Modular Termination of Basic Narrowing and Equational Unification

María Alpuente   Santiago Escobar   José Iborra

Abstract

Basic narrowing is a restricted form of narrowing which constrains narrowing steps to a set of unblocked (or basic) positions. In this work, we study the modularity of termination of basic narrowing in hierarchical combinations of TRSs, which provides new algorithmic criteria to prove termination of basic narrowing. Basic narrowing has a number of important applications including equational unification in canonical theories. Another application is analyzing termination of narrowing by checking the termination of basic narrowing, as done in pioneering work by Hullot. As a particularly interesting application, we consider solving equations modulo a theory that is given by a TRS, and then distill a number of modularity results for the decidability of equational unification via the modularity of basic narrowing (completeness and) termination.

1   Introduction

Narrowing [18] is a generalization of term rewriting that allows free variables in terms (as in logic programming) and replaces pattern matching by syntactic unification. Narrowing has received much attention due to the many important applications, such as automated proofs of termination [7], execution of functional–logic programming languages [27, 21], symbolic reachability [28], verification of cryptographic protocols [15, 16], equational unification [24], equational constraint solving [4, 5], partial evaluation [6], and model checking [17], among others. Termination of narrowing is of great interest to these applications.

Narrowing was originally introduced as a mechanism for solving equational unification problems [24]; hence, termination results for narrowing have been traditionally achieved as a by–product of addressing the decidability of equational unification. Basic narrowing [24] is a refinement of narrowing that restricts narrowing steps to a set of unblocked (or basic) positions and is still complete for equational unification in canonical TRSs, i.e., terminating and confluent TRSs. Termination of basic narrowing was first studied by Hullot in [24], where an erroneous termination result for narrowing was enunciated, namely the termination of all narrowing derivations in canonical theories when all basic narrowing derivations issuing from the right–hand sides (rhs's) of the

rules terminate. This result was implicitly corrected in [25], downgrading it to the more limited result of basic narrowing termination (instead of ordinary narrowing) whenever basic narrowing terminates for the rhs's of the rules. In [3], we proved that the missing condition to recover narrowing termination in [24] is to require that the TRS satisfies Réty's maximal commutation condition [38] for narrowing sequences, and we explicitly dropped the superfluous requirement of canonicity from the termination result of [24], as some *cognoscenti* tacitly do. By using this repaired result, we also distilled in [3] a syntactic characterization of TRSs where termination of basic narrowing implies termination of narrowing, namely *right-linear* TRSs that are either *left-linear* or *regular* and where narrowing computes only *normalized substitutions*. This includes some popular classes of TRSs, which also includes linear constructor systems.

The original motivation for our work was to prove the termination of narrowing via the termination of basic narrowing, modularly, for non-trivial subclasses of the TRSs stated above. In this paper, we ascertain several criteria for modular termination of basic narrowing in hierarchical combinations of TRSs, including generalized proper extensions with a shared subsystem and a novel generalization of this class called *(generalized) relaxed proper extensions*. We assume a standard notion of *modularity*, where a property φ of a TRS is called *modular* if, whenever $\mathcal{R}_1$ and $\mathcal{R}_2$ satisfy φ, then their combination $\mathcal{R}_1 \cup \mathcal{R}_2$ also satisfies φ. Our modularity results for basic narrowing rely on a *novel* commutation result for basic narrowing sequences, which we have not identified in the previous related literature.

Relying on the above results for the modular termination of basic narrowing, the aim of this article is to provide new classes of TRSs where equational unification is decidable. As far as we know, [35] proves the only previous modularity result for the decidability of equational unification (via termination of narrowing) in canonical TRSs. However, this result does not imply the modularity of narrowing termination for a particular class of TRSs but rather the possibility to define a terminating, modular narrowing procedure. The result in [35] is as follows: *given a canonical TRS $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$ such that narrowing terminates for $\mathcal{R}_1$ and $\mathcal{R}_2$ and $\mathcal{R}{\downarrow} \subseteq \mathcal{R}_1{\downarrow}\mathcal{R}_2{\downarrow}$ (i.e. the normalization with $\mathcal{R}$ can be obtained by first normalizing with $\mathcal{R}_1$ followed by a normalization with $\mathcal{R}_2$), there is a terminating and complete, modular narrowing strategy for $\mathcal{R}$.* Any complete strategy can be used within the modular procedure given in [35], including the basic narrowing strategy.

Structure of the paper

After some preliminaries in Section 2, Section 3 introduces basic narrowing, analyzes its termination properties, and provides a novel commutation result for basic narrowing derivations. Section 4 recalls some standard notions from the modularity of rewriting, and then studies the modularity of basic narrowing termination in the standard classes of modular combinations of TRSs. We demonstrate that termination of basic narrowing is highly modular in some of these classes, including *disjoint unions*, *constructor–sharing unions* and the

unions of *composable systems*. The restriction to basic narrowing is of key importance for these results, which do not carry out to unrestricted narrowing. Modularity of basic narrowing termination in hierarchical combinations of TRSs is studied in Section 5. Our most general result for the modularity of basic narrowing termination is proved for the novel class of *(generalized) relaxed proper extensions*. Section 6 discusses the combination of TRSs where basic narrowing terminates and is complete, and also ascertains the conditions guaranteeing the decidability of equational unification via basic narrowing in the combination of the considered TRSs. We conclude in Section 7.

This article is a significantly extended and revised version of [1]. In particular, all the content of Section 6 is new. The article also contains the full proofs of all the results, not included in [1].

2 Preliminaries

In this section, we briefly recall the essential notions and terminology of term rewriting [12, 33, 41].

$\mathcal{V}$ denotes a countably infinite set of variables, and Σ denotes a set of function symbols, or signature, each of which has a fixed associated arity. Terms are viewed as labelled trees in the usual way, where $\mathcal{T}(\Sigma, \mathcal{V})$ and $\mathcal{T}(\Sigma)$ denote the non-ground term algebra and the ground algebra built on $\Sigma \cup \mathcal{V}$ and Σ, respectively. Positions are defined as sequences of positive natural numbers used to address subterms of a term, with ϵ as the root (or top) position (i.e., the empty sequence). Concatenation of positions p and q is denoted by $p.q$, and $p < q$ is the usual prefix ordering. The concatenation of a position p and a set of positions P is $p.P = \{p.q \mid q \in P\}$. Two positions p, q are disjoint, denoted by $p \parallel q$, if neither $p < q$, $p > q$, nor $p = q$. Given $S \subseteq \Sigma \cup \mathcal{V}$, $\mathcal{P}os_S(t)$ denotes the set of positions of a term t that are rooted by function symbols or variables in S. $\mathcal{P}os_{\{f\}}(t)$ with $f \in \Sigma \cup \mathcal{V}$ will be simply denoted by $\mathcal{P}os_f(t)$, and $\mathcal{P}os_{\Sigma \cup \mathcal{V}}(t)$ will be simply denoted by $\mathcal{P}os(t)$. $t|_p$ is the subterm at the position p of t. $t[s]_p$ is the term t with the subterm at the position p replaced with term s. By $Var(s)$, we denote the set of variables occurring in the syntactic object s. By $\bar{x}$, we denote a tuple of pairwise distinct variables. A *fresh* variable is a variable that appears nowhere else. A *linear* term is one where every variable occurs only once.

A *substitution* σ is a mapping from the set of variables $\mathcal{V}$ into the set of terms $\mathcal{T}(\Sigma, \mathcal{V})$, with a finite domain $D(\sigma)$ and image $I(\sigma)$. A substitution is represented as $\{x_1/t_1, \ldots, x_n/t_n\}$ for variables $x_1, \ldots, x_n$ and terms $t_1, \ldots, t_n$. The application of substitution θ to term t is denoted by $t\theta$, using postfix notation. Composition of substitutions is denoted by juxtaposition, i.e., the substitution $\sigma\theta$ denotes $(\theta \circ \sigma)$. We write $\theta_{\upharpoonright Var(s)}$ to denote the restriction of the substitution θ to the set of variables in s; by abuse of notation, we often simply write $\theta_{\upharpoonright s}$. Given a term t, we write $\theta = \nu$ $[t]$ for $\theta_{\upharpoonright Var(t)} = \nu_{\upharpoonright Var(t)}$, i.e., $\forall x \in Var(t)$, $x\theta = x\nu$. A substitution θ is more general than σ, denoted by $\theta \le \sigma$, if there is a substitution γ such that $\theta\gamma = \sigma$. A *unifier* of terms s and t is a substitution

3

ϑ such that $s\vartheta = t\vartheta$. The *most general unifier* of terms s and t, denoted by $mgu(s,t)$, is a unifier θ such that for any other unifier θ', $\theta \leq \theta'$.

A *term rewriting system* (TRS) $\mathcal{R}$ is a pair (Σ, R), where R is a finite set of rewrite rules of the form $l \to r$ such that $l, r \in \mathcal{T}(\Sigma, \mathcal{V})$, $l \notin \mathcal{V}$, and $Var(r) \subseteq Var(l)$. We will often write just $\mathcal{R}$ or (Σ, R) instead of $\mathcal{R} = (\Sigma, R)$. Given a TRS $\mathcal{R} = (\Sigma, R)$, the signature Σ is often partitioned into two disjoint sets $\Sigma = \mathcal{C} \uplus \mathcal{D}$, where $\mathcal{D} = \{f \mid f(t_1, \ldots, t_n) \to r \in R\}$ and $\mathcal{C} = \Sigma \setminus \mathcal{D}$. Symbols in $\mathcal{C}$ are called *constructors*, and symbols in $\mathcal{D}$ are called *defined functions*. The elements of $\mathcal{T}(\mathcal{C}, \mathcal{V})$ are called *constructor terms*. We let $Def(\mathcal{R})$ denote the set of defined symbols in $\mathcal{R}$.

A rewrite step is the application of a rewrite rule to an expression. A term $s \in \mathcal{T}(\Sigma, \mathcal{V})$ *rewrites* to a term $t \in \mathcal{T}(\Sigma, \mathcal{V})$, denoted by $s \xrightarrow{p}_{\mathcal{R}, \sigma} t$, if there exist $p \in \mathcal{P}os_\Sigma(s)$, $l \to r \in \mathcal{R}$, and substitution σ such that $s|_p = l\sigma$ and $t = s[r\sigma]_p$. When no confusion can arise, we omit the subscript in $\to_\mathcal{R}$. We also omit the reduced position p when it is not relevant. A term s is a *normal form* w.r.t. the relation $\to_\mathcal{R}$ (or simply a normal form), if there is no term t such that $s \to_\mathcal{R} t$. A term is a reducible expression or *redex* if it is an instance of the left hand side of a rule in $\mathcal{R}$. A term s is a *head normal form* if there are no terms t, t' s.t. $s \to^*_\mathcal{R} t' \xrightarrow{\epsilon}_\mathcal{R} t$.

A TRS $\mathcal{R}$ is *($\to$)-terminating* (also called *terminating, strongly normalizing,* or *noetherian*) if there are no infinite reduction sequences $t_1 \to_\mathcal{R} t_2 \to_\mathcal{R} \ldots$. A TRS $\mathcal{R}$ is *weakly normalizing* if every term has a normal form that can be reached by a finite rewriting sequence. A TRS $\mathcal{R}$ is *confluent* if, whenever $t \to^*_\mathcal{R} s_1$ and $t \to^*_\mathcal{R} s_2$, there exists a term w s.t. $s_1 \to^*_\mathcal{R} w$ and $s_2 \to^*_\mathcal{R} w$. A confluent and terminating TRS is called canonical[1]. In canonical TRS's, each term has one (and only one) normal form.

Two (possibly renamed) rules $l \to r$ and $l' \to r'$ *overlap* if there is $p \in \mathcal{P}os_\Sigma(l)$ and substitution σ such that $l|_p\sigma = l'\sigma$. The pair $\langle l\sigma[r'\sigma]_p, r\sigma \rangle$ is called a *critical pair*; it is called an *overlay* if $p = \epsilon$. A critical pair $\langle t, s \rangle$ is *trivial* if $t = s$. A TRS whose critical pairs are overlays is called an *overlay TRS*. A left-linear TRS whose critical pairs are trivial overlays is called *almost orthogonal*.

Narrowing is a symbolic computation mechanism that generalizes rewriting by replacing pattern matching with syntactic unification. Many redundancies in the narrowing algorithm can be eliminated by restricting narrowing steps to a distinguished set of *basic* positions, which was proposed by Hullot in [24].

3 Basic Narrowing and a Novel Commutation Result

Basic narrowing is a restriction of narrowing introduced by Hullot [24] which is essentially based on forbidding narrowing steps on terms brought in by instantiation. We use the definition of basic narrowing given in [22], where the

---

[1]Canonical TRS's are sometimes called *complete* [29].

expression to be narrowed is split into a *skeleton* t and an *environment* part θ, i.e., $\langle t, \theta \rangle$. The environment part keeps track of the accumulated substitution so that, at each step, substitutions are composed in the environment part, but are not applied to the expression in the skeleton part, as opposed to ordinary narrowing. For TRS $\mathcal{R}$, $l \to r \ll \mathcal{R}$ denotes that $l \to r$ is a *fresh* variant of a rule in $\mathcal{R}$, i.e., all the variables are *fresh*.

Definition 1 (Basic narrowing) [22] *Given a term $s \in \mathcal{T}(\Sigma, \mathcal{V})$ and a substitution σ, a basic narrowing step for $\langle s, \sigma \rangle$ is defined by $\langle s, \sigma \rangle \overset{b}{\leadsto}_{p, \mathcal{R}, \theta} \langle t, \sigma' \rangle$ if there exist $p \in \mathcal{P}os_\Sigma(s)$, $l \to r \ll \mathcal{R}$, and substitution θ such that $\theta = mgu(s|_p \sigma, l)$, $t = (s[r]_p)$, and $\sigma' = \sigma\theta$.*

Along a basic narrowing derivation, the set of *basic* occurrences of each term $t\theta$ occurring in the sequence (given by $\langle t, \theta \rangle$) is $\mathcal{P}os_\Sigma(t)$, and the non–basic occurrences are $\mathcal{P}os_\Sigma(t\theta) - \mathcal{P}os_\Sigma(t)$. When p is not relevant, we simply denote the basic narrowing relation by $\overset{b}{\leadsto}_{\mathcal{R}, \theta}$. By abuse of notation, we also often relax the skeleton-environment notation for basic narrowing steps, i.e., $\langle s, \sigma \rangle \overset{b}{\leadsto}_{\mathcal{R}, \theta} \langle t, \sigma' \rangle$, and use the more compact notation $s\sigma \overset{b}{\leadsto}_{\mathcal{R}, \theta} t\sigma\theta$ instead, but then a suitable track of the basic positions along the narrowing sequences is implicitly done.

We say that $\mathcal{R}$ is ($\overset{b}{\leadsto}$)-terminating (or that basic narrowing terminates in $\mathcal{R}$) when every basic narrowing derivation issuing from any term terminates. All modular termination results in this paper are based on the following termination result for basic narrowing that we developed in [3]. It is essentially Hullot's basic narrowing termination result, but we have explicitly dropped the superfluous requirement of canonicity.

