URL: http://www.elsevier.nl/locate/entcs/volume51.html 12 pages

Solving Equations by Graph Transformation ⋆

Annegret Habel [1]

Fachbereich Informatik
Universität Oldenburg
Postfach 2503, D-26111 Oldenburg, Germany

Detlef Plump [2]

Department of Computer Science
The University of York
Heslington, York YO10 5DD, United Kingdom

Abstract

We review the concept of term graph narrowing as an approach for solving equations by transformations on term graphs. Term graph narrowing combines term graph rewriting with first-order term unification. This mechanism is complete for all term rewriting systems over which term graph rewriting is normalizing and confluent. This includes, in particular, all convergent term rewriting systems. Completeness means that for every solution of a given equation, term graph narrowing can find an equivalent or more general solution. The general motivation for using term graphs instead of terms is to improve efficiency: sharing common subterms saves space and avoids the repetition of computations.

1 Introduction

Narrowing was devised in the field of theorem proving as an equation solving method for the case that an equational theory is represented by a convergent term rewriting system. Fay [7] was the first to show the completeness of narrowing. In order to reduce the search space of the narrowing procedure, Hullot [13] considered a strategy called basic narrowing and showed that it is still complete. Later, narrowing became popular as the computational paradigm for the combination of functional and logic programming. Since then there

⋆ Research partially supported by the TMR Network GETGRATS and the ESPRIT Working Group APPLIGRAPH.

[1] Email: habel@informatik.uni-oldenburg.de
[2] Email: det@cs.york.ac.uk

has been much research activity on improving the efficiency of narrowing and on relaxing the requirements for completeness (see the survey of Hanus [11]).

In order to implement narrowing efficiently, it is advisable to represent terms by graph-like data structures. This is because the simple tree representation of terms enforces copying of subterms in rewrite steps and hence leads to multiplication of evaluation work. We present *term graph narrowing* as an approach for solving equations by transformations on term graphs. Term graph narrowing is complete for all term rewriting systems over which term graph rewriting is normalizing and confluent. This includes, in particular, all convergent term rewriting systems. Completeness means that if an equation is represented by a term graph, then for every solution of this equation there is a narrowing derivation starting from this graph that computes an equivalent or more general solution. In other words, every solution of a given equation is equivalent to an instance of a solution generated by narrowing.

Term graph narrowing combines term graph rewriting with first-order term unification (see [19] for an overview on term graph rewriting and [22] for a collection of papers from that area). We use the term graph rewriting model studied in [18,19]. It allows, besides applications of rewrite rules, collapsing steps on term graphs to increase the degree of sharing. This model is complete for equational reasoning in the same sense as term rewriting is. The completeness proof for term graph narrowing given in [9] exploits existing results on the relation between term graph rewriting and term rewriting with respect to termination, confluence and related properties.

This paper is organized as follows: In Section 2 we briefly review term graph rewriting, refering to [19,20] for details. Section 3 introduces term graph narrowing on the basis of term graph rewriting and term graph substitutions. In Section 4 we discuss the completeness of two restricted forms of term graph narrowing, called *minimally collapsing* and *maximally collapsing* narrowing. Finally, in Section 5 we sketch the notion of basic term graph narrowing and refer to some related work. Our presentation is based on [10,9].

2 Term graph rewriting

Let Σ and X be disjoint sets of *function symbols* and *variables*, respectively, where each function symbol f comes with a natural number $\mathrm{arity}(f) \geq 0$ and variables have arity 0. Function symbols of arity 0 are called *constants*.

A *hypergraph* over $\Sigma \cup \mathrm{X}$ is a system $G = \langle \mathrm{V}_G, \mathrm{E}_G, \mathrm{lab}_G, \mathrm{att}_G \rangle$ consisting of two finite sets V_G and E_G of *nodes* and *hyperedges*, a labelling function $\mathrm{lab}_G\colon \mathrm{E}_G \to \Sigma \cup \mathrm{X}$, and an attachment function $\mathrm{att}_G\colon \mathrm{E}_G \to \mathrm{V}_G^*$ which assigns a string of nodes to a hyperedge e such that the length of $\mathrm{att}_G(e)$ is $1 + \mathrm{arity}(\mathrm{lab}_G(e))$. From now on hypergraphs and hyperedges are simply called graphs and edges.

Given a graph G and an edge e with $\mathrm{att}_G(e) = v\,v_1 \dots v_n$, node v is the *result node* of e while $v_1, \dots, v_n$ are the *argument nodes*. The result node v

is denoted by $\mathrm{res}(e)$. For each node v, $G[v]$ is the subgraph consisting of all nodes that are reachable from v and all edges having these nodes as result nodes.

