A Generalization of Shostak's Method for Combining Decision Procedures

Clark W. Barrett, David L. Dill, and Aaron Stump

Stanford University, Stanford, CA 94305, USA,
http://verify.stanford.edu

Abstract. Consider the problem of determining whether a quantifier-free formula ϕ is satisfiable in some first-order theory $\mathcal{T}$. Shostak's algorithm decides this problem for a certain class of theories with both interpreted and uninterpreted functions. We present two new algorithms based on Shostak's method. The first is a simple subset of Shostak's algorithm for the same class of theories but without uninterpreted functions. This simplified algorithm is easy to understand and prove correct, providing insight into how and why Shostak's algorithm works. The simplified algorithm is then used as the foundation for a generalization of Shostak's method based on the Nelson-Oppen method for combining theories.

1 Introduction

In 1984, Shostak introduced a clever and subtle algorithm for deciding the satisfiability of quantifier-free formulas in a combined theory which includes a first-order theory with certain properties and the pure theory of equality with uninterpreted functions [10]. The method has proved to be popular for automated reasoning applications, having been used as the basis for decision procedures found in several tools including PVS [8] STeP [3,5], and SVC [1,2,6].

Unfortunately, the original paper is difficult to follow, due in part to the fact that it contains several errors. As a result, there has been an ongoing effort to understand and clarify the method [4,9,12]. The presentation that is most faithful to Shostak while correcting his errors is that recently produced by Shankar and Ruess [9].

Our work on SVC has led to a number of additional insights which we hope will help to demystify the method and increase its utility. We first present a subset of the original algorithm, in particular, the subset which decides formulas *without* uninterpreted functions. This algorithm is interesting in its own right because it is easily proved correct and can be used directly to produce decision procedures.

The simplified algorithm also forms the basis for a more general algorithm which subsumes Shostak's algorithm. In order to justify the more general algorithm, we use an argument based on a variation of the Nelson-Oppen method for combining theories [7,11] and a new theorem which relates *convexity* (a requirement for Shostak) and *stable-infiniteness* (a requirement for Nelson-Oppen).

In Section 2, below, some preliminary definitions and notation are given. The simple algorithm without uninterpreted functions is presented in Section 3. Section 4 reviews the Nelson-Oppen method in preparation for the generalized algorithm which is presented in Section 5. Finally, Section 6 compares our approach to other work on Shostak's algorithm, and Section 7 concludes.

2 Preliminary Concepts

2.1 Some Notions from Logic

A *theory* is a set of first-order sentences. For the purposes of this paper, all theories are assumed to be first-order and to include the axioms of equality. The *signature* of a theory is the set of function, predicate (other than equality), and constant symbols appearing in those sentences. A *literal* is an atomic formula or its negation. To avoid confusion with the logical equality symbol $=$, we use the symbol $\equiv$ to indicate that two logical expressions are syntactically identical.

For a given model, M, a *variable assignment* ρ is a function which assigns to each variable an element of the domain of M. We write $M \models_\rho \phi$ if ϕ is true in the model M with variable assignment ρ. If Φ is a set of formulas, then $M \models_\rho \Phi$ indicates that $M \models_\rho \phi$ for each $\phi \in \Phi$. In general, whenever sets of formulas are used as logical formulas, the intended meaning is the conjunction of the formulas in the set. A formula ϕ is satisfiable if there exists some model M and variable assignment ρ such that $M \models_\rho \phi$. If Γ is a set of formulas and ϕ is a formula, then $\Gamma \models \phi$ means that whenever a model and variable assignment satisfy Γ, they also satisfy ϕ. A set S of literals is *convex* in a theory $\mathcal{T}$ if $\mathcal{T} \cup S$ does not entail any disjunction of equalities without entailing one of the equalities itself. A theory is convex if every set of literals in the language of the theory is convex.