Theorem 1 (Termination of Basic Narrowing) [24, 3] *Let $\mathcal{R}$ be a TRS. If for every $l \to r \in \mathcal{R}$, all basic narrowing derivations issuing from r terminate, then basic narrowing is terminating in $\mathcal{R}$.*

Hullot's termination result provides a semi-decidable criteria for ($\overset{b}{\leadsto}$)-termination by just executing the rhs's of the rules. In the literature, the original criterion that all basic narrowing derivations issuing from the rhs's of rules terminate has been approximated by two different sufficient syntactic conditions: every rhs of a rewrite rule is either (i) a constructor term [24] (e.g. a variable) or (ii) a ground term (provided that $\mathcal{R}$ is ($\to$)-terminating) [13]. The condition that all basic narrowing derivations issuing from the rhs's of rules terminate has been generalized in [3] to the requirement that the rhs's of the rewrite rules ares *rigid normal forms* (rnf), i.e., unnarrowable.

The following well-known example shows a mildly surprising fact: ($\overset{b}{\leadsto}$)-termination does not entail ($\to$)-termination.

Example 1 *Let us consider the reference example by Toyama [42] that shows the non-modularity of ($\to$)-termination for disjoint unions (see [33]):*

$$\mathcal{R}_0 : \mathtt{f}(0, 1, x) \to \mathtt{f}(x, x, x) \qquad \mathcal{R}_1 : \mathsf{g}(x, y) \to x \quad \mathsf{g}(x, y) \to y$$

Both, $\mathcal{R}_0$ and $\mathcal{R}_1$ are $(\rightarrow)$terminating, whereas the system $\mathcal{R} = \mathcal{R}_0 \cup \mathcal{R}_1$ is not $(\rightarrow)$-terminating as witnessed by $\mathtt{f}(0,1,g(0,1)) \rightarrow_{\mathcal{R}} \mathtt{f}(g(0,1),g(0,1),g(0,1)) \rightarrow_{\mathcal{R}}^{*}$ $\mathtt{f}(0,1,g(0,1)) \rightarrow_{\mathcal{R}} \ldots$. Regarding basic narrowing both, $\mathcal{R}_0$ and $\mathcal{R}_1$ are $(\overset{b}{\rightsquigarrow})$-terminating since every rhs is unnarrowable, and trivially $\mathcal{R}$ is also $(\overset{b}{\rightsquigarrow})$-terminating, since the right hand sides are still unnarrowable in the disjoint union $\mathcal{R}_0 \cup \mathcal{R}_1$.

Nevertheless, note that the system $\mathcal{R} = \mathcal{R}_0 \cup \mathcal{R}_1$ is[2] *strongly innermost normalizing (SIN)*, i.e., every innermost[3] rewriting sequence is finite. Actually, by formalizing the idea that every innermost rewriting sequence is a basic sequence [25, 29, 44], we are able to prove that $(\overset{b}{\rightsquigarrow})$-termination entails termination of *innermost rewriting*, even if does not entail $(\rightarrow)$-termination. To the best of our knowledge, this is the first time that this result is explicit (and exploited) in a formal publication.

In order to formalize this proof, we find it convenient to recall the original definition of basic narrowing by Hullot [24] that is based on tracking sets of positions that are enabled for narrowing.

Definition 2 (Basic Narrowing Derivations) [24] *Let*

$$t_1 \overset{p_1}{\rightsquigarrow}_{\sigma_1, l_1 \rightarrow r_1} t_2 \overset{p_2}{\rightsquigarrow}_{\sigma_2, l_2 \rightarrow r_2} \cdots \overset{p_n}{\rightsquigarrow}_{\sigma_n, l_n \rightarrow r_n} t_{n+1}$$

be a narrowing derivation. We inductively define sets of positions $B_1, \cdots, B_n$ as follows:

$$
\begin{aligned}
B_1 &= \mathcal{P}os_\Sigma(t_1) \\
B_{i+1} &= (B_i - \{q \in B_i \mid p_i \le q\}) \cup \{p_i.q \mid q \in \mathcal{P}os_\Sigma(r_i)\}
\end{aligned}
$$

Positions in B_i are called basic positions *and positions in $\mathcal{P}os_\Sigma(t_i) - B_i$ are called* non-basic $(1 \le i \le n)$. *We say that the above narrowing derivation is* basic *if $p_i \in B_i$ for $1 \le i \le n$.*

We also provide an auxiliary notion of basic rewriting sequences based on the basic positions of the previous definition.

Definition 3 (Basic Rewriting Derivations) *Given a rewrite derivation*

$$t_1 \overset{p_1}{\rightarrow}_{l_1 \rightarrow r_1, \rho_1} t_2 \overset{p_2}{\rightarrow}_{l_2 \rightarrow r_2, \rho_2} \cdots \overset{p_n}{\rightarrow}_{l_n \rightarrow r_n, \rho_n} t_{n+1}$$

we say it is based on a set of positions B (or simply called basic), *with $B_1, B_2, \ldots, B_n$ defined as in Definition 2, if $B = B_1$, and $p_i \in B_i$ for $1 \le i \le n$.*

Note that each basic rewriting sequence is a basic narrowing derivation. We can prove the result that innermost rewriting sequences are basic.

---

[2]Its innermost dependency graph contains no cycles and the results in [7] apply directly.

[3]A reduction step $t \Rightarrow^p t'$ at position p is called *innermost* if no strict subterm of $t|_p$ is reducible by $\Rightarrow$

Proposition 1 [25] *Let $\mathcal{R}$ be a TRS. Any innermost rewriting sequence starting from a term $t\sigma$ is based on $\mathcal{P}os_\Sigma(t)$.*

Now it is immediate to derive the result that termination of basic narrowing entails termination of innermost rewriting (SIN), which will be used for proving our results in Section 6.

Proposition 2 *Termination of basic narrowing implies strong innermost normalization.*

Proof.   By contradiction. Suppose there exists $\mathcal{R}$ that is ($\overset{b}{\leadsto}$)-terminating but there is an infinite innermost rewriting sequence in $\mathcal{R}$. However, since this sequence is a basic narrowing derivation, this contradicts the initial assumption of $\mathcal{R}$ being ($\overset{b}{\leadsto}$)-terminating. ∎

The rest of this section is devoted to study the commutation properties of basic narrowing derivations, which are essential for proving the modularity of basic narrowing termination in Section 5.2. The commutation of ordinary narrowing derivations was extensively studied by Rety in [38] and we follow the same style here. In the following, we consider basic narrowing derivations of the form

$$s \overset{b}{\leadsto}_{p,g\to d,\sigma} t \overset{b}{\leadsto}_{q,l\to r,\theta} u \overset{b}{\leadsto} \dots \tag{1}$$

and we are interested in the conditions that allow us to commute the first two steps by first applying to s the rule $l \to r$ and then the rule $g \to d$ to the resulting term. If the subterm $t|_q$ already exists in s, i.e., if q admits at least one antecedent in s, the idea essentially consists in applying $l \to r$ to all the antecedents of q, and then applying $g \to d$ to the resulting term. Let us give an example for motivation.

Example 2 [38] *Let us consider the following TRS $\mathcal{R}$:*

$$\mathcal{R} = \begin{cases} \mathtt{f}(x,x) & \to & x & \textit{(r1)} \\ \mathtt{g}(x,\mathtt{h}(x)) & \to & x & \textit{(r2)} \end{cases}$$

and the following basic narrowing derivation, where $p = 1$ and $q = 2$:

$$\langle \mathtt{h}(\mathtt{f}(\mathtt{0},x),\mathtt{g}(x,y)),\{\}\rangle \quad \overset{b}{\leadsto}_{p,r1,\{x/\mathtt{0}\}} \langle \mathtt{h}(x,\mathtt{g}(x,y)),\{x/\mathtt{0}\}\rangle$$
$$\overset{b}{\leadsto}_{q,r2,\{y/\mathtt{h}(\mathtt{0})\}} \langle \mathtt{h}(x,x),\{x/\mathtt{0},y/\mathtt{h}(\mathtt{0})\}\rangle.$$

The occurrences p and q are disjoint, therefore q has an antecedent in s at $q_0 = 2$. By first applying $r2$ at q_0, and then $r1$ at p we get:

$$\langle \mathtt{h}(\mathtt{f}(\mathtt{0},x),\mathtt{g}(x,y)),\{\}\rangle \quad \overset{b}{\leadsto}_{q_0,r2,\{y/\mathtt{h}(\mathtt{x})\}} \langle \mathtt{h}(\mathtt{f}(\mathtt{0},x),x),\{y/\mathtt{h}(\mathtt{x})\}\rangle$$
$$\overset{b}{\leadsto}_{p,r1,\{x/\mathtt{0}\}} \langle \mathtt{h}(x,x),\{y/\mathtt{h}(\mathtt{0}),x/\mathtt{0}\}\rangle$$

Let us recall the notion of *antecedent* of a position in a rewriting sequence [38].

Definition 4 (Antecedent of a Position) [38] *Let $t \xrightarrow{p}_{l \to r} t'$ be a rewriting step, $v \in \mathcal{P}os(t)$, and $v' \in \mathcal{P}os(t')$. We say position v is an* antecedent *of v' iff either*

1. *$v \parallel p$, i.e., v and p are disjoint, and $v = v'$, or*

2. *there exists an occurrence $u' \in \mathcal{P}os(r)$ of a variable x in r s.t. $v' = p.u'.w$ and $v = p.u.w$, where $u \in \mathcal{P}os(l)$ is an occurrence of x in l.*

With the notations of the previous definition, we have:

1. $t|_v = t'|_{v'}$,

2. v' may have no antecedent if $v' = p.u'$ with $u' \in \mathcal{P}os_\Sigma(r)$, or if $v' < p$,

This notion extends to a rewrite sequence by transitive closure of the rewriting relation in the usual way. The notion of antecedent can also be extended to narrowing sequences as follows.

Definition 5 (Narrowing Antecedent of a Position) [38] *Let $t \overset{b}{\leadsto}{}^*_{\mathcal{R},\sigma} t'$, $v \in \mathcal{P}os(t)$, and $v' \in \mathcal{P}os(t')$. We say v is a* narrowing antecedent *of v' iff v is an antecedent of v' in the rewrite sequence $t\sigma \to^*_{\mathcal{R}} t'$.*

Note that if v is an antecedent of v' in the narrowing sequence above, then $t|_v\sigma = t'|_{v'}$.

The following result establishes that, in a basic narrowing derivation, the antecedent of a position is always in the skeleton, and case 2 of Definition 4 cannot happen.

Lemma 1 *Given a basic narrowing derivation $t \overset{b}{\leadsto}_{p,l \to r,\sigma} t' \overset{b}{\leadsto}_{q',g \to d,\theta} u$, and $q \in \mathcal{P}os(t)$, if q is an antecedent of q', then q and p are necessarily disjoint, q' is in the skeleton part of t', and $q = q'$.*

Proof. Suppose that q and p are not disjoint, then we are in case 2 of Definition 4, and $q' = p.u'.w$ for some $u \in Pos_x(r)$. But this means q' is in the environment, and hence the narrowing derivation is not basic, which contradicts the initial assumption. ∎

Now we show that basic narrowing steps can be commuted under ascertained conditions. This result is the basis for the modularity results of Section 5.2.

Proposition 3 (Commutation of Basic Narrowing) *Every basic narrowing derivation of the form*

$$t \overset{b}{\leadsto}_{p,g \to d,\sigma_1} (t[d]_p)\sigma_1 \overset{b}{\leadsto}_{q,l \to r,\sigma_2} u \tag{2}$$

where q admits an antecedent in t, i.e., p and q are disjoint positions, can be commuted to an equivalent basic narrowing derivation:

$$t \stackrel{b}{\leadsto}_{q,l \to r,\sigma_3} (t[r]_q)\sigma_3 \stackrel{b}{\leadsto}_{p,g \to d,\sigma_4} u \tag{3}$$

and $\sigma_1\sigma_2 = \sigma_3\sigma_4$.

Proof. The following proof is an adaptation of the proof found in [30, Lemma 26].

It is useful to make explicit here the skeleton-environment formulation of the derivations involved:

$$\langle s, \theta \rangle \stackrel{b}{\leadsto}_{p,g \to d,\sigma_1} \langle s[d]_p, \theta\sigma_1 \rangle \stackrel{b}{\leadsto}_{q,l \to r,\sigma_2} \langle s[d]_p[r]_q, \theta\sigma_1\sigma_2 \rangle \quad (2)$$
$$\langle s, \theta \rangle \stackrel{b}{\leadsto}_{q,l \to r,\sigma_3} \langle s[r]_q, \theta\sigma_3 \rangle \stackrel{b}{\leadsto}_{p,g \to d,\sigma_4} \langle s[r]_q[d]_p, \theta\sigma_3\sigma_4 \rangle \quad (3)$$

where $t = s\theta$ and $u = (s[d]_p[r]_q)\theta\sigma_1\sigma_2 = (s[d]_p[r]_q)\theta\sigma_3\sigma4$. Note that positions q and p come from the skeleton.

We need to prove that σ_3 and σ_4 exist and that $\sigma_1\sigma_2 = \sigma_3\sigma_4$. Existence of σ_3 follows from the fact that $t|_q$ and l are unifiable.

$$t|_q\sigma_1\sigma_2 = t\sigma_1|_q\ \sigma_2 = (t[d]_p)\sigma_1|_q\ \sigma_2 = l\sigma_2 = l\sigma_1\sigma_2$$

Note that the second step in (2) uses the fact that p and q are disjoint positions; the first step formulation in (3) follows from (2); and the last step uses the fact that l is the lhs of a fresh variant of a rule, and hence $Var(l) \cap Dom(\sigma_1) = \emptyset$.

So $t|_q$ and l are unifiable; let σ_3 be an idempotent most general unifier of these two terms. By definition, $\sigma_3 \leq \sigma_1\sigma_2$ and there exists ρ such that $\sigma_3\rho = \sigma_1\sigma_2$. We show now that the terms $t[r]_q\sigma_3|_p$ and g are unifiable.

$$(t[r]_q)\sigma_3|_p\ \rho = (t[r]_q|_p)\ \sigma_3\rho = (t[r]_q|_p)\ \sigma_1\sigma_2 = g\sigma_1\sigma_2 = g\sigma_3\rho = g\rho$$

In (3), the second step formulation takes advantage of the fact that $Var(g) \cap Dom(\sigma_3) = \emptyset$. To see why, consider that g is a fresh variant and $Dom(\sigma_3) \subseteq Var(t|_q) \cup Var(l)$.

