Complete Strategies for Term Graph Narrowing*

Annegret Habel[1] and Detlef Plump[2]**

[1] Universität Oldenburg, Fachbereich Informatik
Postfach 2503, 26111 Oldenburg, Germany
Annegret.Habel@informatik.uni-oldenburg.de

[2] Universität Bremen, Fachbereich Mathematik und Informatik
Postfach 330440, 28334 Bremen, Germany
det@informatik.uni-bremen.de

Abstract. Narrowing is a method for solving equations in the equational theories of term rewriting systems. Unification and rewriting, the central operations in narrowing, are often implemented on graph-like data structures to exploit sharing of common subexpressions. In this paper, we study the completeness of narrowing in graph-based implementations. We show that the well-known condition for the completeness of tree-based narrowing, viz. a normalizing and confluent term rewrite relation, does not suffice. Completeness is restored, however, if the implementing graph rewrite relation is normalizing and confluent. We address basic narrowing and show its completeness for innermost normalizing and confluent graph rewriting. Then we consider the combination of basic narrowing with two strategies for controlling sharing, 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. Our results on minimally collapsing basic narrowing correct analogous claims by Krishna Rao [Proc. JICSLP'96] which are based on an incomplete version of term graph narrowing.

1 Introduction

Narrowing is a method for solving equations in the equational theories defined by term rewriting systems. To ensure the completeness of this method, the term rewrite relation has to be normalizing and confluent. A huge number of narrowing strategies has been proposed to reduce the search space of narrowing while maintaining completeness (see the survey of Hanus [9]). Almost all of this work considers narrowing as a relation on terms, i.e. trees, although rewriting and unification—the central operations in narrowing—are often implemented on graph-like data structures to allow sharing of common subexpressions.

* This work was partially supported by the ESPRIT Working Group APPLIGRAPH.

** Work of this author was done while he was visiting CWI, Amsterdam, and the University of Nijmegen, supported by the HCM network EXPRESS and by Deutsche Forschungsgemeinschaft, respectively.

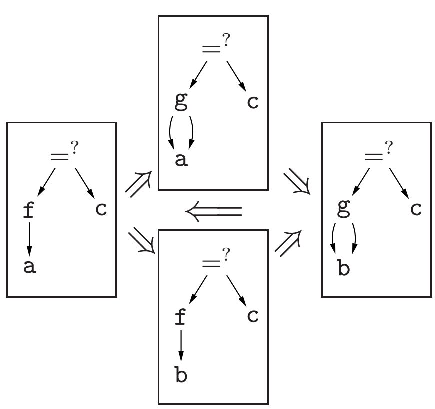
J.L. Fiadeiro (Ed.): WADT'98, LNCS 1589, pp. 152–167, 1999.

Sharing saves space and also time, as repeated evaluations of expressions are avoided. Consider, for instance, the rule $\mathtt{fib}(\mathtt{s}(\mathtt{s}(\mathtt{x}))) \to \mathtt{fib}(\mathtt{s}(\mathtt{x})) + \mathtt{fib}(\mathtt{x})$ from a specification of the Fibonacci function. Applying this rule to an expression of the form $\mathtt{fib}(\mathtt{s}(\mathtt{s}(t)))$ requires copying the subterm t in order to obtain the result $\mathtt{fib}(\mathtt{s}(t)) + \mathtt{fib}(t)$. But this copying duplicates work as the subterm t now has to be evaluated twice. The obvious solution to this problem is to use a graph-like representation of expressions in which subexpressions like t are *shared* rather than copied. Similarly, it is well-known that an efficient implementation of the unification operation requires sharing certain subexpressions (see [12,3]).

In a graph-based implementation, however, narrowing is incomplete for certain rewrite systems over which tree-based narrowing is complete. We demonstrate this by the following rules[1]:

$$\mathcal{R}\begin{cases} \mathtt{f}(\mathtt{x}) \to \mathtt{g}(\mathtt{x},\mathtt{x}) \\ \mathtt{a} \to \mathtt{b} \\ \mathtt{g}(\mathtt{a},\mathtt{b}) \to \mathtt{c} \\ \mathtt{g}(\mathtt{b},\mathtt{b}) \to \mathtt{f}(\mathtt{a}) \end{cases}$$

The term rewrite relation of this system is normalizing and confluent, a sufficient condition for the completeness of (unrestricted) tree-based narrowing [20]. But the goal $\mathtt{f}(\mathtt{a}) =^? \mathtt{c}$ cannot be solved in an implementation with sharing, although $\mathtt{f}(\mathtt{a})$ and $\mathtt{c}$ are equivalent terms. The reason is that sharing prevents a rewriting of $\mathtt{f}(\mathtt{a})$ into $\mathtt{c}$, as shown in the picture below. (Notice that there is no step corresponding to the term rewrite step $\mathtt{g}(\mathtt{a},\mathtt{a}) \to \mathtt{g}(\mathtt{a},\mathtt{b})$.) For the same reason, the graph rewrite relation of the above system is neither normalizing nor confluent.



In this paper, we provide completeness results for narrowing in graph-based implementations (where completeness means that for every solution to a goal, a solution can be computed that is at least as general). More precisely, we use the

[1] This system was used in [17] as a counterexample to confluence of term graph rewriting. Independently, Middeldorp and Hamoen invented the same system as a counterexample to completeness of basic narrowing [15].

setting of *term graph narrowing* [7] and show completeness (and incompleteness) of various strategies.

Basic term graph narrowing is shown to be complete for innermost normalizing and confluent term graph rewriting, where innermost normalization can be relaxed to normalization if all rewrite rules are right-linear. A counterexample reveals that normalization and confluence are not sufficient in general for this strategy. We then introduce minimally collapsing and maximally collapsing narrowing. While the former introduces only as much sharing as necessary, the latter creates a full sharing prior to every rewrite step. We present an example where maximally collapsing narrowing reduces an exponential number of necessary narrowing steps to a linear number. The combination of minimally collapsing and basic narrowing remains complete for innermost normalizing and confluent graph rewriting. A counterexample shows that this condition is not sufficient for maximally collapsing narrowing. Here termination and confluence ensure completeness, even under the combination with basic narrowing.

Our results on minimally collapsing basic narrowing correct analogous claims of Krishna Rao [14] which are based on an incomplete version of term graph narrowing. The problem with that version 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 $\{\mathtt{f}(\mathtt{x},\mathtt{x}) \to \mathtt{a}\}$ (belonging to all three classes of rewrite systems addressed by the main results of [14]). The goal $\mathtt{f}(\mathtt{x},\mathtt{y}) =^? \mathtt{a}$, for instance, is not solvable with such a kind of narrowing.

