A Universal Unification Algorithm Based on Unification-Driven Leftmost Outermost Narrowing

Heinz Faßbender* and Heiko Vogler

Abt. Theoretische Informatik, Universität Ulm

Oberer Eselsberg, W-7900 Ulm, Germany

e-mail: {fassbend,vogler}@informatik.uni-ulm.de

Abstract

We formalize a universal unification algorithm for the class of equational theories which is induced by the class of canonical, totally-defined, not strictly subunifiable term rewriting systems (for short: *ctn-trs*). For a *ctn-trs* $\mathcal{R}$ and for two terms t and s, the algorithm computes a ground-complete set of $E_{\mathcal{R}}$-unifiers of t and s, where $E_{\mathcal{R}}$ is the set of rewrite rules of $\mathcal{R}$ viewed as equations. The algorithm is based on the *unification-driven leftmost outermost narrowing relation* (for short: *ulo narrowing relation*) which is introduced in this paper. The ulo narrowing relation combines usual leftmost outermost narrowing steps and unification steps. Since the unification steps are applied as early as possible, some of the nonsuccessful derivations can be stopped earlier than in other approaches to $E_{\mathcal{R}}$-unification. Furthermore, we formalize a deterministic version of our universal unification algorithm that is based on a depth-first left-to-right traversal through the narrowing trees.

*The work of this author has been supported by the Deutsche Forschungsgemeinschaft (DFG).

1 Introduction

The *unification problem* is to determine whether or not, for two given terms t and s, there exists a unifier φ of t and s, i.e., a substitution φ such that $\varphi(t) = \varphi(s)$. It is well-known that the unification problem for first-order terms is decidable [Rob65]. Actually, there are algorithms which compute the most general unifier of t and s, if there exists a unifier at all (cf., e.g., [PW78, MM82]).

The problem of unification generalizes to the problem of E-unification if one considers the equality modulo a set E of equations, denoted by $=_E$, rather than the usual equality; $=_E$ is also called the equational theory induced by E. The *E-unification problem* is to determine whether or not, for two given terms t and s, there exists a substitution φ such that $\varphi(t) =_E \varphi(s)$; then φ is called an E-unifier of t and s. Clearly, the decidability of the E-unification problem depends on the set E of equations. If E is the empty set, then the E-unification problem coincides with the unification problem and henceforth it is decidable. If E is the set of Peano's axioms, then the E-unification problem becomes undecidable, because it is precisely Hilbert's tenth problem, which was shown to be undecidable [Mat70]. Upto now the E-unification problem has been studied in particular for sets of algebraic laws like the laws of commutativity, associativity, idempotence, or distributivity. A survey about these investigations can be found in [Sie89].

For a class $\mathcal{E}$ of equational theories, a *universal unification algorithm for $\mathcal{E}$* is an algorithm which takes as input an equational theory $=_E$ from the class $\mathcal{E}$ and two terms t and s, and which computes a complete set of E-unifiers of t and s. In this paper, we will concentrate on universal unification algorithms for classes of equational theories which are induced by particular term rewriting systems (for short: trs's). A trs $\mathcal{R}$ induces the equational theory $=_{E_\mathcal{R}}$, where $E_\mathcal{R}$ is the set of rules of $\mathcal{R}$ viewed as equations. Since now, a lot of research has been carried out to construct universal unification algorithms for classes $\mathcal{E}$ of equational theories which are induced by trs's. In fact, all the approaches are based on the concept of narrowing [Lan75]. More precisely, every investigation shows that the use of a particular narrowing relation is complete for a particular class of trs's. Here we list some of the investigations by showing the corresponding pairs of narrowing relation and class of trs's.

- narrowing and canonical trs's [Fay79, Hul80]
- basic narrowing and canonical trs's [Hul80, MH92]
- outer narrowing and confluent, constructor-based trs's [You89]
- any innermost narrowing strategy and totally-defined trs's [Fri85]
- any narrowing strategy and canonical, totally-defined, not strictly subunifiable trs's [Ech88].

We note that a narrowing strategy is a narrowing relation in which the narrowing occurrence is fixed. We also recall that a trs is *canonical*, if it is confluent and noetherian. A trs is *constructor-based*, if its ranked alphabet Ω is partitioned into sets F and Δ of function symbols and constructor symbols, respectively; moreover, the left-hand sides of rules are linear terms $f(t_1, \ldots, t_n)$ where f is a function symbol, $t_1, \ldots, t_n$ are terms over $\Delta \cup \mathcal{V}$ where $\mathcal{V}$ is the set of variables (cf. [You89]). A trs is *totally-defined*, if it is constructor-based and every function symbol is completely defined over its domain or, equivalently: every normal form is a constructor term (cf., e.g., [Ech88]). A trs is *not strictly subunifiable*, if, roughly speaking, two rules cannot be applied at the same occurrence under the same substitution (cf. [Ech88] and Subsection 6.2 of the present paper).

In this paper we present a universal unification algorithm for the class of equational theories which are induced by canonical, totally-defined, not strictly subunifiable trs's (for short: ctn-trs 's). Although, at first glance, it might seem that our approach is subsumed by the results of [Ech88], we will show later that this is not true. To give the reader an idea about the power of ctn-trs's, we mention that every modular tree transducer [EV91] is a trs of this type; the class of modular tree transducers characterizes the class of primitive recursive tree functions [Hup78]. But in fact, ctn-trs's are even more powerful than modular tree transducers. Our universal unification algorithm is based on the so-called *unification-driven leftmost outermost narrowing relation* (for short: ulo narrowing relation) which is introduced in this paper. For a trs $\mathcal{R}$, the ulo narrowing relation is denoted by $\overset{u}{\leadsto}_{\mathcal{R}}$. Roughly speaking, this relation combines the usual leftmost outermost narrowing strategy with steps which are adopted from the usual unification algorithm for terms. More precisely, the leftmost outermost narrowing is modified such that as soon as possible a rule can be applied which is known as decomposition rule in the unification algorithm of [MM82]; in this sense, the ulo narrowing relation is unification-driven. In [Fri85] a similar idea has been applied in the context of innermost narrowing. Let us give an example at which we can illustrate the ulo narrowing relation.

In Figure 1 a set R_1 of rules of the ctn-trs $\mathcal{R}_1$ is shown where we assume to have a ranked alphabet $F = \{sh^{(2)}, mi^{(1)}\}$ of function symbols and a ranked alphabet $\Delta = \{\sigma^{(2)}, \alpha^{(0)}\}$ of constructor symbols. Intuitively, $\mathcal{R}_1$ defines two functions *shovel* and *mirror* with arity 2 and 1, respectively; *mirror* reflects terms over Δ at the vertical center line, and *shovel* accumulates in its second argument the *mirror*-image of the second subterm of its first argument. If we consider, e.g., the term $t_1 = \sigma(\sigma(\alpha, s_1), s_2)$ for some subterm s_1 and s_2, then for an arbitrary term t_2, $shovel(t_1, t_2)$ is the term $\sigma(mirror(s_1), \sigma(mirror(s_2), t_2))$.

$$
\begin{array}{rcll}
sh(\alpha, y_1) & \to & y_1 & (1) \\
sh(\sigma(x_1, x_2), y_1) & \to & sh(x_1, \sigma(mi(x_2), y_1)) & (2) \\
mi(\alpha) & \to & \alpha & (3) \\
mi(\sigma(x_1, x_2)) & \to & \sigma(mi(x_2), mi(x_1)) & (4)
\end{array}
$$

Figure 1: Set of rules of the ctn-trs $\mathcal{R}_1$.

Now we consider the $E_{\mathcal{R}_1}$-unification problem, where the set $E_{\mathcal{R}_1}$ of equations is obtained from R_1 by simply considering the rules as equations. In particular, we want to compute an $E_{\mathcal{R}_1}$-unifier for the terms $sh(z_1, \alpha)$ and $mi(\sigma(z_2, \alpha))$ in which z_1 and z_2 are free variables. Similar to Hullot in [Hul80], we combine the two terms into one term $equ(sh(z_1, \alpha), mi(\sigma(z_2, \alpha)))$ with a new binary symbols equ (which is called H in [Hul80]).

However, now we do not perform the narrowing relation with the trs $\mathcal{R}_1$ but with the trs $\hat{\mathcal{R}}_1$ which contains the set $R_1 \cup R(\Delta)$ of rules; the set $R(\Delta)$ of *equal-rules of* Δ is shown in Figure 2.

$$\begin{array}{rcll} equ(\alpha, \alpha) & \rightarrow & \alpha & (5) \\ equ(\sigma(x_1, x_2), \sigma(x_3, x_4)) & \rightarrow & \sigma(equ(x_1, x_3), equ(x_2, x_4)) & (6) \end{array}$$

Figure 2: Set of equal-rules of Δ.

Then a derivation by $\overset{u}{\rightsquigarrow}_{\hat{\mathcal{R}}_1}$ starting from $equ(sh(z_1, \alpha), mi(\sigma(z_2, \alpha)))$ may look as follows where we have attached to $\overset{u}{\rightsquigarrow}_{\hat{\mathcal{R}}_1}$ in every step the narrowing occurrence (in Dewey's notation), the applied rule, and the unifier as additional indices; $\varphi_\emptyset$ denotes the empty substitution.

$$\begin{array}{lll}
 & & equ(sh(z_1, \alpha), mi(\sigma(z_2, \alpha))) \\
 & \overset{u}{\rightsquigarrow}_{\hat{\mathcal{R}}_1,1,(2),[z_1/\sigma(z_3,z_4)]} & equ(sh(z_3, \sigma(mi(z_4), \alpha)), mi(\sigma(z_2, \alpha))) \\
 & \overset{u}{\rightsquigarrow}_{\hat{\mathcal{R}}_1,1,(1),[z_3/\alpha]} & equ(\sigma(mi(z_4), \alpha), mi(\sigma(z_2, \alpha))) \\
* & \overset{u}{\rightsquigarrow}_{\hat{\mathcal{R}}_1,2,(4),\varphi_\emptyset} & equ(\sigma(mi(z_4), \alpha), \sigma(mi(\alpha), mi(z_2))) \\
** & \overset{u}{\rightsquigarrow}_{\hat{\mathcal{R}}_1,\Lambda,(6),\varphi_\emptyset} & \sigma(equ(mi(z_4), mi(\alpha)), equ(\alpha, mi(z_2))) \\
 & \overset{u}{\rightsquigarrow}_{\hat{\mathcal{R}}_1,11,(3),[z_4/\alpha]} & \sigma(equ(\alpha, mi(\alpha)), equ(\alpha, mi(z_2))) \\
 & \overset{u}{\rightsquigarrow}_{\hat{\mathcal{R}}_1,12,(3),\varphi_\emptyset} & \sigma(equ(\alpha, \alpha), equ(\alpha, mi(z_2))) \\
 & \overset{u}{\rightsquigarrow}_{\hat{\mathcal{R}}_1,1,(5),\varphi_\emptyset} & \sigma(\alpha, equ(\alpha, mi(z_2))) \\
 & \overset{u}{\rightsquigarrow}_{\hat{\mathcal{R}}_1,22,(3),[z_2/\alpha]} & \sigma(\alpha, equ(\alpha, \alpha)) \\
 & \overset{u}{\rightsquigarrow}_{\hat{\mathcal{R}}_1,2,(5),\varphi_\emptyset} & \sigma(\alpha, \alpha)
\end{array}$$

If we compose the unifiers which are involved in the narrowing steps, then we obtain the substitution $\varphi = [z_1/\sigma(\alpha, \alpha), z_2/\alpha]$; in fact, φ is an $E_{\mathcal{R}_1}$-unifier of $sh(z_1, \alpha)$ and $mi(\sigma(z_2, \alpha))$. Note that φ is not an $E_{\hat{\mathcal{R}}_1}$-unifier, because the equational theory is generated by $E_{\mathcal{R}_1}$. The narrowing step at $*$ shows how the ulo narrowing relation deviates from the leftmost outermost narrowing relation. For the latter relation, 11 is the narrowing occurrence in the term $equ(\sigma(mi(z_4), \alpha), mi(\sigma(z_2, \alpha)))$ and the subterm $mi(z_4)$ has to be narrowed. But it is clear that any normal form s_1' of the first argument $s_1 = \sigma(mi(z_4), \alpha)$ of equ is unifiable with a normal form s_2' of the second argument $s_2 = mi(\sigma(z_2, \alpha))$ of equ only if the constructors at the root of s_1' and s_2' are identical. Because of reasons of efficiency, it is of course important to check this equality as soon as possible. And since s_1 is already evaluated in constructor head normal form, we narrow s_2 at step $*$ and try to get it also into head normal form. Actually, this form is reached immediately. Then, at step $**$, the equality of root symbols is checked by applying the equal-rule (6).

Thus, in general, in the ulo narrowing relation the leftmost occurrence *impO* of the *equ* symbol is important in the sense that its direct sons decide how to proceed further:

- If the first son of *impO* is labeled by a function symbol, then a usual leftmost outermost narrowing step on the basis of the original rules is performed on the subterm which starts at the first son of *impO*.

- If the first son of *impO* is labeled by a constructor or a variable and the second son of *impO* is labeled by a function symbol, then a usual leftmost outermost narrowing step is performed on the subterm which starts at the second son of *impO* (as, e.g., derivation step $*$ in our example).

- If the first son and the second son of *impO* are labeled by the same constructor or, if one of the sons is labeled by a variable and the other son is labeled by a constructor, then an equal-rule is applied to *impO* (as, e.g., in the derivation step $**$).

One situation is still missing, viz., if both sons of *impO* are variables, say, z_1 and z_2. We call this situation *binding modus* (for short: *bm*), and the ulo narrowing relation performs the following step:

$$equ(z_1, z_2) \overset{u}{\leadsto}_{\Lambda, bm, [z_1/z_2]} z_2.$$

As usual for narrowing relations, if the term $equ(z_1, z_2)$ occurs as a subterm of the current derivation form t, then the substitution $[z_1/z_2]$ has to be applied to the whole term t. It is clear that, by applying equal-rules as soon as possible, some unsuccessful derivations can be stopped early. By means of the binding modus, we have kept the computed $E_{\mathcal{R}}$-unifier as general as possible.

Now let us briefly discuss why our approach is not subsumed by the results of [Ech88]. In order to compare the approaches, one has to consider equ as an additional function symbol. But it behaves quite differently in our approach. For instance, equ is not totally defined, because the normal form of the term $t = equ(\sigma(\alpha, \alpha), \alpha)$ is not a constructor term (but it is t itself which contains equ). Moreover, the binding mode which is applied to the term $equ(z_1, z_2)$, should be simulated by the additional rule

$$equ(z_1, z_1) \to z_1.$$

However, this rule is not linear in its left-hand side. Moreover, its left-hand side is strictly subunifiable with the left-hand sides of all the other equal-rules. Hence, the resulting trs is not a ctn-trs, and thus, the approach of [Ech88] is not applicable here.

Actually, for a ctn-trs $\mathcal{R}$ with set Δ of constructors and two terms t and s, our universal unification algorithm computes a *ground complete set of* $(E_{\mathcal{R}}, \Delta)$*-unifiers* of t and s. An $(E_{\mathcal{R}}, \Delta)$-unifier of t and s is an $E_{\mathcal{R}}$-unifier in which all the images are terms over $\Delta \cup \mathcal{V}$, where $\mathcal{V}$ is the set of variables; in particular, this means that we do not consider unifiers of the form $[z_1/f(t)]$ for some function symbol f. Roughly speaking, a set S of $(E_{\mathcal{R}}, \Delta)$-unifiers of t and s is ground complete, if, for every ground $(E_{\mathcal{R}}, \Delta)$-unifier φ of t and s (i.e., the images of φ do not contain variables), there is a $\psi \in S$ which is more general than φ. This notion will be formalized in Section 6.

If one studies or introduces particular narrowing strategies, then one should ask about the efficiency of the proposed formalisms. How can such an efficiency be measured or, at least, be illustrated? We will illustrate the efficiency of our approach by means of trees in which, for terms like

$$t = equ(sh(z_1, \alpha), mi(\sigma(z_2, \alpha))),$$

all the possible derivations are collected which are induced by some narrowing relation $\leadsto$ and which start from t; we call such a tree the *narrowing tree of* t and we denote it by

$nar\text{-}tree(\rightsquigarrow, t)$. The nodes of $nar\text{-}tree(\rightsquigarrow, t)$ are labeled by terms over $\Omega \cup \mathcal{V}$; in particular, the root is labeled by t. If a node nd of the narrowing tree is labeled by a term s and if s derives by the narrowing relation in k different ways to the terms $s_1, \ldots, s_k$, then nd has k children which are labeled by $s_1, \ldots, s_k$. In the context of PROLOG programs, such trees are known as SLD-trees [Llo87]. Then, as a rough measurement, a narrowing relation $\rightsquigarrow_1$ is more efficient than the narrowing relation $\rightsquigarrow_2$, if the set of terms is partitioned into two sets $T_<$ and $T_=$ such that (1) for every $t \in T_<$, the size of $nar\text{-}tree(\rightsquigarrow_1, t)$ is smaller than the size of $nar\text{-}tree(\rightsquigarrow_2, t)$ and, for every $t \in T_=$, the sizes of $nar\text{-}tree(\rightsquigarrow_1, t)$ and $nar\text{-}tree(\rightsquigarrow_2, t)$ are equal, and (2) $T_< \neq \emptyset$.

In general, two sources of nondeterminism form the building principles for narrowing trees:

1. there may be more than one narrowing occurrence in the current derivation form t and

2. at one narrowing occurrence, the corresponding subterm of t may be unifiable with the left hand sides of more than one rule.

In the sequel we will order the successors of a term t, i.e., the elements of the set $\{t_j | t \rightsquigarrow t_j\}$, by first considering the lexicographical ordering on the narrowing occurrences of t and second, if more than one rule is applicable at one occurrence, taking an implicite ordering of the rules into account. To give an example of a narrowing tree, we again consider the term

$$t = equ(sh(z_1, \alpha), mi(\sigma(z_2, \alpha))).$$

Figure 3 shows the narrowing tree of t induced by the ulo-narrowing relation. This tree should be compared with the leftmost outermost narrowing tree in Figure 6 which, in its turn, should be compared with the narrowing tree in Figure 5. This comparison shows that both changes (from narrowing to leftmost outermost narrowing, and from leftmost outermost narrowing to ulo-narrowing) may increase the efficiency of the unification algorithm in the sense explained above.

The presentation of the unification algorithm and of the ulo narrowing relation in this paper is formal; in particular, we prove the completeness of our algorithm. We develop our approach in a stepwise fashion by recalling a sequence of known universal unification algorithms (Theorem 4.12, Theorem 5.5 cf. [Hul80], Theorem 6.10 cf. [Ech88], Theorem 7.8) which leads us to our algorithm in Theorem 7.9. And in fact, during trying to formalize the algorithms, we have felt a strong need to recall the basic concepts of *E-unification* and of *narrowing* also in a formal way. This is the reason why this paper has become a bit lengthy. Readers who are familiar with these concepts, may skip Section 2 to Section 5 and start immediately with Section 6.

