Combining Unification- and Disunification Algorithms—Tractable and Intractable Instances

Klaus U. Schulz
CIS, University of Munich
Oettingenstr. 67
80538 München, Germany
e-mail: schulz@cis.uni-muenchen.de
http://www.cis.uni-muenchen.de/people/schulz.html

Abstract

We consider the problem of combining procedures that decide solvability of (dis)unification problems over disjoint equational theories. Partial answers to the following questions are given:

- Which properties of the component theories imply *intractability* in the sense that there cannot be a polynomial combination algorithm, assuming $P \neq NP$?
- Which general properties of the component theories guarantee *tractability* of the combination problem in the sense that there exists a deterministic and polynomial combination algorithm?

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.
In the second part of the paper we characterize a class of equational theories where disunification algorithms can be combined deterministically and in polynomial time. All unitary, regular and collapse-free equational theories with polynomial unification algorithms belong to this class.

1 Introduction

A general problem that arises in different areas of computer science is the following *combination problem*: given two structures or theories, and two algorithms for answering certain questions about these domains resp. theories, how can we systematically combine these algorithms in order to solve more general "mixed" input problems over a suitable combined domain, or in the union of the theories? Many specific instances of this problem have been exhaustively studied in the areas of automated theorem proving and constraint programming, such as combination of decision procedures [NO79, Ri96, TH96], combination of algorithms to solve the word problem [DKR94, BT96], combination of unification and matching algorithms [Ki85, He86, Ti86, Ye87, Ni89, Sc89, Bo93, BS92, DKR94], combination of disunification algorithms [BS95a], and combination of constraints over non-free solution domains such as rational trees, feature-structures, non-wellfounded sets and lists [Col90, BS95b, KS96].

In the meantime, general combination methods have been developed that solve most of these problems from a theoretical point of view[1] (e.g., [NO79, Sc89, BS92]). Still, these general combination methods are faced with a serious efficiency problem: in these approaches, a mixed input problem is decomposed and transformed into two pure output problems that can be solved independently over the component theories. Typically, this reduction is based on a polynomial number of non-deterministic steps. Hence the combination algorithm introduces its own NP-complexity, disregardless of the complexity of the algorithms that are available for the components. Without further optimizations, most methods are not really useful in practical applications. The present research was started in a project where we tried to optimize the method for combining unification algorithms for equational theories introduced in [BS92]. We investigated how structural properties of the component theories can be used to eliminate parts of the non-determinism of the general combination scheme. It was then natural to ask the following two questions, which give the background for this paper:

- *Which general properties of the component theories guarantee* tractability *of the combination problem in the sense that there exists a deterministic and polynomial combination algorithm?*
- *Which properties of the component theories imply* intractability *in the sense that there cannot be a polynomial combination algorithm, assuming* $P \neq NP$*?*

Both questions are relevant for many combination problems, for this reason we shall not restrict the discussion to the combination of unification algorithms. On the other hand it seems impossible to obtain general answers for all the combination problems mentioned above since a common theoretical background for these problems is still

[1] Strictly speaking, general methods only exist in the case where signatures of both components are disjoint. In this paper, we shall not consider the non-disjoint case.

missing. As a compromise between generality and specificity we consider a variety of combination problems that can be solved—ignoring technical differences—with the same reduction technique. We discuss combination of unification and disunification algorithms for disjoint equational theories, and we comment on combination of constraint solvers for simply combinable structures as described in [BS95b, KS96] in the conclusion. In each case, we shall only consider procedures that decide solvability of a given problem.

In the area of unification theory, some results are known that show that both questions do not have trivial answers. Clearly, combination of theories does not necessarily lead to intractability: when we combine two empty (trivial) equational theories, we can still use a linear unification algorithm for the union of the two theories. In less trivial cases, however, combination destroys tractability: solvability of associative-commutative-idempotent (ACI-) unification problems with free constants is decidable in polynomial time, but the problem of deciding solvability of ACI-unification problems with additional free function symbols (called general ACI-unification) is NP-hard (see [KN91]). The same phenomenon holds for the theory $ACUN$ of an associative-commutative and nilpotent function with unit, and for the theory $ACUNh$ which contains in addition a homomorphic unary function symbol (see [GN96] for formal definitions). These examples show that there cannot be a general polynomial algorithm for combining procedures that decide solvability of unification problems, unless P equals NP. A corresponding result for the case of algorithms that compute minimal complete sets of unifiers modulo finitary equational theories was recently obtained by by M. Hermann and P.G. Kolaitis [HK96], by proving intractability of the counting problem for general unification in the theory of abelian groups (AG-unification).

Even with the negative results that are mentioned in these papers it remains unclear which formal properties of the theories under consideration lead to the observed blow-up of complexity when free function symbols are added. In the first part of this paper we shall isolate such a property. A criterion is given that characterizes a class $\mathcal{K}$ of equational theories where general E-unification 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 represent instances of such theories. The criterion is not restricted to the regular case, and it can also be used to give simple proofs for the fact that general E unification is NP-hard for the theories $ACUN, ACUNh$, and AG mentioned above.

Of course the general complexity result do not depend on any assumption on the complexity of E-unification with constants. For this reason we conjecture that for 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. It seems difficult to prove this general conjecture as long as we do not make any concrete assumption on the form of the combination algorithm and its output problems. For this reason we consider one natural form of such a combination procedure, and we show—assuming $P \neq NP$—that there cannot be a

"polynomial optimization" of the algorithm given in [BS92] for combining E-unification with free unification in the case where $E \in \mathcal{K}$.

When we look for properties that guarantee tractability of combination problems, a useful idea has been introduced already in 1980 in a related context. Oppen [Op80] considered combinations of decidable theories where each component theory is given by means of a quantifier-free axiomatization[2]. He introduced the notion of a "convex" theory and showed—under some additional technical assumptions—that polynomial satisfiability checkers for formulae in disjunctive normal form for convex theories over disjoint signatures can be lifted to a polynomial satisfiability checker for mixed formulae in disjunctive normal form in the union of the theories.[3]

In the second part of this paper we show that the abstract idea of convexity can be applied to the context of combined (dis)unification algorithms. Due to the complex nature of this combination problem, the appropriate notion of convexity turns out to be more complicated than in Oppen's situation. We introduce the new concept of an $\mathcal{L}$-convex equational theory. This notion refers to a constraint language $\mathcal{L}$ that is used to describe sets of admissible "linear constant restrictions" for (dis)unification problems. For $\mathcal{L}$-convex equational theories over disjoint signatures, a deterministic treatment of combined (dis)unification problems is always possible. We show that unitary regular and collapse-free equational theories are $\mathcal{L}$-convex. If polynomial unification algorithms are available for both component theories, this leads to a polynomial algorithm for deciding solvability of disunification problems in the union of the two theories. In the conclusion we indicate how the new notion of $\mathcal{L}$-convexity can be lifted to the class of simply combinable (SC-) structures introduced in [BS95b]. Polynomial combination techniques for solving positive constraints over feature structure domains and rational trees could be obtained on this basis.