2 Term Graphs and Substitutions

Let Σ be a set of *function symbols*. Each function symbol f comes with a natural number $\mathrm{arity}(f) \geq 0$. Function symbols of arity 0 are called *constants*. We further assume that there is an infinite set Var of *variables* such that $\mathrm{Var} \cap \Sigma = \emptyset$. For each variable x, we set $\mathrm{arity}(x) = 0$.

A *hypergraph* over $\Sigma \cup \mathrm{Var}$ 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 $\mathrm{lab}_G\colon \mathrm{E}_G \to \Sigma \cup \mathrm{Var}$, 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 $1 + \mathrm{arity}(\mathrm{lab}_G(e))$. Given an edge e with $\mathrm{att}_G(e) = v\,v_1 \ldots v_n$, node v is the *result node* of e while $v_1, \ldots, v_n$ are the *argument nodes*. The result node is denoted by $\mathrm{res}(e)$. The set of variables occurring in G is denoted by $\mathrm{Var}(G)$, that is, $\mathrm{Var}(G) = \mathrm{lab}_G(\mathrm{E}_G) \cap \mathrm{Var}$. In the following, hypergraphs and hyperedges are simply called graphs and edges.

Definition 1 (term graph). *A graph G is a* term graph *if*

(1) there is a node root_G *from which each node is reachable,*
(2) G is acyclic, and
(3) each node is the result node of a unique edge.

The picture in the introduction shows four term graphs with function symbols $=^?$, $\mathtt{g}$, $\mathtt{f}$, $\mathtt{a}$, $\mathtt{b}$ and $\mathtt{c}$, where $=^?$ and $\mathtt{g}$ are binary, $\mathtt{f}$ is unary, and $\mathtt{a}$, $\mathtt{b}$ and $\mathtt{c}$ are constants. 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 (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* $\mathrm{att}_G(e) = v\, v_1 \ldots v_n$*.*

By Definition 1, $\mathrm{term}_G(v)$ is a well-defined term over $\Sigma \cup \mathrm{Var}$. In the following we abbreviate $\mathrm{term}_G(\mathrm{root}_G)$ by $\mathrm{term}(G)$.

A *graph morphism* $f: G \to H$ between two graphs G and H consists of two functions $f_{\mathrm{V}}: \mathrm{V}_G \to \mathrm{V}_H$ and $f_{\mathrm{E}}: \mathrm{E}_G \to \mathrm{E}_H$ satisfying $\mathrm{lab}_H \circ f_{\mathrm{E}} = \mathrm{lab}_G$ and $\mathrm{att}_H \circ f_{\mathrm{E}} = f^*_{\mathrm{V}} \circ \mathrm{att}_G$ (where $f^*_{\mathrm{V}}: \mathrm{V}^*_G \to \mathrm{V}^*_H$ maps a string $v_1 \ldots v_n$ to $f_{\mathrm{V}}(v_1) \ldots f_{\mathrm{V}}(v_n)$). The morphism f is an *isomorphism* if f_{V} and f_{E} are bijective. In this case G and H are *isomorphic*, which is denoted by $G \cong H$.

Definition 3 (collapsing). *Given two term graphs G and H, G* collapses *to H if there is a graph morphism $c: G \to H$ mapping $root_G$ to $root_H$. This is denoted by $G \succeq_c H$ or simply by $G \succeq H$. We write $G \succ_c H$ or $G \succ H$ if c is non-injective. The latter kind of collapsing is said to be* proper*. A term graph G is* fully collapsed *if there is no H with $G \succ H$.*

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 *substitution pair* x/G consists of a variable x and a term graph G. Given a term graph H and an edge e in H labelled with x, the application of x/G to e proceeds in two steps: (1) Remove e from H, yielding the graph $H - \{e\}$, and (2) construct the disjoint union $(H - \{e\}) + G$ and fuse the result node of e with $root_G$.

Definition 4 (term graph substitution). *A* term graph substitution *(or* substitution *for short) 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 $i = 1, \ldots, n$. The* domain *of α is the set $\mathrm{Dom}(\alpha) = \{x_1, \ldots, x_n\}$. The application of α to a term graph H yields the term graph $H\alpha$ which is obtained by applying all substitution pairs in α simultaneously to all edges with label in* $\mathrm{Dom}(\alpha)$*.*

We assume that the reader is familiar with substitutions on terms. Given a term graph substitution α, we denote by α^{term} the associated term substitution $\{x/\mathrm{term}(G) \mid x/G \in \alpha\}$. The restriction of a term or term graph substitution σ to a subset V of Var is denoted by $\sigma|_V$.

3 Term Graph Narrowing

In this section we briefly review term graph rewriting and narrowing. We first recall some properties of relations and basic concepts of term rewriting systems (for an introduction one may consult [2] or [5,13]). Let A be a set and $\to$ be a binary relation on A. Then $\to^*$ and $\leftrightarrow^*$ denote the transitive-reflexive and symmetric-transitive-reflexive closures of $\to$. The relation $\to$ is *confluent* if for all a, b, c with $b \leftarrow^* a \to^* c$ there is some d such that $b \to^* d \leftarrow^* c$. The relation $\to$ is *terminating* if there is no infinite sequence $a_1 \to a_2 \to a_3 \to \ldots$ An element a is a *normal form* if there is no b such that $a \to b$. The relation $\to$ is *normalizing* if for each a there is a normal form b such that $a \to^* b$.

A *term rewrite rule* $l \to r$ consists of two terms l and r 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*. The term rewrite relation associated with $\mathcal{R}$ is denoted by $\to_{\mathcal{R}}$.

For every term t, let $\vartriangle t$ be a tree representing t and $\Diamond t$ be a term graph representing t such that only variables are shared.[2] Given a graph G, we write $\underline{G}$ for the graph that results from removing all edges labelled with variables. For each node v, we denote by $G|_v$ the subgraph consisting of all nodes reachable from v and all edges having these nodes as result nodes.

Definition 5 (redex and preredex). *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* $\mathrm{red}(\mathrm{root}_{\Diamond l}) = v$. *The pair $\langle v, l \to r\rangle$ is a* preredex *if there is a term substitution σ such that* $\mathrm{term}_G(v) = l\sigma$.

