ICOT Technical Report: TR-636

TR-636

Complete E-unification based on an extension of the Knuth-Bendix completion procedure

by

A. Ohsuga & K. Sakai

March, 1991

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03)3456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

Complete E-unification based on an extension of the Knuth-Bendix completion procedure

Akihiko Ohsuga
Systems & Software Engineering Laboratory
Toshiba Corporation
70, Yanagi-cho, Saiwai-ku
Kawasaki, Kanagawa 210, Japan
ohsuga@ssel.toshiba.co.jp

Kô Sakai
ICOT Research Center
1-4-28, Mita, Minato-ku
Tokyo 108, Japan
sakai@icot.or.jp

Abstract

A unifier is a substitution that makes two terms syntactically equal. In this paper, we discuss more semantical unification; equational unification, which computes a substitution that makes two terms equal modulo a congruence relation. As a result we will give a general complete procedure that enumerates equational unifiers for a given pair of terms under a given congruence.

1 Introduction

We assume the reader has elementary knowledge on universal algebra, in particular, on term rewriting systems (TRSs for short)[13].

In this paper, we define a *TRS* as an arbitrary set of pairs of terms. An element of a TRS is called a *rewrite rule*. Reduction by a TRS $\Re$ is denoted by symbols $\Rightarrow$, that is, we write $c[l\theta] \Rightarrow c[r\theta]$ if there is a rewrite rule $(l, r) \in \Re$, a substitution θ, and a context $c[\,]$. The reflexive transitive closure of $\Rightarrow$ is denoted by $\stackrel{*}{\Rightarrow}$ and the reflexive transitive symmetric closure by $\stackrel{*}{\Leftrightarrow}$. Note that $\stackrel{*}{\Leftrightarrow}$ is a congruence relation on terms.

Let $\simeq$ be a congruence relation on terms. A substitution θ is called a $\simeq$-*unifier* of terms s and t if $s\theta \simeq t\theta$. The set of all $\simeq$ unifiers of s and t are denoted $U(s,t)$. A substitution θ is said to be *more general* than another substitution ϕ under $\simeq$ (denoted $\theta \le \phi$) if there is a substitution ψ such that $(v\theta)\psi \simeq v\phi$ for any variable v. A subset C of $U(s,t)$ is said to be *complete* if, for any $\theta \in U(s,t)$, there exists a unifier $\theta' \in C$ such that $\theta' < \theta$. Moreover, a complete subset C is called the *minimum* if $\theta = \theta'$ for any $\theta, \theta' \in C$ such that $\theta \le \theta'$. If the minimum complete set exists, it is unique up to renaming of the variables in the substituted terms.

In the case of ordinary unification, the most general unifier always exists for any unifiable pair of terms[17]. In the case of $\simeq$-unification, the existence of the most general unifier is not guaranteed. In this situation, a complete set of $\simeq$-unifiers plays the role that the most general unifier plays in ordinary unification: a representative of all unifiers.

From a theoretical point of view, the minimum complete set may be the most interesting since it is unique and not redundant. However, there is no reality of computation of the minimum complete set for the following reasons. First of all, the minimum set may not exist. That is, there may be a complete set C of $\simeq$-unifiers with the following property: for any $\theta \in C$, there is a unifier $\theta' \in C$ such that $\theta' < \theta$ (that is, $\theta' \leq \theta$, but $\theta \not\leq \theta'$)[8]. Even if the minimum set exists, there may be no algorithms to enumerate its elements. Even if it is enumerable, it may need more cost to compute than other (redundant) complete sets.

Let $\Re$ be a finite TRS and $\overset{*}{\Leftrightarrow}$ be the congruence relation induced by $\Re$. In what follows, we will discuss $\overset{*}{\Leftrightarrow}$-unification. It is clear from the definition that, for any terms s and t the set $U(s,t)$ is recursively enumerable and complete. However, it is too redundant to be worthy of computation. The main problems in $\overset{*}{\Leftrightarrow}$-unification are the following.

(1) Is $\overset{*}{\Leftrightarrow}$-unifiability decidable?

(2) Does the minimum complete set of $\overset{*}{\Leftrightarrow}$-unifiers exist? Can it be enumerated?

(3) Is there a finite complete set of $\overset{*}{\Leftrightarrow}$-unifiers?

(4) Is there an efficient algorithm to enumerate a complete set of $\overset{*}{\Leftrightarrow}$-unifiers?

It is undecidable in general whether two given terms have a $\overset{*}{\Leftrightarrow}$-unifier. As for the answers to these problems on specific TRSs, there is a wide-ranged survey by Siekmann[20]. The result on AC-unification, in which TRS consists only of the associative and the commutative laws, seems to be the most important from a practical point of view. That is, the minimum complete set of AC-unifiers always exists and it is finite and computable[22, 7, 12].

