Relating Graph and Term Rewriting via Böhm Models

Zena Ariola

CIS-TR-96-09
May 1996

Also in *Applicable Algebra in Engineering, Communication and Computing,* Vol. 7, No. 5, 1996.

Department of Computer and Information Science
University of Oregon

Relating Graph and Term Rewriting via Böhm Models

Zena M. Ariola

Computer & Information Science Department

University of Oregon, Eugene, OR 97403-1202

e-mail: ariola@cs.uoregon.edu

Dealing properly with sharing is important for expressing some of the common compiler optimizations as source-to-source transformations, such as common subexpressions elimination, lifting of free expressions and removal of invariants from a loop. Term graph rewriting is a computational model to accommodate these concerns. In this paper we are interested in defining a term model for term graph rewriting systems, which allows us to prove total correctness of those optimizations. We introduce the notion of Böhm tree, and show that for orthogonal term graph rewriting systems, Böhm tree equivalence defines a congruence. Total correctness then follows in a straightforward way from showing that if a program M contains less sharing than a program N, then both M and N have the same Böhm tree.

Using Böhm trees we also show that orthogonal term graph rewriting systems are a correct implementation of orthogonal term rewriting systems. This boils down to showing that the behavior of a term graph can be deduced from its finite approximations, that is, graph rewriting is a continuous operation. Our approach differs from that of other researchers which is based on infinite rewriting.

1. Introduction

Dealing properly with sharing is important in a framework for reasoning about the implementation of functional languages and the correctness of certain compiler optimizations, such as common subexpressions elimination, lifting of free expressions and removal of invariants from a loop. All these optimizations can be characterized as merely increasing the sharing of sub-computations in a program. If we want to express these optimizations as source-to-source transformations, we need a calculus which can distinguish between, for example, the following two programs:

$$M \equiv (1+1)*(1+1) \qquad N \equiv \{x = 1+1,\ \text{in}\ x*x\}$$

(A note on syntax: N consists of a collection of unordered bindings, and a main expression written following the keyword in.) Graph rewriting is a computational model to accommodate these concerns. Moreover, due to its implicit parallelism it is also suitable as an intermediate language for compilation on parallel machines. In fact, we have successfully described the operational semantics and the compilation of the implicitly parallel language Id [28] using two different graph rewriting systems, called Kid (Kernel Id) and P-TAC (Parallel

Three Address Code) [2, 3, 4], where P-TAC describes sharing in a first-order system, while Kid includes λ-abstraction.

Term graph rewriting has been described in the literature in terms of either category theory notions [12, 13, 14, 15, 22, 27, 30] or more implementation oriented concepts [9, 11, 29, 32]. The first describes graph rewriting steps as single or double push-outs. The second uses notions like pointers, redirection, indirections. Both approaches, though of indisputable merit, fall short in providing a clear, mathematically manageable framework for term graph rewriting. We provide, instead, an equational treatment of term graph rewriting [5, 6, 8], based on the observation that a natural way of linearly representing a graph is by associating a unique name to each node, and then writing down the interconnections through a set of recursion equations. The advantage of this approach, as discussed in [8], is that one can use the intuitions of Equational Logic in manipulating these equations. Our treatment of term graph rewriting is very general in that cyclic graphs are admitted; most of the literature on graph rewriting is still concerned with directed acyclic graphs only [31].

This paper develops a semantics for orthogonal (cyclic) term graph rewriting, which can be used to show total correctness of the above mentioned compiler optimizations, and to prove that term graph rewriting is a correct implementation of term rewriting. The correctness of graph rewriting with respect to term rewriting is also explored in [11], where the relation is based on the notion of normal form, which leads to some undesirable conclusions when cyclic term graphs are admitted, namely that graph rewriting is unsound. We believe that the notion of normal form is inadequate for such a comparison. This point was already stressed by Wadsworth in his analysis of the relation between the syntactic and the semantic aspects of the λ-calculus [34, 35]. Other researchers [17, 23, 24] have based the relation on rewriting of infinite terms. Instead, our approach is based on showing that the behavior of a graph can be deduced from its finite approximations. In other words, we show that graph rewriting is a *continuous* operation.

The paper is organized as follows: In Section 2, we introduce the reader to term graph rewriting in terms of systems of recursion equations. In Section 3, we introduce a function $\mathcal{P}rint$ which given a term graph g returns the *stable* information associated with g. The information gathered by reducing g is then collected in a set called $\mathcal{P}rint^*(g)$. $\mathcal{P}rint^*(g)$ represents the answer or Böhm tree [10] computed by g. We take the answer as our criterion for equating terms, and we show that this equality is a congruence for a subclass of term graph rewriting systems, namely those without overlapping and non-left-linear rules (*i.e.*, orthogonal term graph rewriting systems). Finally, using the notion of Böhm tree, in Section 4, we show that orthogonal term graph rewriting systems are a correct implementation of orthogonal term rewriting systems (same conditions apply: non-overlapping and left-linear rules only). The notion of Böhm tree allows us to consider cyclic graphs, differently from [11]. Using term graph rewriting we also show the classical fact of completeness of inside-out reduction for orthogonal term rewriting systems (TRSs). This result is needed to provide a term model for TRSs. We conclude the paper with our thoughts on future work.

2. Graphs as systems of recursion equations

Given the TRS rule $\mathsf{F}(x) \longrightarrow \mathsf{G}(x,x)$, the term $\mathsf{F}(+(2,3))$ can be rewritten to $\mathsf{G}(+(2,3),+(2,3))$. That is, the term $+(2,3)$ is substituted for each occurrence of the variable x on the right-hand side of the above rule,

and is thus, *duplicated.* A graph rewriting system avoids this duplication of work by substituting a *pointer* to $+(2,3)$ for each reference to variable x, as depicted below:

We represent the term $\mathsf{F}(+(2,3))$ as:

$$\{x_1 = \mathsf{F}(x_2), x_2 = +(x_3, x_4), x_3 = 2, x_4 = 3, \text{ in } x_1\} \ .$$

In applying the above rule, the name x_2, and not the expression $+(2,3)$, will be substituted for each occurrence of x, leading to the term:

$$\{x_1 = \mathsf{G}(x_2, x_2), x_2 = +(x_3, x_4), x_3 = 2, x_4 = 3, \text{ in } x_1\} \ .$$

Thus, term graphs are simply a set of recursion equations. Similar notations appear in the literature [16, 18]. *E.g.,* $\{x : \mathsf{F}(x,y),\ y : \mathsf{G}(x)\}$ in the language DACTL [18]. However, we insist on an equational notation, not just for the sake of style, but because we want to express term graph operations in terms of equational transformations [8].

DEFINITION 2.1. (TERM GRAPH) *Let Σ be a first-order signature. A term graph defined over Σ is defined inductively as follows:*

(i) *a variable x is a term graph;*

(ii) $\mathsf{F}^k(y_1, \cdots, y_k)$ *is a term graph if $y_1, \cdots, y_k$ are variables and $\mathsf{F}^k \in \Sigma$;*

(iii) $\{x_1 = e_1, \cdots, x_n = e_n, \text{ in } x\}$ *is a term graph if*

(iii.1) *for all $i, 1 \leq i \leq n$, x_i is a variable and e_i is a term graph. The variables x_i are bound, other variables occurring in the system are free;*

(iii.2) *x is a variable;*

(iii.3) *for all $i, j, 1 \leq i < j \leq n$, $x_i \not\equiv x_j$.*

Clause (iii.3) prevents multiple definitions of a variable. Furthermore, we make the assumption that all free and bound variables are distinct from each other. The order of the equations in a term graph does not matter. For technical convenience we assume that if the main term g is of the form $\mathsf{F}^k(y_1, \cdots, y_k)$ or is a variable, say y_1, then g has a name associated to it, that is, $g \equiv \{x = g, \text{in } x\}$, with x distinct from $y_i, 1 \leq i \leq k$.

DEFINITION 2.2. (ROOT) *Given a term graph $g \equiv \{x_1 = e_1, \cdots, x_n = e_n, \text{ in } x\}$, x is said to be the root of g, and is written as* `root`*(g).*

In giving a term graph we will sometimes omit the root, in such a case, the first recursion variable is assumed to be the root. For example, we sometimes write

$$\{x_1 = \mathsf{F}(x_2, x_2), x_2 = 2, \text{ in } x_1\}$$

as

$$\{x_1 = \mathsf{F}(x_2, x_2), x_2 = 2\} \ .$$

In the above term we will say that x_2 is referenced twice and x_1 once (*i.e.*, as the root). Notation: we denote by $g \mid x$, with x a bound variable of g, the system rooted at x. *E.g.*, let g be $\{x_1 = \mathsf{F}(x), x = \mathsf{G}(y),\ \mathsf{in}\ x_1\}$, then $g \mid x$ is $\{x = \mathsf{G}(y)\ \mathsf{in}\ x\}$. We automatically perform the removal of the equation $x_1 = \mathsf{F}(x)$ (garbage collection)†.

Analogous to the notion of α-equivalence in λ-calculus [10], term graphs whose difference may be regarded as merely syntactic noise, are equated. For example, we will consider as equivalent the following terms:

$$\begin{array}{ll}
\{\ x = 8, & \{\ x = 8, \\
\quad z = \{\ y = x, & \quad y = x, \\
\qquad\quad w = +(x, y), & \quad w = +(x, y), \\
\qquad\quad \mathsf{in}\ w\}, & \quad \mathsf{in}\ w\}\ . \\
\quad \mathsf{in}\ z\} &
\end{array}$$

This suggests that all the equations of the form $x = y$, with x and y distinct variables, can be removed after having substituted all the occurrences of x by y, and the nesting of recursion equations can be flattened. Moreover, circular equations of the kind $x = x$ are rewritten to $x = \bullet$, where $\bullet$ is a new constant called 'black hole'. As shown in [6, 8] this guarantees the confluence of term graph rewriting without overlapping rules. A term is said to be in *canonical form* if all the substitutions, the rewriting of circular equations, the flattening, and the removal of garbage have been performed. Before introducing the rules to compute the canonical form of a term, we need some notation. E, F range over sets of equations, $E[y/x]$ denotes the set of equations obtained by replacing each free occurrence of x in E by y, and $z[y/x]$ is y if $z \equiv x$, z otherwise.

Each term graph rewriting systems comes equipped with the following rules:

Redundant names:

$\{x = y, E, \mathsf{in}\ z\} \longrightarrow \{E[y/x], \mathsf{in}\ z[y/x]\} \quad x \neq y$

Black hole:

$\{x = x, E, \mathsf{in}\ z\} \longrightarrow \{x = \bullet, E, \mathsf{in}\ z\}$

Flattening:

$\{x = \{E, \mathsf{in}\ z\}, F, \mathsf{in}\ w\} \longrightarrow \{x = z, E, F, \mathsf{in}\ w\}$

Garbage collection:

$\{E, F\ \mathsf{in}\ x\} \longrightarrow \{E, \mathsf{in}\ x\}$ if the bound variables of F do not occur free in E and are distinct from x

It is an easy exercise to check that the above rules are strongly normalizing (*i.e.*, each reduction terminates), and they are confluent (*i.e.*, let R be the above set of rules. If $g \longrightarrow\!\!\!\!\!\rightarrow_R g_1$ and $g \longrightarrow\!\!\!\!\!\rightarrow_R g_2$ then $\exists g_3, g_1 \longrightarrow\!\!\!\!\!\rightarrow_R g_3$ and $g_2 \longrightarrow\!\!\!\!\!\rightarrow_R g_3$. If g does not contain any redex then g is said to be in normal form.). We thus define the canonical form of a term graph g as follows:

(i) compute the normal form of g, say g_1, with respect to the *Flattening* rule;

† As discussed in [7] disallowing garbage collection makes term graph rewriting suitable to describe a notion of state.

(ii) compute the normal form of g_1, say g_2, with respect to the *Redundant names* and the *Black hole* rules;
(iii) compute the normal form of g_2 with respect to the *Garbage collection* rule.
Two terms g and h are then said to be α-*equivalent* ($\equiv_\alpha$) if their canonical forms are the same up to renaming of bound variables and commutativity of the equations. Herein term graphs are assumed to be in canonical form.

