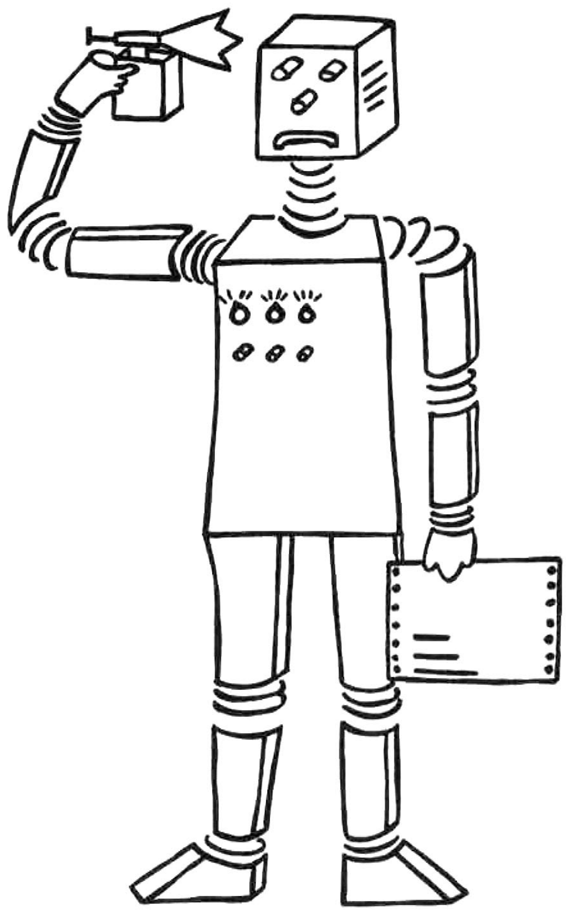
SEKI-REPORT

Artificial
Intelligence
Laboratories

Fachbereich Informatik
Universität Kaiserslautern
Postfach 3049
D-6750 Kaiserslautern 1, W. Germany



Unification in Permutative Equational Theories is Undecidable

Manfred Schmidt-Schauß
SEKI Report SR-87-3

UNIFICATION IN PERMUTATIVE EQUATIONAL THEORIES IS UNDECIDABLE

Manfred Schmidt-Schauß

Mai 1987 SEKI-REPORT SR-87-03

Unification in Permutative Equational Theories is Undecidable.

Manfred Schmidt-Schauß,
Universität Kaiserslautern, Fachbereich Informatik,
Postfach 3049
6075 Kaiserslautern, F.R.G.

Abstract. An equational theory E is permutative if in every valid equation $s =_E t$ the terms s and t have the same symbols with the same number of occurrences. The class of permutative equational theories includes associativity and commutativity and hence is important for unification theory, for term rewriting systems modulo equational theories and corresponding completion procedures.
It is shown in this research note that there is no algorithm that decides E-unifiability of terms for all permutative theories.
The proof technique is to provide for every Turing machine M a permutative theory with a confluent term rewriting system such that narrowing on certain terms simulates the Turing machine M .

Introduction.

We assume the reader to be familiar with the basic notions and definitions of terms, equational theories, unification with respect to an equational theory, term rewriting systems and narrowing. As reference we refer to [KB70, HO80,Hu80, Si86 ,BHS87]
Permutative equational theories are a special class of equational theories, first introduced in [LB77] in order to generalize term rewriting system to term rewriting system modulo some equational theory (see also [JK84]). A permutative theory E is defined as an equational theory, where for every valid equation $s =_E t$, the number of occurrences of every symbol in s is the same as in t. Well-known examples are commutativity (C), associativity (A) and their combination AC. It is easy to see that it is decidable whether a theory is permutative by inspecting the axioms in one of its presentations. Permutative theories have some nice properties: Every equivalence class $[s]_{=E}$ with respect to $=_E$ is finite, the word-problem with respect to $=_E$ is decidable, E-matching is decidable, sets of E-matchers are finite and effectively computable and minimal unifier sets always exist (cf. [Sz82, Si86, BHS87]).
A generalization of permutative theories are the finite theories, in which every $=_E$-equivalence class is finite. It is known that finiteness of a theory is undecidable [NO85] and that there exist finite theories with an undecidable unification problem. For example such a theory is DA (2-sided distributivity and associativity with the axioms: f(x g(y z)) = g(f(x y) f(x z)) ; f(g(x y) z)) = g(f(x z) f(y z)) ; g(x g(y z)) = g(g(x y) z) [Sz82, Si86]. However, DA is not permutative, since the axiom f(x g(y z)) = g(f(x y) f(x z)) is not permutative. It was conjectured by Jouannaud [Ki87] that the more special class of theories which are generated by variable-permutative axioms (i.e. in every axiom l = r the terms l and r have the same term-structure, but the variables may be permuted) has a decidable unification problem. An interesting example of this class is AC. It should be noted that variable-permutativity is in general not inherited to valid equations.