In this paper, we address problem (4) for a general TRS. A $\overset{*}{\Leftrightarrow}$-unification algorithm is said to be *complete* if it enumerates a complete set of $\overset{*}{\Leftrightarrow}$ unifiers of s and t for a given TRS and terms s and t. As stated before, enumeration of $U(s,t)$ is a complete (but not interesting) $\overset{*}{\Leftrightarrow}$-unification procedure. What is interesting is a more efficient algorithm than simple enumeration of all unifiers.

$\overset{*}{\Leftrightarrow}$-unification algorithms based on narrowing[9] and basic narrowing[15, 4] have been proposed under the assumption that TRS is confluent and terminating. These are efficient but the assumption is seldom satisfied in actual cases. Gallier and Snyder proposed a general $\overset{*}{\Leftrightarrow}$-unification algorithm[10], but it does not seem efficient enough for actual applications.

We propose another general algorithm and prove its completeness. Its effectiveness was confirmed by experiments. The procedure is based on combination of the Knuth-Bendix completion[16, 14] (or, more precisely, completion without failure[2]) and narrowing. Let s and t be given two terms whose $\overset{*}{\Leftrightarrow}$-unifiers are wanted. The procedure applies narrowing to s and t, while constructing a confluent and terminating (infinite) TRS. Since, as shown in [14] and [2], a confluent and terminating TRS can be obtained virtually even if the completion process does not terminate, the narrowing process eventually enumerates a complete set of unifiers. Moreover, since the procedure is an extension of the Knuth-Bendix completion, it may obtain a finite confluent and terminating TRS on the way of $\stackrel{*}{\Leftrightarrow}$-unification. Once such a TRS is obtained, the subsequent process becomes ordinary narrowing. Therefore, Fay's result[9] is viewed as a special case.

The essential idea is common with the refutational theorem proving in first-order logic with equality proposed by Hsiang and Rusinowitch[11, 18]. The purpose of this paper is not to claim originality of the idea but to claim its naturality and effectiveness and to give a proof of its completeness from the viewpoint of equational unification.

2 Inference rules for equational unification

In the following discussion, let $\preceq$ be a fixed strong simplification order on terms, namely, a simplification order[6] which is total on ground terms. In the examples in this paper, we use the lexicographic subterm ordering[19] as such an order.

In the following discussions, we assume that given terms and rewrite rules do not have common variables for simplicity of discussion.

First we change the concept of reduction by TRS. Usually, a rewrite rule in a TRS is assumed to be used left to right only. In this paper, however, we do not assume this, that is, rewrite rules in a TRS are used in both directions. For the new definition of reduction, it is better to consider a rewrite rule as unordered pairs of terms. Therefore, from now on, we do not distinguish rules (l, r) and (r, l).

To be precise, the new definition of reduction is the following: a term t is *reduced* to another term u (denoted $t \Rightarrow u$) if $u \prec t$ and there is a rewrite rule (l, r) or (r, l), a context $c[\,]$, and a substitution σ such that $c[l\sigma] = t$, $c[r\sigma] = u$. It is a routine to verify that $\stackrel{*}{\Leftrightarrow}$ is a congruence relation.

This change of the definition of reduction has both an advantage and a disadvantage. The advantage is that, since $\prec$ is well-founded[6, 19], reduction $\Rightarrow$ is always terminating. The disadvantage is that congruence relation $\stackrel{*}{\Leftrightarrow}$ is somewhat weaker than that given by the old definition. For example, let us consider TRS $\{(x + y, y + x)\}$. Then, reduction $\Rightarrow$ is not terminating in the old sense since $s+t \Rightarrow t+s \Rightarrow s+t \Rightarrow t+s \Rightarrow \cdots$, but $v+w \stackrel{*}{\not\Leftrightarrow} w+v$ in the new sense for any variables v and w since neither $v+w \Rightarrow w+v$ nor $w + v \Rightarrow v + w$.

Hereafter, for notational convenience, we use symbol $\simeq$ to denote the congruence relation induced by a TRS in the old sense of reduction. Then, the word problem involves the decision of not $\stackrel{*}{\Leftrightarrow}$ but $\simeq$. However, in many cases, we can assume that terms are ground without loss of generality by substituting fresh constants to variables in the terms. And, for ground terms, relations $\simeq$ and $\stackrel{*}{\Leftrightarrow}$ coincide since $\preceq$ is total.

Next, we define narrowing, which is somewhat modified of that by Fay[9] as well. A term s is said to be *narrowed* to another term t (denoted $s \leadsto t$) if there are a non-variable subterm s_0 of s, a rewrite rule (l, r) such that $s_0\theta = l\theta$, $s\theta \not\prec t$, and $t = c[r]\theta$, where $s = c[s_0]$ and θ is the most general unifier of s_0 and l. If necessary, we suffix the most general unifier, for example, as $s \leadsto_\theta t$. In what follows, we often discuss narrowing of a pair of terms. Notation $(s_1, s_2) \leadsto_\theta (t_1, t_2)$ means that either $s_1 \leadsto_\theta t_1$ and $s_2\theta = t_2$, or $s_1\theta = t_1$ and $s_2 \leadsto_\theta t_2$.