This paper is organized in nine sections where the second section contains preliminaries; in particular, there we formalize the framework of derivation calculus and derivation trees which will later be instantiated to narrowing trees for the narrowing relation, the leftmost outermost narrowing relation, and the ulo-narrowing relation. In Section 3 we recall notions about E-unification from [HO80]. In Section 4 we present an overview over the notations and results of term rewriting systems as far as they are needed in the present

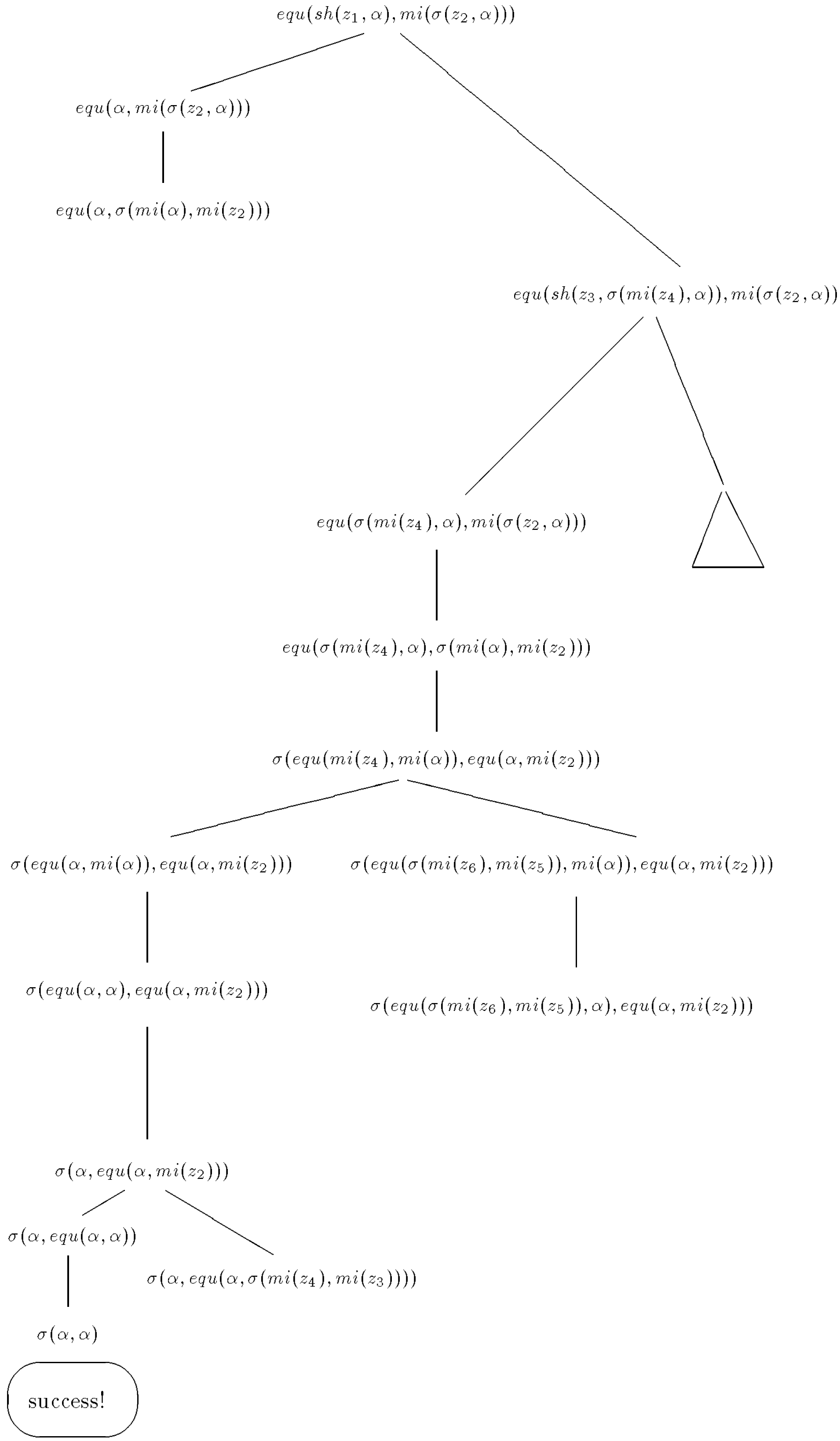

Figure 3: Narrowing tree for t induced by the ulo narrowing relation.

paper. In Section 5 we recall the algorithm from [Hul80] and introduce the narrowing derivation calculus. In Section 6 the leftmost outermost narrowing derivation calculus and ctn-trs's are introduced and the corresponding algorithm of [Ech88] is recalled. In Section 7 we define the unification-driven leftmost outermost narrowing derivation calculus and formalize the universal unification algorithm for the class of equational theories which are characterized by ctn-trs's. In Section 8 we define the deterministic version of our universal unification algorithm which is based on a depth-first left-to-right traversal over the ulo-narrowing trees. Finally, Section 9 contains some concluding remarks and indicates further research topics.

2 Preliminaries

We recall and collect some notations, basic definitions, and terminology which will be used in the rest of the paper. We tried to be in accordance with the notations in [Hue80] and [DJ91] as much as possible.

2.1 General notations

We denote the set of nonnegative integers by $\mathbb{N}$. The empty set is denoted by $\emptyset$. For $j \in \mathbb{N}$, $[j]$ denotes the set $\{1, \ldots, j\}$; thus $[0] = \emptyset$ and for $i, j \in \mathbb{N}$, $[i, j]$ denotes the interval $\{i, i+1, \ldots, j\}$. For a finite set A, $\mathcal{P}(A)$ is the *set of subsets* of A and $card(A)$ denotes the *cardinality* of A. For $n \in \mathbb{N}$, $I(n)$ denotes the set $\bigcup_{i,j \in [0,n]} [i, j]$ of *intervals of* $[0, n]$. As usual for a set A, A^* denotes the set $\bigcup_{n \in \mathbb{N}} \{a_1 a_2 \ldots a_n \mid \text{for every } i \in [n] : a_i \in A\}$ that is called the set of *words over A.*

2.2 Ranked Alphabets, Variables, and Terms

A pair $(\Omega, \underline{rank}_\Omega)$ is called *ranked alphabet,* if Ω is an alphabet and $\underline{rank}_\Omega : \Omega \longrightarrow \mathbb{N}$ is a total function. For $f \in \Omega$, $\underline{rank}_\Omega(f)$ is called *rank of f; maxrank*Ω denotes the maximum of the image of $\underline{rank}_\Omega$. The subset $\Omega^{(m)}$ of Ω consists of all symbols of rank m $(m \geq 0)$. Note that, for $i \neq j$, $\Omega^{(i)}$ and $\Omega^{(j)}$ are disjoint. We can define a ranked alphabet $(\Omega, \underline{rank}_\Omega)$ either by enumerating the finite subsets $\Omega^{(m)}$ that are not empty, or by giving a set of symbols that are indexed with their (unique) rank. For example, if $\Omega = \{a, b, c\}$ and $\underline{rank}_\Omega : \Omega \longrightarrow \mathbb{N}$ with $\underline{rank}_\Omega(a) = 0$, $\underline{rank}_\Omega(b) = 2$, and $\underline{rank}_\Omega(c) = 7$, then we can describe $(\Omega, \underline{rank}_\Omega)$ either by $\Omega^{(0)} = \{a\}$, $\Omega^{(2)} = \{b\}$, and $\Omega^{(7)} = \{c\}$ or by $\{a^{(0)}, b^{(2)}, c^{(7)}\}$. If the ranks of the symbols are clear from the context, then we drop the function $\underline{rank}_\Omega$ from the denotation of the ranked alphabet $(\Omega, \underline{rank}_\Omega)$ and simply write Ω.

In the rest of the paper we let $\mathcal{V}$ denote a fixed enumerable set. Its elements are called *variables.* In the following we use the notations $x, x_1, x_2, \ldots, y, y_1, y_2, \ldots, z, z_1, z_2, \ldots$ for variables.

Let $(\Omega, \underline{rank}_\Omega)$ be a ranked alphabet and let S be an arbitrary set. Then the set of *terms over* Ω *indexed by S,* denoted by $T_\Omega(S)$, is defined inductively as follows:

(i) $S \subseteq T\langle\Omega\rangle(S)$.

(ii) For every $f \in \Omega^{(k)}$ with $k \geq 0$ and $t_1, \ldots, t_k \in T\langle\Omega\rangle(S)$: $f(t_1, \ldots, t_k) \in T\langle\Omega\rangle(S)$.

The set $T\langle\Omega\rangle(\emptyset)$, denoted by $T\langle\Omega\rangle$, is called the set of *ground terms over* Ω.

Let Λ denote the empty word. For a term $t \in T\langle\Omega\rangle(\mathcal{V})$, the set of *occurrences of t,* denoted by $O(t)$, is a subset of $\mathbb{N}^*$ and it is defined inductively on the structure of t as follows:

(i) If $t = x$ where $x \in \mathcal{V}$, then $O(t) = \{\Lambda\}$.

(ii) If $t = f$ where $f \in \Omega^{(0)}$, then $O(t) = \{\Lambda\}$.

(iii) If $t = f(t_1, \ldots, t_n)$ where $f \in \Omega^{(n)}$ and $n > 0$, and for every $i \in [n] : t_i \in T\langle\Omega\rangle(\mathcal{V})$, then $O(t) = \{\Lambda\} \cup \bigcup_{i \in [n]} \{iu \mid u \in O(t_i)\}$.

The prefix order on $O(t)$ is denoted by $\leq$ and the lexicographical order on $O(t)$ is denoted by $\leq_{lex}$. The minimal element with respect to $\leq_{lex}$ in a subset S of $O(t)$ is denoted by $min_{lex}S$. For a term $t \in T\langle\Omega\rangle(\mathcal{V})$ and an occurrence u of t, t/u denotes the *subterm of t at occurrence u,* and $t[u]$ denotes the *label of t at occurrence u.* We use $\mathcal{V}(t)$ to denote the set of variables occurring in t; that is, $x \in \mathcal{V}(t)$, if $x \in \mathcal{V}$ and there exists a $u \in O(t)$ such that $t/u = x$. Finally, we define $t[u \leftarrow s]$ as the term t in which we have replaced the subterm at occurrence u by the term s.

2.3 Algebras, Substitutions, and Congruences

Let $(\Omega, \underline{rank}_\Omega)$ be a ranked alphabet.

An Ω-*algebra* is a pair (A, int_A), where A is a set and int_A is a mapping such that:

$$int_A(f) \in A, \text{ if } \underline{rank}_\Omega(f) = 0, \text{ and}$$
$$int_A(f) : A^n \rightarrow A, \text{ if } \underline{rank}_\Omega(f) = n.$$

The pair $(T\langle\Omega\rangle(\mathcal{V}), int_T)$, where for every $f \in \Omega^{(n)}$ and for every $t_i \in T\langle\Omega\rangle(\mathcal{V})$ with $i \in [n] : int_T(f)(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$, is an Ω-algebra. It is called the Ω-*term algebra.* The Ω-term algebra is a free Ω-algebra (cf. [HO80]).

If (A, int_A) and (B, int_B) are two Ω-algebras, we say that $h : A \rightarrow B$ is a *homomorphism,* if for every $f \in \Omega^{(n)}$ with $n \geq 0$ and for every $a_i \in A$ with $i \in [n]$, we have

$$h(int_A(f)(a_1, \ldots, a_n)) = int_B(f)(h(a_1), \ldots, h(a_n)).$$

A mapping $\nu : \mathcal{V} \rightarrow A$ is called an A-*assignment.*

The property that every A-assignment can be extended in a unique way to a homomorphism from $T\langle\Omega\rangle(\mathcal{V})$ to A is called the *universal property for the free Ω-algebras* in [HO80]. We use ν to denote both the A-assignment and its extension.

A $T\langle\Omega\rangle(\mathcal{V})$-assignment φ, where the set $\{x \mid \varphi(x) \neq x, x \in \mathcal{V}\}$ is finite, is called a $(\mathcal{V}, \Omega)$-*substitution.* The set $\{x \mid \varphi(x) \neq x\}$ is denoted by $\mathcal{D}(\varphi)$ and is called the *domain of* φ. If $\mathcal{D}(\varphi) = \{x_1, \ldots, x_n\}$, then φ is represented as $[x_1/\varphi(x_1), \ldots, x_n/\varphi(x_n)]$. If $\mathcal{D}(\varphi) = \emptyset$, then φ is denoted by $\varphi_\emptyset$. We say that φ is *ground,* if for every $x \in \mathcal{D}(\varphi) : \mathcal{V}(\varphi(x)) = \emptyset$. The set $\bigcup_{x \in \mathcal{D}(\varphi)} \mathcal{V}(\varphi(x))$ is denoted by $\mathcal{I}(\varphi)$ and is called the set of *variables introduced by* φ. The set of $(\mathcal{V}, \Omega)$-substitutions and the set of ground $(\mathcal{V}, \Omega)$-substitutions are denoted by $Sub(\mathcal{V}, \Omega)$ and $gSub(\mathcal{V}, \Omega)$, respectively. The *composition* of two substitutions φ and ψ is the $T\langle\Omega\rangle(\mathcal{V})$-assignment which is defined by $\psi(\varphi(x))$ for every $x \in \mathcal{V}$. It is denoted by $\varphi \circ \psi$.

An equivalence relation $\sim$ on $T\langle\Omega\rangle(\mathcal{V})$ is called an Ω-*congruence over* $T\langle\Omega\rangle(\mathcal{V})$, if for every $f \in \Omega^{(n)}$ with $n > 0$ and for every $t_1, s_1, \ldots, t_n, s_n \in T\langle\Omega\rangle(\mathcal{V})$ with $t_1 \sim s_1, \ldots, t_n \sim s_n$:

$$f(t_1, \ldots, t_n) \sim f(s_1, \ldots, s_n).$$

2.4 Derivation Calculus

A *derivation calculus,* denoted by $\mathcal{D}$, is a pair $(D, \vdash)$ where D is a set and $\vdash$ is a binary relation over D. (In the sequel it will always be clear whether $\mathcal{D}$ denotes the domain of a substitution or a derivation calculus.) The elements of D are called *derivation forms* and $\vdash$ is called a *derivation relation.* An element $(d, d') \in \vdash$ is denoted by $d \vdash d'$ and we say that *d derives to d' by $\vdash$.* We use the standard notations $\vdash^+$ and $\vdash^*$ to denote the transitive closure and the transitive-reflexive closure of $\vdash$, respectively. $d \vdash^* d'$ and $d \vdash d'$ are also called *derivation by $\vdash$ starting with d* and *derivation step by $\vdash$,* respectively. For a derivation form $d \in D$, the set $\{d' \in D \mid d \vdash d'\}$, denoted by $Suc(d, \vdash)$, is called the *set of successors of d.*

A derivation form $d \in D$ is *irreducible* or *in normal form,* if $Suc(d, \vdash)$ is empty; d' is a *normal form of d,* if $d \vdash^* d'$ and d' is in normal form. A substitution φ is *in normal form,* if for every $x \in \mathcal{D}(\varphi) : \varphi(x)$ is in normal form.

Let $\mathcal{D} = (D, \vdash)$ be a derivation calculus. An *indexed derivation calculus for $\mathcal{D}$,* denoted by $\mathcal{D}_J$, is a triple $(D, (J, <), (\vdash_j \mid j \in J))$, such that the following conditions hold:

1. J is an index-set.
2. $<$ is a total order on J.
3. For every $j \in J$, $\vdash_j$ is a binary relation on D and $(\bigcup_{j \in J} \vdash_j) = \vdash$.

Let $\mathcal{D}_J = (D, (J, <), (\vdash_j \mid j \in J))$ be an indexed derivation calculus, and let $d \in D$. A *derivation tree for $\mathcal{D}_J$ and d* is a tree, denoted by $T(\mathcal{D}_J, d)$ such that the following conditions hold:

1. Every node of $T(\mathcal{D}_J, d)$ is labeled by a derivation form $d' \in D$.
2. The root of $T(\mathcal{D}_J, d)$ is labeled by d.
3. Let d' be the label of a node nd in $T(\mathcal{D}_J, d)$, and let $\{j \in J \mid \text{there is a } d^* \in D : d' \vdash_j d^*\}$ be the set $\{j_1, \ldots, j_n\}$ such that $j_1 < j_2 < \ldots < j_n$. Then nd has n sons and for every $k \in [n]$, the k-th son of nd is labeled by $\bar{d}$, if $d' \vdash_{j_k} \bar{d}$.

Intuitively, every path through $T(\mathcal{D}_J, d)$ shows a derivation by $\vdash$ starting with d and, vice versa, every derivation by $\vdash$ starting with d is represented by a path through $T(\mathcal{D}_J, d)$. Hence, the size of $T(\mathcal{D}_J, d)$ gives an idea of the complexity of enumerating the derivations that start with d.

Hier Definition von Menge der narrowing interfaces rein

3 E-Unification

The aim of this section is to recall the definition of E-unification. Since this concept is a central notion in our whole investigation, we will take some effort for its introduction. First of all we have to specify what an equation is. Recall that Ω denotes an arbitrary ranked alphabet and $\mathcal{V}$ denotes a fixed enumerable set of variables.

Definition 3.1 An *equation over* Ω *and* $\mathcal{V}$ is a pair (t, s), where $t, s \in T\langle\Omega\rangle(\mathcal{V})$. $\bigoplus$

For the time being we do not follow the usual convention of denoting an equation (t, s) by $t = s$, because we want to keep for a moment also on the syntactic level the difference between an equation and the binary equality-relation. Of course, later we will identify (t, s) with $t = s$.

Definition 3.2 Let E be a finite set of equations over Ω and $\mathcal{V}$. The *E-equality,* denoted by $=_E$, is the finest Ω-congruence over $T\langle\Omega\rangle(\mathcal{V})$ containing every pair $(\psi(t), \psi(s))$, where $(t, s) \in E$ and ψ is an arbitrary $(\mathcal{V}, \Omega)$-substitution. If $t =_E s$, then t and s are called *E-equal.* $\bigoplus$

Now we are able to recall the definition of E-unification of two terms t and s.

Definition 3.3 (cf. [Sie89] page 220) Let E be a finite set of equations over Ω and $\mathcal{V}$.

- Two terms $t, s \in T\langle\Omega\rangle(\mathcal{V})$ are called *E-unifiable,* if there exists a $(\mathcal{V}, \Omega)$-substitution φ such that $\varphi(t) =_E \varphi(s)$.
- The set $\{\varphi \mid \varphi(t) =_E \varphi(s)\}$ is called the *set of E-unifiers of t and s,* and it is denoted by $\mathcal{U}_E(t, s)$. $\bigoplus$

In the following example we present the set $E_{\mathcal{R}_1}$ of equations which is induced by the term rewriting system $\mathcal{R}_1$ in Figure 1, and an $E_{\mathcal{R}_1}$-unifier of two terms.

Example 3.4 The set $E_{\mathcal{R}_1}$ induced by $\mathcal{R}_1$ consists of the following equations:

$$\begin{aligned} sh(\alpha, y_1) &= y_1 \\ sh(\sigma(x_1, x_2), y_1) &= sh(x_1, \sigma(mi(x_2), y_1)) \\ mi(\alpha) &= \alpha \\ mi(\sigma(x_1, x_2)) &= \sigma(mi(x_2), mi(x_1)) \end{aligned}$$

The substitution $\varphi = [z_1/\sigma(\alpha, \alpha), z_2/\alpha]$ is an $E_{\mathcal{R}_1}$-unifier of the terms $sh(z_1, \alpha)$ and $mi(\sigma(z_2, \alpha))$, because $sh(\sigma(\alpha, \alpha), \alpha) =_{E_{\mathcal{R}_1}} mi(\sigma(\alpha, \alpha))$. $\bigoplus$

Considering the set $\mathcal{U}_E(t, s)$ of all E-unifiers of t and s, immediately the question arises whether two unifiers φ_1 and φ_2 in $\mathcal{U}_E(t, s)$ are related. To answer this question, we need a tool to compare unifiers.

Definition 3.5 (cf. [Sie89] page 220) Let E be a finite set of equations over Ω and $\mathcal{V}$. The *instantiation preorder* $\preceq_E$ is defined over $T\langle\Omega\rangle(\mathcal{V})$ by:

$$t \preceq_E s, \text{ if there exists a } (\mathcal{V},\Omega)\text{-substitution } \varphi \text{ such that } \varphi(t) =_E s.$$

Let V be a finite subset of $\mathcal{V}$. We define a preorder $\preceq_E (V)$ on $(\mathcal{V},\Omega)$-substitutions by

$\varphi \preceq_E \varphi' \ (V)$, if there exists a $(\mathcal{V},\Omega)$-substitution ψ such that for every $x \in V$: $\psi(\varphi(x)) =_E \varphi'(x)$. $\oplus$

If we consider the case where E is the empty set, then E-unification reduces to (usual) unification [Rob65]; two terms t and s are unifiable, if there is a substitution φ such that $\varphi(t) = \varphi(s)$. It is decidable whether two terms are unifiable yes or no, and there are a couple of algorithms which produce such unifiers (cf. [Sie89] for a survey). In fact, for every two terms t and s, the set $\mathcal{U}_\emptyset(t,s)$ contains a smallest element with respect to $\preceq_\emptyset$ which is called the *most general unifier* of t and s.

If E is an arbitrary set of equations, then the situation is different. In general, it is not decidable whether, for a set E of equations and two terms t and s, t and s are E-unifiable yes or no (cf., e.g., [HO80]). Moreover, $\mathcal{U}_E(t,s)$ needs not contain a smallest element, but clearly there are minimal elements.

Actually, one does not have to consider all the elements of $\mathcal{U}_E(t,s)$ when studying E-unification. Rather it suffices to consider the elements of so called *complete sets of E-unifiers of t and s.* The set of minimal elements of $\mathcal{U}_E(t,s)$ is always a subset of such a complete set of E-unifiers. In fact, the minimal complete set of E-unifiers is exactly the set of minimal E-unifiers.

Definition 3.6 (cf. [HO80] page 359) Let E be a finite set of equations over Ω and $\mathcal{V}$. Let $t,s \in T\langle\Omega\rangle(\mathcal{V})$ and let W be a finite set of variables containing $V = \mathcal{V}(t) \cup \mathcal{V}(s)$. A set S of $(\mathcal{V},\Omega)$-substitutions is a *complete set of E-unifiers of t and s away from W,* if the following three conditions hold:

1. For every $\varphi \in S$: $\mathcal{D}(\varphi) \subseteq V$ and $\mathcal{I}(\varphi) \cap W = \emptyset$.

2. $S \subseteq \mathcal{U}_E(t,s)$.

3. For every $\varphi \in \mathcal{U}_E(t,s)$ there is a $\psi \in S$ such that $\psi \preceq_E \varphi \ (V)$.

The set is said to be *minimal,* if it satifies the additional condition (4).

