Non-Disjoint Combination with Forward-Closed Theories (Extended Abstract)

Serdar Erbatur[1], Andrew M. Marshall[2], and Christophe Ringeissen[3]

[1] Ludwig-Maximilians-Universität, München (Germany)
serdar.erbatur@ifi.lmu.de
[2] University of Mary Washington (USA)
marshall@umw.edu
[3] LORIA – INRIA Nancy-Grand Est (France)
Christophe.Ringeissen@loria.fr

1 Introduction

Equational unification is the problem of solving equations in structures of terms modulo an equational theory. In general, equational unification is undecidable, but specialized techniques have been developed to solve the problem for particular classes of equational theories.

When the equational theory has the Finite Variant Property (FVP) [4], equational unification reduces to syntactic unification via the computation of finitely many variants of terms. Nowadays, equational theories with the FVP have attracted a considerable interest, especially for their applications in the analysis of security protocols [2, 5, 10].

When the equational theory is given by a convergent term rewrite system, the concept of narrowing is a generalization of rewriting, where the matching process is replaced by some (syntactic) unification problem. Narrowing is complete for equational unification, but it terminates only in some very particular cases. Hence, a particular narrowing strategy, called folding variant narrowing, is complete and terminating for any equational theory with the FVP [10].

When the equational theory is *syntactic*, it is possible to apply a mutation-based unification procedure [11]. However, being syntactic is not a sufficient condition for a theory to admit a terminating mutation-based unification procedure. In the particular case of shallow theories, there exists a terminating mutation-based unification procedure [3].

Another important scenario is given by an equational theory defined as a union of component theories. To solve this case, it's natural to proceed in a modular way and there are terminating and complete combination procedures for disjoint unions of theories [16]. These combination procedures can be extended to some particular non-disjoint unions of theories, but it is difficult to identify cases where these procedures terminate [6, 15].

In this paper we investigate the unification problem in equational theories involving forward-closed convergent term rewrite systems [2]. In the class of forward-closed theories, unification is decidable and finitary since any convergent term rewrite system has a finite forward closure if and only if it has the FVP [2]. Furthermore, forward-closed theories are syntactic theories admitting a terminating mutation-based unification procedure. We first demonstrate this result by showing that a mutation-based unification algorithm, originally developed for equational theories saturated by paramodulation [13], remains sound and complete for forward-closed theories. Building on this result we use the new mutation-based algorithm to develop an algorithm that solves the unification problem in unions of forward-closed theories with non-disjoint theories. The resulting algorithm can be viewed as a terminating instance of a procedure initiated for hierarchical combination [7].

2 Preliminaries

We assume the reader is familiar with equational unification and term rewriting systems [1]. An axiom $l = r$ is *regular* (also called *variable-preserving*) if $Var(l) = Var(r)$. An axiom $l = r$ is *linear* (resp., *collapse-free*) if l and r are linear (resp. non-variable terms). An equational theory is *regular* (resp., linear/collapse-free) if all its axioms are regular (resp., linear/collapse-free). A theory E is *syntactic* if it has a finite *resolvent presentation* S, that is a presentation S such that each equality $t =_E u$ has an equational proof $t \leftrightarrow_S^* u$ with at most one step $\leftrightarrow_S$ applied at the root position. In the following, we often use tuples of terms, say $\bar{u} = (u_1, \ldots, u_n)$.

An E-unification problem is a set of Σ-equations, $G = \{s_1 =^? t_1, \ldots, s_n =^? t_n\}$, or equivalently a conjunction of Σ-equations. The set of variables in G is denoted by $Var(G)$. A solution to G, called an *E-unifier*, is a substitution σ such that $s_i\sigma =_E t_i\sigma$ for all $1 \leq i \leq n$. A substitution σ is *more general modulo* E than θ on a set of variables V, denoted as $\sigma \leq_E^V \theta$, if there is a substitution τ such that $x\sigma\tau =_E x\theta$ for all $x \in V$. A *Complete Set of E-Unifiers* of G is a set of substitutions denoted by $CSU_E(G)$ such that each $\sigma \in CSU_E(G)$ is an E-unifier of G, and for each E-unifier θ of G, there exists $\sigma \in CSU_E(G)$ such that $\sigma \leq_E^{Var(G)} \theta$. A set of equations $G = \{x_1 =^? t_1, \ \ldots, \ x_n =^? t_n\}$ is said to be in *tree solved form* if each x_i is a variable occurring once in G. Given an idempotent substitution $\sigma = \{x_1 \mapsto t_1, \ \ldots, \ x_n \mapsto t_n\}$ (such that $\sigma\sigma = \sigma$), $\hat{\sigma}$ denotes the corresponding tree solved form. A set of equations is said to be in *dag solved form* if they can be arranged as a list $x_1 =^? t_1, \ \ldots, \ x_n =^? t_n$ where (a) each left-hand side x_i is a distinct variable, and (b) $\forall\, 1 \leq i \leq j \leq n$: x_i does not occur in t_j. A set of equations $\{x_1 =^? t_1, \ \ldots, \ x_n =^? t_n\}$ is a *cycle* if $x_{i+1} \in Var(t_i)$, at least one t_i is not a variable, and $x_1 \in Var(t_n)$.