Term graph rewriting rules have also an equational format, differently from [11, 29, 32], in which rules are expressed in terms of multi-rooted graphs. For example, the TRS rule

$$\tau : \mathsf{F}(\mathsf{G}(y)) \longrightarrow 0$$

is expressed as

$$\{x = \mathsf{F}(x_1), x_1 = \mathsf{G}(y)\} \longrightarrow \{x = 0\} \ .$$

Only the equation matching the first equation in the left-hand side of the above rule will be rewritten, thus capturing the fact that in term graph rewriting only the pointers to the root of the redex get redirected.

DEFINITION 2.3. (TERM GRAPH RULE) *Let l and r be term graphs with the same root. Then: $l \longrightarrow r$ is a term graph rule.*

As is customary in TRSs two conditions are imposed on the rules. Namely, the left-hand side l is not a variable, and the free variables occurring in the right-hand side r are a subset of the variables occurring in the left-hand side l.

DEFINITION 2.4. (SUBSTITUTION) *A substitution σ is a function from variables to variables. We extend σ to system of recursion equations as follows:*
(i) $\sigma(\mathsf{F}(x_1, \cdots, x_n)) = \mathsf{F}(\sigma(x_1), \cdots, \sigma(x_n))$;
(ii) $\sigma(\{x_1 = e_1, \ \cdots, x_n = e_n\}) = \{\sigma(x_1) = \sigma(e_1), \ \cdots, \sigma(x_n) = \sigma(e_n)\}$.
We will also write g^σ instead of $\sigma(g)$.

DEFINITION 2.5. (REDEX) *Let $\tau : l \rightarrow r$, x a bound variable occurring in g. Then (τ, x, σ) is a redex if $l^\sigma \subseteq g$ and $\mathtt{root}(l^\sigma) = x$. If x is the root of g then g itself is said to be a redex.*

Thus, detection of a redex boils down to matching parts of a system of recursion equations.
Notation: $g[x \leftarrow e]$ is the term obtained by replacing the term bound to x by e. If x does not occur bound in g then $g[x \leftarrow e]$ is g. We assume that the bound variables of e do not occur either free or bound in g. We will also make use of the notation $g[\mathcal{F} \leftarrow e]$, where $\mathcal{F}$ is a set of variables, indicating that all terms bound to each x in $\mathcal{F}$ are replaced by e.
Consider the following rule τ and term g, respectively:

$$\tau : \{x = \mathsf{G}(y_1, x_1), x_1 = \mathsf{F}(y_2)\} \longrightarrow \{x = y_2\} \qquad g \equiv \{ \ z_1 = \mathsf{F}(y), \ z_2 = \mathsf{G}(z_1, z_1), \ \mathsf{in}\ z_2\} \ .$$

(τ, z_2, σ) is a redex, where σ is $x = z_2, y_1 = z_1, x_1 = z_1, y_2 = y$. Reduction consists of replacing the term bound to z_2 by an instance of the right-hand side of τ, that is, $g \longrightarrow g[z_2 \leftarrow y] \equiv_\alpha y$.

DEFINITION 2.6. (REDUCTION) *Let (τ, x, σ) be a redex occurring in g. Let $\tau : l \rightarrow r$, then $g \longrightarrow g[x \leftarrow (r^\sigma)']$, with $(r^\sigma)'$ denoting the renaming of all bound variables (using fresh variables) of r^σ. $\longrightarrow\!\!\!\!\!\rightarrow$ denotes the transitive reflexive closure of $\longrightarrow$.*

Renaming is necessary to avoid collision with the variables in a system. We will sometimes use the notation $g \underset{x}{\longrightarrow} h$ to indicate the reduction of a redex rooted at x.

DEFINITION 2.7. (TERM GRAPH REWRITING SYSTEM) *A term graph rewriting system is a structure $(A(\Sigma), R)$, where $A(\Sigma)$ is the set of term graphs defined over signature Σ, and R is a set of term graph rules.*

DEFINITION 2.8. (LEFT-LINEAR RULE) *A rule $\tau : l \longrightarrow r$ is said to be left-linear iff for all variables x occurring in l, x is referenced at most once in l.*

For example, the following rules:

$$\tau_1 : \{x = \mathsf{F}(x)\} \longrightarrow \{x = 0\} \text{ and } \tau_2 : \{x = \mathsf{F}(y, y)\} \longrightarrow \{x = 0\}$$

are non-left-linear rules. In particular, rule τ_1 is non-left-linear because x is referenced twice, x is referenced in the root and in the right-hand side of the equation $x = \mathsf{F}(x)$.

DEFINITION 2.9. (COMPATIBLE TERMS) *Let g_1, g_2 be term graphs. g_1 and g_2 are called compatible (written as $g_1 \uparrow g_2$) if there exists a term graph g_3, substitutions σ_1 and σ_2, such that*
(i) $g_1^{\sigma_1} \subseteq g_3$ *and* $g_2^{\sigma_2} \subseteq g_3$;
(ii) $\mathtt{root}(g_1^{\sigma_1}) = \mathtt{root}(g_2^{\sigma_2}) = \mathtt{root}(g_3)$.

DEFINITION 2.10. (OVERLAPPING) *Let $\tau_1 : l_1 \rightarrow r_1, \tau_2 : l_2 \rightarrow r_2$. We say that τ_1 overlaps with τ_2 iff there exists a bound variable x in l_1 such that*

$$(l_1 \mid x) \uparrow l_2$$

If $\tau_1 = \tau_2$, then it must be the case that x is distinct from the root of l_1.

THEOREM 2.11. *A term graph rewriting system without overlapping rules is confluent up to renaming.*

PROOF. See [6, 8]. □

DEFINITION 2.12. (ORTHOGONAL) *A term (graph) rewriting system is said to be orthogonal if all the rules are left-linear and non-overlapping.*

Hereafter, we will only consider orthogonal term (graph) rewriting systems.

3. Böhm Model for Orthogonal term graph rewriting systems

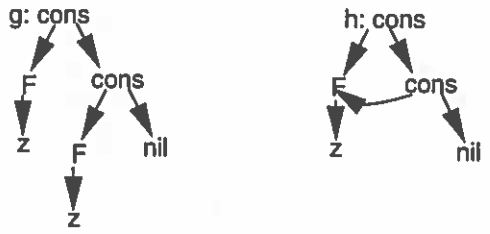
We want to define the printable value of a term graph g as the stable part of g, that is, that part that will never change during reduction. The printable value corresponds to the notions of instant semantics [36] and direct approximation [26, 33] introduced for the λ-calculus. As for the λ-calculus, all redexes must first be substituted by a new constant Ω, which stands for no information. However, this is not enough to guarantee the monotonicity of the information content with respect to reduction. The problem is due to the upward creation of redexes; that is, even though a term M is not a redex, it can become so when some redexes under it are performed. To cope with this phenomenon in the λ-calculus, both Wadsworth [33] and Lévy [26] have introduced the notion of ω-rule, which states $\Omega\ P \longrightarrow \Omega$. Huet and Lévy have applied the same concept to orthogonal TRSs [21], and in [6] we have extended the work to term graph rewriting systems without overlapping rules. According to [6], the stable part of a term graph is computed by replacing its redexes and "potential redexes" by Ω, where a term is a "potential redex" if it can become a redex by either replacing Ω with some other term or by increasing its sharing. For example, with respect to the rules:

$$\begin{aligned} \{x = \mathsf{F}(x_1), x_1 = 0\} &\longrightarrow \{x = 1\} \\ \{x = \mathsf{G}(y)\} &\longrightarrow \{x = 0\} \end{aligned}$$

the stable part of $g \equiv \{x_1 = \mathsf{F}(x_2),\ x_2 = \mathsf{G}(y),\ \text{in } x_1\}$ is computed as follows: the redex rooted at x_2 is replaced by Ω, obtaining $g_1 \equiv \{x_1 = \mathsf{F}(x_2),\ x_2 = \Omega,\ \text{in } x_1\}$. g_1 becomes a redex by replacing Ω with the constant 0, thus the stable part of g is Ω. While, with respect to the above rules, $g_1 \equiv \{x = \mathsf{F}(x), \text{in } x\}$ is considered to be stable. Thus, in [6] sharing is part of our observations. Subsequently, we are not able to equate the following term graphs:

$$\begin{aligned} g &\equiv \{\ x = \mathsf{cons}(x_1, x_2), x_1 = \mathsf{F}(z), x_2 = \mathsf{cons}(x_3, x_4), x_3 = \mathsf{F}(z), x_4 = \mathsf{nil},\ \text{in } x\} \\ h &\equiv \{\ x = \mathsf{cons}(x_1, x_2), x_1 = \mathsf{F}(z), x_2 = \mathsf{cons}(x_1, x_4), x_4 = \mathsf{nil},\ \text{in } x\} \end{aligned}$$

Term graphs g and h may be drawn as follows:



However, if the internal representation of a list is ignored then both g and h represent the same list, *i.e.*, $\mathsf{F}(z) : \mathsf{F}(z) : \mathsf{nil}$. Analogously, consider the following two term graphs g_1 and h_1:

$$g_1 \equiv \{x = \mathsf{cons}(x_1, x), x_1 = 1,\ \text{in } x\} \qquad h_1 \equiv \{x = \mathsf{F}(x_1), x_1 = 1,\ \text{in } x\}$$

and the rule

$$\{x = \mathsf{F}(y)\} \rightarrow \{x = \mathsf{cons}(y, x_1), x_1 = \mathsf{F}(y)\}\ .$$

According to [6], the information associated to h_1 is contained in the information associated to g_1, but not vice versa. Thus, g_1 and h_1 are not equated, even though intuitively it can be said that both g_1 and h_1

represent the infinite list of one's. g_1 has a finite representation of it, in other words, the infinite list of one's is represented through a cycle. (This representation issue is similar to the representation issue for real numbers, where the number with decimal expansion $.999999\cdots$ is the same as 1).

In this paper, we are interested in developing a semantics that equates g to h and g_1 to h_1. Thus we go one step further by eliminating sharing from our observations, that is, we compute the stable part of each *expansion* associated to a term graph. For example, with respect to the rule:

$$\{x = \mathsf{F}(x_1), x_1 = 0\} \longrightarrow \{x = 0\}$$

we will say that term graph $g \equiv \{x = \mathsf{F}(x),\ \text{in } x\}$ does not have any information because none of the expansions, *i.e.*, $\{\Omega, \mathsf{F}(\Omega), \mathsf{F}(\mathsf{F}(\Omega)), \cdots\}$, are stable with respect to the TRS rule $\mathsf{F}(0) \longrightarrow 0$. Analogously, the information associated to the term graph $g_1 \equiv \{x = +(y,y), y = +(x,x),\ \text{in } x\}$ is also Ω.

3.1. Printable Value of a Term

To obtain the printable value of a term, we replace all its redexes by Ω and then compute the stable part of each expansion of the term. The expansions are first-order terms (*i.e.*, TRS terms) defined over a signature containing the constant Ω.

Definition 3.1. (T_Ω) *The set $T_\Omega(\Sigma)$ of TRS terms defined over signature $\Sigma \cup \Omega$ is:*

$$T_\Omega ::= \Omega \mid Variable \mid \mathsf{F}^n(T_\Omega, \cdots, T_\Omega)$$

where F^n is in Σ.

On T_Ω, we define the following prefix ordering:

Definition 3.2. (Prefix Order on T_Ω) *$(T_\Omega, \leq_t)$ is a partial order, where $\leq_t$ is defined as follows:*
(i) $\Omega \leq_t M$, $\forall\, M \in T_\Omega$;
(ii) $x \leq_t x$, $\forall\, x \in Variable$;
(iii) $\mathsf{F}^n(M_1, \cdots, M_n) \leq_t \mathsf{F}^n(N_1, \cdots, N_n)$, *if* $M_i \leq_t N_i$ $1 \leq i \leq n$.
$M <_t N$ *if* $M \leq_t N$ *and* $M \neq N$.

Definition 3.3. *Given a term graph g, we denote by g_Ω the term obtained after substituting all its redexes by Ω.*

In order to determine the expansions of a term graph g, we associate to g a set of TRS rules, called $R_{exp}(g)$.