Definition 2.1 (Term graph) *A graph G is a* term graph *if there is a node* root_G *from which each node is reachable, G is acyclic, and each node is the result node of a unique edge.*

In pictures we represent edges by their labels and omit nodes (as there is a one-to-one correspondence between edges and nodes). Arrows point to the arguments of a function symbol, where the order among the arguments is given by the left-to-right order of the arrows leaving the symbol.

Definition 2.2 (Term representation) *A node v in a term graph G represents the term* $\mathrm{term}_G(v) = \mathrm{lab}_G(e)(\mathrm{term}_G(v_1), \ldots, \mathrm{term}_G(v_n))$*, where e is the unique edge with* $\mathrm{res}(e) = v$*, and where* $\mathrm{att}_G(e) = v\, v_1 \ldots v_n$*. We abbreviate* $\mathrm{term}_G(\mathrm{root}_G)$ *by* $\mathrm{term}(G)$*.*

A *graph morphism* $f\colon G \to H$ between two graphs G and H consists of two functions $f_{\mathrm{V}}\colon \mathrm{V}_G \to \mathrm{V}_H$ and $f_{\mathrm{E}}\colon \mathrm{E}_G \to \mathrm{E}_H$ that preserve labels and attachment to nodes, that is $\mathrm{lab}_H \circ f_{\mathrm{E}} = \mathrm{lab}_G$ and $\mathrm{att}_H \circ f_{\mathrm{E}} = f_{\mathrm{V}}^* \circ \mathrm{att}_G$ (where $f_{\mathrm{V}}^*\colon \mathrm{V}_G^* \to \mathrm{V}_H^*$ maps a string $v_1 \ldots v_n$ to $f_{\mathrm{V}}(v_1) \ldots f_{\mathrm{V}}(v_n)$). The morphism f is *injective* (*surjective*) if f_{V} and f_{E} are. If f is injective and surjective, then it is an *isomorphism*. In this case G and H are *isomorphic*, which is denoted by $G \cong H$.

Definition 2.3 (Collapsing) *Given two term graphs G and H, G* collapses *to H if there is a graph morphism $G \to H$ mapping* root_G *to* root_H*. This is denoted by $G \succeq H$ or, if the morphism is non-injective, by $G \succ H$. The latter kind of collapsing is said to be* proper*. A term graph G is* fully collapsed *if there is no H with $G \succ H$, while G is a* tree *if there is no H with $H \succ G$.*

It is easy to see that the collapse morphisms are the surjective morphisms between term graphs and that $G \succeq H$ implies $\mathrm{term}(G) = \mathrm{term}(H)$.

A *term rewrite rule* $l \to r$ consists of two terms l and r over Σ and X such that l is not a variable and all variables in r occur also in l. A set $\mathcal{R}$ of term rewrite rules is a *term rewriting system*. In the following let $\mathcal{R}$ denote an arbitrary term rewriting system. The term rewrite relation associated with $\mathcal{R}$ is denoted by $\to$ and its reflexive-symmetric-transitive closure by $\leftrightarrow^*$ (see [2] for an introduction to term rewriting).

Given a term t, denote by $\Diamond t$ a term graph representing t such that only variables are shared. The graph resulting from $\Diamond t$ after removing all edges labelled with variables is denoted by $\underline{\Diamond t}$.

Definition 2.4 (Redex) *Let G be a term graph, v be a node in G, and $l \to r$ be a rule in $\mathcal{R}$. The pair $\langle v,\, l \to r\rangle$ is a* redex *if there is a graph morphism* $\mathrm{red}\colon \underline{\Diamond l} \to G$*, called the* redex morphism*, such that* $\mathrm{red}(\mathrm{root}_{\Diamond l}) = v$*.*

Definition 2.5 (Term graph rewriting) *Let G and H be term graphs and $\langle v,\, l \to r\rangle$ be a redex in G with redex morphism* $\mathrm{red}\colon \underline{\Diamond l} \to G$. *Then there is a* proper rewrite step $G \Rightarrow_{v,l\to r} H$ *if H is constructed from G as follows: (1) G_1 is the graph obtained from G by removing the unique edge whose result node is v. (2) G_2 is the graph obtained from the disjoint union $G_1 + \underline{\Diamond r}$ by merging v with* $\mathrm{root}_{\Diamond r}$, *and merging the image of u with u', for all nodes u in $\Diamond l$ and u' in $\Diamond r$ such that* $\mathrm{term}_{\Diamond l}(u) = \mathrm{term}_{\Diamond r}(u') \in \mathrm{X}$. *(3) $H = G_2[\mathrm{root}_G]$ is the term graph obtained from G_2 by removing all nodes and edges not reachable from* root_G *("garbage collection").*

We define the term graph rewrite relation $\Rightarrow_{\mathrm{coll}}$ by adding proper collapse steps: $G \Rightarrow_{\mathrm{coll}} H$ if $G \succ H$ or $G \Rightarrow_{v,l\to r} H$ for some redex $\langle v,\, l \to r\rangle$.