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. 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. The size of a term is defined as usual.

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

[2] The signatures of the theories are assumed to be disjoint.

[3] See the conclusion (Section 5) for a brief comparison of both problems and methods.

substitution 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 called *collapse-free* if E does not contain an equation of the form $t = x$ where t is a non-variable term and $x \in Var$. 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 2.1 The following facts can easily be proved for collapse-free (1.) respectively regular (2.) consistent equational theories:

1. $\forall x \in Var, t \in T(\Sigma, Var) : x =_E t$ iff $t = x$.
2. $\forall x \in Var, s, t \in T(\Sigma, Var) : x \in Var(s)$ and $s =_E t$ implies $x \in Var(t)$.

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 a *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.

An elementary *E-disunification problem* is a finite set γ of equations and negated equations (written in the form $s \neq t$) between Σ-terms. A *solution* of an elementary E-disunification problem γ is a Σ-substitution σ such that $\sigma(s) =_E \sigma(t)$ (resp. $\sigma(s) \neq_E \sigma(t)$) for all (dis-)equations $s = t$ (resp. $s \neq t$) in γ. As in the case of unification problems, this notion can be lifted to E-disunification problems with constants and to general E-disunification problems in the obvious way. It should be mentioned that we consider here only one possible semantics for disunification problems. Often these problems are also solved over the ground term algebra modulo E, the initial algebra for E. For a more thourough description of disunification we refer to [Bür88, Com91].

An equational theory E is *unitary* if every elementary E-unification problem γ has a *most general unifier*, i.e., a unifier μ such that for every unifier τ of γ there exists a substitution λ such that $\tau(x) =_E \lambda(\mu(x))$ for all $x \in Var(\gamma)$.

Let γ be an elementary E-(dis)unification problem over the signature Σ, let $Var(\gamma) \subseteq$

Y be a finite set of variables, and let Δ be another signature. A *linear constant restriction* for Y is a pair $L = (Lab, <_L)$ where $<_L$ 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-(dis)unification problem with linear constant restriction.* A Σ-substitution σ *solves* (γ, L) if σ solves the E-(dis)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 $y <_L x$.

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

3 Intractability Results

In this section 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-(dis)unification problems is NP-hard. The power of the criterion will be illustrated in the third subsection. Since intractability results for combined unification problems immediately generalize to the disunification case, we shall only consider combination of unification algorithms in this section. For technical reasons we start with a description of the combination procedure for unification algorithms for equational theories given in [BS92]: the proof of the central proposition of the section depends on the correctness of this combination algorithm. At the end of this section we shall show that the criterion gives also a serious limitation for attempts to optimize the combination algorithm in the context of general E-unification.

3.1 The combination algorithm for unification

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 *Algorithm 1*, 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.

Algorithm 1. In the *first step*, the input problem γ_0 is transformed into an elementary $(E \cup F)$-unification problem $\gamma_1 \equiv \gamma_{1,E} \wedge \gamma_{1,F}$ in decomposed form 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 $<_L$ 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 $L = (Lab, <_L)$. 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", and needs only a polynomial[4] number of steps (see [BS92]). 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 3.1 *The input problem, γ_0, has a solution iff there exists an output pair of Algorithm 1, $((\gamma_{2,E}, L), (\gamma_{2,F}, L))$, 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 3.1 given in [BS92] will be used.

Remark 3.2 It was shown (p. 58) how given solutions σ_E and σ_F of an output pair of Algorithm 1 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 Σ-variable, then $\sigma(y) = \sigma_F(y)$.

Remark 3.3 It was described (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 Algorithm 1 that lead to an output pair $((\gamma_{2,E}, L), (\gamma_{2,F}, L))$ where both components are solvable.[5] In the second step of this construction, two variables v_1 and v_2 of the decomposed problem are identified iff $\sigma(v_1) =_{E \cup F} \sigma(v_2)$.

3.2 A criterion for intractability

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

Definition 3.4 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

[4] Polynomial in the size of γ_0. This is not a sharp estimate.

[5] 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.

$\sigma(x_i)$ is a variable or a *free* constant (i.e., a constant not occurring in the signature of E), for $i = 0, \ldots, m$.

Proposition 3.5 (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. Solvability of 1-in-3 problems is 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

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

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 3.2 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 3.1 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 3.3 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 $cl_1, \ldots, cl_n$. □

Before we look at some applications of the Main Proposition 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 3.6 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 3.7 *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 3.8 *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 $\mathrm{Var}(\gamma_1^* \wedge \ldots \wedge \gamma_n^*)$ 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 3.9 *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^i 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 $cl_1, \ldots, cl_n$. □

3.3 Applications

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. Before we consider these theories, we give two much more 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 3.10 *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-unfication 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 2.1). It follows that $\sigma(x)$ is a, b, or c. By Proposition 3.5, general E-unification is NP-hard. □

Theorem 3.11 *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-unfication 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 3.5, 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 Propositon can be strengthened if we know that the algorithms for the component theories are NP-algorithms.

Theorem 3.12 *Let E be an equational theory that satisfies the criterion of the Main Proposition. Assume that there exists 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.*

Proof. This follows immediately from Proposition 3.5 with Consequence 5 (pg. 216) of Theorem 2.1 in [BS96]. □

Corollary 3.13 *Let E be a regular equational theory that contains an associative or commutative function symbol. Of 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.

Corollary 3.14 *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 (ACUNh). This theory, 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 Propositon. □

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 assume that the complexity results of this section can be generalized to procedures that decide solvability of general E-matching problems.

3.4 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 if the following **conjecture** is true: *Assume that $P \neq NP$. Let E be an equational theory that satisfies the criterion of the Main Proposition. Then there cannot be any 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 and non-vague 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 there seems to be no way to prove or 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 3.15 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 3.16 *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 3.2. 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 3.7, M_0 contains a solvable output pair. Hence the claim follows from Proposition 3.9. 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. □

4 Tractable combined disunification problems

In this section we shall isolate a class of equational theories where there exists a general deterministic and polynomial method for combining disunification algorithms. Our starting point is the decomposition algorithm for solving disunification problems in the union of disjoint equational theories given in [BS95a].