Example 3.4.
(i) Given the following term graph g:

$$\{\ x = \mathsf{cons}(z, y),\\ y = \mathsf{cons}(z, x),\\ \text{in } x\}\ .$$

$R_{\mathrm{exp}}(g)$ consists of the following two TRS rules:

$$X(z) \longrightarrow \mathsf{cons}(z, Y(z))$$
$$Y(z) \longrightarrow \mathsf{cons}(z, X(z))$$

The first two expansions of g are:

$g^0 \equiv \Omega$;

$g^1 \equiv \mathsf{cons}(z, \Omega)$, which is obtained by first rewriting $X(z)$ (which corresponds to the root of g) to $\mathsf{cons}(z, Y(z))$, and then substituting the redex $Y(z)$ by Ω.

(ii) Consider the term graph g:

$$\{x = \mathsf{F}(x, x) \text{ in } x\} \ .$$

$R_{exp}(g)$ consists of the TRS rule:

$$X \longrightarrow \mathsf{F}(X, X) \ .$$

We rewrite X to $\mathsf{F}(X, X)$, concluding that g^1 is $\mathsf{F}(\Omega, \Omega)$, we then proceed by rewriting simultaneously both the occurrences of X, obtaining $\mathsf{F}(\mathsf{F}(X, X), \mathsf{F}(X, X))$. Thus we will say that g^2 is $\mathsf{F}(\mathsf{F}(\Omega, \Omega), \mathsf{F}(\Omega, \Omega))$.

DEFINITION 3.5. (EXPANSION RULES) *Given a term graph $g \equiv \{x_1 = e_1, \cdots, x_n = e_n, \text{ in } x\}$, let $\{y_1, \cdots, y_k\}$ be the set of free variables of g. Then, $R_{exp}(g)$ consists of the following rules:*

$$\begin{aligned} X1(y_1, \cdots, y_k) &\longrightarrow e_1[X1(y_1, \cdots, y_k)/x_1, \cdots, Xn(y_1, \cdots, y_k)/x_n] \\ &\vdots \\ Xn(y_1, \cdots, y_k) &\longrightarrow e_n[X1(y_1, \cdots, y_k)/x_1, \cdots, Xn(y_1, \cdots, y_k)/x_n] \end{aligned}$$

REMARK 3.6. *$R_{exp}(g)$ is an orthogonal TRS.*

As described in the previous example we rewrite the root of a term graph g according to $R_{exp}(g)$ following the Gross-Knuth or Kleene reduction strategy [25]. Notation: $M \xrightarrow[GK]{}{}^{n} N$ denotes n (TRS) reduction steps according to the Gross-Knuth strategy. In the following definition, N_Ω is computed with respect to the expansion rules.

DEFINITION 3.7. (EXPANSION OF A TERM GRAPH) *Given a term graph g rooted at x, let $\{y_1, \cdots, y_k\}$ be the set of free variables occurring in g. The k^{th} expansion of g, written as g^k, is the term $N_\Omega \in T_\Omega$ such that*
(i) $X(y_1, \cdots, y_k) \xrightarrow[GK]{}{}^{k} N$; *or*
(ii) $X(y_1, \cdots, y_k) \xrightarrow[GK]{}{}^{<k} N$ *and N is in normal form with respect to $R_{exp}(g)$.*

REMARK 3.8. (MONOTONICITY OF EXPANSIONS) *Given a term graph g and its two expansions g^k and $g^{k'}$, if $k \leq k'$ then $g^k \leq_{\mathrm{t}} g^{k'}$.*

In order to determine if an expansion is stable we associate to a term graph rule a set of TRS rules, called ω-rules as in [26, 33]. The aim of a ω-rule is to reduce to Ω potential redexes.

DEFINITION 3.9. (UNWINDING) *Let g be an acyclic term graph rooted at x, with $\{y_1, \cdots, y_k\}$ the set of its free variables.* $\mathtt{Unwind}(g)$ *returns the normal form of $X(y_1, \cdots, y_k)$ with respect to $R_{exp}(g)$.*

For example, if g is $\{x = \mathsf{cons}(x_1, x_2), x_1 = \mathsf{F}(z), x_2 = \mathsf{cons}(x_3, x_4), x_3 = \mathsf{F}(z), x_4 = \mathsf{nil},\ \mathsf{in}\ x\}$, then $\mathtt{Unwind}(g)$ is $\mathsf{cons}(\mathsf{F}(z), \mathsf{cons}(\mathsf{F}(z), \mathsf{nil}))$.

DEFINITION 3.10. (ω-RULE) *Given a TRS rule $l \longrightarrow r$, a rule $r' : l' \longrightarrow \Omega$ is said to be a ω-rule of r iff*
(i) $l' <_t l$;
(ii) *each Ω occurring in l' is not mapped to a variable in l;*
(iii) $l' \not\equiv \Omega$;

Given the TRS rule $\mathsf{F}(y) \longrightarrow 0$, condition (ii) prevents the generation of the ω-rule $\mathsf{F}(\Omega) \longrightarrow \Omega$. Note that Ω is a constant in a ω-rule not a variable.

DEFINITION 3.11. (ω-TRS) *Given a term graph rewriting system $(A(\Sigma), R)$, we define its corresponding ω-TRS to be $(T_\Omega(\Sigma), R_\omega)$, where R_ω is obtained by generating the ω-rules for each rule in $\{\mathtt{Unwind}(l) \longrightarrow \Omega \mid l \longrightarrow r \in R\}$.*

If R is the following set of rules:

$$\begin{aligned} \{x = \mathsf{F}(x_1), x_1 = 0\} &\longrightarrow \{x = 1\} \\ \{x = \mathsf{G}(y)\} &\longrightarrow \{x = 0\} \end{aligned}$$

then R_ω consists of the rule:

$$\mathsf{F}(\Omega) \longrightarrow \Omega\ .$$

We will often refer to a redex in a ω-TRS as a *ω-redex*, and reductions in a ω-TRS will be denoted by $\longrightarrow\!\!\!\!\!\rightarrow_\omega$.

PROPOSITION 3.12. *$\longrightarrow\!\!\!\!\!\rightarrow_\omega$ is strongly normalizing*

PROOF. Since the length of the reduction is bounded by the number of function symbols occurring in a term. □

PROPOSITION 3.13. *$\longrightarrow\!\!\!\!\!\rightarrow_\omega$ is confluent.*

PROOF. Since a ω-TRS is an orthogonal TRS. □

A normal form in a ω-TRS will be referred to as a ω-normal form.

DEFINITION 3.14. (ω-FUNCTION) *Given a ω-TRS term M, $\omega(M)$ denotes the ω-normal form of M.*

PROPOSITION 3.15. (MONOTONICITY OF ω WITH RESPECT TO $\leq_t$) *Given ω-TRS terms M and N, if $M \leq_t N$ then $\omega(M) \leq_t \omega(N)$.*

PROOF. Follows from the fact that each subterm of M that is mapped to a ω-redex in N must either be Ω or a ω-redex. □

We thus collect all the stable information contained in term graphs in a set called ω-$\mathcal{T}rees$.

DEFINITION 3.16. (ω-TREES: SET OF OBSERVATIONS) *The set of all observations is*

$$\omega\text{-}\mathcal{T}rees = \{\omega(g^k_\Omega) \mid \textit{ for all term graphs } g, k \geq 0\}.$$

In order to guarantee that infinite chains in ω-$\mathcal{T}rees$ have a limit, we apply the *ideal completion* method to turn ω-$\mathcal{T}rees$ into a complete partial order [19]. Given a partial order $(A, <)$, a subset D is an *ideal* iff (i) D is non-empty; (ii) for all $a, b \in D$, there exists a $c \in D$, such that $a \leq c$ and $b \leq c$, (iii) for all $c \in D$, if there exists $d \in A$ such that $d \leq c$ then $d \in D$.

DEFINITION 3.17. (BÖHM DOMAIN: ω-$\mathcal{T}rees^\infty$) *The domain of observations, called ω-$\mathcal{T}rees^\infty$, is the ideal completion of ω-$\mathcal{T}rees$, that is,*

$$\omega\text{-}\mathcal{T}rees^\infty = \{S \mid S \subseteq \omega\text{-}\mathcal{T}rees \textit{ and } S \textit{ is an ideal }\} \ .$$

The elements of ω-$\mathcal{T}rees^\infty$ are called Böhm trees [10].

DEFINITION 3.18. (PRINTABLE VALUE OF A TERM GRAPH) *Given a term graph g, the printable value of g is $\mathcal{P}rint(g) = \{a \mid a \in \omega\text{-}\mathcal{T}rees,\ a \leq_t b,\ b \in \{\omega(g^k_\Omega) \mid k \geq 0\}\}$.*

PROPOSITION 3.19. *Given a term graph g, $\mathcal{P}rint(g)$ is in ω-$\mathcal{T}rees^\infty$.*

PROOF. (i) Since $\Omega \in \mathcal{P}rint(g)$, $\mathcal{P}rint(g)$ is non-empty. (ii) From the monotonicity of the expansions (Remark 3.8) and the monotonicity of the ω-function with respect to the $\leq_t$ ordering (Proposition 3.15) follows that $\mathcal{P}rint(g)$ is directed. (iii) It is closed downwards by definition. □

3.2. ANSWER OF A TERM

We define the *answer* or *Böhm tree* of a term graph as the collection of the printable information obtained by reducing that term.

DEFINITION 3.20. (ANSWER OR BÖHM TREE OF A TERM GRAPH) *Given a term graph g, the answer of g is*

$$\mathcal{P}rint^*(g) = \bigcup\{\mathcal{P}rint(h) \mid g \longrightarrow\!\!\!\!\!\rightarrow h\} \ .$$

EXAMPLE 3.21. Consider the term graphs g_1 and h_1 introduced at the beginning of Section 3. Since g_1 does not contain any redex and each expansion is stable, we have:

$$\mathcal{P}rint(g_1) = \mathcal{P}rint^*(g_1) = \{\Omega, \mathsf{cons}(\Omega, \Omega), \mathsf{cons}(1, \Omega), \mathsf{cons}(1, \mathsf{cons}(1, \Omega)), \cdots\} \ .$$

We reduce h_1 and at each step compute the stable information:

$$h_1 \longrightarrow \{\ x = \mathsf{cons}(x_1, x_1'),\ x_1' = \mathsf{F}(x_1), x_1 = 1\} \longrightarrow \{\ x = \mathsf{cons}(x_1, x_1'),\ x_1' = \mathsf{cons}(x_1, x_1''),\ x_1'' = \mathsf{F}(x_1), x_1 = 1\} \quad \cdots$$

$$\mathcal{P}rint: \quad \{\Omega\} \qquad \{\Omega, \mathsf{cons}(\Omega,\Omega), \mathsf{cons}(1,\Omega)\} \qquad \{\Omega, \mathsf{cons}(\Omega,\Omega), \cdots, \mathsf{cons}(1,\mathsf{cons}(1,\Omega))\} \quad \cdots$$

In $\mathcal{P}rint^*(h_1)$ we then collect all the printable information obtaining

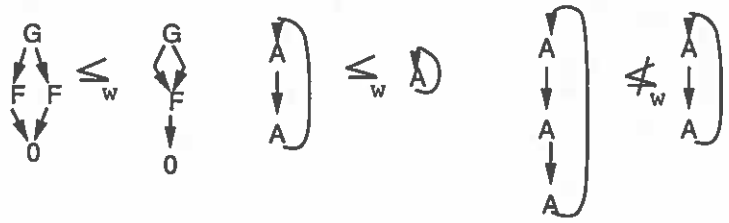
$$\{\Omega, \mathsf{cons}(\Omega,\Omega), \mathsf{cons}(1,\Omega), \mathsf{cons}(1,\mathsf{cons}(1,\Omega)), \mathsf{cons}(1,\mathsf{cons}(1,\mathsf{cons}(1,\Omega))), \cdots\}\ .$$

We can thus equate g_1 to h_1. Instead, following [6] we would have concluded that the answer of g_1 is

$$\{\Omega, \mathsf{cons}(\Omega,\Omega), \mathsf{cons}(1,\Omega), \cdots, \{x = \mathsf{cons}(1,x),\ \mathsf{in}\ x\}\}\ .$$

While the answer of h_1 is the same as above. Each information in the answer of h_1 is contained in the answer of g_1, however, g_1 has more information, namely, the representation in terms of a cycle is part of the observations.