3 Term graph narrowing

We will need substitutions replacing variables in term graphs by term graphs. A pair x/G consisting of a variable x and a term graph G is a *substitution pair*. It is applied to an x-labelled edge e in a term graph H by removing e, adding (disjointly) G, and identifying $\mathrm{res}(e)$ with root_G.

Definition 3.1 (Term graph substitution) *A* term graph substitution *is a finite set $\alpha = \{x_1/G_1, \ldots, x_n/G_n\}$ of substitution pairs such that $x_1, \ldots, x_n$ are pairwise distinct and $x_i \neq \mathrm{term}(G_i)$ for $i = 1, \ldots, n$. Given a term graph H, applying $x_1/G_1, \ldots, x_n/G_n$ simultaneously to all edges labelled with $x_1, \ldots, x_n$ yields the term graph $H\alpha$.*

The *domain* of α is the set $\mathrm{Dom}(\alpha) = \{x_1, \ldots, x_n\}$, and the *composition* of α with a term graph substitution β is defined by

$$\alpha\beta = \{x/G\beta \mid x/G \in \alpha \text{ and } x \neq \mathrm{term}(G\beta)\} \cup \{y/H \in \beta \mid y \notin \mathrm{Dom}(\alpha)\}$$

to satisfy $H(\alpha\beta) = (H\alpha)\beta$ for every term graph H.

A term graph substitution α induces the term substitution α^{term} mapping x_i to $\mathrm{term}(G_i)$, for $i = 1, \ldots, n$, and each other variable to itself. Given a term substitution σ and a term t, we will write $t\sigma$ in place of $\sigma(t)$. We may represent σ by the set $\{x_1/t_1, \ldots, x_n/t_n\}$ if $x_i\sigma = t_i$ for $i = 1, \ldots, n$ and $x\sigma = x$ for each other variable x.

A *variant* of a term rewrite rule $l \to r$ is a rule of the form $l\sigma \to r\sigma$, where σ is an injective substitution mapping variables to variables. A set of terms $\{t_1, \ldots, t_n\}$ is *unifiable* if there is a substitution σ such that $t_1\sigma = t_2\sigma = \ldots = t_n\sigma$. In this case σ can be chosen as a *most general unifier*, meaning that for every substitution τ with $t_1\tau = t_2\tau = \ldots = t_n\tau$ there exists a substitution ρ such that $\tau = \rho \circ \sigma$ (see for example [2]).

Definition 3.2 (Term graph narrowing) *Let G and H be term graphs, U a set of non-variable nodes in G, $l \to r$ a variant of a rule in $\mathcal{R}$, and α a term graph substitution. There is a* narrowing step $G \leadsto_{U,l\to r,\alpha} H$ *if α^{term} is* *a most general unifier of* $\{\mathrm{term}_G(u) \mid u \in U\} \cup \{l\}$, *and*

$$G\alpha \succeq G' \underset{v,\, l \to r}{\Longrightarrow} H$$

for some collapsing $c\colon G\alpha \to G'$ *such that* $U = \{\overline{v} \mid c(\overline{v}) = v\}$.

We denote such a step also by $G \rightsquigarrow_\alpha H$. A *term graph narrowing derivation* is sequence of the form $G = G_1 \rightsquigarrow_{\alpha_1} G_2 \rightsquigarrow_{\alpha_2} \dots \rightsquigarrow_{\alpha_{n-1}} G_n = H$. It may be denoted by $G \rightsquigarrow_\alpha^* H$, where $\alpha = \alpha_1\alpha_2 \dots \alpha_{n-1}$ if $n \geq 2$ and $\alpha = \emptyset$ if $n = 1$.

From now on we assume that $\mathcal{R}$ contains the rule $\mathtt{x} =^? \mathtt{x} \to \mathtt{true}$, where the binary symbol $=^?$ and the constant $\mathtt{true}$ do not occur in any other rule. A *goal* is a term of the form $s =^? t$ such that s and t do not contain $=^?$ and $\mathtt{true}$. A *solution* of this goal is a substitution σ satisfying $s\sigma \leftrightarrow^* t\sigma$.

