Unification of Terms with Term-Indexed Variables

Igor L. Tandetnik

Moscow State University of Railway Transport (MIIT),
Obrazcova 15, 101475, Moscow, Russia

Abstract. We consider the unification problem for generalized terms. The syntax of generalized terms allows the use of indexed variables, with indexes themselves being generalized terms. This leads to an infinite set of conditional equations. We propose a reduction to the finite conditional unification proble. We prove the existence of a most general unifier (m.g.u.) for both the finite and infinite cases. An efficient (cubic in time, linear in space) algorithm for computing the m.g.u. for these problems is developed.

1 Introduction

The classic unification problem is as follows: Given a finite set of pairs of terms (L_i, R_i), $i = 1, \ldots, N$, find a most general substitution (most general unifier, m.g.u) σ for which the equalities

$$L_i\sigma = R_i\sigma, \qquad i = 1, \ldots, N \tag{1}$$

hold.

The theory of most general unifiers is developed in [1] and [2] (see also the survey in [3]). Fast unification algorithms can be found in [4] and [5]. In [6] and [7], while dealing with problems of logical description and synthesis of reference structures, the following problem was introduced: terms being unified are allowed to contain variables of the form v_p where p is again a variable. This naturally leads to the following constraint on substitution, expressing functional dependency of an indexed variable on its index:

$$p\sigma = q\sigma \Rightarrow v_p\sigma = v_q\sigma. \tag{2}$$

The algorithm given in [6] and [7] can cope only with indexed variables which have index-free variables or constants as their indexes. In this paper, we consider a more general problem, in which any term is allowed as an index, including those containing other indexed variables. This leads to the necessity of considering infinite unifiers for (1), (2), i.e. arbitrary mappings defined on the set of all terms and commuting with functional symbols. We prove the existence of an idempotent m.g.u. σ for (1), (2) and give an algorithm calculating $t\sigma$ for any term t. Exploiting the ideas of [5], we give an effective realization of this algorithm which has time complexity $O(l^3)$, where l is the total length of all terms occurring in (1) together with the term t.

2 Consistent substitutions

Let us fix the set $V_{\mathrm{I}} = \{p_1, p_2, \ldots\}$ (index-free variables) and the set F of *functional symbols.*

The sets of *indexed variables* V_{II} and *terms* T are defined as the least sets satisfying the following conditions:

1. $V_{\mathrm{I}} \cup V_{\mathrm{II}} \subseteq T$.
2. If $f \in F$, $n = arity(f)$ and $t_1, \ldots, t_n \in T$ then $f(t_1, \ldots, t_n) \in T$.
3. If $t \in T$ then $v_t \in V_{\mathrm{II}}$.

Let $V = V_{\mathrm{I}} \cup V_{\mathrm{II}}$ be the set of all variables, $Var(t)$ the set of all variables occurring in t on the first level (not in indexes) and $AVar(t)$ the set of all variables occurring in t on any depth. For $W \subseteq V$ let $W^+ = \bigcup_{x \in W} AVar(x)$.

The substitution is a mapping $\sigma : T \to T$ satisfying the condition

$$f(t_1, \ldots, t_n)\sigma = f(t_1\sigma, \ldots, t_n\sigma)$$

(the substitution is defined uniquely by its values on variables). Let us define $Dom(\sigma) = \{x \in V \mid x\sigma \neq x\}$, $Val(\sigma) = \bigcup_{x \in Dom(\sigma)} Var(x\sigma)$, $Var(\sigma) = Dom(\sigma) \cup Val(\sigma)$ and $AVar(\sigma) = (Var(\sigma))^+$.

The substitution σ is called *finite* when the set $Dom(\sigma)$ is finite, and *infinite* otherwise.

The substitution σ is called W-consistent for some set $W \subseteq V$ if for each $v_s, v_t \in W$ the equality $s\sigma = t\sigma$ implies $v_s\sigma = v_t\sigma$. The V-consistent substitution is called *consistent.* Note that any consistent substitution σ with $Dom(\sigma) \neq \emptyset$ is infinite.

A consistent substitution $\bar{\sigma}$ is called a consistent extension of a substitution σ if $x\bar{\sigma} = x\sigma$ for each $x \in Dom(\sigma)$.