In order to guarantee that the answer is well defined, that is, $\mathcal{P}rint^*(g)$ is in ω-$\mathcal{T}rees^\infty$, we need to show that the printable value is monotonic with respect to reduction. To that end, we extend the signature Σ of term graphs with the constant Ω, and we introduce an ordering, written as $\leq_\omega$, on term graphs, which captures both the sharing and the prefix ordering. Intuitively, $g \leq_\omega h$ if h can be obtained from g by replacing Ω with any other term or by increasing the sharing in g. The ω-ordering captures what in the literature is known as rooted homomorphism [8, 11, 32]. For example, we have



We will then show that if $g \leq_\omega h$ then $\mathcal{P}rint(g) \subseteq \mathcal{P}rint(h)$.
In the following, $\mathsf{Fv}(g)$ and $\mathsf{Bv}(g)$ denote the free and bound variables of g, respectively.

DEFINITION 3.22. (ω-ORDERING: $\leq_\omega$) *Given term graphs g and h, rooted at z_1 and z_2, respectively. $g \leq_\omega h$ iff $\exists$ a function σ from variables to variables such that:*
(i) $\forall x \in \mathsf{Fv}(g)$, $\sigma(x) = x$;
(ii) $\forall x \in \mathsf{Bv}(g)$, *if* $x = \mathsf{F}^k(y_1, \cdots, y_k)$ *(* $\mathsf{F}^k \neq \Omega$*) in g then $\sigma(x) \in \mathsf{Bv}(h)$ and $\sigma(x) = \mathsf{F}^k(\sigma(y_1), \cdots, \sigma(y_k))$ in h;*
(iii) $\sigma(z_1) = z_2$.

According to condition (ii) if x is bound to Ω then $\sigma(x)$ is arbitrary. Herein we assume $\bullet =_\omega \Omega$.

PROPOSITION 3.23. *The ω-ordering is a partial order.*

PROOF. See [8]. □

PROPOSITION 3.24. *Let z be a bound variable of g. If $g \mid z$ is not a redex and $g \mid z \leq_\omega h$, with h a redex, then $\mathcal{P}rint(g) = \mathcal{P}rint(g[z \leftarrow \Omega])$.*

PROOF. Since $g \mid z$ is a potential redex it means that all subterms in g^i, for any i, corresponding to z are either Ω or ω-redexes. Thus, $\omega(g^i_\Omega) = \omega((g[z \leftarrow \Omega])^i_\Omega)$, that is, $\mathcal{P}rint(g) = \mathcal{P}rint(g[z \leftarrow \Omega])$. □

PROPOSITION 3.25. (MONOTONICITY OF $\mathcal{P}rint$ WITH RESPECT TO $\leq_\omega$) *Given term graphs g and h, if $g \leq_\omega h$ then $\mathcal{P}rint(g) \subseteq \mathcal{P}rint(h)$.*

PROOF. Let $\mathcal{Z} = \{z \mid z \in \text{Bv}(g),\ z$ is not a redex in g and $\sigma(z)$ is a redex in $h\}$, with σ the substitution induced by $g \leq_\omega h$. We have

$$g_\Omega[\mathcal{Z} \leftarrow \Omega] \leq_\omega h_\Omega \ .$$

Since $(g_\Omega[\mathcal{Z} \leftarrow \Omega])^i \leq_t h^i_\Omega, i \geq 0$, from the monotonicity of the ω-function with respect to the $\leq_t$ ordering (Proposition 3.15),

$$\mathcal{P}rint(g_\Omega[\mathcal{Z} \leftarrow \Omega]) \subseteq \mathcal{P}rint(h) \ .$$

The result then follows from from Proposition 3.24. □

PROPOSITION 3.26. (MONOTONICITY OF $\mathcal{P}rint$ WITH RESPECT TO $\longrightarrow\!\!\!\!\!\rightarrow$) *Given term graphs g and h, if $g \longrightarrow\!\!\!\!\!\rightarrow h$ then $\mathcal{P}rint(g) \subseteq \mathcal{P}rint(h)$.*

PROOF. The proof is by induction on the number n of reduction steps. Suppose $g \xrightarrow[z]{} h$, then h is $g[z \leftarrow k]$, for a term graph k. Since $\mathcal{P}rint(g) = \mathcal{P}rint(g[z \leftarrow \Omega])$, and $g[z \leftarrow \Omega] \leq_\omega g[z \leftarrow k]$, from the monotonicity of the $\mathcal{P}rint$ function with respect to $\leq_\omega$ (Proposition 3.25) we have that $\mathcal{P}rint(g) \subseteq \mathcal{P}rint(h)$. The result then follows trivially from the induction hypothesis. □

PROPOSITION 3.27. *Given a term graph g, $\mathcal{P}rint^*(g)$ is in ω-$\mathcal{T}rees^\infty$.*

PROOF.
(i) $\mathcal{P}rint^*(g)$ is non-empty since it contains at least Ω.
(ii) We show that $\mathcal{P}rint^*(g)$ is a directed set, that is,

$$\forall\, a, b \in \mathcal{P}rint^*(g),\ \exists\, c, \text{ such that } a \leq_t c \text{ and } b \leq_t c \ .$$

By definition of $\mathcal{P}rint^*$, $\exists\ g_1, g_2$ such that, $g \longrightarrow\!\!\!\!\!\rightarrow g_1$, with $a \in \mathcal{P}rint(g_1)$ and $g \longrightarrow\!\!\!\!\!\rightarrow g_2$, with $b \in \mathcal{P}rint(g_2)$. By the confluence of $\longrightarrow\!\!\!\!\!\rightarrow$,

$$\exists\ g_3,\ g_1 \longrightarrow\!\!\!\!\!\rightarrow g_3 \text{ and } g_2 \longrightarrow\!\!\!\!\!\rightarrow g_3 \ .$$

By monotonicity of $\mathcal{P}rint$ with respect to reduction (Proposition 3.26) we have that $a, b \in \mathcal{P}rint(g_3)$. The result then follows from $\mathcal{P}rint(g_3)$ being directed.

(iii) $\mathcal{P}rint^*(g)$ is closed downwards by definition.
□

Using the notion of answer or Böhm tree, we can define an ordering on term graphs as follows.

DEFINITION 3.28. ($\sqsubseteq_{\mathsf{BT}}$: ORDERING ON OBSERVABLE INFORMATION) *Given term graphs g and h, $h \sqsubseteq_{\mathsf{BT}}$ h iff $\mathcal{P}rint^*(g) \subseteq \mathcal{P}rint^*(h)$. If $g \sqsubseteq_{\mathsf{BT}} h$ and $h \sqsubseteq_{\mathsf{BT}} g$ then $g \equiv_{\mathsf{BT}} h$.*

If we want $\mathcal{P}rint^*$ to be our interpretation function, $\mathcal{P}rint^*$ will have to satisfy certain properties; that is, the meaning will have to be preserved by reduction (*soundness*), and it will have to be compositional (*congruence*).

THEOREM 3.29. (SOUNDNESS) *Given term graphs g and h, if $g \longrightarrow\!\!\!\!\!\rightarrow h$ then $g \equiv_{\mathsf{BT}} h$.*

PROOF. Trivially, $h \sqsubseteq_{\mathsf{BT}} g$. The other direction follows from the confluence of the system and from the monotonicity of $\mathcal{P}rint$ with respect to reduction (Proposition 3.26). □

The proof of congruence requires some more machinery. Since $\mathcal{P}rint^*$ equality deals with expansions of terms, *i.e.*, trees, it is natural that if a rule in a term graph rewriting system can distinguish between different sharing of subterms, then $\mathcal{P}rint^*$ would not be a congruence. A way of assuring that $\equiv_{\mathsf{BT}}$ is a congruence is to show that the behavior of $C[g]$ can be inferred from the observations about g, where $C[\Box]$ is a context, as defined below.

DEFINITION 3.30. *A context $C[\Box]$ is a term graph defined over the signature Σ extended with a special symbol $\Box$.*

For example, $\{x = \mathsf{F}(x_1), x_1 = \Box\}$ is a context; $C[0]$ is the term graph obtained after replacing $\Box$ by 0, that is, $C[0]$ is the term $\{x = \mathsf{F}(x_1), x_1 = 0\}$.

We want to show the following:

$$\forall C[\Box],\ C[g] \equiv_{\mathsf{BT}} \bigsqcup \{C[\mathtt{Bl}(a)] \mid a \in \mathcal{P}rint^*(g)\} \qquad (1)$$

where $\mathcal{P}rint^*(\bigsqcup S) = \bigcup\{\mathcal{P}rint^*(s) \mid s \in S\}$, and $\mathtt{Bl}(a)$ represents the term graph obtained by assigning a unique name to each subterm of a.

DEFINITION 3.31. *Given a ω-TRS term M, its corresponding term graph, written as $\mathtt{Bl}(M)$, is defined as follows:*
(i) $\mathtt{Bl}(\Omega) = \Omega$;
(ii) $\mathtt{Bl}(x) = x$;
(iii) $\mathtt{Bl}(\mathsf{F}^n(M_1, \cdots, M_n)) = \{x_1 = \mathtt{Bl}(M_1), \cdots, x_n = \mathtt{Bl}(M_n), x = \mathsf{F}^n(x_1, \cdots, x_n) \text{ in } x\}$.

For example, if a is the term $\mathsf{cons}(1, \mathsf{cons}(1, \Omega))$, then $\mathtt{Bl}(a)$ is the term graph $\{x = \mathsf{cons}(x_1, x_2), x_1 = 1, x_2 = \mathsf{cons}(x_3, x_4), x_3 = 1, x_4 = \Omega\}$.

Note that (1) can be considered as a syntactic version of the continuity of the context operation. In other words, to get an *approximation* of the answer of $C[g]$ an *approximation* of the answer of g suffices. In the following, we show through examples that in that respect *non-left-linear* rules are discontinuous†.

EXAMPLE 3.32.

(i) Consider the following rule:

$$\{x = \mathsf{G}(x_1), x_1 = \mathsf{A}(x_1)\} \longrightarrow \{x = 1\} \ .$$

Let $g \equiv \{x_1 = \mathsf{A}(x_1),\ \mathsf{in}\ x_1\}$ and $C[\Box] \equiv \{x_1 = \mathsf{G}(x_2), x_2 = \Box,\ \mathsf{in}\ x_1\}$. Since $C[g] \longrightarrow 1$, we would conclude that the answer of $C[g]$ is $\{\Omega, 1\}$. However, 1 can not be obtained by plugging in the context any observation about g. g is not a redex and its expansions are stable, thus the answer of g is $\{\Omega, \mathsf{A}(\Omega), \mathsf{A}(\mathsf{A}(\Omega)), \cdots\}$. Let us insert in the context $C[\Box]$ any of the above observations: the answer of $C[\Omega]$ is Ω because $C[\Omega]$ is not a redex but can become a redex. The same applies if we plug any other observation of g in $C[\Box]$. That is, the answer of $C[\mathtt{B1}(a)]$ is Ω, for any observation a of g.

(ii) Consider the following rule:

$$\{x = \mathsf{G}(x_1), x_1 = \mathsf{A}(y, y)\} \longrightarrow \{x = 1\} \ .$$

Let $g \equiv \{x = \mathsf{A}(x_1, x_1), x_1 = \mathsf{B}(0),\ \mathsf{in}\ x\}$ and $C[\Box] \equiv \{x_1 = \mathsf{G}(x_2), x_2 = \Box,\ \mathsf{in}\ x_1\}$. The answer of g is $\{\Omega, \mathsf{A}(\Omega, \Omega), \cdots, \mathsf{A}(\mathsf{B}(0), \mathsf{B}(0))\}$. Let us plug the observation $a \equiv \mathsf{A}(\Omega, \Omega)$ in the context, obtaining the term graph $C[\mathtt{B1}(a)] \equiv \{x_1 = \mathsf{G}(x_2), x_2 = \mathsf{A}(x_3, x_4), x_3 = \Omega, x_4 = \Omega,\ \mathsf{in}\ x_1\}$. $C[\mathtt{B1}(a)]$ does not contain any redex but is a potential redex, that is, by increasing its sharing it can become a redex. Thus, the answer of $C[\mathtt{B1}(a)]$ is Ω, the same applies for any other observation of g. While, $C[g] \longrightarrow 1$ and thus the answer of $C[g]$ is $\{\Omega, 1\}$.

