Theoretical Computer Science 13 (1981) 225–230

NOTE

THE UNDECIDABILITY OF THE SECOND-ORDER UNIFICATION PROBLEM

Warren D. GOLDFARB
Department of Philosophy, Harvard University, Cambridge, MA 02138, U.S.A.

Communicated by A. Meyer
Received October 1979

Abstract. It is shown that there is no effective procedure for determining whether or not two terms of the language of second-order logic have a common instance.

1. Introduction

The unification problem for a formal language is the problem of determining whether any two formulas of the language possess a common instance. The problem for first-order languages has long been known to be decidable [4], and efficient algorithms for finding common instances have been devised (see [3]). Algorithms for first-order unification underlie resolution methods for automatic theorem proving. On the other hand, for third-order languages the problem is undecidable [1]. In this paper we show the unification problem for second-order languages undecidable, by reducing Hilbert's Tenth Problem to it.

We shall consider a simple second-order language L, whose formulas are terms that may contain both individual and function variables. The unification problem for L differs from that for first-order languages in that, to obtain instances of a term of L, function variables as well as individual variables may be instantiated.

More precisely, language L contains an infinite supply of individual variables, an infinite supply of n-place function variables for each $n>0$, and some number of individual and function constants. For the moment, we require L to contain at least two individual constants a and b and one 2-place function constant g. (This requirement may be weakened; see Section 3.) The *terms* of L are defined inductively thus: any individual constant or variable is a term; if F is an n-place function constant or variable and $e_1, \ldots, e_n$ are terms, $n>0$, then $F(e_1, \ldots, e_n)$ is a term.

All terms of L represent individuals. To specify the notion of instance of a term, we also need expressions that represent functions. Hence we consider an expanded language L^*. Let $w_1, w_2, \ldots$ be signs foreign to L. Language L^* differs from language L by having $w_1, w_2, \ldots$ as additional individual variables. Terms of L^* are constructed from the individual variables and constants of L^* just as for L. The *degree* of a term t of L^* is the largest m such that w_m occurs in t ($=0$ if t is a term of L). Below we shall use 'term' for 'term of L^*' and 'proper term' for 'term of L'.

Intuitively, we may take a term t of degree $\leq n$ to represent an n-place function: at arguments represented by proper terms $d_1, \ldots, d_n$ the value of this function is the individual represented by the proper term obtained from t by replacing $w_1, \ldots, w_n$ with $d_1, \ldots, d_n$, respectively. (Thus t represents infinitely many functions, one n-place function for each n greater than or equal to its degree. If w_k does not occur in t, then the values of such a function do not depend on the kth argument.)

A *substitution* is a finite set $\{v_1|t_1, \ldots, v_n|t_n\}$ of pairs such that $v_1, \ldots, v_n$ are distinct variables of L^* and, for each $i \leq n$, if v_i is an individual variable, then t_i is a proper term, and if v_i is an m-place function variable, then t_i is a term of degree $\leq m$. The result $s\theta$ of applying a substitution $\theta = \{v_1|t_1, \ldots, v_n|t_n\}$ to a term s is defined thus:

(1) if s is an individual variable and $s = v_i$ for some $i \leq n$, then $s\theta = t_i$;

(2) if s is an individual constant or an individual variable not among $v_1, \ldots, v_n$, then $s\theta = s$;

(3) if $s = F(s_1, \ldots, s_m)$, where F is a function constant or a function variable not among $v_1, \ldots, v_n$, then $s\theta = F(s_1\theta, \ldots, s_m\theta)$;

(4) if $s = F(s_1, \ldots, s_m)$, where F is a function variable and $F = v_i$ for some $i \leq n$, then $s\theta = t_i\{w_1|s_1\theta, \ldots, w_m|s_m\theta\}$.