Let E be an equational Σ-theory such that $\Sigma = \Sigma_1 \cup \Sigma_2$ and $\Sigma_1 \cap \Sigma_2 = \emptyset$. Σ_i-terms (including the variables) and Σ_i-equations (including the equations between variables) are called *i-pure*. An E-unification problem is in *separate form* if it is a conjunction $G_1 \wedge G_2$, where G_i is a conjunction of Σ_i-equations for $i = 1, 2$. A term t is called a Σ_i*-rooted* term if its root symbol is in Σ_i. An *alien* subterm of a Σ_i-rooted term t is a Σ_j-rooted subterm s ($i \neq j$) such that all superterms of s are Σ_i-rooted. A TRS is Σ_i*-rooted* if all its left-hand sides and right-hand sides are Σ_i-rooted. We define *general E-unification* as the unification problem in the equational theory obtained by extending E with arbitrary free function symbols.

Let us now define the notion of forward-closure [2]. For a given convergent TRS R, assume a reduction ordering $<$ such that $r < l$ for any $l \to r \in R$ and $<$ is total on ground terms. Since (rewrite) rules are multisets of two terms, the multiset extension of $<$ leads to an ordering on rules, also denoted by $<$, which is total on ground instances of rules. A rule ρ is *strictly redundant in* R if any ground instance $\rho\sigma$ of ρ follows from ground instances of R that are strictly smaller w.r.t $<$ than $\rho\sigma$. A rule ρ is *redundant in* R if ρ is strictly redundant in R or ρ is an instance of some rule in R. A TRS R is *forward-closed* if any **Forward** inference (cf. Figure 1) with premises in R generates a rule which is redundant in R.

Forward	$g \to d[l'] \quad l \to r \ \vdash \ (g \to d[r])\sigma$ where $\sigma = mgu(l, l')$ and l' is not a variable.

Figure 1: Forward inference

3 Unification in Forward-Closed Rewrite Systems

We present a rule-based unification procedure for any forward-closed TRS. We basically reuse the unification procedure initially developed for any equational theory saturated by paramodulation [13]. This procedure implements *Basic Syntactic Mutation* (BSM) by extending syntactic unification with some additional mutation rules (rules whose names include **Mut** in Figures 2 and 3) and cycle breaking rules (rules whose names include **Cycle** in Figure 3). These additional rules are applied in a *don't know* non-deterministic way. Thus, the resulting BSM unification procedure is similar to the mutation-based unification procedures designed for syntactic theories [12, 14]. However, these mutation-based procedures are not terminating, whereas the BSM unification procedure always terminates. To get termination, the BSM rules depicted in Figures 2 and 3 make use of boxed terms. This particular annotation of terms works as follows: Subterms of boxed terms are also boxed, terms boxed in the premises of an inference rule remain boxed in the conclusion, and when the "box" status of a term is not explicitly given in an inference rule, it can be either boxed or unboxed. Given a unification problem G and an R-normalized substitution σ, (G, σ) is said to be *R-normalized* if $t\sigma$ is R-normalized whenever t is boxed in G. For a forward-closed convergent TRS R, the set of equalities S used in Figures 2 and 3 is defined as being equal to $RHS(R) = R^{=} \cup \{l\sigma = g\sigma \mid l \to r \in R, g \to d \in R, \sigma \in mgu(r, d), l\sigma \neq g\sigma\}$, where $R^{=} = \{l = r \mid l \to r \in R\}$.

Dec $\{f(\bar{u}) = f(\bar{v})\} \cup G \ \vdash \ \{\bar{u} = \bar{v}\} \cup G$

Mut $\{f(\bar{u}) = g(\bar{v})\} \cup G \ \vdash \ \{\bar{u} = \boxed{\bar{s}}, \boxed{\bar{t}} = \bar{v}\} \cup G$
where $f(\bar{u})$ is unboxed and $f(\bar{s}) = g(\bar{t}) \in S$.