LEMMA 3.33. (MONOTONICITY OF $\sqsubseteq_{\mathsf{BT}}$ WITH RESPECT TO $\leq_\omega$) *Given term graphs g and h, if $g \leq_\omega h$ then $g \sqsubseteq_{\mathsf{BT}} h$.*

PROOF. By induction on the number of reduction steps we prove that

$$\text{if } g \longrightarrow\!\!\!\!\!\rightarrow g_1 \text{ then } \exists h_1 \text{ such that } h \longrightarrow\!\!\!\!\!\rightarrow h_1 \text{ and } \mathcal{P}rint(g_1) \subseteq \mathcal{P}rint(h_1)$$

Suppose $g \underset{z}{\longrightarrow} g_1$. Let z_1 be $\sigma(z)$, with σ the substitution induced by $g \leq_\omega h$, and let h_1 be such that $h \underset{z_1}{\longrightarrow} h_1$. Let $\mathcal{F} = \{s \mid \sigma(s) = z_1, s \neq z, s \text{ not bound to } \Omega\}$. Then, $g_1[\mathcal{F} \leftarrow \Omega] \leq_\omega h_1$. Due to orthogonality, each $s \in (\mathcal{F} \cap \mathtt{Bv}(g_1))$ denotes either a redex or a potential redex in g_1. Thus, from Proposition 3.24,

$$\mathcal{P}rint(g_1) = \mathcal{P}rint(g_1[\mathcal{F} \leftarrow \Omega]) \ .$$

It follows from the monotonicity of the $\mathcal{P}rint$ function with respect to $\leq_\omega$ (Proposition 3.25) that

$$\mathcal{P}rint(g_1) \subseteq \mathcal{P}rint(h_1) \ .$$

The result then follows from the observation that due to orthogonality any redex occurring in g but not in $\mathcal{F}$ is still a redex in $g_1[\mathcal{F} \leftarrow \Omega]$. Therefore, there exists a corresponding redex in h_1. □

† Private communications with Georges Gonthier and Jean-Jacques Lévy.

THEOREM 3.34. *Given a term graph g, $\bigsqcup\{C[\mathtt{Bl}(a)] \mid a \in \mathcal{P}rint^*(g)\} \sqsubseteq_{\mathsf{BT}} C[g]$.*

PROOF. Let $g \twoheadrightarrow h$, and $a \in \mathcal{P}rint(h)$. We have

$$\begin{array}{lll} \mathtt{Bl}(a) & \leq_\omega & h \\ C[\mathtt{Bl}(a)] & \sqsubseteq_{\mathsf{BT}} & C[h] \quad \text{(by the monotonicity of } \sqsubseteq_{\mathsf{BT}} \text{ with respect to } \leq_\omega \text{ (Lemma 3.33))} \\ & \equiv_{\mathsf{BT}} & C[g] \quad \text{(by soundness of } \equiv_{\mathsf{BT}} \text{ (Theorem 3.29))} \ . \end{array}$$

□

In order to prove the other direction, that is,

$$\forall C[\Box],\ C[g] \sqsubseteq_{\mathsf{BT}} \bigsqcup\{C[\mathtt{Bl}(a)] \mid a \in \mathcal{P}rint^*(g)\}$$

we show:

(i) if $g \twoheadrightarrow h$ then the information associated to each expansion of h can be obtained by reducing an expansion of g;

(ii) not too much information is lost by applying ω-reductions.

LEMMA 3.35. *Given a term graph g, if $g \twoheadrightarrow g_1$ then $\forall k, \exists i, \mathtt{Bl}(g^i) \twoheadrightarrow h$ and $\mathtt{Bl}(g_1^k) \leq_\omega h$.*

PROOF. The proof is by induction on the number of reduction steps of $g \twoheadrightarrow g_1$.
Suppose $g \longrightarrow g_1$ by reducing the redex (τ, z, σ). Let τ be $l \longrightarrow r$. Then, g_1 is $g[z \leftarrow (r^\sigma)']$. By definition of expansions and due to orthogonality we have:

$$\mathtt{Bl}((g[z \leftarrow (r^\sigma)'])^k) \leq_\omega \mathtt{Bl}(g^i)[z_1 \leftarrow (r^{\sigma_1})'] \cdots [z_n \leftarrow (r^{\sigma_n})']$$

where:

(i) If we let σ' be the substitution induced by $\mathtt{Bl}(g^i) \leq_\omega g$, we have

$$\{z_1, \cdots, z_n\} = \{w \mid w \in \mathtt{Bv}(\mathtt{Bl}(g^i)) \text{ such that } \sigma'(w) = z \text{ and } w \text{ is not bound to } \Omega\} \ .$$

(ii) $\forall j, 1 \leq j \leq n$, z_j is a τ-redex occurring in $\mathtt{Bl}(g^i)$. (The expansions are such that the pattern of a redex is fully expanded. For example, if you have the rule:

$$\{x = \mathsf{A}(x_1), x_1 = \mathsf{B}(y)\} \longrightarrow \{x = \mathsf{I}(y)\}$$

and the term

$$\{x_1 = \mathsf{A}(x_2),\ x_2 = \mathsf{B}(x_1),\ \text{in } x_1\}$$

then we consider the expansions: $\mathsf{A}(\mathsf{B}(\Omega)), \mathsf{A}(\mathsf{B}(\mathsf{A}(\mathsf{B}(\Omega)))), \cdots$, but not $\mathsf{A}(\mathsf{B}(\mathsf{A}(\Omega)))$.)

(iii) $\forall j, 1 \leq j \leq n, \sigma_j$ is a substitution from l to $\mathtt{Bl}(g^i)$.

Since $\mathtt{Bl}(g^i)[z_1 \leftarrow (r^{\sigma_1})'] \cdots [z_n \leftarrow (r^{\sigma_n})']$ is the result of a complete development of $\mathtt{Bl}(g^i)$ with respect to $\mathcal{F} = \{z_1, \cdots, z_n\}$ (written as $\mathtt{Bl}(g^i) \underset{cdv(\mathcal{F})}{\longrightarrow} h$), g^i is the term we are looking for.

Suppose $g \twoheadrightarrow g_{n-1} \underset{z}{\longrightarrow} g_n$. We have that $\forall k, \exists i$ such that

$$\mathtt{Bl}(g_{n-1}^i) \underset{cdv(\mathcal{F}_1)}{\longrightarrow} h_n \text{ and } \mathtt{Bl}(g_n^k) \leq_\omega h_n$$

where $\mathcal{F}_1 = \{z_1 \mid \sigma_1(z_1) = z \text{ and } z_1 \text{ a redex}\}$, with σ_1 the substitution induced by $\mathtt{Bl}(g^i_{n-1}) \leq_\omega g_{n-1}$. By induction hypothesis, $\exists j$, such that $\mathtt{Bl}(g^j) \longrightarrow\!\!\!\!\!\rightarrow h_{n-1}$ and $\mathtt{Bl}(g^i_{n-1}) \leq_\omega h_{n-1}$; let σ' be the induced substitution function. Note that $\forall z_1 \in \mathcal{F}_1$, the corresponding node in h_{n-1} is a redex. Let $\mathcal{F}'_1 = \{\sigma(z_1) \mid z_1 \in \mathcal{F}_1\}$. Each $z_1 \in \mathtt{Bv}(\mathtt{Bl}(g^i_{n-1}))$ such that $\sigma'(z_1) \in \mathcal{F}'_1$ either belongs to $\mathcal{F}_1$ or is Ω. Therefore, $h_{n-1} \underset{cdv(\mathcal{F}'_1)}{\longrightarrow} h'_n$ and $h_n \leq_\omega h'_n$. Subsequently, $\mathtt{Bl}(g^k_n) \leq_\omega h'_n$. □

COROLLARY 3.36. *Given a term graph g, $g \equiv_{\mathsf{BT}} \bigsqcup\{\mathtt{Bl}(g^k) \mid k \geq 0\}$.*

PROOF. We show that $g \sqsubseteq_{\mathsf{BT}} \bigsqcup\{\mathtt{Bl}(g^k) \mid k \geq 0\}$. Let $a \in \mathcal{P}rint^*(g)$, that is, $g \longrightarrow\!\!\!\!\!\rightarrow g_1$ and $a \in \mathcal{P}rint(g_1)$. Let $k, a \leq_{\mathsf{t}} \omega(g_1{}^k_\Omega)$. Thus, $a \in \mathcal{P}rint(\mathtt{Bl}(g_1{}^k))$. From the above lemma, $\exists i, \mathtt{Bl}(g^i) \longrightarrow\!\!\!\!\!\rightarrow h$ and $\mathtt{Bl}(g_1{}^k) \leq_\omega h$. From monotonicity of $\mathcal{P}rint$ with respect to $\leq_\omega$ (Proposition 3.25), $a \in \mathcal{P}rint(h)$. Thus, $a \in \mathcal{P}rint^*(\mathtt{Bl}(g^i))$. The other direction follows from the monotonicity of $\sqsubseteq_{\mathsf{BT}}$ with respect to the $\leq_\omega$ ordering (Lemma 3.33). □

EXAMPLE 3.37. Consider the following rule:

$$\{x = \mathsf{F}(x_1), x_1 = 0\} \longrightarrow \{x = 0\}$$

and the term $g \equiv \{x = \mathsf{F}(x),\ \text{in } x\}$. Taking $\mathcal{P}rint(g) = \{\Omega\}$ guarantees that the behavior of a term is determined by its expansions. Suppose we say $\mathcal{P}rint(g) = \{\Omega, \mathsf{F}(\Omega), \mathsf{F}(\mathsf{F}(\Omega)), \cdots\}$, then $g \not\equiv_{\mathsf{BT}} \bigsqcup\{\mathtt{Bl}(g^i) \mid i \geq 0\}$. In fact, $\mathcal{P}rint^*(g) = \{\Omega, \mathsf{F}(\Omega), \mathsf{F}(\mathsf{F}(\Omega)), \cdots\}$, while $\mathcal{P}rint^*(\mathtt{Bl}(g^i)) = \Omega$, for any i.

LEMMA 3.38. *Given ω-TRS terms M, N. If $M \longrightarrow\!\!\!\!\!\rightarrow_\omega N$ then $\mathtt{Bl}(M) \equiv_{\mathsf{BT}} \mathtt{Bl}(N)$.*

PROOF. Since $\mathtt{Bl}(N) \leq_\omega \mathtt{Bl}(M)$, the direction $\mathtt{Bl}(N) \sqsubseteq_{\mathsf{BT}} M$ follows from the monotonicity of $\sqsubseteq_{\mathsf{BT}}$ with respect to $\leq_\omega$.

Let R be a set of term graph rules, and R_1 be the set of term graph rules associated to R_ω (that is, for each ω-rule $l \longrightarrow \Omega$ a name is associated to each subterm of l). $R \cup R_1$ is an orthogonal term graph rewriting system and thus is subcommutative [8, 6]. Let $\mathtt{Bl}(M) \longrightarrow\!\!\!\!\!\rightarrow_{R_1} \mathtt{Bl}(N)$ and $\mathtt{Bl}(M) \longrightarrow\!\!\!\!\!\rightarrow_R g$. We want to show that each information contained in g is obtained by reducting $\mathtt{Bl}(N)$. From the subcommutativity of $R \cup R_1$,

$$\exists h, g \longrightarrow\!\!\!\!\!\rightarrow_{R_1} h \text{ and } \mathtt{Bl}(N) \longrightarrow\!\!\!\!\!\rightarrow_R h\ .$$

Since $\mathcal{P}rint(g) = \mathcal{P}rint(h)$ and $\mathcal{P}rint(h) \subseteq \mathcal{P}rint^*(\mathtt{Bl}(N))$ we have $\mathtt{Bl}(M) \sqsubseteq_{\mathsf{BT}} \mathtt{Bl}(N)$. □

THEOREM 3.39. *Given a term graph g, $C[g] \sqsubseteq_{\mathsf{BT}} \bigsqcup\{C[\mathtt{Bl}(a)] \mid a \in \mathcal{P}rint^*(g)\}$.*

PROOF. Suppose $C[g] \longrightarrow\!\!\!\!\!\rightarrow h$, we want to show that each information about h, say a, can be obtained by plugging in the context some observation about g.