4. For every $\varphi, \varphi' \in S$: if $\varphi \neq \varphi'$, then $\varphi \not\preceq_E \varphi' \ (V)$. $\oplus$

In Figure 4 we illustrate the notions of Definition 3.6. We suppose that $\varphi_1,\ldots,\varphi_9$ are the E-unifiers of the terms t and s. If $\varphi_i \preceq_E \varphi_j \ (V)$, then there exists a tour between φ_i and φ_j, and φ_j is written above φ_i. The set $\{\varphi_1,\varphi_2\}$ is the minimal complete set of E-unifiers of t and s, and the set $\{\varphi_1,\varphi_2,\varphi_4,\varphi_6,\varphi_7\}$ is a complete set of E-unifiers of t

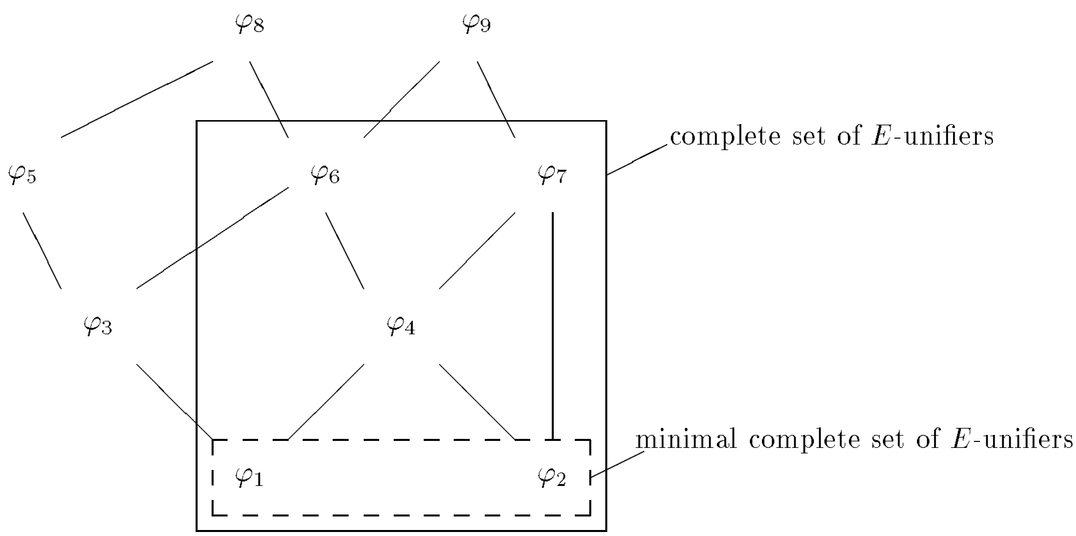

Figure 4: Complete Sets of E-Unifiers.

and s. Any set that does not contain one of the substitutions φ_1 and φ_2 is not a complete set of E-unifiers of t and s.

At the end of this section we present a theorem which implies a naive universal unification algorithm for the computation of a complete set of E-unifiers of two terms t and s, where E is any equational theory. In fact, this algorithm is only an application of the definition of E-unifiers.

Theorem 3.7 Let E be a finite set of equations over Ω and $\mathcal{V}$, let $t, s \in T\langle\Omega\rangle(\mathcal{V})$, and let W be a finite set of variables containing $V = \mathcal{V}(t) \cup \mathcal{V}(s)$. Let S be the set of $(\mathcal{V}, \Omega)$-substitutions φ such that φ is in S iff the following two conditions hold:

1. $\varphi(t) =_E \varphi(s)$.
2. $D(\varphi) \subseteq V$ and $\mathcal{I}(\varphi) \cap W = \emptyset$.

Then S is a complete set of E-unifiers away from W.

Proof: From Definition 3.3 follows that the set of all substitutions which satisfy Condition 1, is the set $\mathcal{U}_E(t, s)$. Thus, Conditions 2 and 3 in Definition 3.6 are satisfied. Furthermore, Condition 1 in Definition 3.6 is exactly the same as Condition 2 in the construction of S. Thus, S is a complete set of E-unifiers away from W. $\oplus$

The algorithm implied by Theorem 3.7 consists of first guessing a substitution φ and second checking whether $\varphi(s) =_E \varphi(t)$. This is very inefficient, because many substitutions that are not E-unifiers, are guessed and subsequently checked. Furthermore, it is not easy to check whether $\varphi(s) =_E \varphi(t)$ yes or no, because this involves the construction of the congruence relation $=_E$.

4 Term Rewriting Systems

In order to present a more efficient universal unification algorithm for the class $\mathcal{E}$, in particular, to avoid the construction of the congruence relation $=_E$, we restrict $\mathcal{E}$ to the class of equational theories characterized by canonical term rewriting systems. For an arbitrary term rewriting system $\mathcal{R}$, we denote by $E_\mathcal{R}$ the set of equations that results from replacing every rewrite rule $l \to r$ by the equation $l = r$. In [HO80] it is shown that $=_{E_\mathcal{R}}$ is equal to the transitive, reflexive, symmetric closure $\Longleftrightarrow^*_\mathcal{R}$ of the reduction relation associated with $\mathcal{R}$. Furthermore, in a canonical term rewriting system $\mathcal{R}$, every term has its unique normal form. In [Hue80] it is shown that, for a canonical term rewriting system $\mathcal{R}$, two terms are related by $\Longleftrightarrow^*_\mathcal{R}$ iff their normal forms are equal. This yields the following simple test for $E_\mathcal{R}$-equality of two terms t and s: Compute the normal forms of t and s and check whether the normal forms are equal or not.

For a complete introduction of canonical term rewriting systems, we recall the definition of term rewriting systems from [HO80]. Afterwards, we define the reduction relation associated with a term rewriting system and the reduction calculus. Then we define canonical term rewriting systems and present a theorem which implies a universal unification algorithm for the class of equational theories which are induced by canonical term rewriting systems.

4.1 Term Rewriting Systems and the Reduction Calculus

We start this subsection by defining term rewriting systems.

Definition 4.1 A term rewriting system, denoted by $\mathcal{R}$, is a pair (Ω, R), where Ω is a ranked alphabet and R is a finite set of rules of the form $l \to r$ such that $l, r \in T\langle\Omega\rangle(\mathcal{V})$ and $\mathcal{V}(r) \subseteq \mathcal{V}(l)$. $\oplus$

Remark 4.2 With every term rewriting system $\mathcal{R} = (\Omega, R)$, a bijection $\pi : R \to [card(R)]$ is associated implicitely that describes an enumeration of R. $\oplus$

An example of a term rewriting system is illustrated in Figure 1. There the associated function π is given by the enumeration on the right of the rewrite rules (e.g., $\pi(mi(\alpha) \to \alpha) = 3$). In the following we always refer to this example and the involved enumeration.

With every term rewriting system $\mathcal{R}$, a reduction relation is associated. By means of this reduction relation, a term t derives to a term t', if there is an occurrence u in t such that the subterm of t at u is an instance of the left hand side l of a rule $l \to r$. Then t' results from replacing t/u by the corresponding instance of r.

Definition 4.3 Let $\mathcal{R} = (\Omega, R)$ be a term rewriting system and let $t \in T\langle\Omega\rangle(\mathcal{V})$.

- The *set of redex interfaces for $\mathcal{R}$ and t*, denoted by $redI(\mathcal{R}, t)$, is the set

$$\{(u, \varphi, l \to r) \mid u \in O(t) \text{ with } t/u \notin \mathcal{V}, \varphi \in Sub(\mathcal{V}, \Omega), l \to r \in R \text{ with } \varphi(l) = t/u\}.$$

- The *set of redex occurrences for $\mathcal{R}$ and t*, denoted by $redO(\mathcal{R}, t)$, is the set

$$\{u \mid (u, \varphi, l \to r) \in redI(\mathcal{R}, t)\}.$$

- The *reduction relation associated with $\mathcal{R}$*, denoted by $\Longrightarrow_{\mathcal{R}}$, is defined as follows: For every $t, s \in T\langle\Omega\rangle(\mathcal{V}) : t \Longrightarrow_{\mathcal{R}} s$, if the following two conditions hold:

  1. There is a redex interface $(u, \varphi, l \to r) \in redI(\mathcal{R}, t)$.
  2. $s = t[u \leftarrow \varphi(r)]$. ⊕

Recall, if $t \Longrightarrow_{\mathcal{R}} s$, then we say that t derives to s by $\Longrightarrow_{\mathcal{R}}$. If $\mathcal{R}$ is clear from the context, we write $\Longrightarrow$ instead of $\Longrightarrow_{\mathcal{R}}$. We use the standard notation $\Longleftrightarrow$ to denote the symmetric closure of $\Longrightarrow$. Note that we sometimes use components of the redex interface as indices for $\Longrightarrow_{\mathcal{R}}$ to notice the redex occurrence, the substitution, or the applied rule. For instance, $\Longrightarrow_{\mathcal{R},u,\varphi,l\to r}$ denotes the reduction step in Definition 4.3. Pieces of the redex interface can be dropped, if they are not relevant. Furthermore, we often replace the applied rule by its number. A derivation by $\Longrightarrow_{\mathcal{R}_1}$ where $\mathcal{R}_1$ is the term rewriting system in Figure 1, is illustrated in the following example.

Example 4.4 Let $\mathcal{R}_1$ be the term rewriting system in Figure 1 and let t be the term

$$sh(\sigma(\alpha, z_1), mi(\sigma(\alpha, z_1))).$$

Then, the following derivation is a derivation by $\Longrightarrow_{\mathcal{R}_1}$ starting with t.

$$\begin{array}{ll}
 & sh(\sigma(\alpha, z_1), mi(\sigma(\alpha, z_1))) \\
\Longrightarrow_{\mathcal{R}_1,\Lambda,(2)} & sh(\alpha, \sigma(mi(z_1), mi(\sigma(\alpha, z_1)))) \\
\Longrightarrow_{\mathcal{R}_1,22,(4)} & sh(\alpha, \sigma(mi(z_1), \sigma(mi(z_1), mi(\alpha)))) \\
\Longrightarrow_{\mathcal{R}_1,222,(3)} & sh(\alpha, \sigma(mi(z_1), \sigma(mi(z_1), \alpha))) \\
\Longrightarrow_{\mathcal{R}_1,\Lambda,(1)} & \sigma(mi(z_1), \sigma(mi(z_1), \alpha))
\end{array}$$

The term $\sigma(mi(z_1), \sigma(mi(z_1), \alpha))$ is a normal form of t. Note that this is not the only derivation by $\Longrightarrow_{\mathcal{R}_1}$ starting with t, e.g., in the first reduction step we can also reduce at occurrence 2 by rule (4). ⊕

As described in the previous example, the reduction relation is nondeterministic: there may be more than one redex occurrence in a term t and, for the same redex occurrence in t, the set $redI(\mathcal{R}, t)$ may contain more than one redex interface. All possible ways to derive t by $\Longrightarrow_{\mathcal{R}}$ are collected in the concept of *reduction tree* which is based on the concept of *reduction calculus of* $\mathcal{R}$. Reduction tree and reduction calculus are instances of the concepts of derivation tree and indexed derivation calculus, respectively (cf. Section 2). We choose the set $\mathbb{N}^* \times R$ as index-set, because the two forms of nondeterminism depend on the occurrences and the applied rules. As total order we define a combination of the lexicographical order on $\mathbb{N}^*$ and the total order induced by the enumeration π on R.

Definition 4.5 Let $\mathcal{R} = (\Omega, R)$ be a term rewriting system with enumeration π on R.

- The *reduction calculus of* $\mathcal{R}$, denoted by $\mathcal{D}_{red\mathcal{R}}$, is the indexed derivation calculus $(T\langle\Omega\rangle(\mathcal{V}), (\mathbb{N}^* \times R, <), (\Longrightarrow_{\mathcal{R},u,l\to r} \mid (u, l \to r) \in \mathbb{N}^* \times R))$ for $(T\langle\Omega\rangle(\mathcal{V}), \Longrightarrow_{\mathcal{R}})$ where $<$ is the total order on $\mathbb{N}^* \times R$ defined as follows: for every $(u, l \to r), (u', l' \to r') \in \mathbb{N}^* \times R$, $(u, l \to r) < (u', l' \to r')$, if one of the following conditions holds:
  1. $u <_{lex} u'$.
  2. $u = u'$ and $\pi(l \to r) < \pi(l' \to r')$.
- A *reduction tree of* $\mathcal{R}$ is a derivation tree of $\mathcal{D}_{red\mathcal{R}}$. $\oplus$

For every term rewriting system $\mathcal{R}$, we consider the related set $E_{\mathcal{R}}$ of equations.

Definition 4.6 Let $\mathcal{R} = (\Omega, R)$ be a term rewriting system. The set $\{l = r \mid l \to r \in R\}$, denoted by $E_{\mathcal{R}}$, is the *set of equations related to* $\mathcal{R}$. $\oplus$

The set $E_{\mathcal{R}_1}$ of equations related to the term rewriting system in Figure 1 is shown in Example 3.4. In the following lemma we recall from [HO80] the connection between the transitive, reflexive, symmetric closure of $\Longrightarrow_{\mathcal{R}}$ and the $E_{\mathcal{R}}$-equality .

Lemma 4.7 (cf. [HO80] page 362) Let $\mathcal{R}$ be a term rewriting system and let $E_{\mathcal{R}}$ be the related set of equations.

$$\Longleftrightarrow^*_{\mathcal{R}} \; = \; =_{E_{\mathcal{R}}}$$

$\oplus$

This lemma will be very important in the following subsection. There we consider canonical term rewriting systems $\mathcal{R}$ and present a theorem which implies a more efficient algorithm for $E_{\mathcal{R}}$-unification than the algorithm implied by Theorem 3.7.

4.2 Canonical Term Rewriting Systems

For the definition of a canonical term rewriting system, we need the definitions of confluence and termination.

Definition 4.8 Let $\mathcal{R} = (\Omega, R)$ be a term rewriting system.

1. $\mathcal{R}$ is *confluent,* if for every $s, t, t' \in T\langle\Omega\rangle(\mathcal{V})$: $s \Longrightarrow^*_{\mathcal{R}} t$ and $s \Longrightarrow^*_{\mathcal{R}} t'$ implies that there is some $s' \in T\langle\Omega\rangle(\mathcal{V})$ such that $t \Longrightarrow^*_{\mathcal{R}} s'$ and $t' \Longrightarrow^*_{\mathcal{R}} s'$.
2. $\mathcal{R}$ is *noetherian,* if no infinite reduction derivation $t \Longrightarrow_{\mathcal{R}} t_1 \Longrightarrow_{\mathcal{R}} t_2 \Longrightarrow_{\mathcal{R}} \cdots$ exists.
3. $\mathcal{R}$ is *canonical,* if $\mathcal{R}$ is confluent and noetherian. $\oplus$

The term rewriting system in Figure 1 is canonical, because it is a macro tree transducer; and macro tree transducers are canonical term rewriting systems (cf. [EV85, FHVV93]).

The following two lemmas are the main foundations of our further investigations.

Lemma 4.9 Let $\mathcal{R} = (\Omega, R)$ be a canonical term rewriting system. Every term $t \in T\langle\Omega\rangle(\mathcal{V})$ has a unique normal form which is called the *$\mathcal{R}$-normal form of t.*

Proof: The existence of a normal form of t follows from the fact that $\mathcal{R}$ is noetherian. Suppose, that t' and t'' are two normal forms of t. By the confluence of $\mathcal{R}$ we obtain that there exists an s such that $t' \Longrightarrow_{\mathcal{R}}^* s$ and $t'' \Longrightarrow_{\mathcal{R}}^* s$. This implies that $t' = s = t''$, because t' and t'' are in normal form. $\oplus$

The $\mathcal{R}$-normal form of t is denoted by $nf_{\mathcal{R}}(t)$. In [Hue80] it is shown that two terms are related by $\Longleftrightarrow_{\mathcal{R}}^*$ iff their normal forms are equal. From this fact and from Lemma 4.7, the following lemma follows immediately.

Lemma 4.10 Let $\mathcal{R} = (\Omega, R)$ be a canonical term rewriting system and let $t, s \in T\langle\Omega\rangle(\mathcal{V})$.

$$t =_{E_{\mathcal{R}}} s \text{ iff } nf_{\mathcal{R}}(t) = nf_{\mathcal{R}}(s).$$

$\oplus$

By applying Lemma 4.10 we show that the substitution φ in Example 3.4 is really an $E_{\mathcal{R}_1}$-unifier.

Example 4.11 Let $\mathcal{R}_1$ be the term rewriting system in Figure 1, let $t = sh(z_1, \alpha)$, let $s = mi(\sigma(z_2, \alpha))$, and let $\varphi = [z_1/\sigma(\alpha, \alpha), z_2/\alpha]$. Then there exist the following two derivations by $\Longrightarrow_{\mathcal{R}_1}$ starting with $\varphi(t)$ and $\varphi(s)$, respectively, which yield the same normal forms.

$$\begin{aligned}
\varphi(t) = sh(\sigma(\alpha,\alpha),\alpha) &\Longrightarrow_{\mathcal{R}_1,\Lambda,(2)} sh(\alpha, \sigma(mi(\alpha), \alpha)) \\
&\Longrightarrow_{\mathcal{R}_1,\Lambda,(1)} \sigma(mi(\alpha), \alpha) \\
&\Longrightarrow_{\mathcal{R}_1,1,(3)} \underline{\sigma(\alpha, \alpha)}
\end{aligned}$$

$$\begin{aligned}
\varphi(s) = mi(\sigma(\alpha,\alpha)) &\Longrightarrow_{\mathcal{R}_1,\Lambda,(4)} \sigma(mi(\alpha), mi(\alpha)) \\
&\Longrightarrow_{\mathcal{R}_1,2,(3)} \sigma(mi(\alpha), \alpha) \\
&\Longrightarrow_{\mathcal{R}_1,1,(3)} \underline{\sigma(\alpha, \alpha)}
\end{aligned}$$

Thus, the normal forms of $\varphi(t)$ and $\varphi(s)$ are equal. It follows from Lemma 4.10 and Definition 3.3 that φ is an $E_{\mathcal{R}_1}$-unifier of t and s. $\oplus$

We finish this section by presenting the theorem which implies a universal unification algorithm for equational theories $=_{E_{\mathcal{R}}}$ where $\mathcal{R}$ is a canonical term rewriting system.

Theorem 4.12 Let $\mathcal{R} = (\Omega, R)$ be a canonical term rewriting system, let $t, s \in T\langle\Omega\rangle(\mathcal{V})$, and let W be a finite set of variables containing $V = \mathcal{V}(t) \cup \mathcal{V}(s)$. Let S be the set of $(\mathcal{V}, \Omega)$-substitutions φ such that φ is in S iff the following two conditions hold:

1. $nf_{\mathcal{R}}(\varphi(t)) = nf_{\mathcal{R}}(\varphi(s))$.
2. $D(\varphi) \subseteq V$ and $\mathcal{I}(\varphi) \cap W = \emptyset$.

Then S is a complete set of $E_{\mathcal{R}}$-unifiers away from W.

Proof: From Condition 1 and Lemma 4.10 follows $\varphi(t) =_{E_{\mathcal{R}}} \varphi(s)$. Thus, from Theorem 3.7 follows S is a complete set of $E_{\mathcal{R}}$-unifiers away from W. $\oplus$

The algorithm implied by Theorem 4.12 consists of guessing a substitution φ and checking whether $\varphi(s) =_{E_{\mathcal{R}}} \varphi(t)$. It is more efficient than the algorithm implied by Theorem 3.7, because the test on $E_{\mathcal{R}}$-equality is realized by computing the $\mathcal{R}$-normal forms of $\varphi(s)$ and $\varphi(t)$ and by checking their equality. This is always a finite process. Nevertheless, this algorithm is very inefficient, because still many substitutions are checked which are not $E_{\mathcal{R}}$-unifiers.

5 $E_{\mathcal{R}}$-Unification by Narrowing

In this section we eliminate the deficiency in the algorithm that is implied by Theorem 4.12, by recalling an algorithm from [Hul80] by means of which a substitution φ is computed step by step during the derivation. Actually, it is not a derivation by the reduction relation any more, but it is a derivation by the narrowing relation. Roughly speaking, the composition of the substitutions $\varphi_1, \varphi_2, \varphi_3, \ldots, \varphi_n$ which are involved in the derivation steps by the narrowing relation, composed with the most general unifier μ of the two terms at the end of the derivation, constitute the desired substitution φ.

The intention of Hullot's algorithm is very similar to the intention of the resolution principle for logic programs in [Rob65]: Hullot's algorithm is superior to "guessing φ and computing $nf_{\mathcal{R}}(\varphi(s))$" in the same way as resolution is superior to level saturation (cf., e.g., [HK89]). The advantage of Hullot's algorithm is the fact that a substitution φ_i is chosen only if φ_i allows for a derivation step, i.e., φ_i must be the most general unifier of a subtree of the current derivation form and the right hand side of a rule. Thereby the set of all possible substitutions is reduced which leads to a more efficient $E_{\mathcal{R}}$-unification algorithm.

We start this section by introducing the narrowing derivation calculus which is needed in Hullot's algorithm. Hullot's algorithm is defined only for canonical term rewriting systems, but we will define the narrowing derivation calculus for arbitrary term rewriting systems, because the restriction to canonical term rewriting systems is not necessary in its definition.