Imit $\bigcup_i \{x = f(\bar{v}_i)\} \cup G \ \vdash \ \{x = \boxed{f(\bar{y})}\} \cup \bigcup_i \{\bar{y} = \bar{v}_i\} \cup G$
where $i > 1$ and there are no more equations $x = f(\dots)$ in G.

MutImit $\{x = f(\bar{u}), x = g(\bar{v})\} \cup G \ \vdash \ \{x = f(\bar{y}), \bar{y} = \boxed{\bar{s}}, \boxed{\bar{s}} = \bar{u}, \boxed{\bar{t}} = \bar{v}\} \cup G$
where $f(\bar{s}) = g(\bar{t}) \in S$, and

1. if $f(\bar{u})$ is boxed, $g(\bar{v})$ is unboxed, then $f(\bar{y})$ is boxed;
2. if $f(\bar{u})$ and $g(\bar{v})$ are unboxed, then $f(\bar{y})$ is unboxed.

Coalesce $\{x = y\} \cup G \ \vdash \ \{x = y\} \cup (G\{x \mapsto y\})$
where x and y are distinct variables occurring both in G.

Figure 2: BSM rules

Given an R-unification problem G, the BSM unification procedure works as follows: apply the BSM rules (Figures 2 and 3) on G until reaching normal forms. The procedure then only returns those sets of equations which are in dag solved form.

Theorem 1. *If R is a forward-closed convergent TRS, then the BSM unification procedure provides an R-unification algorithm.*

4 Forward-Closed Combination

In previous papers [7, 8], we have studied a form of non-disjoint combination, called hierarchical combination, which is defined as a convergent TRS R_1 combined with a base theory E_2. The TRS R_1 must satisfy some properties to ensure that $E = R_1 \cup E_2$ is a conservative extension of E_2. We are interested in combined theories E where it is possible to reduce any E-equality between two terms ($s =_E t$) into the E_2-equality of their R_1-normal forms ($s\downarrow_{R_1} =_{E_2} t\downarrow_{R_1}$). Below, we assume that R_1 is forward-closed.

VarMut $\{f(\bar{u}) = v\} \cup G \vdash \{\bar{u} = \boxed{\bar{s}}, y = v\} \cup G$
where $f(\bar{u})$ is unboxed, $f(\bar{s}) = y \in S$ with a variable y, and if v is a variable, then there is another equation $v = t \in G$ with a non-variable term t, or $v = f(\bar{u})$ occurs in a cycle.

ImitCycle $\{x = f(\bar{v})\} \cup G \vdash \{x = \boxed{f(\bar{y})}, \bar{y} = \bar{v}\} \cup G$
if no other rule applies among **VarMut** and those in Figure 2, $f(\bar{v})$ is unboxed and $x = f(\bar{v})$ occurs in a cycle.

MutImitCycle $\{x = f(\bar{v})\} \cup G \vdash \{x = \boxed{g(\bar{t})}, \boxed{\bar{s}} = \bar{v}\} \cup G$
where $f(\bar{s}) = g(\bar{t}) \in S$, if no other rule applies among **VarMut** and those in Figure 2, $f(\bar{v})$ is unboxed and $x = f(\bar{v})$ occurs in a cycle.

Figure 3: Additional BSM rules for a subterm collapsing theory

Definition 1. *A* forward-closed combination *(FC-combination, for short) is a pair (E_1, E_2) such that: $\Sigma_1 \cap \Sigma_2 = \emptyset$; E_1 is an equational $\Sigma_1 \cup \Sigma_2$-theory given by a forward-closed convergent TRS R_1 such that its left-hand sides are linear Σ_1-terms; E_2 is a regular and collapse-free equational Σ_2-theory. An FC-combination (E_1, E_2) is said to be* layer-preserving *if R_1 is Σ_1-rooted and regular.*

Example 1. *Consider $R_1 = \{exp(a, y) \times exp(a, z) \to exp(a, y + z)\}$ with $\Sigma_1 = \{a, exp, \times\}$ and $\Sigma_2 = \{+\}$. Then, an FC-combination can be obtained by considering any regular and collapse-free Σ_2-theory E_2, such as Commutativity or Associativity-Commutativity.*

From now on, we assume an E_2-unification algorithm, a layer-preserving FC-combination (E_1, E_2) such that E_1 is given by a forward-closed convergent TRS R_1, and a combined theory $E = E_1 \cup E_2$. An E_1-unification algorithm is provided by BSM (cf. Section 3), where **VarMut** does not apply since E_1 is collapse-free.