Due to the subcommutativity of a term graph rewriting system [6] we first reorder the reduction $C[g] \longrightarrow\!\!\!\!\!\rightarrow h$, in a way consisting of first reducing all redexes inside g and then reducing the redexes in the context; that is,

$$C[g] \longrightarrow\!\!\!\!\!\rightarrow h \Longrightarrow \exists g',\ C[g] \longrightarrow\!\!\!\!\!\rightarrow C[g'] \text{ and } C[g'] \overrightarrow{\ }_{g'} h\ .$$

where $C[g'] \overset{\not g'}{\longrightarrow} h$ means that $C[g']$ reduces to h without performing any redexes occurring in g'.
Let $\mathcal{F}$ be the set of all redexes occurring in g'. Due to the absence of overlapping rules we have

$$C[g'[\mathcal{F} \leftarrow \Omega]] \equiv C[g'_\Omega] \longrightarrow\!\!\!\!\!\rightarrow h[\mathcal{F} \leftarrow \Omega] \ .$$

Since for each z in $\mathcal{F}$ such that $z \in \mathtt{Bv}(h)$, z is the root of a redex in h, we have that each information regarding h is contained in $h[\mathcal{F} \leftarrow \Omega]$. That is,

$$a \in \mathcal{P}rint(h[\mathcal{F} \leftarrow \Omega]) \ .$$

Thus, $a \in \mathcal{P}rint^*(C[g'_\Omega])$. By Corollary 3.36, $\exists i$ such that $a \in \mathcal{P}rint^*(C[g'_\Omega]^i)$. Since $\exists j, \mathtt{Bl}(C[g'_\Omega]^i) \leq_\omega C[\mathtt{Bl}(g'^j_\Omega)]$, we have that $a \in \mathcal{P}rint^*(C[\mathtt{Bl}(g'^j_\Omega)])$. Moreover, by Lemma 3.38, ω-reductions do not change the meaning of a term. Therefore,

$$a \in \mathcal{P}rint^*(C[\mathtt{Bl}(\omega(g'^j_\Omega))]) \ .$$

□

COROLLARY 3.40. (CONGRUENCE OF $\equiv_{\mathsf{BT}}$) *Given term graphs g and h, if $g \equiv_{\mathsf{BT}} h$ then $C[g] \equiv_{\mathsf{BT}} C[h]$.*

PROOF. Follows from Theorems 3.34 and 3.39. □

Notation: the restriction of the ordering $\leq_\omega$ to Ω-free terms is called the less-sharing ordering, written as $\leq_{\mathsf{sharing}}$.

PROPOSITION 3.41. *Given term graphs g and h, if $g \leq_{\mathsf{sharing}} h$ then*
(i) *if $g \longrightarrow\!\!\!\!\!\rightarrow g_1$ then there exists term graphs g_2, h_1 such that*

$$h \longrightarrow\!\!\!\!\!\rightarrow h_1, g_1 \longrightarrow\!\!\!\!\!\rightarrow g_2 \text{ and } g_2 \leq_{\mathsf{sharing}} h_1 \ .$$

(ii) *if $h \longrightarrow\!\!\!\!\!\rightarrow h_1$ then there exists term graphs g_1 such that*

$$g \longrightarrow\!\!\!\!\!\rightarrow g_1 \text{ and } g_1 \leq_{\mathsf{sharing}} h_1 \ .$$

PROOF.
(i) By induction on the length n of the reduction $g \longrightarrow\!\!\!\!\!\rightarrow g_1$.

$n = 1$. Let $g \longrightarrow g_1$ by reducing redex z. Let h_1 be the term obtained by reducing redex z_1 in h, with $z_1 = \sigma(z)$, with σ the substitution induced by the ordering $g \leq_{\mathsf{sharing}} h$. Let $\mathcal{F} = \{s \mid \sigma(s) = z_1\}$. Since for each s in $\mathcal{F}$, s denotes the root of a redex in g, let g_2 be obtained by reducing all redexes in $\mathcal{F}$. We have: $g_2 \leq_\omega h_1$ and $g_1 \longrightarrow g_2$ by the subcommutativity property.

$n > 1$. Let $g \longrightarrow\!\!\!\!\!\rightarrow g'$ in $n-1$ steps, and $g' \longrightarrow g_1$. By induction hypothesis, $\exists h'_1, g''$ such that

$$g' \longrightarrow\!\!\!\!\!\rightarrow g'', h \longrightarrow\!\!\!\!\!\rightarrow h'_1 \text{ and } g'' \leq_{\mathsf{sharing}} h'_1 \ .$$

From the subcommutative property $\exists g_2$,

$$g_1 \longrightarrow\!\!\!\!\!\rightarrow g_2 \text{ and } g'' \longrightarrow\!\!\!\!\!\rightarrow g_2 \text{ or } g'' \equiv g_2 \ .$$

If $g_2 \equiv g''$ the result is proved, otherwise it follows from the induction hypothesis.

(ii) Since each $z \in \mathrm{Bv}(g)$ such that $\sigma(z)$ is a redex in h, z denotes a redex in g.
□

THEOREM 3.42. *Given term graphs g and h, if $g \leq_{\mathsf{sharing}} h$ then $g \equiv_{\mathsf{BT}} h$.*

PROOF. From the monotonicity of $\sqsubseteq_{\mathsf{BT}}$ with respect to $\leq_\omega$ follows that $g \sqsubseteq_{\mathsf{BT}} h$. The other direction follows from Proposition 3.41(*ii*), and the fact that if $g \leq_{\mathsf{sharing}} h$ then $\mathcal{P}rint(g) = \mathcal{P}rint(h)$. □

COROLLARY 3.43. *All optimizations that do increase sharing in a program are totally correct.*

4. Relation between graph and term rewriting

We want to explore if term graph rewriting is a correct implementation of term rewriting. A term graph rewriting system is correct with respect to a TRS if the following two conditions are satisfied:
(i) each information obtained by reducing a term graph can be obtained by reducing an expansion of the graph in the corresponding TRS;
(ii) each information obtained by reducing a term in a TRS can be obtained by reducing the term in the corresponding term graph rewriting system.
We do not impose any restriction on the set of term graphs; that is, terms may contain cycles, however, their expansions will be TRS terms.

In the next section we introduce the function BT which returns the Böhm tree of a TRS term. The function BT defines a congruence relation on the set of TRS terms, moreover, in [1] we have also shown that the interpretation function BT defines a stable model; that is, for any information, say a, of $C[M]$ there exists a minimum amount of information of M that is sufficient to derive a.

4.1. BÖHM MODEL FOR ORTHOGONAL TRSs

Let $(T(\Sigma), R)$ be a TRS, with $T(\Sigma)$ a set of first-order terms defined over signature Σ, and R a set of rewrite rules. We restrict our attention to orthogonal TRSs, that is, the rules have to be non-overlapping and left-linear. As in Section 3, to a TRS we associate a ω-TRS$=(T_\Omega(\Sigma), R_\omega)$, with R_ω the set of ω-rules associated to each rule in R. Next we define the answer associated to a TRS term.
Notation: to avoid ambiguity, TRS reductions will be denoted by $\longrightarrow_{\mathsf{t}}$. As before, N_Ω is obtained by substituting its redexes by Ω.

DEFINITION 4.1. (ANSWER OF A TRS TERM OR ITS BÖHM TREE) *Given a TRS term M, the answer of M is* $\mathsf{BT}(M) = \{a \mid a \in \omega\text{-}\mathcal{T}rees, a \leq_{\mathsf{t}} \omega(N_\Omega),\ M \longrightarrow\!\!\!\!\!\rightarrow_{\mathsf{t}} N\}$.

Analogous to term graph rewriting system, we can show that the answer of a TRS term defines an interpretation function. The proof methodology is the same as the one developed in Section 3, and most of the proofs carry over to the TRS case. The only difference shows up in the proof of the continuity of the context operation. In particular, we remind the reader that in proving that $C[g] \sqsubseteq_{\mathsf{BT}} \bigsqcup\{C[\mathtt{B1}(a)] \mid a \in \mathcal{P}rint^*(g)\}$ we made use of a property of reductions, namely that a reduction of the form $C[g] \longrightarrow\!\!\!\!\!\rightarrow h$ can be re-ordered

in a way consisting of first reducing all redexes occurring in g and then performing the rest of the reduction. This property follows directly from the subcommutativity property of term graph rewriting systems without overlapping rules. For orthogonal TRSs, due to the duplication of redexes, subcommutativity is lost. Thus, we show in Section 4.3 that TRS reductions can be re-ordered in an inside-out manner.

4.2. Soundness and Completeness of term graph rewriting

Due to the presence of cycles, the translation of a term graph will not always produce a TRS term but a possible infinite sequence of them. To define an infinite term we perform the ideal completion of $\langle T_\Omega, \leq_t \rangle$.

Definition 4.2. (Infinite Term) *The set of infinite terms, called T_Ω^∞, is the ideal completion of T_Ω.*

Definition 4.3. (Answer of an infinite term) *Given an infinite term M, the answer of M is $\mathcal{P}rint_\infty^*(M) = \bigcup \{\mathsf{BT}(t) \mid t \in M\}$.*

Proposition 4.4. *Given an infinite term M, $\mathcal{P}rint_\infty^*(M)$ is in ω-$\mathcal{T}rees^\infty$.*

Proof. We show that $\mathcal{P}rint_\infty^*(M)$ is directed. Let $a, b \in \mathcal{P}rint_\infty^*(M)$, then $\exists t_1, t_2 \in M$ such that $a \in \mathsf{BT}(t_1)$ and $b \in \mathsf{BT}(t_2)$. Since M is directed, $\exists t_3 \in M$ such that $t_1 \leq_t t_3$ and $t_2 \leq_t t_3$. Since BT is monotonic with respect to $\leq_t$, we have: $\mathsf{BT}(t_1) \subseteq \mathsf{BT}(t_3)$ and $\mathsf{BT}(t_2) \subseteq \mathsf{BT}(t_3)$. The result follows from the fact that $\mathsf{BT}(t_3)$ is directed. □

Definition 4.5. *Given a term graph g, the corresponding infinite term is*

$$g^\infty = \{a \mid a \leq_t b,\ a \in T_\Omega,\ b \in \{g^k \mid k \geq 0\}\}\ .$$

Proposition 4.6. *Given a term graph g, g^∞ is in T_Ω^∞.*

Proof. Trivial since $\forall k \geq 0, g^k \leq_t g^{k+1}$. □

The translation between a term graph rule and a TRS rule requires attention because the right-hand side of a term graph rule may contain cycles. As such, the translation of a term graph rule may result in a set of TRS rules. The translation of the left-hand side l of a term graph rule is obtained by unwinding l.

Example 4.7.
(i) The translation of the following term graph rule:

$$\{x = \mathsf{C}(y)\} \longrightarrow \{x = \mathsf{A}(x_1, y), x_1 = \mathsf{B}(x)\}$$

is the set of TRS rules below:

$$\begin{array}{lcl} \mathsf{C}(y) & \longrightarrow & X(y) \\ X(y) & \longrightarrow & \mathsf{A}(X1(y), y) \\ X1(y) & \longrightarrow & \mathsf{B}(X(y)) \end{array}$$

Note that the second and third rule generate all the expansions of the right-hand side of the above term graph rule.

(ii) The translation of the cyclic implementation of the Y-rule, which in term graph notation is expressed as follows:

$$\{x = \mathsf{Ap}(\mathsf{Y}, f)\} \longrightarrow \{x = \mathsf{Ap}(f, x)\}$$

is the set of TRS rules below:

$$\begin{array}{lcl} \mathsf{Ap}(\mathsf{Y}, f) & \longrightarrow & X(f) \\ X(f) & \longrightarrow & \mathsf{Ap}(\mathsf{f}, X(f)) \end{array}$$

DEFINITION 4.8. *Given a term graph rule $\tau : l \longrightarrow r$, with r rooted at x and $\{y_1, \cdots, y_k\} = \mathtt{Fv}(r)$. The translation of term graph rule τ is:*

$$\mathtt{TR}(\tau) = \{\mathtt{Unwind}(l) \longrightarrow X(y_1, \cdots, y_k)\} \cup R_{exp}(r) \ .$$

DEFINITION 4.9. *Given a TRS rule $\tau : l \longrightarrow r$. The corresponding term graph rule is:*

$$\mathtt{Bl}_\mathtt{T}(\tau) = \mathtt{Bl}(l)' \longrightarrow \mathtt{Bl}(r)'$$

with $\mathtt{Bl}(l)'(\mathtt{Bl}(r)')$ denoting the renaming of the root of $\mathtt{Bl}(l)(\mathtt{Bl}(r))$ with a fresh variable x.