Example 3.3 *Let* $\mathcal{R}$ *consist of the following rules:*

$$\mathtt{0} + \mathtt{x} \to \mathtt{x}$$

$$\mathtt{s}(\mathtt{x}) + \mathtt{y} \to \mathtt{s}(\mathtt{x} + \mathtt{y})$$

$$\mathtt{0} \times \mathtt{x} \to \mathtt{0}$$

$$\mathtt{s}(\mathtt{x}) \times \mathtt{y} \to (\mathtt{x} \times \mathtt{y}) + \mathtt{y}$$

$$\mathtt{x} =^? \mathtt{x} \to \mathtt{true}$$

Suppose that we want to solve the goal $(\mathtt{z} \times \mathtt{z}) + (\mathtt{z} \times \mathtt{z}) =^? \mathtt{s}(\mathtt{z})$. *Figure 1 shows a term graph narrowing derivation starting from the fully collapsed term graph representing this goal. Figure 2 gives the applied rewrite rules and the involved term substitutions. In each step, the set* U *of Definition 3.2 is a singleton. Note that steps (c), (d) and (e) are nothing but* $\Rightarrow$*-steps, and that step (f) consists of a collapse step followed by a* $\Rightarrow$*-step. The derivation computes the term substitution* $\{\mathtt{x}/\mathtt{0}, \mathtt{x}'/\mathtt{s}(\mathtt{0}), \mathtt{y}/\mathtt{s}(\mathtt{0}), \mathtt{z}/\mathtt{s}(\mathtt{0})\}$. *Restricting this substitution to the variables of the goal yields the solution* $\{\mathtt{z}/\mathtt{s}(\mathtt{0})\}$. *Solving the same goal by term-based narrowing requires nine steps, demonstrating that term graph narrowing can speed up the computation of solutions.*

Theorem 3.4 (Soundness and completeness of narrowing) *Let* G *be a term graph such that* $\mathrm{term}(G)$ *is a goal* $s =^? t$.

(1) If $G \rightsquigarrow_\alpha^* \triangle\mathtt{true}$[3]*, then* α^{term} *is a solution of* $s =^? t$.

(2) If $\Rightarrow_{\mathrm{coll}}$ *is normalizing and confluent, then for every solution* σ *of* $s =^? t$ *there exists a narrowing derivation* $G \rightsquigarrow_\beta^* \triangle\mathtt{true}$ *such that* $\beta^{\mathrm{term}} \leq_{\mathcal{R}} \sigma\ [\mathrm{Var}(G)]$[4]*.*

In the sequel, we will refer to the conclusion of statement (2) as *completeness* of term graph narrowing.

---

[3] We denote by $\triangle\mathtt{true}$ a term graph representing $\mathtt{true}$.

[4] Given substitutions σ and τ, and $V \subseteq \mathrm{X}$, we write $\sigma =_{\mathcal{R}} \tau\ [V]$ if $x\sigma \leftrightarrow^* x\tau$ for each $x \in V$, and $\sigma \leq_{\mathcal{R}} \tau\ [V]$ if there is a substitution ρ such that $\sigma\rho =_{\mathcal{R}} \tau\ [V]$. The set of variables occurring in a term graph G is denoted by $\mathrm{Var}(G)$, that is, $\mathrm{Var}(G) = \mathrm{lab}_G(\mathrm{E}_G) \cap \mathrm{X}$.

