Combination of Theories 2014

Jonathan R. J. Kolberg[1]

[1] Technische Universität Kaiserslautern
Kaiserslautern, Deutschland
j_kolberg11@cs.uni-kl.de

Abstract

This paper gives an overview over combination of theories. We derive the requirements for the combination to be decidable. We also describe an algorithm to decide formulas that comply with the requirements. We show that the algorithm is correct and sound. We also introduce performance improvements. Further we give ideas to overcome the limits of the algorithm as it is described.

1998 ACM Subject Classification F.4.1 [*Mathematical Logic*]: Computational logic; Verification; I.2.3 [*Deduction and Theorem Proving*]: Deduction; B.6.3 [*Design Aids*]: Verification

Keywords and phrases Theory, Verification, SAT solvers, Satisfiability Modulo Theories

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

The work presented in this paper belongs to algorithmic analysis. The algorithmic analysis is divide into two parts the analysis of the data, normally analysed with logical formulas, and the control-flow, analysed by automata theoretical tools. We focus onto the analysis of the data and thus onto logical formulas. In a normal program we want to be able to reason about e.g. arrays with integers. For integers and for arrays there exist theories with decision procedures, but how to decide a formula over the combined theory we answer in this paper. Our paper derives necessary requirements for the decidability of such combined theories. We also present a algorithm that decides such theories.

The example on which we discover how this algorithm works is the formula $1 \leq x \wedge x \leq 2 \wedge f(1) \neq f(x) \wedge f(x) \neq f(2)$ and the combined theory of integers and uninterpreted functions.

The tool we present for this cause is the Nelson-Oppen method. We split our problem into smaller problems and reconstruct the answer for the big problem out of the smaller problems. This technique is widely known as divide and conquer. But as one might notice we do not use a standard divide and conquer algorithm instead we will input further information into the recursion.

1.1 Related Work

The base of all the work related to the Nelson-Oppen method is the first paper of Nelson and Oppen [7], which had a flawed proof of the correctness in it. Only one year later Oppen presented a corrected version of the proof in [8] and other four years later Nelson in [6]. Oppen also proved complexity results for the Nelson-Oppen method in [8]. Tinelli and Harandi present an alternative proof of correctness in [9]. The correctness proof we present derives for the correctness proof in [3]. Other notable presentations of the Nelson-Oppen method are found in [5] and in [1].

Conference title on which this volume is based on.
Editors: Billy Editor and Bill Editors; pp. 1–11
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

2 Preliminaries

Before we get to the Nelson-Oppen method we first define a few things. A *first-order theory* $\mathcal{T}$ is a set of closed first-order formulas, which is closed under first-order consequences. We say a set A are the *axioms* of $\mathcal{T}$ iff ($A \vDash B$ and B is closed) $\Leftrightarrow B \in \mathcal{T}$. A theory *has equality* (or is a theory *with equality*) if its axioms imply reflexivity, symmetry and transitivity of equality. The *pure equality* fragment of a theory with equality is composed of formulas that are possibly quantified Boolean combinations of equalities between variables.

We call a formula F $\mathcal{T}$*-satisfiable* if $F \wedge T$ is satisfiable in the first-order sense. For two formulas F and G we say F entails G in $\mathcal{T}$, written as $F \vDash_{\mathcal{T}} G$, iff $F \wedge \neg G$ is $\mathcal{T}$-unsatisfiable. We say two formulas F and G are $\mathcal{T}$*-equivalent* or $\mathcal{T}$-equisatisfiable iff $F \vDash_{\mathcal{T}} G$ and $G \vDash_{\mathcal{T}} F$.

3 Nelson-Oppen Method

Many formulas of interest are not formulas over a single first-order theory, so the question arises if satisfiability of such formulas is decidable. This section introduces an decision procedure for to this.

The main idea of the Nelson-Oppen method is to first divide the given formula into two formulas, one over each signature. The problem is now divided into two and we can use the two given decision procedures. In the second step we re-establish the communication between the two procedures and solve the problem with the help of these procedures.

The formal description of the problem is the following

Given: Quantifier-free formula F over $\bigcup_{i=0}^{n} \Sigma_i$ and theories $\mathcal{T}_1, \ldots, \mathcal{T}_n$ with decision procedure P_i and the theories satisfy the Nelson-Oppen criteria

Problem: Is F satisfiable in the combined theory of $\mathcal{T}_1, \ldots, \mathcal{T}_n$?

Without lose of generality we restrict our view onto the Nelson-Oppen method, to the combination of two theories. Since with an inductive use of the method for two theories it is possible to combine an arbitrary number of theories. Also we restrict our view, such that F is a conjunct of terms. Again this is not a lose of generality, because we can check the every disjunct of the DNF of F. Out of the Nelson-Oppen criteria, the first and second criterion is intuitive and the third is needed for correctness. We will have a further look onto the criteria in Section 4. The soundness and completeness is proven in Section 5.