Let us extend the definition of critical pairs[16] as well. Let (l_1, r_1) and (l_2, r_2) be rewrite rules and s be a non-variable subterm of l_2 unifiable with l_1. If $l_1\theta \not\preceq r_1\theta$ and $l_2\theta \not\preceq r_2\theta$, then pair $(c[r_1]\theta, r_2\theta)$ is called a *critical pair*, where $l_2 = c[s]$ and θ is the most general unifier of l_1 and s.

Let a TRS $\Re$ and terms s and t be given. A $\simeq$-unification procedure for s and t under $\Re$ is given below in the form of inference rules.

E-generation: $$\frac{(E,R,G,U)}{(E\cup\{(u_1,u_2)\},R,G,U)} \quad (u_1,u_2) \text{ is a critical pair between rules in } R$$

E-reduction: $$\frac{(E\cup\{(u_1,u_2)\},R,G,U)}{(E\cup\{(u_1,u_2')\},R,G,U)} \quad u_2 \Rightarrow u_2' \text{ by a rule in } R$$

E-deletion: $$\frac{(E\cup\{(u,u)\},R,G,U)}{(E,R,G,U)}$$

R-generation: $$\frac{(E\cup\{(u_1,u_2)\},R,G,U)}{(E,R\cup\{(u_1,u_2)\},G,U)}$$

G generation: $$\frac{(E,R,G\cup\{(u_1,u_2,\theta)\},U)}{(E,R,G\cup\{(u_1,u_2,\theta),(u_1',u_2',\theta\circ\theta')\},U)}$$
$$(u_1,u_2) \rightsquigarrow_{\theta'} (u_1',u_2') \text{ by a rule in } R$$

U-generation: $$\frac{(E,R,G\cup\{(u_1,u_2,\theta)\},U)}{(E,R,G\cup\{(u_1,u_2,\theta)\},U\cup\{\theta\circ\theta'\})}$$
$$\theta' \text{ is the most general unifier of } u_1 \text{ and } u_2$$

Both E and R are sets of rewrite rules. G is a set of triples (u_1, u_2, θ) (called *goals*) where u_1 and u_2 are terms and θ is a substitution. We do not distinguish triples (u_1, u_2, θ) and (u_2, u_1, θ) similarly to rewrite rules. U is a set of substitutions. At the beginning of the procedure, these are set as follows:

$$E = \Re, \quad R = \phi, \quad G = \{(s,t,\varepsilon)\}, \quad U = \phi$$

$\simeq$-unifiers of s and t are enumerated as elements of U.

When one of the inference rules is applied, a quadruple (E, R, G, U) is transformed to another quadruple (E', R', G', U'), denoted by $(E, R, G, U) \vdash (E', R', G', U')$. If necessary, the name of the applied inference rule are suffixed to symbol $\vdash$. Let

$$(E_0, R_0, G_0, U_0) \vdash (E_1, R_1, G_1, U_1) \vdash (E_2, R_2, G_2, U_2) \vdash \cdots$$

be a sequence of applications of the inference rules. We denote $\bigcup_{i=0}^{\infty} E_i$ by E_∞, $\bigcup_{i=0}^{\infty} R_i$ by R_∞, $\bigcup_{i=0}^{\infty} G_i$ by G_∞, and $\bigcup_{i=0}^{\infty} U_i$ by U_∞. An inference sequence is called *fair*, if it satisfies the following conditions.

(1) Any critical pair between rules in R_∞ is contained in E_∞.

(2) $\bigcup_{i=0}^{\infty} \bigcap_{j=i}^{\infty} E_j = \phi$.
(3) Any goal obtained from a goal in G_∞ and a rule in R_∞ by G-generation is contained in G_∞.
(4) Any substitution obtained from a goal in G_∞ by U-generation is contained in U_∞.

We claim that any fair inference sequence can enumerate a complete set of $\simeq$-unifiers as U_∞.

Example 2.1 Consider TRS $Re = \{(f(x,x), g(x)),\ (h(a), a)\}$ and terms $s = f(h(y), a)$ and $t = g(h(a))$, where $\preceq$ is the lexicographic subterm ordering based on total order $a < h < g < f$ on the function symbols. Then, the following is a fair inference sequence.

$$(E_0 = \{(f(x,x), g(x)), (h(a), a)\},\ R_0 = \phi,\ G_0 = \{(f(h(y), a), g(h(a)), \varepsilon)\},\ U_0 = \phi)$$
$$\vdash_{R\text{-generation}} \quad (E_1 = \{(h(a), a)\},\ R_1 = \{(f(x,x), g(x))\},\ G_1 = G_0,\ U_1 = \phi)$$
$$\vdash_{R\text{-generation}} \quad (E_2 = \phi,\ R_2 = R_1 \cup \{(h(a), a)\},\ G_2 = G_1,\ U_2 = \phi)$$
$$\vdash_{G\text{-generation}} \quad (E_3 = \phi,\ R_3 = R_2,\ G_3 = G_2 \cup \{(f(h(y), a), g(a), \varepsilon)\},\ U_3 = \phi)$$
$$\vdash_{G\text{-generation}} \quad (E_4 = \phi,\ R_4 = R_3,\ G_4 = G_3 \cup \{(f(a,a), g(a), [a/y])\},\ U_4 = \phi)$$
$$\vdash_{G\text{-generation}} \quad (E_5 = \phi,\ R_5 = R_4,\ G_5 = G_4 \cup \{(g(a), g(a), [a/y])\},\ U_5 = \phi)$$
$$\vdash_{U\text{-generation}} \quad (E_6 = \phi,\ R_6 = R_5,\ G_6 = G_5,\ U_6 = \{[a/y]\})$$

