A criterion for intractability of E-unification with free function symbols and its relevance for combination of unification algorithms

Klaus U. Schulz*

CIS, Universität München, Oettingenstraße 67, 80538 München, Germany,
schulz@cis.uni-muenchen.de, http://www.cis.uni-muenchen.de/people/schulz.html

Abstract. All applications of equational unification in the area of term rewriting and theorem proving require algorithms for general E-unification, i.e., E-unification with free function symbols. On this background, the complexity of general E-unification algorithms has been investigated for a large number of equational theories. For most of the relevant cases, the problem of deciding solvability of general E-unification problems was found to be NP-hard. We offer a partial explanation. A criterion is given that characterizes a large class $\mathcal{K}$ of equational theories E where general E-unification is always NP-hard. We show that all regular equational theories E that contain a commutative or an associative function symbol belong to $\mathcal{K}$. Other examples of equational theories in $\mathcal{K}$ concern non-regular cases as well.
The combination algorithm described in [BS92] can be used to reduce solvability of general E-unification algorithms to solvability of E- and free (Robinson) unification problems with linear constant restrictions. We show that for $E \in \mathcal{K}$ there exists no polynomial optimization of this combination algorithm for deciding solvability of general E-unification problems, unless $P = NP$. This supports the conjecture that for $E \in K$ there is no polynomial algorithm for combining E-unification with constants with free unification.

1 Introduction

Equational unification is used as an important built-in procedure in many deductive systems in the area of term rewriting [JoK86, Ba91] and resolution-based theorem proving [Plo72, Sti85, NR94, Ru95]. For all these applications, unification algorithms are needed for solving problems that contain not only the symbols of the signature of the given equational theory E, but also additional "free" (uninterpreted) function symbols of arbitrary arity. Following common terminology, such problems are called general E-unification problems here.

On the background of this practical relevance it is natural to ask for the complexity of general E-unification. For most of the equational theories E that

* This work was supported by the EC Working Group CCL, EP6028.

have been discussed in the literature the problem of deciding solvability of general E-unification problems has turned out to be NP-hard (see [KN92, BS94] for surveys). When looking at the proofs of these intractability results it becomes clear that some of the encoding techniques that are used are closely related. Yet, a common background is missing that would explain these results from an abstract point of view. This yields one motivation for the results of the present paper, where we characterize general properties of equational theories E that guarantee that solvability of general E-unification problems is NP-hard. Before we explain the second motivation let us summarize the main results.

A criterion is given that characterizes a class $\mathcal{K}$ of equational theories where the problem of deciding solvability of general E-unification problems is always NP-hard. Using the criterion for $\mathcal{K}$ we shall prove that general E-unification is NP-hard for every regular equational theory E which contains an associative or a commutative function symbol. The theories A, C, AC and ACI (where A, C, and I stand for associativity, commutativity and idempotence of a fixed binary function symbol respectively) represent instances of such theories. If $E \in \mathcal{K}$ and there exists an NP-algorithm for E-unification with " linear constant restriction[2]", then general E-unification is NP-complete. The main criterion is not restricted to the case of regular theories. For the sake of exposition, and concentrating on theories where E-unification with constants is decidable in polynomial time, we show that the theories $ACUN$, $ACUNh$, and AG which have been discussed in recent papers [GN96, HK96] also belong to $\mathcal{K}$. In each case, the proof is extremely simple.

Let us now explain the second, even more serious motivation for our investigations, even if it is not simple to judge the full status of our results under this perspective. We have seen that applications of equational unification require algorithms for general E-unification. In many cases there is no obvious way to design such an algorithm directly. Instead it might be much simpler to give an algorithm for E-unification problems with free constants only.[3] Fortunately, there exists a general methodology for obtaining an algorithm for general E-unification that can be used in most of these cases. General E-unification can be considered as a combination of "elementary" E-unification with free (Robinson) unification. Hence it is one instance of the more general problem of combining unification algorithms for disjoint equational theories. The latter problem has been considered by many authors ([Ki85, He86, Ti86, Ye87, Sc89, Bo93, BS92]), general solutions have been given in [Sc89, Bo93, BS92]. On the basis of [BS92] it is possible to obtain an algorithm for general E-unification problems by combining a given algorithm for E-unification with linear constant restriction with an algorithm for free (Robinson) unification with linear constant restriction.

When we use this approach for general E-unification in a practical system, an efficiency problem arises. The combination algorithm of [BS92] decomposes a given general E-unification problem into a pair of pure output problems, where

[2] An inessential generalization of E-unification problems with constants, see Section 3.
[3] Compare, for example, the remarks on general A- and AC-unification in [BS92].

the two components represent E- and free unification problems with linear constant restriction respectively. This reduction is based on a polynomial number of non-deterministic steps, which means in practice that an exponential number of different output pairs has to be considered in the worst case.[4] The present research was started in a project where we tried to optimize the algorithm ([KR96]). We investigated how structural properties of the theory E (or, in the more general situation, of two given theories E and F to be combined) can be used to eliminate parts of the non-determinism of the combination scheme. In order to see the limitations for such optimizations in the context of general E-unification it was natural to ask for properties of E that imply intractability in the sense that there cannot be a polynomial version of the algorithm, assuming $P \neq NP$.

We shall show that for $E \in \mathcal{K}$ there exists no polynomial optimization of the algorithm that reduces solvability of general E-unification problems to solvability of E- and free unification problems with linear constant restriction respectively, unless $P = NP$. For this reason we strongly conjecture that for all equational theories $E \in \mathcal{K}$ there is no polynomial procedure for combining algorithms that decide solvability of E-unification problems with constants with similar decision procedures for free (syntactic) unification. Of course our results verify this conjecture—assuming $P \neq NP$—in all cases where the problem of deciding solvability of E-unification problems with constants is decidable in polynomial time, such as, e.g., for the theories $ACUN$, $ACUNh$ (see [GN96]), and AG.