4.1 The combination algorithm for disunification

Let E and F be two equational theories over disjoint signatures Σ and Δ respectively. An elementary $(E \cup F)$-disunification problem is in *decomposed form* if it has the form $\gamma_E \wedge \gamma_F \wedge \gamma_{\neq}$ where γ_E (γ_F) is a finite set of pure Σ-equations (resp. Δ-equations) and where $\gamma_{\neq}$ is a finite set of disequations between variables.

Suppose that we want to decide solvability of an elementary $(E \cup F)$-disunification problem γ_0. The following *Algorithm 2*, described in more detail in [BS95a], reduces γ_0 non-deterministically to a finite number of output pairs. Each component of an output pair represents an (E resp. F-) disunification problem with linear constant restriction.

Algorithm 2. In the *first step*, the input problem γ_0 is transformed into an elementary $(E \cup F)$-disunification problem γ_1 in the decomposed form $\gamma_{1,E} \wedge \gamma_{1,F} \wedge \gamma_{1,\neq}$ such that γ_0 is solvable iff γ_1 is solvable. In the *second step*, a partition Π of $Var(\gamma_1)$ is chosen. This partition must satisfy the following requirement: if $u \neq v$ is a disequation of $\gamma_{1,\neq}$, then u and v must not belong to the same equivalence class w.r.t. Π. 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 $\gamma_{2,E} \wedge \gamma_{2,F} \wedge \gamma_{2,\neq}$. 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 $<_L$ on Y are chosen. The output pair determined by the choices in steps $2-4$, then, is $((\gamma_{2,E} \wedge \gamma_{2,\neq}, L), (\gamma_{2,F} \wedge \gamma_{2,\neq}, L))$, where $L = (Lab, <_L)$. In the first (second) component, the variables with label Δ (resp. Σ) are treated as constants.

Similarly as for Algorithm 1, the first step is deterministic and needs only a polynomial number of steps. The number of variables of γ_1 is linear in the size of γ_0. The following proposition was proved in [BS95a] (see Prop. 3.3, p. 237).

Proposition 4.1 *The input problem, γ_0, has a solution in $\mathcal{T}(\Sigma \cup \Delta, Var)/=_{E \cup F}$ iff there exists an output pair of the Algorithm 2, $((\gamma_{2,E} \wedge \gamma_{2,\neq}, L), (\gamma_{2,F} \wedge \gamma_{2,\neq}, L))$, such that both the E-disunification problem with linear constant restriction $(\gamma_{2,E} \wedge \gamma_{2,\neq}, L)$ has a*

solution and the F-disunification problem with linear constant restriction $(\gamma_{2,F} \wedge \gamma_{2,\neq}, L)$ has a solution.

4.2 Constrained disunification problems

We now introduce a constraint language that will be used to formulate partial descriptions of the linear constant restrictions that may be chosen in the steps of Algorithm 2. Throughout this subsection, γ_1 denotes a disunification problem in decomposed form, reached after the first step of Algorithm 2. With X we denote the set $Var(\gamma_1)$. Π denotes a partition of X, and Y is a set of representants for Π, as chosen in the second step of Algorithm 2.

Definition 4.2 The atomic formulae of the constraint language $\mathcal{L}(X)$ have the form $u = v$ (equality constraints), $u : \Sigma$ or $u : \Delta$ (labelling constraints), or $u \leq v$ (ordering constraints), where $u, v \in X$. In addition, $\mathcal{L}(X)$ contains the following Boolean combinations of atomic formulae:

- disequations $\neg u = v$, written in the form $u \neq v$,
- strict ordering constraints $u \leq v \wedge u \neq v$, written in the form $u < v$,
- implications of the form $u : \Delta \Rightarrow u < v$.

A *constraint set* (for X) is a subset C of $\mathcal{L}(X)$.

Given a linear constant restriction $L = (Lab, <_L)$ on Y and a constraint $c \in \mathcal{L}(X)$, we say that c *holds in* L (symbolically $L \models c$) in the following cases:

- $L \models u = v$ iff $rep(u) = rep(v)$,
- $L \models u : \Sigma$ (resp. $L \models u : \Delta$) iff $Lab(rep(u)) = \Sigma$ (resp. $Lab(rep(u)) = \Delta$).
- $L \models u \leq v$ iff $rep(u) \leq_L rep(v)$.

Here $rep(u) \in Y$ denotes the representant of $x \in X$ with respect to Π. The notion $L \models c$ is extended in the canonical way to general constraints $c \in \mathcal{L}(X)$, lifting Boolean connectives to the meta-level.

Definition 4.3 A constraint set C is *compatible* with the linear constant restriction L on Y (we write $L \models C$), and L is compatible with C, if $L \models c$ for all $c \in C$.

Definition 4.4 A constraint set C is *closed* iff

$$C = \{c \in \mathcal{L}(X) \mid L \models c \text{ for all linear constant restrictions } L \text{ on } Y \text{ s.th. } L \models C\}.$$

With $\mathcal{C}_{\mathcal{L}(X)}$ we denote the set of all closed constraint sets. Two constraint sets C_1 and C_2 for X are called *equivalent* if they are compatible with the same linear constant restrictions on Y. It follows from Definition 4.4 that

- for each constraint set C there exists exactly one closed constraint set $Cl_{\mathcal{L}(X)}(C)$ such that C and $Cl_{\mathcal{L}(X)}(C)$ are equivalent. The constraint set $Cl_{\mathcal{L}(X)}(C)$ is called the closure of C,
- there exists exactly one closed constraint set, denoted $C_\perp$, that is equivalent to the empty constraint set, $\emptyset$,
- there exists exactly one closed constraint set that is not compatible with any linear constant restriction on Y, namely $C_\top = \mathcal{L}(X)$.
- for each linear constant restriction L on Y there exists a unique closed constraint set $C_L \neq C_\top$ that is compatible with L, but not with any other linear constant restriction on Y. The constraint sets of the form C_L will be called *generalized linear constant restrictions*.

It is easy to verify that $Cl_{\mathcal{L}(X)}$ is a closure operator, which means that $C \subseteq Cl_{\mathcal{L}(X)}(C)$, $Cl_{\mathcal{L}(X)}(C) = Cl_{\mathcal{L}(X)}(Cl_{\mathcal{L}(X)}(C))$ and $C_1 \subseteq C_2$ implies $Cl_{\mathcal{L}(X)}(C_1) \subseteq Cl_{\mathcal{L}(X)}(C_2)$, for all constraint sets C, C_1, C_2.