$\simeq$-unifier $[a/y]$ of $s = f(h(y), a)$ and $t = g(h(a))$ is obtained as an element of U_6 in the above sequence, where notation $[a/y]$ expresses the substitution θ such that $y\theta = a$ and $v\theta = v$ for any variables v other than y.□

3 Completeness of the unification procedure

First of all, we prove the soundness of the procedure in the previous section.

Theorem 3.1 Let

$$(\Re, \phi, \{s, t, \varepsilon\}, \phi) = (E_0, R_0, G_0, U_0) \vdash (E_1, R_1, G_1, U_1) \vdash (E_2, R_2, G_2, U_2) \vdash \cdots$$

be an inference sequence. Then, any element of U_∞ is an $\simeq$-unifier of s and t.□

proof: Let $\simeq_i$ be the congruence relation induced by $E_i \cup R_i$ in the old sense of reduction. Then it is easy to prove the following by induction on i.

(1) $\simeq_i = \simeq$.
(2) For any $(u_1, u_2, \theta) \in G_i$, $s\theta \simeq_i u_1$ and $t\theta \simeq_i u_2$.
(3) For any $\theta \in U_i$, $s\theta \simeq t\theta$.■

The proof of completeness of the procedure consists of two parts. First, R_∞ is proved to be confluent by proof transformation method[1, 3]. Second, narrowing is proved to be able to trace any rewriting by R_∞.

Let $\Re$ be a TRS, and s and t ground terms such that $s \overset{*}{\Leftrightarrow} t$. Then, from the definition, there is a finite sequence of terms

$$s = u_0 \Leftrightarrow u_1 \Leftrightarrow \cdots \Leftrightarrow u_m = t.$$

Let us define sequences of this form in a more general framework.

A sequence $s = u_0 \Xi_1 u_1 \Xi_2 \cdots \Xi_m u_m = t$ is called a *proof* of $s \overset{*}{\Leftrightarrow} t$ under E and R, if each u_i is a ground term and Ξ_i is one of the following symbols:

(1) $\Leftrightarrow$, which indicates that $u_{i-1} \Leftrightarrow u_i$ by E in the old sense of reduction.

(2) $\Leftarrow$, which indicates that $u_i \Rightarrow u_{i-1}$ by R in the new sense of reduction.

(3) $\Rightarrow$, which indicates that $u_{i-1} \Rightarrow u_i$ by R in the new sense of reduction.

A proof is said to be *normal* if it has the following form

$$s = s_0 \Rightarrow s_1 \Rightarrow \cdots \Rightarrow s_m \Rightarrow u \Leftarrow t_n \Leftarrow \cdots \Leftarrow t_1 \Leftarrow t_0 = t \ (m \geq 0, n \geq 0)$$

Now, we will define the weight of a proof. First, the weight $w(u\Xi u')$ of each step $u\Xi u'$ of a proof is defined as follows:

$$w(u \Leftrightarrow u') = \{u, u'\}, \quad w(u \Leftarrow u') = \{u'\}, \quad w(u \Rightarrow u') = \{u\}$$

where $\{u, u'\}$, $\{u'\}$, and $\{u\}$ are not sets but multi-sets, and are compared by the multi-set ordering[5]. The weight of a proof is defined as the multi-set of the weights of all the steps in the proof. Note that, since the weight of a step is a multi-set, the weight of a proof is a doubly-multi-set (a multi-set of multi-set of terms). The set of the weights of proofs is well-founded since the base order is well-founded. Let us denote the order also by $\preceq$.

Theorem 3.2 Let

$$(E_0 = \Re, R_0 = \phi, G_0, U_0) \vdash (E_1, R_1, G_1, U_1) \vdash (E_2, R_2, G_2, U_2) \vdash \cdots$$

be a fair inference sequence. Then, R_∞ is a confluent TRS for $\overset{*}{\Leftrightarrow}$ w.r.t. ground terms. In other words, for any ground terms s and t such that $s \overset{*}{\Leftrightarrow} t$, there exist a normal proof of $s \overset{*}{\Leftrightarrow} t$ by R_∞. □

proof: Since $s \overset{*}{\Leftrightarrow} t$, there is a proof of $s \overset{*}{\Leftrightarrow} t$ by E_0 and R_0, which is also a proof by E_∞ and R_∞ of course. Let $\mathcal{P}$ be a proof by E_∞ and R_∞ with minimal weight. We prove that $\mathcal{P}$ is a normal proof by R_∞. First we prove that $\mathcal{P}$ contains no steps of the form

$$c[u_1\theta] \Leftrightarrow c[u_2\theta] \qquad \text{(A)}$$

where $(u_1, u_2) \in E_i$ for some i. Suppose that such a step exists. From fairness condition (2), for some j such that $i < j$, rule (u_1, u_2) must be deleted from E_j; that is, inference rule E-reduction, E-deletion, or R-generation must be applied to (u_1, u_2).