By analogous reasoning, since $(t[r]_q)\sigma_3|_p$ and g are unifiable let δ be an idempotent most general unifier, $\delta \leq \rho$. It follows that $\sigma_3\delta \leq \sigma_3\rho$ and hence $\sigma_3\delta \leq \sigma_1\sigma_2$. It should also be clear that $\sigma_3\delta$ is a unifier of $t|_p$ and g, using the fact that $Var(g) \cap Dom(\sigma_3) = \emptyset$ since $Dom(sigma_3) \subseteq Var(t|_p) \cup Var(l)$.

$$t|_p\sigma_3\delta = (t[r]_q|_p)\sigma_3\delta = (t[r]_q)\sigma_3|_p\ \delta = g\delta = g\sigma_3\delta$$

Because σ_1 is a most general unifier for these two terms, $\sigma_1 \leq \sigma_3\delta$, and there is a substitution ρ_2 such that $\sigma_1\rho_2 = \sigma_3\delta$. By considering that $Var(l) \cap Dom(\sigma_1) = \emptyset$, we conclude

$$(t[d]_q)\sigma_1|_p\ \rho_2 = (t[d]_q|_p)\sigma_1\rho_2 = (t[d]_q|_p)\sigma_3\delta = l\sigma_3\delta = l\sigma_1\rho_2 = l\rho_2$$

Now, since σ_2 is an idempotent most general unifier of $(t[d]_q)\sigma_1|_p$ and l, it follows that $\sigma_2 \leq \rho_2$, and therefore $\sigma_1\sigma_2 \leq \sigma_1\rho_2 = \sigma_3\delta$. But we also have $\sigma_3\delta \leq \sigma_1\sigma_2$, therefore there is a variable renaming τ such that $\sigma_1\sigma_2 = \sigma_3\delta\tau$. Now define $\sigma_4 = \delta\tau$; σ_4 is still a most general unifier of $(t[r]_q)\sigma_3|_p$ and g as required, because most general unifiers are closed under variable renaming [26, Corollary 9]. Hence $\sigma_1\sigma_2 = \sigma_3\sigma_4$, which concludes the proof. ∎

4 Modularity of Basic Narrowing Termination

In this section we recall the standard classes of modular combinations of TRSs, and discuss the standard notion of $\mathcal{C}_\varepsilon$-termination that we lift to basic narrowing. Then we present the first set of results about the modularity of basic narrowing termination, including only those combinations in which termination is *always* modular. We discuss the combinations in which modularity requires additional restrictions in Section 5.

The following classes of combinations are standard in the literature. Note that we sometimes use the word "union" to denote a combination of two TRSs (sometimes simply called systems). In the case of a hierarchical combination, we sometimes write "an extension system" to denote the combination of two TRSs without clarifying which is the base and which is the extension. We sometimes write $(\mathcal{D} \uplus \mathcal{D}' \uplus \mathcal{C}, R)$ to denote a TRS where $\mathcal{C}$ is the set of constructor symbols and $\mathcal{D} \uplus \mathcal{D}'$ is the set of defined symbols such that $\mathcal{D}$ and $\mathcal{D}'$ are disjoint. We borrow the presentation from [33].

disjoint $\mathcal{R}_0 = (\Sigma_0, R_0)$ and $\mathcal{R}_1 = (\Sigma_1, R_1)$ are *disjoint* if they do not share symbols, i.e., $\Sigma_0 \cap \Sigma_1 = \varnothing$. Their union, called *direct sum*, is denoted $\mathcal{R} = \mathcal{R}_0 \uplus \mathcal{R}_1$.

constructor sharing $(\mathcal{D}_0 \uplus \mathcal{C}_0, R_0)$ and $(\mathcal{D}_1 \uplus \mathcal{C}_1, R_1)$ are *constructor sharing* if they do not share defined symbols, i.e., $\mathcal{D}_0 \cap \mathcal{D}_1 = \varnothing$.

composable Two systems $(\mathcal{D}_0 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_0, R_0)$ and $(\mathcal{D}_1 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_1, R_1)$ are *composable* if $\mathcal{D}_0 \cap \mathcal{D}_1 = \mathcal{D}_0 \cap \mathcal{C}_1 = \mathcal{D}_1 \cap \mathcal{C}_0 = \varnothing$ and both systems share all the rewrite rules that define every shared defined symbol, i.e., $R_{sh} \subseteq R_0 \cap R_1$ where $R_{sh} = \{l \to r \in R_0 \cup R_1 \mid root(l) \in \mathcal{D}_{sh}\}$.

hierarchical combination A system $\mathcal{R} = \mathcal{R}_0 \cup \mathcal{R}_1$ is the *hierarchical combination* (HC) of a base system $\mathcal{R}_0 = (\mathcal{D}_0 \uplus \mathcal{C}_0, R_0)$ and an extension system $\mathcal{R}_1 = (\mathcal{D}_1 \uplus \mathcal{C}_1, R_1)$ iff $\mathcal{D}_0 \cap \mathcal{D}_1 = \varnothing$ and $\mathcal{C}_0 \cap \mathcal{D}_1 = \varnothing$.

generalized hierarchical combination A system $\mathcal{R} = \mathcal{R}_0 \cup \mathcal{R}_1$ is the *generalized hierarchical combination* (GHC) of a base system $\mathcal{R}_0 = (\mathcal{D}_0 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_0, R_0)$ and an extension $\mathcal{R}_1 = (\mathcal{D}_1 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_1, R_1)$ with a shared subsystem $(\mathcal{F}, R_{sh})$ iff $\mathcal{D}_0 \cap \mathcal{D}_1 = \varnothing$, $\mathcal{C}_0 \cap \mathcal{D}_1 = \varnothing$, $R_{sh} = R_0 \cap R_1 = \{l \to r \in R_0 \cup R_1 \mid root(l) \in \mathcal{D}_{sh}\}$, and $\mathcal{F} = \{f \in \mathcal{F} \mid f \text{ occurs in } R_{sh}\}$.

Figure 1 shows diagramatic renditions of the definitions introduced above. Roughly speaking, in a hierarchical combination $\mathcal{R} = \mathcal{R}_0 \cup \mathcal{R}_1$ the sets of function symbols defined in $\mathcal{R}_0$ and $\mathcal{R}_1$ are disjoint, and the defined function symbols of the base ($\mathcal{R}_0$) can occur in rules of the extension, but not viceversa. Note that constructor sharing unions generalize disjoint unions, and are themselves generalized by both composable unions and HCs. Finally, these last two combinations are subsumed by GHCs. These relations of subsumption are illustrated in Figure 2.

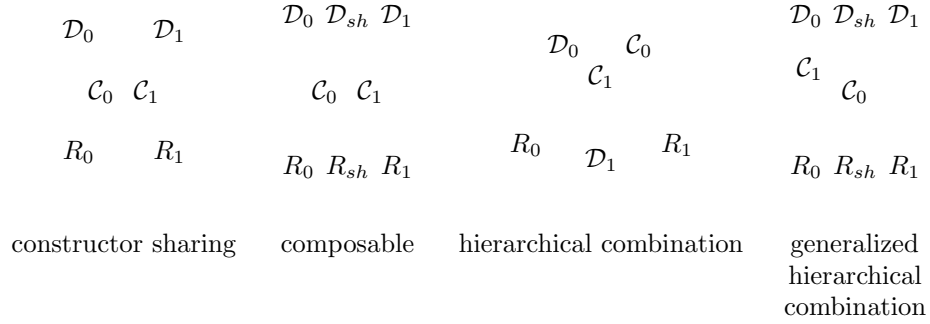The following example borrowed from [33] illustrates these notions.

$$
\begin{array}{cccc}
\mathcal{D}_0 \quad \mathcal{D}_1 & \mathcal{D}_0 \ \mathcal{D}_{sh} \ \mathcal{D}_1 & & \mathcal{D}_0 \ \mathcal{D}_{sh} \ \mathcal{D}_1 \\
\mathcal{C}_0 \ \mathcal{C}_1 & \mathcal{C}_0 \ \mathcal{C}_1 & \mathcal{D}_0 \quad \mathcal{C}_0 & \mathcal{C}_1 \\
& & \mathcal{C}_1 & \mathcal{C}_0 \\
R_0 \quad R_1 & R_0 \ R_{sh} \ R_1 & R_0 \quad R_1 & R_0 \ R_{sh} \ R_1 \\
& & \mathcal{D}_1 &
\end{array}
$$

constructor sharing   composable   hierarchical combination   generalized hierarchical combination
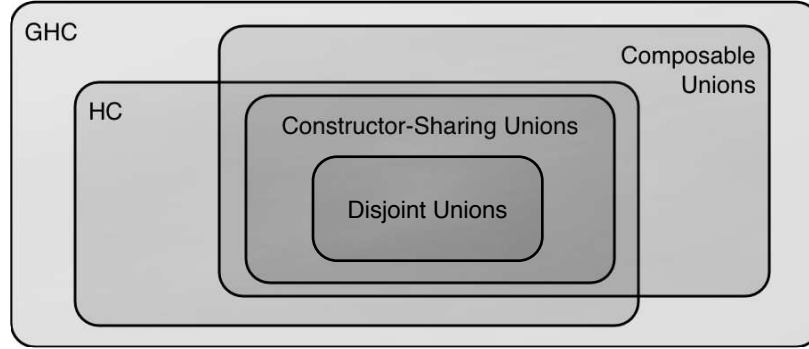
Figure 1: Standard modular combinations

Figure 2: Modular combinations

Example 3 *Consider the following TRSs:*

$$
\mathcal{R}_+ = \left\{ \begin{array}{rcl}
0 + y & \to & y \\
\mathbf{s}(x) + y & \to & \mathbf{s}(x + y)
\end{array} \right.
$$

$$
\mathcal{R}_- = \left\{ \begin{array}{rcl}
0 - \mathbf{s}(y) & \to & 0 \\
x - 0 & \to & x \\
\mathbf{s}(x) - \mathbf{s}(y) & \to & x - y
\end{array} \right.
$$

$$
\mathcal{R}_* = \left\{ \begin{array}{rcl}
0 * y & \to & 0 \\
\mathbf{s}(x) * y & \to & (x * y) + y
\end{array} \right.
$$

$$
\mathcal{R}_{pow} = \left\{ \begin{array}{rcl}
\mathtt{pow}(x, 0) & \to & \mathbf{s}(0) \\
\mathtt{pow}(x, \mathbf{s}(y)) & \to & x * \mathtt{pow}(x, y)
\end{array} \right.
$$

$$
\mathcal{R}_{app} = \left\{ \begin{array}{rcl}
\mathtt{nil} \mathbin{+\!\!+} ys & \to & ys \\
(x : xs) \mathbin{+\!\!+} ys & \to & x : (xs \mathbin{+\!\!+} ys)
\end{array} \right.
$$

11

$\mathcal{R}_+$ and $\mathcal{R}_{app}$ are disjoint, $\mathcal{R}_+$ and $\mathcal{R}_-$ are constructor–sharing, $\mathcal{R}_+ \cup \mathcal{R}_*$ is composable with $\mathcal{R}_+ \cup \mathcal{R}_{app}$, and $\mathcal{R}_* \cup \mathcal{R}_+$ is a HC where $\mathcal{R}_*$ extends $\mathcal{R}_+$. Lastly, the system $\mathcal{R}_1 = \mathcal{R}_{pow} \cup \mathcal{R}_+$ extends $\mathcal{R}_0 = \mathcal{R}_* \cup \mathcal{R}_+$ in a GHC with a shared subsystem $\mathcal{R}_{sh} = \mathcal{R}_+$.

As noted by [36], this classification of combinations of TRSs is straightforwardly applicable to programming languages and incremental program development. The modularity results of direct–sums can be used when two subsystems are defined over different domains, e.g. the natural numbers and the Boolean domain. The modularity results of constructor sharing unions can be used when two subsystems define independent functions (none of the two systems use the procedures defined in the other) over a common domain. HCs model the notion of *modules* in programming languages. In this context, modular termination proofs can pave the way towards applying standard termination analyses and tools to real–life languages since this offers a means to connect termination problems in programming languages to lower-level, rewriting-based solutions. Actually, for proving termination of programs of real-life programming languages supporting a wide range of programming and software development features, termination problems are often step-by-step transformed into "simpler" ones, and then finally translated by rewriting-based termination tools into constraint-solving problems of a given logic which can be handled by standard solvers (see [14, 19]).

In the rest of this section, we show that the ($\overset{b}{\rightsquigarrow}$)-termination is modular for disjoint unions, constructor–sharing unions, and the union of composable systems.

We start the discussion by extending to basic narrowing the standard notion of $\mathcal{C}_\varepsilon$-termination and proving that $\mathcal{C}_\varepsilon$-termination of basic narrowing is implied by ($\overset{b}{\rightsquigarrow}$)-termination.

4.1 $\mathcal{C}_\varepsilon$-termination

The notion of $\mathcal{C}_\varepsilon$-termination is used in the literature for proving termination of rewriting in a modular way [33]. A popular proof technique is essentially based on finding sufficient conditions for the modularity of the more restrictive $\mathcal{C}_\varepsilon$-termination property, which is a sufficient condition for the modularity of ($\rightarrow$)-termination.

Definition 6 ($\mathcal{C}_\varepsilon$-termination) [33] *A TRS $\mathcal{R}$ is called $\mathcal{C}_\varepsilon$-terminating if the collapsing extended TRS $\mathcal{R} \uplus \mathcal{C}_\varepsilon$, where $\mathcal{C}_\varepsilon = \{\texttt{Cons}(x, y) \to x, \texttt{Cons}(x, y) \to y\}$, is terminating, where* $\texttt{Cons}$ *is a fresh function symbol not occurring in the signature of $\mathcal{R}$.*

This definition can be lifted in the obvious way to (basic) narrowing termination of $\mathcal{R} \uplus \mathcal{C}_\varepsilon$, which we call $\mathcal{C}_\varepsilon$-($\overset{b}{\rightsquigarrow}$)-*termination*. The following consequence of Theorem 1 is interesting.

Proposition 4 ($\mathcal{C}_\varepsilon$-termination of Basic Narrowing) *Let $\mathcal{R}$ be a TRS. $\mathcal{R}$ is ($\overset{b}{\leadsto}$)-terminating if and only if the TRS $\mathcal{R} \uplus \mathcal{C}_\varepsilon$ is ($\overset{b}{\leadsto}$)-terminating.*

Proof. Immediate by Theorem 1 because the rhs's of the rules in $\mathcal{C}_\varepsilon$ satisfy the termination condition for basic narrowing derivations of Theorem 1 and `Cons` is a fresh function symbol. ∎

In contrast, a similar property does not hold for rewriting, but only for innermost rewriting (see [33]). That is, $\mathcal{C}_\varepsilon$-termination is a sufficient condition for ($\rightarrow$)-termination, whereas innermost $\mathcal{C}_\varepsilon$-termination (resp. $\mathcal{C}_\varepsilon$-($\overset{b}{\leadsto}$)-termination) is a sufficient and necessary condition for SIN (resp. ($\overset{b}{\leadsto}$)-termination). This means that both ($\overset{b}{\leadsto}$)-termination and SIN are unaffected by non deterministic collapses produced by sets of rules like the ones for `cons` in Example 1. As a result, SIN is much more modular than ($\rightarrow$)-termination and, as we will see in the following sections, ($\overset{b}{\leadsto}$)-termination is at least as modular as SIN.

4.2 Disjoint and Constructor–sharing Unions

We start by showing that ($\overset{b}{\leadsto}$)-termination is modular in constructor–sharing unions, a result easily derivable from Theorem 1.

Theorem 2 (Modularity of Constructor–sharing Unions) *Termination of basic narrowing is a modular property for the union of constructor–sharing systems.*

Proof. Let $\mathcal{R} = \mathcal{R}_0 \cup \mathcal{R}_1$ be a constructor-sharing union of two ($\overset{b}{\leadsto}$)-terminating TRSs. We need to prove that any basic narrowing derivation issuing from every right hand side in $\mathcal{R}$ is finite. Every derivation stemming from the right–hand side of a rule of $\mathcal{R}_0$ only uses rules from $\mathcal{R}_0$ (there is no redex w.r.t. $\mathcal{R}_1$), and then such derivations are finite by hypothesis, since $\mathcal{R}_0$ is ($\overset{b}{\leadsto}$)-terminating. The same argument applies to the right–hand sides of the rules of $\mathcal{R}_1$. Finally, by Theorem 1, the conclusion follows. ∎

Obviously, modularity of ($\overset{b}{\leadsto}$)-termination for constructor–sharing unions implies modularity for disjoint unions as well. Example 1 illustrates this.