5 Unification Procedure for Forward-Closed Combination

We study how the BSM unification procedure can be combined with an E_2-unification algorithm to solve any E-unification problem. Remember that BSM is parameterized by a set of equalities S used in the mutation rules. In order to transform Σ_1-rooted equations, the definition of S can be lifted to take into account E_2. Hence, we can compute

$$S = RHS_{E_2}(R_1) = R_1^{=} \cup \{l\sigma = g\sigma \mid l \to r \in R_1, g \to d \in R_1, \sigma \in CSU_{E_2}(r = d), l\sigma \neq g\sigma\}$$

Lemma 1. *Assume $S = RHS_{E_2}(R_1)$. For each Σ_1-rooted equality $u =_E v$, one of the following is true. Either, $u = f(\bar{u})$, $v = f(\bar{v})$ and $\bar{u} =_E \bar{v}$. Or, $u = f(\bar{u})$, $v = g(\bar{v})$ and there are $f(\bar{s}) = g(\bar{t}) \in S$ and an R_1-normalized substitution σ such that $\bar{u} =_E \bar{s}\sigma$, $\bar{v} =_E \bar{t}\sigma$ where $\bar{s}\sigma$ and $\bar{t}\sigma$ are R_1-normalized.*

Consider the inference system BSC defined by the set of rules in Figure 4 plus the BSM rules in Figures 2 and 3, parameterized by $S = RHS_{E_2}(R_1)$, with the restriction that BSM rules are applied only if the input is in separate form, and by matching only Σ_1-equations.

Lemma 2. *Given an E-unification problem G as input, the repeated application of BSC rules always terminates and computes a set of normal forms in separate form denoted by* $\texttt{BSC}(G)$.

Following Lemma 2, the BSC unification procedure works as follows: apply the BSC rules on a given E-unification problem G until reaching normal forms, and return all the dag solved forms in $\mathtt{BSC}(G)$. The completeness of the BSC unification procedure relies on the lemma given below. First, we state that an E_2-unification algorithm can be reused without loss of completeness to E-unify any conjunction of Σ_2-equations. This is a classical result, already used in hierarchical combination [7], which can be easily lifted to FC-combination.

VA $\{s = t[u]\} \cup G \vdash \{s = t[x], x = u\} \cup G$
where u is an alien subterm of t, x is a fresh variable, and u is boxed iff $t[u]$ is boxed.

IE $\{s = t\} \cup G \vdash \{x = s, x = t\} \cup G$
where s is a non-variable Σ_1-term, t is a non-variable Σ_2-term and x is a fresh variable.

Solve2 $G_1 \wedge G_2 \vdash \bigvee_{\sigma_2 \in CSU_{E_2}(G_2)} G_1 \wedge \hat{\sigma}_2$
if G_2 is E_2-unifiable and unsolved, and no other rule applies.

Figure 4: Additional Rules for the combination with E_2

Lemma 3 (Completeness)**.** *If σ is an E-unifier of G, (G, σ) is R_1-normalized, and G is not a separate form in dag solved form, then there exist some G' and a substitution σ' such that G' is obtained from G by applying some BSC rule, σ' is an E-unifier of G', (G', σ') is R_1-normalized, and $\sigma' \leq_E^{Var(G)} \sigma$.*

According to Lemmas 2 and 3, we can conclude that BSC leads to a terminating and complete E-unification procedure: given an E-unification problem, the set of dag solved forms in $\mathtt{BSC}(G)$ provides a $CSU_E(G)$. Then, this E-unification algorithm can be lifted to a general E-unification algorithm by assuming that E_2 includes the free symbols.

Theorem 2. *Given any layer-preserving FC-combination (E_1, E_2) and an E_2-unification algorithm, BSC provides a general $E_1 \cup E_2$-unification algorithm.*

For sake of simplicity, we restrict us in this short paper to layer-preserving FC-combinations, where all conflicts between component theories have no solution just like in disjoint unions of regular and collapse-free theories [17]. The general case of arbitrary FC-combinations, where the TRS R_1 may contain non-regular or collapse axioms, will be addressed in a full version [9]. In the general case, we will investigate the possibility of considering a single mutation rule dedicated to equations between a variable and a Σ_1-rooted term.