From the combination perspective, the present research is closely related to recent work by M. Hermann and P.G. Kolaitis [HK96]. In their paper it was shown (assuming $P \neq NP$) that there cannot be a polynomial algorithm for combining unification algorithms for arbitrary (disjoint) finitary equational theories, by proving intractability of the counting problem for general unification in the theory of abelian groups (AG-unification). One major difference is that M. Hermann and P.G. Kolaitis consider unification algorithms that compute minimal and complete sets of E-unifiers, whereas we concentrate on algorithms that decide solvability of E-unification problems. Furthermore, M. Hermann and P.G. Kolaitis only treat unification in abelian groups and in boolean rings, whereas we are primarily interested in abstract properties of theories that lead to intractability.[5]

2 Preliminaries

A *signature* consists of a finite set of function symbols, each of fixed arity. Let Σ be a signature, and let Var denote a disjoint countably infinite set of variables. The set of Σ-terms with variables in Var is defined as usual. With $\mathcal{T}(\Sigma, Var)$ we denote the free term algebra for the signature Σ. A *Σ-substitution* is an endomorphism σ of $\mathcal{T}(\Sigma, Var)$ such that the set $\{x \in Var \mid \sigma(x) \neq x\}$ is finite.

[4] Other combination algorithms for unification algorithms have the same problem.

[5] M. Hermann (personal communication) pointed out that it is possible to generalize their results in the same direction.

Symbols $\sigma, \tau, \mu, \lambda$, possibly with subscripts, always denote substitutions. If σ and τ are substitutions, then $\sigma \circ \tau$ denotes their composition, where σ is applied first. If t is a term, then $Var(t)$ denotes the set of variables occurring in t.

A (representation of an) *equational theory* with signature Σ is a set E of equations between Σ-terms. With $=_E$ we denote the least congruence relation on $\mathcal{T}(\Sigma, Var)$ that is closed under substitutions and contains E, and $\mathcal{T}(\Sigma, Var)/=_E$ denotes the quotient term algebra modulo $=_E$. An equational theory E is called *consistent* if $x \neq_E y$ for distinct variables $x, y \in Var$. E is *regular* if $Var(s) = Var(t)$ for all equations $s = t$ of E. For a detailled explanation of these notions and for an introduction to equational unification we refer to [BS94].

Remark 1. The following fact can easily be proved for regular equational theories: $\forall s, t \in T(\Sigma, Var) : \ s =_E t$ implies $Var(s) = Var(t)$.

Let E be an equational theory with signature Σ. An *elementary E-unification problem* is a finite set γ of equations between Σ-terms. Sometimes we shall write γ as a conjunction of equations. An *E-unification problem with constants* is a finite set of equations between $(\Sigma \cup \Gamma)$-terms, where Γ is a set of "free" constants, i.e., a set of constants not occurring in Σ. A *general E-unification problem* is a finite set of equations between $(\Sigma \cup \Phi)$-terms, where Φ is a set of free function symbols of arbitrary arity. Note that each general E-unification problem can be considered as an elementary unification problem in the combined theory $E \cup F$ where F denotes the free (empty) theory over the set of functions symbols Φ.

Let γ be an elementary E-unification problem of the form $\{s_1 = t_1, \ldots, s_n = t_n\}$. A *solution* (or an *$E$-unifier*) of γ is a Σ-substitution σ such that $\sigma(s_i) =_E \sigma(t_i)$, for $i = 1, \ldots, n$. It should be clear that solutions of E-unification problems with constants, or solutions of general E-unification problems, may use the additional free symbols occurring in the problem itself.

3 Combination of unification algorithms

In this section we give a brief description of the combination procedure for unification algorithms for equational theories over disjoint signatures given in [BS92]. The proof of the central proposition of the following section heavily depends on the correctness of this combination algorithm. Before we give the algorithm, we have to introduce a generalization of E-unification problems with constants.

Let γ be an elementary E-unification problem, let Σ be the signature of E, and let Δ be another signature. Let Y be a finite set of variables such that $Var(\gamma) \subseteq Y$. A *linear constant restriction* for Y is a pair $L = (Lab, <)$ where $<$ is a strict linear ordering on Y and where $Lab : Y \to \{\Sigma, \Delta\}$ is a "labelling function" that assigns to each variable $y \in Y$ a signature $Lab(y) \in \{\Sigma, \Delta\}$. The pair (γ, L) is called an *E-unification problem with linear constant restriction*. A Σ-substitution σ *solves* (γ, L) if σ solves the E-unification problem γ and if the following conditions are satisfied:

1. $\sigma(y) = y$ for all $y \in Y$ such that $Lab(y) = \Delta$,
2. for all $x, y \in Y$: if $Lab(y) = \Delta, Lab(x) = \Sigma$ and if y occurs in $\sigma(x)$, then $y < x$.

Note that, by condition 1, the variables with alien label Δ are treated as free constants in (γ, L).

Let E and F be two consistent equational theories over disjoint signatures Σ and Δ respectively. An elementary $(E \cup F)$-unification problem γ is in *decomposed form* if γ has the form $\gamma_E \cup \gamma_F$ where the "pure" subproblems γ_E and γ_F are built over the signatures Σ and Δ respectively. Suppose that we want to decide solvability of an elementary $(E \cup F)$-unification problem γ_0. The following *Decomposition Algorithm*, described in more detail in [BS92], reduces γ_0 non-deterministically to a finite number of output pairs. Each component of an output pair represents an (E- resp. F-) unification problem with linear constant restriction.