Corollary 1 (Modularity of Disjoint Unions) *Termination of basic narrowing is a modular property for the union of disjoint systems.*

A comment on the modularity of unrestricted narrowing We note that a similar result does not hold for unrestricted narrowing in disjoint unions. As an easy counterexample, let us consider again Example 1. Narrowing terminates in each system in the example[4]. However, narrowing does not terminate in

---

[4]By using [3] we can prove termination in $\mathcal{R}_0$ since the rhs is (strongly) non–narrowable, or using the terminology of [3], is a strongly rigid normal form (srnf), and the system is srnf–based, that is, the arguments of the lhs's are srnf's too. And by [2] narrowing terminates in $\mathcal{R}_1$ since its estimated dependency graph contains no cycles.

the union $\mathcal{R} = \mathcal{R}_0 \cup \mathcal{R}_1$, since $\mathcal{R}$ is not ($\rightarrow$)-terminating and every rewriting derivation in $\mathcal{R}$ is also a narrowing derivation in $\mathcal{R}$.

4.3   Composable unions

Composable unions generalize constructor–sharing unions in that the two systems are allowed to have a set $\mathcal{D}_{sh}$ of shared defined symbols and the rules R_{sh} defining them are all included in both systems. A consequence of this is that the rules in R_{sh} cannot contain calls to functions defined out of R_{sh}.

Lemma 2 *Let $(\mathcal{D}_0 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_0, R_0)$ and $(\mathcal{D}_1 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_1, R_1)$ be two* composable *systems with a shared system $R_{sh} = \{l \rightarrow r \in R_0 \cup R_1 \mid root(l) \in \mathcal{D}_{sh}\}$. Then no right–hand side of a rule in R_{sh} contains a function symbol in $\mathcal{D}_0 \cup \mathcal{D}_1$.*

Proof. Let us assume that there is a rule in R_{sh} whose right hand side contains a function symbol $f \in \mathcal{D}_i$ where $i \in \{0, 1\}$. f cannot be a defined symbol in the other system $\mathcal{D}_{1-i}$, since by definition $\mathcal{D}_0 \cap \mathcal{D}_1 = \varnothing$. This implies that f must be a constructor symbol in the other system, i.e., $f \in \mathcal{C}_{1-i}$. But also by definition this is not allowed, as $\mathcal{D}_0 \cap \mathcal{C}_1 = \mathcal{D}_1 \cap \mathcal{C}_0 = \varnothing$. Then f must be a shared symbol, $f \in \mathcal{D}_{sh}$, which contradicts the initial assumption. $\blacksquare$

The next theorem extends the modularity of ($\overset{b}{\rightsquigarrow}$)-termination to composable systems.

Theorem 3 (Modularity of Composable Unions) *Termination of basic narrowing is a modular property of composable systems.*

Proof. Let $\mathcal{R} = \mathcal{R}_0 \cup \mathcal{R}_1$ be the union of two composable, ($\overset{b}{\rightsquigarrow}$)-terminating TRSs. We can partition the rules in $\mathcal{R}$ in the following disjoint sets:

$$\mathcal{R}_{sh} = \mathcal{R}_0 \cap \mathcal{R}_1$$
$$\mathcal{R}'_0 = \mathcal{R}_0 \setminus \mathcal{R}_{sh}$$
$$\mathcal{R}'_1 = \mathcal{R}_1 \setminus \mathcal{R}_{sh}$$

Basic narrowing derivations issuing from the right–hand sides of the rules in $\mathcal{R}'_0$ or $\mathcal{R}'_1$ terminate by an argument analog to the one used in the proof of Theorem 2. By Lemma 2, derivations issuing from the right–hand sides of $\mathcal{R}_{sh}$ include no redex w.r.t. either $\mathcal{R}'_0$ or $\mathcal{R}'_1$. Because basic narrowing does never reduce any redex occurring at a non–basic position (i.e. in the environment part of the tuple representation), these derivations must also terminate. By Theorem 1, the conclusion follows. $\blacksquare$

5 Modularity in Hierarchical Combinations

So far we have seen that $(\overset{b}{\leadsto})$-termination is modular without additional conditions for the unions of disjoint systems, constructor-sharing systems and composable systems. For the class of (generalized) hierarchical combinations however, $(\overset{b}{\leadsto})$-termination is not modular in general, but only under some restrictions. In this section we prove that modularity holds for the novel class of *generalized relaxed proper extensions* (GRPE). *Proper extensions* (PE) are a restriction of hierarchical combinations introduced by Krishna Rao in [36] in order to model the modularity of SIN, although they have proved useful in other contexts too, see [33] for an extensive survey. *Generalized proper extensions* (GPE) extend proper extensions to two systems with a shared subsystem. Generalized relaxed proper extensions are less restrictive on the right hand sides than generalized proper extensions.

In order to prove modularity in the class of generalized relaxed proper extensions we adapt a succesful proof scheme introduced by Krishna Rao in [36] to study modularity of innermost termination. This scheme is based on modularly decomposing a proper extension into a number of layered *nice extensions*, a class of hierarchical combinations smaller than proper extensions. Nice extensions are appealing because modularity results that hold in this class can be lifted to proper extensions by means of a suitable modular argument. Informally, modularity of innermost termination is proved for nice extensions as follows:

1. given a base system $\mathcal{R}_0$ and an extension $\mathcal{R}_1$, a set $\mathcal{S}_{\mathcal{R}_0 \cup \mathcal{R}_1}$ of terms that contains the right hand sides of the nice extension $\mathcal{R}_0 \cup \mathcal{R}_1$ is identified,

2. innermost termination of a nice extension is proved to be equivalent to innermost termination over the set $\mathcal{S}_{\mathcal{R}_0 \cup \mathcal{R}_1}$,

3. a nice extension is split into two TRSs belonging to an even smaller class of modular combinations, called *crosswise independent unions*, and modularity of innermost termination for crosswise independent unions is proved,

4. finally, a commutation result on two specialized relations obtained from $\mathcal{S}_{\mathcal{R}_0 \cup \mathcal{R}_1}$ and the modularity of innermost termination for crosswise independent unions are used to prove innermost termination over the set $\mathcal{S}_{\mathcal{R}_0 \cup \mathcal{R}_1}$.

For our proof of modularity of basic narrowing termination, we consider generalized relaxed proper extensions and *generalized relaxed nice extensions* (GRNE), two less restrictive versions of generalized proper extensions and *generalized nice extensions* (GNE), respectively. In Section 5.3, for proving the termination of basic narrowing in generalized relaxed proper extensions, a similar decomposition scheme based on generalized relaxed nice extensions is applied. Then, our proof of modularity of basic narrowing termination for generalized relaxed nice extensions proceeds as follows:
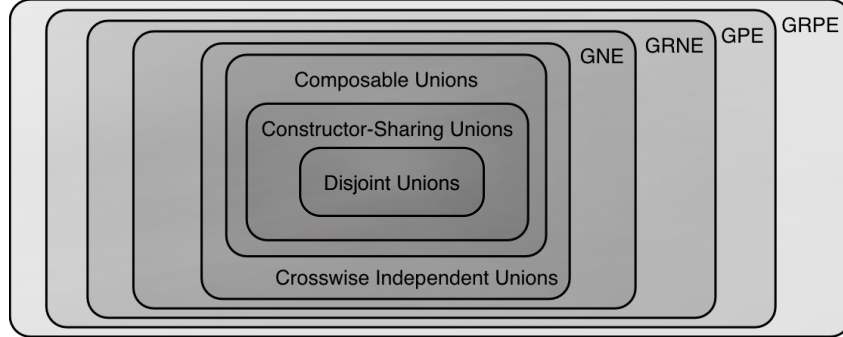
Figure 3: More modular combinations

1. given a base system $\mathcal{R}_0$ and an extension $\mathcal{R}_1$, a set $\mathcal{S}^{\mathsf{rs-rnf}}_{\mathcal{R}_0 \cup \mathcal{R}_1}$ of terms that contains the right hand sides of the generalized relaxed nice extension $\mathcal{R}_0 \cup \mathcal{R}_1$ is identified (Section 5.2),

2. the equivalence between basic narrowing termination for generalized relaxed nice extensions and basic narrowing termination over the set $\mathcal{S}^{\mathsf{rs-rnf}}_{\mathcal{R}_0 \cup \mathcal{R}_1}$ follows from Theorem 1 (Section 3),

3. a generalized relaxed nice extension is split into two crosswise independent systems (Section 5.2) and modularity of basic narrowing termination for crosswise independent unions is proved (Section 5.1),

4. finally, the commutation result for basic narrowing of Proposition 3 (Section 3) is applied to two specialized relations obtained from $\mathcal{S}^{\mathsf{rs-rnf}}_{\mathcal{R}_0 \cup \mathcal{R}_1}$ and the modularity of basic narrowing termination for crosswise independent unions is used to prove basic narrowing termination over the set $\mathcal{S}^{\mathsf{rs-rnf}}_{\mathcal{R}_0 \cup \mathcal{R}_1}$ (Section 5.2).

Figure 3 shows the different modular combinations studied in this section.

Let us first introduce the auxiliary notion of functional dependency. The function f depends on g, in symbols $f \trianglerighteq g$, if the evaluation of f involves a call to g after one or more rewrite steps.

Definition 7 (Dependency relation $\trianglerighteq_{\mathcal{R}}$) [36] *For a TRS $(\mathcal{D} \uplus \mathcal{C}, R)$ the dependency relation $\trianglerighteq_{\mathcal{R}}$ is the smallest preorder satisfying the condition $f \trianglerighteq_{\mathcal{R}} g$ whenever there is a rewrite rule $f(s_1, \ldots, s_n) \to r \in R$ and $g(t_1, \ldots, t_n)$ is a subterm of r, with $g \in \mathcal{D}$.*

We often omit $\mathcal{R}$ when it is clear from the context. We say that a symbol $f \in \mathcal{D}$ depends on a symbol $g \in \mathcal{D}$ if $f \trianglerighteq g$.

Definition 8 (Split) [36] *Let $(\mathcal{D} \uplus \mathcal{C}, R)$ be a GHC of a base system $\mathcal{R}_0 = (\mathcal{D}_0 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_0, R_0)$ with an extension $\mathcal{R}_1 = (\mathcal{D}_1 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_1, R_1)$. The set $\mathcal{D}_1$ of defined symbols of $\mathcal{R}_1$ is split into two sets $\mathcal{D}_1^0$ and $\mathcal{D}_1^1$ where*
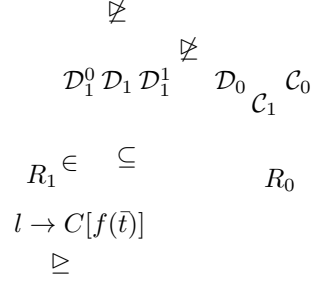
$$\ntriangleright$$
$$\ntriangleright$$
$$\mathcal{D}_1^0\ \mathcal{D}_1\ \mathcal{D}_1^1 \qquad \mathcal{D}_0 \quad \mathcal{C}_0$$
$$\mathcal{C}_1$$
$$R_1 \quad {}^{\in} \qquad {}^{\subseteq} \qquad\qquad R_0$$
$$l \to C[f(\bar{t})]$$
$$\trianglerighteq$$

Figure 4: Proper extension

- $\mathcal{D}_1^0 = \{f \in \mathcal{D}_1 \mid \exists g \in \mathcal{D}_0, f \trianglerighteq_\mathcal{R} g\}$

- $\mathcal{D}_1^1 = \mathcal{D}_1 \setminus \mathcal{D}_1^0.$

That is, $\mathcal{D}_1^0$ contains all the functions from $\mathcal{D}_1$ which depend on functions from $\mathcal{R}_0$, and $\mathcal{D}_1^1$ contains those which do not. We also split R_1 into two subsystems R_1^0 and R_1^1:

- $R_1^0 = \{l \to r \in R_1 \mid root(l) \in \mathcal{D}_1^0\}$

- $R_1^1 = \{l \to r \in R_1 \mid root(l) \in \mathcal{D}_1^1\}.$

We are now ready to introduce the class of generalized proper extensions.

Definition 9 (Generalized Proper Extension(GPE)) [36] *Let $\mathcal{R} = \mathcal{R}_0 \cup \mathcal{R}_1$ be a GHC of a base system $(\mathcal{D}_0 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_0, R_0)$ and the extension $(\mathcal{D}_1 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_1, R_1)$. Define the sets $\mathcal{D}_1^0, \mathcal{D}_1^1, \mathcal{R}_1^0$ and $\mathcal{R}_1^1$ as in Definition 8. $\mathcal{R}_1$ is a generalized proper extension (GPE) of $\mathcal{R}_0$ iff $R_{sh} \subset R_1^1$ and every rewrite rule $l \to r \in \mathcal{R}_1^0$ satisfies that, for every subterm t of r such that $root(t) \in \mathcal{D}_1^0$ and $root(t) \trianglerighteq_\mathcal{R} root(l)$, t contains no function symbol from $\mathcal{D}_0 \cup \mathcal{D}_1^0$ strictly below its root.*

By abusing notation, we often say that $\mathcal{R}_1 \cup \mathcal{R}_0$ is a GPE to mean that it is a GHC where $\mathcal{R}_1$ is a GPE of $\mathcal{R}_0$. This applies not only to GPEs but also to the rest of restrictions of (generalized) hierarchical combinations defined from now on. Figure 4 illustrates the notion of *proper extension*, i.e., a generalized proper extension without a common subsystem. The following two examples illustrate the notion of (generalized) proper extensions.

Example 4 *Consider computing the factorial of a number in tail recursive style.*

$$\mathcal{R}_! = \left\{ \begin{array}{rcl} \mathtt{fact}(x) & \to & \mathtt{factacc}(x, 1) \\ \mathtt{factacc}(0, x) & \to & x \\ \mathtt{factacc}(\mathtt{s}(y), x) & \to & \mathtt{factacc}(y, x * \mathtt{s}(y)) \end{array} \right.$$

17

$$\mathcal{R}_* = \begin{cases} 0 * y & \to & 0 \\ \mathtt{s}(x) * y & \to & (x * y) + y \end{cases}$$

$\mathcal{R}_!$ *is a hierarchical extension of* $\mathcal{R}_*$*, but it is not a GPE, because of the call to* $*$*, which is defined in the base* $\mathcal{R}_*$*, in the right hand side of the 3rd rule. Below a recursive call, nesting calls to the base system are forbidden by the proper restriction.*

On the other hand, a direct, non tail recursive definition of factorial is a GPE.

Example 5 *Consider computing the factorial of a number.*

$$\mathcal{R}_! = \begin{cases} \mathtt{fact}(0) & \to & 1 \\ \mathtt{fact}(s(x)) & \to & \mathtt{s}(x) * \mathtt{fact}(x) \end{cases} \qquad \mathcal{R}_* = \begin{cases} 0 * y & \to & 0 \\ \mathtt{s}(x) * y & \to & (x * y) + y \end{cases}$$

$\mathcal{R}_!$ *is a hierarchical extension of* $\mathcal{R}_*$ *and it is a GPE, since there are no nesting calls to the base below a recursive call in a right hand side of the extension.*

To understand why non–proper extensions can be troublesome for the modularity of ($\overset{b}{\rightsquigarrow}$)-termination (and SIN), consider the following example.

