Combined Abstract Congruence Closure for Associative or Commutative Theories

Christophe Ringeissen and Laurent Vigneron

Université de Lorraine, CNRS, Inria, Loria, F-54000 Nancy, France.
FirstName.LastName@loria.fr

1 Introduction

We design rule-based satisfiability procedures modulo unions of axiomatized theories, targeting equational axioms such as Associativity or Commutativity. In the proposed approach, any function symbol can be uninterpreted, associative only, commutative only, but also associative and commutative. To tackle these unions of theories, we introduce a combined congruence closure procedure that can be viewed as a particular Nelson-Oppen combination method [8] using particular congruence closure procedures for the individual theories. The combined congruence procedure is based on the ping-ponging of entailed equalities between (shared) constants. Actually, the congruence closure procedures used for the individual theories allow us to deduce these equalities. In this context, we consider terminating congruence closure procedures, but also non-terminating ones. Hence, we have terminating congruence closure procedures for Commutativity and Associativity-Commutativity, while the one for Associativity is non-terminating. We show how all the congruence closure procedures, including the combined one, can be presented in a uniform and abstract way along the lines of [3, 5, 7].

Related Work. Congruence closure modulo Associativity-Commutativity has been successfully investigated in [3, 4]. It has been revisited more recently, showing how the method can be extended to take into account additional orientable axioms, for instance to handle the theory of Abelian Groups [5]. The case of flat permutation axioms, such as Commutativity, has been considered in [7]. The theory of Groups and all of its subtheories including Associativity is considered in [6], where the related congruence closure procedure is not necessarily terminating, contrarily to the one known for Associativity-Commutativity. In these papers, some particular unions of theories are studied, for instance to handle several symbols following the same equational axioms.

In our paper, we clearly focus on the combination of congruence closure procedures to cope with arbitrary unions of (signature-disjoint) theories. This combination of congruence closure procedures can be seen as a particular case of combination of deduction-complete satisfiability procedures, already investigated in [12]. In addition to Associativity-Commutativity, we believe that it is interesting to consider Associativity alone and Commutativity alone. On one hand, Associativity provides a significant case study of a non-terminating congruence closure procedure. On the other hand, Commutativity leads to a simple extension of the congruence closure procedure known for the theory of equality as done in [7].

Paper Outline. In this paper, after explaining the notations used, we describe our combination method based on two kinds of processes: the orchestrator whose role is to prepare and handle a combination of theories; a theory process whose role is to complete the set of rewrite rules for a specific theory. Then we discuss the completeness of the method.

2 Preliminaries

We assume the reader familiar with the notions of terms and term rewriting [1].

We consider n theories $\mathcal{E}_1, \dots, \mathcal{E}_n$ such that each theory $\mathcal{E}_i$ is given by a set of equalities over the signature Σ_i. The theories $\mathcal{E}_1, \dots, \mathcal{E}_n$ are assumed to be pairwise signature-disjoint, meaning that $\Sigma_i \cap \Sigma_j = \emptyset$ for any $i, j \in [1, n], i \neq j$. The union of theories $\mathcal{E}_1 \cup \cdots \cup \mathcal{E}_n$ is denoted by $\mathcal{E}$ and the union of signatures $\Sigma_1 \cup \cdots \cup \Sigma_n$ is denoted by Σ. We assume a set of ground equalities Γ and a set of ground disequalities Δ, where both Γ and Δ are expressed over the signature Σ.

The process described in this paper relies on a flattening of terms. For theory $\mathcal{E}_i$ including an operator, say $+$, such that $(x + y) + z \approx x + (y + z)$ occurs in $\mathcal{E}_i$, this flattening will be performed using $+$ as a variadic operator, eg. $a + (b + c)$ is flattened into $+(a, b, c)$.

The initial set of ground equalities Γ will be purified via flattening thanks to the introduction of new constants (K denotes the set of used new constants taken from an infinite countable set U disjoint from Σ), generating pure flat rewrite rules for each theory $\mathcal{E}_i$ (denoted by the set R_i); and further deductions between those rules may generate flat equalities in this theory (denoted by the set E_i).