If it is E-reduction, $(u_1, u) \in E_j$ (or $(u, u_2) \in E_j$) for some u such that $u_2 \Rightarrow u$ (or $u_1 \Rightarrow u$ by R_j). Therefore, by replacing the step of form (A) with two steps

$$c[u_1\theta] \Leftrightarrow c[u\theta] \Leftarrow c[u_2\theta] \quad (\text{or} \quad c[u_1\theta] \Rightarrow c[u\theta] \Leftrightarrow c[u_2\theta]),$$

we can obtain a new proof $\mathcal{P}'$. Comparing the weight of the steps, that is, $\{c[u_1\theta], c[u_2\theta]\}$ in $\mathcal{P}$ and $\{c[u_1\theta], c[u\theta]\}, \{c[u_2\theta]\}$ (or $\{c[u_1\theta]\}, \{c[u\theta], c[u_2\theta]\}$) in $\mathcal{P}'$, we can easily see that $w(\mathcal{P}') \preceq w(\mathcal{P})$, which contradicts that $\mathcal{P}$ has minimal weight. If the inference step is E-deletion, u_1 must be equal to u_2. Therefore, by simply removing the step of form (A), we can obtain a new proof, which again contradicts that $\mathcal{P}$ has minimal weight. If the inference step is R-generation, R_j contains rule (u_1, u_2). In this case, the step of form (A) can be replaced with

$$c[u_1\theta] \Rightarrow c[u_2\theta] \quad \text{or} \quad c[u_1\theta] \Leftarrow c[u_2\theta]$$

since $\preceq$ is total for ground terms, and a contradiction follows. Next, we prove that $\mathcal{P}$ contains no steps of the form

$$t_1 \Leftarrow t \Rightarrow t_2 \tag{B}$$

Suppose that there are steps of form (B), in which term t is reduced in two ways, say, to t_1 by rule $(l_1, r_1) \in R_i$ and to t_2 by rule $(l_2, r_2) \in R_j$. There are several cases. First assume that the reduced parts do not overlap, that is t, t_1, and t_2 have forms $c[l_1\theta_1, l_2\theta_2]$, $c[r_1\theta_1, l_2\theta_2]$, and $c[l_1\theta_1, r_2\theta_2]$. In this case, by replacing the steps of form (B) with

$$t_1 \Rightarrow c[r_1\theta_1, r_2\theta_2] \Leftarrow t_2$$

we can obtain a new proof, which contradicts that $\mathcal{P}$ has minimal weight. Next assume that the reduced parts overlap. Since the discussion is symmetrical, we can assume without loss of generality that $t = d[c[l_1\theta_1]] = d[l_2\theta_2]$, $t_1 = d[c[r_1\theta_1]]$, and $t_2 = d[r_2\theta_2]$. If $l_1\theta_1$ occurs at a variable position in l_2, we can easily arrive at a contradiction similarly to the non-overlapping case. Otherwise, $(c[r_1\theta_1], r_2\theta_2)$ is an instance of a critical pair of rules (l_1, r_1) and (l_2, r_2). From fairness condition (1), the critical pair must be in some E_k. Then, by replacing the steps of form (B) with

$$t_1 = d[c[r_1\theta_1]] \Leftrightarrow d[r_2\theta_2] = t_2,$$

we arrive at a contradiction again. Thus, we have proved that $\mathcal{P}$ contains no steps of form (A) or (B). Such a proof is clearly normal.■

If there is a normal proof

$$s_0 \Rightarrow s_1 \Rightarrow \cdots \Rightarrow s_m \Rightarrow u \Leftarrow t_n \Leftarrow \cdots \Leftarrow t_1 \Leftarrow t_0 \tag{C}$$

we can always convert it to a one-way reduction sequence of pairs of terms of the following form:

$$(s_0, t_0) = p_0 \Rightarrow p_1 \Rightarrow \cdots \Rightarrow p_{m+n} = (u, u). \tag{C'}$$

In each step, either the left or the right element of pairs is reduced. In what follows, sequences of form (C') are called normal proofs instead of those of form (C) for simplicity of discussion.

A substitution σ is said to be *irreducible* if $v\sigma$ is irreducible for any variable v.