2.2 Equations in Solved Form

Definition 1. *A set $\mathcal{E}$ of equations is said to be in* solved form *iff the left-hand side of each equation in $\mathcal{E}$ is a variable which appears only once in $\mathcal{E}$. We refer to the variables which appear only on the left-hand sides as* solitary *variables.*

A set $\mathcal{E}$ of equations in solved form defines an idempotent substitution: the one which replaces each solitary variable with its corresponding right-hand side. If S is an expression or set of expressions, we denote the result of applying this substitution to S by $\mathcal{E}(S)$. Another interesting property of equations in solved form is that the question of whether such a set $\mathcal{E}$ entails some formula ϕ in a theory $\mathcal{T}$ can be answered simply by determining the validity of $\mathcal{E}(\phi)$ in $\mathcal{T}$:

Lemma 1. *If $\mathcal{T}$ is a theory with signature Σ and $\mathcal{E}$ is a set of Σ-equations in solved form, then $\mathcal{T} \cup \mathcal{E} \models \phi$ iff $\mathcal{T} \models \mathcal{E}(\phi)$.*

Proof. Clearly, $\mathcal{T} \cup \mathcal{E} \models \phi$ iff $\mathcal{T} \cup \mathcal{E} \models \mathcal{E}(\phi)$. Thus we need only show that $\mathcal{T} \cup \mathcal{E} \models \mathcal{E}(\phi)$ iff $\mathcal{T} \models \mathcal{E}(\phi)$. The "if" direction is trivial. To show the other direction, assume that $\mathcal{T} \cup \mathcal{E} \models \mathcal{E}(\phi)$. Any model of $\mathcal{T}$ can be made to satisfy $\mathcal{T} \cup \mathcal{E}$ by assigning any value to the non-solitary variables of $\mathcal{E}$, and then choosing the value of each solitary variable to match the value of its corresponding right-hand side. Since none of the solitary variables occur anywhere else in $\mathcal{E}$, this assignment is well-defined and satisfies $\mathcal{E}$. By assumption then, this model and assignment also satisfy $\mathcal{E}(\phi)$, but none of the solitary variables appear in $\mathcal{E}(\phi)$, so the initial arbitrary assignment to non-solitary variables must be sufficient to satisfy $\mathcal{E}(\phi)$. Thus it must be the case that every model of $\mathcal{T}$ satisfies $\mathcal{E}(\phi)$ with every variable assignment. □

Corollary 1. *If $\mathcal{T}$ is a consistent theory with signature Σ and $\mathcal{E}$ is a set of Σ-equations in solved form, then $\mathcal{T} \cup \mathcal{E}$ is satisfiable.*

3 Algorithm S1

In this section we present an algorithm, based on a subset of Shostak's algorithm, for deciding satisfiability of quantifier-free formulas in a theory $\mathcal{T}$ which meets certain conditions. We call such a theory a *Shostak* theory.

Definition 2. *A consistent theory $\mathcal{T}$ with signature Σ is a* Shostak *theory if the following conditions hold.*

1. *Σ does not contain any predicate symbols.*
2. *$\mathcal{T}$ is convex.*
3. *There exists a* canonizer *σ, a computable function from Σ-terms to Σ-terms, with the property that $\mathcal{T} \models a = b$ iff $\sigma(a) \equiv \sigma(b)$.*
4. *There exists a* solver *ω, a computable function from Σ-equations to sets of formulas defined as follows:*
   (a) *If $\mathcal{T} \models a \neq b$, then $\omega(a = b) \equiv \{false\}$.*
   (b) *Otherwise, $\omega(a = b)$ returns a set $\mathcal{E}$ of equations in solved form such that $\mathcal{T} \models [(a = b) \leftrightarrow \exists \overline{x}.\mathcal{E}]$, where $\overline{x}$ is the set of variables which appear in $\mathcal{E}$ but not in a or b. Each of these variables must be fresh.*

The requirements given here are slightly different from those given by Shostak and others. These differences are discussed in Section 6 below. In the rest of this section, $\mathcal{T}$ is assumed to be a Shostak theory with canonizer σ and solver ω. As we will show, the solver can be used to convert an arbitrary set of equations into a set of equations in solved form. The canonizer is used to determine whether a specific equality is entailed by a set of equations in solved form, as shown by the following lemma.