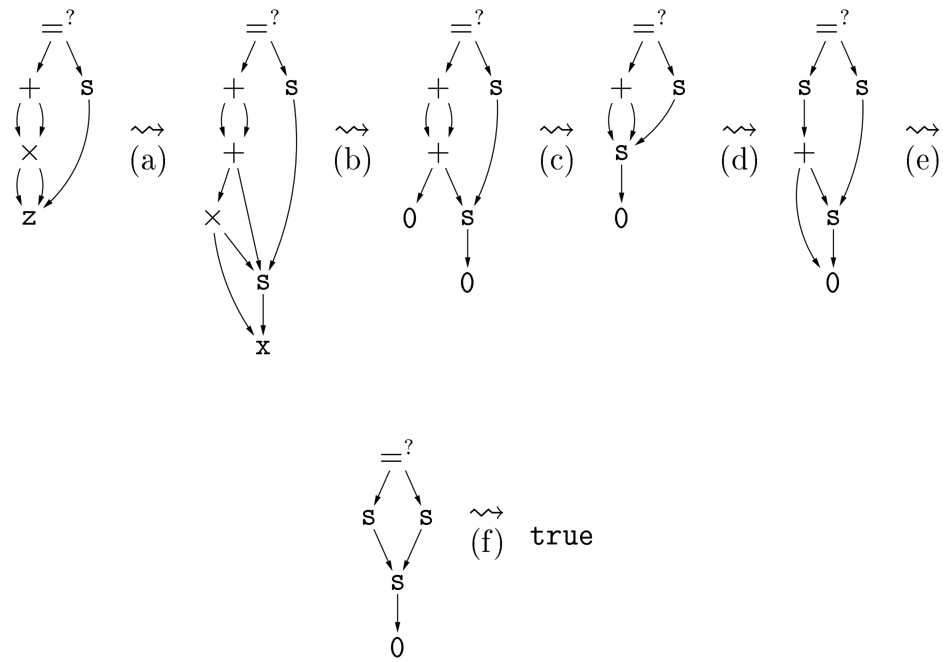

Fig. 1. A term graph narrowing derivation

step	rewrite rule	substitution
(a)	$\mathtt{s}(\mathtt{x}) \times \mathtt{y} \to (\mathtt{x} \times \mathtt{y}) + \mathtt{y}$	$\{\mathtt{y}/\mathtt{s}(\mathtt{x}),\ \mathtt{z}/\mathtt{s}(\mathtt{x})\}$
(b)	$0 \times \mathtt{x}' \to 0$	$\{\mathtt{x}/0,\ \mathtt{x}'/\mathtt{s}(0)\}$
(c)	$0 + \mathtt{x} \to \mathtt{x}$	$\{\mathtt{x}/\mathtt{s}(0)\}$
(d)	$\mathtt{s}(\mathtt{x}) + \mathtt{y} \to \mathtt{s}(\mathtt{x} + \mathtt{y})$	$\{\mathtt{x}/0,\ \mathtt{y}/\mathtt{s}(0)\}$
(e)	$0 + \mathtt{x} \to \mathtt{x}$	$\{\mathtt{x}/\mathtt{s}(0)\}$
(f)	$\mathtt{x} =^? \mathtt{x} \to \mathtt{true}$	$\{\mathtt{x}/\mathtt{s}(\mathtt{s}(0))\}$

Fig. 2. Rewrite rules and substitutions of the derivation in Figure 1

4 Minimally and maximally collapsing narrowing

In this section we consider the completeness of two restricted forms of term graph narrowing where all steps contain a minimal or maximal collapsing, respectively.

Definition 4.1 (Minimal collapsing) *A collapsing $G \succeq M$ is* minimal *with respect to a redex $\langle v, l \to r\rangle$ in M if for each term graph M' with $G \succeq M' \succ M$ and each preimage v' of v in M', the pair $\langle v', l \to r\rangle$ is not a redex.*

In particular, if G equals M, then $G \succeq M$ is minimal since no M' with $G \succeq M' \succ M$ exists. A proper collapsing $G \succ M$ is minimal only if $l \to r$ is not left-linear and cannot be applied at any preimage of v in G.

Definition 4.2 (Minimally collapsing narrowing) *A term graph narrowing derivation is* minimally collapsing *if for each narrowing step* $G \mapsto G\alpha \succeq G' \Rightarrow_{v,l\to r} H$, *the collapsing* $G\alpha \succeq G'$ *is minimal with respect to the redex* $\langle v,\, l \to r\rangle$.

For example, the derivation of Figure 1 is minimally collapsing. Note that in a minimally collapsing step $G \rightsquigarrow_{U,l\to r,\alpha} H$, the set U must be a singleton. It turns out that Theorem 3.4 can be strengthened by replacing unrestricted term graph narrowing with minimally collapsing narrowing.

Theorem 4.3 (Completeness of minimally collapsing narrowing) *Minimally collapsing narrowing is complete whenever* $\Rightarrow_{\mathrm{coll}}$ *is normalizing and confluent.*

We now turn to maximally collapsing narrowing, that is, we consider narrowing derivations in which all involved collapse steps yield fully collapsed term graphs.

Definition 4.4 (Maximally collapsing narrowing) *A term graph narrowing derivation is* maximally collapsing *if for each narrowing step* $G \mapsto G\alpha \succeq G' \Rightarrow_{v,l\to r} H$, *the term graph* G' *is fully collapsed.*

Example 4.5 *Consider the rules*

$$\mathtt{exp(0)} \to \mathtt{s(0)}$$
$$\mathtt{exp(s(x))} \to \mathtt{exp(x)} + \mathtt{exp(x)}$$

specifying the function exp: $n \mapsto 2^n$ *on natural numbers. Figure 3 demonstrates that maximally collapsing narrowing can solve a goal of the form*

$$\mathtt{exp(x)} =^? \underbrace{\mathtt{s(0)} + \ldots + \mathtt{s(0)}}_{2^n\text{-}times}$$

in $n+2$ *steps if the goal is suitably represented. (Substitutions are represented only by those parts affecting the variables in the graphs.) In contrast, both tree-based narrowing and minimally collapsing narrowing need a number of steps exponential in* n *to solve such a goal.*

While minimally collapsing narrowing is complete when term graph rewriting is normalizing and confluent, a counterexample (see [10], Example 30) shows that this is not the case for maximally collapsing narrowing.

Completeness of maximally collapsing narrowing can be ensured, however, by strengthening normalization to termination.