Lemma 4.5 *Let C_1 and C_2 be two elements of $\mathcal{C}_{\mathcal{L}(X)}$. Then $C_1 \subset C_2$ (resp. $C_1 \subseteq C_2$) iff the set of linear constant restrictions on Y that are compatible with C_2 is a proper subset (resp. subset) of the set of generalized linear constant restrictions on Y that are compatible with C_1.*

If C_1 and C_2 are elements of $\mathcal{C}_{\mathcal{L}(X)}$ we often write $C_1 \prec C_2$ instead of $C_1 \subset C_2$. This emphasizes that we consider a relation on $\mathcal{C}_{\mathcal{L}(X)}$. The following lemma represents one particular instance of a well-known result on closure operators.

Lemma 4.6 *$(\mathcal{C}_{\mathcal{L}(X)}, \prec)$ is a lattice with $0 = C_\perp, 1 = C_\top$. The meet operation on $\mathcal{C}_{\mathcal{L}(X)}$ is given by (set-theoretical) intersection. The join of $C_1, C_2 \in \mathcal{C}_{\mathcal{L}(X)}$ is $Cl_{\mathcal{L}(X)}(C_1 \cup C_2)$.*

In the deterministic combination algorithm that we want to describe below we shall consider subproblems where the pure output problems $\gamma_{2,E} \wedge \gamma_{2,\neq}$ and $\gamma_{2,F} \wedge \gamma_{2,\neq}$ of Algorithm 2 are constrained by means of arbitrary elements of $\mathcal{C}_{\mathcal{L}(X)}$. We shall now give a formal definition of these problems.

Definition 4.7 An $\mathcal{C}_{\mathcal{L}(X)}$*-constrained* E*-unification problem* is a pair (γ_E, C) where γ_E is an elementary E-unification problem, $Var(\gamma_E) \subseteq X$, and $C \in \mathcal{C}_{\mathcal{L}(X)}$. The *size* of (γ_E, C) is the size of γ_E plus $|X|$[6].

It is now convenient to give a reformulation of Algorithm 2 where the output problems are $\mathcal{C}_{\mathcal{L}(X)}$-constrained E-(dis)unification problem with generalized linear constant restrictions. Before we can give a new version of Proposition 4.1 along these lines we have to say what it means to solve such a problem. We shall see that it suffices to give the definition in the case of unification problems.

Definition 4.8 Let γ_E be an elementary E-unification problem with $Var(\gamma_E) \subseteq X$. A *solution* of the *$E$-unification problem with generalized linear constant restriction* (γ_E, C_L) is an E-unifier σ of γ such that

1. for all $u, v \in X$: $\sigma(u) =_E \sigma(v)$ iff $u = v \in C_L$,
2. for all $u : \Delta \in C_L$: $\sigma(u) \in Var$,
3. whenever $u : \Delta, v : \Sigma \in C_L$ and $\sigma(u) \in Var(\sigma(v))$ we have $u < v \in C_L$.

Proposition 4.9 *Let γ_1 be an elementary $(E \cup F)$-disunification problem, given in decomposed form $\gamma_{1,E} \wedge \gamma_{1,F} \wedge \gamma_{1,\neq}$. Let $X = Var(\gamma_1)$. Then γ_1 is solvable if and only if there exists a generalized linear constant restriction C_L on X, where $\gamma_{1,\neq} \subseteq C_L$, such that the E-unification problem with generalized linear constant restriction $(\gamma_{1,E}, C_L)$ has a solution and the F-unification problem with generalized linear constant restriction $(\gamma_{1,F}, C_L)$ has a solution.*

Proof. Assume that γ_1 has a solution. By Proposition 4.1 there exists an output pair $((\gamma_{2,E} \wedge \gamma_{2,\neq}, L), (\gamma_{2,F} \wedge \gamma_{2,\neq}, L))$ of Algorithm 2 such that both the E-disunification problem with linear constant restriction $(\gamma_{2,E} \wedge \gamma_{2,\neq}, L)$ has a solution and the F-disunification problem with linear constant restriction $(\gamma_{2,F} \wedge \gamma_{2,\neq}, L)$ has a solution. It follows from the description of the second step of Algorithm 2 that the generalized linear constant restriction C_L corresponding to L contains all disequations of $\gamma_{2,\neq}$. The given solutions σ_E and σ_F of $(\gamma_{2,E} \wedge \gamma_{2,\neq}, L)$ and $(\gamma_{2,F} \wedge \gamma_{2,\neq}, L)$ can be extended to solutions of $(\gamma_{1,E}, C_L)$ and $(\gamma_{1,F}, C_L)$ just by mapping each variable in X to the value of its representant under σ_E and σ_F respectively.

Conversely assume that there exists a generalized linear constant restriction C_L on X, where $\gamma_{1,\neq} \subseteq C_L$, such that $(\gamma_{1,E}, C_L)$ has a solution σ_E and $(\gamma_{1,F}, C_L)$ has a solution σ_F. The first condition of Definition 4.8 shows that σ_E and σ_F solve all disequations of $\gamma_{2,\neq}$. Let L denote the linear constant restriction that corresponds to C_L. It is unique modulo the set Y of representants for the partition Π which is induced

[6] $|M|$ denotes the cardinality of the set M.

by the equality constraints of C_L. Let τ_E be the function that maps each variable of the form $\sigma_E(u)$ to the representant of u with respect to Π, for all variables $u \in X$ such that $u : \Delta \in C_L$. It follows from the first two conditions of Definition 4.8 that τ_E is well-defined. It may be regarded as a renaming substitution. The composition $\sigma_E \circ \tau_E$ treats representants $y \in Y$ with label Δ as constants, and it is easy to see that it yields a solution of $(\gamma_{2,E} \wedge \gamma_{2,\neq}, L)$. In the same way we obtain a solution for $(\gamma_{2,F} \wedge \gamma_{2,\neq}, L)$. By Proposition 4.1, γ_1 has a solution. □

4.3 $\mathcal{L}$-convex theories and deterministic and polynomial combination

Definition 4.10 A generalized linear constant restriction $C_L \in \mathcal{C}_{\mathcal{L}(X)}$ is called a *faithful* extension of $C \in \mathcal{C}_{\mathcal{L}(X)}$ if $C \preceq C_L$ and if C and C_L have the same set of equality constraints.