Note that if $v_1, \ldots, v_n$ are all individual variables, then, for every term s, $s\theta$ is the result obtained from s by simultaneous replacement of $v_1, \ldots, v_n$ with $t_1, \ldots, t_n$, respectively. Note too that if s is a proper term, then so is $s\theta$ for every substitution θ.

Thus an instance of a proper term e is simply any term $e\theta$ for some substitution θ. A substitution θ is a *unifier* for a pair $\langle d, e\rangle$ of proper terms iff $d\theta = e\theta$. The unification problem for L is the problem of determining, given any pair $\langle d, e\rangle$ of proper terms, whether there exists a unifier for $\langle d, e\rangle$.

Remark. We take the unification problem to concern proper terms only. Sometimes, however, it is formulated to include expressions representing higher-order objects as well. Although this broader formulation allows a somewhat simpler undecidability proof, it is not relevant to the usual systems of second-order logic. For in these systems, expressions representing second-order objects never occur alone: they always occur with their argument places filled in. Hence if we seek to apply resolution procedures, it is always proper terms whose common instances are at issue.

2. Undecidability proof

A substitution θ is a unifier for a *set* of pairs of proper terms if and only if it is a unifier for each pair of terms in the set. We start by reducing the unification problem for finite sets of pairs to the problem for single pairs.

For all terms $t_1, \ldots, t_n$, $n > 0$, define a term $[t_1, \ldots, t_n]$ thus: $[t] = t$ for each t; $[t_1, \ldots, t_{n+1}] = g(t_1, [t_2, \ldots, t_{n+1}])$. Clearly $[s_1, \ldots, s_n] = [t_1, \ldots, t_n]$ if and only if $s_i = t_i$ for each $i \leq n$. Hence any unifier for a set $\{\langle d_1, e_1\rangle, \ldots, \langle d_n, e_n\rangle\}$ of pairs of proper terms is a unifier for the pair $\langle [d_1, \ldots, d_n], [e_1, \ldots, e_n]\rangle$, and conversely. Thus it suffices to show the undecidability of the unification problem for finite sets of pairs of proper terms.

Note that $[t_1, t_2]$ is just $g(t_1, t_2)$. For the sake of perspicuity, we shall use the former notation below rather than the latter. Also note that $[t_1, \ldots, t_k, [t_{k+1}, t_{k+2}]] = [t_1, \ldots, t_k, t_{k+1}, t_{k+2}]$.

For each $n \geq 0$ and each term t, let $\bar{n}t$ be the term defined inductively thus: $\bar{0}t = t$; $\overline{n+1}t = [a, \bar{n}t]$. Equivalently, $\bar{n}t = [a, \ldots, a, t]$, with n occurrences of a. Hence $\bar{n}t = \bar{m}t$ iff $n = m$.

We can easily construct a pair of proper terms any unifier for which 'simulates' addition. Let F_1, F_2, F_3 be 1-place function variables, and let $\theta = \{F_1 | \bar{n}w_1, F_2 | \bar{m}w_1, F_3 | \bar{p}w_1\}$ for $m, n, p \geq 0$. Then clearly θ is a unifier for the pair $\langle F_1(F_2(a)), F_3(a)\rangle$ if and only if $p = m + n$. The heart of our proof is the construction of a set of pairs of terms any unifier for which, in an analogous sense, simulates multiplication.

Let F_1, F_2, F_3 be 1-place function variables and let G be a 3-place function variable. Then let

$$d_1 = G(a, b, [[F_3(a), F_2(b)], a]), \qquad e_1 = [[a, b], G(F_1(a), \bar{1}b, a)],$$

$$d_2 = G(b, a, [[F_3(b), F_2(a)], a]), \qquad e_2 = [[b, a], G(F_1(b), \bar{1}a, a)].$$

Lemma. *For all $m, n, p \geq 0$ there is a unifier θ for $\{\langle d_1, e_1\rangle, \langle d_2, e_2\rangle\}$ containing the pairs $F_1 | \bar{m}w_1$, $F_2 | \bar{n}w_1$, and $F_3 | \bar{p}w_1$ if and only if $p = m \cdot n$.*