Theorem 4.6 (Completeness of maximally collapsing narrowing) *Maximally collapsing narrowing is complete whenever* $\Rightarrow_{\mathrm{coll}}$ *is convergent.*

In particular, maximally collapsing narrowing is complete for all convergent term rewriting systems since $\Rightarrow_{\mathrm{coll}}$ is convergent over those systems [19].

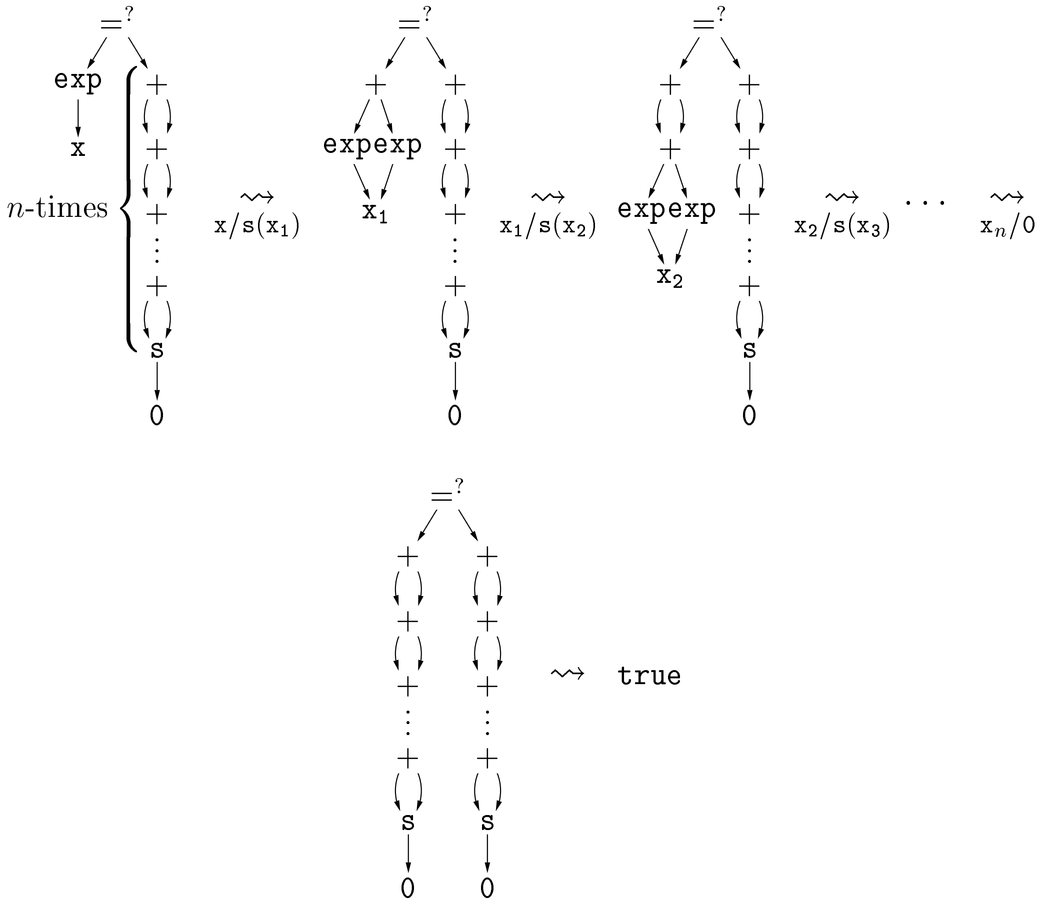

Fig. 3. A maximally collapsing narrowing derivation

5 Bibliographic notes and concluding remarks

In [8] the concept of graph substitution introduced in [21] is applied to term graphs, yielding a uniform framework for unification, rewriting, and narrowing on term graphs. The notion of substitution allows definitions of these concepts that are close to the corresponding definitions in the term world.

In [9] term graph narrowing is introduced as an approach for solving equations by transformations on term graphs. A narrowing step consists of three components: a substitution step, a collapsing step, and a proper rewrite step. The collapsing *after* application of the substitution step is necessary to make narrowing complete.[5] The main result in [9] is that this mechanism is complete for all term rewriting systems over which term graph rewriting is normalizing and confluent. The completeness proof is based on a transformation of rewrite derivations into minimally collapsing ones ([9], Theorem 4.8) and a lifting lemma for minimally collapsing rewrite derivations ([9], Lemma 6.4) which allows to lift minimally collapsing rewrite derivations to (minimally collapsing) narrowing derivations.

[5] The definition of term graph narrowing in [8] is not sufficient in this respect.