Theorem 3.3 (Hullot[15]) Let s be a term and θ be an irreducible substitution. Then, for any sequence of reduction

$$s\theta = t_0 \Rightarrow t_1 \Rightarrow \cdots \Rightarrow t_n,$$

there is a sequence of narrowing

$$s = u_0 \rightsquigarrow_{\theta_0} u_1 \rightsquigarrow_{\theta_1} \cdots \rightsquigarrow_{\theta_{n-1}} u_n$$

and a sequence of irreducible substitutions $\psi_0, \psi_1, \ldots, \psi_n$ such that

$$t_i = u_i\psi_i \ (i = 0, 1, \ldots, n)$$

and

$$\theta = \psi_0 = \theta_0 \circ \psi_1 = \cdots = \theta_0 \circ \cdots \circ \theta_{n-1} \circ \psi_n \square$$

In the original form of the above theorem, the concepts of reduction and narrowing are the conventional left-to-right ones, the TRS is assumed to be confluent and terminating, and substitution θ is assumed to be normal. However, the above form of the theorem can also be proved in the same way as the original.

Now, we are ready to prove the completeness of the $\simeq$-unification procedure.

Theorem 3.4 Let $\Re$ be a TRS, s and t be terms, and

$$(\Re, \phi, (s, t, \varepsilon), \phi) = (E_0, R_0, G_0, U_0) \vdash (E_1, R_1, G_1, U_1) \vdash (E_2, R_2, G_2, U_2) \vdash \cdots$$

be a fair inference sequence. Then, U_∞ is a complete set of $\simeq$ unifiers of s and t. That is, for any $\simeq$-unifier θ of s and t, there is a substitution $\theta' \in U_\infty$ more general than θ.$\square$

proof: By replacing variables in $s\theta$ and $t\theta$ with fresh constants, we can assume that $s\theta$ and $t\theta$ are ground terms without loss of generality. Moreover, by replacing the value of θ at each variable with its normal form w.r.t. R_∞, we can assume that θ is irreducible. Since $s\theta \simeq t\theta$, there is a term u and a normal proof

$$(s, t)\theta = p_0 \Rightarrow p_1 \Rightarrow \cdots \Rightarrow p_n = (u, u)$$

by R_∞. Then, from Theorem 3.3, there is a sequence of narrowing by R_∞

$$(s, t) = (s_0, t_0) \rightsquigarrow_{\theta_0} (s_1, t_1) \rightsquigarrow_{\theta_1} \cdots \rightsquigarrow_{\theta_{n-1}} (s_n, t_n)$$

and a sequence of irreducible substitutions $\psi_0, \psi_1, \ldots, \psi_n$ such that $p_i = (s_i, t_i)\psi_i$ ($i = 0, 1, \ldots, n$) and

$$\theta = \psi_0 = \theta_0 \circ \psi_1 = \cdots = \theta_0 \circ \cdots \circ \theta_{n-1} \circ \psi_n.$$

From fairness condition (3), we can easily prove by induction that, for each i, $(s_i, t_i, \theta_0 \circ \cdots \circ \theta_{i-1}) \in G_\infty$, in particular, $(s_n, t_n, \theta_0 \circ \cdots \circ \theta_{n-1}) \in G_\infty$. Since $s_n\psi_n = u = t_n\psi_n$, s_n and t_n are unifiable. Let ψ be the most general unifier of s_n and t_n. Then, from fairness condition (4), $\theta' = \theta_0 \circ \cdots \circ \theta_{n-1} \circ \psi \in U_\infty$, which is more general than $\theta = \theta_0 \circ \cdots \circ \theta_{n-1} \circ \psi_n$.■

As shown in Theorem 3.2, the $\simeq$-unification is an extension of the Knuth-Bendix completion procedure. In particular, if $R_i = R_\infty$ for some i, a finite confluent and terminating TRS is obtained after a finite number of steps of inference. Then, the subsequent process can be assumed to consist only of G generations and U-generations since the other rules cause no essential change in R_i, G_i, and U_i. Therefore, the procedure can be viewed as an extension of Fay's procedure. Moreover, if $G_j = G_\infty$ for some j (in fact, Example 2.1 is this case), we can obtain a finite complete set U_∞ of $\simeq$-unifiers of s and t. Note that, even in this case, U_∞ is not necessarily the minimum complete set.

4 Implementation issues and examples

There are a lot of things to be considered for efficiency in actual implementation of the procedure discussed in the previous section.

If the proof of Theorem 3.2 is examined, it can be easily seen that the inference rules E-reduction and E-deletion do not contribute to the completeness of the procedure. In fact, these rules are introduced for efficiency. To improve efficiency further, the following inference rules should be taken into consideration. If these rules are given priority over the generation rules, they will save a lot of time by not applying useless inferences.

R-reduction: $$\frac{(E, R \cup \{(u_1, u_2)\}, G, U)}{(E \cup \{(u_1, u_2')\}, R, G, U)} \quad u_2 \to u_2' \text{ by a rule in } R$$

G-reduction: $$\frac{(E, R, G \cup \{(u_1, u_2, \theta)\}, U)}{(E, R, G \cup \{(u_1, u_2', \theta)\}, U)} \quad u_2 \to u_2' \text{ by a rule in } R$$

G-deletion: $$\frac{(E, R, G \cup \{(u_1, u_2, \theta)\}, U)}{(E, R, G, U)}$$

θ is reducible by R or an element of U is more general than θ

U-deletion: $$\frac{(E, R, G, U \cup \{\theta\})}{(E, R, G, U)}$$

θ is reducible by R or an element of U is more general than θ

The reader can clearly see the role of R-reduction and G-reduction. Rules G-deletion and U-deletion essentially play the role that the basic narrowing plays in Hullot's procedure[15].