Lemma 2. *If $\mathcal{T}$ is a Shostak theory with signature Σ, $\mathcal{E}$ is a set of Σ-equations in solved form and σ is a canonizer for the theory $\mathcal{T}$, then $\mathcal{T} \cup \mathcal{E} \models a = b$ iff $\sigma(\mathcal{E}(a)) \equiv \sigma(\mathcal{E}(b))$.*

Proof. By Lemma 1, $\mathcal{T} \cup \mathcal{E} \models a = b$ iff $\mathcal{T} \models \mathcal{E}(a) = \mathcal{E}(b)$. But $\mathcal{T} \models \mathcal{E}(a) = \mathcal{E}(b)$ iff $\sigma(\mathcal{E}(a)) \equiv \sigma(\mathcal{E}(b))$ by the definition of σ. □

```
S1(Γ, Δ, σ, ω)
1.  ℰ := ∅;
2.  WHILE Γ ≠ ∅ DO BEGIN
3.    Remove some equality a = b from Γ;
4.    a* := ℰ(a); b* := ℰ(b);
5.    ℰ* := ω(a* = b*);
6.    IF ℰ* = {false} THEN RETURN FALSE;
7.    ℰ := ℰ*(ℰ) ∪ ℰ*;
8.  END
9.  IF σ(ℰ(a)) ≡ σ(ℰ(b)) for some a ≠ b ∈ Δ THEN RETURN FALSE;
10. RETURN TRUE;
```

Fig. 1. Algorithm S1: based on a simple subset of Shostak's algorithm.

Algorithm *S1* (shown in Fig. 1) makes use of the properties of a Shostak theory to check the joint satisfiability of an arbitrary set of equalities, Γ, and an arbitrary set of disequalities, Δ, in a Shostak theory with canonizer σ and solver ω. Since the satisfiability of any quantifier-free formula can be determined by first converting it to disjunctive normal form, it suffices to have a satisfiability procedure for a conjunction of literals. Since Σ contains no predicate symbols, all Σ-literals are either equalities or disequalities. Thus, algorithm *S1* is sufficient for deciding the satisfiability of quantifier-free formulas. Termination of the algorithm is trivial since each step terminates and each time line 3 is executed the size of Γ is reduced. The following lemmas are needed before proving correctness.

Lemma 3. *If $\mathcal{T}$ is a theory, Γ and Θ are sets of formulas, and $\mathcal{E}$ is a set of equations in solved form, then for any formula ϕ, $\mathcal{T} \cup \Gamma \cup \Theta \cup \mathcal{E} \models \phi$ iff $\mathcal{T} \cup \Gamma \cup \mathcal{E}(\Theta) \cup \mathcal{E} \models \phi$.*

Proof. Follows trivially from the fact that $\Theta \cup \mathcal{E}$ and $\mathcal{E}(\Theta) \cup \mathcal{E}$ are satisfied by exactly the same models and variable assignments. □

Lemma 4. *If ω is a solver for a Shostak theory $\mathcal{T}$ and Γ is any set of formulas, then for any formula ϕ, and terms a and b,*

$$\mathcal{T} \cup \Gamma \cup \{a = b\} \models \phi \text{ iff } \mathcal{T} \cup \Gamma \cup \omega(a = b) \models \phi.$$

Proof.
$\Rightarrow$: Given that $\mathcal{T} \cup \Gamma \cup \{a = b\} \models \phi$, suppose that $M \models_\rho \mathcal{T} \cup \Gamma \cup \omega(a = b)$. It is easy to see from the definition of ω that $M \models_\rho a = b$ and hence by the hypothesis, $M \models_\rho \phi$.
$\Leftarrow$: Given $\mathcal{T} \cup \Gamma \cup \omega(a = b) \models \phi$, suppose that $M \models_\rho \mathcal{T} \cup \Gamma \cup \{a = b\}$. Then, since $\mathcal{T} \models (a = b) \leftrightarrow \exists \overline{x}.\omega(a = b)$, there exists a modified assignment ρ^* which assigns values to all the variables in $\overline{x}$ and satisfies $\omega(a = b)$ but is otherwise equivalent to ρ. Then, by the hypothesis, $M \models_{\rho^*} \phi$. But the variables in $\overline{x}$ are *new* variables, so they do not appear in ϕ, meaning that changing their values cannot affect whether ϕ is true. Thus $M \models_\rho \phi$. □

Lemma 5. *If Γ, $\{a = b\}$, and $\mathcal{E}$ are sets of Σ-formulas, with $\mathcal{E}$ in solved form, then for every Σ-formula ϕ, $\mathcal{T} \cup \Gamma \cup \{a = b\} \cup \mathcal{E} \models \phi$ iff $\mathcal{T} \cup \Gamma \cup \mathcal{E}^* \cup \mathcal{E}^*(\mathcal{E}) \models \phi$, where $\mathcal{E}^* = \omega(\mathcal{E}(a = b))$.*