Theorem 1. *Let σ be W-consistent substitution, where $Var(\sigma) \subseteq W$ and $W^+ = W$. Let $\bar{\sigma}$ be the substitution satisfying the conditions*

$$\begin{array}{c} p\bar{\sigma} = p\sigma \qquad \textit{for } p \in V_{\mathrm{I}}, \\ v_t\bar{\sigma} = \begin{cases} v_t\sigma, & \textit{if } v_t \in W, \\ v_s\sigma, & \textit{if } v_t \notin W \textit{ and } \exists v_s \in W : s\bar{\sigma} = t\bar{\sigma}, \\ v_{t\bar{\sigma}}, & \textit{if } v_t \notin W \textit{ and } \neg\exists v_s \in W : s\bar{\sigma} = t\bar{\sigma}. \end{cases} \end{array} \tag{3}$$

Then 1. The substitution $\bar{\sigma}$ exists and is unique.

2. The substitution $\bar{\sigma}$ is a consistent extension of σ.

3. If σ is idempotent (i.e. $\sigma^2 = \sigma$) then $\bar{\sigma}$ is also idempotent and $\sigma\bar{\sigma} = \bar{\sigma}$ (and so $s\sigma = t\sigma$ implies $s\bar{\sigma} = t\bar{\sigma}$).

Proof. **1.** We introduce function $VD(t)$ on terms as follows:

1. $VD(p) = 0$ for $p \in V_{\mathrm{I}}$.
2. $VD(v_t) = 1 + VD(t)$.
3. $VD(f(t_1, \ldots, t_n)) = \max\{VD(t_1), \ldots, VD(t_n)\}$.

The first two lines of (3) uniquely define the substitution $\bar{\sigma}$ on terms t for which $Var(t) \subseteq V_{\mathrm{I}} \cup W$ (this includes all terms with the property $VD(t) = 0$). Moreover, for such terms $t\bar{\sigma} = t\sigma$.

Consider a variable $v_t \notin W$. We assume that the substitution $\bar{\sigma}$ is uniquely defined on all terms r with $VD(r) < VD(v_t)$. There are three possibilities.

Case 1: there is exactly one variable $v_s \in W$ for which $s\bar{\sigma} = t\bar{\sigma}$ (both sides of this equality are defined due to inductive hypothesis). Then by (3) $v_t\bar{\sigma} = v_s\sigma$.

Case 2: there are two variables $v_r, v_s \in W$ for which $r\bar{\sigma} = s\bar{\sigma} = t\bar{\sigma}$. It follows from the condition $W^+ = W$ that $Var(r) \subseteq W$ and $Var(s) \subseteq W$, and so $r\sigma = r\bar{\sigma} = s\bar{\sigma} = s\sigma$. But the substitution σ is W-consistent which implies $v_r\sigma = v_s\sigma$. Thus $v_t\bar{\sigma}$ is uniquely defined.

Case 3: there is no variable $v_s \in W$ for which $s\bar{\sigma} = t\bar{\sigma}$. Then $v_t\bar{\sigma} = v_{t\bar{\sigma}}$ where $t\bar{\sigma}$ is uniquely defined by inductive hypothesis.

2. It is obvious that the substitution $\bar{\sigma}$ is an extension of σ i.e. $x\bar{\sigma} = x\sigma$ for each $x \in Dom(\sigma)$. It remains to be proved that $\bar{\sigma}$ is consistent.

Suppose there are two variables v_s and v_t for which $s\bar{\sigma} = t\bar{\sigma}$. We shall prove that $v_s\bar{\sigma} = v_t\bar{\sigma}$. If $v_s, v_t \in W$ then $v_s\bar{\sigma} = v_s\sigma = v_t\sigma = v_t\bar{\sigma}$. If $v_s \in W$ but $v_t \notin W$ then $v_t\bar{\sigma} = v_s\sigma = v_s\bar{\sigma}$. If $v_s, v_t \notin W$ then $v_s\bar{\sigma} = v_{s\bar{\sigma}} = v_{t\bar{\sigma}} = v_t\bar{\sigma}$. So $\bar{\sigma}$ is consistent in all three cases.

3. Let x be a variable and $\tau = x\sigma$. It follows from the condition $Var(\sigma) \subseteq W$ that $Var(\tau) \subseteq V_{\mathrm{I}} \cup W$ whenever $x \in V_{\mathrm{I}} \cup W$. If $x \in W$ then $x\sigma\bar{\sigma} = \tau\bar{\sigma} = \tau\sigma = x\sigma^2 = x\sigma = x\bar{\sigma}$. If $x \notin W$ then $x\sigma = x$ and the equality $x\sigma\bar{\sigma} = x\bar{\sigma}$ holds.

We now prove that $\bar{\sigma}$ is idempotent. If $x \in V_{\mathrm{I}} \cup W$ then $x\bar{\sigma} = x\sigma$ and $x\bar{\sigma}^2 = x\sigma\bar{\sigma} = x\bar{\sigma}$.