Lemma 4.11 *Let $C \neq C_\top$ be a constraint set in $\mathcal{C}_{\mathcal{L}(X)}$. Then C has a faithful extension to a generalized linear constant restriction $C_L \in \mathcal{C}_{\mathcal{L}(X)}$.*

Proof. Since $C \neq C_\top$ there exists a generalized linear constant restriction $C_L \in \mathcal{C}_{\mathcal{L}(X)}$ that extends C. The equality constraint of C and C_L impose partitions Π and Π_1 of Y, and Π is a refinement of Π_1. If some equivalence class of Π_1 contains $n \geq 2$ equivalence classes of Π, then we may split this class in its n subclasses according to Π and define a new ordering where these n classes are consecutive (not changing the order with respect to other classes). Repeating this step, we obtain a new generalized linear constant restriction $C'_L \in \mathcal{C}_{\mathcal{L}(X)}$ which faithfully extends C. □

The following may be considered as the central definition of this section.

Definition 4.12 The constraint set $C_2 \in \mathcal{C}_{\mathcal{L}(X)}$ is a *cover point* for the $\mathcal{C}_{\mathcal{L}(X)}$-constrained E-unification problem (γ_E, C_1) if $C_1 \preceq C_2$ and for each generalized linear constant restriction $C_L \in \mathcal{C}_{\mathcal{L}(X)}$ that faithfully extends C_2 there exists a solution of (γ_E, C_L).
The cover point C_2 of (γ_E, C_1) is called a *universal* cover point for (γ_E, C_1) if $C_2 \preceq C_L$ for all generalized linear constant restrictions $C_L \in \mathcal{C}_{\mathcal{L}(X)}$ where $C_1 \preceq C_L$ and the $\mathcal{C}_{\mathcal{L}(X)}$-constrained E-unification problem (γ_E, C_L) is solvable.

To understand Definition 4.12 it might help to imagine an advocate P for E who negotiates about a suitable generalized linear constant restriction $C_L \in \mathcal{C}_{\mathcal{L}(X)}$ that extends C_1. P wants to agree on some extension C_L such that (γ_E, C_L) is solvable. In this situation P may suggest to continue the negotiation from a universal cover point C_2 for (γ, C_1). With such a suggestion, it is still possible to reach any generalized linear constant restriction C_L which is interesting for E. Moreover, the suggestion is "safe" in the sense that *each* pair (γ_E, C_L) is always solvable *as long as C_L is a faithful extension* of C_2.

Definition 4.13 The equational theory E is *$\mathcal{L}$-convex* iff for every $\mathcal{C}_{\mathcal{L}(X)}$-constrained E-unification problem (γ_E, C_1) there exists a universal cover point with respect to E.

Definition 4.14 The equational theory E is *effectively $\mathcal{L}$-convex* if there exists an algorithm that computes a universal cover point for each $\mathcal{C}_{\mathcal{L}(X)}$-constrained E-unification problem (γ_E, C_1). E is *polynomial $\mathcal{L}$-convex* if there exists an algorithm that is polynomial in the size of (γ_E, C_1).

Let E and F denote two effectively $\mathcal{L}$-convex equational theories over disjoint signatures Σ and Δ respectively. We shall now give a deterministic combination algorithm that may be used to decide solvability of elementary $(E \cup F)$-disunification problems. If both theories are polynomial $\mathcal{L}$-convex, then this decision procedure is polynomial.

Deterministic Combination Algorithm

The *input* of the algorithm is an elementary $(E \cup F)$-disunification problem γ. The remarks on Step 1 of Algorithm 2 show that we may assume that γ is given in decomposed form $\gamma_E \wedge \gamma_F \wedge \gamma_{\neq}$. Let $X := Var(\gamma)$ and $n := |X|$. The algorithm is organized in a series of rounds. Each round has an $\mathcal{C}_{\mathcal{L}(X)}$-constrained ($E$- respectively F-) unification problem (γ_I, C) as input, where $I \in \{E, F\}$ and $C \neq C_\top$. The input for the first round is (γ_E, C_1), where $C_1 := C_\bot$.

Round 1: *We compute a universal cover point $C_2 \in \mathcal{C}_{\mathcal{L}(X)}$ for $(\gamma_E, C_\bot)$ with respect to E. If $C_2 = C_\top$, then we stop with failure. In the other case, (γ_F, C_2) is the input for round 2.*

Round $k \geq 2$: *Assume that the input for this round is (γ_I, C_k), where $I \in \{E, F\}$. We compute a universal cover point $C_{k+1} \in \mathcal{C}_{\mathcal{L}(X)}$ for (γ_I, C_k) with respect to the given equational theory I. The algorithm stops in two cases:*

(i) If $C_{k+1} = C_\top$, then we stop with failure.

(ii) In the other case, if C_{k+1} and C_k have the same set of equality constraints, then we stop with success.

In the remaining case, the $\mathcal{C}_{\mathcal{L}(X)}$-constrained J-unification problem (γ_J, C_{k+1}), where $\{I, J\} = \{E, F\}$, is the input for round $k+1$.

Proposition 4.15 *The Deterministic Combination Algorithm terminates after at most $n+1$ rounds.*

Proof. Beginning with round 2, each non-final round leads to a new identification of variables by means of new equality constraints (Condition (ii)). The maximal number

of such identification steps is $n-1$. Together with the initial round and the final round we obtain a maximum of $n+1$ rounds. □

Proposition 4.16 *The Deterministic Combination Algorithm stops with success if and only if the input conjunction γ has a solution with respect to $E \cup F$.*

Proof. First assume that the $(E \cup F)$-disunification problem γ has a solution with respect to $E \cup F$. By Proposition 4.9 there exists a generalized linear constant restriction $C_L \in \mathcal{C}_{\mathcal{L}(X)}$ where $\gamma_{\neq} \subseteq C_L$ such that the $\mathcal{C}_{\mathcal{L}(X)}$-constrained I-unification problem (γ_I, C_L) has a solution with respect to the theory I, for $I = E, F$. Let $C_1, \ldots, C_k$ $(k \geq 1)$ denote the sequence of universal cover points that are computed in the rounds of the Deterministic Combination Algorithm. Obviously $C_1 = C_\perp \preceq C_L$. Assume that $i < k$ and $C_i \preceq C_L$. Let (γ_I, C_i) be the input for round i, where $I \in \{E, F\}$. The fact that C_{i+1} is a *universal* cover point for (γ_I, C_i) implies that $C_{i+1} \preceq C_L$. It follows that $C_k \preceq C_L$ and the algorithm does not stop with failure. Hence it stops with success.