5.1 The Narrowing Calculus

A term t derives by the reduction relation to a term t', if there exists a redex interface $(u, \varphi, l \rightarrow r)$ such that t/u and $\varphi(l)$ are equal. In the narrowing relation, we take the most general unifier φ of t/u and $\rho(l)$ where ρ is a variable renaming such that $\mathcal{V}(\rho(l)) \cap \mathcal{V}(t) = \emptyset$. That means, the condition $\varphi(\rho(l)) = \varphi(t/u)$ must hold. Moreover, φ must be applied to r and to the context of t/u, too.

In order to keep track of the substitutions which have occurred in previous narrowing derivation steps, every derivation form of the narrowing relation is a pair (t, ψ) where t is a term and ψ is a substitution. Roughly speaking, ψ comprises the composition of all most general unifiers of previous derivation steps by the narrowing relation. Clearly, we are only interested in substitutions of variables that occur in t and we are not interested in substitutions of variables that occur in the left hand side of a rule. Thus, if (t, ψ) derives to (t', ψ') by the narrowing relation, then ψ' is the composition of ψ and the restriction of the most general unifier φ to the set of variables in t.

Definition 5.1 Let $\mathcal{R} = (\Omega, R)$ be a term rewriting system and let $t \in T\langle\Omega\rangle(\mathcal{V})$.

- The *set of narrowing interfaces for $\mathcal{R}$ and t,* denoted by $narI(\mathcal{R}, t)$, is the set

$\{(u, \varphi, l \rightarrow r, \rho) \mid u \in O(t) : t/u \notin \mathcal{V}, l \rightarrow r \in R, \rho$ is a renaming of variables in l such that $\mathcal{V}(\rho(l)) \cap \mathcal{V}(t) = \emptyset, \varphi \in Sub(\mathcal{V}, \Omega)$ is the most general unifier of $\rho(l)$ and $t/u\}$.

- The *set of narrowing occurrences for* $\mathcal{R}$ *and* t, denoted by $narO(\mathcal{R}, t)$, is the set

$$\{u \mid (u, \varphi, l \rightarrow r, \rho) \in narI(\mathcal{R}, t)\}.$$

- The *narrowing relation associated with* $\mathcal{R}$, denoted by $\rightsquigarrow_{\mathcal{R}}$, is defined as follows: For every $t, s \in T\langle\Omega\rangle(\mathcal{V})$ and $\psi, \psi' \in Sub(\mathcal{V}, \Omega) : (t, \psi) \rightsquigarrow_{\mathcal{R}} (s, \psi')$, if the following three conditions hold:

  1. There is a narrowing interface $(u, \varphi, l \rightarrow r, \rho) \in narI(\mathcal{R}, t)$.
  2. $s = \varphi(t[u \leftarrow \rho(r)])$
  3. $\psi' = \psi \circ (\varphi|_{\mathcal{V}(t)})$ $\oplus$

In the following, we use the notations for the narrowing relation in analogy to the notations for the reduction relation. In Example 5.2 we show a derivation by the narrowing relation that starts with the same term t as the derivation by the reduction relation in Example 4.4 and where derivation steps 1 - 6 are analog to the derivation steps by the reduction relation in Example 4.4.

Example 5.2 Let $\mathcal{R}_1$ be the term rewriting system in Figure 1 and let t be the term

$$sh(\sigma(\alpha, z_1), mi(\sigma(\alpha, z_1))).$$

Then, the following derivation is a derivation by $\rightsquigarrow_{\mathcal{R}_1}$ starting with $(t, \varphi_\emptyset)$.

$$\begin{array}{lll}
 & (sh(\sigma(\alpha, z_1), mi(\sigma(\alpha, z_1))), \varphi_\emptyset) & \\
\rightsquigarrow^4_{\mathcal{R}_1} & (\sigma(mi(z_1), \sigma(mi(z_1), \alpha)), \varphi_\emptyset) & \\
\rightsquigarrow_{\mathcal{R}_1, 21, [z_1/\alpha], (3)} & (\sigma(mi(\alpha), \sigma(\alpha, \alpha)), [z_1/\alpha]) & (*) \\
\rightsquigarrow_{\mathcal{R}_1, 1, \varphi_\emptyset, (3)} & (\sigma(\alpha, \sigma(\alpha, \alpha)), [z_1/\alpha]) &
\end{array}$$

Remark that in narrowing step (*) also rule (4) can be applied by the substitution $[z_1/\sigma(z_2, z_3)]$. $\oplus$

As illustrated in the introduction and in Example 5.2, the narrowing relation is nondeterministic. There are the same two forms of nondeterminism in the narrowing relation as in the reduction relation, i.e., (1) there may be more than one narrowing occurrence and (2) more than one rule may be applied at one narrowing occurrence. For the purpose of enumerating all the possibilities, we introduce *narrowing trees* which are derivation trees of the *narrowing calculus.* In the definition of the narrowing calculus we choose the same index set and the same total order as in the reduction calculus.

Definition 5.3 Let $\mathcal{R} = (\Omega, R)$ be a term rewriting system with enumeration π on R.

- The *narrowing calculus of* $\mathcal{R}$, denoted by $\mathcal{D}_{nar\mathcal{R}}$, is the indexed derivation calculus $(T\langle\Omega\rangle(\mathcal{V}) \times Sub(\mathcal{V},\Omega), (\mathbb{N}^* \times R, <), (\leadsto_{\mathcal{R},u,l\rightarrow r} \mid (u, l \rightarrow r) \in \mathbb{N}^* \times R))$ for $(T\langle\Omega\rangle(\mathcal{V}) \times Sub(\mathcal{V},\Omega), \leadsto_{\mathcal{R}})$ where $<$ is the total order on $\mathbb{N}^* \times R$ defined as follows: for every $(u, l \rightarrow r), (u', l' \rightarrow r') \in \mathbb{N}^* \times R$, $(u, l \rightarrow r) < (u', l' \rightarrow r')$, if one of the following conditions holds:
  1. $u <_{lex} u'$.
  2. $u = u'$ and $\pi(l \rightarrow r) < \pi(l' \rightarrow r')$.
- A *narrowing tree of* $\mathcal{R}$ is a derivation tree of $\mathcal{D}_{nar\mathcal{R}}$. $\oplus$

5.2 Hullot's $E_{\mathcal{R}}$-Unification

In [Hul80] Hullot presents a theorem which says that, for every canonical term rewriting system $\mathcal{R}$, the $E_{\mathcal{R}}$-unifiability of two terms t and s can be checked nondeterministically by narrowing and unification. He starts the derivation by the narrowing relation with the pair $(equ(t,s), \varphi_{\emptyset})$ where $equ \notin \Omega \cup \mathcal{V}$ is a new binary symbol (in [Hul80] the symbol equ is denoted by H). Hullot shows that t and s are $E_{\mathcal{R}}$-unifiable, if $(equ(t,s), \varphi_{\emptyset})$ derives to some pair $(equ(t_n, s_n), \varphi_n)$ by $\leadsto^*_{\mathcal{R}}$, and t_n and s_n are unifiable. (We assume that $\leadsto_{\mathcal{R}}$ is extended in an obvious way to objects of the form $(equ(t,s), \varphi)$ for $t, s \in T\langle\Omega\rangle(\mathcal{V})$.) If μ is the most general unifier of t_n and s_n, then $\varphi_n \circ \mu$ is a substitution which satisfies the condition for φ in Theorem 4.12, i.e., $nf_{\mathcal{R}}(\mu(\varphi_n(t))) = nf_{\mathcal{R}}(\mu(\varphi_n(s)))$, and hence, $\varphi_n \circ \mu$ is an $E_{\mathcal{R}}$-unifier of t and s. In the following example we show, how the $E_{\mathcal{R}_1}$-unifier in Example 4.11 can be computed by Hullot's method.

Example 5.4 Let $\mathcal{R}_1$ be the term rewriting system in Figure 1, let $t = sh(z_1, \alpha)$, and let $s = mi(\sigma(z_2, \alpha))$. Then there exists the following derivation by $\leadsto_{\mathcal{R}_1}$ starting with $(equ(t,s), \varphi_{\emptyset})$ (also cf. Figure 5).

$$
\begin{array}{ll}
 & (equ(sh(z_1,\alpha), mi(\sigma(z_2,\alpha))), \varphi_{\emptyset}) \\
\leadsto_{\mathcal{R}_1,1,(2)} & (equ(sh(z_3,\sigma(mi(z_4),\alpha)), mi(\sigma(z_2,\alpha))), [z_1/\sigma(z_3,z_4)]) \\
\leadsto_{\mathcal{R}_1,1,(1)} & (equ(\sigma(mi(z_4),\alpha), mi(\sigma(z_2,\alpha))), [z_1/\sigma(\alpha,z_4)]) \\
\leadsto_{\mathcal{R}_1,11,(3)} & (equ(\sigma(\alpha,\alpha), mi(\sigma(z_2,\alpha))), [z_1/\sigma(\alpha,\alpha)]) \\
\leadsto_{\mathcal{R}_1,2,(4)} & (equ(\sigma(\alpha,\alpha), \sigma(mi(\alpha), mi(z_2))), [z_1/\sigma(\alpha,\alpha)]) \\
\leadsto_{\mathcal{R}_1,21,(3)} & (equ(\sigma(\alpha,\alpha), \sigma(\alpha, mi(z_2))), [z_1/\sigma(\alpha,\alpha)) \\
\leadsto_{\mathcal{R}_1,22,(3)} & (equ(\sigma(\alpha,\alpha), \sigma(\alpha,\alpha)), [z_1/\sigma(\alpha,\alpha), z_2/\alpha])
\end{array}
$$

The two subterms of the first component in the result of the last derivation step are equal. Thus, $\varphi_{\emptyset}$ is their most general unifier, and hence the substitution $\varphi = [z_1/\sigma(\alpha,\alpha), z_2/\alpha]$ is an $E_{\mathcal{R}_1}$-unifier of t and s. $\oplus$

Hullot's method is also a method to construct a complete set of $E_{\mathcal{R}}$-unifiers of t and s.

Theorem 5.5 (cf. Theorem 2 of [Hul80]) Let $\mathcal{R} = (\Omega, R)$ be a canonical term rewriting system, let $t, s \in T\langle\Omega\rangle(\mathcal{V})$, and let V be the set $\mathcal{V}(t) \cup \mathcal{V}(s)$. Let S be the set of all $(\mathcal{V}, \Omega)$-substitutions φ such that φ is in S iff there exists a derivation by $\rightsquigarrow_{\mathcal{R}}$ of the form:

$$(equ(t,s), \varphi_\emptyset) \rightsquigarrow_{\mathcal{R}} (equ(t_1,s_1), \varphi_1) \rightsquigarrow_{\mathcal{R}} (equ(t_2,s_2), \varphi_2) \rightsquigarrow_{\mathcal{R}} \cdots \rightsquigarrow_{\mathcal{R}} (equ(t_n,s_n), \varphi_n),$$

where for every $i \in [n] : \varphi_i$ is in normal form, t_n and s_n are unifiable with most general unifier μ, and $\varphi = (\varphi_n \circ \mu)|_V$. Then S is a complete set of $E_{\mathcal{R}}$-unifiers of t and s away from V. $\oplus$

In Figure 5 a narrowing tree which is associated to a computation of the algorithm implied by Theorem 5.5, is shown. We call trees of this form *Hullot's narrowing trees.* The nodes at the front of such trees have the form $(equ(t', s'), \varphi)$. Every such leaf for which t' and s' are unifiable, yields an $E_{\mathcal{R}}$-unifier. Thus, for the computation of an $E_{\mathcal{R}}$-unifier the branches are lengthened by the unification. However, to reduce the space of the figure, we omit the substitutions in the nodes of Hullot's narrowing tree in Figure 5.

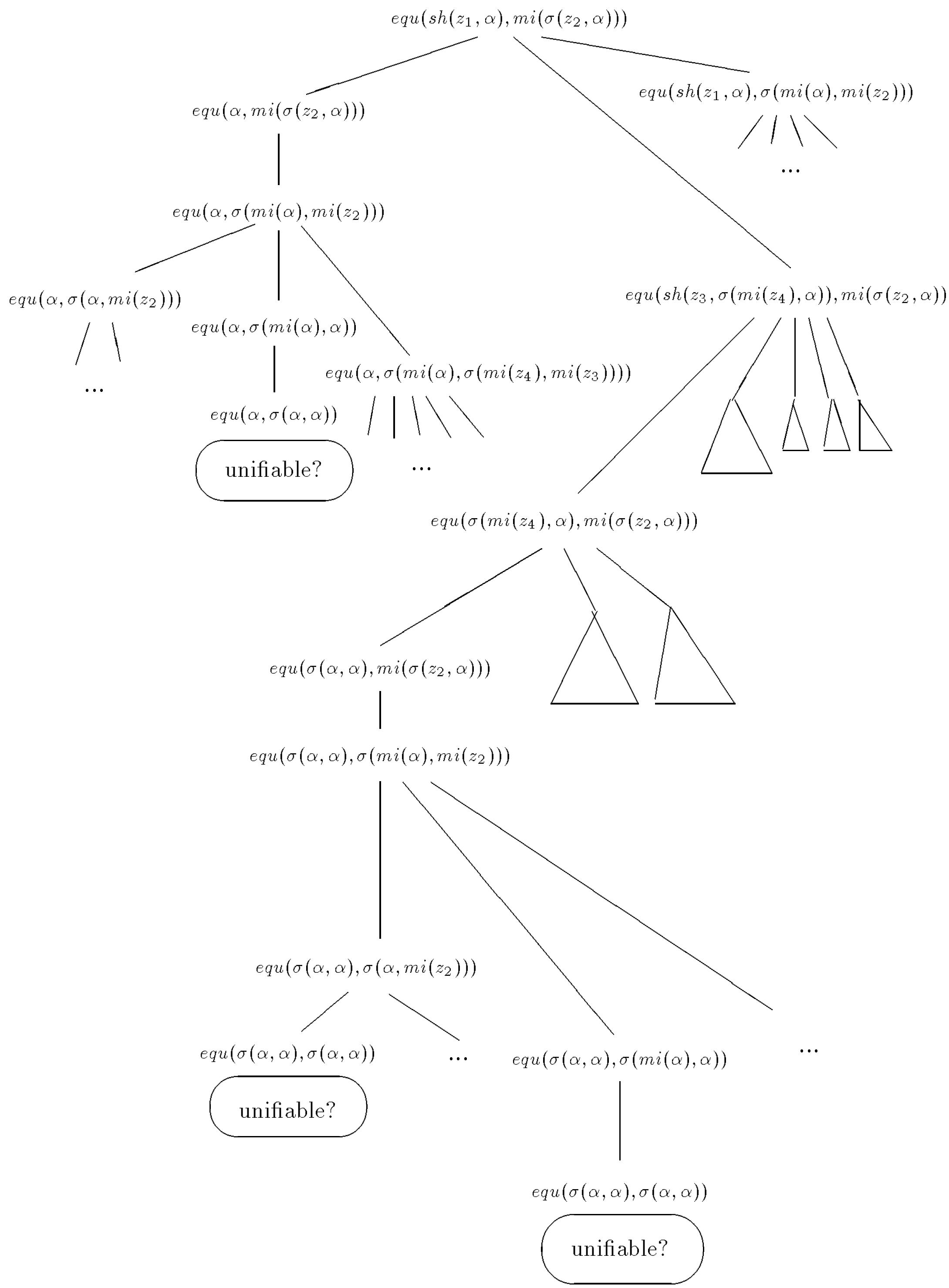

Figure 5: Hullot's narrowing tree.

6 $E_{\mathcal{R}}$-Unification by Leftmost Outermost Narrowing

In this section we increase the efficiency of the universal unification algorithm implied by Theorem 5.5, by allowing narrowing derivations only at the leftmost outermost narrowing occurrence. By fixing one narrowing occurrence the breadth of Hullot's narrowing trees is reduced. Furthermore, we choose the leftmost outermost narrowing strategy, because it omits the evaluation of arguments which are deleted by a function call. Thus, also the depth of Hullot's narrowing trees is reduced. In [Ech88] it is shown that the universal unification algorithm presented in this section, computes a complete set of $E_{\mathcal{R}}$-unifiers only for a restricted class of canonical term rewriting systems. We call the term rewriting systems in this class ctn-trs's.

We start this section by introducing the leftmost outermost narrowing calculus. Then we define ctn-trs's and recall the universal unification algorithm from [Ech88].

6.1 The Leftmost Outermost Narrowing Calculus

In the leftmost outermost narrowing relation a pair (t, ψ) derives to a pair (t', ψ') at the minimal element of the set of narrowing occurrences in t.

Definition 6.1 Let $\mathcal{R} = (\Omega, R)$ be a term rewriting system and let $t \in T\langle\Omega\rangle(\mathcal{V})$.

- The *leftmost outermost narrowing occurrence for* $\mathcal{R}$ *and* t, denoted by $lo\text{-}narO(\mathcal{R}, t)$, is the narrowing occurrence $min_{lex} narO(\mathcal{R}, t)$.
- The *set of leftmost outermost narrowing interfaces for* $\mathcal{R}$ *and* t, denoted by $lo\text{-}narI(\mathcal{R}, t)$, is the set

$$\{(u, \varphi, l \rightarrow r, \rho) \mid (u, \varphi, l \rightarrow r, \rho) \in narI(\mathcal{R}, t) \text{ and } u = lo\text{-}narO(\mathcal{R}, t)\}.$$

- The *leftmost outermost narrowing relation associated with* $\mathcal{R}$, denoted by $\overset{lo}{\rightsquigarrow}_{\mathcal{R}}$, is defined as follows: For every $t, s \in T\langle\Omega\rangle(\mathcal{V})$ and $\psi, \psi' \in Sub(\mathcal{V}, \Omega)$: $(t, \psi) \overset{lo}{\rightsquigarrow}_{\mathcal{R}} (s, \psi')$, if the following three conditions hold:

  1. There is a leftmost outermost narrowing interface $(u, \varphi, l \rightarrow r, \rho) \in lo\text{-}narI(\mathcal{R}, t)$.
  2. $s = \varphi(t[u \leftarrow \rho(r)])$
  3. $\psi' = \psi \circ (\varphi|_{\mathcal{V}(t)})$ $\oplus$

It is obvious, that $\overset{lo}{\rightsquigarrow}_{\mathcal{R}} \subseteq \rightsquigarrow_{\mathcal{R}}$. Furthermore, in the leftmost outermost narrowing relation there only exists the nondeterminism of the second type, i.e, more than one rule can be applied at the leftmost outermost narrowing occurrence. Similar to the reduction relation and the narrowing relation, we also define an indexed derivation calculus for the leftmost outermost narrowing relation. This indexed derivation calculus is called the *leftmost outermost narrowing calculus.* We choose the set of rules as index set and the total order on it is implied by the enumeration π of the rules as required in Remark 4.2.

Definition 6.2 Let $\mathcal{R} = (\Omega, R)$ be a term rewriting system with enumeration π on R.

- The *leftmost outermost narrowing calculus of* $\mathcal{R}$, denoted by $\mathcal{D}_{lo-nar\mathcal{R}}$, is the indexed derivation calculus $(T\langle\Omega\rangle(\mathcal{V}) \times Sub(\mathcal{V}, \Omega), (R, <), (\stackrel{lo}{\leadsto}_{\mathcal{R}, l \to r} \mid l \to r \in R))$ for $(T\langle\Omega\rangle(\mathcal{V}) \times Sub(\mathcal{V}, \Omega), \stackrel{lo}{\leadsto}_{\mathcal{R}})$ where $<$ is the total order on R defined as follows: for every $l \to r, l' \to r' \in R$, $l \to r < l' \to r'$, if $\pi(l \to r) < \pi(l' \to r')$.
- A *leftmost outermost narrowing tree of* $\mathcal{R}$ is a derivation tree of $\mathcal{D}_{lo-nar\mathcal{R}}$. $\oplus$

Clearly, a leftmost outermost narrowing tree for a term t results from the narrowing tree for t by deleting the branches corresponding to narrowing derivations at other occurrences than the leftmost outermost one. In Figure 6 we illustrate the leftmost outermost narrowing tree for the term $equ(sh(z_1, \alpha), mi(\sigma(z_2, \alpha)))$ by recalling Hullot's narrowing tree from Figure 5 and shading the deleted areas. We call trees like the tree in Figure 6 *Hullot's leftmost outermost narrowing trees.*

Now, the question arises, whether Theorem 5.5 holds, if the narrowing relation is replaced by the leftmost outermost narrowing relation. In the following example it is illustrated that this question is answered by 'no'.