The rewrite rules in R_i can have two shapes: D-rules denoted by $f(c_1, \dots, c_n) \to c$, where $f \in \Sigma_i$ and $c_1, \dots, c_n, c \in K$; E-rules denoted by $f(c_1, \dots, c_m) \to f(d_1, \dots, d_n)$, where $f \in \Sigma_i$ is a variadic operator and $c_1, \dots, c_m, d_1, \dots, d_n \in K$. For any rewrite rule $t \to s$, t has to be greater than s ($t \succ s$); the definition of an ordering may be difficult for deduction systems modulo equational theories; but in our case the ordering is very simple as we only have to consider D-rules and E-rules: for D-rules, it suffices to assume $\forall f \in \Sigma, \forall c \in K, f \succ c$; for E-rules, we have to compare lists of constants: if of the same length, this can be done with a lexicographic or a multiset extension of an arbitrary ordering comparing two constants of K (the choice is done for each theory), and if of different length, the longest is the biggest. For example, for an associative theory the lexicographic extension will be used, and for an associative-commutative theory the multiset extension will be used.

The equalities in E_i also have two shapes: D-equalities denoted by $f(c_1, \dots, c_n) \approx c$, where $f \in \Sigma_i$ and $c_1, \dots, c_n, c \in K$; E-equalities denoted by $f(c_1, \dots, c_m) \approx f(d_1, \dots, d_n)$, where $f \in \Sigma_i$ is a variadic operator and $c_1, \dots, c_m, d_1, \dots, d_n \in K$. An equality $c_1 \approx c_2$ between two constants of K is called a C-equality. E-rules and E-equalities will be generated only for variadic operators by the Superposition inference rule, because of the use of extended rewrite rules (see Section 3.2).

3 Combined Satisfiability Procedure

We describe in this section a procedure that aims at (semi-)deciding the satisfiability of any set of ground equalities Γ together with any set of ground disequalities Δ, modulo a combination of signature-disjoint equational theories $\mathcal{E}_i$. This procedure, called `CombCC`, is based on congruence closure and involves two kinds of processes: an orchestrator decomposing the problem to separate the different theories, and theory processes that complete rewrite rules, one process for each theory.

3.1 The Orchestrator

The role of the orchestrator is to purify and flatten the problem to be solved, to send each theory process the rewrite rules it has to handle, and to detect if any contradiction wrt. Δ is generated by one of the theory processes. For this purpose, several sets are handled in addition to Γ and Δ: the set of new constants K, and for each equational theory $\mathcal{E}_i$ a set of rewrite rules R_i and a set of equalities E_i. In the following inference rules, we will only indicate the involved sets.

The input problem is given by two sets: a set of ground equalities Γ built over Σ plus a set of ground disequalities Δ built over Σ.

The first task of the orchestrator is to transform the disequalities for hiding the theories involved. This is done with the following inference rule that replaces an arbitrary disequality by a disequality between two new constants together with the equalities associating each of these constants to the corresponding term:

Splitting:
$$\frac{K, \Delta \cup \{t_1 \not\approx t_2\}, \Gamma}{K \cup \{c_1, c_2\}, \Delta \cup \{c_1 \not\approx c_2\}, \Gamma \cup \{t_1 \approx c_1, t_2 \approx c_2\}}$$

if $t_1, t_2 \notin K$, $c_1, c_2 \in U \setminus K$

Once all disequalities have been decomposed, the second task of the orchestrator is to purify the equalities of Γ, by generating rewrite rules that are purely in one theory. In this purpose, it applies the following inference rules:

Flattening:
$$\frac{K, \Gamma[t], R_i}{K \cup \{c\}, \Gamma[c], R_i \cup \{t \to c\}}$$

if $t \to c$ is a D-rule, $c \in U \setminus K$, t occurs in some equality in Γ that is not a D-equality, and t is Σ_i-rooted.

Orientation:
$$\frac{K \cup \{c\}, \Gamma \cup \{t \approx c\}, R_i}{K \cup \{c\}, \Gamma, R_i \cup \{t \to c\}}$$
if $t \approx c$ is a D-equality and t is Σ_i-rooted.

When all equations have been transformed ($\Gamma = \emptyset$), the orchestrator runs one process per equational theory $\mathcal{E}_i$, providing it two sets of information: the set of new constants K and the set of D-rules R_i defined over Σ_i and K.

