Terminating Non-Disjoint Combined Unification (Extended Abstract)

Serdar Erbatur, Andrew Marshall, Christophe Ringeissen

Terminating Non-Disjoint Combined Unification

(Extended Abstract)

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

[1] University of Texas at Dallas, USA
serdar.erbatur@utdallas.edu
[2] University of Mary Washington, USA
marshall@umw.edu
[3] Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
Christophe.Ringeissen@loria.fr

1 Introduction

Unification is a critical tool in many fields such as automated reasoning, logic programming, declarative programming, and the formal analysis of security protocols. For many of these applications we want to consider equational unification, where the problem is defined modulo an equational theory E, such as Associativity-Commutativity. Since equational unification is undecidable in general, specialized techniques have been developed to solve the problem for particular classes of equational theories, many of high practical interest. For instance, when the equational theory E has the Finite Variant Property (FVP) [3, 7], there exists a reduction from E-unification to syntactic unification via the computation of finitely many variants of the unification problem.

Another ubiquitous scenario is given by an equational theory E involved in a union of theories $F \cup E$. To solve this case, it is quite natural to proceed in a modular way by reusing the unification algorithms available for F and for E. There are terminating and complete combination procedures for signature-disjoint unions of theories [10, 2]. However, the non-disjoint case remains a challenging problem. One approach to the non-disjoint combination problem that has been successful in some cases is the hierarchical approach [5]. In this approach, $F \cup E$-unification can be considered as a conservative extension of E-unification. Then, a new inference system related to F, say U_F, can be combined with an E-unification algorithm to obtain a $F \cup E$ unification algorithm. While this hierarchical approach won't work for every $F \cup E$ it can be a very useful tool when applicable. However, up to now it could be complex to know if a combination $F \cup E$ could be solved via the hierarchical approach. For example, there is no general method for obtaining the inference system U_F, and the resulting hierarchical unification procedure may not terminate.

In this paper, we consider "syntactic" theories $F \cup E$ where U_F can be defined as a system of mutation rules, and we present new terminating instances of the hierarchical unification procedure.

2 Preliminaries

We use the standard notation of equational and term rewriting systems [1]. An equational theory E is *regular* if for any axiom $l = r \in E$, l and r have the same set of variables. An equational theory E is *collapse-free* if for any axiom $l = r \in E$, l and r are non-variable terms. An equational theory E is *subterm collapse-free* if for all terms t it is not the case that $t =_E u$ where u is a strict subterm of t. A subterm collapse-free theory is necessarily regular and collapse-free. An equational term rewrite system, equational TRS for short, is a pair (R, E) where R is a set of rewrite Σ-rules and E is an equational Σ-theory, Σ being a signature. An equational TRS (R, E) is said to be E-convergent if the rewrite relation $\to_{R,E}$, defined via E-matching, is *E-convergent*, meaning that $=_E \circ \to_{R,E} \circ =_E$ is terminating and $\to_{R,E}$ is Church-Rosser modulo E. A function symbol that does not occur in $\{l(\epsilon) \mid l \to r \in R\}$ is called a *constructor* for R. Let Σ_0 be the subsignature of Σ that consists of function symbols occurring in the axioms of E. An E-convergent TRS (R, E) is said to be *E-constructed* if all symbols in Σ_0 are constructors for R. Given two rewrite rules $g \to d$ and $l \to r$, the E-**Forward** inference generates a new rewrite rule when l and d overlap. It is formally defined as follows:

E-**Forward** $\quad g \to d[l'],\ l \to r \ \vdash\ (g \to d[r])\sigma$
where $g \to d[l'], l \to r \in R, l'$ is not a variable, $\sigma \in CSU_E(l' =^? l)$.

An E-convergent TRS (R, E) is *forward-closed* if any application of E-**Forward** generates a rule which is redundant in (R, E) when the premises are rules in (R, E), following an appropriate definition of redundancy [8]. It can be shown that for any E-constructed TRS (R, E) where E is regular, collapse-free and E-unification is finitary, (R, E) has the FVP if and only if it has a finite closure by E-**Forward**.

