A Superposition Based Methodology to Design Satisfiability Decision Procedures

Alessandro Armando[1], Silvio Ranise[1,2], and Michaël Rusinowitch[2]

[1] DIST—Universitá di Genova, Genova, Italy, {armando, silvio}@dist.unige.it
[2] LORIA-INRIA-Lorraine, Nancy, France, {ranise, rusi}@loria.fr

1 Introduction

In verification with proof assistants (such as PVS, COQ, HOL, and Nqthm), decision procedures are typically used for eliminating trivial subgoals represented for instance as sequents modulo a background theory. These theories axiomatize standard data-types such as arrays, lists, bit-vectors and have proved to be quite useful for, e.g., hardware verification. Elimination of trivial sequents often reduces to the **problem of proving the unsatisfiability of conjunctions of literals modulo a background theory** T, which is the problem we shall consider here.

The rewriting approach permits us the uniform design of decision procedures for eliminating these subgoals and also offers an efficient alternative to congruence closure techniques. This approach was inspired by Greg Nelson's thesis [Nel81] where it is suggested to apply Knuth-Bendix completion to derive decision procedures. Here, instead of the Knuth-Bendix completion procedure, we apply a standard (refutation) complete superposition-based inference system for clausal equational logic (given for instance in [NR01]). This allows not only to handle pure equality but also several interesting axiomatic theories that were not handled previously that way such as lists, arrays, and extensional arrays. The proof that the decision procedures are correct is straightforward w.r.t. to other correctness proofs available in literature. To see this, consider the decision procedure for an extensional theory of arrays in [SDBL01]. The key ingredient is a modified congruence closure algorithm which is capable of handling "partial equations". The correctness proof is rather complex and it takes the main part of the paper; it is model-theoretic and rather *ad-hoc*. In [ARR01] (see Section 4 for some hints), we design a decision procedure for the same theory which is simpler to understand since it amounts to apply standard equality reasoning in contrast to handling partial equalities. Furthermore, our proof of correctness relies on basic properties of first order logic.

Finally, we notice that in our framework decision procedures for combinations of theories can be readily derived in a manner closely resembling the combination schema described in [NO78]. This is exemplified for a combination of the theory of lists and arrays in [ARR01]. (Notice that non-convex theories [NO78], such as the theory of arrays, are smoothly handled in our framework.)

2 Our two step methodology

Let T be a first-order theory. In the following, with literal we mean an expression of the form $l \bowtie r$, where l and r are two ground terms and $\bowtie$ is either $=$ or $\neq$ ($l = r$ and $l \neq r$ are called equality and disequality, respectively). We are concerned with the *problem of checking whether a set S of ground literals is T-satisfiable*, i.e. if $S \cup T$ is satisfiable.

If T is the quantifier-free theory of equality (which we denote with $\mathcal{E}$), there are many available solutions to solve this problem ranging from congruence closure algorithms to ground completion. Unfortunately, there are no general and uniform solutions when T is a theory extending $\mathcal{E}$. Nonetheless, some *ad-hoc* solutions for certain theories are available in literature. For example in [NO80], Nelson and Oppen describe a decision procedure for the "quantifier-free theory of LISP list structure". The procedure is obtained as an extension of a congruence closure algorithm with a mechanism which augments the graph by selected instances of the axioms of the theory. The proof of correctness is model theoretic and seems difficult to generalize. A discussion of the difficulties to derive a general method to obtain decision procedures by extending congruence closure algorithms as well as a decision procedure for the theory of arrays (without extensionality) can be found in [Nel81]. This discussion has motivated our work.

Let S be the input set of ground literals and $Ax(T)$ be a presentation of a theory T extending $\mathcal{E}$. Our **goal** is to derive a satisfiability decision procedure capable of establishing whether S is T-satisfiable or not. The **key idea** underlying our methodology is to tune a general *refutational complete superposition inference system* derived from [Rus91,BG94], which we call $\mathcal{SP}$, in order to obtain *termination* for the theory T. The proposed methodology consists of the following two steps.

Step 1: flattening. A flat literal is a literal of the form $l \bowtie r$, where l and r are terms of depth at most 1. The first step of our methodology is to flatten all the literals in S by introducing a finite number of "fresh" constants so to obtain a set S' of (ground) flat literals s.t. S' is T-satisfiable iff S is. As an example, consider the set of literals $\{f(c, c') = h(h(a)), h(h(h(a))) \neq a\}$. It can be flattened to the set $\{f(c, c') = h(c_1), c_3 \neq a\} \cup \{c_1 = h(a), c_3 = h(c_2), c_2 = h(c_1)\}$, where c_1, c_2, and c_3 are "fresh" constants.