Example 6 *Consider the following TRSs, whose combination is hierarchical but not proper since there is a call to b in the one rule of* $\mathcal{R}_1$*:*

$$\mathcal{R}_1 : \{ \mathtt{f(a)} \to \mathtt{f(b)} \} \qquad\qquad \mathcal{R}_0 : \{ \mathtt{b} \to \mathtt{a} \}$$

There exists the following infinite basic narrowing derivation

$$\mathtt{f(a)} \overset{b}{\rightsquigarrow} \mathtt{f(b)} \overset{b}{\rightsquigarrow} \mathtt{f(a)} \overset{b}{\rightsquigarrow} \cdots$$

produced by the nesting of a redex w.r.t. $\mathcal{R}_0$ *inside the recursive call to* $\mathtt{f}$ *in the rhs of the rule of* $\mathcal{R}_1$*.*

Let us now introduce the main idea behind our *relaxed* generalization of GPEs by means of the following example.

Example 7 *Consider the following TRS, an encoding of the exponentiation* x^y *with group axioms that are commonly used in the specification of many cryptographic protocols [10, 11], where the constructor symbol* g *is used as a generator for the exponentiation.*

$$\mathcal{R}_1 : \quad \mathrm{exp}(\mathrm{exp}(\mathtt{g}, X), Y) \to \mathrm{exp}(\mathtt{g}, X * Y)$$

$$\mathcal{R}_0 : \quad X * X^{-1} \to 1 \qquad X * 1 \to X \qquad 1 * X \to X$$

Basic narrowing trivially terminates on each system separately, since every rhs is unnarrowable. However, $\mathcal{R}_1$ *is not a GPE of* $\mathcal{R}_0$*, and even so it is easy to see that basic narrowing does terminate in the union* $\mathcal{R}_1 \cup \mathcal{R}_0$*. The reason is that the outer function symbol* exp *in the recursive invocation occurring in the right–hand side of the rule of* $\mathcal{R}_1$ *is blocked forever, and therefore cannot cause non-termination. Our notion of generalized* relaxed *proper extension captures this idea.*

$$\not\trianglerighteq$$

$$\not\trianglerighteq$$

$$\mathcal{D}_1^0\ \mathcal{D}_1\ \mathcal{D}_1^1 \qquad \mathcal{D}_0 \qquad \mathcal{C}_0$$
$$\mathcal{C}_1$$

$$R_1\ {\in} \qquad {\subseteq} \qquad\qquad R_0$$

$$l \to C[\underbrace{f(\bar{t})}]$$
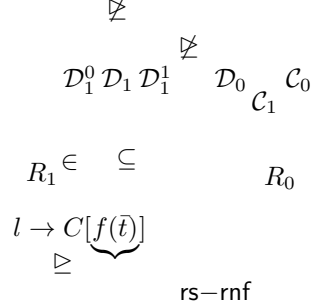$$\trianglerighteq$$

$$\mathsf{rs-rnf}$$

Figure 5: Relaxed proper extension

We recall the notion of *root-stable rigid normal form* introduced in [3], which lifts to narrowing the standard concept of head normal form. By abusing terminology, in contrast to the definition of [3], here we consider basic narrowing derivations instead of narrowing derivations within the following definition.

Definition 10 (Root-Stable Rigid Normal Form) [3] *A term s is a root-stable rigid normal form (rs−rnf) w.r.t. $\mathcal{R}$ iff either s is a variable or there are no substitutions θ and θ' and terms s' and s'' s.t. $s\theta \xrightarrow{\geq\epsilon_*}_{\mathcal{R}} s' \overset{b}{\leadsto}_{\epsilon,\mathcal{R},\theta'} s''$.*

Roughly speaking, a term t is a root-stable rigid normal form if and only if it is a variable or if there is no possible instantiation of t which enables a (basic) narrowing step at the root after giving arbitrary rewriting steps below the root. The notion of root-stable rigid normal form is stronger than the notion of rewriting head normal form, i.e., every root-stable rigid normal form is also a head normal form, but the opposite is not true. Constructor terms as well as ground normal forms are trivial cases of root-stable rigid normal forms.

We are ready now to formalize the notion of generalized relaxed proper extensions.

Definition 11 (Generalized Relaxed Proper Extension(GRPE)) *Let $\mathcal{R} = (\mathcal{D} \uplus \mathcal{C}, R)$ be a GHC of a base system $\mathcal{R}_0 = (\mathcal{D}_0 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_0, R_0)$ and the extension $\mathcal{R}_1 = (\mathcal{D}_1 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_1, R_1)$. Define the sets $\mathcal{D}_1^0$, $\mathcal{D}_1^1$, $\mathcal{R}_1^0$ and $\mathcal{R}_1^1$ as in Definition 8. $\mathcal{R}_1$ is a generalized relaxed proper extension (GRPE) of $\mathcal{R}_0$ iff $R_{sh} \subset R_1^1$ and every rule $l \to r$ in R_1^0 satisfies the following condition:*

(H1) *for each subterm t of r such that (a) $root(t) \in \mathcal{D}_1^0$, (b) t is not a rs−rnf, and (c) $root(t) \trianglerighteq_{\mathcal{R}} root(l)$, t does not contain a function symbol from $\mathcal{D}_0 \cup \mathcal{D}_1^0$ strictly below its root.*

Figure 5 illustrates the notion of *relaxed proper extension* rather than generalized relaxed proper extension, i.e., a generalized relaxed proper extension without a common subsystem. The reader can check that the TRS of Example 7 is a GRPE.

$$\not\trianglerighteq$$

$$\mathcal{D}_0\ \mathcal{D}_{sh}\ \mathcal{D}_1$$
$$\not\trianglerighteq$$
$$\mathcal{C}_1 \qquad \mathcal{C}_0$$
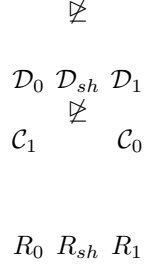
$$R_0\ R_{sh}\ R_1$$

Figure 6: Crosswise independent union

5.1 Crosswise Independent Unions

Crosswise independent unions (CIU) are a generalization of composable unions in which the non-shared defined symbols of the other system are not allowed in right–hand sides. As for composable unions, defined symbols in $\mathcal{D}_{sh}$ depend only on functions in $\mathcal{D}_{sh}$. We study them as an instrumental step for achieving the proof of modularity of basic narrowing termination for nice extensions in the next section.

Definition 12 (Crosswise Independent Union(CIU)) [36] *Two TRS's* $\mathcal{R}_0 = (\mathcal{D}_0 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_0, R_0)$ *and* $\mathcal{R}_1 = (\mathcal{D}_1 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_1, R_1)$ *are* crosswise independent *if*

(i) $R_{sh} = R_0 \cap R_1 = \{l \to r \in R_0 \cup R_1 \mid root(l) \in \mathcal{D}_{sh}\}$

(ii) *for all* $f \in \mathcal{D}_i \cup \mathcal{D}_{sh}$ *and* $g \in \mathcal{D}_{1-i}$, $i \in \{0,1\}$, *we have* $f \not\trianglerighteq_{R_0 \cup R_1} g$.

We say that $\mathcal{R} = \mathcal{R}_0 \cup \mathcal{R}_1$ *is a* crosswise independent union *(CIU) iff* $\mathcal{R}_0$ *and* $\mathcal{R}_1$ *are crosswise independent.*

Figure 6 illustrates the notion of a crosswise independent union. Let us show that ($\overset{b}{\rightsquigarrow}$)-termination is modular for CIUs.

Theorem 4 *Termination of basic narrowing is a modular property for crosswise independent unions.*

Proof. Let $\mathcal{R} = \mathcal{R}_0 \cup \mathcal{R}_1$ be the union of two crosswise independent systems. The scheme of the proof is similar to the proof of Theorem 3. Essentially, since all the derivations starting from the rhs of a rule in $\mathcal{R} = \mathcal{R}_0 \cup \mathcal{R}_1$ use only rules that belong either to $\mathcal{R}_0$ or $\mathcal{R}_1$, and are, hence, finite in both cases, by Theorem 1 every basic narrowing derivation w.r.t. $\mathcal{R}$ is finite. ∎
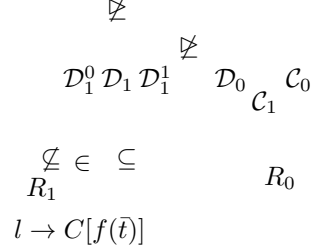
$$\not\vdash$$

$$\not\vdash$$

$$\mathcal{D}_1^0\ \mathcal{D}_1\ \mathcal{D}_1^1 \qquad \mathcal{D}_0 \quad \mathcal{C}_0$$
$$\mathcal{C}_1$$

$$\not\subseteq\ \in\ \subseteq \qquad\qquad R_0$$
$$R_1$$
$$l \to C[f(\bar{t})]$$

Figure 7: Nice extension

5.2 Nice Extensions

In this section we prove that ($\overset{b}{\leadsto}$)-termination is modular for generalized *relaxed* nice extensions, the relaxed variant of generalized nice extensions introduced by Krishna Rao in [36].

Definition 13 (Generalized Nice Extension(GNE)) [36] *Let $\mathcal{R} = (\mathcal{D} \uplus \mathcal{C}, R)$ be a GHC of a base system $\mathcal{R}_0 = (\mathcal{D}_0 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_0, R_0)$ and the extension $\mathcal{R}_1 = (\mathcal{D}_1 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_1, R_1)$. Define the sets $\mathcal{D}_1^0$, $\mathcal{D}_1^1$, $\mathcal{R}_1^0$ and $\mathcal{R}_1^1$ as in Definition 8. $\mathcal{R}_1$ is a generalized nice extension (GNE) of $\mathcal{R}_0$ iff $R_{sh} \subset R_1^1$ and every rule $l \to r \in R_1^0$ satisfies that for each subterm t of r and $root(t) \in \mathcal{D}_1^0$, t contains no function symbol from $\mathcal{D}_0 \cup \mathcal{D}_1^0$ strictly below its root.*

Example 5 is a GRNE where $\mathcal{R}_!$ is a hierarchical extension of $\mathcal{R}_*$. Figure 7 illustrates a *nice extension* (i.e., a GNE with no shared subsystem).

Note that every GNE is also a GPE by construction. GNEs are a useful restriction of GPEs, because it can be shown that every GPE can be modelled as a *pyramid* of GNEs. The same applies to GRPEs and GRNEs. Therefore in some cases we can lift results from GRNEs to GRPEs.

Definition 14 (Generalized Relaxed Nice Extension(GRNE)) *Let $\mathcal{R} = (\mathcal{D} \uplus \mathcal{C}, R)$ be a GHC of a base system $\mathcal{R}_0 = (\mathcal{D}_0 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_0, R_0)$ and the extension $\mathcal{R}_1 = (\mathcal{D}_1 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_1, R_1)$. Define the sets $\mathcal{D}_1^0$, $\mathcal{D}_1^1$, $\mathcal{R}_1^0$ and $\mathcal{R}_1^1$ as in Definition 8. $\mathcal{R}_1$ is a generalized relaxed nice extension (GRNE) of $\mathcal{R}_0$ iff $R_{sh} \subset R_1^1$ and every rule $l \to r \in R_1^0$ satisfies the following condition:*

(N1) *for each subterm t of r such that t is not a rs−rnf and $root(t) \in \mathcal{D}_1^0$, t contains no function symbol from $\mathcal{D}_0 \cup \mathcal{D}_1^0$ strictly below its root.*

Example 7 shows two systems $\mathcal{R}_1$ and $\mathcal{R}_0$ where $\mathcal{R}_1$ is a GRNE of $\mathcal{R}_0$ but not a GNE nor a GPE.

In a hierarchical combination $\mathcal{R}_0 \cup \mathcal{R}_1$ of a base system $\mathcal{R}_0$ and a GRNE $\mathcal{R}_1$, it is not allowed in the right–hand sides of the extension $\mathcal{R}_1$ to nest calls

to functions which depend on the base system $\mathcal{R}_0$, unless the subterm in the right–hand side of $\mathcal{R}_1$ under consideration is a root-stable rigid normal form. This contrasts with the less restricted right–hand sides of proper extensions, where one is allowed to nest function calls that are non rigid and dependent on $\mathcal{R}_0$ as long as they are not mutually recursive.

Following Rao's proof scheme, we prove the result that basic narrowing derivations starting from a special set $\mathcal{S}^{\mathsf{rs-rnf}}_{\mathcal{R}_0 \cup \mathcal{R}_1}$ of terms are finite. Let us recall the standard notion of context here. A context is a term C with zero or more 'holes', i.e., the fresh constant symbol $\square$. If C is a context with k holes and $\bar{t}$ a list of k terms, $C[\bar{t}]$ denotes the result of replacing the k holes in C by the terms in $\bar{t}$.

Definition 15 ($\mathcal{S}^{\mathsf{rs-rnf}}_{\mathcal{R}_0 \cup \mathcal{R}_1}$ terms) *Let $\mathcal{R} = (\mathcal{D} \uplus \mathcal{C}, R)$ be a GRNE of a base system $\mathcal{R}_0 = (\mathcal{D}_0 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_0, R_0)$ and the extension $\mathcal{R}_1 = (\mathcal{D}_1 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_1, R_1)$. Define the sets $\mathcal{D}_1^0$, $\mathcal{D}_1^1$, R_1^0 and R_1^1 as in Definition 8. We define $\mathcal{S}^{\mathsf{rs-rnf}}_{\mathcal{R}_0 \cup \mathcal{R}_1}$ as the set of all terms of the form $C[s_1, \ldots, s_n]$, where C is a context in $(\mathcal{D} \cup \mathcal{C})$ and the following conditions hold:*

1. *every subterm s of C with $root(s) \in \mathcal{D}_1^0$ is a $\mathsf{rs-rnf}$ in $\mathcal{R}$.*

2. *for all i, $root(s_i) \in \mathcal{D}_1^0$.*

3. *for all i, s_i is not a $\mathsf{rs-rnf}$ in $\mathcal{R}$.*

4. *s_i contains no function symbol from $\mathcal{D}_0 \cup \mathcal{D}_1^0$ strictly below its root.*

The reader can check that the right–hand sides of the rules in a GRNE belong to the corresponding set of $\mathcal{S}^{\mathsf{rs-rnf}}_{\mathcal{R}_0 \cup \mathcal{R}_1}$.

By definition, $\mathcal{S}^{\mathsf{rs-rnf}}_{\mathcal{R}_0 \cup \mathcal{R}_1}$ terms have the property that no $\mathcal{R}_1^0$ reduction step is possible within the context C. Also, when we consider the skeleton–environment formulation of basic narrowing, the set $\mathcal{S}^{\mathsf{rs-rnf}}_{\mathcal{R}_0 \cup \mathcal{R}_1}$ is skeleton-closed under $\overset{b}{\leadsto}_{\mathcal{R}_0 \cup \mathcal{R}_1}$ if $\mathcal{R} = \mathcal{R}_0 \cup \mathcal{R}_1$ is a GRNE. Formally,

Lemma 3 *If $t \in \mathcal{S}^{\mathsf{rs-rnf}}_{\mathcal{R}_0 \cup \mathcal{R}_1}$ and either $\langle t, \theta \rangle \overset{b}{\leadsto}_{\mathcal{R}_1, \sigma} \langle t', \theta\sigma \rangle$ or $\langle t, \theta \rangle \overset{b}{\leadsto}_{\mathcal{R}_0, \sigma} \langle t', \theta\sigma \rangle$, then $t' \in \mathcal{S}^{\mathsf{rs-rnf}}_{\mathcal{R}_0 \cup \mathcal{R}_1}$.*

Proof. First we consider the closedness of $\mathcal{R}_1$. Let $t = C[s_1, \ldots, s_n]$, and let $l \to r \in R_1$ be the rule applied in the basic narrowing step. We consider two cases depending on whether $root(l)$ is in $\mathcal{D}_1^0$ or not.