The Nelson-Oppen method is divided into two steps.

1. Variable abstraction
2. Guess and check

Where the first step divides F into two formulas F_1 and F_2, where F_1 is over Σ_1 and F_2 over Σ_2. The main point of this step is that F is $(\mathcal{T}_1 \cup \mathcal{T}_2)$-equisatisfiable to $F_1 \wedge F_2$.

The second step encodes the relation of F_1 and F_2 into a new formula and decides if $F_1 \wedge F_2$ is $(\mathcal{T}_1 \cup \mathcal{T}_2)$-satisfiable with the help of this new formula and the decision procedures of the two theories.

We will cover the two steps of the Nelson-Oppen method in detail now and will orientate our presentation on [3].

3.1 Variable abstraction

Since we want to divide our formula F into two formulas over disjunct signatures, we want to get a notion of belonging to a signature. Our first step towards this goal is to define a function gs (get signature), which tells us the signature of a term, predicate or equality. For simplicity we will further refer to terms, predicates and equalities as *expressions*.

▶ **Definition 1** (gs)**.** The signatures Σ_i, given through the input of the method, look like $(\mathrm{Func}_i, \mathrm{Pred}_i)$. gs, standing for ground signature, is defined as follows

- $\mathrm{gs}(f(\tilde{t})) = i,$ where $f \in \mathrm{Func}_i$
- $\mathrm{gs}(p(\tilde{t})) = i,$ where $f \in \mathrm{Pred}_i$
- $\mathrm{gs}(t_1 = t_2) = \mathrm{gs}(t_1)$

Now we have a formal notion of the root symbol of a term. The real problem of the variable abstraction lies within expressions that contain functions and predicates of different signatures. To formally get a notion of this, we first define what a subexpression is.

▶ **Definition 2** (expression)**.** A *expression* e of the signature $\Sigma = (Func, Pred)$ and variables V is defined inductively as

$$e ::= p(t_1, \dots, t_n) \,|\, f(t_1, \dots, t_m) \,|\, t_1 = t_2, \qquad \text{where } p/_k \in Pred \text{ and } f/_m \in Func$$

$$t ::= x \,|\, f(t_1, \dots, t_k), \qquad \text{where } x \in V \text{ and } f/_k \in Func$$

An *subexpression* of e is any t_i on the right hand side of the e definition, that occurs within e.

We want to talk about the problematic expressions and want a formal notion of problematic expressions.

▶ **Definition 3** (problematic expression & number of problems)**.** A problematic expression e is a expression where there exists a subexpression e_s such that $\mathrm{gs}(e) \neq \mathrm{gs}(e_s)$. In other words there exists a subexpression e_s in e such that e is not an expression over $\Sigma_{\mathrm{gs}(e_s)}$.

The number of problems in a formula is the number of choices we have for subexpressions e_s of problematic expressions within this formula.

Our final goal was to provide two formulas, each containing only expression over one signature. To reach that goal we first have to handle problematic expressions. For this we will define a function φ, which reduces the number of problems in the formula. This is done through substitution of the subexpression of one subexpression with a fresh variable and additionally constrain this variable to be the subexpression.

Formally let $\varphi(F)$ be the function which replaces a problematic expression e of F with $e[e_s/x_f]$ and appends the conjunct $x_f = e_s$ where x_f is a fresh variable in F. If there exists no problematic expression in F it just returns F unmodified.

Finally the variable abstraction is the sorting of conjuncts into two formulas F_1 and F_2. Each conjunct C_j of the least fixed point of φ on F is sorted into F_i iff $\mathrm{gs}(C_j) = i$.

To illustrate the first step we will have a look onto an example.

▶ **Example 4.** Consider $(\Sigma_E \cup \Sigma_{\mathbb{Z}})$ formula.

$$\begin{aligned} F :=& 1 \leq x \wedge x \leq 2 \wedge f(1) \neq f(x) \wedge f(x) \neq f(2) \\ \varphi(F) =& 1 \leq x \wedge x \leq 2 \wedge f(y_1) \neq f(x) \wedge f(x) \neq f(2) \wedge y_1 = 1 \\ \varphi(\varphi(F)) =& 1 \leq x \wedge x \leq 2 \wedge f(y_1) \neq f(x) \wedge f(x) \neq f(y_2) \wedge y_1 = 1 \wedge y_2 = 2 \\ \varphi^3(F) =& \varphi^2(F) \end{aligned}$$

So the formulas constructed are:

$$\begin{aligned} F_{\mathbb{Z}} :=& 1 \leq x \wedge x \leq 2 \wedge y_1 = 1 \wedge y_2 = 2 \\ F_E :=& f(y_1) \neq f(x) \wedge f(x) \neq f(y_2) \end{aligned}$$

F_E and $F_{\mathbb{Z}}$ share the variables x, y_1 and y_2. $F_E \wedge F_{\mathbb{Z}}$ is $(\mathcal{T}_E \cup \mathcal{T}_{\mathbb{Z}})$-equisatisfiable to F.