Proof. Let $m, n, p \geq 0$. Define four substitutions thus:

$$\sigma_1 = \{w_1 | a, w_2 | b, w_3 | [[\bar{p}a, \bar{n}b], a]\}, \qquad \tau_1 = \{w_1 | \bar{m}a, w_2 | \bar{1}b, w_3 | a\},$$

$$\sigma_2 = \{w_1 | b, w_2 | a, w_3 | [[\bar{p}b, \bar{n}a], a]\}, \qquad \tau_2 = \{w_1 | \bar{m}b, w_2 | \bar{1}a, w_3 | a\}.$$

If, for some term u, a substitution θ contains $F_1 | \bar{m}w_1, F_2 | \bar{n}w_1, F_3 | \bar{p}w_1$ and $G | u$, then $d_1\theta = u\sigma_1$, $d_2\theta = u\sigma_2$, $e_1\theta = [[a, b], u\tau_1]$ and $e_2\theta = [[b, a], u\tau_2]$. For each $k \geq 0$ let $t_k = [\overline{m \cdot k}w_1, \bar{k}w_2]$. Note that

$$t_k\tau_1 = [\overline{m \cdot (k+1)}a, \overline{k+1}b] = t_{k+1}\sigma_1 \quad \text{and} \quad t_k\tau_2 = [\overline{m \cdot (k+1)}b, \overline{k+1}a] = t_{k+1}\sigma_2.$$

(a) 'If'. Let $p = m \cdot n$ and let $\theta = \{F_1 | \bar{m}w_1, F_2 | \bar{n}w_1, F_3 | \bar{p}w_1, G | u\}$, where $u = w_3$ if $n = 0$ and $u = [t_0, \ldots, t_{n-1}, w_3]$ if $n > 0$.

If $n = 0$, then $d_1\theta = u\sigma_1 = [[\bar{p}a, \bar{n}b], a] = [[a, b], a] = [[a, b], u\tau_1] = e_1\theta$. Similarly, $d_2\theta = [[b, a], a] = e_2\theta$.

If $n > 0$, then $d_1\theta = u\sigma_1 = [t_0\sigma_1, \ldots, t_{n-1}\sigma_1, [[\bar{p}a, \bar{n}b], a]]$. Since $p = m \cdot n$, $[\bar{p}a, \bar{n}b] = t_n\sigma_1$. Hence $d_1\theta = [t_0\sigma_1, \ldots, t_{n-1}\sigma_1, t_n\sigma_1, a]$. Now $u\tau_1 = [t_0\tau_1, \ldots, t_{n-1}\tau_1, a] = [t_1\sigma_1, \ldots, t_n\sigma_1, a]$. Hence $e_1\theta = [[a, b], u\tau_1] = [t_0\sigma_1, [t_1\sigma_1, \ldots, t_n\sigma_1, a]] = [t_0\sigma_1, \ldots, t_n\sigma_1, a]$, so that $e_1\theta = d_1\theta$. Similarly, $e_2\theta = [t_0\sigma_2, \ldots, t_n\sigma_2, a] = d_2\theta$.

Thus θ is a unifier for $\{\langle d_1, e_1\rangle, \langle d_2, e_2\rangle\}$.

(b) 'Only if'. Suppose θ is a unifier for $\{\langle d_1, e_1\rangle, \langle d_2, e_2\rangle\}$ such that $\{F_1 | \bar{m}w_1, F_2 | \bar{n}w_1, F_3 | \bar{p}w_1\} \subseteq \theta$. This unifier θ must also contain $G | u$ for some term u. And then

(1) $u\sigma_1 = d_1\theta = e_1\theta = [[a, b], u\tau_1]$;

(2) $u\sigma_2 = d_2\theta = e_2\theta = [[b, a], u\tau_2]$.