Decomposition Algorithm. In the *first step*, the input problem γ_0 is transformed into an elementary $(E \cup F)$-unification problem γ_1 which is in decomposed form $\gamma_{1,E} \wedge \gamma_{1,F}$ such that γ_0 is solvable iff γ_1 is solvable. In the *second step*, a partition Π of $Var(\gamma_{1,E} \wedge \gamma_{1,F})$ is chosen, and for each equivalence class of Π a representant is chosen. Now all occurrences of variables are replaced by the representant of the equivalence class that contains the variable. We obtain the new formula $\gamma_{2,E} \wedge \gamma_{2,F}$. Let Y denote the set of representants. In the *third* and *fourth step*, a labelling function $Lab : Y \to \{\Sigma, \Delta\}$ and a strict linear ordering $<$ on Y are chosen. The output pair determined by the choices in steps $2-4$, then, is $((\gamma_{2,E}, L), (\gamma_{2,F}, L))$, where $L = (Lab, <)$. In the first (second) component, the variables with label Δ (resp. Σ) are treated as constants.

The first, deterministic step is based on the technique of "variable abstraction". Steps 2-4, then, are non-deterministic. Following common terminology, the second step will be called "variable identification" in this paper. The main technical result of [BS92] is the following

Proposition 2. *The input problem, γ_0, has a solution iff there exists an output pair of the Decomposition Algorithm, $((\gamma_{2,E}, L), (\gamma_{2,F}, L))$, such that both the E-unification problem with linear constant restriction $(\gamma_{2,E}, L)$ has a solution and the F-unification problem with linear constant restriction $(\gamma_{2,F}, L)$ has a solution.*

In the sequel, two details of the correctness proof for Proposition 2 given in [BS92] will be used.

Remark 3. It was shown ([BS92], p. 58) how given solutions σ_E and σ_F of an output pair of the Decomposition Algorithm can be combined to a solution σ of the input problem, γ_0. This combined solution σ has the following property: if y is a representant of type Δ, and if the term $\sigma_F(y)$ does not contain any variable with label Σ, then $\sigma(y) = \sigma_F(y)$.

Remark 4. It was described ([BS92], p. 60) how a given solution σ of an elementary $(E \cup F)$-problem can be used to define choices in the non-deterministic steps of the Decomposition Algorithm that lead to an output pair $((\gamma_{2,E}, L), (\gamma_{2,F}, L))$ where both components are solvable.[6] In the second step of this construction, two variables v_1 and v_2 of the decomposed problem are identified iff $\sigma(v_1) =_{E \cup F} \sigma(v_2)$.

4 Main Results

In the first subsection we shall give a criterion that can be used to show that for a given equational theory E the problem of deciding solvability of *general* E-unification problems is NP-hard. The power of the criterion will be illustrated in the second subsection. Eventually we comment on the consequences for attempts to optimize the combination algorithm given in [BS92] in the context of general E-unification.

4.1 A criterion for intractability

One notion will be needed before we can state the main technical result of this section.

Definition 5. Let γ be an E-unification problem. Let $\{x_0, \ldots, x_m\} \subseteq Var(\gamma)$ for some $m \geq 0$, let $\vec{x}$ denote the sequence $\langle x_0, \ldots, x_m \rangle$. A solution σ of γ is $\vec{x}$-*atomic* if $\sigma(x_i)$ is a variable or a *free* constant (i.e., a constant not occurring in the signature of E), for $i = 0, \ldots, m$.

Proposition 6 (Main Proposition). *Let E be a consistent equational theory with signature Σ. Suppose there exists an E-unification problem with constants, γ, containing three distinct free constants a, b, and c and variables $\{x_0, \ldots, x_m\}$ such that for $\vec{x} = \langle x_0, \ldots, x_m \rangle$*

1. *γ has $\vec{x}$-atomic solutions σ_a, σ_b and σ_c that map x_0 to a, b, and c respectively, and*
2. *every $\vec{x}$-atomic solution of γ maps x_0 to one of the constants a, b or c.*

Then solvability of general E-unification problems is NP-hard.

Proof. We shall show that so-called 1-in-3 problems over positive literals can be encoded as general E-unification problems. [7] Solvability of 1-in-3 problems is

[6] The solution that is considered in [BS92] is assumed to be normalized in a particular way. But this point is not relevant for the present discussion.

[7] A 1-*in*-3 *problem over positive literals* is given by a finite set $cl_1, \ldots, cl_n$ of clauses, each clause cl_i containing exactly three *positive* literals. A solution of the problem is a truth value assignment that maps *exactly one* literal of each clause to 1 (true).

well-known to be NP-complete, see [GJ79]. The size of an encoded 1-in-3 problem will be linear in the size of the 1-in-3 problem, which will give the desired result.

1. In the first step we show how to encode a single clause $cl = \langle p_1, p_2, p_3 \rangle$ with three positive literals. Let a, b, c, and $\vec{x}$ as above. For simplicity we shall assume that γ contains just four free constants a, b, c and d. We consider the free signature $\Delta := \{0, 1, f\}$ where 0 and 1 are distinct constants and f is a ternary function symbol. Let F denote the free (empty) theory for signature Δ. Clearly, $E \cup F$ is a consistent equational theory and $1 \neq_{E \cup F} 0$. Let z_1, z_2, z_3 be three distinct variables that do not occur in γ. The variables z_1, z_2, z_3 will be used to represent p_1, p_2, p_3. For each $i \in \{1, \ldots, m\}$, let $y_{i,1}, y_{i,2}, y_{i,3}$ be a collection of three new variables (not occurring in γ and distinct from z_1, z_2, z_3). Let γ_F denote the elementary F-unification problem

$$x_0 = f(z_1, z_2, z_3) \wedge a = f(1,0,0) \wedge b = f(0,1,0) \wedge c = f(0,0,1) \wedge d = f(1,1,1) \\ \wedge \bigwedge_{i=1}^{m} x_i = f(y_{i,1}, y_{i,2}, y_{i,3}).$$

In this problem, a, b, c, and d are treated as variables. With γ_E we denote the variant of the system γ where a, b, c, d are treated as variables. Now consider the elementary $(E \cup F)$-unification problem in decomposed form

$$\gamma^* := \gamma_E \wedge \gamma_F.$$

We shall verify the following two claims:

Claim 1 *For each triple $(i, j, k) \in \{(1,0,0), (0,1,0), (0,0,1)\}$ there exists a solution σ of γ^* such that (z_1, z_2, z_3) is mapped to (i, j, k) under σ.*

Claim 2 *Modulo E, each solution of γ^* maps (z_1, z_2, z_3) either to $(1,0,0)$, or to $(0,1,0)$, or to $(0,0,1)$.*

Note that these claims can be interpreted in the sense that solutions of γ^* may be used to "select" (via identification with 1) exactly one of the elements z_1, z_2 and z_3, and that each solution in fact provides for such a unique selection.