While every redex is a preredex, the converse need not hold if there are repeated variables in the left-hand side l of $l \to r$. In this case a suitable collapsing of G can turn the preredex $\langle v, l \to r\rangle$ into a redex.

Definition 6 (term graph rewriting). *Let G, 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 isomorphic to the term graph G_3 constructed as follows:*

(1) $G_1 = G - \{e\}$ is the graph obtained from G by removing the unique edge e having result node v.
(2) G_2 is the graph obtained from the disjoint union $G_1 + \underline{\Diamond r}$ by
  - *identifying v with* $\mathrm{root}_{\Diamond r}$,
  - *identifying* $\mathrm{red}(v_1)$ *with v_2, for each pair $\langle v_1, v_2\rangle \in \mathrm{V}_{\Diamond l} \times \mathrm{V}_{\Diamond r}$ with* $\mathrm{term}_{\Diamond l}(v_1) = \mathrm{term}_{\Diamond r}(v_2) \in \mathrm{Var}$.

(3) $G_3 = G_2|_{root_G}$ 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 $\Rightarrow_{\mathcal{R}}$ by adding proper collapse steps: $G \Rightarrow_{\mathcal{R}} H$ if $G \succ H$ or $G \Rightarrow_{v,\, l\to r} H$ for some redex $\langle v, l \to r\rangle$. A sequence $G \cong G_1 \Rightarrow_{\mathcal{R}} G_2 \Rightarrow_{\mathcal{R}} \ldots \Rightarrow_{\mathcal{R}} G_n = H$ is a term graph rewrite derivation *and is denoted by $G \Rightarrow_{\mathcal{R}}^* H$.*

[2] In other words, $\Diamond t$ is a term graph such that there exists a surjective graph morphism $\vartriangle t \to \Diamond t$ that identifies two edges e_1 and e_2 iff $\mathrm{lab}_{\vartriangle t}(e_1) = \mathrm{lab}_{\vartriangle t}(e_2) \in \mathrm{Var}$.

The purpose of collapse steps is twofold: they are sometimes necessary to enable the application of a rule with repeated variables in its left-hand side (see the remark below Definition 5), and in certain cases they can speed up rewriting and narrowing drastically (see Example 1).

Now we turn to term graph narrowing which, like conventional term narrowing, aims at solving equations modulo the equational theory defined by a term rewriting system.

Recall that a set of terms $\{t_1, \ldots, t_n\}$ is *unifiable* if there is a term substitution σ such that $t_1\sigma = t_2\sigma = \ldots = t_n\sigma$, and that in this case there exists a *most general unifier* σ with this property.

Definition 7 (term graph narrowing). *Let G and H be term graphs, U be a set of non-variable nodes in G, $l \to r$ be a rule*[3] *in $\mathcal{R}$, and α be a term graph substitution. Then there is a* narrowing step $G \rightsquigarrow_{U,\, l\to r,\, \alpha} H$ *if α^{term} is a most general unifier of $\{\mathrm{term}_G(u) \mid u \in U\} \cup \{l\}$, and*

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

for some collapsing $G\alpha \succeq_c G'$ such that $U = c^{-1}(v)$.[4] *We denote such a step also by $G \rightsquigarrow_\alpha H$. A sequence of narrowing steps $G \cong G_1 \rightsquigarrow_{\alpha_1} G_2 \rightsquigarrow_{\alpha_2} \ldots \rightsquigarrow_{\alpha_{n-1}} G_n = H$ is a* term graph narrowing derivation *and is denoted by $G \rightsquigarrow^*_\alpha H$, where $\alpha = \alpha_1\alpha_2\ldots\alpha_{n-1}$.*[5]

The present definition of term graph narrowing extends the definition in [7] in that the latter corresponds to the special case where all nodes in the set U represent the same term.

From now on we assume that $\mathcal{R}$ contains the rule $\mathtt{x} =^? \mathtt{x} \to \mathtt{true}$, where the binary function 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 term substitution σ satisfying $s\sigma \leftrightarrow^*_{\mathcal{R}} t\sigma$.

Example 1. The system

$$\mathcal{R}\left\{\begin{array}{c}\mathtt{exp(0)} \to \mathtt{s(0)}\\ \mathtt{exp(s(x))} \to \mathtt{exp(x)} + \mathtt{exp(x)}\\ \mathtt{x} =^? \mathtt{x} \to \mathtt{true}\end{array}\right.$$

specifies the function $\mathtt{exp}\colon n \mapsto 2^n$ on natural numbers, where the result is represented as a term over $\{\mathtt{0}, \mathtt{s}, +\}$. Figure 1 demonstrates that goals of the form

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

can be solved in $n+2$ steps if narrowing steps merge two occurrences of an exp-call into one call. In contrast, tree-based narrowing (and also the "minimally

3 We assume that this rule has no common variables with G. If this is not the case, then the variables in $l \to r$ are renamed into variables from $\mathrm{Var} - \mathrm{Var}(G)$.

4 Given a graph morphism $f\colon G \to H$ and a node v in H, $f^{-1}(v)$ denotes the set $\{\overline{v} \in V_G \mid f(\overline{v}) = v\}$.