Consequently, either $u = w_3$ or else $u = [r, r']$ for some terms r and r'.

Suppose $u = w_3$. Then $u\sigma_1 = [[\bar{p}a, \bar{n}b], a]$ and $u\tau_1 = a$. By (1), $[[\bar{p}a, \bar{n}b], a] = [[a, b], a]$, whence $n = 0$ and $p = 0$. Hence $p = m \cdot n$ and we are done.

Suppose $u = [r, r']$ for some r and r'. Indeed, let k be the largest integer such that $u = [s_0, \ldots, s_k]$ for some terms $s_0, \ldots, s_k$ $(k > 0)$. By (1), $[s_0\sigma_1, \ldots, s_k\sigma_1] = [[a, b], s_0\tau_1, \ldots, s_k\tau_1]$. By (2), $[s_0\sigma_2, \ldots, s_k\sigma_2] = [[b, a], s_0\tau_2, \ldots, s_k\tau_2]$. Thus

(3) $s_0\sigma_1 = [a, b]$ and $s_0\sigma_2 = [b, a]$;

(4) for $0 < j < k$, $s_j\sigma_1 = s_{j-1}\tau_1$ and $s_j\sigma_2 = s_{j-1}\tau_2$;

(5) $s_k\sigma_1 = [s_{k-1}\tau_1, s_k\tau_1]$ and $s_k\sigma_2 = [s_{k-1}\tau_2, s_k\tau_2]$.

By (3), $s_0 = [w_1, w_2]$, that is, $s_0 = t_0$. By (4), then, $s_1\sigma_1 = t_0\tau_1 = t_1\sigma_1$ and $s_1\sigma_2 = t_0\tau_2 = t_1\sigma_2$. Hence $s_1 = t_1$. Applying (4) repeatedly, we infer $s_2 = t_2, \ldots, s_{k-1} = t_{k-1}$. By (5), $s_k\sigma_1 = [t_{k-1}\tau_1, s_k\tau_1] = [t_k\sigma_1, s_k\tau_1]$, whence either $s_k = w_3$ or else $s_k = [s, s']$ for some terms s and s'. But in the latter case $u = [s_1, \ldots, s_{k-1}, [s, s']] = [s_1, \ldots, s_{k-1}, s, s']$, contrary to the choice of k. Hence $s_k = w_3$. And then $s_k\sigma_1 = [[\bar{p}a, \bar{n}b], a] = [t_k\sigma_1, s_k\tau_1] = [[\overline{m \cdot k}a, \bar{k}b], a]$. Thus $k = n$ and $p = m \cdot k = m \cdot n$.

Theorem. *There is an effective method that reduces Hilbert's Tenth Problem to the unification problem for L.*

Proof. Let H be any finite set of equations having the forms $X_i \cdot X_j = X_k$, $X_i + X_j = X_k$, and $X_i = C_j$, where the X's are numerical variables and the C's numerical constants. A solution for H is an assignment of nonnegative integers to the numerical variables that makes all the equations in H true. It suffices to construct a set S of pairs of proper terms such that there is a unifier for S if and only if H has a solution.

Suppose $X_1, \ldots, X_q$ are all the numerical variables in H. The terms in S will contain the 1-place function variables $F_1, \ldots, F_q$ and various 3-place function variables G_l. Let S contain the following pairs:

(1) for each i, $1 \leq i \leq q$, the pair $\langle \bar{1}F_i(a), F_i(\bar{1}a)\rangle$;

(2) for all i and j such that $X_i = C_j$ is a member of H, the pair $\langle F_i(a), \bar{c}_j a\rangle$, where c_j is the numerical value of C_j;

(3) for all i, j, k such that $X_i + X_j = X_k$ is a member of H, the pair $\langle F_i(F_j(a)), F_k(a)\rangle$;

(4) for all i, j, k such that $X_i \cdot X_j = X_k$ is a member of H, the two pairs obtained from the pairs $\langle d_1, e_1\rangle$ and $\langle d_2, e_2\rangle$ given above by relettering the function variables thus: F_1 is relettered F_i, F_2 is relettered F_j, F_3 is relettered F_k, and G is relettered G_l for $l = 2^i 3^j 5^k$.