3.2 Guess and check

The first Phase separates $(\Sigma_1 \cup \Sigma_2)$-formula F into two formulas, Σ_1-formula F_1 and Σ_2-formula F_1. The second phase guesses the relation of F_1 and F_2 non deterministically and uses the decision procedures P_1 and P_2 to check the satisfiability of $F_1 \wedge F_2$. Since we want to talk about the set of shared variables between F_1 and F_2, we introduce $V := \text{shared}(F_1, F_2) := \text{free}(F_1) \cap \text{free}(F_2)$.

We note that $F_1 \wedge F_2$ is satisfiable iff there exists a equivalence relation E over the shared variables of F_1 and F_2 such that F_1 together with this equivalence relation is $\mathcal{T}_1$-satisfiable and F_2 together with E is $\mathcal{T}_2$-satisfiable. One might ask why there exists such an E. This is the case because the model for $F_1 \wedge F_2$ has an equivalence relation embed. This Model is also a model for F_1 and F_2 and thus also for F_1, respectively F_2 together with this equivalence relation.

Let E be an equivalence relation over V. The *arrangement* $\text{arr}(V, E)$ of V induced by E is the formula

$$\text{arr}(V, E) := \bigwedge_{u,v \in V : uEv} u = v \wedge \bigwedge_{u,v \in V : \neg(uEv)} u \neq v$$

which forces the variables related by E to be equal and those not related by E to not be equal.

Our big picture tells us, that formula F is $(\mathcal{T}_1 \cup \mathcal{T}_2)$-satisfiable iff there exists an equivalence relation E of V such that

- $F_1 \wedge \text{arr}(V, E)$ is $\mathcal{T}_1$-satisfiable, and
- $F_2 \wedge \text{arr}(V, E)$ is $\mathcal{T}_2$-satisfiable

Otherwise, F is $(\mathcal{T}_1 \cup \mathcal{T}_2)$-unsatisfiable.

▶ **Example 5.** Let us revisit Example 4.

$$F :\equiv 1 \leq x \wedge x \leq 2 \wedge f(1) \neq f(x) \wedge f(x) \neq f(2)$$

Phase 1 separates this formula into Σ_E-formula

$$F_E \equiv 1 \leq x \wedge x \leq 2 \wedge y_1 = 1 \wedge y_2 = 2$$

and the $\Sigma_{\mathbb{Z}}$-formula

$$F_{\mathbb{Z}} \equiv f(y_1) \neq f(x) \wedge f(x) \neq f(y_2)$$

with

$$V = \text{shared}(F_{\mathbb{Z}}, F_E) = \{x, y_1, y_2\}.$$

There are 5 equivalence relations to consider, of which we will only cover two since the others are similar to these two.

1. $\{\{x, y_1, y_2\}\}$, i.e., $x = y_1 = y_2$: $F_E \wedge \text{arr}(V, E)$ is $\mathcal{T}_E$-unsatisfiable because item cannot be the case that both $x = y_1$ and $f(x) \neq f(y_1)$
2. $\{\{x\}, \{y_1, y_2\}\}$, i.e., $x \neq y_1, y_1 = y_2$: $F_{\mathbb{Z}} \wedge \text{arr}(V, E)$ is $\mathcal{T}_{\mathbb{Z}}$-unsatisfiable because it cannot be the case that both $y_1 = y_2$ and $y_1 = 1 \wedge y_2 = 2$

Practical Efficiency

The second Phase is formulated as "guess and check". First, guess an equivalence relation E, then check the induced arrangement. Unfortunately the number of possible equivalence relations is given by the sequence of *Bell numbers*, which grows super-exponentially. This problem is addressed by the variants of the Nelson-Oppen method in 6.

4 Nelson-Oppen Criteria

Now we have a look at the requirements for the Nelson-Oppen method.

▶ **Definition 6** (Nelson-Oppen criteria)**.** The (first-order) theories $\mathcal{T}_1, \dots, \mathcal{T}_n$ with signatures $\Sigma_1, \dots, \Sigma_n$ satisfy the Nelson-Oppen criteria when

1. $\bigcap\limits_{i=1}^{n} \Sigma_i = \emptyset$
2. T_i has a decision procedure P_i
3. the $\mathcal{T}_i$s are stably infinite

▶ **Definition 7** (stably infinite)**.** A theory $\mathcal{T}$ with signature[1] S is called *stably infinite* if for every quantifier free S-formula F, if F is $\mathcal{T}$-satisfiable, then there exists a $\mathcal{T}$-interpretation that satisfys F and has a domain with infinite cardinality.