Example 6.3 (cf. [Ech88] Example 1). Let $\mathcal{R} = (\Omega, R)$ be a term rewriting system where $\Omega = \{f^{(2)}, \gamma^{(1)}, \alpha^{(0)}\}$ and let R contain the following rules:

$$\begin{array}{rcll} f(\alpha, \alpha) & \to & \alpha & (1) \\ f(\gamma(x), \alpha) & \to & \gamma(\alpha) & (2) \\ f(x, \gamma(y)) & \to & \gamma(\gamma(\alpha)) & (3) \end{array}$$

The only derivation by $\stackrel{lo}{\leadsto}_{\mathcal{R}}$ starting with the term $f(f(z_1, z_2), z_3)$ has the form

$$(f(f(z_1, z_2), z_3), \varphi_\emptyset) \stackrel{lo}{\leadsto}_{\mathcal{R}, \Lambda, [x/f(z_1, z_2), z_3/\gamma(y)], (3)} (\gamma(\gamma(\alpha)), [z_3/\gamma(y)]).$$

But there exists the following derivation by $\leadsto_{\mathcal{R}}$ starting with the term $f(f(z_1, z_2), z_3)$

$$\begin{array}{rl} (f(f(z_1, z_2), z_3), \varphi_\emptyset) \leadsto_{\mathcal{R}, 1, [z_1/\alpha, z_2/\alpha], (1)} & (f(\alpha, z_3), [z_1/\alpha, z_2/\alpha]) \\ \leadsto_{\mathcal{R}, \Lambda, [z_3/\alpha], (1)} & (\alpha, [z_1/\alpha, z_2/\alpha, z_3/\alpha]) \end{array}$$

Thus, from Theorem 5.5 follows that the substitution $[z_1/\alpha, z_2/\alpha, z_3/\alpha]$ is an $E_{\mathcal{R}}$-unifier of the terms $t = f(f(z_1, z_2), z_3)$ and $s = \alpha$. But this substitution is not in the set S which is constructed in Theorem 5.5, if the narrowing relation is replaced by the leftmost outermost narrowing relation. $\oplus$

In [Ech88] it is shown that the modification of Theorem 5.5 which is obtained by replacing the narrowing relation by the leftmost outermost narrowing relation, holds for canonical term rewriting systems that have the *property of free strategies.* We call these term rewriting systems *canonical, totally defined, not strictly sub-unifiable term rewriting systems,* for short: *ctn-trs.*

6.2 CTN-TRS

A ctn-trs $\mathcal{R} = (\Omega, R)$ is a canonical term rewriting system, where Ω is divided into two disjoint ranked alphabets, denoted by F and Δ. F is called the set of function symbols and Δ is called the set of working symbols or constructors. This partition is motivated by declarative programming languages. The left hand sides of the rewrite rules in R are linear; function symbols only occur at the root of a left hand side. Thus, ctn-trs's are constructor-based term rewriting systems (cf. [You89]). Furthermore, every function symbol in F is totally defined over its domain (cf. Definition 12 in [Ech88]), i.e., if a term is in normal form, then it is in $T\langle\Delta\rangle(\mathcal{V})$. For obtaining the completeness of the lo-narrowing relation, the left hand sides of the rules in R must be pairwise not strictly sub-unifiable. We recall the definitions of sub-unifiability and of strictly sub-unifiability from [Ech88].

Definition 6.4 (cf. [Ech88] Definition 10 and Definition 11). Let $t, t' \in T\langle\Omega\rangle(\mathcal{V})$.

- t and t' are *sub-unifiable,* if there exists an occurrence u in $O(t) \cap O(t')$ such that the following two conditions hold:
  1. t/u and $\rho(t'/u)$ are unifiable with most general unifier σ_u where ρ is a variable-renaming such that $\mathcal{V}(t/u) \cap \mathcal{V}(\rho(t'/u)) = \emptyset$.
  2. For all occurrences w with $w < u$, t/w and t'/w have the same label at the root.
- t and t' are *strictly sub-unifiable,* if there exists an occurrence u where t and t' are sub-unifiable and the corresponding most general unifier σ_u is neither a variable renaming nor the empty substitution. $\oplus$

Example 6.5

- In the term rewriting system $\mathcal{R}$ in Example 6.3, the left hand sides of rule 1 and rule 3 are strictly sub-unifiable at occurrence 1; the same holds for rule 2 and rule 3.
- The left hand sides of rule 1 and rule 2 are sub-unifiable at occurrence 2 but not strictly sub-unifiable, because the most general unifier σ_2 is the empty substitution.
- Let $\mathcal{R}' = (\Omega, R')$ be a term rewriting system where $\Omega = \{f^{(2)}, \gamma^{(1)}, \alpha^{(0)}\}$ and let R' contain the following rules:

$$\begin{array}{rcll} f(\alpha,\alpha) & \rightarrow & \alpha & (1) \\ f(\gamma(x),\alpha) & \rightarrow & \gamma(\alpha) & (2) \\ f(\alpha,\gamma(y)) & \rightarrow & \gamma(\gamma(\alpha)) & (3) \\ f(\gamma(x),\gamma(y)) & \rightarrow & \gamma(\gamma(\alpha)) & (4) \end{array}$$

  The left hand sides of the rules in R' are pairwise not strictly sub-unifiable. Furthermore, the left hand sides of the rules 2 and 3 are not sub-unifiable. $\oplus$

Now, we are able to define ctn-trs.

Definition 6.6 Let $\mathcal{R} = (\Omega, R)$ be a term rewriting system. $\mathcal{R}$ is a *canonical, totally defined, not strictly sub-unifiable term rewriting system,* for short *ctn-trs,* if the following conditions hold:

1. $\mathcal{R}$ is canonical.
2. $\Omega = F \cup \Delta$ and $F \cap \Delta = \emptyset$.
3. Every left hand side is linear in $\mathcal{V}$.
4. Every left hand side has the form $f(t_1, \ldots, t_n)$ where $f \in F^{(n)}$ and for every $i \in [n]$: $t_i \in T\langle\Delta\rangle(\mathcal{V})$.
5. For every $t \in T\langle\Omega\rangle(\mathcal{V}) : nf_{\mathcal{R}}(t) \in T\langle\Delta\rangle(\mathcal{V})$.
6. The left hand sides of the rewrite rules in R are pairwise not strictly sub-unifiable. $\oplus$

We note that, e.g., every modular tree transducer [EV91] is a ctn-trs; the class of modular tree transducer characterizes the class of primitive recursive tree functions [Hup78]. An example of a ctn-trs is shown in Figure 1. Remark, that $\mathcal{R}$ denotes the triple (F, Δ, R). As a second example, we present the description of the multiplication by a ctn-trs.

Example 6.7 The term rewriting system $\mathcal{R} = (F, \Delta, R)$, where $F = \{mult^{(2)}, add^{(2)}\}$, $\Delta = \{\gamma^{(1)}, \alpha^{(0)}\}$, and R contains the following rules:

$$\begin{array}{rcll} mult(\alpha, y) & \rightarrow & \alpha & (1) \\ mult(\gamma(x), y) & \rightarrow & add(y, mult(x, y)) & (2) \\ add(\alpha, y) & \rightarrow & y & (3) \\ add(\gamma(x), y) & \rightarrow & \gamma(add(x, y)) & (4) \end{array}$$

is a ctn-trs, because it is a modular tree transducer. $\oplus$

If we start from a ctn-trs $\mathcal{R}$ and we want to compute an $E_{\mathcal{R}}$-unifier of two terms t and s, then we are not interested in substitutions of the following form $[x/f(y)]$, where $f \in F$. For instance, if we have the ctn-trs in Example 6.7 and we want to compute $E_{\mathcal{R}}$-unifiers of the terms $add(x, y)$ and z, then we are not interested in the minimal $E_{\mathcal{R}}$-unifier $[z/add(x, y)]$. In fact, we are interested in $E_{\mathcal{R}}$-unifiers of which the images are elements of $T\langle\Delta\rangle$. For instance, we should be able to compute the $E_{\mathcal{R}}$-unifier $[z/\gamma(\gamma(\gamma(\alpha))), x/\gamma(\alpha), y/\gamma(\gamma(\alpha))]$. Such an $E_{\mathcal{R}}$-unifier is called a ground $(E_{\mathcal{R}}, \Delta)$-unifier.

Definition 6.8 Let $\mathcal{R} = (F, \Delta, R)$ be a ctn-trs, let $t, s \in T\langle F \cup \Delta\rangle(\mathcal{V})$, and let $\varphi \in \mathcal{U}_{E_{\mathcal{R}}}(t, s)$ be an $E_{\mathcal{R}}$-unifier of t and s.

- φ is an $(E_{\mathcal{R}}, \Delta)$-*unifier of* t *and* s, if $\varphi \in Sub(\mathcal{V}, \Delta)$.
- φ is a *ground* $(E_{\mathcal{R}}, \Delta)$-*unifier of* t *and* s, if $\varphi \in gSub(\mathcal{V}, \Delta)$.

The sets of $(E_\mathcal{R}, \Delta)$-unifiers and of ground $(E_\mathcal{R}, \Delta)$-unifiers of t and s are denoted by $\mathcal{U}_{(E_\mathcal{R},\Delta)}(t,s)$ and $g\mathcal{U}_{(E_\mathcal{R},\Delta)}(t,s)$, respectively. $\oplus$

Similar to the situation of E-unifiers of two terms t and s, we do not have to compute the whole set $g\mathcal{U}_{(E_\mathcal{R},\Delta)}(t,s)$, but rather an approximation of it. It suffices to compute a ground complete set of $(E_\mathcal{R}, \Delta)$-unifiers of t and s.

Definition 6.9 (cf. [Ech88] page 92) Let $\mathcal{R} = (F, \Delta, R)$ be a ctn-trs. Let $t, s \in T\langle F \cup \Delta\rangle(\mathcal{V})$ and let W be a finite set of variables containing $V = \mathcal{V}(t) \cup \mathcal{V}(s)$. A set S of $(\mathcal{V}, \Delta)$-substitutions is a *ground complete set of* $(E_\mathcal{R}, \Delta)$*-unifiers of t and s away from W,* if the following three conditions hold:

1. For every $\varphi \in S$: $\mathcal{D}(\varphi) \subseteq V$ and $\mathcal{I}(\varphi) \cap W = \emptyset$.
2. $S \subseteq \mathcal{U}_{(E_\mathcal{R},\Delta)}(t,s)$.
3. For every $\varphi \in g\mathcal{U}_{(E_\mathcal{R},\Delta)}(t,s)$ there is a $\psi \in S$ such that $\psi \preceq_{E_\mathcal{R}} \varphi\ (V)$. $\oplus$

For ctn-trs's, a modification of Theorem 5.5 obtained by replacing the narrowing relation by an arbitrary strategy, is presented in [Ech88] in Theorem 3. A ground complete set of $(E_\mathcal{R}, \Delta)$-unifiers is computed by the universal unification algorithm which is implied by this theorem. We present an instance of this theorem where we choose the strategy of taking the leftmost outermost narrowing occurrence. Also for $\overset{lo}{\rightsquigarrow}_\mathcal{R}$ we assume that it is extended to objects of the form $(equ(t,s), \varphi)$ in an obvious way.

Theorem 6.10 (cf. [Ech88] Theorem 3) Let $\mathcal{R} = (F, \Delta, R)$ be a ctn-trs. Let $t, s \in T\langle F \cup \Delta\rangle(\mathcal{V})$, and let V be the set $\mathcal{V}(t) \cup \mathcal{V}(s)$. Let S be the set of all $(\mathcal{V}, \Delta)$-substitutions φ such that φ is in S iff there exists a derivation by $\overset{lo}{\rightsquigarrow}_\mathcal{R}$:

$$(equ(t,s), \varphi_\emptyset) \overset{lo}{\rightsquigarrow}_\mathcal{R} (equ(t_1,s_1), \varphi_1) \overset{lo}{\rightsquigarrow}_\mathcal{R} (equ(t_2,s_2), \varphi_2) \overset{lo}{\rightsquigarrow}_\mathcal{R} \cdots \overset{lo}{\rightsquigarrow}_\mathcal{R} (equ(t_n,s_n), \varphi_n),$$

where for every $i \in [n] : \varphi_i$ is in normal form, t_n and s_n are in normal form and unifiable with most general unifier μ, and $\varphi = (\varphi_n \circ \mu)|_V$. Then S is a ground complete set of $(E_\mathcal{R}, \Delta)$-unifiers of t and s away from V. $\oplus$

In Figure 6 the leftmost outermost narrowing tree for the term $equ(sh(z_1, \alpha), mi(\sigma(z_2, \alpha)))$ which is associated to a computation of the algorithm implied by Theorem 6.10, is shown, where the shaded areas do not belong to this tree. Remark that this tree is a part of Hullot's narrowing tree in Figure 5. Also note that, as in the tree in Figure 5, the branches are lengthened by the unification of the two subterms at the leaves.

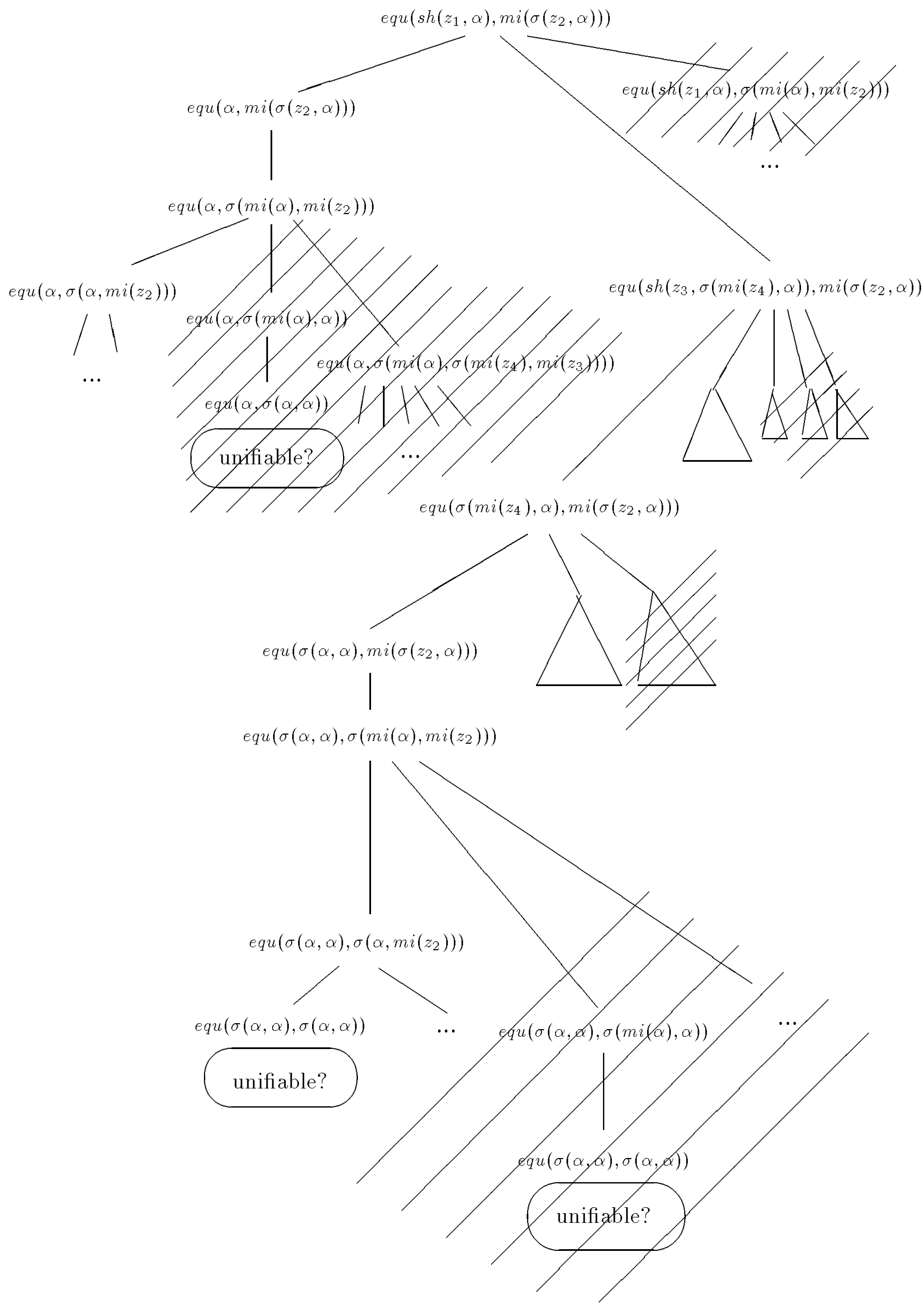

Figure 6: Hullot's leftmost outermost narrowing tree.

7 $E_{\mathcal{R}}$-Unification by Unification-Driven LO-Narrowing

In this section we further increase the efficiency of the universal unification algorithm implied by Theorem 6.10, by splitting the unifications at the front of Hullot's leftmost outermost narrowing trees into steps which correspond to decomposition steps in the unification algorithm of [MM82] and by applying them as early as possible. By means of this strategy, some derivations that do not yield an $(E_{\mathcal{R}}, \Delta)$-unifier, are stopped earlier than in the algorithm implied by Theorem 6.10. For instance, in Hullot's leftmost outermost narrowing tree in Figure 6, the infinite tree at the left the root of which is labeled by $equ(\alpha, \sigma(\alpha, mi(z_2)))$, is cut, because the subterms are not unifiable. For formalizing this strategy, for every ctn-trs $\mathcal{R} = (F, \Delta, R)$, we introduce a term rewriting system which is called the *equal-part of* $\mathcal{R}$. The union of the equal-part of $\mathcal{R}$ and $\mathcal{R}$ itself is called the *equal-extension of* $\mathcal{R}$. Then, roughly speaking, leftmost outermost narrowing is performed on the basis of the equal-extension of $\mathcal{R}$. This is formalized by the unification-driven leftmost outermost narrowing relation.

We start this section with the definition of the equal-part and the equal-extension of a ctn-trs. Then we introduce the unification-driven leftmost outermost narrowing relation and present a theorem which implies a universal unification algorithm for the class of equational theories $=_{E_{\mathcal{R}}}$ where $\mathcal{R}$ is a ctn-trs. For this purpose, we show that a most general unifier of two terms $t, s \in T\langle\Delta\rangle(\mathcal{V})$ can be computed by a derivation by the unification-driven leftmost outermost narrowing relation which is restricted to the equal-part of the ctn-trs.

7.1 The Equal-Part and the Equal-Extension of a CTN-TRS

We start with the definition of the equal-part of a ctn-trs $\mathcal{R}$.

Definition 7.1 Let $\mathcal{R} = (F, \Delta, R)$ be a ctn-trs. The *equal-part of* $\mathcal{R}$, denoted by $\mathcal{R}(\Delta)$, is the triple $(\hat{F}, \Delta, R(\Delta))$ where

- $\hat{F} = F \cup \{equ\}$ where equ is a new binary symbol.
- $R(\Delta)$ contains, for every $\sigma \in \Delta^{(k)}$ with $k \geq 0$, the rule

$$equ(\sigma(x_1, \ldots, x_k), \sigma(x_{k+1}, \ldots, x_{2k})) \rightarrow \sigma(equ(x_1, x_{k+1}), \ldots, equ(x_k, x_{2k})).$$

$\oplus$

A rule in $R(\Delta)$ is called an *equal-rule.* Later we will see that the unification of two terms $t, s \in T\langle\Delta\rangle(\mathcal{V})$ can be realized by some derivation associated with $\mathcal{R}(\Delta)$. But the equal-rules are also part of the term rewriting system for which we define the unification-driven leftmost outermost narrowing relation. The original term rewriting system $\mathcal{R}$ enriched by the equal-part of $\mathcal{R}$ is the *equal-extension of* $\mathcal{R}$.

Definition 7.2 Let $\mathcal{R} = (F, \Delta, R)$ be a ctn-trs and let $\mathcal{R}(\Delta) = (\hat{F}, \Delta, R(\Delta))$ be the equal-part of $\mathcal{R}$. The *equal-extension of* $\mathcal{R}$, denoted by $\hat{\mathcal{R}}$, is the triple $(\hat{F}, \Delta, \hat{R})$ where $\hat{R}$ is the set $R \cup R(\Delta)$. ⊕

The enumeration of the rules in $\hat{R}$ is given by the bijection $\hat{\pi} : \hat{R} \to [card(\hat{R})]$ such that $\hat{\pi}|_R = \pi$ where π is the bijection that induces the enumeration of R, and the equal-rules are enumerated in any arbitrary order (which is irrelevant in the future).

In Figure 7 the rules of the equal-extension $\hat{\mathcal{R}}_1 = (\hat{F}_1, \Delta_1, \hat{R}_1)$ of $\mathcal{R}_1$ (cf. Figure 1) are shown where $\hat{F}_1 = \{sh^{(2)}, mi^{(1)}, equ^{(2)}\}$ and $\Delta_1 = \{\sigma^{(2)}, \alpha^{(0)}\}$.

$$
\begin{array}{rcll}
sh(\alpha, y_1) & \to & y_1 & (1) \\
sh(\sigma(x_1, x_2), y_1) & \to & sh(x_1, \sigma(mi(x_2), y_1)) & (2) \\
mi(\alpha) & \to & \alpha & (3) \\
mi(\sigma(x_1, x_2)) & \to & \sigma(mi(x_2), mi(x_1)) & (4) \\
equ(\alpha, \alpha) & \to & \alpha & (5) \\
equ(\sigma(x_1, x_2), \sigma(x_3, x_4)) & \to & \sigma(equ(x_1, x_3), equ(x_2, x_4)) & (6)
\end{array}
$$

Figure 7: Set of rules of an equal-extension.

In the following subsection we introduce the unification-driven leftmost outermost derivation calculus.