Even if the above inference rules are also employed, the procedure is still complete. To prove its completeness, however, the proof order and the limits need more subtle treatment, and this would introduce a simple but long discussion, which we have avoided in the proof of Theorem 3.2. For example, if R-reduction is employed, R_∞ must not be defined as $\bigcup_{i=1}^{\infty} R_i$ but as $\bigcup_{i=1}^{\infty}\bigcap_{j=i}^{\infty} R_j$, since R_i is no longer increasing.

We will show several examples of $\simeq$-unifications in combinatory logic. In the examples, we use the strong simplification order $\preceq$ induced by lexicographic subterm ordering. Terms of the form $*(\cdots(*(x,y),\cdots),z)$ are abbreviated to the form $xy\cdots z$ in the following inference sequence.

Example 4.1 An identity combinator **i**is defined as a combinator with property $\forall x\ \mathbf{i}x = x$. Here, we show the example of automatic construction of **i**from **s**and **k**by $\simeq$-unification. Let E_0 be $\{(\mathbf{s}xyz, xz(yz)), (\mathbf{k}xy, x)\}$ (that is, consist of the rules for **s**and **k**), and G_0 be $\{(vc, c, \varepsilon)\}$. Function symbols are ordered as $c < \mathbf{k} < \mathbf{s} < *$.

$$(E_0=\{(\mathbf{s}xyz, xz(yz)), (\mathbf{k}xy, x)\},\ R_0=\phi,\ G_0=\{(vc,c,\varepsilon)\},\ U_0=\phi)$$
$$\vdash_{R\text{-generation}}\ (E_1=\{(\mathbf{s}xyz, xz(yz))\},\ R_1=\{(\mathbf{k}xy,x)\},\ G_1=G_0,\ U_1=\phi)$$
$$\vdash_{R\text{-generation}}\ (E_2=\varepsilon,\ R_2=R_1\cup\{(\mathbf{s}xyz, xz(yz))\},\ G_2=G_1,\ U_2=\phi)$$
$$\vdash_{E\text{-generation}}\ (E_3=\{(\mathbf{sk}xy, y)\},\ R_3=R_2,\ G_3=G_2,\ U_3=\phi)$$
$$\vdash_{R\text{-generation}}\ (E_4=\varepsilon,\ R_4=R_3\cup\{(\mathbf{sk}xy,y)\},\ G_4=G_3,\ U_4=\phi)$$
$$\vdash_{G\text{-generation}}\ (E_5=\varepsilon,\ R_5=R_4,\ G_5=G_4\cup\{(c,c,[\mathbf{sk}x/v])\},\ U_5=\phi)$$
$$\vdash_{U\text{-generation}}\ (E_6=\varepsilon,\ R_6=R_5,\ G_6=G_5,\ U_6=\{[\mathbf{sk}x/v]\})$$

Thus, $v = \mathbf{sk}x$ is obtained as an $\simeq$-unifier.□

Remark: Strictly speaking, an $\simeq$-unifier of vc and c is not necessarily an identity combinator, since it may depend on c (that is, the term substituted to v may contain c as its subterm). If we want to restrain such a unifier from being generated, we should $\simeq$-unify both sides of $vc(v) \neq c(v)$, which is the Skolem form of the negation of $\forall x\ vx = x$.

Example 4.2 Next let us try the mockingbird problem[21]. A mockingbird is a combinator **m**with property $\forall x\ \mathbf{m}x = xx$. The problem is to construct a fixed point of a given combinator c from **m**, **b**, and c itself, where **b** is a composition combinator, which has property $\forall x\ \forall y\ \forall z\ \mathbf{b}xyz = x(yz)$. A fixed point of c is defined as a combinator f with property $cf = f$.

We set E_0 to be $\{(\mathbf{m}x, xx), (\mathbf{b}yzw, y(zw))\}$, R_0 to be ϕ, G_0 be $\{(cv, v, \varepsilon)\}$, and U_0 to be ϕ, and execute the $\simeq$-unification procedure. order $c < m < b <$ of function symbols We do not trace the details, but a fixed point of c is obtained through the following process.

(1) New rule $(\mathbf{m}(\mathbf{b}yz), y(z(\mathbf{b}yz)))$ is obtained as a critical pair of rules $(\mathbf{m}x, xx)$ and $(\mathbf{b}yzw, y(zw))$ by E-generation.

(2) New goal $(\mathbf{m}(\mathbf{b}cz), z(\mathbf{b}cz), [z(\mathbf{b}cz)/v])$ is obtained from goal (cv, v, ε) and rule $(\mathbf{m}(\mathbf{b}yz), y(z(\mathbf{b}yz)))$ by G-generation.