The first Nelson-Oppen criteria is necessary, because otherwise it is not possible to know which theory a function or predicate belongs to. This is essential for the variable abstraction (3.1) of the Nelson-Oppen method.

The second criteria is necessary, because it is easy to understand that a combined theory is only decidable if the theories to be combined are decidable.

The example which shows that the third point of the Nelson-Oppen criteria is necessary, was taken out of [9], but this criteria was not mentioned in the first publication for the Nelson-Oppen method [7].

▶ **Example 8.** Consider a theory $\mathcal{T}_1$ admitting only models of cardinality at most 2 and a signature-disjoint theory $\mathcal{T}_2$ admitting models of each cardinality. Assume that f is a functor of $\mathcal{T}_1$ and g is a functor on $\mathcal{T}_2$ and neither is defined as a constant function in their respective theory. The union $\mathcal{T}$ of $\mathcal{T}_1$ and $\mathcal{T}_2$. So consider the formula F

$$F :\equiv fx \neq fy \wedge gx \neq gz \wedge gy \neq gz$$

The first step of the Nelson-Oppen method splits this into

$$F_1 :\equiv fx \neq fy \text{ and } F_2 :\equiv gx \neq gz \wedge gy \neq gz$$

Now observe the only possible arrangements are $\{x = y\}$ and $\{x \neq y\}$. Since $L_1 \wedge x = y$ is clearly unsatisfiable, the procedure fails on that arrangement. With the other arrangement both L_i's are satisfiable in their respective theory and so the procedure concludes that F is satisfiable in $\mathcal{T}$, but unfortunately,

$$\mathcal{T} \vDash F \rightarrow (x \neq y \wedge x \neq z \wedge y \neq z)$$

which means that F is unsatisfiable in $\mathcal{T}$ because, as $\mathcal{T}_1$, $\mathcal{T}$ has only models of cardinality less that 3.

[1] we consider $=$ not to be a part of the signature

5 Proof of Soundness and Completeness

This proof is a adaptation of the proof in The Calculus of Computation ([3]).

The main result of this paper is the soundness and completeness of the Nelson-Oppen method.

▶ **Theorem 9** (Sound & Complete). *Consider theories $\mathcal{T}_1$ and $\mathcal{T}_2$ satisfying the Nelson-Oppen criteria*[2]*. For conjunctive quantifier-free Σ_1-formula F_1 and conjunctive quantifier-free Σ_2-formula F_2, $F_1 \wedge F_2$ is $(\mathcal{T}_1 \cup \mathcal{T}_2)$-satisfiable iff there exists an arrangement $K = \mathrm{arr}(\mathrm{shared}(F_1, F_2), E)$ such that $F_1 \wedge K$ is T_1-satisfiable and $F_2 \wedge K$ is T_2-satisfiable.*

Proof of Soundness. Suppose that $F_1 \wedge F_2$ is $(\mathcal{T}_1 \cup \mathcal{T}_2)$-satisfiable with satisfying $(\mathcal{T}_1 \cup \mathcal{T}_2)$-interpretation I. Extract the equivalence relation E from I, such that the arrangement K is satisfied by I. Then $F_1 \wedge K$ and $F_2 \wedge K$ are both satisfied by I, which can be seen as both a $\mathcal{T}_1$-interpretation and a $\mathcal{T}_2$-interpretation, so that they are $\mathcal{T}_1$-satisfiable and T_2-satisfiable. ◀

Proof sketch of Completeness. Let $K = \mathrm{arr}(\mathrm{shared}(F_1, F_2), E)$ be an arrangement such that $F_1 \wedge K$ and $F_2 \wedge K$ are $\mathcal{T}_1$-satisfiable and $\mathcal{T}_2$-satisfiable. Suppose that $F_1 \wedge F_2$ is $(\mathcal{T}_1 \cup \mathcal{T}_2)$-unsatisfiable. We derive a contradiction.

Because $F_1 \wedge F_2$ is $(\mathcal{T}_1 \cup \mathcal{T}_2)$-unsatisfiable, we know that F_1 implies $\neg F_2$ in $\mathcal{T}_1 \cup \mathcal{T}_2$. The *Craig Interpolation Lemma* (Lemma 10) tells us that there is a quantifier-free formula H such that F_1 implies H over all infinite $\mathcal{T}_1$-interpretations (T_1-interpretations with infinite domain) and F_2 implies $\neg H$ over all infinite $\mathcal{T}_2$-interpretations: H interpolates between F_1 and F_2. We than show that the arrangement K implies H, which means that F_2 implies $\neg K$ over all infinite $\mathcal{T}_2$-interpretations. In other words, no infinite $\mathcal{T}_2$-interpretation satisfies $F_2 \wedge K$. Yet if $\mathcal{T}_2$ is stably infinite and $F_2 \wedge K$ is $\mathcal{T}_2$-satisfiable as assumed, then $F_2 \wedge K$ is satisfied by some infinite $\mathcal{T}_2$-interpretation, a contradiction. ◀

In the proof sketch we used the Craig Interpolation Lemma.