Given a set of term graph rules R, let

$$R_\mathtt{t} = \cup\{\mathtt{TR}(\tau) \mid \tau \in R\}$$

and

$$R_\mathtt{g} = \{\mathtt{Bl}_\mathtt{T}(\tau) \mid \tau \in R_\mathtt{t}\} \ .$$

Note that $R_\mathtt{t}$ is a set of TRS rules, while $R_\mathtt{g}$ contains the corresponding term graph rules. We denote the reduction relation on term graphs induced by $R_\mathtt{g}$ as $\longrightarrow\!\!\!\!\!\rightarrow_{R_\mathtt{g}}$, while $\longrightarrow\!\!\!\!\!\rightarrow_{R_\mathtt{t}}$ denotes the reduction relation on TRS terms induced by $R_\mathtt{t}$. Let $\mathcal{P}rint_{R_\mathtt{g}}(h)$ be the printable value of term graph h computed with respect to $R_\mathtt{g}$, and let $\mathcal{P}rint^*_{R_\mathtt{g}}(h)$ be the answer of h with respect to $R_\mathtt{g}$; that is, $\mathcal{P}rint^*_{R_\mathtt{g}}(h) = \bigcup\{\mathcal{P}rint_{R_\mathtt{g}}(h_1) \mid h \longrightarrow\!\!\!\!\!\rightarrow_{R_\mathtt{g}} h_1\}$. (We remind the reader that $\mathcal{P}rint^*(h)$ and $\mathcal{P}rint(h)$ are computed with respect to a set of term graph rules R.) The translation TR does not change the meaning of a term.

PROPOSITION 4.10. *Given a term graph g, $\mathcal{P}rint^*(g) = \bigcup\{\mathcal{P}rint^*_{R_\mathtt{g}}(\mathtt{Bl}(g^k)) \mid k \geq 0\}$.*

PROOF. Follows from Corollary 3.36. □

The right-hand side of rules in $R_\mathtt{g}$ are acyclic, therefore, if g is an acyclic term graph and $g \longrightarrow\!\!\!\!\!\rightarrow_{R_\mathtt{g}} h$, then h is also acyclic.

PROPOSITION 4.11. *Given an acyclic term graph g. If $g \longrightarrow\!\!\!\!\!\rightarrow_{R_\mathtt{g}} h$ then $\mathtt{Unwind}(g) \longrightarrow\!\!\!\!\!\rightarrow_{R_\mathtt{t}} \mathtt{Unwind}(h)$.*

PROOF. The proof is by induction on the number n of reduction steps of $g \longrightarrow\!\!\!\!\!\rightarrow_{R_g} h$. Let $n = 1$, that is, $g \longrightarrow_{R_g} h$ by reducing the redex rooted at z. Then $\mathtt{Unwind}(g) \longrightarrow\!\!\!\!\!\rightarrow_{R_t} \mathtt{Unwind}(h)$ by a complete development of all redexes corresponding to z. The result then follows from the induction hypothesis. □

DEFINITION 4.12. *Given a term graph rewriting system (A_c, R_c) its corresponding TRS is (A, R), where:*
(1) $A = \{a \mid a \in g^\infty, \forall g \in A_c\}$;
(2) $R = \{\tau \mid \tau \in \mathtt{TR}(\tau'), \forall \tau' \in R_c\}$.

DEFINITION 4.13. (SOUNDNESS) *Given a term graph rewriting system $\mathcal{A} = (A_c, R_c)$ and its corresponding TRS $\mathcal{T} = (A, R)$, then $\mathcal{A}$ is sound with respect to $\mathcal{T}$, if $\forall g \in A_c$, $\mathcal{P}rint^*(g) \subseteq \mathcal{P}rint^*_\infty(g^\infty)$.*

DEFINITION 4.14. (COMPLETENESS) *Given a term graph rewriting system $\mathcal{A} = (A_c, R_c)$ and its corresponding TRS $\mathcal{T} = (A, R)$, then $\mathcal{A}$ is complete with respect to $\mathcal{T}$, if $\forall M \in A$, $\mathtt{BT}(M) \subseteq \mathcal{P}rint^*(\mathtt{Bl}(M))$.*

THEOREM 4.15. *A term graph rewriting system $\mathcal{A} = (A_c, R_c)$ is sound with respect to its corresponding TRS.*

PROOF.

$$\begin{aligned} \mathcal{P}rint^*(g) &= \bigcup\{\mathcal{P}rint^*_{R_g}(\mathtt{Bl}(g^i)) \mid g^i \in g^\infty\} && \text{(Proposition 4.10)} \\ &\subseteq \bigcup\{\mathtt{BT}(g^i) \mid g^i \in g^\infty\} && \text{(Proposition 4.11)} \\ &= \mathcal{P}rint^*_\infty(g^\infty)\ . \end{aligned}$$

□

EXAMPLE 4.16.
(i) Consider the rule:

$$\{x = \mathsf{I}(y)\} \longrightarrow \{x = y\}\ .$$

The reduction:

$$g \equiv \{x = \mathsf{I}(x)\} \longrightarrow \bullet$$

is sound because

$$\mathcal{P}rint^*(g) = \mathcal{P}rint^*_\infty(\{\Omega, \mathsf{I}(\Omega), \mathsf{I}(\mathsf{I}(\Omega)), \cdots\}) = \{\Omega\}\ .$$

(ii) Consider the rule

$$\{x = \mathsf{A}(y)\} \longrightarrow \{x = \mathsf{B}(y)\}\ .$$

The reduction

$$g \equiv \{x = \mathsf{A}(x)\} \longrightarrow \{x = \mathsf{B}(x)\}$$

is sound because

$$\mathcal{P}rint^*(g) = \mathcal{P}rint^*_\infty(\{\Omega, \mathsf{A}(\Omega), \mathsf{A}(\mathsf{A}(\Omega)), \cdots\}) = \{\Omega, \mathsf{B}(\Omega), \mathsf{B}(\mathsf{B}(\Omega)), \cdots\}\ .$$

THEOREM 4.17. *A term graph rewriting system $\mathcal{A} = (A_c, R_c)$ is complete with respect to its corresponding TRS.*

PROOF. Let $a \in \mathtt{BT}(M)$, that is, $\exists N, M \longrightarrow\!\!\!\!\!\rightarrow_{\mathtt{t}} N$ and $a \leq_{\mathtt{t}} \omega(N_\Omega)$. By induction on the length n of the reduction $M \longrightarrow\!\!\!\!\!\rightarrow_{\mathtt{t}} N$ we show that $a \in \mathcal{P}rint^*(\mathtt{Bl}(M))$.

$n = 0$. Since $a \in \mathcal{P}rint(\mathtt{Bl}(M))$, then $a \in \mathcal{P}rint^*(\mathtt{Bl}(M))$.
$n > 0$. Let $M \longrightarrow_{\mathtt{t}} N_1$ and $N_1 \longrightarrow\!\!\!\!\!\rightarrow_{\mathtt{t}} N$ in $n-1$ steps. We have

$$\mathtt{Bl}(M) \longrightarrow M_1 \text{ and } \mathtt{Bl}(N_1) \leq_\omega M_1 \ .$$

By induction hyphothesis $a \in \mathcal{P}rint^*(\mathtt{Bl}(N_1))$. By monotonicity of $\sqsubseteq_{\mathsf{BT}}$ with respect to $\leq_\omega$ (Lemma 3.33):

$$a \in \mathcal{P}rint^*(M_1) \ .$$

Follows from the soundness of $\equiv_{\mathsf{BT}}$ (Theorem 3.29) that $a \in \mathcal{P}rint^*(\mathtt{Bl}(M))$.

□

COROLLARY 4.18. *The cyclic implementation of the* Y *rule is totally correct.*

4.3. INSIDE-OUT REDUCTION

We show completeness of inside-out reductions for TRSs by lifting the TRS reduction in the term graph rewriting world, finding the corresponding inside-out reduction and then projecting the reduction so obtained back to the TRS world.

Let us first show that inside-out reductions are complete for term graph rewriting systems containing acyclic terms and right-acyclic rules only (*i.e.*, the right-hand side of a rule does not contain a cycle), hereafter called acyclic term graph rewriting systems. Intuitively, an inside-out reduction consists of reducing a term in a bottom-up fashion. Consider the following rules:

$$\tau_1 : \{x = \mathsf{G}(y)\} \longrightarrow \{x = 0\} \qquad \tau_3 : \{x = \mathsf{H}(y)\} \longrightarrow \{x = y\}$$
$$\tau_2 : \{x = \mathsf{D}(y,z)\} \longrightarrow \{x = y\} \qquad \tau_4 : \{x = \mathsf{F}(y)\} \longrightarrow \{x = \mathsf{D}(y,y)\}$$

and the following term:

$$\begin{aligned} \{\ & x_1 = \mathsf{H}(x_2), \\ & x_2 = \mathsf{D}(x_3, x_4), \\ & x_3 = \mathsf{G}(0), \\ & x_4 = \mathsf{F}(0), \\ & \text{in } x_1\} \end{aligned}$$

The reduction consisting of redexes x_3, x_4, x_2, x_1, will be an inside-out reduction, analogously for x_4, x_3, x_2, x_1 or x_2, x_1. However, x_2, x_4, x_1 is not an inside-out reduction, because after having reduced redex x_2 we reduce the redex x_4 which is below x_2. Notice that a reduction of newly created redexes only is inside-out.

Notation: given a term graph g and redex z occurring in g, we denote as n_z the corresponding node in $\mathcal{G}(g)$, where $\mathcal{G}(g)$ is the graph associated to g (see [1]).

DEFINITION 4.19. (ORDERING ON REDEXES) *Given a term graph g and two distinct redexes z_1, z_2 occur-*

ring in g. We say $z_1 \ll z_2$ in g iff $\exists$ a path in $\mathcal{G}(g)$ from n_{z_1} to n_{z_2}. We denote the transitive closure of $\ll$ with $\ll^+$, and the reflexive and transitive closure of $\ll$ with $\ll^*$.

Notice that in a cyclic term two redexes z_1 and z_2 could be such that $z_1 \ll^+ z_2$ and $z_2 \ll^+ z_1$.

DEFINITION 4.20. (INSIDE-OUT REDUCTION)
Let A be the reduction $g_0 \xrightarrow[z_0]{} g_1 \xrightarrow[z_1]{} \cdots g_i \xrightarrow[z_i]{} \cdots \longrightarrow g_j \xrightarrow[z_j]{} \cdots \longrightarrow g_n$. A is said to be inside-out, written as $g_1 \xrightarrow[io]{} g_n$, iff $\forall i, j$, $0 \leq i \leq j \leq n$, z_j is not a redex occurring in g_i, such that $z_i \ll^+ z_j$.

THEOREM 4.21. (COMPLETENESS OF INSIDE-OUT REDUCTION FOR ACYCLIC TERM GRAPH REWRITING)
Given an acyclic term graph rewriting system. If $g \longrightarrow\!\!\!\!\!\rightarrow h$ then there exists a corresponding inside-out reduction, $g \xrightarrow[io]{} h$.