Let $x = v_t \notin W$. If there is a variable $v_s \in W$ for which $s\bar{\sigma} = t\bar{\sigma}$ then $v_t\bar{\sigma}^2 = v_s\sigma\bar{\sigma} = v_s\bar{\sigma} = v_s\sigma = v_t\bar{\sigma}$. Suppose such a variable does not exist. We assume that $r\bar{\sigma}^2 = r\bar{\sigma}$ for all terms r with the property $VD(r) < VD(v_t)$. Thus $t\bar{\sigma}^2 = t\bar{\sigma}$ and there is no variable $v_s \in W$ for which $t\bar{\sigma}^2 = s\bar{\sigma}$. We have $v_t\bar{\sigma}^2 = v_{t\bar{\sigma}}\bar{\sigma} = v_{t\bar{\sigma}^2} = v_{t\bar{\sigma}} = v_t\bar{\sigma}$. □

The substitution $\bar{\sigma}$ from (3) is called the *standard consistent extension* of σ. Note that the conditions (3) provide a method to calculate $t\bar{\sigma}$ for any $t \in T$ whenever σ is computable (in particular, finite).

3 Unification algorithm

Consider the system of equations S of the form $L_i = R_i, i = 1, \ldots, N$. We define $Var(S) = \bigcup_{i=1}^{N}(Var(L_i) \cup Var(R_i))$ and $AVar(S) = (Var(S))^+$.

The substitution σ is called a most general (most general W-consistent, most general consistent) unifier of S, if σ is a (W-consistent, consistent) unifier of S and for any (W-consistent, consistent) unifier θ of S there exists a substitution λ such that $\theta = \sigma\lambda$ (if σ is idempotent, the last condition can be rewritten as $\theta = \sigma\theta$).

Theorem 2. *Let S be a system of equations, $W \subseteq V$, $W^+ = W$ and $Var(S) \subseteq W$. Let σ be an idempotent most general W-consistent unifier of S, $Var(\sigma) \subseteq W$. Then the standard consistent extension $\bar{\sigma}$ of σ is an idempotent most general consistent unifier of S.*

Proof. By the theorem 1 the substitution $\bar{\sigma}$ is an idempotent consistent unifier of S. We shall prove that it is also a most general consistent unifier. Let θ be some consistent unifier of S. As σ is a m.g.u. of S we have $\theta = \sigma\theta$. Thus $t\bar{\sigma}\theta = t\sigma\theta = t\theta$ for all terms t for which $Var(t) \subseteq V_I \cup W$ (this includes all terms with the property $VD(t) = 0$).

Consider a variable $v_t \notin W$. We assume that $r\bar{\sigma}\theta = r\theta$ for all terms r for which $VD(r) < VD(v_t)$. Suppose there is a variable $v_s \in W$ for which $s\bar{\sigma} = t\bar{\sigma}$. By applying the substitution θ to both sides of the last equality we have $s\bar{\sigma}\theta = t\bar{\sigma}\theta$. But $s\bar{\sigma}\theta = s\theta$ due to the fact that $Var(s) \subseteq W$ and $t\bar{\sigma}\theta = t\theta$ by the inductive hypothesis. As θ is consistent we have $v_s\theta = v_t\theta$. Thus $v_t\bar{\sigma}\theta = v_s\sigma\theta = v_s\theta = v_t\theta$.

Suppose there is no variable $v_s \in W$ for which $s\bar{\sigma} = t\bar{\sigma}$. By the inductive hypothesis $t\theta = t\bar{\sigma}\theta$ which leads to $v_t\theta = v_{t\bar{\sigma}}\theta$ because θ is consistent. Then $v_t\bar{\sigma}\theta = v_{t\bar{\sigma}}\theta = v_t\theta$. So $x\bar{\sigma}\theta = x\theta$ for any $x \in V$. □

Thus in order to find the most general consistent solution $\bar{\sigma}$ of S ($\bar{\sigma}$ in most cases is infinite) it is sufficient to construct only its finite part — the most general $AVar(S)$-consistent unifier σ of S. Then the procedure (3) allows one to calculate $t\bar{\sigma}$ for any term t. Furthermore, if term t is known beforehand we can assume without loss of generality that the system S contains the equation $t = t$. In this case $t\bar{\sigma} = t\sigma$ and the application of the procedure (3) is not necessary.

Let $unify(S)$ be a unification algorithm for terms without indexed variables (e.g. algorithm from [5]). Its input is the system of equations S, and its output is a pair $(bool, \sigma_0)$ with the following properties:

1. If $bool = false$ then the system S has no solutions.
2. If $bool = true$ then σ_0 is the most general unifier of S. Moreover, σ_0 is idempotent and conservative (i.e. $Var(\sigma_0) \subseteq Var(S)$).