(a) $root(l) \notin \mathcal{D}_1^0$. That is, $root(l) \in (\mathcal{D}_1^1 \cup \mathcal{D}_{sh})$. There are two subcases:

1. The step took place in C. By definition, no $\mathcal{D}_1^0$ symbol occurs in r, hence t' has the form $C'[t_1, \ldots, t_m]$, where each t_i is a subterm of some s_j, $root(t_i) \in \mathcal{D}_1^0$, and no reduction is possible in R_1^0 at a position within the context C'. Since it is always possible to cast t' in this form, the lemma holds.

2. The step took place in a proper subterm of some s_i. By definition $root(r) \in \mathcal{D}_1^1$ and t' has the form $C[s_1, \ldots, s'_j, \ldots, s_n]$, where $s'_j = s_j[r]$ for some position $p > \varepsilon$. Since no D_1^0 symbol occurs in r, s'_j satisfies all the conditions of Definition 15 and $t' \in \mathcal{S}_{\mathcal{R}_0 \cup \mathcal{R}_1}^{\mathsf{rs-rnf}}$.

(b) $root(l) \in \mathcal{D}_1^0$. Therefore, the reduction happened at the root of some s_i, since

- any ocurrence of a $\mathcal{D}_1^0$ symbol in C is a $\mathsf{rs-rnf}$ and hence by definition not narrowable.

- s_i itself is not a $\mathsf{rs-rnf}$ and hence there exists some instance $s_i\theta$ which is eventually narrowable.

- there are no $\mathcal{D}_1^0$ function symbols below the root of s_i, hence the reduction cannot take place in a proper subterm of s_i.

Let $r = C'[u_1, \ldots, u_m]$; note it is by definition a term in $\mathcal{S}_{\mathcal{R}_0 \cup \mathcal{R}_1}^{\mathsf{rs-rnf}}$. Then, t' has the form $C[s_1, \ldots, s_{i-1}, r, s_{i+1}, \ldots, s_n]$, and can be written as $C''[s_1, \ldots, s_{i-1}, u_1, \ldots, u_m, s_{i+1}, \ldots, s_n]$, where C'' is the context resulting of *appropriately* joining C and C' so that any subterm r of C'' with $root(r) \in \mathcal{D}_1^0$ is a $\mathsf{rs-rnf}$. This means that t' satisfies all the conditions of Definition 15, and the lemma holds.

Now let us consider the closedness of $\mathcal{R}_0$. Let $t = C[s_1, \ldots, s_n]$, and $l \to r \in R_0$ be the rule applied in the basic narrowing step. The reduction takes place in C, and since no symbol in $\mathcal{D}_1$ occurs in r, we can write t' as $C'[t_1, \ldots, t_m]$ where

- C' is the context resulting of replacing a subterm in C with r. By construction, every subterm s in C' with $root(s) \in \mathcal{D}_1^0$ was also in C (since r couldn't introduce it) and therefore it is by assumption a $\mathsf{rs-rnf}$.

- each t_i is a subterm of some s_j with the following properties

  - $root(t_i) \in \mathcal{D}_1^0$
  - t_i is not a $\mathsf{rs-rnf}$

- since t_i is a subterm of s_i, the following holds: t_i contains no function symbol in $\mathcal{D}_0 \cup \mathcal{D}_1^0$ strictly below its root

This concludes the proof. ∎

Having defined and proved some properties about the set $\mathcal{S}_{\mathcal{R}_0 \cup \mathcal{R}_1}^{\mathsf{rs-rnf}}$, we are now ready to consider ($\overset{b}{\leadsto}$)-termination. Concretely, we show that the union of two systems which are ($\overset{b}{\leadsto}$)-terminating over $\mathcal{S}_{\mathcal{R}_0 \cup \mathcal{R}_1}^{\mathsf{rs-rnf}}$ is itself ($\overset{b}{\leadsto}$)-terminating. We prove the result by using Proposition 3 given in Section 3 and some general results on quasi–commutation of abstract relations, which we introduce in the following.

Definition 16 (Abstract Reduction System) [8] *An abstract reduction system (ARS) is a structure $\mathcal{A} = (A, \{\to_\alpha | \ \alpha \in I\})$ consisting of a set A and a set of binary relations $\to_\alpha$ on A, indexed by a set I. We write $(A, \to_1, \to_2)$ instead of $(A, \{\to_\alpha | \ \alpha \in \{1, 2\}\})$.*

Definition 17 (Quasi-commutation) [8] *Let $\to_0$ and $\to_1$ be two relations on a set S. The relation $\to_1$ quasi-commutes over $\to_0$ if, for all $s, u, t \in S$ s.t. $s \to_0 u \to_1 t$, there exists $v \in S$ s.t. $s \to_1 v \to_{01}^* t$, where $\to_{01}^*$ is the transitive-reflexive closure of $\to_0 \cup \to_1$.*

Theorem 5 [8] *If the relations $\to_0$ and $\to_1$ in the $ARS(S, \to_0, \to_1)$ are terminating and $\to_1$ quasi-commutes over $\to_0$, then the relation $\to_0 \cup \to_1$ is terminating too.*

We now define an ARS with skeleton–environment tuples as elements, where the skeletons come from the set $\mathcal{S}_{\mathcal{R}_0 \cup \mathcal{R}_1}^{\mathsf{rs-rnf}}$ of terms, and the relationships $\to_0$ and $\to_1$ of the ARS are restrictions of basic narrowing.

Definition 18 *Let $\mathcal{R} = \mathcal{R}_0 \cup \mathcal{R}_1$ be a hierarchical combination where $\mathcal{R}_1$ is a GRNE of $\mathcal{R}_0$. Define the sets $\mathcal{D}_1^0$, $\mathcal{D}_1^1$, R_1^0 and R_1^1 as in Definition 8. We define the ARS $\mathcal{A}(\mathcal{R}_0, \mathcal{R}_1) = (\mathcal{S}_{\mathcal{R}_0 \cup \mathcal{R}_1}^{\mathsf{rs-rnf}} \times Subst, \to_0, \to_1)$, where the relations $\to_0$ and $\to_1$ are defined as follows. Let $s = C[s_0, \dots, s_n]$ be a term in $\mathcal{S}_{\mathcal{R}_0 \cup \mathcal{R}_1}^{\mathsf{rs-rnf}}$. Then*

1. *$\langle C[s_0, \dots, s_n], \sigma \rangle \to_0 \langle C'[u_0, \dots, u_k], \theta\sigma \rangle$ if $\langle C[s_0, \dots, s_n], \sigma \rangle \overset{b}{\leadsto}_{\mathcal{R}_0 \cup \mathcal{R}_1^1, \theta}$ $\langle C'[u_0, \dots, u_k], \sigma\theta \rangle$ is a basic narrowing step given within the context C.*

2. *$\langle C[s_0, \dots, s_n], \sigma \rangle \to_1 \langle C[s_0, \dots, s_{i-1}, s_i', s_{i+1}, \dots, s_n], \theta\sigma \rangle$ if $\langle C[s_0, \dots, s_n], \sigma \rangle$ $\overset{b}{\leadsto}_{\mathcal{R}_1, \theta} \langle C[s_0, \dots, s_{i-1}, s_i', s_{i+1}, \dots, s_n], \theta\sigma \rangle$ is a basic narrowing step given at a subterm s_i, with $i \in [0, \dots, n]$.*

The relation $\to_1 \cup \to_0$ is exactly the basic narrowing relation over $\mathcal{S}_{\mathcal{R}_0 \cup \mathcal{R}_1}^{\mathsf{rs-rnf}}$. In the following we establish that both $\to_0$ and $\to_1$ are terminating relations.

Lemma 4 *Given the ARS $\mathcal{A}(\mathcal{R}_0, \mathcal{R}_1)$ of Definition 18, the relations $\to_0$ and $\to_1$ are terminating if $\mathcal{R}_0$ and $\mathcal{R}_1$ are $(\overset{b}{\leadsto})$-terminating.*

Proof. The relation $\to_1$ is a subrelation of $\overset{b}{\leadsto}_{\mathcal{R}_1}$, and hence terminating.

On the other hand, $\mathcal{R}_0$ and $\mathcal{R}_1^1$ are crosswise independent: condition (i) in Definition 12 is satisfied by construction, and condition (ii) is satisfied since $\mathcal{R}_1^1 \setminus \mathcal{R}_{sh}$ does not depend on $\mathcal{R}_0$ by definition, and neither does $\mathcal{R}_{sh}$ (as $\mathcal{R}_{sh} \subset \mathcal{R}_1^1$ by Definition 14). Hence, their union is terminating by Theorem 4.

It should be obvious from the definition of $\mathcal{A}$ that $\to_0$ can be seen as a restriction of the system $\mathcal{R}_0 \cup \mathcal{R}_1^1$ to the terms in $\mathcal{S}_{\mathcal{R}_0 \cup \mathcal{R}_1}^{\mathsf{rs-rnf}}$, and therefore termination of $\to_0$ follows. ∎

We are now in a position to prove the quasi-commutation of the relation $\to_1$ over the relation $\to_0$ in the ARS $\mathcal{A}(\mathcal{R}_0, \mathcal{R}_1)$. The proof of this result relies on Proposition 3.

Theorem 6 *Given the ARS $\mathcal{A}(\mathcal{R}_0,\mathcal{R}_1)$ of Definition 18, the relation $\to_1$ quasi-commutes over the relation $\to_0$.*

Proof. We have to show that

$$\forall s, u, t \in \mathcal{S}^{\mathsf{rs-rnf}}_{\mathcal{R}_0 \cup \mathcal{R}_1} \ s.t. \ s \xrightarrow{p}_0 u \xrightarrow{q}_1 t, \ \exists v \in \mathcal{S}^{\mathsf{rs-rnf}}_{\mathcal{R}_0 \cup \mathcal{R}_1}, p' \in \mathcal{P}os_\Sigma(s) \ s.t. \ s \xrightarrow{p'}_1 v \to^*_{01} t \tag{4}$$

where, by abusing notation, p, q and p' refer to positions that are in the skeleton part of the tuples denoted by s, u, t and v.

As $\to_0 \subseteq \to_{01}$ and $\to_1 \subseteq \to_{01}$, we can equivalently see these as $\to_{01}$ derivations. And now, since $\to_{01}$ is a basic narrowing relation, we can use Proposition 3 to prove the result. We only need to show that q admits an antecedent in s. It is easy to see that this is satisfied, because $\to_0$ cannot create new $\to_1$ redexes, which means that redex $u|_q$ was already there in $s|_q$. On the other hand, since no redex can be propagated by basic narrowing, p and q must be disjoint. Applying Proposition 3, we have:

$$\forall s, u, t \in \mathcal{S}^{\mathsf{rs-rnf}}_{\mathcal{R}_0 \cup \mathcal{R}_1} \ s.t. \ s \xrightarrow{p}_{01} u \xrightarrow{q}_{01} t, \ \exists v \in \mathcal{S}^{\mathsf{rs-rnf}}_{\mathcal{R}_0 \cup \mathcal{R}_1} \ s.t. \ s \xrightarrow{q}_{01} v \xrightarrow{p}_{01} t \tag{5}$$

from which we observe that (1) $p' = q$, (2) t is reached in a single $\to_{01}$ step, and (3) the step $s \to_{01} v$ is indeed given in $\to_1$, which suffices to prove the result. ∎

We only need to combine Theorem 5 and Theorem 6 to obtain the desired result on basic narrowing termination of $\mathcal{S}^{\mathsf{rs-rnf}}_{\mathcal{R}_0 \cup \mathcal{R}_1}$ terms.

Corollary 2 *Let $\mathcal{R}_1$ and $\mathcal{R}_0$ be two $(\overset{b}{\rightsquigarrow})$-terminating systems where $\mathcal{R}_1$ is a GRNE over $\mathcal{R}_0$. Let $\mathcal{S}^{\mathsf{rs-rnf}}_{\mathcal{R}_0 \cup \mathcal{R}_1}$ be a set of terms constructed following definition 15. Then every basic narrowing derivation in $\mathcal{R}_0 \cup \mathcal{R}_1$ starting from a term of $\mathcal{S}^{\mathsf{rs-rnf}}_{\mathcal{R}_0 \cup \mathcal{R}_1}$ terminates.*

And now by Theorem 1, we can derive the modularity of $(\overset{b}{\rightsquigarrow})$-termination in generalized relaxed nice extensions.

Corollary 3 *Termination of basic narrowing is a modular property for generalized relaxed nice extensions.*

5.3 Proper Extensions

We now study the connection between GRNEs and GRPEs. In the following, we show that it is possible to model any finite GRPE as a finite pyramid of one or more GRNEs. Essentially, the idea is similar to the modular decomposition of a TRS given in [43]. A given GRPE is reduced to the canonical modular form, a modular partition such that each of the individual modules cannot be split up. In order to achieve this we employ the graph induced by the dependency relation $\trianglerighteq$ on defined function symbols, and the rules corresponding to the symbols of every strongly connected component become a module (i.e., a GRNE).

In order to prove the main result of this section, we first need two auxiliary definitions and one proposition.

Definition 19 (Equivalence relation $\approx$ and partial order $\sqsupset$) [36] *Let $(\mathcal{D} \uplus \mathcal{C}, R)$ be a generalized hierarchical combination of a base system $(\mathcal{D}_0 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_0, R_0)$ and the extension $(\mathcal{D}_1 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_1, R_1)$. Define the sets $\mathcal{D}_1^0$, $\mathcal{D}_1^1$, $\mathcal{R}_1^0$ and $\mathcal{R}_1^1$ as in Definition 8. We define an equivalence relation $\approx_\mathcal{R}$ from the dependency relation $\unrhd_\mathcal{R}$, where the equivalence class containing f is denoted by $[f]_\mathcal{R}$, and a partial ordering $\sqsupset_\mathcal{R}$ on the set of equivalence classes:*

- *$f \approx_\mathcal{R} g$ iff $f \unrhd_\mathcal{R} g$ and $g \unrhd_\mathcal{R} f$, where $f, g \in \mathcal{D}_1^0$.*

- *$[f]_\mathcal{R} \sqsupset_\mathcal{R} [g]_\mathcal{R}$ iff $f \unrhd_\mathcal{R} g$ and $g \ntrianglerighteq_\mathcal{R} f$.*

Since the signature of any TRS is a countable set, the equivalence relation $\approx$ partitions $\mathcal{D}_1^0$ into a countable set E of equivalence classes. Provided that the ordering $\sqsupset$ is noetherian, it can be extended to a well-ordering of type λ, where λ is a countable ordinal.