7.2 The Unification-Driven LO-Narrowing Calculus

Roughly speaking, the unification-driven leftmost outermost narrowing relation is almost the same as the leftmost outermost narrowing relation associated with $\hat{\mathcal{R}}$. But there are the following three differences between the two relations. Let (t, φ) be the current derivation form.

1. The term $t = equ(\alpha, \sigma(\alpha, mi(z_2)))$ at the leftmost node in the tree in Figure 6 derives by the leftmost outermost narrowing relation at the leftmost outermost narrowing occurrence 22. But the derivation of the unification-driven leftmost outermost narrowing relation stops at this point, because the two direct subterms α and $\sigma(\alpha, mi(z_2))$ cannot be unified because of different root symbols. Thus, the occurrence Λ of t is important for further narrowing on t, because the nonunifiability of the two subterms of t is recognized exactly at this occurrence. In Definition 7.3 we fix this occurrence and call it the *important occurrence in t*, for short $impO(t)$.

2. If $t/impO(t) = equ(z_i, z_j)$ for two variables z_i and z_j, then, by means of the leftmost outermost narrowing relation, (t, φ) derives to $card(\Delta)$ many terms by unifying $t/impO(t)$ with the left hand sides of the equal-rules. Thus, z_i and z_j are substituted by the same term which has the form $\sigma(z_{k+1}, \ldots, z_{k+n})$ where $\sigma \in \Delta^{(n)}$. As mentioned before, the unification at the end of Hullot's algorithm is realized by equal-rules. Thus, the leftmost outermost narrowing relation yields the substitution

$[z_i/\sigma(z_{k+1},\ldots,z_{k+n}), z_j/\sigma(z_{k+1},\ldots,z_{k+n})]$ as most general unifier of z_i and z_j. But the most general unifier of z_i and z_j is $[z_i/z_j]$ (cf. [MM82]). To be correct with respect to the algorithm in [MM82], a derivation form (t,φ) with $t/impO(t) = equ(z_i, z_j)$ derives by the unification-driven leftmost outermost narrowing relation as follows: $t/impO(t)$ is replaced by z_j, every occurrence of z_i in t is replaced by z_j, and φ is composed with the substitution $[z_i/z_j]$.

3. If $t/impO(t) = equ(z_i, t')$ or $t/impO(t) = equ(t', z_i)$ where $t' \in T\langle F \cup \Delta\rangle(\mathcal{V})\backslash\mathcal{V}$, then we have to check whether z_i occurs in the $(\Delta \cup \mathcal{V})$-prefix of t' or not. This check is called the *occur check* in unification algorithms. In Theorem 5.5 it is done implicitly during the unification of t_n and s_n at the end of the derivation. But, since in our algorithm the unification is realized by equal-rules, we have to apply the occur check explicitly. The $(\Delta \cup \mathcal{V})$-prefix of the term s consists of all occurrences u in s such that there is no occurrence v which is a prefix of u and which is labeled by a function symbol. In Figure 8, the $(\Delta \cup \mathcal{V})$-prefix of the tree $\sigma(\sigma(\sigma(z_1,\alpha), z_2), \sigma(sh(\alpha, z_1), \alpha))$ is inside the frame.

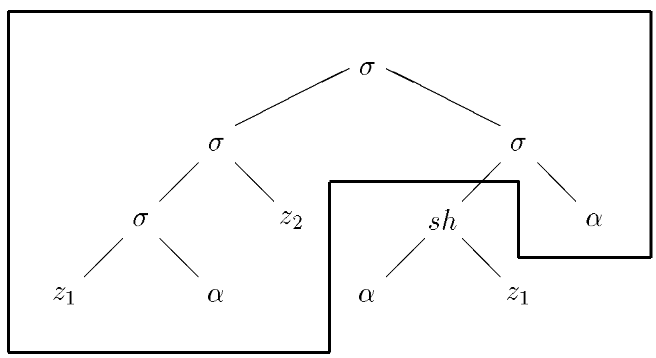

Figure 8: $(\Delta \cup \mathcal{V})$-Prefix.

Before we introduce the unification-driven leftmost outermost narrowing relation, we define some notions which are used in Definition 7.4.

Definition 7.3 Let $\mathcal{R} = (F, \Delta, R)$ be a ctn-trs and let $t \in T\langle \hat{F} \cup \Delta\rangle(\mathcal{V})$.

- The set of *equal occurrences in t,* denoted by $equO(t)$, is the set $\{u \in O(t) \mid t[u] = equ\}$.
- The *important occurrence in t,* denoted by $impO(t)$, is the occurrence $min_{lex} equO(t)$.
- t is in *binding form,* if $t[impO(t)1], t[impO(t)2] \in \mathcal{V}$.
- The $(\Delta \cup \mathcal{V})$-*prefix of* t is the set

$$\{u \in O(t) \mid \text{there does not exist any } v \in O(t), v \leq u \text{ and } t[v] \in F\}.$$

- The *occur check for t succeeds,* if the following conditions hold:

1. t is not in binding form.
2. $t[impO(t)i] \in \mathcal{V}$ for exactly one $i \in [2]$.
3. There exists a u in the $(\Delta \cup \mathcal{V})$-prefix of $t/(impO(t)(3-i))$ such that $t/(impO(t)(3-i))[u] = t[impO(t)i]$ $\oplus$

Now we are able to present the definition of the unification-driven leftmost outermost narrowing relation.

Definition 7.4 Let $\mathcal{R} = (F, \Delta, R)$ be a ctn-trs. The *unification-driven leftmost outermost narrowing relation associated with* $\hat{\mathcal{R}}$, denoted by $\overset{u}{\leadsto}_{\hat{\mathcal{R}}}$, is defined as follows: For every $t, s \in T\langle \hat{F} \cup \Delta\rangle(\mathcal{V})$ and $\psi, \psi' \in Sub(\mathcal{V}, \Delta)$: $(t, \psi) \overset{u}{\leadsto}_{\hat{\mathcal{R}}} (s, \psi')$, if $t/impO(t) = equ(t_1, t_2)$ where $t_1, t_2 \in T\langle F \cup \Delta\rangle(\mathcal{V})$ and one of the following conditions holds:

1. $(t_1[\Lambda], t_2[\Lambda] \in \Delta$ and $t_1[\Lambda] = t_2[\Lambda])$ or $(((t_1[\Lambda] \in \Delta$ and $t_2[\Lambda] \in \mathcal{V})$ or $(t_1[\Lambda] \in \mathcal{V}$ and $t_2[\Lambda] \in \Delta))$ and the occur check fails for $t)$ and the following three conditions hold:
   (a) $(equ(t_1, t_2), \varphi_\emptyset) \overset{lo}{\leadsto}_{\mathcal{R}(\Delta)} (t', \varphi')$.
   (b) $s = \varphi'(t[impO(t) \leftarrow t'])$.
   (c) $\psi' = \psi \circ \varphi'$.
2. $t_1[\Lambda], t_2[\Lambda] \in \mathcal{V}$ and the following three conditions hold:
   (a) $\varphi' = [t_1/t_2]$.
   (b) $s = \varphi'(t[impO(t) \leftarrow t_2])$.
   (c) $\psi' = \psi \circ \varphi'$.
3. $t_1[\Lambda] \in F$ and the following three conditions hold:
   (a) $(t_1, \varphi_\emptyset) \overset{lo}{\leadsto}_{\mathcal{R}} (t', \varphi')$.
   (b) $s = \varphi'(t[impO(t)1 \leftarrow t'])$.
   (c) $\psi' = \psi \circ \varphi'$.
4. $t_1[\Lambda] \notin F$ and $t_2[\Lambda] \in F$ and the following three conditions hold:
   (a) $(t_2, \varphi_\emptyset) \overset{lo}{\leadsto}_{\mathcal{R}} (t', \varphi')$.
   (b) $s = \varphi'(t[impO(t)2 \leftarrow t'])$.
   (c) $\psi' = \psi \circ \varphi'$. $\oplus$

In the cases 1, 3, and 4, i.e., in the case of an application of a rule $l \to r \in \hat{R}$, we write $\overset{u}{\leadsto}_{\hat{\mathcal{R}}, l\to r}$. In case 2 we write $\overset{u}{\leadsto}_{\hat{\mathcal{R}}, bm}$ to indicate that the current term is in binding mode.

In the following example a nonsuccessful derivation and a successful derivation by the unification-driven leftmost outermost narrowing relation for the equal-extension $\hat{\mathcal{R}}_1$ are shown (cf. Figure 7 for the set of rules).

Example 7.5 (a) $E_{\mathcal{R}_1}$-unification of the terms $sh(z_1, \sigma(\alpha, z_2))$ and $\sigma(mi(z_1), \sigma(z_2, \alpha))$.

$$
\begin{array}{ll}
 & (equ(sh(z_1, \sigma(\alpha, z_2)), \sigma(mi(z_1), \sigma(z_2, \alpha))), \varphi_\emptyset) \\
\overset{u}{\leadsto}_{\hat{\mathcal{R}}_1,1,(1)} & (equ(\sigma(\alpha, z_2), \sigma(mi(\alpha), \sigma(z_2, \alpha))), [z_1/\alpha]) \\
\overset{u}{\leadsto}_{\hat{\mathcal{R}}_1,\Lambda,(6)} & (\sigma(equ(\alpha, mi(\alpha)), equ(z_2, \sigma(z_2, \alpha))), [z_1/\alpha]) \\
\overset{u}{\leadsto}_{\hat{\mathcal{R}}_1,12,(3)} & (\sigma(equ(\alpha, \alpha), equ(z_2, \sigma(z_2, \alpha))), [z_1/\alpha]) \\
\overset{u}{\leadsto}_{\hat{\mathcal{R}}_1,1,(5)} & (\sigma(\alpha, \underline{equ(z_2, \sigma(z_2, \alpha))}), [z_1/\alpha])
\end{array}
$$

Here the derivation stops, because the occur check succeeds.

(b) $E_{\mathcal{R}_1}$-unification of the terms $sh(z_1, \sigma(\alpha, z_2))$ and $\sigma(mi(z_1), \sigma(z_3, \alpha))$. Derivation steps (1)-(4) are analog to those one in (a).

$$
\begin{array}{ll}
 & (equ(sh(z_1, \sigma(\alpha, z_2)), \sigma(mi(z_1), \sigma(z_3, \alpha))), \varphi_\emptyset) \\
\overset{u}{\leadsto}{}^{4}_{\hat{\mathcal{R}}_1} & (\sigma(\alpha, \underline{equ(z_2, \sigma(z_3, \alpha))}), [z_1/\alpha]) \\
\overset{u}{\leadsto}_{\hat{\mathcal{R}}_1,2,(6)} & (\sigma(\alpha, \sigma(equ(z_4, z_3), equ(z_5, \alpha))), [z_1/\alpha, z_2/\sigma(z_4, z_5)]) \\
\overset{u}{\leadsto}_{\hat{\mathcal{R}}_1,21,bm} & (\sigma(\alpha, \sigma(z_3, equ(z_5, \alpha))), [z_1/\alpha, z_2/\sigma(z_3, z_5)]) \\
\overset{u}{\leadsto}_{\hat{\mathcal{R}}_1,22,(5)} & (\sigma(\alpha, \sigma(z_3, \alpha)), [z_1/\alpha, z_2/\sigma(z_3, \alpha)])
\end{array}
$$

Here the derivation yields the $E_{\mathcal{R}_1}$-unifier $[z_1/\alpha, z_2/\sigma(z_3, \alpha)]$. $\oplus$

Similar to the reduction relation, the narrowing relation, and the leftmost outermost narrowing relation, we also define an indexed derivation calculus for the unification-driven leftmost outermost narrowing relation. This indexed derivation calculus is called the *unification-driven leftmost outermost narrowing calculus.* We choose the set $\hat{R} \cup \{bm\}$ as index set where the symbol bm indicates a derivation step of the form $\overset{u}{\leadsto}_{\hat{\mathcal{R}},bm}$. The total order is implied by the enumeration $\hat{\pi}$ of the rules in $\hat{R}$ and we define bm as the minimal element of the index set.

Definition 7.6 Let $\mathcal{R} = (F, \Delta, R)$ be a ctn-trs with enumeration π on R.

- The *unification-driven leftmost outermost narrowing calculus of $\mathcal{R}$,* denoted by $\mathcal{D}_{u-nar\mathcal{R}}$, is the indexed derivation calculus $(T\langle \hat{F} \cup \Delta\rangle(\mathcal{V}) \times Sub(\mathcal{V}, \Delta), (\hat{R} \cup \{bm\}, <), (\overset{u}{\leadsto}_{\hat{\mathcal{R}},\psi} \mid \psi \in \hat{R} \cup \{bm\}))$ for $(T\langle \hat{F} \cup \Delta\rangle(\mathcal{V}) \times Sub(\mathcal{V}, \Delta), \overset{u}{\leadsto}_{\hat{\mathcal{R}}})$ where $<$ is the total order on $\hat{R} \cup \{bm\}$ defined as follows: for every $\psi, \psi' \in \hat{R} \cup \{bm\}$, $\psi < \psi$, if one of the following conditions hold:
  1. $\psi = bm$ and $\psi' \neq bm$
  2. $\psi \neq bm$, $\psi' \neq bm$, and $\hat{\pi}(\psi) < \hat{\pi}(\psi')$.
- A *unification-driven leftmost outermost narrowing tree of* $\mathcal{R}$ is a derivation tree of $\mathcal{D}_{u-nar\mathcal{R}}$. $\oplus$

The unification-driven leftmost outermost narrowing tree of $\mathcal{R}_1$ for the $E_{\mathcal{R}_1}$-unification of the terms $sh(z_1, \alpha)$ and $mi(\sigma(z_2, \alpha))$ is shown in Figure 3.

7.3 Unification by $\overset{u}{\leadsto}_{\mathcal{R}(\Delta)}$

As an intermediate result between Theorem 6.10 and the intended universal unification algorithm in Theorem 7.9 which is based on the unification driven leftmost outermost narrowing relation, we show in this subsection that the unification of two terms $t, s \in T\langle\Delta\rangle(\mathcal{V})$ can be realized by a derivation by the unification-driven leftmost outermost narrowing relation associated with $\mathcal{R}(\Delta)$.

Lemma 7.7 Let $\mathcal{R} = (F, \Delta, R)$ be a ctn-trs and let $t, s \in T\langle\Delta\rangle(\mathcal{V})$. t and s are unifiable with most general unifier φ iff there exists a derivation by $\overset{u}{\leadsto}_{\mathcal{R}(\Delta)}$ of the following form $(equ(t,s), \varphi_\emptyset) \overset{u}{\leadsto}^*_{\mathcal{R}(\Delta)} (t', \varphi)$ and $t' \in T\langle\Delta\rangle(\mathcal{V})$.

<u>*Proof:*</u> We show that every transformation of the unification algorithm in [MM82] can be realized by a derivation by $\overset{u}{\leadsto}_{\mathcal{R}(\Delta)}$, and vice versa, every derivation by $\overset{u}{\leadsto}_{\mathcal{R}(\Delta)}$ can be realized by a finite number of transformations of the unification algorithm in [MM82]. For this purpose, we first recall the unification algorithm from [MM82]. In this algorithm the unification of t and s starts with a set P with one unordered pair $\langle t, s\rangle$. Then, a finite number of transformations are applied step by step to this set. The transformations are of the following three forms:

1. If there is an unordered pair $\langle z_i, z_i\rangle$ in P, then P is transformed to the set $P\backslash\{\langle z_i, z_i\rangle\}$.
2. If there is an unordered pair $\langle\sigma(t_1,\ldots,t_k), \sigma(s_1,\ldots,s_k)\rangle$ in P, then P is transformed to the set $P\backslash\{\langle\sigma(t_1,\ldots,t_k), \sigma(s_1,\ldots,s_k)\rangle\} \cup \{\langle t_1, s_1\rangle, \ldots, \langle t_k, s_k\rangle\}$.
3. If there is an unordered pair $\langle z_i, s\rangle \in P$ such that z_i does not occur in s, then P is transformed to $\varphi(P\backslash\{\langle z_i, s\rangle\}) \cup \{\langle z_i, s\rangle\}$, where $\varphi = [z_i/s]$ and the φ-image of a set is defined as the set of the φ-images of its elements.

The algorithm stops, if P is in solved form, i.e., $P = \{\langle z_i, t_i\rangle \mid i \in [n]\}$ where for every $i, j \in [n] : z_i \neq z_j$ for $i \neq j$ and z_i does not occur in any t_j. Then, $[z_1/t_1, \ldots, z_n/t_n]$ is the most general unifier of t and s.

To decrease the number of used notations, we only explain the correspondence of every transformation of type 1 - 3 with a derivation by $\overset{u}{\leadsto}_{\mathcal{R}(\Delta)}$. Let $(t, \varphi) \in T\langle\hat{F} \cup \Delta\rangle(\mathcal{V}) \times Sub(\mathcal{V}, \Delta)$.

1. A transformation of type 1 corresponds to the derivation $(t, \varphi) \overset{u}{\leadsto}_{\mathcal{R}(\Delta)} (t[impO(t) \leftarrow z_i], \varphi)$, because $t/impO(t) = equ(z_i, z_i)$. Then, the substitution φ is not changed.
2. A transformation of type 2 corresponds to the derivation $(t, \varphi) \overset{u}{\leadsto}_{\mathcal{R}(\Delta)} (t', \varphi)$, where $t' = t[impO(t) \leftarrow \sigma(equ(t_1, s_1), \ldots, equ(t_k, s_k))]$ and φ is not changed, because $t/impO(t) = equ(\sigma(t_1,\ldots,t_k), \sigma(s_1,\ldots,s_k))$. Thus, an application of an equal-rule covers the transformation of type 2.
3. A transformation of type 3 corresponds to the derivation $(t, \varphi) \overset{u}{\leadsto}^*_{\mathcal{R}(\Delta)} (t', \varphi \circ [z_i/s])$, where t' is the term that results from t by replacing every occurrence of z_i by s. The length of this derivation is $size(s)$, because the equal-rules are applied node by node in s. $\oplus$

The unification of the terms $t = \sigma(z_1, z_2)$ and $s = \sigma(\sigma(z_2, \alpha), \alpha)$ via a derivation by $\overset{u}{\leadsto}_{\mathcal{R}_1(\Delta_1)}$ is shown in Figure 9 (for $\mathcal{R}_1$ and Δ_1 cf. Figure 1). The most general unifier is $\theta = [z_1/\sigma(\alpha, \alpha), z_2/\alpha]$.

$$
\begin{array}{ll}
 & (equ(\sigma(z_1, z_2), \sigma(\sigma(z_2, \alpha), \alpha)), \varphi_\emptyset) \\
\overset{u}{\leadsto}_{\mathcal{R}_1(\Delta_1),(6)} & (\sigma(equ(z_1, \sigma(z_2, \alpha)), equ(z_2, \alpha)), \varphi_\emptyset) \\
\overset{u}{\leadsto}_{\mathcal{R}_1(\Delta_1),(6)} & (\sigma(\sigma(equ(z_3, z_2), equ(z_4, \alpha)), equ(z_2, \alpha)), [z_1/\sigma(z_3, z_4)]) \\
\overset{u}{\leadsto}_{\mathcal{R}_1(\Delta_1),bm} & (\sigma(\sigma(z_2, equ(z_4, \alpha)), equ(z_2, \alpha)), [z_1/\sigma(z_2, z_4)]) \\
\overset{u}{\leadsto}_{\mathcal{R}_1(\Delta_1),(5)} & (\sigma(\sigma(z_2, \alpha), equ(z_2, \alpha)), [z_1/\sigma(z_2, \alpha)]) \\
\overset{u}{\leadsto}_{\mathcal{R}_1(\Delta_1),(5)} & (\sigma(\sigma(\alpha, \alpha), \alpha), [z_1/\sigma(\alpha, \alpha), z_2/\alpha])
\end{array}
$$

Figure 9: A unification by a derivation by $\overset{u}{\leadsto}_{\mathcal{R}_1(\Delta_1)}$.

Now we present a theorem which is a simple modification of Theorem 6.10 obtained by replacing the unification by a derivation by $\overset{u}{\leadsto}_{\mathcal{R}(\Delta)}$.

Theorem 7.8 Let $\mathcal{R} = (F, \Delta, R)$ be a ctn-trs. Let $t, s \in T\langle F \cup \Delta\rangle(\mathcal{V})$, and let V be the set $\mathcal{V}(t) \cup \mathcal{V}(s)$. Let S be the set of all $(\mathcal{V}, \Delta)$-substitutions φ such that φ is in S iff there exists a derivation by $\overset{lo}{\leadsto}_{\mathcal{R}}$:

$$(equ(t, s), \varphi_\emptyset) \overset{lo}{\leadsto}_{\mathcal{R}} (equ(t_1, s_1), \varphi_1) \overset{lo}{\leadsto}_{\mathcal{R}} (equ(t_2, s_2), \varphi_2) \overset{lo}{\leadsto}_{\mathcal{R}} \cdots \overset{lo}{\leadsto}_{\mathcal{R}} (equ(t_n, s_n), \varphi_n),$$

where for every $i \in [n]$: φ_i is in normal form, t_n and s_n are in normal form, and there exists a derivation by $\overset{u}{\leadsto}_{\mathcal{R}(\Delta)}$:

$$(equ(t_n, s_n), \varphi_n) \overset{u}{\leadsto}{}^{*}_{\mathcal{R}(\Delta)} (t', \varphi'),$$

and $\varphi = \varphi'|_V$. Then S is a ground complete set of $(E_{\mathcal{R}}, \Delta)$-unifiers of t and s away from V.