Its final task is to manage equalities between new constants, when generated by a theory process in some set E_i; there are two possibilities: if the equality contradicts a disequality of Δ then the system can stop, otherwise a constant has to be replaced by the other in all sets.

Contradiction:
$$\frac{K \cup \{c, d\}, \Delta \cup \{c \not\approx d\}, RE \cup (R_i, E_i \cup \{c \approx d\})}{\bot}$$

Compression:
$$\frac{K \cup \{c, d\}, \Delta, RE \cup (R_i, E_i \cup \{c \approx d\})}{K \cup \{d\}, \Delta\langle c \mapsto d\rangle, RE\langle c \mapsto d\rangle \cup (R_i\langle c \mapsto d\rangle, E_i\langle c \mapsto d\rangle)}$$

if $c \succ d$; the notation $\langle c \mapsto d\rangle$ denotes the homomorphic extension of the mapping σ defined as $\sigma(c) = d$ and $\sigma(x) = x$ for $x \neq c$, and $S\langle c \mapsto d\rangle$ denotes the set of equalities/rules obtained by applying the mapping $\langle c \mapsto d\rangle$ to each term in set S.

The strategy of the orchestrator can therefore be described by: $\mathbf{Split}^* \cdot (\mathbf{Flat}^* \cdot \mathbf{Ori})^* \cdot (\mathbf{Cont} \cup \mathbf{Comp})^*$.

3.2 A Theory Process

A process run for an equational theory $\mathcal{E}_i$ will use inference rules to complete its term rewriting system R_i. Some inference rules are used for transforming the rewrite rules (Composition), for deducing new equalities added to a set E_i (Collapse, Superposition), and for handling those new equalities (Simplification, Orientation, Deletion). The set of new constants K is never modified, so not indicated, but it is useful in the process for checking if a constant is a new one or belongs to the theory.

Considering the theory $\mathcal{E}_i$, if two terms t_1 and t_2 are $\mathcal{E}_i$-equal in this theory, we write $t_1 \leftrightarrow^*_{\mathcal{E}_i} t_2$. By $(R_i, \mathcal{E}_i)$ we denote the rewriting system defined by $\{u' \to v \mid u \to v \in R_i \text{ and } u' \leftrightarrow^*_{\mathcal{E}_i} u\}$. For some theories, the inference system has to consider extended rewrite rules as we do not explicitly use the axioms of a theory: an extension is built wrt. a context defined from the theory axioms; a context is a term in which a non variable position (denoted by $\cdot$) is reserved for placing the term to extend; let us denote $Cont_{\mathcal{E}_i}$ the set of contexts for the theory $\mathcal{E}_i$; given a D-rule or a E-rule $u \to v$, its extended version by a context $Cont[\cdot] \in Cont_{\mathcal{E}_i}$ is written $Cont[u] \to Cont[v]$. The construction of contexts for generating extensions has been explained in [10, 9, 13].

In this paper, as we want to handle only shallow rewrite rules, we will consider only theories for which extended rewrite rules have a shallow form. For example, if an operator f is associative, from the axiom of this theory $f(f(x, y), z) \approx f(x, f(y, z))$, we can build three shallow contexts: $f(\cdot, x)$, $f(x, \cdot)$ and $f(x_1, \cdot, x_2)$. So, a rewrite rule $f(a, b) \to c$ has three extensions: $f(a, b, x) \to f(c, x)$, $f(x, a, b) \to f(x, c)$ and $f(x_1, a, b, x_2) \to f(x_1, c, x_2)$.

By $(R_i^e, \mathcal{E}_i)$ we denote the rewriting system extending $(R_i, \mathcal{E}_i)$ with all possible extended rewrite rules from R_i.

The inference rules used by a theory process are the following.

Simplification: $\dfrac{R_i, E_i[t]}{R_i, E_i[s]}$ where t occurs in some equality of E_i, and $t \to_{(R_i^e, \mathcal{E}_i)} s$.

Orientation: $\dfrac{R_i, E_i \cup \{t \approx s\}}{R_i \cup \{t \to s\}, E_i}$ if $t \succ s$ and $t \to s$ is a D-rule or a E-rule.