(3) Finally, we can generate $\simeq$-unifier $[\mathbf{m}(\mathbf{b}c\mathbf{m})/v]$ of cv and v from the above goal $(\mathbf{m}(\mathbf{b}cz), z(\mathbf{b}cz), [z(\mathbf{b}cz)/v])$ by U-generation, and $\mathbf{m}(\mathbf{b}c\mathbf{m})$ is a fixed point of c in fact.□

Acknowledgments

This work is based on the intelligent programming systems (IPS) research in the FGCS project. We would like to thank Dr. Fuchi (ICOT Director) and Dr. Hasegawa (Chief of ICOT 5th Laboratory) for the opportunity to carry out this research.

References

1. Bachmair, L., Dershowitz, N., and Hsiang, J.: Orderings for equational proofs, In *Proc. 1st IEEE Symposium on Logic in Computer Science* (1986), pp. 346–357.

2. Bachmair, L., Dershowitz, N., and Plaisted, D. A.: Completion without failure, In *Proc. Colloquium on Resolution of Equations in Algebraic Structures* (1987).

3. Bachmair, L.: Proof Normalization for Resolution and Paramodulation, In *Proc. 3rd Int. Conf. Rewriting Techniques and Applications*, Lecture Notes in Computer Science 355, Springer-Verlag (1989), pp. 15–28.

4. Bosco, P. G., Giovannetti, E., and Moiso, C.: Refined strategies for semantic unification, In *Proc. TAPSOFT '87*, Lecture Notes in Computer Science 250, Springer-Verlag (1987), pp. 276–290.

5. Dershowitz, N. and Manna, Z.: Proving termination with multiset orderings, *Comm. ACM*, Vol. 22, No. 8 (1979), pp. 465–467.

6. Dershowitz, N.: Orderings for term-rewriting systems, *Theor. Comput. Sci.*, Vol. 17, No. 3 (1982), pp. 279–301.

7. Fages, F.: Associative-commutative unification, In *Proc. 7th Int. Conf. on Automated Deduction*, Lecture Notes in Computer Science 170, Springer-Verlag (1984), pp. 194–208.

8. Fages, F.: Complete sets of unifiers and matchers in equational theories, *Theor. Comput. Sci.*, Vol. 43 (1986), pp. 189–200.

9. Fay, M. J.: First-Order Unification in an Equational Theory, In *Proc. 4th Workshop on Automated Deduction* (1979), pp. 161–167.

10. Gallier, J. and Snyder, W.: A General complete E-unification procedure, In *Proc. 2nd Int. Conf. Rewriting Techniques and Applications*, Lecture Notes in Computer Science 256, Springer-Verlag (1987), pp. 216–227.

E

11. Hsiang, J. and Rusinowitch, M.: On Word Problems in Equational Theories, In *Proc. 14th Int. Colloquium Automata, Languages and Programming*, Lecture Notes in Computer Science 267, Springer-Verlag (1987), pp. 54–71.

12. Huet, G.: An algorithm to generate the basis of solutions to homogeneous linear Diophantine equations, *Inf. Process. Lett.*, Vol. 7 (1978), pp. 144–147.

13. Huet, G. and Oppen, D. C.: Equations and Rewrite Rules: A Survey, In Book, R., editor, *Formal Languages: Perspective and Open Problems*, Academic Press (1980), pp. 349–405.

14. Huet, G.: A complete proof of correctness of the Knuth-Bendix completion algorithm, *J. Comput. Syst. Sci.*, Vol. 23, No. 1 (1981), pp. 11–21.

15. Hullot, J. M.: Canonical forms and unification, In *Proc. 5th Int. Conf. on Automated Deduction*, Lecture Notes in Computer Science 87, Springer-Verlag (1980), pp. 318–334.

16. Knuth, D. E. and Bendix, P. B.: Simple word problems in universal algebras, In Leech, J., editor, *Proc. Computational problems in abstract algebra*, Pergamon Press,Oxford (1970), pp. 263–297 ;also in *Automation of Reasoning 2* (Siekmann, J. H. and Wrightson eds.), Springer-Verlag (1983), pp. 342–376.

17. Robinson, J.: A machine-oriented logic based on the resolution principle, *J. ACM*, Vol. 12, No. 1 (1965), pp. 23–41.

18. Rusinowitch, M.: Theorem-Proving with Resolution and Superposition: an Extension of the Knuth and Bendix Procedure to a Complete Set of Inference, In *Proc. Int. Conf. on Fifth Generation Computer Systems* (1988), pp. 524–531.

19. Sakai, K.: An ordering method for term rewriting systems, Tech. Report TR-062, ICOT (1984).

20. Siekmann, J. H.: Unification Theory, *J. Symbolic Computation*, Vol. 7, No. 3&4 (1989), pp. 207–274.

21. Smullyan, R., M.: *To Mock a Mockingbird*, Alfred A. Knopf, Inc. (1985).

22. Stickel, M. E.: A Unification Algorithm for Associative-Commutative Functions, *J. ACM*, Vol. 28, No. 3 (1981), pp. 423–434.