Step 2: applying the superposition calculus $\mathcal{SP}$. We will make use of $\mathcal{SP}$ comprising the inference rules of Table 1 and the simplification rules of Table 2. $\mathcal{SP}$ is taken 'off-the-shelf' from [NR01]. It extends the system from [Rus91] by the *equality factoring rule* [BG94], so that more ordering restrictions are possible (in the non-Horn case). The binary relation $\succ$ is a reduction ordering [DJ90], which is total on ground terms. $\succ$ is extended to equational literals in the following way: $(a \bowtie b) \succ (c \bowtie d)$ if $\{a, b\} \succ\!\!\succ \{c, d\}$, where $\succ\!\!\succ$ is the multiset extension of $\succ$ and $\bowtie$ is either $=$ or $\neq$. Multisets of literals are compared using the multiset extension of $\succ$ on literals. The calculus $\mathcal{SP}$ is refutationally complete for general first-order equational logic [BG94,NR01]. A *saturation* of a set of clauses by $\mathcal{SP}$ is the final set of clauses

Name	Rule	Conditions
Superposition	$\dfrac{\Gamma \Rightarrow \Delta, l[u'] = r \quad \Pi \Rightarrow \Sigma, u = v}{\Gamma, \Pi \Rightarrow \Delta, \Sigma, l[v] = r}$	$u \not\preceq v$, $l[u'] \not\preceq r$, $*$
Paramodulation	$\dfrac{\Gamma, l[u'] = r \Rightarrow \Delta \quad \Pi \Rightarrow \Sigma, u = v}{l[v] = r, \Gamma, \Pi \Rightarrow \Delta, \Sigma}$	$u \not\preceq v$, $l[u'] \not\preceq r$, $*$
Reflection	$\dfrac{\Gamma, u' = u \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$	$(u' = u) \not\prec (\Gamma \cup \Delta)$
Factoring	$\dfrac{\Gamma \Rightarrow \Delta, u = t, u' = t'}{(\Gamma, t = t' \Rightarrow \Delta, u = t')}$	$u \not\preceq t$, $u \not\preceq \Gamma$, $(u = t) \not\prec \{u' = t'\} \cup \Delta$

Legenda:

- σ is the m.g.u. of the terms u and u' and it is applied to both the premises and the conclusion of each rule (it is omitted for simplicity);
- a *clause* is a disjunction of literals, i.e. an expression of the form $\neg A_1 \vee \cdots \vee \neg A_n \vee B_1 \vee \cdots \vee B_m$ (abbreviated with $A_1, \ldots, A_n \Rightarrow B_1, \ldots, B_m$) where $A_1, \ldots, A_n$, $B_1, \ldots, B_m$ are equalities ($n \geq 0$ and $m \geq 0$);
- Γ, Δ, Π, and Σ denote multisets of equalities;
- $*$ stands for$(u = v) \not\preceq (\Pi \cup \Sigma)$, $(l[u'] = r) \not\prec (\Gamma \cup \Delta)$.

Fig. 1. Inference rules of $\mathcal{SP}$

Name	Rule	Conditions
Subsumption	$\dfrac{S \cup \{C, C'\}}{S \cup \{C\}}$	if for some θ, $\theta(C) = C'$, and for no ρ, $\rho(C') = C$
Simplification	$\dfrac{S \cup \{C[\theta(l)], l = r\}}{S \cup \{C[\theta(r)], l = r\}}$	$\theta(l) \succ \theta(r)$, $C[\theta(l)] \succ (\theta(l) = \theta(r))$
Deletion	$\dfrac{S \cup \{\Gamma \Rightarrow \Delta, t = t\}}{S}$	

Legenda:

- C and C' denote clauses whereas S denotes a set of clauses;
- θ and ρ denote substitutions.

Fig. 2. Simplification rules of $\mathcal{SP}$

generated by a fair derivation from S using rules in $\mathcal{SP}$ with higher priority given to the simplification rules. *If the saturation terminates for the union of T and any set of ground flat literals, then it is a decision procedure for T.* More precisely, if the final set of clauses contains the empty clause then the input set of literals is unsatisfiable; it is satisfiable, otherwise. (This is a direct consequence of the refutational completeness of $\mathcal{SP}$.) We shall call $\mathcal{SP}$ any fair application of the inference system with priority given to the simplification rules.

3 Our methodology at work: the theory of lists

Let $\mathcal{L}$ be the theory of lists whose language contains the binary function symbol cons, and the unary function symbols car and cdr. Let $Ax(\mathcal{L}) := \{\mathsf{car}(\mathsf{cons}(x, y)) =$

$x, \mathsf{cdr}(\mathsf{cons}(x, y)) = y\}$.[1] In the following, S denotes a set of ground literals already flattened. Then, the second step of our methodology amounts to analyzing the possible inferences in $\mathcal{SP}$ which are applicable to elements of $S \cup Ax(\mathcal{L})$.

Lemma 1. *The clauses occurring in the saturations of $S \cup Ax(\mathcal{L})$ by $\mathcal{SP}$ can only be the empty clause, ground flat literals, or the equalities in $Ax(\mathcal{L})$.*

Proof sketch. The proof is by induction on the length of the derivations. No inference between axioms in $Ax(\mathcal{L})$ is possible. Thus, by inspection of the rules in $\mathcal{SP}$, there are four cases to consider. The most interesting one is the following. A *Superposition* between an equality in $Ax(\mathcal{L})$ and a ground flat equality of the form $\mathsf{cons}(c_1, c_2) = c_3$ (where c_1, c_2, and c_3 are individual constants) only yields ground flat equalities. The remaining cases are obvious.