In this paper we solve the open question [Ki87] whether or not unification in permutative theories is decidable.

Reduction of the Halting Problem to the Unification problem in Permutative Theories.

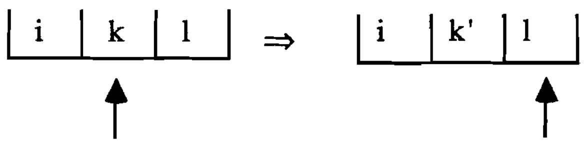
We proceed as follows:
For a given Turing machine M we define a set of rewrite rules for every line of the Turing machine program. The resulting term rewriting system R_M is shown to be confluent, hence we can use narrowing as unification procedure. The equational theory described by this rewrite rules is permutative. Then we give two terms s,t such that the narrowing process simulates the Turing machine M starting on blank tape.

We assume that the Turing machine uses as symbols from athe finite alphabet $A = \{1,\ldots,n\}$ and B (blank) and that blanks cannot be printed on the tape. Furthermore we can assume that it has a starting state q_{ST} and only one accepting state q_{ACC}.

The transformation of M into a term rewriting system R_M is as follows:
We use a signature consisting of two constants a and b , a ternary function symbol f, a n+1-ary function symbol g and a function symbol h that has as many argument as M has states. The function symbol f is used for the instantaneous description of the Turing machine, g for the description of the tape, where the first n argument positions are reserved for and h for encoding the state. The instantaneous description $\alpha_1 q \alpha_2$ of M is represented as $f(q\ \alpha_1\ \alpha_2)$. Here q is the current state, α_1 is the tape content to the left of the head and α_2 is the tape content to the right of the head including the currently scanned symbol. The state q is encoded as a term h(a,...,a,b,a,...), where b is at the q´th argument position. The tape content is encoded with the function symbol g as follows: a blank B is represented as g(b a a ...), 1 as g(a b a ...), 2 as g(a a b ...) and so forth. Hence a string 112B is represented as g(a b a ...g(a b a ...g(a a b ... g(b a a ...) ...) ...) ...). We abbreviate g(b a a ...) and g(a ...a b a ...) where b is at the k+1st position as $g_B(\ldots)$ and $g_k(\ldots)$, respectively.

We explain the translation of the lines of the Turing machine program into a term rewriting system by examining all important cases. Besides the starting state, there are essentially 4 different possibilities: either the head moves right or left in some situation and either the head reads B or not.
We describe the corresponding encodings:

i) Suppose in state q_1, if the head reads a symbol $k \in A$, a symbol k' is printed, the new state is q_2 and the head moves right.
   This is encoded in rewrite rules of the form:
   $$f(q_1, g_i(x), g_k(g_\ell(y))) \rightarrow f(q_2, g_{k'}(g_i(x)), g_\ell(y)).$$
   Here the symbols i and ℓ range over symbols from A and B.

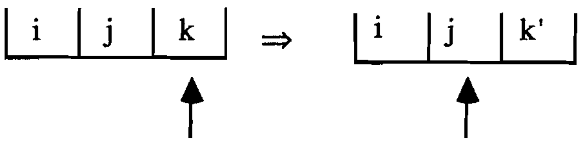


ii) Suppose in state q_1, if the head reads a symbol $k \in \{0, 1\}$, a symbol k' is printed, the new state is q_2 and the head moves left.

This is encoded in rewrite rules of the form:

$f(q_1, g_j(g_i(x)),\ g_k(y),) \to f(q_2, g_i(x),\ g_j(g_{k'}(y)))$.

The symbols i and j range over the symbols from A and B and if i is B, then j is also B.



iii) Suppose in state q_1, if the head reads a B, a symbol k is printed, the new state is q_2 and the head moves right.

This is encoded in rewrite rules as follows:

$f(q_1, g_i(x), g_B(g_B(y))) \to f(q_2, g_k(g_i(x)), g_B(y))$.