5 The composition of term graph substitutions is defined analogously to the term case.

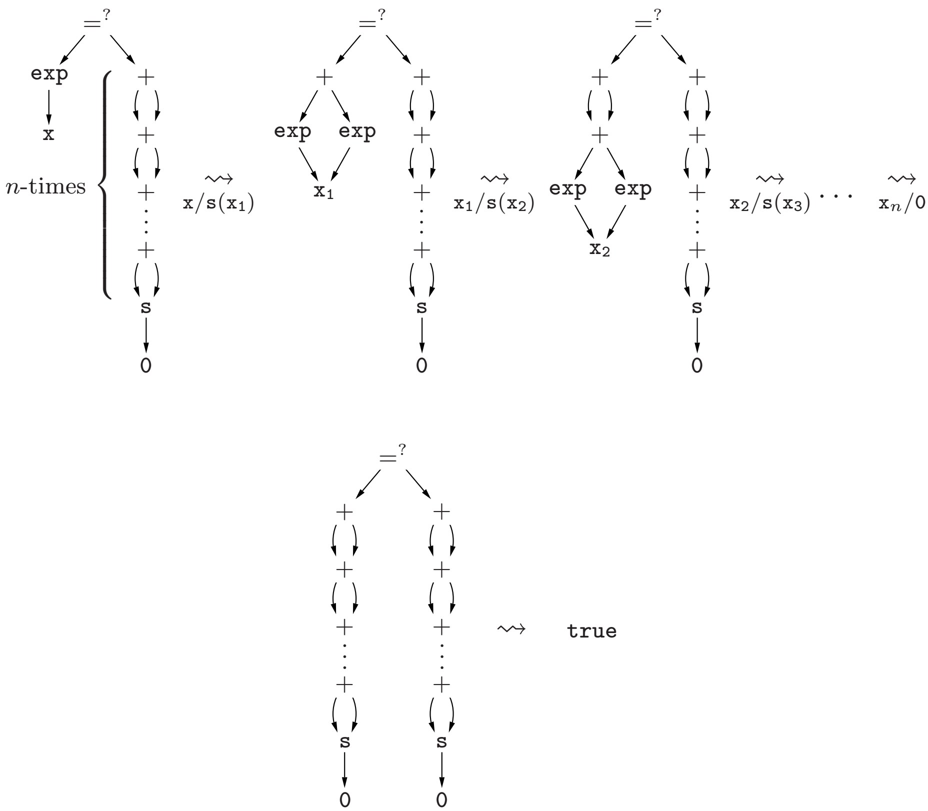


Fig. 1. A term graph narrowing derivation

collapsing narrowing" of Section 6) needs a number of steps exponential in n to solve these goals.

In [7] it is shown that term graph narrowing is sound for arbitrary term rewriting systems and complete for systems $\mathcal{R}$ with a normalizing and confluent graph rewrite relation $\Rightarrow_{\mathcal{R}}$.

Theorem 1 (soundness and completeness of narrowing). *Let G be a term graph such that* $\mathrm{term}(G)$ *is a goal* $s =^? t$.

1. *If* $G \rightsquigarrow_{\alpha}^{*} \triangle\mathtt{true}$, *then* α^{term} *is a solution of* $s =^? t$.
2. *If* $\Rightarrow_{\mathcal{R}}$ *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* $\beta^{\mathrm{term}} \leq_{\mathcal{R}} \sigma\ [\mathrm{Var}(G)]$.[6]

[6] Given term substitutions σ and τ, and $V \subseteq Var$, we write $\sigma =_{\mathcal{R}} \tau\ [V]$ if $x\sigma \leftrightarrow_{\mathcal{R}}^{*} x\tau$ for each $x \in V$, and $\sigma \leq_{\mathcal{R}} \tau\ [V]$ if there is a substitution ρ such that $\sigma\rho =_{\mathcal{R}} \tau\ [V]$.

Sufficient conditions for confluence and termination of $\Rightarrow_{\mathcal{R}}$ can be found in [18]. We just mention that if $\Rightarrow_{\mathcal{R}}$ is normalizing, then $\Rightarrow_{\mathcal{R}}$ is confluent if and only if $\to_{\mathcal{R}}$ is confluent.

4 The Lifting Lemma

Now we show how to transform a term graph rewrite derivation $G \Rightarrow^*_{\mathcal{R}} H$, where $G = G'\alpha$ for some substitution α, into a narrowing derivation $G' \rightsquigarrow^*_{\beta} H'$ such that $H'\gamma \succeq H$ for some substitution γ. This Lifting Lemma is essential for proving the completeness of the different narrowing strategies considered in the following sections.

In [7] completeness is proved by a two-stage lifting of rewrite derivations: derivations are transformed into "minimally collapsing" rewrite derivations which in turn can be lifted to minimally collapsing narrowing derivations (being often longer than the given rewrite derivations). This lifting mechanism cannot be used to show, for example, the completeness of maximally collapsing narrowing. Therefore we establish a new Lifting Lemma which lifts rewrite derivations directly to narrowing derivations and, consequently, is more flexible to use.

Before stating the Lifting Lemma at the end of this section, we provide the lemmata used in its proof. A key construction will be the suitable lifting of collapsings "over" substitutions which is described in Lemma 3.

Lemma 1 (factorization of substitutions). *Let $G'\alpha = G$ for some normalized term graph substitution α,*[7] *and U be a set of nodes in G'. Suppose that there is a term l and a term substitution σ such that $term_G(u) = l\sigma$ for all $u \in U$. Moreover, let V be a finite subset of* Var *such that* $\mathrm{Var}(G') \cup \mathrm{Dom}(\alpha) \subseteq V$. *Then there exist term graph substitutions β and γ such that (1) β^{term} is a most general unifier of $\{term_{G'}(u) \mid u \in U\} \cup \{l\}$, (2) $(\beta\gamma)|_V = \alpha$, and (3) γ is normalized.*

Proof. Similar to the proof of the corresponding lemma in [7]. □

Lemma 2 (induced collapsing). *For every collapsing $G \succeq_c H$ and substitution α there is a collapsing $G\alpha \succeq_{\widehat{c}} H\alpha$.*

The collapsing $\widehat{c}$ of Lemma 2 is said to be *induced* by c and α. Below we lift a collapsing $G'\gamma \succeq H$ over the substitution γ to a collapsing $G' \succeq H'$ such that $H'\gamma \succeq H$. The collapsing $G' \succeq H'$ is constructed in such a way that it "approximates" $G'\gamma \succeq H$ as far as possible.

Lemma 3 (collapse lifting). *Let $G \succeq_c H$ be a collapsing and G' be a term graph such that $G'\alpha = G$ for some substitution α. Then there is a collapsing $G' \succeq_d H'$ such that for all nodes v and w in G', $d(v) = d(w)$ if and only if $c(v) = c(w)$ and* $\mathrm{term}_{G'}(v) = \mathrm{term}_{G'}(w)$. *Moreover, there is a collapsing $H'\alpha \succeq H$.*

[7] A term graph substitution $\alpha = \{x_1/G_1, \ldots, x_n/G_n\}$ is *normalized* if $G_1, \ldots, G_n$ are normal forms with respect to $\Rightarrow_{\mathcal{R}}$.