Proof of Claim 1: we show that there exists a solution σ of γ^* such that (z_1, z_2, z_3) is mapped to $(1,0,0)$ under σ, the other cases can be treated analoguosly. By assumption, γ has an $\vec{x}$-atomic solution σ_a that maps x_0 to a. Consider the partition Π of $Var(\gamma^*)$ where two elements u, v of $\{a, b, c, d, x_0, \ldots, x_m\}$ belong to the same class of Π iff $\sigma_a(u) = \sigma_a(v)$, and where the equivalence classes of the variables in $Var(\gamma^*) \setminus \{a, b, c, d, x_0, \ldots, x_m\}$ have just one element. Note that a, b, c, and d belong to distinct equivalence classes of Π since σ_a leaves these elements fixed. On the other hand, x_0 and a belong to the same class.

We select a set of representants Y for Π as follows. Let a, b, c, and d be the representants of their equivalence classes. Choose any representant for the variables in $\vec{x}$ that belong to other classes of Π. All the remaining equivalence classes of Π have just one element which is the representant of the class. Let Lab be the labelling function on Y where the representants occurring in γ_F receive label Δ and all the other representants receive label Σ. Let $<$ be any

linear ordering on Y such that all representants with label Δ are smaller than all the representants with label Σ. We consider the linear constant restriction $L := (Lab, <)$ on Y. Let $\gamma_{2,E}$ and $\gamma_{2,F}$ be the formulae that are obtained from γ_E and γ_F by replacing each occurrence of a variable by its representant. Now

$$((\gamma_{2,E}, L), (\gamma_{2,F}, L))$$

is a possible output pair of the Decomposition Algorithm.

We claim that both components are solvable problems. First we consider $(\gamma_{2,E}, L)$. The choice of the linear ordering $<$ guarantees that $(\gamma_{2,E}, L)$ can be considered as a usual E-unification problem with constants. In fact, since Δ-variables are smaller than Σ-variables with respect to $<$, the linear constant restriction L does not impose any real restriction on the Σ-variables of $\gamma_{2,E}$. The constants occurring in the problem are a, b, c, d and the representants of the variables in $\vec{x}$.

Let τ be the function that maps each atom $\sigma_a(x_i)$ to the representant of x_i $(0 \leq i \leq m)$. The choice of representants guarantees that τ leaves a, b, c, and d fixed, hence τ can be regarded as a Σ-substitution. Let $\sigma_E := \sigma_a \circ \tau$. We want to show that σ_E is a solution of $(\gamma_{2,E}, L)$.

We have to verify that σ_E treats Δ-variables as constants. This is clear for a, b, c, and d. Let x_k be the representant of x_l for some $0 \leq k, l \leq m$. Then $\sigma_E(x_k) = \tau(\sigma_a(x_k)) = \tau(\sigma_a(x_l))$ is the representant of x_l, namely x_k.

It remains to show that σ_E solves the equations of $\gamma_{2,E}$. Let $s_1 = s_2$ be an equation of $\gamma_{2,E}$, and let $t_1 = t_2$ be the corresponding equation of γ. Recall that t_i is obtained from s_i be replacing all occurrences of variables in $\vec{x}$ by their representants, for $i = 1, 2$. By assumption $\sigma_a(t_1) =_E \sigma_a(t_2)$. The choice of the partition Π shows that $\sigma_a(s_1) =_E \sigma_a(s_2)$. Hence $\tau(\sigma_a(s_1)) =_E \tau(\sigma_a(s_2))$ and $\sigma_E(s_1) =_E \sigma_E(s_2)$.

The second system, $(\gamma_{2,F}, L)$, does not contain any variable with label Σ, which means that the linear constant restriction L does not impose a real condition. We may treat the system as an elementary F-unification problem. Recall also that a, b, c, d are four distinct variables of $(\gamma_{2,F}, L)$. Obviously, there exists a solution σ_F of $(\gamma_{2,F}, L)$ mapping (z_1, z_2, z_3) to $(1, 0, 0)$.

It follows now from Remark 3 that γ^* has a solution σ such that

$$(\sigma(z_1), \sigma(z_2), \sigma(z_3)) = (1, 0, 0).$$

This completes the proof of Claim 1.

Proof of Claim 2: Let σ be a solution of γ^*. By Proposition 2 there exists a solvable output pair $((\gamma_{2,E}, L), (\gamma_{2,F}, L))$ of the Decomposition Algorithm. An analysis of γ^* gives some information on the variable identification step and on L. First note that the representants of the variables a, b, c, d and $x_0, \ldots, x_m$ necessarily must receive label Δ in L since otherwise $(\gamma_{2,F}, L)$ would be unsolvable. For the same reason, the four variables a, b, c, d must belong to different equivalence classes of the partition that has been selected. Without loss of generality we may assume that a, b, c, and d are used as representants of their equivalence

classes. Let σ_E be a solution of $(\gamma_{2,E}, L)$. We assume that σ_E leaves all variables fixed that do not occur in $\gamma_{2,E}$. We may now consistently extend σ_E, mapping each variable of γ_E to the image of its representant under σ_E. In this way, we obtain a solution σ_0 of γ_E. Note that σ_0, similarly as σ_E, treats a, b, c, d and the representants of the variables in $\vec{x}$ as constants since these elements are Δ-variables of $\gamma_{2,E}$. Therefore σ_0 is an $\vec{x}$-atomic solution of γ.