Definition 20 [36] *Let $(\mathcal{D} \uplus \mathcal{C}, R)$ be a GHC of a base system $(\mathcal{D}_0 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_0, R_0)$ and the extension $(\mathcal{D}_1 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_1, R_1)$. Define the sets $\mathcal{D}_1^0$, $\mathcal{D}_1^1$, $\mathcal{R}_1^0$ and $\mathcal{R}_1^1$ as in Definition 8. For any ordinal α we denote the α-th element in the above well-ordering by E_α (if $\alpha > \lambda$ then $E_\alpha = \varnothing$), and the rules defining its elements as $\mathcal{R}_{E_\alpha} = \{l \to r \in \mathcal{R}_1 \mid root(l) \in E_\alpha\}$. We define the TRS $X_\alpha = \mathcal{R}_{sh} \cup \mathcal{R}_1^1 \cup \mathcal{R}_{E_\alpha}$ and the combined system $S_\alpha = \mathcal{R}_0 \cup (\bigcup_{\beta < \alpha} X_\beta)$. S_0 is $\mathcal{R}_0$ and S_k for $k > \lambda$ is $\mathcal{R}_0 \cup \mathcal{R}_1$.*

The following result establishes the precise relation between GRPEs and GRNEs.

Proposition 5 *Let $(\mathcal{D} \uplus \mathcal{C}, R)$ be a GRPE of a base system $(\mathcal{D}_0 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_0, R_0)$ and the extension $(\mathcal{D}_1 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_1, R_1)$. Define the sets $\mathcal{D}_1^0$, $\mathcal{D}_1^1$, $\mathcal{R}_1^0$ and $\mathcal{R}_1^1$ as in Definition 8. If the relation $\sqsupset_\mathcal{R}$ is noetherian, then X_α is a GRNE of S_α for every ordinal α, where S_α and X_α are defined as in Definition 20.*

Proof. As $\mathcal{R}_1$ is a GRPE of $\mathcal{R}_0$, we know the shared system R_{sh} is a subset of R_1^1 and hence by construction the first condition in Definition 14 is satisfied for any α. We have to show that (N1) holds in X_α. That is, for every $l \to r \in X_\alpha$, if s is a subterm of r s.t. $root(s) \in E_\alpha$ and there is a subterm u of s that contains a defined symbol depending on $Def(S_\alpha) - Def(X_\alpha)$, then s must be a rs−rnf. Since $root(s) \in E_\alpha$, it follows that $root(l) \in E_\alpha$ and, by definition, $root(l) \in \mathcal{D}_1^0$ and $root(s) \approx root(l)$. Therefore $root(s) \unrhd root(l)$. Now, since $\mathcal{R}_1$ is a GRPE and $root(s) \in D_1^0$, the following holds by (H1): if s contains a defined symbol depending on $\mathcal{D}_0$, then s is a rs−rnf. Finally, since $Def(S_\alpha) - Def(X_\alpha) \subset \mathcal{D}_0 \cup \mathcal{D}_1^0$, X_α is a GRNE of S_α. $\blacksquare$

Now, we are ready to establish how GRNEs relate to GRPEs.

Theorem 7 *Let $\mathcal{R}_1$ be a finite TRS s.t. it is a GRPE of $\mathcal{R}_0$. $\mathcal{R}_1$ can be seen as a finite pyramid of GRNEs.*

Proof. By Proposition 5, assuming that the relation $\sqsupset_{\mathcal{R}_0 \cup \mathcal{R}_1}$ is noetherian. ∎

Finally, we are able to prove the main result of the paper regarding basic narrowing termination, which follows directly from Corollary 3 and Theorem 7.

Corollary 4 *Termination of basic narrowing is a modular property for finite generalized relaxed proper extensions.*

6   Combining Theories for E-unification

Unification of terms with respect to an equational theory E is the following problem [34, 40]:

> Given a set of equations $\Gamma = \{s_1 = t_1, \ldots, s_n = t_n\}_E$, find all substitutions σ (called *solutions*) such that all equations are solved with respect to E, i.e., E implies $s_i\sigma = t_i\sigma$ for all i.

Usually, one is only interested in finding a representation of all solutions. One important case of E-unification problem is when the equational theory E can be represented by a canonical set of rewrite rules, where basic narrowing is complete as an equational unification procedure [24]: informally, for every (normalized) solution σ, a more general substitution is computed by narrowing.

Modularity of unification is a central problem in automated deduction in equational theories that are presented by a finite TRS. In particular, it is known that any disjoint combination of theories where basic narrowing terminates is of unification–type finitary [39], which means that equational unification is decidable in the combined theory (provided this theory additionally satisfies the conditions for the completeness of basic narrowing as a unification procedure).

For the best of our knowledge, modularity of unification in hierarchical combinations of theories has not been previously studied in the related literature. Without loss of generality, we restrict ourselves to the union of two equational theories. The results for the union of several theories can be derived in the obvious way.

The original completeness result proved by Hullot for canonical TRSs was later generalized in [44] by Yamamoto to *weakly innermost normalizing*[5], *weakly canonical*[6] TRSs. Yamamoto also conjectured that weak canonicity would suffice for completeness, but it was refuted by a counterexample in Middeldorp and Hamoen [29].

We summarize the existing completeness results for basic narrowing below. But first let us recall the notions of critical pair and orthogonal system. Two (possibly renamed) rules $l \to r$ and $l' \to r'$ *overlap* if there is $p \in \mathcal{P}os_\Sigma(l)$ and substitution σ such that $l|_p\sigma = l'\sigma$. The pair $\langle l\sigma[r'\sigma]_p, r\sigma \rangle$ is called a *critical*

---

[5]A TRS is called *weakly innermost normalizing* if every term has a normal form which can be reached by means of an innermost reduction sequence.

[6]*Weakly canonical* TRSs are confluent & *weakly normalizing* TRSs. A TRS is called *weakly normalizing* if every term has a normal form. Weakly canonicity is referred to as *semicompleteness* in other works, e.g. [32].

pair. A left-linear TRS without critical pairs is called *orthogonal.* Orthogonality implies confluence [41].

Proposition 6 (Hullot [24]) *Basic narrowing is a complete E-unification algorithm for canonical TRSs.*

Proposition 7 (Yamamoto [44]) *Basic narrowing is a complete E-unification algorithm for weakly canonical, weakly innermost normalizing TRSs.*

Proposition 8 (Middeldorp Hamoen [29]) *Basic narrowing is a complete E-unification algorithm for weakly normalizing, orthogonal TRSs.*

Proposition 9 (Middeldorp Hamoen [29]) *Basic narrowing is a complete E-unification algorithm for weakly canonical, right–linear TRSs.*

Note that Proposition 7 is strictly more general than Proposition 6, since canonicity implies weak canonicity and the property of weak innermost normalization.

Weak normalization (WN) is therefore an important property for completeness. As termination of basic narrowing $(SN^{\overset{b}{\rightsquigarrow}})$ implies strong innermost normalization (SIN) and strong innermost normalization implies weak normalization, we have that termination of basic narrowing implies weak normalization:

$$SN^{\overset{b}{\rightsquigarrow}} \Rightarrow SIN \Rightarrow WN$$

Lemma 5 *If basic narrowing terminates in $\mathcal{R}$, then $\mathcal{R}$ is weakly normalizing.*

We focus now on termination of the E-unification process via basic narrowing, and its completeness. We say that a narrowing calculus is an *effective E-unification algorithm* for $\mathcal{R}$ if it provides all the solutions in $\mathcal{R}$ for a given E-unification problem or fails when there are no solutions, in a finite amount of time. The following corollary characterizes the conditions under which basic narrowing constitutes an effective E-unification algorithm.

Corollary 5 *Basic narrowing is an* effective *E-unification algorithm for $(\overset{b}{\rightsquigarrow})$-terminating confluent TRSs.*

Proof. Completeness follows now from Lemma 5, Proposition 7 and Proposition 2. ∎

It can be seen that this corollary subsumes all the completeness criteria enumerated above as a consequence of the assumption of $(\overset{b}{\rightsquigarrow})$-termination: with respect to Proposition 6, termination is dropped; regarding Proposition 7, WN and WIN are dropped; since orthogonality implies confluence, Proposition 8 is also subsumed; and right–linearity as well as WN from Proposition 9 can also be dropped when $(\overset{b}{\rightsquigarrow})$-termination holds.

We are now ready to discuss the modularity of decidability of E-unification via basic narrowing; or more precisely, the preservation of the effectiveness of basic narrowing as a E-unification algorithm in modular combinations of TRSs. Let us start with the case of composable systems.

Theorem 8 (Modular Unification in Composable Unions) *Decidability of E-unification via basic narrowing is a modular property for the union of composable ($\overset{b}{\rightsquigarrow}$)-terminating, confluent systems.*

Proof. We have shown modularity of basic narrowing termination for composable unions in Theorem 3. As for completeness, confluence is shown to be modular for weakly normalizing composable unions in [32, Theorem 5.2], and weak normalization is implied by ($\overset{b}{\rightsquigarrow}$)-termination using Lemma 5. The result follows from Corollary 5. ∎

Note that the previous result implies also the modularity of E-unification decidability in the unions of disjoint and constructor-sharing ($\overset{b}{\rightsquigarrow}$)-terminating confluent systems.

None of the classes of hierarchical combinations considered previously in this article enjoy modularity E-unification decidability without additional conditions. The following example shows that modularity of equational unification cannot be directly extended to proper extensions.

Example 8 [33, Example 8.1.3] *Consider the linear, canonical, $\mathcal{C}_\varepsilon$-terminating systems $\mathcal{R}_0$ and $\mathcal{R}_1$.*

$$\mathcal{R}_0 \;:\; \mathtt{a} \;\rightarrow\; \mathtt{b} \quad \mathcal{R}_1 \;:\; \mathtt{f(a)} \;\rightarrow\; \mathtt{c}$$

$\mathcal{R}_1$ is a proper extension of $\mathcal{R}_0$ but the combined system is not even locally confluent. We have:

$$\mathtt{c} \leftarrow \mathtt{f(a)} \rightarrow \mathtt{f(b)}$$

that is, the term $\mathtt{f(a)}$ has two different normal forms.

Under the additional restriction that the union is an overlay TRS, we prove next that decidability of equational unification is modular for GRPEs. Let us recall the standard notion of *overlay* system. A critical pair $\langle l\sigma[r'\sigma]_p, r\sigma \rangle$ is called an *overlay* if $p = \epsilon$. Given a TRS $\mathcal{R}$, a critical pair $\langle l\sigma[r'\sigma]_p, r\sigma \rangle$ is called *joinable* if there is a term u such that $l\sigma[r'\sigma]_p \rightarrow^*_\mathcal{R} u$ and $r\sigma \rightarrow^*_\mathcal{R} u$. A TRS whose critical pairs are overlays is called an *overlay TRS*. The proof reuses a number of standard results from the literature, which we include here for self-containment.

Lemma 6 (Huet's Critical Pairs Lemma [23]**)** *A term rewriting system is locally confluent iff all its critical pairs are joinable.*

Lemma 7 (Newman's Lemma [31]**)** *A terminating TRS is confluent if and only if it is locally confluent.*

Lemma 8 ([20, Theorem 3.23]**)** *Every locally confluent overlay system is terminating if and only if it is innermost terminating (SIN).*

Before formulating and proving the result for GRPEs, let us introduce an auxiliary lemma.

Lemma 9 *Local confluence is a modular property of hierarchical combinations of two systems where one is a generalized relaxed proper extension of the other and the union is an overlay system.*

Proof. This result is stated in [36, Lemma 30] for GNEs, and it can be generalized with little effort to GRPEs. A complete proof follows.

By the Critical Pairs Lemma, to show local confluence one needs to show joinability of every critical pair. Since every individual system is locally confluent, we only need to consider critical pairs resulting from the overlapping of a symbol from the extension system with a symbol from the base. As the union is an overlay system, overlapping is only possible at the topmost position, and moreover, only in rules from R_{sh} (since $\mathcal{D}_0 \cap \mathcal{D}_1 = \emptyset$). Hence, since these rules are included in both systems, the critical pairs induced are, by assumption, joinable, and by the Critical Pairs Lemma the union system is locally confluent. ∎

Now we are ready to prove that decidability of E-unification via basic narrowing is modular for GRPEs (and GPEs) if the union is an overlay system.

Theorem 9 (Modular Unification in Relaxed Proper Extensions)
Decidability of E-unification via basic narrowing is a modular property for the union of two finite $(\overset{b}{\leadsto})$-terminating, confluent systems where one is a generalized relaxed proper extension of the other and the union is an overlay system.

Proof. Local confluence in the union system follows from Lemma 9. Termination of basic narrowing in the union system follows from Corollary 4. This implies that the union system is SIN by Proposition 2, and SIN in a locally confluent overlay system implies termination by Lemma 8. Thus, confluence follows from Newman's Lemma. Finally, completeness follows from Corollary 5. ∎

Looking at Example 8 now, it can be seen that the union of $\mathcal{R}_0$ and $\mathcal{R}_1$ fails to be an overlay system and therefore the union is not confluent and hence not suitable for E-unification by using basic narrowing.

Though the decidability of equational unification is modular for proper extensions only under the additional condition of overlay system, the following subclass of proper extensions enjoys modularity in *every* case, with no additional restrictions.

Definition 21 (Generalized Restricted Proper Extension(GrtPE)) [36]
Let $\mathcal{R} = \mathcal{R}_0 \cup \mathcal{R}_1$ be a GHC of a base system $\mathcal{R}_0 = (\mathcal{D}_0 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_0, R_0)$ and the extension $\mathcal{R}_1 = (\mathcal{D}_1 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_1, R_1)$. Define the sets $\mathcal{D}_1^0$, $\mathcal{D}_1^1$, R_1^0 and R_1^1 as in Definition 8. $\mathcal{R}_1$ is a generalized restricted proper extension (GrtPE) of $\mathcal{R}_0$ if it is a generalized proper extension and additionally, no left-hand side of $\mathcal{R}_1$ contains a function symbol from $\mathcal{D}_0 \cup \mathcal{D}_1^0$ strictly below its root.

Figure 8 shows how this new combination is related to previous combinations. Now, we can see that Example 8 is not a GrtPE because the extension
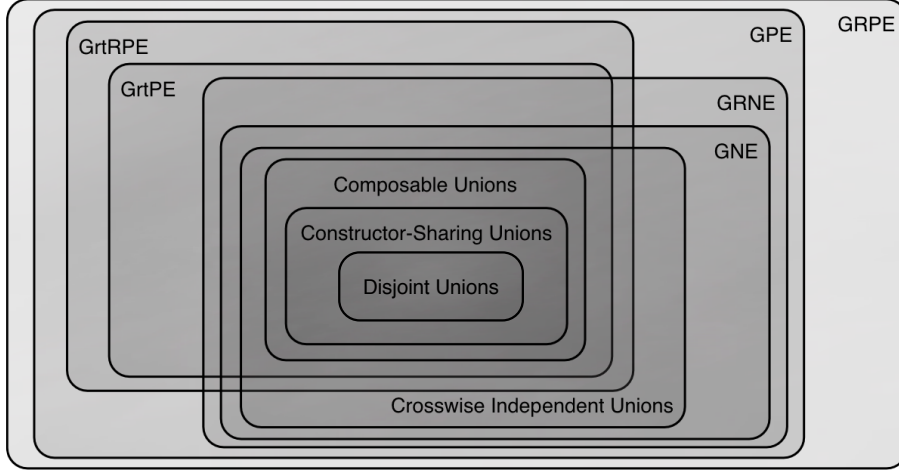
Figure 8: Modular combinations for E-unification

$\mathcal{R}_1$ contains a defined symbol from the base system $\mathcal{R}_0$ in one left-hand side. On the other hand, in Example 5, $\mathcal{R}_!$ is a GrtPE of $\mathcal{R}_*$.