Now assume that the algorithm stops with success, say in round $l \geq 2$. Suppose that the input of round l is the $\mathcal{C}_{\mathcal{L}(X)}$-constrained I-unification problem (γ_I, C_l), where $I \in \{E, F\}$. Choose an arbitrary *faithful* extension of C_l to a generalized linear constant restriction C_L. Lemma 4.11 shows that such a faithful extension exists. Condition (ii) for round l ensures that C_L is also a faithful extension of C_{l-1}. Since C_l and C_{l-1} are cover points we know that both (γ_I, C_L) and (γ_J, C_L) (where $\{I, J\} = \{E, F\}$) are solvable unification problems with generalized linear constant restrictions. It follows from Proposition 4.9 that the $(E \cup F)$-disunification problem γ is solvable. □

Theorem 4.17 *Let E and F be two polynomial $\mathcal{L}$-convex equational theories over disjoint signatures Σ and Δ. Then solvability of elementary $(E \cup F)$-disunification problems is decidable in polynomial time.*

Proof. Let γ_0 be an elementary $(E \cup F)$-disunification problem. As we mentioned after the description of Algorithm 2, there exists a polynomial algorithm that transforms γ_0 in an elementary $(E \cup F)$-disunification problem γ_1 in decomposed form $\gamma_E \wedge \gamma_F \wedge \gamma_{\neq}$ such that γ_0 is solvable iff γ_1 is solvable, where the cardinality of $X := Var\gamma_1)$ is linear in the size of γ_0. The problem γ_1 may be used as input for the Deterministic Combination Algorithm. Under the given assumption on E and F it follows that the computations of each round of the Deterministic Combination Algorithm need polynomial time (in the size of the original input problem γ_0). Now the theorem follows from Propositions 4.15 and 4.16. □

4.4 Applications

In view of the strong intractability results of Section 3 it should not be surprising that it turns out to be difficult to find natural classes of equational theories that are $\mathcal{L}$-convex. The following theorem gives one such class.

Theorem 4.18 *Let E be a unitary regular collapse-free equational theory. Then E is $\mathcal{L}$-convex.*

Proof. Let E be a unitary regular collapse-free equational theory and let (γ, C_1) be an $\mathcal{C}_{\mathcal{L}(X)}$-constrained E-unification problem. We have to show that there exists a universal cover point for (γ, C_1) with respect to E.

In the first, trivial case, there does not exist a generalized linear constant restriction $C_L \in \mathcal{C}_{\mathcal{L}(X)}$ such that $C_1 \preceq C_L$ and (γ, C_L) is solvable. In this case, $C_\top$ is a universal cover point for (γ, C_1).

Now assume that there exists a generalized linear constant restriction $C_L \in \mathcal{C}_{\mathcal{L}(X)}$ such that $C_1 \preceq C_L$ and (γ, C_L) has a solution σ. The first condition of Definition 4.8 shows that σ solves $\gamma_1 := \gamma \wedge \bigwedge_{u=v \in C_1} u = v$. Let μ be a most general E-unifier of γ_1, let

$$\begin{aligned} C_= &:= \{u = v \mid u, v \in X, \mu(u) =_E \mu(v)\} \\ C_\Sigma &:= \{u : \Sigma \mid u \in X, \mu(u) \notin \mathit{Var}\} \\ C_\Rightarrow &:= \{u : \Delta \Rightarrow u < v \mid u, v \in X, \mu(u) \in \mathit{Var}(\mu(v)), \mu(v) \notin \mathit{Var}\} \end{aligned}$$

and let C denote the closure of $C_1 \cup C_= \cup C_\Sigma \cup C_\Rightarrow$. It is easy to see that (*) all equations of C are in $C_=$.

We claim that C is a cover point for γ. Let C'_L be a generalized linear constant restriction on X that faithfully extends C. We show that μ is a solution of (γ, C'_L): Clearly μ solves γ. Moreover, since C'_L faithfully extends C it follows from (*) that (1) $\mu(u) =_E \mu(v)$ iff $u = v \in C'_L$. It remains to prove that the remaining two conditions of Definition 4.8 are satisfied. Let $u, v \in X$. If $u : \Delta \in C'_L$, then $u : \Sigma \notin C_\Sigma \subseteq C'_L$ and $\mu(u) \in \mathit{Var}$. Now assume that $u : \Delta, v : \Sigma \in C'_L$ and $\mu(u) \in \mathit{Var}(\mu(v))$. Then $\mu(v)$ cannot be a variable, otherwise we would have $\mu(u) = \mu(v)$ and $u = v \in C'_L$ which is impossible since both variables have distinct labels. The definition of $C_\Rightarrow$ implies that $u : \Delta \Rightarrow u < v \in C \subseteq C'_L$. Hence $u < v \in C'_L$ since C'_L is closed.

It remains to prove that C is a universal cover point for (γ, C_1). Let C_L, σ, γ_1 and μ as above. Since σ is an E-unifier of γ_1 there exists a substitution λ such that $\sigma(v) =_E \lambda(\mu(v))$ for all $v \in X$. In order to show that $C \preceq C_L$ it suffices to prove that $(C_= \cup C_\Sigma \cup C_\Rightarrow) \subseteq C_L$ since C_L is closed and $C_1 \subseteq C_L$.

1. Let $u = v \in C_=$. Then $\mu(u) =_E \mu(v)$ and $\sigma(u) =_E \lambda(\mu(u)) =_E \lambda(\mu(v)) =_E \sigma(v)$. The first condition of Definition 4.8 shows that $u = v \in C_L$.

2. For $u : \Sigma \in C_\Sigma$ we have $\mu(u) \notin Var$. Since $\sigma(u) =_E \lambda(\mu(u))$ and since E is collapse-free it follows from Remark 2.1 that $\sigma(u) \notin X$. Condition 2 of Definition 4.8 shows that $u : \Delta \notin C_L$, therefore $u : \Sigma \in C_L$ since C_L is a generalized linear constant restriction.