Proof. Let $\sim$ be the equivalence relation on $\mathrm{V}_{G'}$ defined by: $v \sim w$ if $c(v) = c(w)$ and $\mathrm{term}_{G'}(v) = \mathrm{term}_{G'}(w)$. Let $\mathrm{V}_{H'}$ be the set of equivalence classes of $\sim$, and let $d_{\mathrm{V}}\colon \mathrm{V}_{G'} \to \mathrm{V}_{H'}$ map each node to its equivalence class. An analogous construction yields the edge set $\mathrm{E}_{H'}$ and the mapping $d_{\mathrm{E}}\colon \mathrm{E}_{G'} \to \mathrm{E}_{H'}$, where two edges in G' are equivalent if there source nodes are. Given an edge equivalence class $[e]$, define $\mathrm{lab}_{H'}([e]) = \mathrm{lab}_{G'}(e)$ and $\mathrm{att}_{H'}([e]) = [v][v_1]\ldots[v_n]$ if $\mathrm{att}_{G'}(e) = v\,v_1\ldots v_n$ It is easy to see that H' is well-defined and that $d = \langle d_{\mathrm{V}}, d_{\mathrm{E}}\rangle$ is a graph morphism. Now the collapsing $H'\alpha \succeq_e H$ is defined by considering the collapsing $G'\alpha \succeq_{\widehat{d}} H'\alpha$ induced by d and α. Given any node v in H', select some $u \in \widehat{d}^{-1}(v)$ and define $e(v) = c(u)$. It is not difficult to verify that e is well-defined.

If a collapsing is followed by a rewrite step such that the redex node has only one preimage, the collapsing can be "shifted" behind the rewrite step.

Lemma 4 (collapse shifting [7]). *Let $G \succeq_c G' \Rightarrow_{v,\,l\to r} H$ be a collapsing followed by a proper rewrite step such that $c^{-1}(v)$ contains a single node $\overline{v}$. If $\langle \overline{v},\, l \to r\rangle$ is a redex, then there is a term graph H' such that $G \Rightarrow_{\overline{v},\,l\to r} H' \succeq H$.*

Now we are ready to show that a collapsing together with a subsequent rewrite step can be lifted to a narrowing step.

Lemma 5. *Let $G \succeq M \Rightarrow_{v,\,l\to r} H$ be a rewrite derivation with one proper rewrite step, and G' be a term graph such that $G'\alpha = G$ for some normalized substitution α. Moreover, let $V \subseteq \mathrm{Var}$ be a finite set of variables such that $\mathrm{Var}(G') \cup \mathrm{Dom}(\alpha) \subseteq V$. Then there is a narrowing step $G' \rightsquigarrow_{U,\,l\to r,\,\beta} H'$ and a normalized substitution γ such that $H'\gamma \succeq H$ and $(\beta\gamma)|_V = \alpha$.*

Proof. Without loss of generality, we may assume $\mathrm{Var}(l) \cap V = \emptyset$ (otherwise the variables of l are renamed). Let $U = c^{-1}(v)$. Since $\langle v,\, l \to r\rangle$ is a redex, there is a term substitution σ such that $term_G(u) = term_M(v) = l\sigma$ for each u in U. Since α is normalized, all nodes in U are non-variable nodes in G'. By Lemma 1, there exist term graph substitutions β and γ such that (1) β^{term} is a most general unifier of $\{term_{G'}(u) \mid u \in U\} \cup \{l\}$, (2) $(\beta\gamma)|_V = \alpha$, and (3) γ is normalized. Let $G'\beta \succeq_d M'$ be the lifting of the collapsing $G \succeq M$ over γ, as specified in Lemma 3. By definition of d, there is a node $\overline{v}$ in M' such that $U = d^{-1}(\overline{v})$. Moreover, for each $u \in U$ and each (repeated) variable x in l, d identifies the nodes in $G'\beta|_u$ corresponding to the occurrences of x in l. This is because c identifies these nodes, too (otherwise $\langle v,\, l \to r\rangle$ were not a redex). Hence $\langle \overline{v},\, l \to r\rangle$ is a redex in M'.

Now, by Lemma 3, there is a collapsing $M'\gamma \succeq_e M$. The construction of e in the proof of Lemma 3 and the fact that c identifies all nodes in U imply $e^{-1}(v) = \{\overline{v}\}$. Hence, by Lemma 4, there is a term graph $\overline{H}$ such that $M'\gamma \Rightarrow_{\overline{v},\,l\to r} \overline{H} \succeq H$. Since $\langle \overline{v},\, l \to r\rangle$ is a redex in M', there is a restricted rewrite step $M' \Rightarrow_{\overline{v},\,l\to r} H'$ such that $H'\gamma \succeq H$. This completes the proof of the lemma. □

Lemma 6 (Lifting Lemma). *Let $G \Rightarrow_{\mathcal{R}}^* H$ be a rewrite derivation and G' be a term graph such that $G'\alpha = G$ for some normalized substitution α. Moreover, let V be a finite subset of* Var *such that* $\mathrm{Var}(G') \cup \mathrm{Dom}(\alpha) \subseteq V$. *Then there is a narrowing derivation $G' \rightsquigarrow_{\beta}^* H'$ and a normalized substitution γ such that $H'\gamma \succeq H$ and $(\beta\gamma)|_V = \alpha$.*

Proof. By induction on the number of proper rewrite steps in $G \Rightarrow_{\mathcal{R}}^* H$, using Lemma 5. □

5 Basic Term Graph Narrowing

In order to reduce the search space of narrowing, Hullot [11] introduced *basic narrowing* as a restricted form of narrowing and showed its completeness over terminating and confluent term rewriting systems. In this section, we define basic term graph narrowing and show that it is complete whenever term graph rewriting is innermost normalizing and confluent. We also present a counterexample showing that innermost normalization cannot be relaxed to normalization.

Given two nodes v and v' in a term graph G, we write $v \geq_G v'$ if v' is reachable from v, and by $v >_G v'$ we mean $v \geq_G v'$ and $v \neq v'$.