By the assumption of the proposition, σ_0 maps x_0 to one of the constants a, b, c. Let us assume that $\sigma_0(x_0) = a$. But this implies, by the choice of σ_0, that a is the representant of x_0. Let z_1', z_2', and z_3' denote the representants of the equivalence classes of z_1, z_2 and z_3 respectively. We have seen that the problem which is reached after the variable identification step contains the equations $a = f(z_1', z_2', z_3')$ and $a = f(1, 0, 0)$.

By Remark 4 we may assume without loss of generality that in the variable identification step two variables u and v of γ^* are identified iff $\sigma(u) =_E \sigma(v)$. This means that σ solves the equations $a = f(z_1', z_2', z_3')$ and $a = f(1, 0, 0)$ modulo E. Hence

$$\begin{aligned} f(1,0,0) &= \sigma(f(1,0,0)) =_E \sigma(f(z_1', z_2', z_3')) = f(\sigma(z_1'), \sigma(z_2'), \sigma(z_3')) \\ &=_E f(\sigma(z_1), \sigma(z_2), \sigma(z_3)). \end{aligned}$$

It is well-known that the Δ-reducts of the joint term algebra $\mathcal{T}(\Sigma \cup \Delta, Var)/=_E$ and of the pure term algebra $\mathcal{T}(\Delta, Var)$ are Δ-isomorphic. This shows that σ maps (z_1, z_2, z_3) to $(1, 0, 0)$ modulo E.

2. In the second part of the proof we show how to encode a 1-in-3 problem with clauses $cl_1, \ldots, cl_n$ containing the positive literals $p_1, \ldots, p_k$. Let $z_1, \ldots, z_k$ be a fixed set of distinct variables. The clause cl_i will be encoded by the elementary $(E \cup F)$-unification problem γ_i^* that is obtained from the formula γ^* defined above in the following way. If cl_i has the form $\langle p_q, p_r, p_s \rangle$, then we use the variables z_q, z_r, z_s instead of z_1, z_2, z_3. Clearly, z_q, z_r, z_s encode p_q, p_r, p_s just as z_1, z_2, z_3 encoded p_1, p_2, p_3 before. For all other variables occurring in γ^* (in particular for a, b, c, d and the variables in $\vec{x}$) we use a fresh copy for each of the subproblems γ_i^* (to be denoted $a^i, b^i, x_0^i, \ldots$). In this way, the general E-unification problems $\gamma_1^*, \ldots, \gamma_n^*$ share only variables in $\{z_1, \ldots, z_k\}$. Modulo the values of these variables they can be solved independently. Now $\gamma_1^*, \ldots, \gamma_n^*$ is used for encoding $cl_1, \ldots, cl_n$.

Assume that the 1-in-3 problem $cl_1, \ldots, cl_n$ has a solution. Then there exists a mapping $S : \{z_1, \ldots, z_k\} \to \{0, 1\}$ such that in each problem γ_i^*, with equation $x_0^i = f(z_q, z_r, z_s)$, say, exactly one of the variables z_q, z_r, z_s is mapped to 1 under S, while the remaining two variables are mapped to 0. It follows from Claim 1 that γ_i^* has a solution σ_i such that (z_q, z_r, z_s) is mapped to $(S(z_q), S(z_r), S(z_s))$ under σ_i. Since the distinct subproblems $\gamma_1^*, \ldots, \gamma_n^*$ share only variables in $\{z_1, \ldots, z_k\}$ it follows that the general E-unification problem $\gamma_1^* \wedge \ldots \wedge \gamma_n^*$ has a solution.

Conversely, if $\gamma_1^* \wedge \ldots \wedge \gamma_n^*$ has a solution, then Claim 2 shows that there exists a mapping $S : \{z_1, \ldots, z_k\} \to \{0, 1\}$ respectively $S' : \{p_1, \ldots, p_k\} \to \{0, 1\}$ which

represents a solution of the 1-in-3 problem $cl_1, \ldots, cl_n$. □

In the rest of this subsection we consider in more detail the situation where $\gamma_1^* \wedge \ldots \wedge \gamma_n^*$ is used as the input of the Decomposition Algorithm. We want to show that from every partition Π of $Var(\gamma_1^* \wedge \ldots \wedge \gamma_n^*)$ in the variable identification step that leads to a solvable output pair we can read of a solution of $cl_1, \ldots, cl_n$. In the sequel, E and γ are as in the Main Proposition and expressions $cl_i, \gamma_i^*, x_0^i, a^i, b^i, c^i$ etc. refer to the same entities as in the previous proof. If Π is a partition of $X := Var(\gamma_1^* \wedge \ldots \wedge \gamma_n^*)$, then $[x]_\Pi$ denotes the equivalence class of $x \in X$ with respect to Π.

Definition 7. A partition Π of $Var(\gamma_1^* \wedge \ldots \wedge \gamma_n^*)$ is *locally correct* if, for all $i = 1, \ldots, n$, the equivalence classes $[a^i]_\Pi, [b^i]_\Pi$ and $[c^i]_\Pi$ are pairwise distinct and $x_0^i \in [a^i]_\Pi \cup [b^i]_\Pi \cup [c^i]_\Pi$.