▶ **Lemma 10** (Craig Interpolation Lemma). *If $F_1 \Rightarrow F_2$, then there exists a formula H such that $F_1 \Rightarrow H, H \Rightarrow F_2$, and each free variable, function symbol and predicate symbol of H appears in F_1 and F_2.*

Also we said that we want show that the arrangement K implies H, but as we will see in the detailed proof we need the following theorem to show this.

▶ **Lemma 11** (Weak Quantifier Elimination for Pure Equality). *Consider any stably infinite theory $\mathcal{T}$. For each pure equality formula F, there exists a quantifier-free pure equality formula F' such that F is weakly $\mathcal{T}$-equivalent to F'.*

Proof. Consider pure equality formula $\exists x. G[x, \overline{y}]$, where G is quantifier-free with free variables x and $\overline{y}$. Define

$$G_0 := G\{x = x \mapsto true, x = y_1 \mapsto false, \ldots, x = y_n \mapsto false\}$$

and, for $i \in \{1, \ldots, n\}$

$$G_i := G\{x = y_i\}$$

[2] see Definition 4

We claim that $\exists x.G$ is weakly $\mathcal{T}$-equivalent to

$$G' := G \vee G_0 \vee \ldots \vee G_n$$

For G' asserts that x is either equal to some free variable y_i or not. Because we consider only infinite domains, it is always possible for x not to equal any y_i.

We now have a weak quantifier elimination procedure over the pure equality fragment of $\mathcal{T}$. It is weak because equivalence is only guaranteed to hold on infinite interpretations. ◀

Since our proof takes advantage of the stably infinite theories, we want to have a short notation and saying for the implication and equivalence for infinite models only.

▶ **Definition 12.** $F \Rightarrow^* G$ iff $M[\![G]\!] = 1$ for every infinite model of F. Similarly we define $\Leftrightarrow^*$ as an weakening of $\Leftrightarrow$ to only hold on infinite structures.

If $F \Rightarrow^* G$, we say that F *weakly implies* G; if $F \Leftrightarrow^* G$, we say that F is *weakly equivalent* to G.

Now we have all the requirements for the detailed proof of Theorem 9. The soundness was already proven, that leaves only the completeness.

Proof of Completeness. Let $K = \mathrm{arr}(\mathrm{shared}(F_1, F_2), E)$ be an arrangement such that $F_1 \wedge K$ and $F_2 \wedge K$ are $\mathcal{T}_1$-satisfiable and $\mathcal{T}_2$-satisfiable. Suppose that $F_1 \wedge F_2$ is $(\mathcal{T}_1 \cup \mathcal{T}_2)$-unsatisfiable. We derive a contradiction.

Now we will have a look at the details of the proof. Since we are only considering interpretations with infinite domain, we will mostly work with the definitions 12.

Since $F_1 \wedge F_2$ is $(\mathcal{T}_1 \cup \mathcal{T}_2)$-unsatisfiable, the Compactness Theorem tells us that there exists a conjunct S_1 of a finite subset of axioms in $\mathcal{T}_1$ and a conjunct S_2 of a finite subset of axioms in $\mathcal{T}_2$ such that $S_1 \wedge F_1 \wedge S_2 \wedge F_2$ is (first-order) unsatisfiable. Choose S_1 and S_2 to include the axioms that imply reflexivity, symmetry and transitivity of equality. Then rearranging, we get

$$S_1 \wedge F_2 \Rightarrow \neg S_2 \vee \neg F_2 \tag{1}$$

Now with application of *Craig Interpolation Lemma* (Lemma 10) onto the implication 1, there exists an interpolant H' such that $\mathrm{free}(H') = \mathrm{shared}(F_1, F_2)$ and

$$S_1 \wedge F_1 \Rightarrow H' \text{ and } S_2 \wedge H' \Rightarrow \neg F_2$$

The second implication is a simple derivation of $H' \Rightarrow \neg S_2 \vee \neg F_2$. Because all the functions and predicate between $S_1 \wedge F_1$ and $S_2 \wedge F_2$ are disjunct, H' is of a special form: its axioms are equalities between variables of $\mathit{shared}(F_1, F_2)$. However H' may have quantifiers.

As we know from Lemma 11 in fact there exists a quantifier-free interpolant H over $\mathit{shared}(F_1, F_2)$ such that

$$S_1 \wedge F_1 \Rightarrow^* H \text{ and } S_2 \wedge H \Rightarrow^* \neg F_2$$

For the next step, recall from the beginning of the proof that $F_1 \wedge K$ is $\mathcal{T}_1$-satisfiable and $F_2 \wedge K$ is $\mathcal{T}_2$-satisfiable, where K is the arrangement. We thus know that

$$S_1 \wedge F_1 \wedge K \text{ and } S_2 \wedge F_2 \wedge K$$

are (first-order) satisfiable. Moreover, as $\mathcal{T}_1$ and $\mathcal{T}_2$ are stably infinite, each of these formulas has an an interpretation with an infinite domain.