<u>*Proof:*</u> The correctness of Theorem 7.8 immediately follows from Theorem 6.10 and from Lemma 7.7. $\oplus$

7.4 $E_{\mathcal{R}}$-Unification by $\overset{u}{\leadsto}_{\hat{\mathcal{R}}}$

We finish this section by showing that we can compute a ground complete set of $(E_{\mathcal{R}}, \Delta)$-unifiers of two terms t and s by derivations by the unification-driven leftmost outermost narrowing relation.

Theorem 7.9 Let $\mathcal{R} = (F, \Delta, R)$ be a ctn-trs. Let $t, s \in T\langle F \cup \Delta\rangle(\mathcal{V})$, and let V be the set $\mathcal{V}(t) \cup \mathcal{V}(s)$. Let S be the set of all $(\mathcal{V}, \Delta)$-substitutions φ such that φ is in S iff there exists a derivation by $\overset{u}{\leadsto}_{\hat{\mathcal{R}}}$:

$$(equ(t, s), \varphi_\emptyset) \overset{u}{\leadsto}_{\hat{\mathcal{R}}} (t_1, \varphi_1) \overset{u}{\leadsto}_{\hat{\mathcal{R}}} (t_2, \varphi_2) \overset{u}{\leadsto}_{\hat{\mathcal{R}}} \cdots \overset{u}{\leadsto}_{\hat{\mathcal{R}}} (t_n, \varphi_n),$$

where for every $i \in [n] : \varphi_i$ is in normal form, $t_n \in T\langle\Delta\rangle(\mathcal{V})$, and $\varphi = \varphi_n|_V$. Then S is a ground complete set of $(E_{\mathcal{R}}, \Delta)$-unifiers of t and s away from V.

<u>*Proof:*</u> We show that there exists a derivation

$$(equ(t,s), \varphi_\emptyset) \overset{lo}{\rightsquigarrow}{}^{*}_{\mathcal{R}} (equ(t',s'), \varphi') \overset{u}{\rightsquigarrow}{}^{*}_{\mathcal{R}(\Delta)} (t^*, \varphi^*), \tag{1}$$

where $t', s', t^* \in T\langle\Delta\rangle(\mathcal{V})$ and $\varphi', \varphi^* \in Sub(\mathcal{V}, \Delta)$ iff there exists a derivation

$$(equ(t,s), \varphi_\emptyset) \overset{u}{\rightsquigarrow}{}^{*}_{\hat{\mathcal{R}}} (t^*, \varphi^*) \tag{2}$$

Furthermore, we show that the lengths of derivation 1 and derivation 2 are equal. Then from Theorem 7.8 the correctness of the Theorem follows.

Derivation 1 $\Longrightarrow$ Derivation 2

First, we show that for every derivation 1, there exists a derivation 2. For this purpose, we introduce the function $eqpos : T\langle F \cup \Delta\rangle(\mathcal{V}) \times T\langle F \cup \Delta\rangle(\mathcal{V}) \to \mathbb{N}$ that yields, for two terms t_1 and t_2, the number of occurrences $u \in O(t_1) \cap O(t_2)$ where an equal-rule can be applied or where the subterms t_1/u and t_2/u are in binding form, and such that u is less with respect to $<_{lex}$ than the leftmost outermost occurrence in $O(t_1) \cap O(t_2)$ at which no equal-rule is applicable. The latter occurrence is denoted by $lonotuniocc(t_1, t_2)$ and it is defined as follows:

$$\begin{array}{ll} min_{lex}\{u \in O(t_1) \cap O(t_2) \mid & t_1[u] \in F \text{ or } t_2[u] \in F \text{ or} \\ & (t_1[u] \in \Delta \text{ and } t_2[u] \in \Delta \text{ and } t_1[u] \neq t_2[u]) \text{ or} \\ & \text{the occur check for } equ(t_1[u], t_2[u]) \text{ succeeds}\} \end{array}$$

Then $eqpos(t_1, t_2)$ is defined as follows:

$$\sum_{\{u \in O(t_1) \cap O(t_2) \;\mid\; u <_{lex} lonotuniocc(t_1,t_2)\}} equsteps(t_1, t_2, u)$$

$equsteps(t_1, t_2, u)$ is the number of equal-rule applications at occurrence u. It is defined as follows

$$equsteps(t_1, t_2, u) = \begin{cases} 1 & ; \text{ if } t_1[u], t_2[u] \in \Delta \text{ and } t_1[u] = t_2[u] \\ n & ; \text{ if } i \in [2] : t_i \in \mathcal{V}, t_{3-i} \in T\langle F \cup \Delta\rangle(\mathcal{V}) \text{ and} \\ & \quad n = card(\{w \in O(t_{3-i}) \mid w <_{lex} min_{lex}\{v \mid t_{3-i}[v] \in F\}\}) \end{cases}$$

Furthermore, we prove the following Claim by induction on k.

Claim 1 For every $k \geq 0$, $\zeta_t, \zeta_s \in T\langle F \cup \Delta\rangle(\mathcal{V})$, $\zeta \in T\langle \hat{F} \cup \Delta\rangle(\mathcal{V})$, and for every $\varphi, \psi \in Sub(\mathcal{V}, \Delta)$: If there exists a derivation

$$(equ(t,s), \varphi_\emptyset) \overset{lo}{\rightsquigarrow}{}^{k}_{\mathcal{R}} (equ(\zeta_t, \zeta_s), \varphi) \overset{u}{\rightsquigarrow}{}^{eqpos(\zeta_t,\zeta_s)}_{\mathcal{R}(\Delta)} (\zeta, \psi),$$

then there exists a derivation

$$(equ(t,s), \varphi_\emptyset) \overset{u}{\rightsquigarrow}{}^{k+eqpos(\zeta_t,\zeta_s)}_{\hat{\mathcal{R}}} (\zeta, \psi).$$

Induction on k:

$\underline{k = 0:}$ $\zeta_t = t$ and $\zeta_s = s$. We have $(equ(t,s), \varphi_\emptyset) \overset{u}{\leadsto}_{\mathcal{R}(\Delta)}^{eqpos(\zeta_t,\zeta_s)} (\zeta, \psi)$.

From $R(\Delta) \subseteq \hat{R}$ follows $(equ(t,s), \varphi_\emptyset) \overset{u}{\leadsto}_{\hat{\mathcal{R}}}^{eqpos(\zeta_t,\zeta_s)} (\zeta, \psi)$.

$\underline{k \to k+1:}$ There exist $\zeta_t', \zeta_s' \in T\langle F \cup \Delta\rangle(\mathcal{V})$, $\zeta' \in T\langle \hat{F} \cup \Delta\rangle(\mathcal{V})$, $\varphi', \psi' \in Sub(\mathcal{V}, \Delta)$, and there exists the following derivation:

$$(equ(t,s), \varphi_\emptyset) \overset{lo}{\leadsto}_{\mathcal{R}}^{k} (equ(\zeta_t, \zeta_s), \varphi) \overset{lo}{\leadsto}_{\mathcal{R}} (equ(\zeta_t', \zeta_s'), \varphi') \overset{u}{\leadsto}_{\mathcal{R}(\Delta)}^{eqpos(\zeta_t',\zeta_s')} (\zeta', \psi').$$

Now we split the derivation by $\overset{u}{\leadsto}_{\mathcal{R}(\Delta)}$ into two derivations: There exist $\bar{\zeta} \in T\langle \hat{F} \cup \Delta\rangle(\mathcal{V})$, $\bar{\varphi} \in Sub(\mathcal{V}, \Delta)$, and there exists the following derivation:

$$(equ(t,s), \varphi_\emptyset) \overset{lo}{\leadsto}_{\mathcal{R}}^{k} (equ(\zeta_t, \zeta_s), \varphi) \overset{lo}{\leadsto}_{\mathcal{R}} (equ(\zeta_t', \zeta_s'), \varphi') \overset{u}{\leadsto}_{\mathcal{R}(\Delta)}^{eqpos(\zeta_t,\zeta_s)} (\bar{\zeta}, \bar{\varphi})$$

$$\overset{u}{\leadsto}_{\mathcal{R}(\Delta)}^{eqpos(\zeta_t',\zeta_s') - eqpos(\zeta_t,\zeta_s)} (\zeta', \psi').$$

There exist $\bar{\zeta}' \in T\langle \hat{F} \cup \Delta\rangle(\mathcal{V})$, $\bar{\varphi}' \in Sub(\mathcal{V}, \Delta)$, and there exists the following derivation by changing the order of applications of rules in the previous derivation:

$$(equ(t,s), \varphi_\emptyset) \overset{lo}{\leadsto}_{\mathcal{R}}^{k} (equ(\zeta_t, \zeta_s), \varphi) \overset{u}{\leadsto}_{\mathcal{R}(\Delta)}^{eqpos(\zeta_t,\zeta_s)} (\bar{\zeta}', \bar{\varphi}') \overset{u}{\leadsto}_{\hat{\mathcal{R}}} (\bar{\zeta}, \bar{\varphi})$$

$$\overset{u}{\leadsto}_{\mathcal{R}(\Delta)}^{eqpos(\zeta_t',\zeta_s') - eqpos(\zeta_t,\zeta_s)} (\zeta', \psi').$$

Changing the order of the derivation is correct, because in the derivation step $(equ(\zeta_t, \zeta_s), \varphi)$ $\overset{lo}{\leadsto}_{\mathcal{R}} (equ(\zeta_t', \zeta_s'), \varphi')$, a function is applied at the leftmost outermost narrowing occurrence. From the definition of *eqpos* it follows that this occurrence is the leftmost outermost narrowing occurrence in $\bar{\zeta}'$, too. Furthermore, in the case of a function application, the relations $\overset{lo}{\leadsto}_{\mathcal{R}}$ and $\overset{u}{\leadsto}_{\hat{\mathcal{R}}}$ yield the same result.

The existence of the following derivation follows from the induction hypothesis:

$$(equ(t,s), \varphi_\emptyset) \overset{u}{\leadsto}_{\hat{\mathcal{R}}}^{k + eqpos(\zeta_t,\zeta_s)} (\bar{\zeta}', \bar{\varphi}') \overset{u}{\leadsto}_{\hat{\mathcal{R}}} (\bar{\zeta}, \bar{\varphi}) \overset{u}{\leadsto}_{\mathcal{R}(\Delta)}^{eqpos(\zeta_t',\zeta_s') - eqpos(\zeta_t,\zeta_s)} (\zeta', \psi').$$

The existence of the following derivation follows from $R(\Delta) \subseteq \hat{R}$:

$$(equ(t,s), \varphi_\emptyset) \overset{u}{\leadsto}_{\hat{\mathcal{R}}}^{k+1+eqpos(\zeta_t',\zeta_s')} (\zeta', \psi').$$

This finishes the proof of Claim 1.

Especially, if k is equal to the length of the derivation by $\overset{lo}{\leadsto}_{\mathcal{R}}$ in derivation 1, it follows that for every derivation 1, there exists a derivation 2.

Derivation 2 $\Longrightarrow$ Derivation 1

Now we show that for every derivation 2, there exists a derivation 1. For this purpose, we introduce the function $eqapp : T\langle \hat{F} \cup \Delta \rangle(\mathcal{V}) \to \mathbb{N}$ that yields, for a term t, the sum of applications of equal-rules and steps started by a term in binding form, in the derivation by $\overset{u}{\leadsto}_{\hat{\mathcal{R}}}$ up to t.

$$eqapp(t) = card(\{u \in O(t) \mid u <_{lex} impO(t)\})$$

Furthermore, we prove the following claim by induction on k.

Claim 2 For every $k \geq 0$, $\zeta \in T\langle \hat{F} \cup \Delta \rangle(\mathcal{V})$, and $\psi \in Sub(\mathcal{V}, \Delta)$: If there exists a derivation

$$(equ(t,s), \varphi_\emptyset) \overset{u}{\leadsto}{}^{k}_{\hat{\mathcal{R}}} (\zeta, \psi),$$

then there exist $\zeta_t, \zeta_s \in T\langle F \cup \Delta \rangle(\mathcal{V})$, $\varphi \in Sub(\mathcal{V}, \Delta)$, and there exists a derivation

$$(equ(t,s), \varphi_\emptyset) \overset{lo}{\leadsto}{}^{k-eqapp(\zeta)}_{\mathcal{R}} (equ(\zeta_t, \zeta_s), \varphi) \overset{u}{\leadsto}{}^{eqapp(\zeta)}_{\mathcal{R}(\Delta)} (\zeta, \psi).$$

<u>*Induction on k:*</u>

<u>$k = 0$:</u> $\zeta = equ(t,s)$, $\psi = \varphi_\emptyset$. Thus, $eqapp(\zeta) = 0$. We have

$$(equ(t,s), \varphi_\emptyset) \overset{lo}{\leadsto}{}^{0-0}_{\mathcal{R}} (equ(\zeta_t, \zeta_s), \varphi_\emptyset) \overset{u}{\leadsto}{}^{0}_{\mathcal{R}(\Delta)} (\zeta, \psi).$$

<u>$k \to k+1$:</u> There exist $\zeta' \in T\langle \hat{F} \cup \Delta \rangle(\mathcal{V})$, $\psi' \in Sub(\mathcal{V}, \Delta)$, and there exists the following derivation:

$$(equ(t,s), \varphi_\emptyset) \overset{u}{\leadsto}{}^{k}_{\hat{\mathcal{R}}} (\zeta, \psi) \overset{u}{\leadsto}_{\hat{\mathcal{R}}} (\zeta', \psi').$$

From the induction hypothesis it follows that there exists the following derivation:

$$(equ(t,s), \varphi_\emptyset) \overset{lo}{\leadsto}{}^{k-eqapp(\zeta)}_{\mathcal{R}} (equ(\zeta_t, \zeta_s), \varphi) \overset{u}{\leadsto}{}^{eqapp(\zeta)}_{\mathcal{R}(\Delta)} (\zeta, \psi) \overset{u}{\leadsto}_{\hat{\mathcal{R}}} (\zeta', \psi').$$

Now we have to distinguish the following two cases:

<u>*Case 1* :</u> $eqapp(\zeta') = eqapp(\zeta)$. Then, the $k+1$th derivation step is a function application. The same function application can be applied to the term $equ(\zeta_t, \zeta_s)$ in a derivation step by $\overset{lo}{\leadsto}_{\mathcal{R}}$. Furthermore, the $eqapp(\zeta)$ derivation steps by $\overset{u}{\leadsto}_{\mathcal{R}(\Delta)}$ work only on occurrences that are less with respect to $<_{lex}$ than the occurrence where the function is applied. Thus, there exist $\zeta'_t, \zeta'_s \in T\langle F \cup \Delta \rangle(\mathcal{V})$, $\varphi' \in Sub(\mathcal{V}, \Delta)$, and there exists a derivation:

$$(equ(t,s), \varphi_\emptyset) \overset{lo}{\leadsto}{}^{k-eqapp(\zeta)}_{\mathcal{R}} (equ(\zeta_t, \zeta_s), \varphi) \overset{lo}{\leadsto}_{\mathcal{R}} (equ(\zeta'_t, \zeta'_s), \varphi') \overset{u}{\leadsto}{}^{eqapp(\zeta')}_{\mathcal{R}(\Delta)} (\zeta', \psi').$$

Then we obtain the following derivation:

$$(equ(t,s), \varphi_\emptyset) \overset{lo}{\leadsto}{}^{k+1-eqapp(\zeta')}_{\mathcal{R}} (equ(\zeta'_t, \zeta'_s), \varphi') \overset{u}{\leadsto}{}^{eqapp(\zeta')}_{\mathcal{R}(\Delta)} (\zeta', \psi').$$

<u>*Case 2* :</u> $eqapp(\zeta') = eqapp(\zeta) + 1$. One of the cases 1 and 2 in Definition 7.4 is applied in the added step. In these cases $\overset{u}{\leadsto}_{\hat{\mathcal{R}}}$ exactly works as $\overset{u}{\leadsto}_{\mathcal{R}(\Delta)}$. We get the following derivation:

$$(equ(t,s), \varphi_\emptyset) \overset{lo}{\leadsto}{}^{k-eqapp(\zeta)}_{\mathcal{R}} (equ(\zeta_t, \zeta_s), \varphi) \overset{u}{\leadsto}{}^{eqapp(\zeta)}_{\mathcal{R}(\Delta)} (\zeta, \psi) \overset{u}{\leadsto}_{\mathcal{R}(\Delta)} (\zeta', \psi').$$

From $eqapp(\zeta') = eqapp(\zeta) + 1$ follows:

$$(equ(t,s), \varphi_\emptyset) \overset{lo}{\rightsquigarrow}_{\mathcal{R}}^{k-(eqapp(\zeta')-1)} (equ(\zeta_t, \zeta_s), \varphi) \overset{u}{\rightsquigarrow}_{\mathcal{R}(\Delta)}^{eqapp(\zeta')} (\zeta', \psi').$$

Here we obtain the following derivation:

$$(equ(t,s), \varphi_\emptyset) \overset{lo}{\rightsquigarrow}_{\mathcal{R}}^{k+1-eqapp(\zeta')} (equ(\zeta_t, \zeta_s), \varphi) \overset{u}{\rightsquigarrow}_{\mathcal{R}(\Delta)}^{eqapp(\zeta')} (\zeta', \psi').$$

Especially, if k is equal to the length of the derivation by $\overset{u}{\rightsquigarrow}_{\hat{\mathcal{R}}}$ in derivation 2, it follows that for every derivation 2, there exists a derivation 1. This finishes the proof of Claim 2. $\oplus$

8 Deterministic Unification-Driven LO-Narrowing

In this section we formalize a deterministic universal unification algorithm for the class of equational theories $=_{E_\mathcal{R}}$ where $\mathcal{R} = (F, \Delta, R)$ is a ctn-trs. This algorithm is the formalization of a depth-first left-to-right traversal through the unification-driven leftmost outermost narrowing tree of $\mathcal{R}$. We choose this strategy, because the strategy that formalizes a breadth-first left-to-right traversal is to inefficient. Whereas the latter strategy yields a ground complete set of $(E_\mathcal{R}, \Delta)$-unifiers, it is clear that the depth-first left-to-right strategy is not complete: If there is an infinite branch left to a branch that yields an $(E_\mathcal{R}, \Delta)$-unifier φ, then φ is not computed by our strategy. The same problem arises in deterministic algorithms for SLD-resolution of PROLOG-programs in [Llo87]. Here we define the deterministic algorithm to compute at most one $(E_\mathcal{R}, \Delta)$-unifier. At the end of this section we show that the deterministic universal unification algorithm is correct.

We do not introduce the definition of a derivation calculus for the deterministic unification-driven leftmost outermost narrowing relation. This would make no sense, because this relation is deterministic. We only introduce the deterministic unification-driven leftmost outermost narrowing relation and its derivation forms.

Roughly speaking, a derivation form represents a path through the unification-driven leftmost outermost narrowing tree starting from its root. Technically, a derivation form is a word of triples where the first two components constitutes derivation forms of $\overset{u}{\rightsquigarrow}_{\hat{\mathcal{R}}}$. The additional third component contains an interval which includes the indices of the rules that are not yet applied. In Figure 10 the derivation form representing the leftmost branch in Figure 3 is shown.

$$(equ(sh(z_1, \alpha), mi(\sigma(z_2, \alpha))), \varphi_\emptyset, [2, 2])\,(equ(\alpha, mi(\sigma(z_2, \alpha))), [z_1/\alpha], [5, 4])$$
$$(equ(\alpha, \sigma(mi(\alpha), mi(z_2))), [z_1/\alpha], [1, 0])$$

Figure 10: A deterministic derivation form.

Definition 8.1 The set of *derivation forms of the deterministic unification-driven leftmost outermost narrowing relation,* denoted by $DDF(\hat{\mathcal{R}})$, is the set

$$(T\langle \hat{F} \cup \Delta\rangle(\mathcal{V}) \times Sub(\mathcal{V}, \Delta) \times I(card(\hat{F}) \cdot card(\Delta))^* \cup (T\langle\Delta\rangle(\mathcal{V}) \times Sub(\mathcal{V}, \Delta)).$$

$\oplus$

The second set in the union includes the results of the deterministic unification-driven leftmost outermost narrowing relation; a result is a pair consisting of a normalform and an $(E_\mathcal{R}, \Delta)$-unifier. In the initial deterministic derivation forms, we must distinguish three cases to be consistent with the usage of deterministic derivation forms in Definition 8.3. If the term t in the first component is not in binding form and if it derives by $\overset{u}{\rightsquigarrow}_{\hat{\mathcal{R}}}$, then the interval contains the indices of the applicable rules. Recall that every index of a rule is greater than zero. If (t, φ) does not derive by $\overset{u}{\rightsquigarrow}_{\hat{\mathcal{R}}}$, then the interval is empty. This is denoted by $[1, 0]$. In this case, t is not in binding form. To indicate that t is in binding form, the interval contains only the element 0 which is not an index of a rule.