Proposition 8. *Assume that we reach, for input $\gamma_1^* \wedge \ldots \wedge \gamma_n^*$, a solvable output pair of the Decomposition Algorithm, selecting the partition Π of $Var(\gamma_1^* \wedge \ldots \wedge \gamma_n^*)$ in the variable identification step. Then Π is locally correct.*

Proof. This follows as in the proof of Claim 2 above. □

Given a locally correct partition Π on $Var(\gamma_1^* \wedge \ldots \wedge \gamma_n^*)$, we define, for each clause $cl_i = \langle p_q, p_r, p_s \rangle$ in $cl_1, \ldots, cl_n$, a local truth value assignment $S_\Pi^i : \{p_q, p_r, p_s\} \to \{0, 1\}$ in the following way. Assume that γ_i^* contains the equations

$$x_0^i = f(z_q, z_r, z_s), a^i = f(1,0,0), b^i = f(0,1,0), c^i = f(0,0,1).$$

Then S_Π^i has the following form, depending on whether (1) $x_0^i \in [a^i]_\Pi$, (2) $x_0^i \in [b^i]_\Pi$, or (3) $x_0^i \in [c^i]_\Pi$:

$$(1)\ S_\Pi^i := \begin{bmatrix} p_q \mapsto 1 \\ p_r \mapsto 0 \\ p_s \mapsto 0 \end{bmatrix}, (2)\ S_\Pi^i := \begin{bmatrix} p_q \mapsto 0 \\ p_r \mapsto 1 \\ p_s \mapsto 0 \end{bmatrix}, (3)\ S_\Pi^i := \begin{bmatrix} p_q \mapsto 0 \\ p_r \mapsto 0 \\ p_s \mapsto 1 \end{bmatrix}.$$

Let $S_\Pi := \bigcup_{i=1}^n S_\Pi^i$ denote the union of these local truth value assignments.

Proposition 9. *Assume that we reach, for input $\gamma_1^* \wedge \ldots \wedge \gamma_n^*$, the output pair $((\gamma_E, L), (\gamma_F, L))$ of the Decomposition Algorithm, selecting the locally correct partition Π of $Var(\gamma_1^* \wedge \ldots \wedge \gamma_n^*)$ in the variable identification step. Let $p_q \in \{p_1, \ldots, p_k\}$. If, for some $1 \leq i \leq n$, $S_\Pi^i(p_q) = 1$ (resp. $S_\Pi^i(p_q) = 0$), then the representant z_q' of $[z_q]_\Pi$ is mapped to 1 (resp. 0) under every solution of γ_F.*

Proof. We may assume that cl_i has the form $\langle p_q, p_r, p_s \rangle$. Assume first that $S_\Pi^i(p_q) = 1$. Then $x_0^i \in [a^i]_\Pi$, by definition of S_Π^i. Let y, z_q', z_r', z_s' denote the representants of a^i, z_q, z_r and z_s in γ_F respectively. Then γ_F contains the equations $y = f(z_q', z_r', z_s')$ and $y = f(1,0,0)$ and the result follows. In the other case, where $S_\Pi^i(p_q) = 0$, we know that $x_0^i \in [b^i]_\Pi$ or $x_0^i \in [c^i]_\Pi$. The rest is as in the first case. □

Proposition 10. *Assume that we reach, for input $\gamma_1^* \wedge \ldots \wedge \gamma_n^*$, a solvable output pair $((\gamma_E, L), (\gamma_F, L))$ of the Decomposition Algorithm, selecting the locally correct partition Π of $\mathrm{Var}(\gamma_1^* \wedge \ldots \wedge \gamma_n^*)$ in the variable identification step. Then S_Π solves $cl_1, \ldots, cl_n$.*

Proof. Since (γ_F, L) is solvable it follows from the previous proposition that two local assignments S_Π^i and S_Π^j agree on the common literals in their domain, for $1 \leq i, j \leq n$. Hence S_Π is a truth value assignment on $\{p_1, \ldots, p_k\}$. The form of the S_Π^i shows that S_Π solves $cl_1, \ldots, cl_n$. □

4.2 Intractable E-unification problems

In this subsection we shall use the criterion given in the Main Proposition to prove that general E-unification is NP-hard for all the theories mentioned in the introduction. We begin with the general results. In the following two theorems we consider theories that have an associative or a commutative function symbol. It should be clear that these function symbols may have other properties as well.

Theorem 11. *Let E be an equational theory that contains an associative function symbol "$\circ$". If E is regular, then the problem of deciding solvability of general E-unification problems is NP-hard.*

Proof. Consider the E-unification problem with constants γ of the form

$$y \circ x \circ z = a \circ a \circ b \circ c \circ c.$$

Let $\vec{x} := \langle x \rangle$. Since $\circ$ is associative, it is obvious that γ has $\vec{x}$-atomic solutions that map x to a, b, and c respectively. Now let σ be any $\vec{x}$-atomic solution of γ. We have $\sigma(y) \circ \sigma(x) \circ \sigma(z) =_E a \circ a \circ b \circ c \circ c$. Since E is regular, the atom $\sigma(x)$ of the left-hand side must occur on the right-hand side (Remark 1). It follows that $\sigma(x)$ is a, b, or c. By Proposition 6, general E-unification is NP-hard. □

Theorem 12. *Let E be an equational theory that contains a commutative function symbol "f". If E is regular, then the problem of deciding solvability of general E-unification problems is NP-hard.*

Proof. Consider the E-unification problem with constants γ of the form

$$f(f(x, y), f(u, v)) = f(f(a, b), f(b, c)).$$

Let $\vec{x} := \langle x \rangle$. Since f is commutative, it is obvious that γ has $\vec{x}$-atomic solutions that map x to a, b, and c respectively. Now let σ be any $\vec{x}$-atomic solution of γ. We have $f(f(\sigma(x), \sigma(y)), f(\sigma(u), \sigma(v))) =_E f(f(a, b), f(b, c))$. Since E is regular, the atom $\sigma(x)$ of the left-hand side must occur on the right-hand side. It follows that $\sigma(x)$ is a, b, or c. By Proposition 6, general E-unification is NP-hard. □

The theorems show, for example, that general E-unification is NP-hard for the following equational theories E: the theory A of an associative function symbol, the theory C of a commutative function symbol, the theory AC of an associative and commutative function symbol, and the theory ACI of an associative, commutative and idempotent function symbol.

