UNIFICATION OF IDEMPOTENT FUNCTIONS

S. Kuhner, Ch. Mathis, P. Raulefs, J. Siekmaiin
Institut fur Informatik I
Universitat Karlsruhe
D-7500 Karlsruhe 1
West Germany

- Extended Abstract -

1. Introduction

For almost as long as attempts at proving theorems by machines have been made, a critical problem has been well known: Equational axioms, if left without precautions in the data base of an automatic theorem prover (ATP), will force the ATP to go astray.

Four approaches to cope with equality axioms have been proposed:

(1) To write the axioms into the data base, and use an additional rule of inference, such as paramodulation.
(2) To use special "rewrite rules".
(3) To design special inference rules incorporating these axioms.
(4) To develop special unification algorithms incorporating these axioms.

At least for equational axioms, the? last approach (4) appears to be most promising.

So far we have concentrated on the axioms of associativity, commutativity, idempotence and various combinations of these. In this paper, algorithms are presented for theories of idempotence and for idempotence together with commutativity. Idempotent functions appear in group theory, practical examples being proofs about substitutions which are idempotent if they are in normal form. See e.g. the proof of Theorem 3-2 in this paper. A historical example is how Luckham's program verifier found the correctness of Robinson's unification algorithm only after the idempotence for unifiers had been added.

The main results are that the unification problem for idempotence is decidable, and the set of all unifiers is finite, but not a singleton in general.

2. Algorithm for I-Unification

2.1. Intuitive Overview. Our algorithm unifying two terms s and t is split up into two interlocking parts:

(1) Collapsing phase: In both s and t we look for subterms (r^,^) s.t. r and r can be unified by a substitution p to r. Then, applying p to s and t causes subterms (r ,r2) to "collapse" to r, generating a new unification problem.

(2) R-unification phase,: Unification problems resulting from the collapsing phase are solved by the algorithm RUNIFY. Essentially, RUNIFY follows the idea of Robinson's unification algorithm except for the way an atom and a non-atomic term is unified. (RUNIFY returns a success/failure message (SUCC/FAIL) and a substitution (empty upon failure).) For space limitations we can not state the full algorithms but the following examples may indicate the basic idea:

(1) s = f(f(a,x),c) and t = f(x,c) has mgu $\delta=\{(x\leftarrow a)\}$
(2) s = f(f(a,b),f(z,d)) t = f(x,y)
s has one collapse: f(f(a,b),d) $\{(z\leftarrow d)\}$
t has one collapse: y $\{(x\leftarrow y)\}$
the unifiers are:

$\delta_1=\{(x\leftarrow f(a,b)),(y\leftarrow f(z,d))\}$

$\delta_2=\{(x\leftarrow f(f(a,b),f(z,d)),(y\leftarrow f(f(a,b),f(z,d))\}$

$\delta_3=\{(x\leftarrow f(a,b)),(y\leftarrow d),(z\leftarrow d)\}$

$\delta_4=\{(x\leftarrow f(f(a,b),d),(y\leftarrow f(f(a,b),d)),(z\leftarrow d)\}$

In a modified version of the algorithm δ_3 and δ_4, which are instances of δ_1 and δ_2 respectively, are not generated. However even the modified algorithm is not minimal in general.

3. Completeness of the Unification Algorithm

The completeness of the unification algorithm IUNIFY is shown in two steps:

(1) First, we show that for any unifier δ of terms s and t not collapsing any non-leaf nodes ("immediate unifier"), RUNIFY applied to s and t successfully returns a unifier σ s.t. $\delta \sqsubseteq \sigma$ I.e. RUNIFY is complete w.r.t. immediate unifiers.

(2) We then show that for any arbitrary unifier σ of s and t we can find substitutions τ and τ' s.t. τs is a collapse of s, and $\tau' t$ is a collapse of t,

1. $\tau s \in$ collapse(s) and $\tau' t \in$ collapse(t), and
2. there is an immediate unifier Θ with $\Theta\tau s = \Theta\tau' t$ and $\sigma \sqsubseteq \Theta\circ\tau\circ\tau'$.

3.1. Partial Completeness of RUNIFY

Def.: For any term t, the set ICnode(t) of immediately collapsable nodes in t is given by: For any k, labelling a node in the tree representation of t: If the largest subterm with root k is identical to a subterm with a root being a brother to k, then $k \in$ ICnode(t). Taking NLnode(t) to be the set of all non-leaf nodes in a term t, an immediate unifier is a substitution not immediately collapsing any non-leaf nodes:

Def.: A unifier σ of two terms t_1 and t_2 is called an immediate unifier for t_1 and t_2 iff

$\text{ICnode}(\sigma t_i) \cap \text{NLnode}(t_i) = \emptyset \quad (i = 1,2)$.

To show the partial completeness of RUNIFY in Theorem 3-2, we make use of the following lemma:

Lemma 3-1. Let σ be an immediate unifier for two terms t_1 and t_2, and disagree $(t_1,t_2) = \{s_1,s_2\}$. Then

$$\sigma_1 s \equiv \sigma_2 s$$

Theorem 3-2 [Completeness of RUNIFY w.r.t. Immediate Unifiers]

Let δ be an immediate unifier of two terms t_1 and t_2. Then RUNIFY terminates successfully with RUNIFY $t_1 t_2$ = (SUCC,σ) and $\delta \sqsubseteq \sigma$.

3.2. Completeness of IUNIFY

To establish the overall completeness of the unification algorithm, two auxiliary lemmas about substitutions are needed:

Lemma 3-3. Let ρ and σ be substitutions with $\rho \sqsubseteq \sigma$. Then, for any substitution τ, $\rho\circ\tau \sqsubseteq \sigma\circ\tau$.

Lemma 3-4. For any substitutions δ, σ_1, and σ_2:
If $\delta \sqsubseteq \sigma_1$ and $\delta \sqsubseteq \sigma_2$ then $\delta \sqsubseteq \sigma_2 \circ \sigma_1$.

Theorem 3-5. Let δ be any unifier for two terms t_1 and t_2. Then, there are substitutions τ_1 and τ_2 s.t.

(1) $\tau_i t_i$ is a collapse of t_i $(i = 1,2)$, and
(2) RUNIFY applied to $\tau_1 t_1$ and $\tau_2 t_2$ successfully returns an immediate unifier Θ with $\delta \sqsubseteq \Theta\circ\tau_1\circ\tau_2$.

The completeness of IUNIFY is then an immediate consequence of Theorem 3-5.