6 Connection with the Finite Variant Property

The computation of finite variants [4,10] is another way to reduce any unification problem modulo $E = R_1 \cup E_2$ into some E_2-unification problems with free function symbols. Thus, there exists an interesting connection between the finite variant property and FC-combinations. It can be shown that a brute force method can be used to solve E-unification: compute all the finitely many R_1-variants of an input E-unification problem, and solve them by general E_2-unification, usually implemented by combining E_2-unification and syntactic unification. This is a highly non-deterministic method. The brute force approach of computing all variants can be prohibitive due to the possible large number of variants and thus it is desirable to have alternatives to that approach. From our point of view, the approach described in Section 5 provides an interesting alternative where all the non-determinism is managed inside the combination procedure, when mutation rules and **Solve2** have to be applied.

References

[1] F. Baader and T. Nipkow. *Term rewriting and all that.* Cambridge University Press, New York, NY, USA, 1998.

[2] C. Bouchard, K. A. Gero, C. Lynch, and P. Narendran. On forward closure and the finite variant property. In P. Fontaine, C. Ringeissen, and R. A. Schmidt, editors, *Frontiers of Combining Systems*, volume 8152 of *Lecture Notes in Computer Science*, pages 327–342. Springer Berlin Heidelberg, 2013.

[3] H. Comon, M. Haberstrau, and J. Jouannaud. Syntacticness, cycle-syntacticness, and shallow theories. *Inf. Comput.*, 111(1):154–191, 1994.

[4] H. Comon-Lundh and S. Delaune. The finite variant property: How to get rid of some algebraic properties. In J. Giesl, editor, *Rewriting Techniques and Applications*, volume 3467 of *Lecture Notes in Computer Science*, pages 294–307. Springer, 2005.

[5] Ştefan Ciobâcă, S. Delaune, and S. Kremer. Computing knowledge in security protocols under convergent equational theories. *J. Autom. Reasoning*, 48(2):219–262, 2012.

[6] E. Domenjoud, F. Klay, and C. Ringeissen. Combination techniques for non-disjoint equational theories. In *International Conference on Automated Deduction, (CADE-12)*, volume 814 of *LNCS*, pages 267–281. 1994.

[7] S. Erbatur, D. Kapur, A. M. Marshall, P. Narendran, and C. Ringeissen. Hierarchical combination. In M. P. Bonacina, editor, *Automated Deduction (CADE-24)*, volume 7898 of *Lecture Notes in Computer Science*, pages 249–266. Springer Berlin Heidelberg, 2013.

[8] S. Erbatur, D. Kapur, A. M. Marshall, P. Narendran, and C. Ringeissen. Unification and matching in hierarchical combinations of syntactic theories. In C. Lutz and S. Ranise, editors, *Frontiers of Combining Systems - 10th International Symposium, FroCoS 2015, Wroclaw, Poland, September 21-24, 2015. Proceedings*, volume 9322 of *Lecture Notes in Computer Science*, pages 291–306. Springer, 2015.

[9] S. Erbatur, A. M. Marshall, and C. Ringeissen. Unification in non-disjoint combinations with forward-closed theories. Available at https://hal.inria.fr.

[10] S. Escobar, R. Sasse, and J. Meseguer. Folding variant narrowing and optimal variant termination. *J. Log. Algebr. Program.*, 81(7-8):898–928, 2012.

[11] J.-P. Jouannaud and C. Kirchner. Solving equations in abstract algebras: A rule-based survey of unification. In *Computational Logic - Essays in Honor of Alan Robinson*, pages 257–321, 1991.

[12] C. Kirchner and F. Klay. Syntactic theories and unification. In *Logic in Computer Science, 1990. LICS '90, Proceedings., Fifth Annual IEEE Symposium on Logic in Computer Science*, pages 270–277, Jun 1990.

[13] C. Lynch and B. Morawska. Basic syntactic mutation. In A. Voronkov, editor, *Automated Deduction - CADE-18, 18th International Conference on Automated Deduction, Copenhagen, Denmark, July 27-30, 2002, Proceedings*, volume 2392 of *Lecture Notes in Computer Science*, pages 471–485. Springer, 2002.

[14] T. Nipkow. Proof transformations for equational theories. In *Logic in Computer Science, 1990. LICS '90, Proceedings., Fifth Annual IEEE Symposium on Logic in Computer Science*, pages 278–288, Jun 1990.

[15] C. Ringeissen. Unification in a combination of equational theories with shared constants and its application to primal algebras. In *The 1st International Conference on Logic Programming and Automated Reasoning, volume 624 of LNAI*, pages 261–272. Springer, 1992.

[16] M. Schmidt-Schauß. Unification in a combination of arbitrary disjoint equational theories. *Journal of Symbolic Computation*, 8:51–99, July 1989.

[17] K. A. Yelick. Unification in combinations of collapse-free regular theories. *Journal of Symbolic Computation*, 3(1-2):153–181, 1987.