Deletion: $\dfrac{R_i, E_i \cup \{t \approx s\}}{R_i, E_i}$ if $t \leftrightarrow^*_{\mathcal{E}_i} s$.

Composition: $\dfrac{R_i \cup \{t \to s, u \to v\}, E_i}{R_i \cup \{t \to s', u \to v\}, E_i}$ if $s \to_{(\{u \to v\}^e, \mathcal{E}_i)} s'$.

Collapse: $\dfrac{R_i \cup \{t \to s, u \to v\}, E_i}{R_i \cup \{u \to v\}, E_i \cup \{t' \approx s\}}$ if $t \to_{(\{u \to v\}^e, \mathcal{E}_i)} t'$, and if $t \leftrightarrow^*_{\mathcal{E}_i} u$ then $s \succ v$.

Superposition: $\dfrac{R_i \cup \{t_1 \to s_1, t_2 \to s_2\}, E_i}{R_i \cup \{t_1 \to s_1, t_2 \to s_2\}, E_i \cup \{Cont_1[s_1]\sigma \approx Cont_2[s_2]\sigma\}}$

if the substitution σ is the ground substitution in a minimal complete set of $\mathcal{E}_i$-unifiers of $Cont_1[t_1]$ and $Cont_2[t_2]$, where $Cont_1[\cdot], Cont_2[\cdot] \in Cont_{\mathcal{E}_i}$ are selected to guarantee a useful ground new equality; the resulting equality will be written in flat form.

A strategy for combining all those inference rules is: $(\mathbf{Com}^* \cdot (\mathbf{Col} \cup \mathbf{Sup}) \cdot \mathbf{Sim}^* \cdot (\mathbf{Del} \cup \mathbf{Ori}))^*$ So this process starts with a set of rewrite rules R_i and, if terminating, it ends with R_i^∞ where there is no possible inference rule involving rewrite rules of R_i^∞; intermediate equalities are stored in E_i. If an equality between two constants of K is generated, it will be handled by the orchestrator.

For applying inference rules, this theory process has to use a $\mathcal{E}_i$-matching algorithm for applying rewriting steps with respect to $(R_i^e, \mathcal{E}_i)$. It also needs a simple $\mathcal{E}_i$-unification algorithm, simple because it will have to solve unification problems of the shape $Cont_1[t_1] =^? Cont_2[t_2]$, where t_1 and t_2 are ground; if there is a solution, it will be the unique most general unifier since the variables occurring in $Cont_i[\cdot]$ will be instantiated by subterms of the ground term t_{3-i}.

4 Completeness Results

The combined satisfiability procedure described in the previous section is defined for "some theories $\mathcal{E}_i$". But for guaranteeing its completeness, we consider only three kinds of theories (in addition to the empty theory of course).

- *Commutative* theories are represented by the set of axioms
  $$\{f(x_1, \ldots, x_k) \approx f(x_{\sigma(1)}, \ldots, x_{\sigma(k)}) \mid \sigma \text{ is a permutation of } \{1, \ldots, k\}\}$$
  With such theories, there is no extension of rewrite rules to be considered, so the Superposition inference rule cannot apply. For the ordering, the arguments of a commutative operator are compared with a multiset extension of the ordering between constants of K.

- *Associative* theories are represented by axioms $f(f(x_1, x_2), x_3) \approx f(x_1, f(x_2, x_3))$. They generate three possible extensions of rewrite rules, with the contexts $f(\cdot, x)$, $f(x, \cdot)$ and $f(x_1, \cdot, x_2)$. Those three contexts can be used for applying term rewriting steps with respect to $(R_i^e, \mathcal{E}_i)$. But for the Superposition inference rule between two rules $t_1 \to s_1$ and $t_2 \to s_2$, we only need to consider their extensions $f(t_1, x_1) \to f(s_1, x_1)$ and $f(x_2, t_2) \to f(x_2, s_2)$ because this is the only combination of contexts for which the unification problem $(f(t_1, x_1) =^? f(x_2, t_2))$ can generate a ground most general unifier (any use of another context would generate a redundant equation). For the ordering, the arguments of an associative operator are compared with a lexicographic extension of the ordering between constants of K.

