VRIES, F.J., *Transfinite reductions in orthogonal term rewriting systems*, Report CS-R9041, CWI, (Amsterdam, 1990).

[Ken90b] KENNAWAY, J.R., KLOP, J.W., SLEEP, M.R., and DE VRIES, F.J., *An infinitary Church-Rosser property for non-collapsing orthogonal term rewriting systems*, Report CS-R9043, CWI, (Amsterdam, 1990).

[Ken91a] KENNAWAY, J.R., KLOP, J.W., SLEEP, M.R., and DE VRIES, F.J., *Transfinite reductions in orthogonal term rewriting systems* (extended abstract), in Proc. Conference on Rewriting Techniques and Applications, ed. R.V. Book, LNCS **488**, (Springer-Verlag 1991).

[Ken91b] KENNAWAY, J.R., KLOP, J.W., SLEEP, M.R., and DE VRIES, F.J., *Event Structures and Orthogonal Term Graph Rewriting*, to appear as report, Centrum voor Wiskunde en Informatica. A preliminary version has appeared in the Preliminary Proceedings of the SemaGraph '91 Symposium, eds. M.J. Plasmeijer and M.R. Sleep, (Nijmegen 1991).

[Ken91c] KENNAWAY, J.R., *On transfinite abstract reduction systems*, to appear as report, CWI, (Amsterdam, 1992).

[Klo91] KLOP, J.W., Term rewriting systems, to appear in *Handbook of Logic in Computer Science*, vol.1, eds. S. Abramsky, D. Gabbay, T. Maibaum (Oxford University Press, 1991). Also published as Report CS-R9073, Centrum voor Wiskunde en Informatica, (Amsterdam, 1990).

[Mit91] MITCHELL J.C., *On abstraction and the expressive power of programming languages*, to appear in Proc. Int. Conf. on Theoretical aspects of Computer Software, (Sendai, Japan, 1991).

[O'Do85] O'DONNELL, M.J., *Equational Logic as a Programming Language*, (MIT Press, 1985).

[Pey85] S.L. PEYTON JONES, *The implementation of functional programming languages*, (Prentice-Hall, 1987).

[Sta80] J. STAPLES, *Computation on graph-like expressions*, Th. Comp. Sci., v.10, 171–185, (1980).