3. Consider a constraint $u : \Delta \Rightarrow u < v \in C_\Rightarrow$. In order to show that $u : \Delta \Rightarrow u < v \in C_L$ we may assume that

$$u : \Delta \in C_L$$

and show that $u < v \in C_L$. From the definition of $C_\Rightarrow$ it follows that $\mu(u) \in Var(\mu(v))$ and $\mu(v) \notin Var$. Now $\lambda(\mu(v)) \notin Var$ and the first observation of Remark 2.1 implies that $\lambda(\mu(v)) =_E \sigma(v) \notin Var$. It follows that $v : \Delta \notin C_L$ and thus

$$v : \Sigma \in C_L.$$

Since $u : \Delta \in C_L$, Condition 3 of Definition 4.8 shows that $\sigma(u) \in Var$. But $\sigma(u) =_E \lambda(\mu(u))$. Since E is collapse-free, the first observation of Remark 2.1 implies $\sigma(u) = \lambda(\mu(u)) \in Var$. Since $\mu(u) \in Var(\mu(v))$ we see now that $\sigma(u) = \lambda(\mu(u)) \in Var(\lambda(\mu(v)))$. Since E is regular and $\sigma(v) =_E \lambda(\mu(v))$, the second observation of Remark 2.1 shows that $\sigma(u) \in Var(\sigma(v))$. Condition 3 of Definition 4.8 implies that $u < v \in C_L$. □

Corollary 4.19 *Let E be a unitary, regular and collapse-free equational theory. Suppose there exists a unification algorithm that computes a most general unifier μ for each solvable E-unification problem with constants, γ, stopping with failure for unsolvable problems. Then E is effectively $\mathcal{L}$-convex. If this algorithm is polynomial, then E is polynomial $\mathcal{L}$-convex.*

Proof. Let (γ, C_1) be a $\mathcal{C}_{\mathcal{L}(X)}$-constrained E-unification problem as considered in the previous proof. Let μ be a most general E-unifier for $\gamma_1 := \gamma \wedge \bigwedge_{u=v \in C_1} u = v$. Under the given assumptions there exists (polynomial) algorithms to compute μ, and to decide if $\mu(u) =_E \mu(v)$, for given variables $u, v \in X$. The previous proof shows that we may effectively compute a universal cover point C_2 for each $\mathcal{L}(X)$-constrained E-unification problem (γ, C_1). If the E-unification algorithm is polynomial, then obviously the set $C_1 \cup C_= \cup C_\Sigma \cup C_\Rightarrow$ mentioned in the previous proof can be computed in time polynomial in the size of (γ, C_1). It is easy to see that the $\mathcal{L}(X)$-closure of this set, i.e., the universal cover point for (γ, C_1), can be computed in time polynomial in the size of (γ, C_1). □

Theorem 4.20 *Let E and F be a unitary, regular and collapse-free equational theories over disjoint signatures Σ and Δ. Suppose there exists a polynomial algorithm that computes a most general unifier μ for each solvable elementary E- resp. F-unification problem with constants, γ, stopping with failure for unsolvable problems. Then there exists a polynomial algorithm to decide solvability of elementary $(E \cup F)$-disunification problems.*

Proof. This follows from Theorem 4.17 and Corollary 4.19. □

The result of Theorem 4.20 can be lifted to general $(E \cup F)$-disunification problems, just by combining $E \cup F$ with the free theory. Let us mention one concrete application of Theorem 4.20.

Corollary 4.21 *Let*

$$\begin{aligned} D_L &:= \{f(x, g(y, z)) = g(f(x, y), f(x, z))\} \\ D_R &:= \{f(g(y, z), z) = g(f(x, z), f(y, z))\} \end{aligned}$$

denote the theories of left and right ditributivity respectively. Then solvability of general D_L (resp. D_R) disunification problems is decidable in polynomial time.

Proof. The theories of left respectively right distributivity are known to be unitary, regular and collapse-free, and there exists a quadratic unification algorithm for unification with constants in both cases (see [BS94], chapter 5.) Obviously the free (empty) theory for a given set of function symbols satisfies these requirements as well. Hence the result follows from Theorem 4.20. □

5 Conclusion

In this paper we have tried to get some insights into the borderline between tractable and intractable instances of combination problems for equational unification and disunification. We have introduced a criterion that characterizes a large class $\mathcal{K}$ of equational theories E where general E-unification is always NP-hard. The most important question arising from this result has already been mentioned before: is it possible to interpret these results in the sense that there cannot be a polynomial method for reducing general E-unification problems to pure E-unification problems and pure free (syntactic) unification problems, for *any* given theory $E \in \mathcal{K}$? We strongly believe that the answer is yes, in principle, though we indicated in Section 3 that the formulation of the problem is vague. In order to obtain a partial answer we introduced the concept of a polynomial optimization of the combination algorithm given in [BS92]. We showed that there cannot be a polynomial optimization for general E-unification, for $E \in \mathcal{K}$.

We have shown that general E-unification is NP-hard for all regular equational theories E that contain an associative or a commutative function symbol. It is interesting to note that the proof for the intractability of the counting problem for general AG unification given in M. Hermann and P.G. Kolaitis [HK96] heavily depends on the presence of an associative and commutative function symbol in the theory. One may ask if their result can also be generalized to arbitrary regular equational theories with associative or commutative function symbols.

In the second part of the paper we looked at properties of equational theories that guarantee that a deterministic and polynomial combination for disunification algorithms is possible. We introduced the notion of an $\mathcal{L}$-convex equational theory, refering to a constraint language $\mathcal{L}$ for describing linear constant restrictions. It was shown that $\mathcal{L}$-convexity yields the key for results on deterministic and polynomial combination of (dis)unification algorithms, for equational theories over disjoint signatures. It should be stressed that the notion of $\mathcal{L}$-convexity is generic in the sense that variants of this notion can be introduced for similar constraint languages that describe linear constant restrictions. The particular constraint language that we used in this paper is tuned to the results on unitary regular and collapse-free theories, and it might be possible to obtain further results on deterministic and polynomial combination, using other constraint languages.

The notion of $\mathcal{L}$-convexity can be generalized in a straightforward way to the class of simply combinable (SC-) structures introduced in [BS95b]. This class of structures properly extends the class of all free structures and contains many domains that are relevant for constraint programming. For example, the algebra of rational trees ([Col84, Mah88]), the structure of rational feature trees ([APS94]), and the structure of rational feature trees with arity ([ST94]) are (non-free) $\mathcal{L}$-convex SC-structures. This follows from an analysis of the solved form systems for conjunctions of atomic constraints that exist for these structures. It is possible to obtain polynomial combination algorithms for procedures that decide solvability of conjunctions of atomic constraints on this basis, if we use the free amalgamated product ([BS95b]) of the component structures as the combined solution domain.

As a matter of fact, the tractability results given in Section 4 are mainly of theoretical interest. On the other hand it seems possible to extract from the discussion some hints on possible optimizations of Algorithm 1 for arbitrary combinations of disjoint equational theories. Constraint languages similar to the language $\mathcal{L}$ discussed in Section 4 are used in [KR96] in an optimized version of Algorithm 1. In these approach, constraints are used to eliminate at least part of the non-determinism of the combination scheme. However, conditional constraints of the form $u : \Delta \Rightarrow u < v$—which have been crucial for obtaining our tractability results—have not been used in this context so far. It seems worthwile to study the effect of their integration.