- *Associative-Commutative* theories are represented by axioms $f(x_1, x_2) \approx f(x_2, x_1)$ and $f(f(x_1, x_2), x_3) \approx f(x_1, f(x_2, x_3))$. They generate only one possible extension of rewrite rules, with the context $f(\cdot, x)$, used for applying term rewriting steps by $(R_i^e, \mathcal{E}_i)$, and the Superposition inference rule. For the ordering, the arguments of an AC operator are compared with a multiset extension of the ordering between constants of K.

The `CombCC` procedure is refutationally complete, provided that deductions are fairly applied. Moreover, if the `CombCC` procedure terminates without finding a contradiction with disequalities of Δ, it generates a terminating confluent term rewriting system for the equational theory $\mathcal{E} \cup \Gamma$.

Theorem 1. *Let $\mathcal{E}$ be any disjoint union of empty, commutative, associative, and associative-commutative theories over the combined signature Σ which is assumed to include uninterpreted function symbols and constants. Consider Γ is any set of ground Σ-equalities and Δ is any set of ground Σ-disequalities. Given the input $\Gamma \cup \Delta$, the `CombCC` procedure halts on $\bot$ if $\Gamma \cup \Delta$ is $\mathcal{E}$-unsatisfiable. If the `CombCC` procedure halts on an output distinct from $\bot$, then $\Gamma \cup \Delta$ is $\mathcal{E}$-satisfiable, and the output provides a rewriting system R such that (1) R is terminating and confluent modulo $\mathcal{E}$ on $T(\Sigma \cup K)$, and (2) any two ground terms in $T(\Sigma)$ are $\mathcal{E} \cup R$-equal iff they are $\mathcal{E} \cup \Gamma$-equal. Moreover, the `CombCC` procedure is necessarily terminating if $\mathcal{E}$ does not involve associative theories.*

Example 1. *Our procedure may indeed not terminate (if no contradiction exists) with associative theories. For example, if f and g are associative, given the equalities $\{f(a, b) \approx c, f(a, c) \approx f(c, a), g(b, a) \approx c, g(a, c) \approx g(c, a)\}$, either the theory process of f, or the one of g, will generate* *an infinite number of rewrite rules, depending on the chosen ordering between constants a and c deciding of the orientation of the second and fourth equalities. If $c \succ a$, the infinitely generated rules can be schematized by $f(a, c^n, b) \to f(c, c^n)$. If $a \succ c$, they can be schematized by $g(b, c^n, a) \to f(c, c^n)$. We also have non-terminating examples with a single associative theory, but they may be less simple to explain.*

To prove the completeness of `CombCC`, we can rely on a Nelson-Oppen combination method [8] based on the ping-ponging of entailed equalities between (shared) constants. This combination method is applicable without loss of completeness because one can rely on a union of convex and stably infinite theories, using the same proof idea as the one initiated in [2]. The theories we focus on are convex, since equational theories are Horn theories, and Horn theories are known to be convex [11]. Actually, the convexity induces a particular way to decide the satisfiability of equalities plus a conjunction of disequalities: it allows us to consider each disequality separately. Assuming convexity, a satisfiable set of equalities Γ together with a set of disequalities Δ is satisfiable if and only if for any $s \not\approx t \in \Delta$, we have that $s \approx t$ is not entailed by Γ. In our context, arbitrary satisfiability problems are equi-satisfiable, via flattening, to satisfiability problems including only flat literals, meaning that all the disequalities in Δ are of the form $c \not\approx d$ where c and d are constants. Thus, we are looking for inference systems with the property of being deduction-complete [12], in order to *derive* each equality $c \approx d$ such that $\Gamma \Rightarrow c \approx d$ is valid in the underlying theory. This is exactly the purpose of a congruence closure procedure when it applies to an input set of flat equalities Γ. It generates all the equalities between constants that are logically entailed by Γ.

In this paper, we consider terminating congruence closure procedures, but also non-terminating ones. Compared to a classical use of the Nelson-Oppen combination method, we have to accommodate procedures that are not necessarily terminating, as exemplified by Associativity.