Definition 8 (basic narrowing derivation). *Let $G_1 \rightsquigarrow_{\alpha_1} \ldots \rightsquigarrow_{\alpha_{n-1}} G_n$ be a narrowing derivation where each step has the form $G_i \mapsto G_i\alpha_i \succeq_{c_i} G_i' \Rightarrow_{v_i,\, l_i \to r_i} G_{i+1}$. Define sets of nodes $A_1, \ldots, A_n$ as follows: (i) $A_1 = \mathrm{Nonvar}(G_1)$*[8] *and (ii) for $1 \leq i < n$, $A_{i+1} = \mathrm{track}(A_i^{*\prime}) + \mathrm{Nonvar}(\Diamond r_i)$*[9] *where $A_i^* = \{v \in c_i(A_i) \mid c_i^{-1}(v) \subseteq A_i\}$ and $A_i^{*\prime} = A_i^* - \{v \mid v_i \geq_{G_i'} v\}$. Nodes in A_i are* basic *nodes and nodes in $\mathrm{V}_{G_i} - A_i$ are* non-basic *for $1 \leq i \leq n$. We say that the above narrowing derivation is* basic *if $c_i^{-1}(v_i) \subseteq A_i$ for $1 \leq i < n$.*

Definition 9 (basic rewrite derivation). *Let $G_1 \Rightarrow_{\mathcal{R}} \ldots \Rightarrow_{\mathcal{R}} G_n$ be a rewrite derivation and $B \subseteq \mathrm{Nonvar}(G_1)$. Define sets of nodes $B_1, \ldots, B_n$ as follows: (i) $B_1 = B$ and (ii) for $1 \leq i < n$, $B_{i+1} = \mathrm{track}(B_i') + \mathrm{Nonvar}(\Diamond r_i)$, where $B_i' = B_i - \{v \mid v_i \geq_{G_i} v\}$, if $G_i \Rightarrow_{v_i,\, l_i \to r_i} G_{i+1}$, and $B_{i+1} = \{v \in c_i(B_i) \mid c_i^{-1}(v) \subseteq B_i\}$ if $G_i \succ_{c_i} G_{i+1}$. The above rewrite derivation is* basic *with respect to B if $v_i \in B_i$ for each proper rewrite step $G_i \Rightarrow_{v_i,\, l_i \to r_i} G_{i+1}$.*

While original narrowing considers any non-variable nodes in the goal, basic narrowing discards all nodes that have been introduced by the substitution of a previous narrowing step.

Definition 10. (innermost derivation) *A proper rewrite step $G \Rightarrow_{v,\, l \to r} H$ is an* innermost *step if $\mathrm{term}_G(v')$ is a normal form for all nodes v' with $v >_G v'$. A rewrite derivation $G \Rightarrow_{\mathcal{R}}^* H$ is an* innermost derivation *if all its proper rewrite*

[8] For a term graph G, $\mathrm{Nonvar}(G)$ is the set of non-variable nodes in G.

[9] For every rewrite step $G' \Rightarrow H$ there is a partial function track: $\mathrm{V}_{G'} \to \mathrm{V}_H$ which sends each node in G' to its descendant in H (see [7] for a formal definition).

steps are innermost steps. The term graph rewrite relation $\Rightarrow_{\mathcal{R}}$ *is* innermost normalizing *if every term graph has a normal form which can be reached by an innermost derivation.*

Lemma 7 (innermost derivations are basic). *Let G' be a term graph and α a normalized substitution. Every innermost term graph rewrite derivation starting from $G'\alpha$ is basic with respect to* $\mathrm{Nonvar}(G')$.

Lemma 8 (lifting of basic derivations). *Let α be a normalized substitution. Lemma 6 transforms every term graph rewrite derivation $G'\alpha \Rightarrow^*_{\mathcal{R}} H$ that is basic with respect to* $\mathrm{Nonvar}(G')$ *into a basic narrowing derivation $G' \rightsquigarrow^*_{\beta} H'$.*

The following property is proved in [7].

Lemma 9 (from solutions to term graph derivations). *Let $\Rightarrow_{\mathcal{R}}$ be normalizing and confluent, and G be a term graph such that* $\mathrm{term}(G)$ *is a goal $s =^? t$. Then for every solution σ of $s =^? t$, there exists a normalized term graph substitution α such that $G\alpha \Rightarrow^*_{\mathcal{R}} \triangle\mathtt{true}$ and $\alpha^{\mathrm{term}} =_{\mathcal{R}} \sigma\ [\mathrm{Var}(G)]$.*

Combining Lemmata 9, 7 and 8, we obtain the completeness of basic term graph narrowing.

Theorem 2 (completeness of basic term graph narrowing). *Basic term graph narrowing is complete*[10] *whenever term graph rewriting is innermost normalizing and confluent.*

The following counterexample demonstrates that "innermost normalizing" cannot be relaxed to "normalizing" (contrary to our claim in [8]).

Example 2. The following term rewriting system results from adding the rule $\mathtt{g(a,x)} \to \mathtt{g(a,x)}$ (and $\mathtt{x} =^? \mathtt{x} \to \mathtt{true}$) to the system shown in the introduction:

$$\mathcal{R}\begin{cases} \quad\mathtt{f(x)} \to \mathtt{g(x,x)} \\ \qquad\ \mathtt{a} \to \mathtt{b} \\ \mathtt{g(a,b)} \to \mathtt{c} \\ \mathtt{g(b,b)} \to \mathtt{f(a)} \\ \mathtt{g(a,x)} \to \mathtt{g(a,x)} \\ \mathtt{x} =^? \mathtt{x} \to \mathtt{true} \end{cases}$$

The new rule allows to copy a shared constant $\mathtt{a}$. Now every term graph has a unique normal form, which can be shown by induction on the size of term graphs. Hence $\Rightarrow_{\mathcal{R}}$ is normalizing and confluent. However, the goal $\mathtt{f(a)} =^? \mathtt{c}$ cannot be solved by basic term graph narrowing. This can be seen from Figure 2, where we underline function symbols at non-basic positions. Note that in every application of $\mathtt{g(a,x)} \to \mathtt{g(a,x)}$, the second argument of $\mathtt{g}$ becomes non-basic.

[10] If not stated otherwise, completeness is always understood in the sense of the second part of Theorem 1.

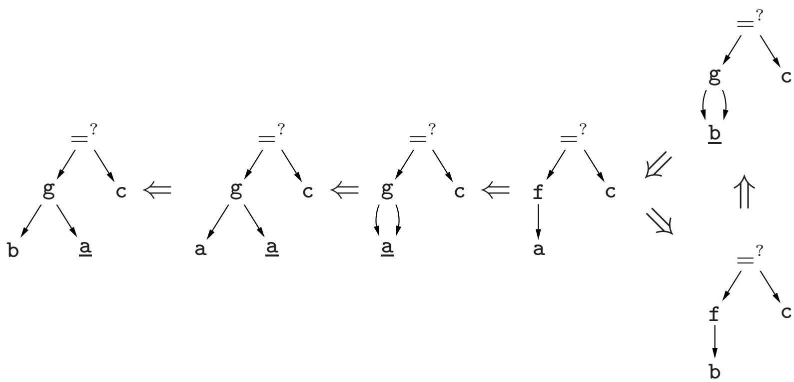


Fig. 2. Incompleteness of basic term graph narrowing

6 Minimally Collapsing Narrowing

Term graph narrowing steps as introduced in Definition 7 are non-deterministic not only with respect to the choice of the node set U and the rewrite rule $l \to r$, but also with respect to the collapsing $G\alpha \succeq_c G'$. In this and the next section we consider the two extreme choices for this collapsing, obtaining *minimally collapsing* and *maximally collapsing* narrowing.