As we mentioned in the introduction, the notion of $\mathcal{L}$-convexity was inspired by the notion of a "convex theory" introduced by in [Op80]. Oppen considered combinations of first order theories $\mathcal{T}_1$, and $\mathcal{T}_2$ over disjoint signatures. These theories are axiomatized by a set of quantifier-free formulae—implicitly, axioms are universally quantified. The algorithmic problem is to decide *validity* of mixed[7] formulae in the union $\mathcal{T}_1 \cup \mathcal{T}_2$ of the two theories. This problem is reduced to the problem of deciding *satisfiability* of a conjunction γ of mixed (positive or negated) literals in $\mathcal{T}_1 \cup \mathcal{T}_2$.[8] The non-deterministic

[7] A formula is mixed if it uses non-logical symbols from both theories.

[8] Note that any model of $\mathcal{T}_1 \cup \mathcal{T}_2$ can be used to validate satisfiability of γ. In contrast, solvability of unification problems refers to a fixed algebra!

combination method suggested in [Op80] consists essentially of the first two steps of Algorithm 1 described in Section 2: the mixed conjunction γ is decomposed into two pure subparts γ_1 and γ_2. In the second step, a set of equations and disequations between the variables in the problem is guessed, yielding the output formulae γ_1' and γ_2' that are now evaluated independently with the satisfiability checkers for the single theories.

Given a quantifier-free theory $\mathcal{T}$, a formula φ is called *convex* in [Op80] if it never entails a disjunction of equalities between variables without entailing any of the equalities alone. Here entailment is with respect to $\mathcal{T}$. The theory $\mathcal{T}$ is called convex if every conjunction of literals in the language of $\mathcal{T}$ is convex[9]. Oppen shows that a deterministic combination algorithm exists for formulae in disjunctive normalform if both component theories are convex.

Despite of the fact that the component theories considered in [Op80] are more general than the equational theories considered here, the comparison between the two decomposition algorithms shows that combination of (dis)unification algorithms is the more difficult problem. While Oppen uses just one non-deterministic step, there are three such steps in Algorithm 1. This difference is reflected in the two notions of convexity. The common idea behind both notions is, roughly, to formulate a condition that guarantees a backtrack-free search for an output pair of the non-deterministic combination procedure in the style of a negotiation between the two theories. In the case of [Op80], just entailed equations and disequations between variables have to be communicated between the theories. In our case, attention has to be payed also to labelling and order information.

References

[APS94] H. Ait-Kaci, A. Podelski, and G. Smolka, "A feature-based constraint system for logic programming with entailment," *Theoretical Comp. Science* **122**, 1994, pp.263–283.

[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.

[BS95a] F. Baader, K.U. Schulz, "Combination techniques and decision problems for disunification," *Theoretical Computer Science* **142** (1995), pp. 229-255.

[BS95b] F. Baader, K.U. Schulz, "On the combination of symbolic constraints, solution domains, and constraint solvers," In *Principles and Practice of Constraint Programming - CP95,* U. Montanari, F. Rossi (Eds.). Springer LNCS 976, 1995, pp. 380-398.

[9] It is interesting to note that for theories where all atomic formulae are equations the notion of convexity just expresses socalled "independence of disequations", a property which is often discussed in the field of constraint logic programming. See, e.g., [Col84, ST94].

[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.

[BT96] F. Baader, C. Tinelli, "A New Approach for Combining Decision Procedures for the Word Problem and Its Connection to the Nelson-Oppen Combination Method," research paper, available under http://www.ps.uni-sb.de/ccl/publications/1996.html.

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

[Bür88] H.J. Bürckert, "Solving Disequations in Equational Theories," *Proceedings of the 9th International Conference on Automated Deduction, Argonne*, LNCS 310, Springer 1988.

[Col84] A. Colmerauer, "Equations and inequations on finite and infinite trees," in: *Proc. 2nd Int. Conf. on Fifth Generation Computer Systems*, 1984, pp. 85-99.

[Col90] A. Colmerauer, "An introduction to PROLOG III," *C. ACM* **33**, 1990, pp.69–90.

[Com91] H. Comon, "Disunification: a Survey," in J.-L. Lassez, G. Plotkin (editors), *Computational Logic*, MIT Press, 1991.

[DKR94] E. Domenjoud, F. Klay, R. Ringeissen "Combination Techniques for Non-Disjoint Equational Theories," in: *Proc. CADE-12*, LNAI 814, 1994, pp. 267-281.

[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.

[KN91] D. Kapur, P. Narendran, "Complexity of Unification Problems with Associative-Commutative Operators," *J. Automated Reasoning* **9**, 1992, pp. 261-288.

[KS96] S. Kepser, K.U. Schulz, "Combination of constraint systems II: Rational amalgamation," In *Principles and Practice of Constraint Programming - CP96,* E.C. Freuder (Ed.), Springer LNCS 1118, 1996, pp. 282-296.

[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'algorithms d'unification dans les théories équationelles," Thése d'Etat, Université de Nancy 1, France, 1985.

[Mah88] M.J. Maher, "Complete axiomatizations of the algebras of finite, rational and infinite trees," in: *Proceedings of Third Annual Symposium on Logic in Computer Science, LICS'88*, pp.348–357, Edinburgh, Scotland, 1988. IEEE Computer Society.

[NO79] C.G. Nelson, D.C. Oppen, "Simplification by cooperating decision procedures," *ACM Trans. Programming Languages and Systems* **2** (1) (1979).

[Ni89] T. Nipkow, "Combining Matching Algorithms: The Regular Case," in: *Proc. RTA-3*, N. Dershowitz (Ed.), Springer LNCS 335, 1989, pp. 343-358.

[Op80] D.C. Oppen, "Complexity, Convexity and Combination of Theories," *Theoretical Computer Science* **12** (1980), pp. 291-302.

[Ri96] C. Ringeissen, "Cooperation of Decision Procedures for the Satisfiability Problem," in *Proceedings of the 1st International Workshop "Frontiers of Combining Systems", Munich, Germany*, F. Baader, K.U. Schulz (Eds.), Applied Logic, Kluwer, 1996, to appear.

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

[ST94] G. Smolka, R. Treinen, "Records for Logic Programming," *Journal of Logic Programming* **18**(3), 1994, pp.229-258.

[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.

[TH96] C. Tinelli, M. Harandi, "A New Correctness Proof of the Nelson-Oppen Combination Procedure," in *Proceedings of the 1st International Workshop "Frontiers of Combining Systems", Munich, Germany*, F. Baader, K.U. Schulz (Eds.), Applied Logic, Kluwer, 1996, to appear.

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