Proof.

$$\begin{array}{rcll} \mathcal{T} \cup \Gamma \cup \{a = b\} \cup \mathcal{E} \models \phi & \Leftrightarrow & \mathcal{T} \cup \Gamma \cup \{\mathcal{E}(a = b)\} \cup \mathcal{E} \models \phi & \text{Lemma 3} \\ & \Leftrightarrow & \mathcal{T} \cup \Gamma \cup \mathcal{E}^* \cup \mathcal{E} \models \phi & \text{Lemma 4} \\ & \Leftrightarrow & \mathcal{T} \cup \Gamma \cup \mathcal{E}^* \cup \mathcal{E}^*(\mathcal{E}) \models \phi & \text{Lemma 3} \end{array}$$

□

Lemma 6. *During the execution of algorithm S1, $\mathcal{E}$ is always in solved form.*

Proof. Clearly $\mathcal{E}$ is in solved form initially. Consider one iteration. By construction, a^* and b^* do not contain any of the solitary variables of $\mathcal{E}$, and thus $\mathcal{E}^*$ doesn't either. $\mathcal{E}^*$ is in solved form by the definition of ω. Finally, applying $\mathcal{E}^*$ to $\mathcal{E}$ guarantees that none of the solitary variables of $\mathcal{E}^*$ appear in $\mathcal{E}$, so the new value of $\mathcal{E}$ is also in solved form. □

Lemma 7. *Let Γ_n and $\mathcal{E}_n$ be the values of Γ and $\mathcal{E}$ after the while loop in algorithm S1 has been executed n times. Then for each n, and any formula ϕ, the following invariant holds: $\mathcal{T} \cup \Gamma_0 \models \phi$ iff $\mathcal{T} \cup \Gamma_n \cup \mathcal{E}_n \models \phi$.*

Proof. The proof is by induction on n. For $n = 0$, the invariant holds trivially. Now suppose the invariant holds for some $k \geq 0$. Consider the next iteration.

$$\begin{array}{rcll} \mathcal{T} \cup \Gamma_0 \models \phi & \Leftrightarrow & \mathcal{T} \cup \Gamma_k \cup \mathcal{E}_k \models \phi & \text{Induction Hypothesis} \\ & \Leftrightarrow & \mathcal{T} \cup \Gamma_{k+1} \cup \{a = b\} \cup \mathcal{E}_k \models \phi & \text{Line 3} \\ & \Leftrightarrow & \mathcal{T} \cup \Gamma_{k+1} \cup \mathcal{E}^* \cup \mathcal{E}^*(\mathcal{E}_k) \models \phi & \text{Lemmas 5 and 6} \\ & \Leftrightarrow & \mathcal{T} \cup \Gamma_{k+1} \cup \mathcal{E}_{k+1} \models \phi & \text{Line 7} \end{array}$$

□

Theorem 1. *Suppose $\mathcal{T}$ is a Shostak theory with signature Σ, canonizer σ, and solver ω. If Γ is a set of Σ-equalities and Δ is a set of Σ-disequalities, then $\mathcal{T} \cup \Gamma \cup \Delta$ is satisfiable iff $\mathtt{S1}(\Gamma, \Delta, \sigma, \omega) = \mathtt{TRUE}$.*

Proof. Suppose $\mathtt{S1}(\Gamma, \Delta, \sigma, \omega) = \mathtt{FALSE}$. If the algorithm terminates at line 9, then, $\sigma(\mathcal{E}(a)) \equiv \sigma(\mathcal{E}(b))$ for some $a \neq b \in \Delta$. It follows from Lemmas 2 and 7 that $\mathcal{T} \cup \Gamma \models a = b$, so clearly $\mathcal{T} \cup \Gamma \cup \Delta$ is not satisfiable. The other possibility is that the algorithm terminates at line 6. Suppose the loop has been executed n times and that Γ_n and $\mathcal{E}_n$ are the values of Γ and $\mathcal{E}$ at the end of the last loop. It must be the case that $\mathcal{T} \models a^* \neq b^*$, so $\mathcal{T} \cup \{a^* = b^*\}$ is unsatisfiable. Clearly then, $\mathcal{T} \cup \{a^* = b^*\} \cup \mathcal{E}_n$ is unsatisfiable, so by Lemma 3, $\mathcal{T} \cup \{a = b\} \cup \mathcal{E}_n$ is unsatisfiable. But $\{a = b\}$ is a subset of Γ_n, so $\mathcal{T} \cup \Gamma_n \cup \mathcal{E}_n$ must be unsatisfiable, and thus by Lemma 7, $\mathcal{T} \cup \Gamma$ is unsatisfiable.