Definition 8.2 The set of *initial derivation forms of the deterministic unification-driven leftmost outermost narrowing relation,* denoted by $initDDF(\hat{\mathcal{R}})$, is the union of the following three sets:

1. $\{(t, \varphi_\emptyset, [m, n]) \mid t = equ(s, s')$ where $s, s' \in T\langle F \cup \Delta\rangle(\mathcal{V})$, t is not in binding form, $(t, \varphi_\emptyset)$ derives by $\overset{u}{\leadsto}_{\hat{\mathcal{R}}}$, and $[m, n] = \{\hat{\pi}(l \to r) \mid (t, \varphi_\emptyset) \overset{u}{\leadsto}_{\hat{\mathcal{R}}, l\to r} (t', \varphi)\}\}$.
2. $\{(t, \varphi_\emptyset, [1, 0]) \mid t = equ(s, s')$ where $s, s' \in T\langle F \cup \Delta\rangle(\mathcal{V})$, and $(t, \varphi_\emptyset)$ does not derive by $\overset{u}{\leadsto}_{\hat{\mathcal{R}}}\}$.
3. $\{(t, \varphi_\emptyset, [0, 0]) \mid t = equ(s, s')$ where $s, s' \in \mathcal{V}\}$. $\oplus$

Now we are able to define the deterministic unification-driven leftmost outermost narrowing relation.

Definition 8.3 The *deterministic unification-driven leftmost outermost narrowing relation associated with* $\hat{\mathcal{R}}$, denoted by $\overset{d}{\leadsto}_{\hat{\mathcal{R}}}$, is defined as follows: For every $\beta_1, \beta_2 \in DDF(\hat{\mathcal{R}})$: $\beta_1 \overset{d}{\leadsto}_{\hat{\mathcal{R}}} \beta_2$, if the following two conditions hold:

1. There are $\beta_1' \in DDF(\hat{\mathcal{R}})$, $(t, \varphi) \in T\langle \hat{F} \cup \Delta\rangle(\mathcal{V}) \times Sub(\mathcal{V}, \Delta)$, and $m, n \in \mathbb{N}$ such that $\beta_1 = \beta_1'(t, \varphi, [m, n])$.
2. $0 < m \leq n$ (application of a rule): Let $(t, \varphi) \overset{u}{\leadsto}_{\hat{\mathcal{R}}, l\to r} (t', \varphi')$ and let $\hat{\pi}(l \to r) = m$.
   (a) If $impO(t')$ exists and
      i. t' is not in binding form, (t', φ') derives by $\overset{u}{\leadsto}_{\hat{\mathcal{R}}}$, and $[m', n'] = \{\hat{\pi}(l' \to r') \mid (t', \varphi') \overset{u}{\leadsto}_{\hat{\mathcal{R}}, l'\to r'} (\bar{t}, \bar{\varphi})\}$,
      then $\beta_2 = \beta_1'(t, \varphi, [m+1, n])(t', \varphi', [m', n'])$.
      ii. (t', φ') does not derive by $\overset{u}{\leadsto}_{\hat{\mathcal{R}}}$, then $\beta_2 = \beta_1'(t, \varphi, [m+1, n])(t', \varphi', [1, 0])$.
      iii. t' is in binding form, then $\beta_2 = \beta_1'(t, \varphi, [m+1, n])(t', \varphi', [0, 0])$.
   (b) If $impO(t')$ does not exist, then $\beta_2 = (t', \varphi')$.

   $0 = m = n$ (t is in binding form): Let $(t, \varphi) \overset{u}{\leadsto}_{\hat{\mathcal{R}}, bm} (t', \varphi')$.
   (a) If $impO(t')$ exists and
      i. t' is not in binding form, (t', φ') derives by $\overset{u}{\leadsto}_{\hat{\mathcal{R}}}$, and $[m', n'] = \{\hat{\pi}(l' \to r') \mid (t', \varphi') \overset{u}{\leadsto}_{\hat{\mathcal{R}}, l'\to r'} (\bar{t}, \bar{\varphi})\}$,
      then $\beta_2 = \beta_1'(t, \varphi, [1, 0])(t', \varphi', [m', n'])$.
      ii. (t', φ') does not derive by $\overset{u}{\leadsto}_{\hat{\mathcal{R}}}$, then $\beta_2 = \beta_1'(t, \varphi, [1, 0])(t', \varphi', [1, 0])$.
      iii. t' is in binding form, then $\beta_2 = \beta_1'(t, \varphi, [1, 0])(t', \varphi', [0, 0])$.
   (b) If $impO(t')$ does not exist, then $\beta_2 = (t', \varphi')$.

   $m > n$ (backtracking): $\beta_2 = \beta_1'$. $\oplus$

Note that the case $0 = m < n$ does not occur, because $m = 0$ only if t' is in binding form and then $n = 0$, too. In Figure 11 the derivation by the deterministic unification-driven leftmost outermost narrowing relation associated with $\hat{\mathcal{R}}_1$ starting with the triple $(equ(sh(z_1, \alpha), mi(\sigma(z_2, \alpha))), \varphi_\emptyset, [1, 2])$ is shown. The result of this derivation is the leaf in the unification-driven leftmost outermost narrowing tree in Figure 3 which is marked by 'success!'.

As mentioned before, the deterministic unification-driven leftmost outermost narrowing relation is not complete, but it is correct.

Lemma 8.4 Let $t, s \in T\langle F \cup \Delta\rangle(\mathcal{V})$. If there exists a derivation $(equ(t, s), \varphi_\emptyset, [m, n]) \overset{d}{\leadsto}^*_{\hat{\mathcal{R}}} (t', \varphi)$ where $[m, n]$ is defined as in Definition 8.2, then φ is an $(E_\mathcal{R}, \Delta)$-unifier of t and s.

<u>*Proof:*</u> If $(equ(t, s), \varphi_\emptyset, [m, n]) \overset{d}{\leadsto}^*_{\hat{\mathcal{R}}} (t', \varphi)$, then (t', φ) is the label of the leftmost leave in the unification-driven leftmost outermost narrowing tree of $\mathcal{R}$ for $(equ(t, s), \varphi_\emptyset)$ which is labeled by an element of $T\langle\Delta\rangle(\mathcal{V}) \times Sub(\mathcal{V}, \Delta)$. Thus, there exists a derivation $(equ(t, s), \varphi_\emptyset)$ $\overset{u}{\leadsto}^*_{\hat{\mathcal{R}}} (t', \varphi)$ and, by Theorem 7.9, φ is an element of a ground complete set of $(E_\mathcal{R}, \Delta)$-unifiers of t and s. From Definition 6.9 it follows that, in particular, φ is an $(E_\mathcal{R}, \Delta)$-unifier of t and s. $\oplus$

In the following remark we discuss how the deterministic unification-driven leftmost outermost narrowing relation can be modified to increase its efficiency and to compute more than one $(E_\mathcal{R}, \Delta)$-unifier.

Remark 8.5 The number of steps in the derivation by the deterministic unification-driven leftmost outermost narrowing relation can be reduced, if the rightmost triple of the derivation form of $\overset{d}{\leadsto}_{\hat{\mathcal{R}}}$ is deleted immediately after the application of the last possible rule, i.e., if $0 \leq m = n$. Roughly speaking, in the unification-driven leftmost outermost narrowing tree, the information of a node is deleted, if we walk to its rightmost son, because this information is not needed in further backtracking. This modification is realized in the implementation of PROLOG on the Warren Abstract Machine in [War83] by the application of the `trust_me_else_fail`-instruction. By applying this instruction, a choice point that indicates the rules which are not yet applied, is deleted, if the last possible rule is applied.

We can also modify the definition of the deterministic unification-driven leftmost outermost narrowing relation to compute more than one $(E_\mathcal{R}, \Delta)$-unifier as follows:

- A component is added to every derivation form of $\overset{d}{\leadsto}_{\hat{\mathcal{R}}}$ that includes the set Eun of $(E_\mathcal{R}, \Delta)$-unifiers which are computed up to now.
- If there is the derivation step $\beta'_1(t, \varphi, [m, n]) \overset{d}{\leadsto}_{\hat{\mathcal{R}}} (t', \varphi')$, then φ' is put into Eun and the next $(E_\mathcal{R}, \Delta)$-unifier is computed by starting with the derivation form $\beta_2 = \beta'_1(t, \varphi, [m + 1, n])$.
- If β_2 is the empty word, then Eun is a ground complete set of $(E_\mathcal{R}, \Delta)$-unifiers, because the depth-first left-to-right traversal through the unification-driven leftmost outermost narrowing tree is finished. $\oplus$

$$
\begin{array}{ll}
 & (equ(sh(z_1,\alpha), mi(\sigma(z_2,\alpha))), \varphi_\emptyset, [1,2]) \\
\stackrel{d}{\rightsquigarrow}_{\hat{\mathcal{R}}_1} & (equ(sh(z_1,\alpha), mi(\sigma(z_2,\alpha))), \varphi_\emptyset, [2,2])\ (equ(\alpha, mi(\sigma(z_2,\alpha))), [z_1/\alpha], [4,4]) \\
\stackrel{d}{\rightsquigarrow}_{\hat{\mathcal{R}}_1} & (equ(sh(z_1,\alpha), mi(\sigma(z_2,\alpha))), \varphi_\emptyset, [2,2])\ (equ(\alpha, mi(\sigma(z_2,\alpha))), [z_1/\alpha], [5,4]) \\
 & (equ(\alpha, \sigma(mi(\alpha), mi(z_2))), [z_1/\alpha], [1,0]) \\
\stackrel{d}{\rightsquigarrow}_{\hat{\mathcal{R}}_1} & (equ(sh(z_1,\alpha), mi(\sigma(z_2,\alpha))), \varphi_\emptyset, [2,2])\ (equ(\alpha, mi(\sigma(z_2,\alpha))), [z_1/\alpha], [5,4]) \\
\stackrel{d}{\rightsquigarrow}_{\hat{\mathcal{R}}_1} & (equ(sh(z_1,\alpha), mi(\sigma(z_2,\alpha))), \varphi_\emptyset, [2,2]) \\
\stackrel{d}{\rightsquigarrow}_{\hat{\mathcal{R}}_1} & (equ(sh(z_1,\alpha), mi(\sigma(z_2,\alpha))), \varphi_\emptyset, [3,2]) \\
 & (equ(sh(z_3, \sigma(mi(z_4),\alpha)), mi(\sigma(z_2,\alpha))), [z_1/\sigma(z_3,z_4)], [1,2]) \\
\stackrel{d}{\rightsquigarrow}_{\hat{\mathcal{R}}_1} & (equ(sh(z_1,\alpha), mi(\sigma(z_2,\alpha))), \varphi_\emptyset, [3,2]) \\
 & (equ(sh(z_3, \sigma(mi(z_4),\alpha)), mi(\sigma(z_2,\alpha))), [z_1/\sigma(z_3,z_4)], [2,2]) \\
 & (equ(\sigma(mi(z_4),\alpha), mi(\sigma(z_2,\alpha))), [z_1/\sigma(\alpha,z_4)], [4,4]) \\
\stackrel{d}{\rightsquigarrow}_{\hat{\mathcal{R}}_1} & (equ(sh(z_1,\alpha), mi(\sigma(z_2,\alpha))), \varphi_\emptyset, [3,2]) \\
 & (equ(sh(z_3, \sigma(mi(z_4),\alpha)), mi(\sigma(z_2,\alpha))), [z_1/\sigma(z_3,z_4)], [2,2]) \\
 & (equ(\sigma(mi(z_4),\alpha), mi(\sigma(z_2,\alpha))), [z_1/\sigma(\alpha,z_4)], [5,4]) \\
 & (equ(\sigma(mi(z_4),\alpha), \sigma(mi(\alpha), mi(z_2))), [z_1/\sigma(\alpha,z_4)], [6,6]) \\
\stackrel{d}{\rightsquigarrow}_{\hat{\mathcal{R}}_1} & (equ(sh(z_1,\alpha), mi(\sigma(z_2,\alpha))), \varphi_\emptyset, [3,2]) \\
 & (equ(sh(z_3, \sigma(mi(z_4),\alpha)), mi(\sigma(z_2,\alpha))), [z_1/\sigma(z_3,z_4)], [2,2]) \\
 & (equ(\sigma(mi(z_4),\alpha), mi(\sigma(z_2,\alpha))), [z_1/\sigma(\alpha,z_4)], [5,4]) \\
 & (equ(\sigma(mi(z_4),\alpha), \sigma(mi(\alpha), mi(z_2))), [z_1/\sigma(\alpha,z_4)], [7,6]) \\
 & (\sigma(equ(mi(z_4), mi(\alpha)), equ(\alpha, mi(z_2))), [z_1/\sigma(\alpha,z_4)], [3,4]) \\
\stackrel{d}{\rightsquigarrow}_{\hat{\mathcal{R}}_1} & (equ(sh(z_1,\alpha), mi(\sigma(z_2,\alpha))), \varphi_\emptyset, [3,2]) \\
 & (equ(sh(z_3, \sigma(mi(z_4),\alpha)), mi(\sigma(z_2,\alpha))), [z_1/\sigma(z_3,z_4)], [2,2]) \\
 & (equ(\sigma(mi(z_4),\alpha), mi(\sigma(z_2,\alpha))), [z_1/\sigma(\alpha,z_4)], [5,4]) \\
 & (equ(\sigma(mi(z_4),\alpha), \sigma(mi(\alpha), mi(z_2))), [z_1/\sigma(\alpha,z_4)], [7,6]) \\
 & (\sigma(equ(mi(z_4), mi(\alpha)), equ(\alpha, mi(z_2))), [z_1/\sigma(\alpha,z_4)], [4,4]) \\
 & (\sigma(equ(\alpha, mi(\alpha)), equ(\alpha, mi(z_2))), [z_1/\sigma(\alpha,\alpha)], [3,3]) \\
\stackrel{d}{\rightsquigarrow}{}^{2}_{\hat{\mathcal{R}}_1} & (equ(sh(z_1,\alpha), mi(\sigma(z_2,\alpha))), \varphi_\emptyset, [3,2]) \\
 & (equ(sh(z_3, \sigma(mi(z_4),\alpha)), mi(\sigma(z_2,\alpha))), [z_1/\sigma(z_3,z_4)], [2,2]) \\
 & (equ(\sigma(mi(z_4),\alpha), mi(\sigma(z_2,\alpha))), [z_1/\sigma(\alpha,z_4)], [5,4]) \\
 & (equ(\sigma(mi(z_4),\alpha), \sigma(mi(\alpha), mi(z_2))), [z_1/\sigma(\alpha,z_4)], [7,6]) \\
 & (\sigma(equ(mi(z_4), mi(\alpha)), equ(\alpha, mi(z_2))), [z_1/\sigma(\alpha,z_4)], [4,4]) \\
 & (\sigma(equ(\alpha, mi(\alpha)), equ(\alpha, mi(z_2))), [z_1/\sigma(\alpha,\alpha)], [4,3]) \\
 & (\sigma(equ(\alpha, \alpha), equ(\alpha, mi(z_2))), [z_1/\sigma(\alpha,\alpha)], [6,5]) \\
 & (\sigma(\alpha, equ(\alpha, mi(z_2))), [z_1/\sigma(\alpha,\alpha)], [3,4]) \\
\stackrel{d}{\rightsquigarrow}{}^{2}_{\hat{\mathcal{R}}_1} & (\sigma(\alpha,\alpha), [z_1/\sigma(\alpha,\alpha), z_2/\alpha])
\end{array}
$$

Figure 11: A derivation by $\stackrel{d}{\rightsquigarrow}_{\hat{\mathcal{R}}_1}$.

9 Conclusion

In this paper we have formalized a universal unification algorithm for equational theories which are characterized by ctn-trs's. This algorithm is at least as efficient as the algorithm which is implied by Theorem 3 in [Ech88], but sometimes it is more efficient because many derivation steps which do not yield an $(E_{\mathcal{R}}, \Delta)$-unifier, are omitted. For this purpose, we have introduced the unification-driven leftmost outermost narrowing relation which is a combination of unification and leftmost outermost narrowing. Furthermore, we have formalized a deterministic version of our universal unification algorithm that formalizes a depth-first left-to-right traversal through a unification-driven leftmost outermost narrowing tree. Similar to deterministic algorithms for SLD-resolution, the deterministic universal unification algorithm presented in this paper, is not complete, but it is correct.

Two implementations of leftmost outermost reduction for special ctn-trs's which are called macro tree transducers, are formalized in [FV92, GFV91]. In our current research we modify the implementation in [GFV91] to an implementation of the presented deterministic universal unification algorithm by adding features for unification and backtracking to the implementation of leftmost outermost reduction [FVW92]. As further research investigation, we will modify this implementation to the implementation of a deterministic universal unification algorithm for equational theories which are characterized by modular tree transducers. Modular tree transducers are ctn-trs's which describe primitive recursive tree functions.

References

[DJ91] N. Dershowitz and J.P. Jouannaud. Notations for rewriting. *Bulletin of the EATCS*, 43:162–172, 1991.

[Ech88] R. Echahed. On completeness of narrowing strategies. In *CAAP*, pages 89–101. Springer-Verlag, 1988. LNCS 299.

[EV85] J. Engelfriet and H. Vogler. Macro tree transducers. *Journal of Computer and System Sciences*, 31:71–145, 1985.

[EV91] J. Engelfriet and H. Vogler. Modular tree transducers. *Theoretical Computer Science*, 78:267–304, 1991.

[Fay79] M. Fay. First-order unification in an equational theory. In *Proceeding of the 4th workshop on automated deduction, Austin*, pages 161–167, 1979.

[FHVV93] Z. Fülöp, F. Herrmann, S. Vagvölgyi, and H. Vogler. Tree transducers with external functions, *accepted for publication in Theoretical Computer Science* . 1993.

[Fri85] L. Fribourg. A logic programming language interpreter based on clausual superposition and rewriting. In *Proceedings of the IEEE International Symposium on logic programming*, pages 172–184. IEEE Computer Society Press, 1985.

[FV92] H. Faßbender and H. Vogler. An implementation of syntax directed functional programming on nested-stack machines. *Formal Aspects of Computing*, 4:341–375, 1992.

[FVW92] H. Faßbender, H. Vogler, and A. Wedel. Syntax-directed runtime-stack machine for the solution of E-unification with macro tree transducers. Technical report, University of Ulm, 1992. in preparation.

[GFV91] K. Gladitz, H. Faßbender, and H. Vogler. Compiler-based implementation of syntax directed functional programming. Technical Report 10, Aachen University of Technology, Fachgruppe Informatik, Ahornstr. 55, W-5100 Aachen, FRG, 1991.

[HK89] D. Hofbauer and R.D. Kutsche. *Grundlagen des maschinellen Beweisens*. Vieweg, 1989.

[HO80] G. Huet and D.C. Oppen. Equations and rewrite rules: a survey. In R. Book, editor, *Formal Language Theory: Perspectives and Open Problems*. Academic Press, New York, 1980.

[Hue80] G. Huet. Confluent reductions: abstract properties and applications to term rewriting systems. *J. Assoc. Comput. Mach.*, 27:797–821, 1980.

[Hul80] J.M. Hullot. Canonical forms and unification. In *Proceedings of the 5th conference on automated deduction, LNCS 87*, pages 318–334. Springer-Verlag, 1980.

[Hup78] U. Hupach. Rekursive Funktionen in mehrsortigen Algebren. *Elektron. Informationsverarb. Kybernetik*, 15:491–506, 1978.

[Lan75] D.S. Lankford. Canonical inference. Technical Report ATP-32, Department of Mathematics and Computer Science, University of Texas at Austin, 1975.

[Llo87] J.W. Lloyd. *Foundations of Logic Programming.* Springer-Verlag, 1987. Second, extended edition.

[Mat70] Y. Matiyasevich. Diophantine representation of recursively enumerable predicates. In *Proceedings of the Second Scandinavian Logic Symposium.* North-Holland, 1970.

[MH92] A. Middeldorp and E. Hamoen. Counterexamples to completeness results for basic narrowing. In *Conference on algebraic and logic programming*, pages 244–258. Springer-Verlag, 1992. LNCS 632.

[MM82] A. Martelli and U. Montanari. An efficient unification algorithm. *ACM Transactions on Programming Languages Systems*, 4:258–282, 1982.

[PW78] M. S. Paterson and M. N. Wegman. Linear unification. *J. Comput. System Sci.*, 16:158–167, 1978.

[Rob65] J.A. Robinson. A machine-oriented logic based on the resolution principle. *J. Assoc. Comput. Mach.*, 20:23–41, 1965.

[Sie89] J.H. Siekmann. Unification theory. *J. Symbolic Computation*, 7:207–274, 1989.

[War83] D.H.D. Warren. An abstract prolog instruction set. Technical Report 309, SRI International, 1983.

[You89] J.H. You. Enumerating outer narrowing derivations for constructor-based term rewriting systems. *Journal of Symbolic Computation*, 7:319–341, 1989.