$f(q_1, g_B(x), g_B(g_i(y))) \to f(q_2, g_k(g_B(x)), g_i(y))$.

The symbol i ranges over the symbols from A.

In the first rule, the head is at the right end of the marked tape and in the second rule, the head is at the left end.

iv) Suppose in state q_1, if the head reads a B, a symbol k is printed, the new state is q_2 and the head moves left.

This is encoded in rewrite rules as follows:

$f(q_1, g_B(g_B(x)), g_B(y)) \to f(q_2, g_B(x), g_B(g_k(y)))$.

$f(q_1, g_i(g_j(x)), g_B(y)) \to f(q_2, g_i(x), g_j(g_k(y)))$.

The symbols i and j range over the symbols from A.

In the first rule the head is at the left end of the marked tape and in the second rule,the head is at the right end. ❑

The above encoding needs at most finitely many rewrite rules to encode one line of the Turing machine program. We have omitted the nonsensible combinations, which correspond to the case that a symbol from A is outside the marked tape or that a B is inside the marked tape. The omission of these redundant rules is necessary for the correct behaviour of the narrowing process, as we will see later.

Let $\mathcal{E}_M$ be the equational theory generated by the term rewriting system R_M; then we have:

1. Lemma.

i) $\mathcal{E}_M$ is permutative.

ii) R_M is confluent.

Proof. i) For the encoding above: if we count the number of symbols, we have the same number for all symbols on the left and right hand side for every rewrite rule.

ii) The term rewriting system thus constructed is left-linear and there are no critical pairs. It is easy to see that R_M is strongly confluent, and hence it is confluent (cf. Lemma 2.5 in [Hu80]). ■

Note that the term rewriting system R_M may be nonterminating. However, since R_M is confluent, the normalform of a term t is unique.
We say a substitution is **normal**, iff all terms in its codomain are in normalform.

The idea is to use narrowing [Hl80, Fay79] now for the unification procedure, since R_M is confluent. The following theorem follows from results and proofs in [Hl80] and is a specialization of the completeness result in [Hus85]:

2. Theorem. Let s, t be two terms and let θ be a normal $\mathcal{E}_M$-unifier of s and t. Then narrowing provides a $\mathcal{E}_M$-unifier σ that is more general than θ on V(s,t). ■

We use the following unification problem to simulate the Turing machine:
$\langle s = t \rangle_{\mathcal{E}M}$ where $s = f(q_{ST}, g_B(x), g_B(y))$ and $t = f(q_{ACC}, z_1, z_2)$. By the structure of the equational theory we see that for every $\mathcal{E}_M$-unifier of s and t there exists a more general one that does not contain f's in the codomain terms and hence there exists a normal $\mathcal{E}_M$-unifier of s and t, iff s and t are $\mathcal{E}_M$-unifiable.
Narrowing on t is not possible, since there does not exist a rewrite rule with q_{ACC} at the first argument position. Hence narrowing is only performed on the term s and its descendants. Obviously narrowing takes place always at the toplevel occurrence of s. A condition that the narrowing process succeeds is that it reaches the state q_{ACC}, since otherwise it is not possible to Robinson-unify the obtained term with t. On the other hand if it reaches the state q_{ACC}, then it terminates and gives a $\mathcal{E}_M$-unifier. During the narrowing, the descendants of the term s represent the tape content as a string $Bi_1i_2\dots i_nB$ or $BBi_1i_2\dots i_nB$, where $i_j \in A$. This and the structure of the rewrite rules implies that there is always at most one narrowing step possible. If the head is inside the string, i.e., it scannes i_j where $j = 2,\dots,n$, then narrowing is just rewriting. If the head scannes the borderline of the string representation, i.e. the descendant of s is of the form $f(q, g_B(x), \dots)$ or $f(q, \dots, g_B(x))$ then it may be possible that new blanks are added to the end of the string by the narrowing substitution. There are only two possibilities:

i) either a rewrite step is performed or

ii) the narrowing-substitution is of the form $\{z \leftarrow g_B(z')\}$.