The Main Proposition can be strengthened under suitable assumptions on E. In the sequel, $\mathcal{K}$ denotes the class of all equational theories that satisfy the criterion of the Main Proposition.

Theorem 13. *Let $E \in \mathcal{K}$. Assume that the problem of deciding solvability of E-unification problems with linear constant restrictions is in NP. Then the problem of deciding solvability of general E-unification problems is NP-complete.*

Proof. Under the given assumption the Decomposition Algorithm yields an NP algorithm for deciding solvability of general E-unification problems, see Consequence 5 (pg. 216) of Theorem 2.1 in [BS96]. The result follows immediately from Proposition 6. □

Corollary 14. *Let E be a regular equational theory that contains an associative or commutative function symbol. If there exist an NP algorithm for deciding solvability of E-unification problems with linear constant restrictions, then the problem of deciding solvability of general E-unification problems is NP-complete.*

Let us now look at the non-regular theories mentioned in the introduction (for which E-unification with constants is polynomially decidable). The results mentioned below are known, our point is just the simple new proof.

Corollary 15. *Solvability of general E-unification problems is NP-hard for the theories $E = ACUN, ACUNh$, and AG (to be defined below).*

Proof. In each case, we shall give a particular E-unification problem with constants, γ, and we shall show that this problem has the properties mentioned in the Main Proposition.

Associativity, Commutativity, Nilpotency with Unit (ACUN). This theory, discussed in [GN96], is formulated over the binary nilpotent AC-symbol $+$ and the constant 0. The axioms for nilpotency and unity are $x+x = 0$ and $x+0 = x$. We consider the problem γ of the form $x+y+z = a+b+c$ and choose $\vec{x} = \langle x, y, z\rangle$. It is obvious that γ has $\vec{x}$-atomic solutions that map x to a, b, and c respectively. Conversely, let σ be an $\vec{x}$-atomic solution of γ. If the atoms $\sigma(x), \sigma(y)$, and $\sigma(z)$ are distinct, then $\{\sigma(x), \sigma(y), \sigma(z)\} = \{a, b, c\}$ and we are done. But it is easy to see that an $\vec{x}$-atomic solution of γ cannot identify two (or three) of the atoms $\sigma(x), \sigma(y)$, and $\sigma(z)$.

Associativity, Commutativity, Nilpotency with Unit and Homomorphism. This theory (denoted $ACUNh$), also discussed in [GN96], is similar to $ACUN$. There is one additional function symbol h, the additional axioms are $h(x+y) = h(x)+h(y)$ and $h(0) = 0$. We may use the same unification problem as in the case of $ACUN$.

Theory of Abelian Groups (AG). The signature of the theory AG, which is treated in [HK96], has a binary associative and commutative function symbol $+$, a unary symbol $-$, and a constant e. The axioms are $x + e = x, x + (-x) = e, x + y = y + x$, and $(x + y) + z = x + (y + z)$. We consider the problem γ of the form $x + y + z = a + b + c$ and choose $\vec{x} = \langle x, y, z \rangle$. Again it is trivial to see that this AG-unification problem satisfies the requirements mentioned in the Main Proposition. □

It is interesting to note that all the problems that we used to verify the criterion of the Main Proposition are matching problems since the right-hand side is always ground. We conjecture that the complexity results of this subsection can be generalized to procedures that decide solvability of general E-matching problems.

4.3 Impossibility of polynomial combination and limitations for optimizing the combination algorithm

In view of the complexity results of the previous subsections it is natural to ask if the following *conjecture* mentioned in the introduction is true:

> *Assume that $P \neq NP$. Let $E \in \mathcal{K}$. Then there exists no polynomial combination algorithm that reduces solvability of general E-unification problems to solvability of E-unification problems with constants plus solvability of free (syntactic) unification problems.*

When we look carefully at the conjecture, we see that it is difficult to interpret it in a precise way. In fact, when we are talking here about E-unification with constants, we do of course not want to exclude "closely related" output problems such as, e.g., E-unification with linear constant restriction. Actually, we do not see any general and convincing definition of the type of output problem that one would still be willing to accept. Because of this vagueness it seems difficult either to prove or to refute the conjecture. We shall now introduce one possible formalization of the conjecture that arises naturally if one tries to optimize the Decomposition Algorithm in the context of general E-unification. Here the conjecture can be verified.

Definition 16. A *polynomial optimization of the Decomposition Algorithm for general E-unification* is an algorithm that accepts as input an arbitrary general E-unification problem γ_0 and computes in polynomial time a finite set M of output pairs $((\gamma_E, L), (\gamma_F, L))$ of E- resp. free unification problems with linear constant restriction such that γ_0 is solvable iff, for some output pair in M, both components are solvable. More specifically we demand that each output pair in M is also a possible output pair of the original Decomposition Algorithm.