The next theorem provides the conditions states that decidability of equational unification is modular for GrtPEs by essentially assembling together different results dispersed across the literature.

Theorem 10 (Modular Unification in Restricted Proper Extensions)
Decidability of E-unification via basic narrowing is a modular property for the union of two finite $(\overset{b}{\leadsto})$-terminating confluent systems where one is a generalized restricted proper extension of the other.

Proof. Basic narrowing termination is modular for finite GrtPEs since it is modular for finite GRPEs by Corollary 4. Confluence is shown to be modular for weakly normalizing GrtPEs in [37, Theorem 10]. Weak normalization is implied by $(\overset{b}{\leadsto})$-termination using Lemma 5. Finally, completeness follows from Corollary 5. ∎

For the modularity of completeness we also consider a restricted version of relaxed proper extensions called *restricted relaxed proper* extensions. Generalized *restricted* relaxed proper extensions are essentially generalized relaxed proper extensions with the extra condition on the left hand sides introduced in Definition 21. In the following we prove that decidability of E-unification is also fully modular for this class of combinations.

Definition 22 (Generalized Restricted Relaxed Proper Extension)
Let $\mathcal{R} = (\mathcal{D} \uplus \mathcal{C}, R)$ be a GHC of a base system $\mathcal{R}_0 = (\mathcal{D}_0 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_0, R_0)$ and the extension $\mathcal{R}_1 = (\mathcal{D}_1 \uplus \mathcal{D}_{sh} \uplus \mathcal{C}_1, R_1)$. Define the sets $\mathcal{D}_1^0$, $\mathcal{D}_1^1$, R_1^0 and R_1^1 as in Definition 8. $\mathcal{R}_1$ is a generalized restricted relaxed proper extension

(GrtRPE) of $\mathcal{R}_0$ iff it is a generalized relaxed proper extension and additionally, no left-hand side of $\mathcal{R}_1$ contains a funtion symbol from $\mathcal{D}_0 \cup \mathcal{D}_1^0$ strictly below its root.

Figure 8 shows how this new combination is related to previous combinations. The following result states the modularity of E-unification decidability in this class of combinations and strictly subsumes Theorem 10.

Theorem 11 (Modular Unification in Restricted Relaxed Proper E.)
Decidability of E-unification via basic narrowing is a modular property for the union of two finite $(\overset{b}{\leadsto})$-terminating confluent systems where one is a generalized relaxed restricted proper extension of the other.

Proof. Let $\mathcal{R}_1$ and $\mathcal{R}_0$ be two finite $(\overset{b}{\leadsto})$-terminating confluent TRSs such that $\mathcal{R}_1$ is a GrtRPE of $\mathcal{R}_0$. We define $\mathcal{R}_{1b}$ as the system that results of replacing[7] with a fresh symbol the root symbol in every rs−rnf in a right hand side of $\mathcal{R}_1$. $\mathcal{R}_{1b}$ is by construction a GrtPE of $\mathcal{R}_0$, and hence by Theorem 10 basic narrowing provides an effective E-unification algorithm for the union system $\mathcal{R}_b = \mathcal{R}_{1b} \cup \mathcal{R}_0$. That is, basic narrowing is complete and terminating in $\mathcal{R}_b$.

In order to show that basic narrowing is complete and terminating in the union $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_0$ too, we proceed by contradiction. Suppose that it is complete and terminating for $\mathcal{R}_b$ but not for $\mathcal{R}$. Then either:

- $\mathcal{R}$ is not $(\overset{b}{\leadsto})$-terminating,

- $\mathcal{R}$ is not complete.

But neither case is possible, since by definition of rs−rnf, $\mathcal{R}_b$ and $\mathcal{R}$ have the same derivations, i.e., one can establish a bisimulation between $\mathcal{R}_b$ and $\mathcal{R}$. Hence basic narrowing is also an effective E-unification algorithm for the union $\mathcal{R}$, which concludes the proof. ∎

In this section we have presented a number of modularity results for the decidability of E-unification via basic narrowing. Table 6 summarizes all these results. Note that $(\overset{b}{\leadsto})$-termination is an implicit condition in all cases in the table.

7 Conclusions

The completeness and termination properties of basic narrowing have previously been studied in landmark work [24, 38, 29]. In this article, we contribute to the analysis of basic narrowing termination by proving modular termination in several hierarchical combinations of TRSs, including a novel class of hierarchical combinations called *generalized relaxed proper extensions*. We also study

---

[7]For instance, for the TRS of Example 7 we have: $\mathcal{R}_1 : \exp(\exp(g, X), Y) \to \exp(g, X * Y)$ $\mathcal{R}_{1b} : \exp(\exp(g, X), Y) \to Q_{exp}(g, X * Y)$, where Q_{exp} is a fresh symbol.

TRS	Disjoint	Shared C.	Composable	GrtPE	GrtRPE	GPE	GRPE
Confl	Th. 8	Th. 8	Th. 8	Th. 10	Th. 11	No(Ex. 8)	No (Ex. 8)
Confl + Ov	Th. 8	Th. 8	Th. 8	Th. 10	Th. 11	Th. 9	Th. 9

Legend

Confl	Confluent
Ov	The combination is an overlay system

Table 1: Modularity of Decidability of E-unification via basic narrowing

the hierarchical combination of theories where basic narrowing terminates and is complete as an equational unification procedure, and we ascertain the conditions for the decidability of E-unification in the combined theory E. This is interesting because powerful functional languages featuring narrowing and equational unification, such as Maude (e.g., see [9]), include effective unification algorithms based on the basic narrowing strategy. Furthermore, in the context of cryptographic protocol verification, equational unification (e.g. via basic narrowing) is the basis for effective security and safety analyses (see [15, 16]).

8 Acknowledgements

References

[1] Alpuente, M., Escobar, S., Iborra, J., 2008a. Modular termination of basic narrowing. In: Voronkov, A. (Ed.), Proceedings of the 19th International Conference on Rewriting Techniques and Applications, RTA 2008. Vol. 5117 of Lecture Notes in Computer Science. Springer, pp. 1–16.

[2] Alpuente, M., Escobar, S., Iborra, J., 2008b. Termination of narrowing with dependency pairs. In: de la Banda, M. G., Pontelli, E. (Eds.), Proceedings of the 24th International Conference on Logic Programming, ICLP'08. Vol. 5366 of Lecture Notes in Computer Science. pp. 317–331.

[3] Alpuente, M., Escobar, S., Iborra, J., 2009. Termination of narrowing revisited. Theoretical Computer Science 410 (46), 4608 – 4625.

[4] Alpuente, M., Falaschi, M., Gabbrielli, M., Levi, G., 1993. The Semantics of Equational Logic Programming as an instance of CLP. In: Logic Programming Languages:49–81. The MIT Press.

[5] Alpuente, M., Falaschi, M., Levi, G., 1995. Incremental Constraint Satisfaction for Equational Logic Programming. Theoretical Computer Science 142 (1), 27–57.

[6] Alpuente, M., Falaschi, M., Vidal, G., 1998. Partial Evaluation of Functional Logic Programs. ACM Transactions on Programming Languages and Systems 20 (4), 768–844.

[7] Arts, T., Giesl, J., 2000. Termination of Term Rewriting using Dependency Pairs. Theoretical Computer Science 236 (1-2), 133–178.

[8] Bachmair, L., Dershowitz, N., 1986. Commutation, transformation, and termination. 8th Int'l Conference on Automated Deduction.

[9] Clavel, M., Durán, F., Eker, S., Escobar, S., Lincoln, P., Martí-Oliet, N., Meseguer, J., Talcott, C. L., 2009. Unification and narrowing in maude 2.4. In: Treinen, R. (Ed.), Proceedings of the 20th International Conference on Rewriting Techniques and Applications , RTA 2009, Brasília, Brazil, June 29 - July 1, 2009, Proceedings. Vol. 5595 of Lecture Notes in Computer Science. Springer, pp. 380–390.

[10] Comon-Lundh, H., 2004. Intruder Theories (Ongoing Work). In: Proceedings of the 7th International Conference on Foundations of Software Science and Computation Structures (FOSSACS). Vol. 2987 of Lecture Notes in Computer Science. Springer-Verlag, pp. 1–4.

[11] Cortier, V., Delaune, S., Lafourcade, P., 2006. A Survey of Algebraic Properties used in Cryptographic Protocols. Journal of Computer Security 14 (1), 1–43.

[12] Dershowitz, N., Jouannaud, J.-P., 1990. Rewrite systems. In: van Leeuwen, J. (Ed.), Handbook of Theoretical Computer Science. Vol. B: Formal Methods and Semantics. Elsevier Science, Ch. 6, pp. 244–320.

[13] Dershowitz, N., Mitra, S., Sivakumar, G., 1992. Decidable Matching for Convergent Systems. In: Proceedings of the 11th International Conference on Automated Deduction. Vol 607 of Lecture Notes in Computer Science, Springer-Verlag, pp. 589–602.

[14] Durán, F., Lucas, S., Meseguer, J., 2008. MTT: The Maude termination tool (system description). In: Armando, A., Baumgartner, P., Dowek, G. (Eds.), Proceedings of the 4th International Joint Conference on Automated Reasoning, IJCAR 2008, Sydney, Australia, August 12-15, 2008. Vol. 5195 of Lecture Notes in Computer Science. Springer-Verlag, pp. 313–319.

[15] Escobar, S., Meadows, C., Meseguer, J., 2006. A Rewriting-Based Inference System for the NRL Protocol Analyzer and its Meta-Logical Properties. Theoretical Computer Science 367 (1-2), Springer-Verlag, pp. 162–202.

[16] Escobar, S., Meadows, C., Meseguer, J., 2009. Maude-NPA: Cryptographic protocol analysis modulo equational properties. In: Aldini, A., Barthe, G., Gorrieri, R. (Eds.), FOSAD 2008/2009 Tutorial Lectures. Vol. 5705 of Lecture Notes in Computer Science. Springer-Verlag, pp. 1–50.

[17] Escobar, S., Meseguer, J., 2007. Symbolic model checking of infinite-state systems using narrowing. In: Proceedings of the 18th International Conference on Rewriting Techniques and Applications, RTA 2007. Vol. 4533 of Lecture Notes in Computer Science. Springer-Verlag, pp. 153–168.

[18] Fay, M., 1979. First-Order Unification in an Equational Theory. In: Proceedings of the Fourth International Conference on Automated Deduction. pp. 161–167.

[19] Giesl, J., Swiderski, S., Schneider-Kamp, P., Thiemann, R., 2006. Automated termination analysis for haskell: From term rewriting to programming languages. In: Pfenning, F. (Ed.), Proceedings of the 17th International Conference on Rewriting Techniques and Applications, RTA 2006, Seattle, WA, USA, August 12-14, 2006, Vol. 4098 of Lecture Notes in Computer Science. Springer-Verlag, pp.297–312.

[20] Gramlich, B., 1995. Abstract relations between restricted termination and confluence properties of rewrite systems. Fundamenta Informaticae 24, 3–23.

[21] Hanus, M., 1994. The Integration of Functions into Logic Programming: From Theory to Practice. Journal of Logic Programming 19&20, 583–628.

[22] Hölldobler, S., 1989. Foundations of Equational Logic Programming. Vol. 353 of Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin.

[23] Huet, G. P., 1980. Confluent reductions: Abstract properties and applications to term rewriting systems: Abstract properties and applications to term rewriting systems. Journal of the ACM 27 (4), 797–821.

[24] Hullot, J.-M., 1980. Canonical Forms and Unification. In: Proceedings of the 5th International Conference on Automated Deduction, CADE 1980. Vol. 87 of Lecture Notes in Computer Science, Springer-Verlag, pp. 318–334.

[25] Hullot, J.-M., 1981. Compilation de Formes Canoniques dans les Théories áquationelles. Thése de Doctorat de Troisième Cycle. Ph.D. thesis, Université de Paris Sud, Orsay (France).

[26] Lassez, J.-L., Maher, M. J., Marriott, K., 1988. Unification Revisited. In: Minker, J. (Ed.), Foundations of Deductive Databases and Logic Programming. Morgan Kaufmann, Los Altos, Ca., pp. 587–625.

[27] Meseguer, J., 1992. Multiparadigm logic programming. In: Kirchner, H., Levi, G. (Eds.), Proceedings of the Third International Conference on Algebraic and Logic Programming, ALP'92. Vol. 632 of Lecture Notes in Computer Science, Springer-Verlag, pp. 632:158–200.

[28] Meseguer, J., Thati, P., 2007. Symbolic reachability analysis using narrowing and its application to verification of cryptographic protocols. Higher-Order and Symbolic Computation 20 (1-2), 123–160.

[29] Middeldorp, A., Hamoen, E., 1994. Completeness Results for Basic Narrowing. Journal of Applicable Algebra in Engineering, Communication and Computing 5, 313–353.

[30] Middeldorp, A., Okui, S., Ida, T., 1996. Lazy narrowing: Strong completeness and eager variable elimination. Theoretical Computer Science 167 (1,2), pp. 95–130.

[31] Newman, M. H. A., 1942. On theories with a combinatorial definition of equivalence. Annals of Mathematics 43(2), 223–243.

[32] Ohlebusch, E., Jan 1995. Modular properties of composable term rewriting systems. Journal of Symbolic Computation 20, 1–41.

[33] Ohlebusch, E., 2002. Advanced Topics in Term Rewriting. Springer-Verlag.

[34] Plotkin, G. D., 1970. A note on inductive generalization. Machine Intelligence 5, 153–165.

[35] Prehofer, C., 1994. On Modularity in Term Rewriting and Narrowing. In: Proceedings of the First International Conference on Constraints in Computational Logic. Vol. 845 of Lecture Notes in Computer Science. Springer-Verlag, pp. 253–268.

[36] Rao, M. K., Jan 1995a. Modular proofs for completeness of hierarchical term rewriting systems. Theoretical Computer Science 151 (2), pp. 487–512.

[37] Rao, M. R. K. K., 1995b. Semi-completeness of hierarchical and super-hierarchical combinations of term rewriting systems. In: TAPSOFT '95: Proceedings of the 6th International Joint Conference CAAP/FASE on Theory and Practice of Software Development. Springer-Verlag, London, UK, pp. 379–393.

[38] Réty, P., 1987. Improving Basic Narrowing Techniques. In: Proceedings of the Second International Conference on Rewriting Techniques and Applications (RTA). Vol. 256 of Lecture Notes in Computer Science. Springer-Verlag, pp. 228–241.

[39] Schmidt-Schauß, M., 1988. Unification in a Combination of Arbitrary Disjoint Equational Theories. In: Proceedings of the 9th International Conference on Automated Deduction (CADE). Vol. 310 of Lecture Notes in Computer Science. Springer-Verlag, pp. 378–396.

[40] Siekmann, J. H., 1989. Unification Theory. Journal of Symbolic Computation 7, 207–274.

[41] TeReSe (Ed.), 2003. Term Rewriting Systems. Cambridge University Press.

[42] Toyama, Y., 1987. Counterexamples to termination for the direct sum of term rewriting systems. Information Processing Letters 25 (3), 141–143.

[43] Urbain, X., 2004. Modular & incremental automated termination proofs. Journal of Approximated Reasoning 32 (4), 315–355.

[44] Yamamoto, A., 1988. Completeness of extended unification based on basic narrowing. https://qir.kyushu-u.ac.jp.