Since narrowing on the $\mathcal{E}_M$-unification-problem $\langle s,t \rangle$ simulates the Turing machine M starting on blank tape, we have the following Proposition:

3. Proposition. The terms s and t are $\mathcal{E}_M$-unifiable, iff M accepts blank tape. ■

Since it is well-known that it is undecidable whether a Turing machine accepts blank tape, we have our final result:

4 Theorem . There is no algorithm that decides the unifiability of terms in permutative theories. In other words unification in permutative theories is undecidable. ■

We have also the stronger result:

5 Theorem . There exists a permutative theory $\mathcal{E}_M$, such that there is no algorithm that decides the unifiability of terms in $\mathcal{E}_M$.

Proof. For M we use a universal Turing machine and as unification problem we choose $\langle s = t \rangle_{\mathcal{E}M}$ where $s = f(q_{ST}, \alpha_1, \alpha_2)$ and $t = f(q_{ACC}, z_1, z_2)$, where the term s represents the tape content at the start. The same arguments as above apply to this configuration. ■

References:

[Bi 35] Birkhoff, G.: 'On the Structure of Abstract Algebra', Proc. Cambrigde Phil. Soc., Vol. 31, p. 433-454, (1935)

[BHS87] Bürckert, H.-J, Herold, A., Schmidt-Schauß, M., On Equational Theories, Unification and Decidability, (to appear in Proc. RTA 87, Bordeaux), (1987)

[Fay79] Fay, M., First order unification in an equational theory, Proc. 4th Workshop on Automated Deduction, W.H.Joyner (ed), Academic Press, (1979)

[Hl 80] Hullot, J.M.: 'Canonical Forms and Unification', in Proc. of 5th CADE (eds. W. Bibel and R. Kowalski), Springer, LNCS 87, p. 318-334, (1989)

[HO 80] Huet, G. and Oppen, D. C.: 'Equations and Rewrite Rules: A Survey', in 'Formal Languages: Perspectives and Open Problems' (ed R. Book), Academic Press, (1980)

[HoUl79] Hopcraft. J.E., Ullmann, J.D., Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, (1979)

[Hu80] Huet, G., Confluent reductions: Abstract Properties and Applications to Term Rewriting Systems. JACM 27, 4, pp. 797-821, (1980)

[Hus85] Hussmann, H., Unification in Conditional-Equational Theories, in Proc. of EUROCAL '85 (ed. B.F. Caviness), LNCS 204, (1985)

[JK84] Jouannaud, J.P., Kirchner, H., Completion of a set of rules modulo a set of equations. Technical report SRI International, (1984)
see also Proc. of an NSF Workshop on the Rewrite Rule Laboratory, General Electric, Schnectady, (ed. by J.V. Guttag, D. Kapur, D.R. Musser), Rep no. 84GEN008, pp. 207-228, (1983)

[Ki87] Kirchner. C. (ed), First Workshop on Unification Val d´Ajol, C.N.R.S Internal Report n° 87 R34, University of Nancy, (1987)

[LB 77] Lankford, D.S. and Ballantyne, A.M.: 'Decision Procedures for Simple Equational Theories with Permutative Axioms: Complete Sets of Reductions', Report ATP-37, University of Texas, Austin, (1977)

[KB 70] Knuth, D.E. and Bendix, P.B.: 'Simple Word Problems in Universal Algebras', in 'Computational Problems in Abstract Algebras' (ed. J. Leech), Pergamon Press, p. 263-297, (1970)

[Ki 85] Kirchner, C.: 'Methodes et outils de conception systematique d'algorithmes d'unification dans les théories équationelles', Thèse de doctorat d'état, (in French) Université de Nancy 1, (1985)

[NO 85] Narendran, P.: O'Dúnlaing, C. and Rolletschek, H., 'Complexity of certain desision problems about congruential languages', Journal of Computer and System Sciences, Vol.30, p. 343-358, (1985)

[Ro 65] Robinson, J. A.: 'A Machine-Oriented Logic Based on the Resolution Principle', JACM 12, N$^{\underline{o}}$. 1, p. 23-41, (1965)

[Si 84] Siekmann, J.: 'Universal Unification', in Proc. of 7th CADE (ed R.E. Shostak), Springer, LNCS 170, p. 1-42, (1984)

[Si 86] Siekmann, J.: 'Unification Theory', in Proc. of ECAI'86, Vol. II, p. vi-xxxv, Brighton, (1986)

[Sz 82] Szabó, P.: Unifikationstheorie Erster Ordnung, Dissertation (in German), Universität Karlsruhe, (1982)

[Ta 79] Taylor, W.: 'Equational Logic', Houston Journal of Mathematics 5, (1979)