Theorem 17. *Let $E \in \mathcal{K}$. Then there exists no polynomial optimization of the Decomposition Algorithm for general E-unification, unless $P = NP$.*

Proof. Assume that there exists a polynomial optimization. We shall then show how solvability of 1-in-3 problems over positive literals can be decided in polynomial time, which yields the desired contradiction. We refer to the notations introduced at the end of Subsection 4.1. Given a 1-in-3 problem $cl_1, \ldots, cl_n$, we encode it into a general E-unification problem $\gamma_1^* \wedge \ldots, \wedge \gamma_n^*$ as in the proof of the Main Proposition. We use $\gamma_1^* \wedge \ldots, \wedge \gamma_n^*$ as the input of the polynomial optimization. The output set M contains only a polynomial number of output pairs. From M, we eliminate all pairs that are based on a partition Π of $Var(\gamma_1^* \wedge \ldots, \wedge \gamma_n^*)$ which is not locally correct. Let M_0 be the new set. We claim that $cl_1, \ldots, cl_n$ has a solution iff S_Π yields a solution, for some partition Π of $Var(\gamma_1^* \wedge \ldots, \wedge \gamma_n^*)$ that was used for an output pair in M_0. In fact, assume that $cl_1, \ldots, cl_n$ is solvable. Then $\gamma_1^* \wedge \ldots, \wedge \gamma_n^*$ is solvable (as we saw in the proof of the Main Proposition) and M contains a solvable output pair. By Proposition 8, M_0 contains a solvable output pair. Hence the claim follows from Proposition 10. Clearly, the computation of all relations S_Π for partitions Π used in M_0 needs only polynomial time in the size of $cl_1, \ldots, cl_n$ under the given assumptions. □

5 Conclusion

In this paper we have introduced a criterion that characterizes a large class $\mathcal{K}$ of equational theories E where solvability of general E-unification problems is always NP-hard. We have seen that for $E \in \mathcal{K}$ there exists no polynomial optimization of the combination algorithm given in [BS92] if applied to general E-unification, unless $P = NP$. By and large, the list of equational theories E that belong to $\mathcal{K}$ which was given in Subsection 4.2 indicates that most of the relevant regular theories belong to $\mathcal{K}$.

The case where we combine a regular equational theory with the free (empty) theory is one of the simplest situations for combination. On this background, there cannot be much hope that it is possible to characterize any interesting and large class of equational theories where a polynomial combination of unification algorithms is possible. On the other hand, polynomial combination is possible under strong restrictions. In [Sc96] we have shown that a polynomial combination algorithm, even for disunification, exists for unitary, regular and collapse-free theories over disjoint signatures if both theories have polynomial algorithms for computing most general unifiers, for unification with constants.

References

[BS92] F. Baader, K.U. Schulz, "Unification in the union of disjoint equational theories: Combining decision procedures," in: *Proc. CADE-11*, LNAI 607, 1992, pp. 50-65.

[BS96] F. Baader, K.U. Schulz, "Unification in the union of disjoint equational theories: Combining decision procedures," *Journal of Symbolic Computation*, **21** (1996), pp. 211-243.

[BS94] F. Baader, J. Siekmann, "Unification Theory," in D.M. Gabbay, C. Hogger, and J. Robinson, Editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, Oxford University Press, Oxford, UK, 1994, pp. 41–125.

[Ba91] L. Bachmair, *Canonical equational proofs*, Progress in Theoretical Computer Science, Birkhäuser, Boston, 1991.

[Bo93] A. Boudet, "Combining Unification Algorithms," *Journal of Symbolic Computation* **16** (1993) pp. 597-626.

[GJ79] M.R. Garey, D.S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness," W.H. Freeman and Co. San Francisco (1979).

[GN96] Q. Guo, P. Narendran, and D.A. Wolfram "Unification and Matching modulo Nilpotence," manuscript, received from Qing Guo, guo@cs.albany.edu, 1996.

[HK96] M. Hermann, P.G. Kolaitis, "Unification Algorithms Cannot be Combined in Polynomial Time," in *Proceedings of the 13th International Conference on Automated Deduction*, M.A. McRobbie and J.K. Slaney (Eds.), Springer *LNAI* **1104**, 1996, pp. 246-260.

[He86] A. Herold, "Combination of Unification Algorithms," *Proceedings of the 8th International Conference on Automated Deduction*, *LNCS* **230**, 1986, pp. 450-469.

[JoK86] J.P. Jouannaud, H. Kirchner, "Completion of a set of rules modulo a set of equations," *SIAM J. Computing* 15, 1986, pp. 1155–1194.

[KN92] D. Kapur, P. Narendran, "Complexity of Unification Problems with Associative-Commutative Operators," *J. Automated Reasoning* **9**, 1992, pp. 261-288.

[KR96] S. Kepser, J. Richts, "Optimization Techniques for the Combination of Unification Algorithms," to be submitted.

[Ki85] C. Kirchner, "Méthodes et outils de conception systématique d'algorithmes d'unification dans les théories équationelles," Thèse d'Etat, Université de Nancy 1, France, 1985.

[NR94] R. Niewenhuis, A. Rubio, "*AC*-superposition with constraints: No *AC*-unifiers needed," in *Proceedings of the 12th International Conference on Automated Deduction, Nancy, France*, A. Bundy (Ed.), Springer *LNAI* 1994, pp.545-559.

[Plo72] G. Plotkin, "Building in equational theories," *Machine Intelligence* 7, 1972, pp. 73–90.

[Ru95] A. Rubio, "Theorem Proving modulo Associativity," in *Proceedings Computer Science Logic CSL'95, Paderborn, Germany*, H. Kleine Büning (Ed.), Springer LNCS 1092 (1995), pp. 452-467.

[Sc89] M. Schmidt-Schauß, "Combination of Unification Algorithms," *J. Symbolic Computation* **8**, 1989, pp. 51-99.

[Sc96] K.U. Schulz, "Combination of Unification and Disunification Algorithms: Tractable and Intractable Instances," Research Paper, available under ftp.cis.uni-muenchen.de, in directory pub/schulz. File name: ComplexityCombination.ps.gz

[Sti85] M. Stickel, "Automated deduction by theory resolution," *J. Automated Reasoning* 1, 1985, pp. 333–356.

[Ti86] E. Tiden, "Unification in Combinations of Collapse Free Theories with Disjoint Sets of Function Symbols," *Proceedings of the 8th International Conference on Automated Deduction*, *LNCS* **230**, 1986.

[Ye87] K. Yelick, "Unification in Combinations of Collapse Free Regular Theories," *J. Symbolic Computation* **3**, 1987.

This article was processed using the LaTeX macro package with LLNCS style