¿From the analysis of the possible inference, we can derive termination.

Lemma 2. *All the saturations of $S \cup Ax(\mathcal{L})$ by $\mathcal{SP}$ are finite.*

Proof sketch. It is sufficient to observe that the saturations of $S \cup Ax(\mathcal{L})$ by $\mathcal{SP}$ can only contain the empty clause or a finite number of ground flat literals for Lemma 1.

Finally, from termination, fairness, and the refutation completeness of $\mathcal{SP}$, we derive the following theorem.

Theorem 1. *$\mathcal{SP}$ is a decision procedure for $\mathcal{L}$.*

4 A decision procedure for arrays

We briefly overview how straightforward is to derive a decision procedure for the theory of arrays with extensionality in our framework. The key idea is to take into account the axiom of extensionality by a suitable transformation of the input set of literals and to apply the two step methodology described in Section 2 on the transformed set of input literals and the remaining axioms.

Let $\mathcal{A}_e^s$ be the many-sorted theory containing the sorts ELEM, INDEX, and ARRAY, the function symbols store and select of arity $\text{ARRAY}, \text{INDEX}, \text{ELEM} \longrightarrow \text{ARRAY}$ and $\text{ARRAY}, \text{INDEX} \longrightarrow \text{ELEM}$ respectively, and the set $Ax(\mathcal{A}_e^s)$ of axioms:

$$\mathsf{select}(\mathsf{store}(a, i, e), i) = e \tag{1}$$

$$i \neq j \Rightarrow \mathsf{select}(\mathsf{store}(a, i, e), j) = \mathsf{select}(a, j) \tag{2}$$

$$\forall i.(\mathsf{select}(a, i) = \mathsf{select}(b, i)) \Rightarrow a = b \tag{3}$$

where a and b are variables of sort ARRAY, i and j are variables of sort INDEX. Let $\Sigma_{\mathcal{A}_s^e}$ denote a signature containing the function symbols select, store, and a finite set of constants for each sort. Finally, let $Ax(\mathcal{A}^s)$ denote the set containing only axioms (1) and (2).

[1] Here and in the remainder of the paper, we omit the flattening step for simplicity.

Lemma 3. *Let S be a set of literals (built out of $\Sigma_{\mathcal{A}_s^e}$) and let S' be obtained from S by replacing all the inequalities of the form $t \neq t'$ with $\exists i.\mathsf{select}(t,i) \neq \mathsf{select}(t',i)$, where t and t' are terms of sort* ARRAY. *Then S is $\mathcal{A}_e^s$-satisfiable iff S' is $\mathcal{A}^s$-satisfiable.*

The proof of this Lemma is reported in [ARR01].

Theorem 2. *Let S be a set of literals (built out of $\Sigma_{\mathcal{A}_s^e}$) and let S' be obtained from S by replacing all the inequalities of the form $t \neq t'$ with $\mathsf{select}(t, sk(t,t')) \neq \mathsf{select}(t', sk(t,t'))$, where t and t' are terms of sort* ARRAY, *and sk is a Skolem function of arity* ARRAY, ARRAY $\longrightarrow$ INDEX. *Then S is $\mathcal{A}_e^s$-satisfiable iff S' is $\mathcal{A}^s$-satisfiable.*

This Theorem follows from Lemma 3 and basic properties of skolemization.

At this point, we are left with the problem of checking the satisfiability of a set of ground literals w.r.t. the theory presented by $Ax(\mathcal{A}^s)$. This can be done straightforwardly with a decision procedure obtained by applying the two step methodology described in Section 2 (see [ARR01] for details).

References

[ARR01] A. Armando, S. Ranise, and M. Rusinowitch. Uniform Derivation of Decision Procedures by Superposition. In *CSL'01*, 2001. To appear.

[BG94] L. Bachmair and H. Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *J. of Logic and Comp.*, 4(3):217–247, 1994.

[DJ90] N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Hand. of Theoretical Comp. Science*, pages 243–320. 1990.

[Nel81] G. Nelson. Techniques for Program Verification. Technical Report CSL-81-10, Xerox Palo Alto Research Center, June 1981.

[NO78] G. Nelson and D.C. Oppen. Simplification by Cooperating Decision Procedures. Technical Report STAN-CS-78-652, Stanford CS Dept., April 1978.

[NO80] Greg Nelson and Derek C. Oppen. Fast decision procedures based on congruence closure. *Journal of the ACM*, 27(2):356–364, 1980.

[NR01] R. Nieuwenhuis and A. Rubio. Paramodulation-based theorem proving. In A. Robinson and A. Voronkov, editors, *Hand. of Automated Reasoning*. 2001.

[Rus91] M. Rusinowitch. Theorem-Proving with Resolution and Superposition. *J. of Symbolic Computation*, (1-2), 1991.

[SDBL01] A. Stump, D. L. Dill, C. W. Barrett, and J. Levitt. A Decision Procedure for an Extensional Theory of Arrays. In *LICS'01*, 2001. To appear.