Let us shortly explain why `CombCC` is refutationally complete. According to the completeness of the Nelson-Oppen method, the satisfiability problem in any disjoint union of stably infinite theories is reducible to the satisfiability problems in the component theories, provided that all possible arrangements are guessed. Consequently, given any disjoint union of stably infinite theories, using refutationally complete procedures for the satisfiability problems in the component theories allows us to get a refutationally complete procedure for the satisfiability problem in the union. In our context, stably infinite theories are also convex and so the guessing of all possible arrangements can be replaced by a ping-ponging of entailed equalities between constants. Then, we use the property that all the entailed equalities between constants are eventually generated since our congruence closure procedures are deduction-complete.

5 Conclusion

We have implemented the combination of those three kinds of theories (plus the empty theory) by extending AbstractCC [3]. The result is a very efficient procedure, even if the initial set of ground (dis)equalities contains very big terms.

We have defined the orchestrator so that it does not need to handle specific algorithms of theories $\mathcal{E}_i$. It could be more efficient using other inference rules like Simplification and Deletion. But we did this on purpose for the clarity of the paper. We are considering several extensions of our procedure, to apply it to any theory having a deduction system preserving the groundness of generated rules/equalities. This applies to flat permutative theories, an extension of commutative theories. It also applies to extensions of associative or associative-commutative theories where axioms can be used as shallow collapsing rewrite rules.

References

[1] Franz Baader and Tobias Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.

[2] Franz Baader and Cesare Tinelli. A New Approach for Combining Decision Procedure for the Word Problem, and Its Connection to the Nelson-Oppen Combination Method. In William McCune, editor, *CADE-14, 14th Int. Conference on Automated Deduction, Townsville, North Queensland, Australia, July 1997, Proceedings*, volume 1249 of *LNCS*, pages 19–33. Springer, 1997.

[3] Leo Bachmair, Ashish Tiwari, and Laurent Vigneron. Abstract Congruence Closure. *J. Autom. Reason.*, 31(2):129–168, 2003.

[4] Deepak Kapur. A Modular Associative Commutative (AC) Congruence Closure Algorithm. In Naoki Kobayashi, editor, *6th Int. Conference on Formal Structures for Computation and Deduction, FSCD, Buenos Aires, Argentina, July 2021 (Virtual Conference)*, volume 195 of *LIPIcs*, pages 15:1–15:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[5] Deepak Kapur. Modularity and Combination of Associative Commutative Congruence Closure Algorithms enriched with Semantic Properties. *Log. Methods Comput. Sci.*, 19(1), 2023.

[6] Dohan Kim. Congruence closure modulo groups. *CoRR*, abs/2310.05014, 2023.

[7] Dohan Kim and Christopher Lynch. Congruence Closure Modulo Permutation Equations. In Temur Kutsia, editor, *Proceedings of the 9th Int. Symposium on Symbolic Computation in Software Science, SCSS 2021, Hagenberg, Austria, September 2021*, volume 342 of *EPTCS*, pages 86–98, 2021.

[8] Greg Nelson and Derek C. Oppen. Simplification by Cooperating Decision Procedures. *ACM Trans. Program. Lang. Syst.*, 1(2):245–257, 1979.

[9] Gerald E. Peterson and Mark E. Stickel. Complete Sets of Reductions for Some Equational Theories. *J. ACM*, 28(2):233–264, 1981.

[10] Gordon Plotkin. Building-in equational theories. *Machine intelligence*, 7:73–90, 1972.

[11] Cesare Tinelli. Cooperation of Background Reasoners in Theory Reasoning by Residue Sharing. *J. Autom. Reason.*, 30(1):1–31, 2003.

[12] Duc-Khanh Tran, Christophe Ringeissen, Silvio Ranise, and Hélène Kirchner. Combination of convex theories: Modularity, deduction completeness, and explanation. *J. Symb. Comput.*, 45(2):261–286, 2010.

[13] Laurent Vigneron. Positive Deduction modulo Regular Theories. In Hans Kleine Büning, editor, *Computer Science Logic, 9th Int. Workshop, CSL '95, Annual Conference of the EACSL, Paderborn, Germany, September 1995, Selected Papers*, volume 1092 of *LNCS*, pages 468–485. Springer, 1995.