In [10] several strategies for term graph narrowing are investigated with respect to completeness. The completeness proofs are based on a lifting lemma for arbitrary rewrite derivations ([10], Lemma 15) which lifts rewrite derivations to narrowing derivations such that the number of proper rewrite steps and the number of narrowing steps coincide. To obtain this relationship, the definition of term graph narrowing in [9] (Definition 5.1), in the following called *elementary* narrowing, is extended to *simultaneous* term graph narrowing in [10] (Definition 7): Instead of a single non-variable node one may choose a non-empty set U of non-variable nodes, compute a most general unifier of the terms $\mathrm{term}_G(u)$ (for u in U) and the left-hand side of a rewrite rule, collapse at least the nodes in U, and apply the rewrite rule to the image of U.

We conjecture that every simultaneous narrowing step $G \rightsquigarrow_\alpha H$ can be transformed into an elementary narrowing derivation $G \rightsquigarrow_\beta^* H'$ such that there is a term graph substitution γ with $H'\gamma \succeq H$ and $\beta\gamma = \alpha[\mathrm{Var}(G)]$. This should permit a transfer of the completeness results from simultaneous narrowing to elementary narrowing.

Basic term graph narrowing, an analogue to basic term-based narrowing [13], is investigated in [15,10]. Roughly speaking, this strategy forbids narrowing steps at nodes that have been created by the substitutions of previous steps. In [10], basic term graph narrowing is proven to be complete for innermost normalizing and confluent graph rewriting. Moreover, the combination of basic narrowing with two strategies for controlling sharing is considered, obtaining minimally collapsing and maximally collapsing basic narrowing. The former is shown to be complete in the presence of innermost normalization and confluence, the latter in the presence of termination and confluence. Maximally collapsing narrowing sometimes speeds up narrowing derivations drastically.

The results in [10] on (minimally collapsing) basic narrowing correct analogous claims of Krishna Rao [15] which are based on an incomplete version of term graph narrowing. The problem with the definition of narrowing in that paper—borrowed from [8]—is that it does not allow a collapsing between the application of the unifier and the rewrite step. As a consequence, narrowing is incomplete for a non-left-linear system like $\{\mathrm{f}(\mathrm{x},\mathrm{x}) \to \mathrm{a}\}$ (belonging to all three classes of rewrite systems addressed by the main results of [15]). The goal $\mathrm{f}(\mathrm{x},\mathrm{y}) =^? \mathrm{a}$, for instance, is not solvable with the kind of narrowing given there.

Narrowing on jungles, using conditional rewrite rules, is considered in [4]. Narrowing steps are based on jungle pushouts, leading to a kind of minimally collapsing narrowing. The results in [4] aim at showing the correctness of a concrete implementation of conditional narrowing.

Echahed and Janodet [5,6,14] introduce cyclic term graphs as basic data structure in functional logic languages: they consider programs as cyclic term graph rewriting systems and study the rewriting and narrowing relations they induce. They characterize a class of cyclic term graphs, so-called admissible graphs, and prove that admissible graph rewriting is confluent. They further prove that narrowing is sound and complete with respect to admissible graph rewriting, and show the optimality of certain strategies with respect to criteria depending on the graph rewriting system under consideration.

We summarize the results about the completeness of term graph narrowing strategies in the following two figures. An entry in Figure 4 means that the term graph narrowing strategy is complete whenever term graph rewriting satisfies the mentioned requirement; an entry in Figure 5 means that there exists a counterexample demonstrating that the narrowing strategy is not complete in general for term rewrite systems with the mentioned requirement.

Strategy	Requirement	Reference
arbitrary	$\Rightarrow_{\mathrm{coll}}$ normalizing & confluent	[9], Thm. 5.7
basic	$\Rightarrow_{\mathrm{coll}}$ innermost norm. & confl.	[10], Thm. 22
	$\mathcal{R}$ right-linear and $\Rightarrow_{\mathrm{coll}}$ normalizing & confluent	[10], Thm. 28
min. collapsing	$\Rightarrow_{\mathrm{coll}}$ normalizing & confluent	[10], Thm. 26
min. coll. basic	$\Rightarrow_{\mathrm{coll}}$ innermost norm. & confl.	[10], Thm. 27
	$\mathcal{R}$ right-linear and $\Rightarrow_{\mathrm{coll}}$ normalizing & confluent	[10], Thm. 28
max. collapsing	$\Rightarrow_{\mathrm{coll}}$ terminating & confluent	[10], Thm. 31
max. coll. basic	$\Rightarrow_{\mathrm{coll}}$ terminating & confluent	[10], Thm. 31

Fig. 4. Completeness results for term graph narrowing strategies

Strategy	Requirement	Reference
arbitrary	$\mathcal{R}$ normalizing & confluent	[10], Introduction
basic	$\Rightarrow_{\mathrm{coll}}$ normalizing & confluent	[10], Example 23
max. collapsing	$\mathcal{R}$ right-linear and $\Rightarrow_{\mathrm{coll}}$ normalizing & confluent	[10], Example 30
max. coll. basic	$\Rightarrow_{\mathrm{coll}}$ normalizing & confluent	[10], Example 30

Fig. 5. Counterexamples to completeness

A topic for future work is to investigate combinations of minimally and maximally collapsing narrowing with known refinements of basic (term) narrowing such as LSE narrowing [3]. By employing restrictions on rewrite rules like non-ambiguity, left-linearity etc., one may also adopt strategies like needed and lazy narrowing [1,17,16,12] and consider their completeness when combined with various sharing strategies.