The conservativity property provides the finiteness of σ_0.

Consider the algorithm $unify_{ind}(S)$, accepting system S as its input and constructing a pair $(bool, \sigma)$:

1. Let $k = 0$, $S_0 = S$.
2. Let $(bool, \sigma_k) = unify(S_k)$. If $bool = false$, stop with the result $(false, \sigma_k)$.
3. Find a pair $v_s, v_t \in AVar(S)$ for which $s\sigma_k = t\sigma_k$ but $v_s\sigma_k \neq v_t\sigma_k$. If there is no such pair, stop with the result $(true, \sigma_k)$, else let $S_{k+1} = S_k \cup (v_s = v_t)$, $k = k + 1$ and go to step 2.

The theorem below states the properties of the algorithm $unify_{ind}(S)$.

Theorem 3. *1. Algorithm $unify_{ind}(S)$ always stops.*

2. If $(bool, \sigma)$ is an output of $unify_{ind}(S)$ and $bool = false$ then system S has no consistent solutions.

3. If $(bool, \sigma)$ is an output of $unify_{ind}(S)$ and $bool = true$ then σ is an idempotent most general $AVar(S)$-consistent $AVar(S)$-conservative unifier of S.

Proof. **1.** No one pair of variables can be chosen more than once during step 3, and the set $AVar(S)$ is finite.

2. Suppose there is a consistent unifier θ of the system S. θ is obviously a unifier of S_0. We assume θ is a unifier of S_k for some k so S_k is unifiable and σ_k is its idempotent m.g.u. There must exist a pair $v_s, v_t \in AVar(S)$ for which $s\sigma_k = t\sigma_k$ (otherwise the algorithm stops with $bool = true$). We have $s\theta = s\sigma_k\theta = t\sigma_k\theta = t\theta$ which implies $v_s\theta = v_t\theta$ and θ is a unifier of the system S_{k+1}. By induction the systems S_k for any k are unifiable and the algorithm cannot stop with $bool = false$.

3. Let S_n be the last system being unified by the algorithm. Note that $AVar(S_k) = AVar(S)$ for any k. As $\sigma = \sigma_n$ is the output of $unify(S_n)$ it is an idempotent $AVar(S)$-conservative m.g.u. of S_n. σ is a unifier of S because $S \subseteq S_n$. Step 3 of the algorithm ensures that σ is $AVar(S)$-consistent. We shall prove that σ is a most general $AVar(S)$-consistent unifier of S.

Let θ be some $AVar(S)$-consistent unifier of $S = S_0$. The argument similar to that of part 2 of this proof leads us to the fact that θ is a unifier of S_k for any k, particularly S_n. As σ is an idempotent m.g.u. of S_n we have $\sigma\theta = \theta$ and so σ is a most general $AVar(S)$-consistent unifier of S. □

Thus algorithm $unify_{ind}(S)$ obtains a finite substitution satisfying conditions of theorem 2.

The effective realizations of the algorithm $unify_{ind}(S)$ exploits a representation of terms using so-called dags (directed acyclic graphs) as in [5]. The algorithm has linear space complexity and cubic time complexity in the size of the dag representing the system of equations. The size of this dag is bound by the total length l the system S, which gives the time complexity $O(l^3)$.

References

1. J.A. Robinson. A machine oriented logic based on the resolution principle. *J. of the ACM*, 12(1), 1965, p. 23-41
2. D.E. Knuth, P.B. Bendix. Simple word problems in universal algebras. In J.Leech, editor, *Computational problems in Abstract Algebra*, Pergamon Press, Oxford, 1970
3. F. Baader and J.H. Siekmann. Unification Theory. In: *Handbook of Logic in Artificial Intelligence and Logic Programming* D.M. Gabbay, C.J. Hogger, and J.A. Robinson (Ed.), Oxford University Press, 1994
4. A. Martelli, U. Montanari. An efficient unification algorithm. *ACM Transactions on Programming Languages and Systems* 4, 258-282, 1982.
5. J. Corbin, M. Bidoit. A rehabilitation of Robinson's unification algorithm. *Information Processing* 83, R.E.A. Mason (ed.), Elsevier Science Publishers (North-Holland), p. 909–914, 1983
6. S. Artemov, V. Krupski. Referential data structures and labeled modal logic. *Lecture Notes in Computer Science*, 1994, v.13, p. 23-33
7. S. Artёmov, V. Krupski. Data storage interpretation of labeled modal logic. *Annals of Pure and Applied Logic*, v.78 (1996), pp. 57-71.