An *alien* subterm of a Σ_0-rooted term t is a $\Sigma\backslash\Sigma_0$-rooted subterm s such that all superterms of s are Σ_0-rooted. A set of equations $G = \{x_1 = t_1,\ \dots,\ x_n = t_n\}$ is said to be in *tree solved form* if each x_i is a variable occurring once in G.

3 Hierarchical Unification

Consider now a union of theories $R \cup E$ where E is regular and collapse-free and (R, E) is assumed to be E-constructed. Thanks to this assumption, R and E are "sufficiently separated" and thus we can envision the problem of building a $R\cup E$-unification algorithm using an approach based on combination. A hierarchical unification procedure is parameterized by an E-unification algorithm and a mutation-based reduction procedure U. It applies some additional rules given in Figure 1: **Coalesce**, **Split**, **Flatten**, and **VA** are used to separate the terms, U is used to simplify the $\Sigma\backslash\Sigma_0$-equations, and finally, **Solve** calls the E-unification algorithm.

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

Split $\quad \{f(\bar{v}) = t\} \cup G \ \vdash\ \{x = f(\bar{v}), x = t\} \cup G$
where $f \in \Sigma\backslash\Sigma_0$, t is a non-variable term and x is a fresh variable.

Flatten $\quad \{v = f(\dots, u, \dots)\} \cup G \ \vdash\ \{v = f(\dots, x, \dots), x = u\} \cup G$
where $f \in \Sigma\backslash\Sigma_0$, v is a variable, u is a non-variable term, and x is a fresh variable.

VA $\quad \{s = t[u]\} \cup G \ \vdash\ \{s = t[x], x = u\} \cup G$
where t is Σ_0-rooted, u is an alien subterm of t, and x is a fresh variable.

Solve $\quad G \cup G_0 \ \vdash\ \bigvee_{\sigma_0 \in CSU_E(G_0)} G \cup \hat{\sigma}_0$
where G is a set of $\Sigma\backslash\Sigma_0$-equations, G_0 is a set of Σ_0-equations, G_0 is E-unifiable and not in tree solved form, $\hat{\sigma}_0$ is the tree solved form associated with σ_0, and w.l.o.g for any $x \in Dom(\sigma_0)$, $x\sigma_0 \in Var(G_0)$ if $x\sigma_0$ is a variable.

Figure 1: H_E rules

Definition 1 (Hierarchical unification procedure)**.** *Assume a Σ_0-theory E for which an E-unification algorithm is known to compute a finite $CSU_E(G_0)$ for all E-unification problems G_0, a Σ-theory $F \cup E$ for which E-unification is complete for solving the Σ_0-fragment of $F \cup E$-unification, and an inference system U satisfying the following assumptions: U transforms only equations of the form $x_0 = f(x_1, \ldots, x_n)$ where $x_0, x_1, \ldots, x_n$ are variables and f is a function symbol in $\Sigma \backslash \Sigma_0$; U is sound and complete; and U is parameterized by some finite set S of $F \cup E$-equalities such that the soundness of each inference $\vdash_U$ follows from at most one equality in S. Under these assumptions, the $H_E(U)$ inference system is defined as the repeated application of some inference from H_E (cf. Figure 1) or U, using the following order of priority:* ***Coalesce****,* ***Split****,* ***Flatten****,* ***VA****, U,* ***Solve****. A $F \cup E$-unification problem is in* separate form *if it is a normal form with respect to $H_E \backslash \{$**Solve**$\}$.*

Note, that when we speak of an inference system, U, this is not just a set of rules but also a strategy for applying those rules. This could include, as in the $\mathcal{E}_{AC}$ case of Proposition 3, methods for detecting errors such as occur-checks and non-termination [6].

Proposition 1. *Let (R, E) be any E-constructed TRS such that an inference system U following Definition 1 is known for the equational theory $R \cup E$, in addition to an existing E-unification algorithm. Then E, $R \cup E$ and U satisfy the assumptions of Definition 1, and the $H_E(U)$ inference system provides a sound and complete $R \cup E$-unification procedure if the normal forms w.r.t $H_E(U)$ are either the dag solved forms or problems that are not $R \cup E$-unifiable. If $H_E(U)$ is terminating, then it is a $R \cup E$-unification algorithm.*

3.1 Subterm Collapse-Free Theories

Hierarchical unification algorithms are known for particular subterm collapse-free theories of particular interest for protocol analysis.