References

[1] Antoy, S., R. Echahed and M. Hanus, *A needed narrowing strategy*, Journal of the ACM **47** (2000), pp. 776–822.

[2] Baader, F. and T. Nipkow, "Term Rewriting and All That," Cambridge University Press, Cambridge, 1998.

[3] Bockmayr, A., S. Krischer and A. Werner, *Narrowing strategies for arbitrary canonical rewrite systems*, Fundamenta Informaticae **24** (1995), pp. 125–155.

[4] Corradini, A. and D. Wolz, *Jungle rewriting: an abstract description of a lazy narrowing machine*, in: *Graph Transformations in Computer Science*, Lecture Notes in Computer Science **776** (1994), pp. 119–137.

[5] Echahed, R. and J.-C. Janodet, *Admissible graph rewriting and narrowing*, in: *Proc. of Joint International Conference and Symposium on Logic Programming (JICSLP'98)* (1998), pp. 325–342.

[6] Echahed, R. and J.-C. Janodet, *Completeness of admissible graph collapsing narrowing*, in: H. Ehrig and G. Taentzer, editors, *Proc. Joint APPLIGRAPH/GETGRATS Workshop on Graph Transformation Systems*, Technische Universität Berlin, Forschungsbericht **2000-2** (2000), pp. 123–132.

[7] Fay, M. J., *First-order unification in an equational theory*, in: *Proc. 4th Workshop on Automated Deduction*, Academic Press, Austin, Texas, 1979 pp. 161–167.

[8] Habel, A. and D. Plump, *Unification, rewriting, and narrowing on term graphs*, in: *Proc. Joint COMPUGRAPH/SEMAGRAPH Workshop on Graph Rewriting and Computation (SEGRAGRA 95)*, Electronic Notes in Theoretical Computer Science **2** (1995), pp. 35–42.

[9] Habel, A. and D. Plump, *Term graph narrowing*, Mathematical Structures in Computer Science **6** (1996), pp. 649–676.

[10] Habel, A. and D. Plump, *Complete strategies for term graph narrowing*, in: *Recent Trends in Algebraic Development Techniques, Selected Papers*, Lecture Notes in Computer Science **1589** (1999), pp. 152–167.

[11] Hanus, M., *The integration of functions into logic programming: From theory to practice*, The Journal of Logic Programming **19 & 20** (1994), pp. 583–628.

[12] Hanus, M., *Lazy narrowing with simplification*, Journal of Computer Languages **23** (1997), pp. 61–85.

[13] Hullot, J.-M., *Canonical forms and unification*, in: *Proc. 5th Conference on Automated Deduction*, Lecture Notes in Computer Science **87** (1980), pp. 318–334.

[14] Janodet, J.-C., "On Data Types in Functional Logic Programming Languages: Admissible Graph Rewriting and Narrowing," Ph.D. thesis, Grenoble (2000), in French.

[15] Krishna Rao, M. R., *Completeness results for basic narrowing in non-copying implementations*, in: *Proc. of the Joint International Conference and Symposium on Logic Programming (JICSP'96)* (1996), pp. 393–407.

[16] Middeldorp, A. and S. Okui, *A deterministic lazy narrowing calculus*, Journal of Symbolic Computation **25** (1999), pp. 733–757.

[17] Middeldorp, A., S. Okui and T. Ida, *Lazy narrowing: Strong completeness and eager variable elimination*, Theoretical Computer Science **167** (1996), pp. 95–130.

[18] Plump, D., *Evaluation of functional expressions by hypergraph rewriting*, Dissertation, Universität Bremen (1993).

[19] Plump, D., *Term graph rewriting*, in: *Handbook of Graph Grammars and Computing by Graph Transformation, 2: Applications, Languages and Tools*, World Scientific, 1999 pp. 3–61.

[20] Plump, D., *Essentials of Term Graph Rewriting*, 2001. In this volume.

[21] Plump, D. and A. Habel, *Graph unification and matching*, in: *Graph Grammars and Their Application to Computer Science*, Lecture Notes in Computer Science **1073** (1996), pp. 75–89.

[22] Sleep, R., R. Plasmeijer and M. v. Eekelen, editors, "Term Graph Rewriting. Theory and Practice," John Wiley, Chichester, 1993.