Minimally collapsing rewrite derivations contain only collapse steps that are necessary to apply non-left-linear rewrite rules. That is, each collapse step in such a derivation turns a preredex of a non-left-linear rule into a redex.

Definition 11 (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 $G \succeq M' \succ_c M$ and each preimage $v' \in c^{-1}(v)$ of v in M', the preredex $\langle v', l \to r\rangle$ is not a redex.*

In particular, if G and M are isomorphic, 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 non-left-linear and cannot be applied at any preimage of v in G.

Definition 12 (minimally collapsing derivation). *A rewrite derivation is* minimally collapsing *if each collapse step $G \succ M$ in the derivation is followed by a proper rewrite step $M \Rightarrow_{v,\,l\to r} N$ such that $G \succ M$ is minimal with respect to $\langle v, l \to r\rangle$. A narrowing derivation is minimally collapsing if for each narrowing step $G \mapsto G\alpha \succeq_c G' \Rightarrow_{v,\,l\to r} H$ in the derivation, the collapsing $G\alpha \succeq_c G'$ is minimal with respect to $\langle v, l \to r\rangle$.*

Note that in a minimally collapsing narrowing step $G \rightsquigarrow_{U,\,l\to r,\,\alpha} H$, the set U must be a singleton due to the definition of a minimal collapsing.

Theorem 3 (completeness of minimally collapsing narrowing). *Minimally collapsing narrowing is complete whenever term graph rewriting is normalizing and confluent.*

Proof. By Theorem 4.8 of [7], for every rewrite derivation $G \Rightarrow^*_{\mathcal{R}} H$ there is a minimally collapsing derivation $G \Rightarrow^*_{\mathcal{R}} H'$ such that $H' \succeq H$. By the construction of lifted collapsings in Lemma 3 and the proof of Lemma 5, every minimally collapsing rewrite derivation is lifted by Lemma 6 to a minimally collapsing narrowing derivation. Combining this with Lemma 9 yields the desired result. □

We now consider the combination of minimally collapsing and basic narrowing. By Example 2 we already know that this strategy is not complete in general for normalizing and confluent term graph rewriting. To ensure completeness, normalization has to be strengthened to innermost normalization or the given rewrite system has to be right-linear.

Theorem 4 (completeness of minimally collapsing basic narrowing). *Minimally collapsing basic narrowing is complete whenever term graph rewriting is innermost normalizing and confluent.*

Proof. It is not difficult to check that Theorem 4.8 of [7] transforms every innermost rewrite derivation into a minimally collapsing innermost derivation. Hence, the proposition is obtained by combining Lemmata 9, 7 and 8, and by using the fact that lifting transforms minimally collapsing rewrite derivations into minimally collapsing narrowing derivations. □

Innermost normalization can be relaxed to normalization if the given rewrite system is right-linear. This can be proved by using the corresponding result for term narrowing [15], exploiting the fact that for right-linear systems, every (basic) narrowing derivation on terms can be simulated by a minimally collapsing (basic) narrowing derivation on term graphs.

Theorem 5 (completeness for right-linear systems). *Minimally collapsing basic narrowing is complete for right-linear term rewriting systems over which term graph rewriting is normalizing and confluent.*

Here we mean by "complete" that for every goal $s =^? t$ and every solution σ of this goal, there exists a narrowing derivation $\triangle(s =^? t) \rightsquigarrow^*_\beta \triangle\texttt{true}$ such that $\beta^{\text{term}} \leq_{\mathcal{R}} \sigma\ [\text{Var}(\triangle(s =^? t))]$. That is, we consider only narrowing derivations starting from the tree representation of the goal.

7 Maximally Collapsing Narrowing

We now consider maximally collapsing narrowing, that is, narrowing derivations in which all involved collapse steps are maximal. While minimally collapsing narrowing is complete when term graph rewriting is normalizing and confluent, we show by a counterexample that this is not the case for maximally collapsing narrowing. Completeness can be restored, however, by strengthening normalization to termination.

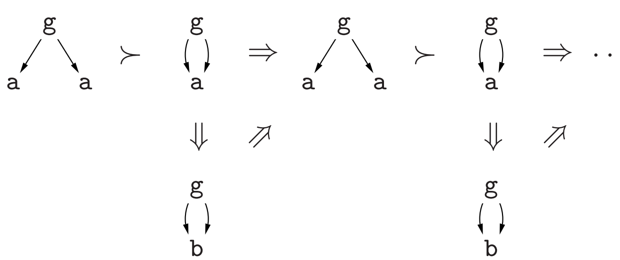
Definition 13. (maximally collapsing derivations) *A rewrite derivation is* maximally collapsing *if all its collapse steps yield fully collapsed term graphs and if all proper rewrite steps start from fully collapsed term graphs. A narrowing derivation is maximally collapsing if for each narrowing step $G \mapsto G\alpha \succeq_c G' \Rightarrow_{v,\,l\to r} H$ in the derivation, G' is fully collapsed.*

In Example 1, for instance, the goal is solved by a maximally collapsing narrowing derivation. The following counterexample demonstrates that maximally collapsing narrowing, even when not restricted to basic narrowing, is not complete in general for normalizing and confluent term graph rewriting.

Example 3. Consider the following term rewriting system:

$$\mathcal{R}\begin{cases} \mathtt{g}(\mathtt{x},\mathtt{y}) \to \mathtt{g}(\mathtt{a},\mathtt{a}) \\ \qquad\ \mathtt{a} \to \mathtt{b} \\ \mathtt{g}(\mathtt{a},\mathtt{b}) \to \mathtt{b} \\ \mathtt{x} =^? \mathtt{x} \to \mathtt{true} \end{cases}$$

Here $\Rightarrow_{\mathcal{R}}$ is normalizing and confluent since every term graph representing a term of the form $\mathtt{g}(\ldots)$ can be reduced to $\triangle\mathtt{b}$. However, the tree $\triangle\mathtt{g}(\mathtt{a},\mathtt{a})$ representing $\mathtt{g}(\mathtt{a},\mathtt{a})$ cannot be reduced to this normal form by a maximally collapsing derivation: the picture below shows all maximally collapsing derivations starting from $\triangle\mathtt{g}(\mathtt{a},\mathtt{a})$. As a consequence, maximally collapsing term graph narrowing cannot solve the goal $\mathtt{g}(\mathtt{a},\mathtt{a}) =^? \mathtt{b}$, although $\mathtt{g}(\mathtt{a},\mathtt{a}) \leftrightarrow^*_{\mathcal{R}} \mathtt{b}$.