K is a conjunction of equalities and inequalities between pairs of variables of $\text{shared}(F_1, F_2)$. Moreover, by the definition of an arrangement, K is as strong as possible: no additional equality literals L over $\text{shared}(F_1, F_2)$ can be added to K without either K and $K \wedge L$ being equivalent in a theory with equality or $K \wedge L$ being unsatisfiable in a theory with equality. With the help of this observation, we construct a formula K' by conjoining additional equality pairs. For each pair of variables $u, v \in \text{shared}(F_1, F_2)$, conjoin either $u = v$ or $u \neq v$ depending on which keeps the satisfiability of K' in a theory with equality. Now since $S_1 \wedge F_1 \wedge K$ is satisfiable, then so is $S_1 \wedge F_1 \wedge K'$ indeed by the same interpretation.

We claim that the DNF representation of H must include K' or a (conjunctive) sub formula of K' as a disjunct. Suppose that is not the case, then every disjunct of the DNF representation of H contradicts the satisfying interpretations of $S_1 \wedge F_1 \wedge K'$, of which at least one exists. Therefore, $K' \Rightarrow H$, and – because K and K' are equivalent in a theory with equality – $K \Rightarrow H$. In other words, the arrangement K is a special case of the weak interpolant H.

So we have

$$S_2 \wedge H \Rightarrow^* \neg F_2,$$

or rearranged

$$S_2 \wedge F_2 \Rightarrow^* \neg H$$

From $K \Rightarrow H$, we get $\neg H \Rightarrow \neg K$, so

$$S_2 \wedge F_2 \Rightarrow^* \neg K$$

But this contradicts to the satisfiability of $S_2 \wedge F_2 \wedge K$ by some infinite interpretation. Thus $F_1 \wedge F_2$ is actually $(\mathcal{T}_1 \cup \mathcal{T}_2)$-satisfiable, and so the Nelson-Oppen method is correct. ◀

6 Variants for Performance Improvements

The default "Guess and check" of the Nelson-Oppen method is not very efficient. Also the method is limited to conjunctive quantifier-free formulas, but often the opposite kind of formula is checked for satisfiability. To overcome such limitations and to improve the performance there are various way of which we present the most important. We first discuss a performance improvement in 6.1. Then we have a look at ideas to handle quantifier-free formulas in 6.3 and finally two different ideas to handle the quantifiers. Handle them in DPLL[3] in 6.4 and handling them without the need of DPLL in 6.5.

6.1 Equality Propagation

The idea of equality propagation is that the search space for the equality relation E should be shrunk. This is achieved through adding additional equalities that have to hold because of one of the theories. Since we want to have a notion for when this works best we introduce

▶ **Definition 13** (convex theories). The idea of a convex theory is that if a conjunctive formula in a convex theory implies a disjunct of equalities between variables, then it actually implies a single equality.

[3] DPLL is a SAT solver framework

Let F be a conjunctive quantifier-free Σ-formula and $G :\equiv \bigvee_{i=1}^{n} u_i = v_i$ for variables u_i, v_i. The theory $\mathcal{T}$ is *convex* if for every such G and F if $F \Rightarrow G$ then $F \Rightarrow u_j = v_j$ for a $j \in \{1, \ldots, n\}$.

In our new version a central manager asks the decision procedures P_1 and P_2 to report new discovered equalities between shared variables and propagates it to the other decision procedure.

If the already discovered equalities are $\mathcal{E}$ then a decision procedure for a convex theory $\mathcal{T}_i$ discovers a new equality $u = v$ for shared variables u and v iff $F_i \wedge \mathcal{E} \Rightarrow u = v$. The central manager then propagates the new equality to the other decision procedure.

If the theory is not convex it is still possible to discover a new disjunct of equalities $\mathcal{S}$ when $F_i \wedge \mathcal{E} \Rightarrow \bigvee_{u_i = v_i \in \mathcal{S}} u_i = v_i$. In this case the manager has to split the disjunction and search along multiple branches. The search along a branch ends when either a full arrangement is discovered or when all sub-branches end in a contradiction. In the later case the central manger tries an other branch and if non is left to try, the central manager declares the formula to be $(\mathcal{T}_1 \cup \mathcal{T}_2)$- unsatisfiable.