Proposition 2. *([11, 6]) Let E be the empty Σ_0-theory where Σ_0 only consists of a binary function symbol $*$. Consider $R_{\mathcal{D}} = \{h(x * y) \rightarrow h(x) * h(y)\}$ and $R_{\mathcal{D}1} = \{f(x * y, z) \rightarrow f(x, z) * f(y, z)\}$. The equational TRSs $(R_{\mathcal{D}}, E)$ and $(R_{\mathcal{D}1}, E)$ are E-constructed. Moreover, $R_{\mathcal{D}} \cup E$ (resp., $R_{\mathcal{D}1} \cup E$) is a subterm collapse-free theory admitting a unification algorithm of the form $H_E(U_{\mathcal{D}})$ (resp., $H_E(U_{\mathcal{D}1})$).*

Proposition 3. *([6]) Let $AC = AC(\circledast)$, $R_{\mathcal{E}} = \{exp(exp(x, y), z) \rightarrow exp(x, y \circledast z), exp(x * y, z) \rightarrow exp(x, z) * exp(y, z)\}$ and $R_{\mathcal{F}} = \{enc(enc(x, y), z) \rightarrow enc(x, y \circledast z)\}$. The equational TRSs $(R_{\mathcal{E}}, AC)$ and $(R_{\mathcal{F}}, AC)$ are AC-constructed. Moreover, $\mathcal{E}_{AC} = R_{\mathcal{E}} \cup AC$ (resp., $\mathcal{F}_{AC} = R_{\mathcal{F}} \cup AC$) is a subterm collapse-free theory admitting a unification algorithm of the form $H_{AC}(U_{\mathcal{E}})$ (resp., $H_{AC}(U_{\mathcal{F}})$).*

3.2 Forward-Closed E-Constructed TRSs

For any forward-closed E-constructed TRS (R, E) such that E is regular and collapse-free, a $R \cup E$-unification algorithm of the form $H_E(U)$ can be obtained by defining some inference system U based on the *Basic Syntactic Mutation* approach initiated for the class of theories saturated by paramodulation [9], and already applied in [4] to a particular class of forward-closed equational TRSs.

Let BSM_R be the inference system given in Figure 2. One can notice that each inference rule in BSM_R generates some boxed terms. This particular annotation of terms, detailed in [9, 4], allows us to control the rules application in such a way that BSM_R is terminating.

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 $f \in \Sigma\backslash\Sigma_0$, $i > 1$, $\bar{y}$ are fresh variables and there are no more equations $x = f(\dots)$ in G.

MutConflict$_R$ $\{x = f(\bar{v})\} \cup G \ \vdash\ \{x = \boxed{t}, \boxed{\bar{s}} = \bar{v}\} \cup G$
where $f \in \Sigma\backslash\Sigma_0$, $f(\bar{s}) \to t$ is a fresh instance of a rule in R, $f(\bar{v})$ is unboxed, and (there is another equation $x = u$ in G with a non-variable term u or $x = f(\bar{v})$ occurs in a cycle).

ImitCycle $\{x = f(\bar{v})\} \cup G \ \vdash\ \{x = \boxed{f(\bar{y})}, \bar{y} = \bar{v}\} \cup G$
where $f \in \Sigma\backslash\Sigma_0$, $f(\bar{v})$ is unboxed, $\bar{y}$ are fresh variables and $x = f(\bar{v})$ occurs in a cycle.

Figure 2: BSM_R rules

Lemma 1. *Assume E is any regular and collapse-free theory such that an E-unification algorithm is known. Let (R, E) be a forward-closed E-constructed TRS and BSM_R the inference system given in Fig. 2. Then $H_E(BSM_R)$ is a $R \cup E$-unification algorithm.*

Example 1. *Consider $R = \{\pi_1(x.y) \to x, \pi_2(x.y) \to y, dec(enc(x, y), y) \to x\}$ and $E = \{enc(x.y, z) = enc(x, z).enc(y, z)\}$. E-unification algorithms are know for this type of one-sided distributivity [11] and can be used in a hierarchical unification procedure of the form $H_E(BSM_R)$. Since (R, E) is forward-closed and E-constructed, $H_E(BSM_R)$ provides an $R \cup E$-unification algorithm.*

4 Combined Hierarchical Unification

We are now interested in combining hierarchical unification algorithms known for E-constructed TRSs. Given two E-constructed TRSs, say (R_1, E) and (R_2, E), the problem is to study the possible construction of a (combined) hierarchical unification algorithm for $(R_1 \cup R_2, E)$ using the two hierarchical unification algorithms known for (R_1, E) and (R_2, E).