Now we show that maximally collapsing narrowing—even in combination with basic narrowing—becomes complete when normalization is strengthened to termination.

Theorem 6 (completeness of maximally collapsing basic narrowing). *Maximally collapsing basic narrowing is complete whenever term graph rewriting is terminating and confluent.*

Proof. If $\Rightarrow_{\mathcal{R}}$ is terminating and confluent, then every term graph can be reduced to its normal form by applying as long as possible in an alternating way maximal collapse steps and (arbitrary) rewrite rules. Combining this with Lemmata 9, 7, 8 and the observation that the Lifting Lemma transforms maximally collapsing rewrite derivations into maximally collapsing narrowing derivations, we obtain the desired result. □

As a corollary of Theorem 6, maximally collapsing basic narrowing is complete for all terminating and confluent term rewriting systems. For, the latter are properly included in the class $\{\mathcal{R} \mid \Rightarrow_{\mathcal{R}} \text{ is terminating and confluent}\}$ (see [17,18]).

8 Conclusion

We summarize our results about the completeness of term graph narrowing strategies in the following table. Here the normalization and confluence properties refer to term graph rewriting, a "yes" indicates completeness and "no" means that there exists a counterexample.

	normalizing & confluent	rightlinear, normalizing & confluent	innermost normalizing & confluent	terminating & confluent
basic	no	yes	yes	yes
min. coll.	yes	yes	yes	yes
min. coll. basic	no	yes	yes	yes
max. coll.	no	no	?	yes
max. coll. basic	no	no	?	yes

We conjecture that innermost normalization and confluence are sufficient to make maximally collapsing basic narrowing complete.

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 [4,19].

By employing stronger restrictions on rewrite rules, like non-ambiguity, left-linearity etc., one may also adopt strategies like needed and lazy narrowing [1,16,10] and consider their completeness when combined with various sharing strategies. A first step into this direction is done in [6], where term graph narrowing over orthogonal constructor systems is considered.

References

1. Sergio Antoy, Rachid Echahed, and Michael Hanus. A needed narrowing strategy. In *Proc. Symposium on Principles of Programming Languages*, pages 268–279. ACM Press, 1994. 166
2. Franz Baader and Tobias Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998. 156
3. Franz Baader and Jörg H. Siekmann. Unification theory. In D.M. Gabbay, C.J. Hogger, and J.A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, pages 41–125. Clarendon Press, 1994. 153
4. Alexander Bockmayr, Stefan Krischer, and Andreas Werner. Narrowing strategies for arbitrary canonical rewrite systems. *Fundamenta Informaticae*, 24:125–155, 1995. 166

5. Nachum Dershowitz and Jean-Pierre Jouannaud. Rewrite systems. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science,* volume B, chapter 6, pages 244–320. Elsevier, 1990. 156
6. Rachid Echahed and Jean-Christophe Janodet. Admissible graph rewriting and narrowing. In *Proc. Joint International Conference and Symposium on Logic Programming 1998*. MIT Press, 1998. 166
7. Annegret Habel and Detlef Plump. Term graph narrowing. *Mathematical Structures in Computer Science*, 6:649–676, 1996. 154, 157, 158, 159, 159, 160, 161, 162, 164, 164
8. Annegret Habel and Detlef Plump. Completeness of narrowing in non-copying implementations. Informatik-Bericht 19/97, Universität Hildesheim, Institut für Informatik, 1997. 162
9. Michael Hanus. The integration of functions into logic programming: From theory to practice. *The Journal of Logic Programming*, 19 & 20:583–628, 1994. 152
10. Michael Hanus. Lazy narrowing with simplification. *Journal of Computer Languages*, 23(2–4):61–85, 1997. 166
11. Jean-Marie Hullot. Canonical forms and unification. In *Proc. 5th International Conference on Automated Deduction*, volume 87 of *Lecture Notes in Computer Science*, pages 318–334. Springer-Verlag, 1980. 161
12. Jean-Pierre Jouannaud and Claude Kirchner. Solving equations in abstract algebras: A rule-based survey of unification. In Jean-Louis Lassez and Gordon Plotkin, editors, *Computational Logic: Essays in Honor of Alan Robinson*, pages 257–321. MIT Press, 1991. 153
13. Jan Willem Klop. Term rewriting systems. In S. Abramsky, Dov M. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science,* volume 2, pages 1–116. Oxford University Press, 1992. 156
14. Madala R.K. Krishna Rao. Completeness results for basic narrowing in non-copying implementations. In *Proc. Int. Joint Conference and Symposium on Logic Programming*. MIT Press, 1996. 154, 154
15. Aart Middeldorp and Erik Hamoen. Completeness results for basic narrowing. *Applicable Algebra in Engineering, Communication and Computing*, 5:213–253, 1994. 153, 164
16. Aart Middeldorp, Satoshi Okui, and Tetsuo Ida. Lazy narrowing: Strong completeness and eager variable elimination. *Theoretical Computer Science*, 167(1,2):95–130, 1996. 166
17. Detlef Plump. Collapsed tree rewriting: Completeness, confluence, and modularity. In *Proc. Conditional Term Rewriting Systems*, volume 656 of *Lecture Notes in Computer Science*, pages 97–112. Springer-Verlag, 1993. 153, 166
18. Detlef Plump. Term graph rewriting. In H. Ehrig, G. Engels, H.-J. Kreowski, and G. Rozenberg, editors, *Handbook of Graph Grammars and Computing by Graph Transformation*, volume 2. World Scientific, 1998. To appear. 159, 166
19. Andreas Werner. Normalizing narrowing for weakly terminating and confluent systems. In *Proc. Principles and Practice of Constraint Programming*, volume 976 of *Lecture Notes in Computer Science*, pages 415–430. Springer-Verlag, 1995. 166
20. Akihiro Yamamoto. Completeness of extended unification based on basic narrowing. In *Proc. Logic Programming '88*, volume 383 of *Lecture Notes in Artificial Intelligence*, pages 1–10. Springer-Verlag, 1989. 153