If at some point neither of the decision procedures finds something new, then the central manager concludes that the given formula is $(\mathcal{T}_1 \cup \mathcal{T}_2)$- satisfiable. Because if $\mathcal{E}$ is the set of already learned equalities, $\mathcal{S}$ the set of all possible remaining equalities and $F_1 \wedge \mathcal{E} \not\Rightarrow \bigvee_{u_i = v_i \in \mathcal{S}} u_i = v_i$ and $F_2 \wedge \mathcal{E} \not\Rightarrow \bigvee_{u_i = v_i \in \mathcal{S}} u_i = v_i$, which holds if no new disjunct of equality was discovered, then $F_1 \wedge \mathcal{E} \wedge \bigwedge_{u_i = v_i \in \mathcal{S}} u_i \neq v_i$ and $F_2 \wedge \mathcal{E} \wedge \bigwedge_{u_i = v_i \in \mathcal{S}} u_i \neq v_i$ are $\mathcal{T}_1$-satisfiable and $\mathcal{T}_2$-satisfiable, respectively. So the discovered arrangement is $\operatorname{arr}(V, E) :\equiv \mathcal{E} \wedge \bigwedge_{u_i = v_i \in \mathcal{S}} u_i \neq v_i$, and F is $(\mathcal{T}_1 \cup \mathcal{T}_2)$-satisfiable.

We will now see how this work with the help of a Example 10.15 of [3]

▶ **Example 14.** Consider the formula $F \equiv 1 \leq x \wedge x \leq 3 \wedge f(x) \neq f(1) \wedge f(x) \wedge f(x) \neq f(3) \wedge f(1) \neq f(2)$. After the first phase of the Nelson-Oppen method we get $F_{\mathbb{Z}} \equiv 1 \leq x \wedge x \leq 3 \wedge x_1 = 1 \wedge x_2 = 2 \wedge x_3 = 3$ and $F_E \equiv f(x) \neq f(x_1) \wedge f(x) \neq f(x_3) \wedge f(x_1) \neq f(x_2)$. With the $V = \operatorname{shared}(F_{\mathbb{Z}}, F_E) = \{x, x_1, x_2, x_3\}$.

$P_{\mathbb{Z}}$ concludes from $1 \leq x \leq 3$ that $F_{\mathbb{Z}} \Rightarrow x = x_1 \vee x = x_2 \vee x = x_3$. $\mathcal{T}_{\mathbb{Z}}$ is not convex as noted in 6.1.1, so we have to go through all the cases.

On case $\mathcal{E}_1^a \equiv x = x_1$ P_E finds that $F_E \wedge \mathcal{E}_1^a \Rightarrow \perp$, since $f(x) \wedge f(x_1)$.

On case $\mathcal{E}_1^b \equiv x = x_2$, neither $P_{\mathbb{Z}}$ nor P_E finds any now contradictions or equalities, which means that $K \equiv \mathcal{E}_1^b \wedge x \neq x_1 \wedge x \neq x_3 \wedge x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_2 \neq x_3$ is an arrangement and $F_{\mathbb{Z}} \wedge K$ is $\mathcal{T}_{\mathbb{Z}}$-satisfiable and $F_E \wedge K$ is $\mathcal{T}_E$-satisfiable. Thus F is $(\mathcal{T}_E \cup \mathcal{T}_{\mathbb{Z}})$-satisfiable.

This argument is nicely summarized in Figure 1.

6.1.1 Notes

Many practically relevant theories are not convex, e.g $\mathcal{T}_{\mathbb{Z}}$ and $\mathcal{T}_A$, but $\mathcal{T}_{\mathbb{Q}}$ is convex. For details about this see ([3] 10.3.1).

6.2 DPLL-$\mathcal{T}$

DPLL-$\mathcal{T}$ is an extension of the DPLL SAT solver to be an SMT solver. The idea is to ask the SAT solver for possible satisfying truth-assignments and to then verify them with the actual theory solver. This is illustrated in Figure 2. This enables the us to check arbitrary quantifier-free formulas. For further detail on this see [2].

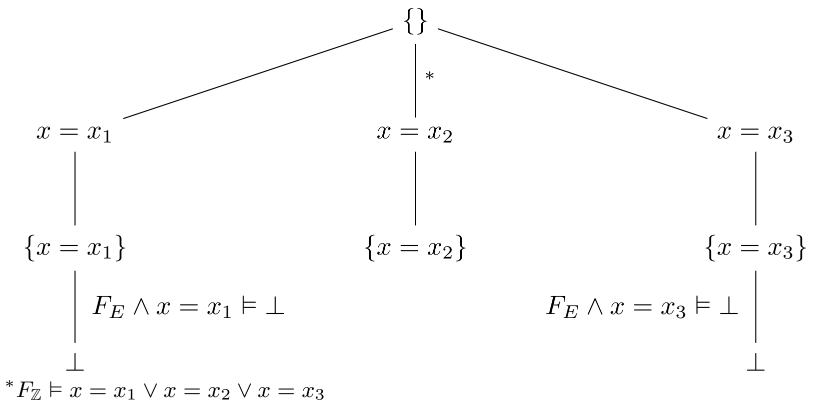
{}
*
$x = x_1$
$x = x_2$
$x = x_3$
$\{x = x_1\}$
$\{x = x_2\}$
$\{x = x_3\}$
$F_E \wedge x = x_1 \vDash \bot$
$F_E \wedge x = x_3 \vDash \bot$
$\bot$
$\bot$
$^*F_{\mathbb{Z}} \vDash x = x_1 \vee x = x_2 \vee x = x_3$