4.1 Combining Subterm Collapse-Free Theories

Let us first consider a technical lemma which is useful to get a hierarchical unification procedure.

Lemma 2. *Let (R_1, E) and (R_2, E) be two E-constructed TRSs sharing only symbols in E such that, for $i = 1, 2$, $R_i \cup E$ admits a sound and complete unification procedure of the form $H_E(U_i)$. Assume that $R_1 \cup R_2 \cup E$ is subterm collapse-free, and for any $\Sigma_1\backslash\Sigma_0$-rooted term t_1 and any $\Sigma_2\backslash\Sigma_0$-rooted term t_2, t_1 cannot be equal to t_2 modulo $R_1 \cup R_2 \cup E$. Then, $H_E(U_1 \cup U_2)$ is a sound and complete $R_1 \cup R_2 \cup E$-unification procedure.*

We study below a possible way to satisfy the assumptions of Lemma 2.

Definition 2 (Layer-preservingness)**.** *Let (R, E) be an E-constructed TRS over the signature Σ. A Σ-term t is said to be* Σ_0-capped *if there exist a constant-free Σ_0-term u and a substitution σ such that $t = u\sigma$, $Dom(\sigma) = Var(u)$ and $Ran(\sigma)$ is a set of $\Sigma\backslash\Sigma_0$-rooted terms. The TRS (R, E) is said to be* layer-preserving *if $R \cup E$ is subterm collapse-free and any normal form of any $\Sigma\backslash\Sigma_0$-rooted term is Σ_0-capped.*

Remark 1. *The assumption that rules in R are $\Sigma\backslash\Sigma_0$-rooted was used in [5], and layer-preservingness generalizes this assumption.*

The property of being E-constructed and layer-preserving is modular.

Lemma 3. *Assume E is subterm collapse-free, for $i = 1, 2$, (R_i, E) is an E-constructed layer-preserving TRS whose signature is Σ_i, and $\Sigma_1 \cap \Sigma_2 = \Sigma_0$. If $=_E \circ \rightarrow_{R_1 \cup R_2} \circ =_E$ is terminating, then $(R_1 \cup R_2, E)$ is an E-constructed layer-preserving TRS, and for any $\Sigma_1 \backslash \Sigma_0$-rooted term t_1 and any $\Sigma_2 \backslash \Sigma_0$-rooted term t_2, t_1 cannot be equal to t_2 modulo $R_1 \cup R_2 \cup E$.*

By Lemma 3, the two assumptions of Lemma 2 can be satisfied, and this leads to a hierarchical unification procedure for the combined TRS. In the following, we consider a notion of decreasingness in order to study the termination of this unification procedure.

Definition 3 (Decreasingness)**.** *Consider a complexity measure defined as a mapping C from separate forms to natural numbers. A $H_E(U)$ inference system is said to be C-*decreasing *if for any separate form $G \cup G_0$ we have that (1) for any G' such that $G \cup G_0 \vdash_U G' \cup G_0$, the separate form of $G' \cup G_0$ does not increase C; (2) for any G_0' such that $G \cup G_0 \vdash_{\mathbf{Solve}} G \cup G_0'$, then either the separate form of $G \cup G_0'$ is in normal form w.r.t $H_E(U)$, or it decreases C.*

Consequently, $H_E(U)$ is terminating if there exists some C such that $H_E(U)$ is C-decreasing.