PROOF. The proof is by induction on the number n of reduction steps of $g \longrightarrow\!\!\!\!\!\rightarrow h$.
The base case is trivial. Suppose it holds for $n-1$, and let A be the reduction

$$g \xrightarrow[z_0]{} g_1 \xrightarrow[z_1]{} \cdots \longrightarrow g_i \xrightarrow[z_i]{} \cdots g_j \xrightarrow[z_j]{} \cdots \longrightarrow g_n \ .$$

Let z_i be an internal redex in g, that is, a redex in g such that $\nexists j \neq i, z_i \ll^+ z_j$. Let A_1^* be the reduction $g \xrightarrow[z_i]{} h_1$, and $A \setminus A_1^*$ be the projection of A with respect to A_1^*, which is well-defined due to the subcommutative property of term graph rewriting systems. Due to the non-duplicative property of term graph rewriting systems [6], we have that the length of the reduction $A \setminus A_1^*$ is less than the length of A; therefore, there exists an inside-out reduction, called A_2^*, of $A \setminus A_1^*$. We define A^* to be the reduction $A_1^* A_2^*$. Since z_i is an internal redex, A^* is an inside-out reduction. □

The completeness of inside-out reductions does not necessarily hold for confluent term graph rewriting systems with overlapping rules, as shown in the example below. Given the rules:

$$\begin{aligned}
\tau_1 &: \{x = \mathsf{F}(x_1), x_1 = \mathsf{G}(y)\} \longrightarrow \{x = x_1\} \\
\tau_2 &: \{x = \mathsf{G}(y)\} \longrightarrow \{x = y\} \\
\tau_3 &: \{x = \mathsf{F}(y)\} \longrightarrow \{x = \mathsf{F}(x_1), x_1 = \mathsf{G}(y)\}
\end{aligned}$$

The following reduction does not have a corresponding inside-out reduction,

$$\{z_1 = \mathsf{F}(z_2), z_2 = \mathsf{G}(y)\} \xrightarrow[z_1]{} \{z_2 = \mathsf{G}(y)\} \xrightarrow[z_2]{} y \ .$$

The problem lies in the fact that a redex is not guaranteed to remain a redex. For example, by performing redex z_2 first, z_1 is not a τ_1-redex anymore.

Notation: $M \xrightarrow[io]{}\!\!\!\!\!\rightarrow_{\mathsf{t}} M'$ denotes an inside-out TRS reduction, which can be defined as in Definition 4.20.

THEOREM 4.22. (COMPLETENESS OF INSIDE-OUT REDUCTION FOR TRS) *Given a TRS, if $M \longrightarrow\!\!\!\!\!\rightarrow_{\mathsf{t}} N$, then $\exists M'$, such that $M \xrightarrow[io]{}\!\!\!\!\!\rightarrow_{\mathsf{t}} M'$ and $N \longrightarrow\!\!\!\!\!\rightarrow_{\mathsf{t}} M'$.*

PROOF. From Proposition 3.41 and Proposition 4.11 we can construct a term graph reduction such that

$$\mathtt{Bl}(M) \longrightarrow\!\!\!\!\!\rightarrow g \text{ and } N \longrightarrow\!\!\!\!\!\rightarrow_{\mathsf{t}} \mathtt{Unwind}(g) \ .$$

We reorder the above term graph reduction in an inside-out manner (Theorem 4.21). The reduction so obtained is projected in the TRS world (Proposition 4.11). This final reduction is inside-out because the unwinding preserves the ordering among redexes. □

Note that the stronger result: $M \longrightarrow\!\!\!\!\!\rightarrow_t N \Longrightarrow M \underset{io}{\longrightarrow\!\!\!\!\!\rightarrow}_t N$ does not holds. Consider the TRS rules:

$$\begin{aligned} \mathsf{A}(0) &\longrightarrow 1 \\ \mathsf{F}(x) &\longrightarrow \mathsf{G}(x,x) \end{aligned}$$

then the following reduction can not be ordered in an inside-out manner:

$$\mathsf{F}(\mathsf{A}(0)) \longrightarrow_t \mathsf{G}(\mathsf{A}(0),\mathsf{A}(0)) \longrightarrow_t \mathsf{G}(\mathsf{A}(0),1).$$

Conclusions

We have presented the semantic properties of orthogonal (cyclic) graph rewriting and shown the correctness of graph rewriting with respect to term rewriting. The motivation for this work came from a desire to formalize the operational semantics of languages such as Id [28] and pH (a parallel variant of Haskell [20]). These languages depart from purely functional languages by allowing a restricted form of assignment, namely the creation of an unbound variable which can be defined later on. (In a purely functional language a variable is given exactly one binding or definition at creation time.) This, however, comes at the cost of destroying referential transparency, that is, the definition of an identifier can not be substituted for each occurrence of the identifier in an unrestricted manner. As such, these languages need a precise treatment of sharing. A term graph rewriting constitutes a suitable intermediate language for the compilation of the above mentioned languages. Moreover, we can express optimizations in terms of source-to-source transformations on the intermediate language. The term model developed in this paper provides the right criterion for expressing the total correctness of these optimizations. In particular, we are able to show total correctness for the optimizations that increase the sharing in a term and for the cyclic version of the Y-rule.

The author is currently working, in collaboration with Jan Willem Klop, on extending the work to include λ-abstraction. We would also like to provide a theory that covers multi-rooted rules. Multi-rooted rules are needed to express side-effect operations. This would provide a sound mathematical basis for Id and parallel Haskell. It would also be interesting to investigate the suitability of term graph rewriting systems as an intermediate language for other classes of languages, such as imperative languages.

5. Acknowledgements

Funding for this work has been provided by the NSF grant CCR-94-10237.

Many thanks to Arvind, Jan Willem Klop, Art Farley, Evan Tick, and Will Clinger for reading a preliminary version of this paper. In addition, we thank the anonymous referees for suggesting significant improvements.

References

1. Z. M. Ariola. *An Algebraic Approach to the Compilation and Operational Semantics of Functional Languages with I-structures.* PhD thesis, MIT Technical Report TR-544, 1992.
2. Z. M. Ariola and Arvind. P-TAC: A parallel intermediate language. In *Proc. ACM Conference on Functional Programming Languages and Computer Architecture, London*, September 1989.
3. Z. M. Ariola and Arvind. Compilation of Id. In *Proc. of the Fourth Worskshop on Languages and Compilers for Parallel Computing, Santa Clara, California, Springer-Verlag LNCS 589*, August 1991.
4. Z. M. Ariola and Arvind. A syntactic approach to program transformations. In *Proc. ACM SIGPLAN Symposium on Partial Evaluation and Semantics Based Program Manipulation, Yale University, New Haven, CT*, June 1991.
5. Z. M. Ariola and Arvind. Graph rewriting systems for efficient compilation. In M. R. Sleep, M. J. Plasmeijer, and M. C. D. J. van Eekelen, editors, *Term Graph Rewriting: Theory and Practice*, pages 77–90. John Wiley & Sons, 1993.
6. Z. M. Ariola and Arvind. Properties of a first-order functional language with sharing. *Theoretical Computer Science*, 146, 1995.
7. Z. M. Ariola, C. Braun, and J. W. Klop. A graph rewriting system to express state and sequentialization in a parallel setting. In *Proc. 5th International Workshop on Graph Grammars and their Application to Computer Science, Williamsburg, Virgina*, 1994.
8. Z. M. Ariola and J. W. Klop. Equational term graph rewriting. *To appear in Fundamenta Informaticae. Extended version: CWI Report CS-R9552.* 1995.
9. H. Barendregt, T. Brus, M. van Eekelen, J. Glauert, J. Kennaway, M. van Leer, M. Plasmeijer, and M. R. Sleep. Towards an intermediate language based on graph rewriting. In *Proc. Conference on Parallel Architecture and Languages Europe (PARLE '87), Eindhoven, The Netherlands, Springer-Verlag LNCS 259*, June 1987.
10. H. P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics.* North-Holland, Amsterdam, 1984.
11. 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. In J. W. de Bakker, A. J. Nijman, and P. C. Treleaven, editors, *Proc. Conference on Parallel Architecture and Languages Europe (PARLE '87), Eindhoven, The Netherlands, Springer-Verlag LNCS 259*, pages 141–158, 1987.
12. H. Ehrig. Introduction to the algebraic theory of graph grammars. In *Proc. 1st International Workshop on Graph Grammars and their Application to Computer Science, Springer-Verlag LNCS 73*, pages 1–69, 1989.
13. H. Ehrig, M. Nagl, and G. Rozenberg, editors. *Proc. 2nd International Workshop on Graph Grammars and their Application to Computer Science, Haus Ohrbeck, Germany, Springer-Verlag LNCS 153.* 1983.
14. H. Ehrig, M. Nagl, G. Rozenberg, and A. Rosenfeld, editors. *3nd International Workshop on Graph Grammars and their Application to Computer Science, Warrenton, Virginia, USA Springer-Verlag LNCS 291.* 1987.
15. H. Ehrig, M. Nagl, G. Rozenberg, and A. Rosenfeld, editors. *Proc. 4th International Workshop on Graph Grammars and their Application to Computer Science, Bremen, Germany, Springer-Verlag LNCS 532.* 1991.

16. W. M. Farmer. A correctness proof for combinator reduction with cycles. *ACM Transactions on Programming Languages and Systems*, 12(1):123–134, 1990.
17. W. M. Farmer and R. J. Watro. Redex capturing in term graph rewriting. In R. V. Book, editor, *Proc. 4th International Conference on Rewriting Techniques and Applications (RTA-91), Como, Italy, Springer-Verlag LNCS 488*, pages 13–24, 1991.
18. J. R. W. Glauert, J. R. Kennaway, and M. R. Sleep. Dactl: An experimental graph rewriting language. In *Proc. 4th International Workshop on Graph Grammars and their Application to Computer Science, Bremen, Germany, Springer-Verlag LNCS 532*, pages 378–395, 1990.
19. M. Hennessy. *Algebraic Theory of Processes.* MIT Press, 1988.
20. P. Hudak, S. Peyton Jones, P. Wadler, B. Boutel, J. Fairbairn, J. Fasel, K. Hammond, J. Hughes, T. Johnsson, D. Kieburtz, R. Nikhil, W. Partain, and J. Peterson. Report on the programming language Haskell. *ACM SIGPLAN Notices*, 27(5):1–64, 1992.
21. G. Huet and J.-J. Lévy. Computations in orthogonal rewriting systems 1 and 2. In *Computational logic. Essays in Honor of Alan Robinson. Ed. J.-L. Lassez & G.D. Plotkin*, 1991.
22. J. R. Kennaway. On graph rewriting. *Theoretical Computer Science*, 52:37–58, 1987.
23. J. R. Kennaway, J. W. Klop, M. R. Sleep, and F. J. de Vries. The adequacy of term graph rewriting for simulating term rewriting. *Transactions on Programming Languages and Systems*, 16(3):493–523, 1994.
24. J. R. Kennaway, J. W. Klop, M. R. Sleep, and F. J. de Vries. Transfinite reductions in orthogonal term rewriting systems. *Information and Computation*, 119(1), 1995.
25. J. W. Klop. Term rewriting systems. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science*, volume II, pages 1–116. Oxford University Press, 1992.
26. J.-J. Lévy. *Réductions Correctes et Optimales dans le Lambda-Calcul.* PhD thesis, Universite Paris VII, October 1978.
27. M. Löwe. Algebraic approach to single pushout graph transformation. *Theoretical Computer Science*, 109:181–224, 1993.
28. R. S. Nikhil. Id (Version 90.1) Reference Manual. Technical Report CSG Memo 284-2, MIT Laboratory for Computer Science, 545 Technology Square, Cambridge, MA 02139, USA, July 1991.
29. M. J. Plasmeijer and M. C. J. D. van Eekelen. *Functional Programming and Parallel Graph Rewriting.* Addison Wesley, 1993.
30. J. C. Raoult. On graph rewritings. *Theoretical Computer Science*, 32:1–24, 1984.
31. M. R. Sleep, M. J. Plasmeijer, and M. C. D. J. van Eekelen, editors. *Term Graph Rewriting: Theory and Practice.* John Wiley & Sons, 1993.
32. J. E. W. Smetsers. *Graph Rewriting and Functional Languages.* PhD thesis, University of Nijmegen, 1993.
33. C. Wadsworth. *Semantics and Pragmatics of the Lambda-Calculus.* 1971. PhD thesis, University of Oxford.
34. C. Wadsworth. The relation between computational and denotational properties for scott's d_∞-models of the lambda-calculus. *Theoretical Computer Science*, 5, 1976.
35. C. Wadsworth. Approximate reduction and lambda calculus models. *Theoretical Computer Science*, 7, 1978.

36. P. Welch. Continuous semantics and inside-out reductions. In *λ-Calculus and Computer Science Theory, Italy (Springer-Verlag LNCS 37)*, March 1975.