Figure 1 Summery of Example 14

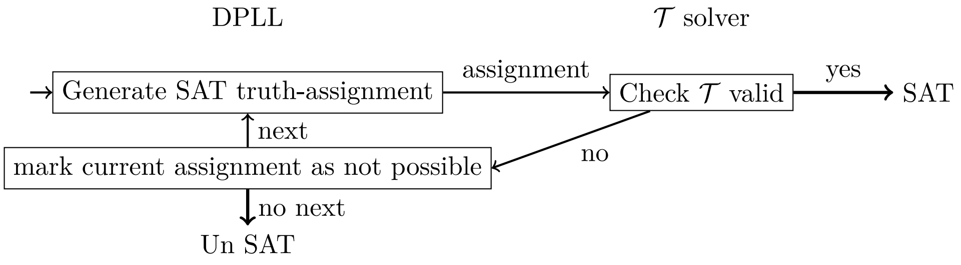
DPLL
$\mathcal{T}$ solver
Generate SAT truth-assignment
assignment
Check $\mathcal{T}$ valid
yes
SAT
next
mark current assignment as not possible
no
no next
Un SAT

Figure 2 An illustration of the DPLL-$\mathcal{T}$ framework

6.3 Delayed Theory Combination

Delayed Theory Combination (DCT) is a method solving the problem of theory combination within the context of lazy SMT. We only present the idea of DCT for detailed information refer to 12.6.3 in [2]. The idea of DTC is to build around DPLL[4]. The DPLL engine does not only enumerate truth-assignments for the atoms of the input formula, but also "nondeterministically guesses" truth values for the truth values the $\mathcal{T}$-solvers are not capable of inferring. It also handles the case-split induced by the entailment of disjunctions of interface equalities in non-convex theories. The reason for doing this is to use the power of a modern DPLL engine by delegating parts of the reasoning effort from the $\mathcal{T}_i$-solvers to the DPLL engine.

6.4 Handling Quantifier

Because the Nelson-Oppen method only works for quantifier-free formulas, SMT solvers have traditionally been limited. The addition of *quantifier instantiation* to DPLL as done in [4] overcomes this limitation. The idea is to extend the transition system of DPLL to include rules for quantifier instantiation. Since a detailed view onto this is out of scope for this paper one can refer to 12.7.2 in [2] for more information.

[4] DPLL is a SAT solver framework

6.5 Quantifier Elimination for Stably Infinite Theories

With the help of Theorem 11 and Chapter 7 of [3] it should be possible to build also an procedure without DPLL.

References

1 Clark Barrett. “Decision Procedures: An Algorithmic Point of View,” by Daniel Kroening and Ofer Strichman, Springer-Verlag, 2008. *Journal of Automated Reasoning*, 51(4):453–456, 2013.

2 Clark W. Barrett, Roberto Sebastiani, Sanjit A. Seshia, and Cesare Tinelli. Satisfiability Modulo Theories. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*, pages 825–885. IOS Press, 2009.

3 Aaron R. Bradley and Zohar Manna. *The Calculus of Computation - Decision Procedures with Applications to Verification.* Springer, 2007.

4 David Detlefs, Greg Nelson, and James B. Saxe. Simplify: a theorem prover for program checking. *J. ACM*, 52(3):365–473, 2005.

5 Zohar Manna and Calogero G. Zarba. Combining Decision Procedures. In Bernhard K. Aichernig and T. S. E. Maibaum, editors, *10th Anniversary Colloquium of UNU/IIST*, volume 2757 of *Lecture Notes in Computer Science*, pages 381–422. Springer, 2002.

6 G. Nelson. Combining Satisfiability Procedures by Equality Sharing. In W. W. Bledsoe and D. W. Loveland, editors, *Contemporary Mathematics: Automated Theorem Proving - After 25 Years*, pages 201–212. American Mathematical Society, Providence, RI, 1984.

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

8 Derek C. Oppen. Complexity, Convexity and Combinations of Theories. *Theor. Comput. Sci.*, 12:291–302, 1980.

9 Cesare Tinelli and Mehdi T. Harandi. A new correctness proof of the Nelson–Oppen combination procedure. In F. Baader and K. U. Schulz, editors, *Frontiers of Combining Systems: Proceedings of the 1st International Workshop (Munich, Germany)*, Applied Logic, pages 103–120. Kluwer Academic Publishers, March 1996.