Theorem 1. *Assume E is a subterm collapse-free theory such that an E-unification algorithm is known, and C is a complexity measure defined on separate forms. Let (R_1, E) and (R_2, E) be two E-constructed TRSs sharing only symbols in E such that, for $i = 1, 2$, (R_i, E) is layer-preserving, and $R_i \cup E$ admits a C-decreasing unification algorithm of the form $H_E(U_i)$. If $=_E \circ \rightarrow_{R_1 \cup R_2} \circ =_E$ is terminating, then $(R_1 \cup R_2, E)$ is an E-constructed TRS such that $(R_1 \cup R_2, E)$ is layer-preserving, and $R_1 \cup R_2 \cup E$ admits a C-decreasing unification algorithm of the form $H_E(U_1 \cup U_2)$.*

Example 2. *Consider the theories $\mathcal{E}_{AC}$ and $\mathcal{F}_{AC}$ introduced in Proposition 3 and the corresponding hierarchical unification algorithms $H_{AC}(U_{\mathcal{E}})$ and $H_{AC}(U_{\mathcal{F}})$ where the mutation rules defining $U_{\mathcal{E}}$ and $U_{\mathcal{F}}$ can be found in [6]. Let SVC be the complexity measure defined as follows: given a $R \cup E$-unification problem in separate form $G \cup G_0$, $SVC(G \cup G_0)$ is the number of equivalence classes of variables shared by G and G_0 that are variables abstracting $\Sigma \backslash \Sigma_0$-rooted terms.*

One can check that the unification algorithms $H_{AC}(U_{\mathcal{E}})$ and $H_{AC}(U_{\mathcal{F}})$ are both SVC-decreasing. By Theorem 1, we get that $\mathcal{E}_{AC} \cup \mathcal{F}_{AC}$ admits a SVC-decreasing unification algorithm of the form $H_{AC}(U_{\mathcal{E}} \cup U_{\mathcal{F}})$. We suspect that this complexity measure, SVC, could be useful for proving termination in other theories.

4.2 Combining Forward-Closed E-Constructed TRSs

The union of two forward-closed E-constructed TRSs remains a forward-closed E constructed TRS. Thus, a hierarchical unification algorithm can be constructed in a modular way in unions of forward-closed E-constructed TRSs.

Theorem 2. *Assume E is a regular and collapse-free theory such that an E-unification algorithm is known. Let (R_1, E) and (R_2, E) be two forward-closed E-constructed TRSs sharing only symbols in E. Then $R_1 \cup R_2 \cup E$ admits a unification algorithm of the form $H_E(BSM_{R_1} \cup BSM_{R_2})$.*

References

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

[2] F. Baader and K. U. Schulz. Unification in the union of disjoint equational theories: Combining decision procedures. *Journal of Symbolic Computation*, 21(2):211 – 243, 1996.

[3] 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.

[4] A. K. Eeralla, S. Erbatur, A. M. Marshall, and C. Ringeissen. Rule-based unification in combined theories and the finite variant property. In C. Martín-Vide, A. Okhotin, and D. Shapira, editors, *Language and Automata Theory and Applications - 13th International Conference, LATA 2019, St. Petersburg, Russia, March 26-29, 2019, Proceedings*, volume 11417 of *Lecture Notes in Computer Science*, pages 356–367. Springer, 2019.

[5] S. Erbatur, D. Kapur, A. M. Marshall, P. Narendran, and C. Ringeissen. Hierarchical combination. In M. P. Bonacina, editor, *Automated Deduction - CADE-24 - 24th International Conference on Automated Deduction, Lake Placid, NY, USA, June 9-14, 2013. Proceedings*, volume 7898 of *Lecture Notes in Computer Science*, pages 249–266. Springer, 2013.

[6] S. Erbatur, A. M. Marshall, D. Kapur, and P. Narendran. Unification over distributive exponentiation (sub)theories. *Journal of Automata, Languages and Combinatorics (JALC)*, 16(2–4):109–140, 2011.

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

[8] D. Kim, C. Lynch, and P. Narendran. Reviving basic narrowing modulo. In A. Herzig and A. Popescu, editors, *Frontiers of Combining Systems - 12th International Symposium, FroCoS 2019, London, UK, September 4-6, 2019, Proceedings*, volume 11715 of *Lecture Notes in Computer Science*, pages 313–329. Springer, 2019.

[9] 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.

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

[11] E. Tidén and S. Arnborg. Unification problems with one-sided distributivity. *Journal of Symbolic Computation*, 3(1/2):183–202, 1987.