Let θ be a unifier for S. By (1) there are $n_1, \ldots, n_q$ such that $F_i \mid \bar{n}_i w_1 \in \theta$, $1 \leq i \leq q$. We claim that $n_1, \ldots, n_q$ are a solution for H. For by (2) if $X_i = C_j$ is in H, then $\bar{n}_i a = F_i(a)\theta = \bar{c}_j a$, so that $n_i = c_j$; by (3) if $X_i + X_j = X_k$ is in H, then $\overline{n_i + n_j} a = F_i(F_j(a))\theta = F_k(a)\theta = \bar{n}_k a$, so that $n_i + n_j = n_k$; and by (4) and the Lemma, if $X_i \cdot X_j = X_k$ is in H then, since $\{F_i \mid \bar{n}_i w_1, F_j \mid \bar{n}_j w_1, F_k \mid \bar{n}_k w_1\} \subseteq \theta$, $n_k = n_i \cdot n_j$.

Conversely, suppose the assignment of $n_1, \ldots, n_q$ to $X_1, \ldots, X_q$ is a solution for H. Let θ contain $F_i \mid \bar{n}_i w_1$ for each i, $1 \leq i \leq q$. Then θ is a unifier for each pair of proper terms specified in (1)–(3). Now suppose $X_i \cdot X_j = X_k$ is in H, so that $n_i \cdot n_j = n_k$. By the Lemma there is a term u such that if θ contains $G_l \mid u$ as well (where $l = 2^i 3^j 5^k$), then θ is a unifier for the two pairs specified in (4).

3. Further remarks

(i) We required language L to contain two individual constants a and b, and one 2-place function constant g. This requirement may be weakened: we do not in fact need the individual constants. For let x be an individual variable of L. We may replace all occurrences of a and b in the terms used in Section 2 by occurrences of $[[x, x], x]$ and $[x, [x, x]]$, respectively; the proofs still are valid. We cannot, however, dispense with function constants. Indeed, if L contains no function constants, then the unification problem is trivially decidable. For suppose d and e are terms such that $d = F(d_1, \ldots, d_n)$ and $e = G(e_1, \ldots, e_m)$, where F and G are function variables; let u be any term of degree 0, and let $\theta = \{F|u, G|u\}$. Then θ is a unifier for $\langle d, e\rangle$.

(ii) In [2] Parikh considered the problem of determining, given any $k > 0$ and any formula F of the standard formulation of Peano Arithmetic, whether F has a proof in Peano Arithmetic containing at most k lines. He showed this problem reducible to the following form of the second-order unification problem. The second-order language L contains these constants: as individual constants the variables and zero sign of Peano Arithmetic; as a 1-place function constant the successor sign of Peano Arithmetic; and as 2-place function constants the addition and multiplication signs of Peano Arithmetic. The problem is then to determine, given any pair $\langle d, e\rangle$ of terms of L, whether there exists a unifier θ for $\langle d, e\rangle$ such that $d\theta$ contains no variables of L. Clearly the proof in Section 2 above establishes the undecidability of this problem. Hence Parikh's reduction does not settle the status of the k-line provability problem for Peano Arithmetic. Indeed, the decidability of that problem remains open.

Acknowledgment

The author would like to thank Harry Lewis for his great help in the preparation of this paper.

References

[1] G.P. Huet, The undecidability of unification in third order logic, *Information and Control* **22** (1973) 257–267.

[2] R. Parikh, Some results on the length of proofs, *Trans. Amer. Math. Soc.* **177** (1973) 29–36.

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

[4] J.A. Robinson, A machine-oriented logic based on the resolution principle, *J. ACM* **12** (1965) 23–41.