Suppose on the other hand that $\mathtt{S1}(\Gamma, \Delta, \sigma, \omega) = \mathtt{TRUE}$. Then the algorithm terminates at line 10. By Lemma 6, $\mathcal{E}$ is in solved form. Let $\overline{\Delta}$ be the disjunction of equalities equivalent to $\neg(\Delta)$. Since the algorithm does not terminate at line 9, $\mathcal{E}$ does not entail any equality in $\overline{\Delta}$. Because $\mathcal{T}$ is convex, it follows that

$\mathcal{T} \cup \mathcal{E} \not\models \overline{\Delta}$. Now, since $\mathcal{T} \cup \mathcal{E}$ is satisfiable by Corollary 1, it follows that $\mathcal{T} \cup \mathcal{E} \cup \Delta$ is satisfiable. But by Lemma 7, $\mathcal{T} \cup \Gamma \models \phi$ iff $\mathcal{T} \cup \mathcal{E} \models \phi$, so in particular $\mathcal{T} \cup \mathcal{E} \models \Gamma$. Thus $\mathcal{T} \cup \mathcal{E} \cup \Delta \cup \Gamma$ is satisfiable, and hence $\mathcal{T} \cup \Gamma \cup \Delta$ is satisfiable. □

3.1 An Example

Perhaps the most obvious example of a Shostak theory is the theory of linear arithmetic with signature $\{0, S, +\}$ (where S is the *successor* function) and domain the real numbers. Terms in this theory can be more conveniently represented by using some standard abbreviations: base 10 numerals instead of repeated applications of successor (i.e. 3 instead of $S(S(S(0)))$), multiplication by a constant instead of repeated applications of $+$ (i.e. $3x$ instead of $x+x+x$). Division by a non-zero constant and the use of unary minus can also be included since equations involving these operations can always be converted into equivalent equations without them.

A simple canonizer for this theory can be obtained by imposing an order on all variables (lexicographic or otherwise), and combining like terms. For example, $\sigma(z+3y-x-5z) \equiv -x+3y+(-4z)$. Similarly, a solver can be obtained simply by solving for one of the variables in an equation.

A well-known method for obtaining a solution to a system of equations in this theory is simply to use Gaussian elimination and back-substitution. Interestingly, by using the solver and canonizer just described, algorithm *S1* actually implements Gaussian elimination with back-substitution.

Consider the following system of equations:

$$\begin{aligned} x + 3y - 2z &= 1 \\ x - y - 6z &= 1 \end{aligned}$$

This system can be represented by a matrix and transformed to reduced row echelon form as follows.

$$\left(\begin{array}{ccc|c} 1 & 3 & -2 & 1 \\ 1 & -1 & -6 & 1 \end{array}\right) \Rightarrow \left(\begin{array}{ccc|c} 1 & 3 & -2 & 1 \\ 0 & -4 & -4 & 0 \end{array}\right) \Rightarrow \left(\begin{array}{ccc|c} 1 & 0 & -5 & 1 \\ 0 & 1 & 1 & 0 \end{array}\right)$$

Compare this with running algorithm *S1* on the same set of equations. The following table shows the values of Γ, $\mathcal{E}$, $\mathcal{E}(a=b)$, and $\mathcal{E}^*$ on each iteration of algorithm *S1* starting with $\Gamma = \{x+3y-2z=1, x-y-6z=1\}$:

Γ	$\mathcal{E}$	$\mathcal{E}(a=b)$	$\mathcal{E}^*$
$x+3y-2z=1$ $x-y-6z=1$	$\emptyset$	$x+3y-2z=1$	$x=1-3y+2z$
$x-y-6z=1$	$x=1-3y+2z$	$1-3y+2z-y-6z=1$	$y=-z$
$\emptyset$	$x=1+5z$ $y=-z$		

The substitution for x in the second iteration corresponds to using x as a pivot variable to produce a zero in the second row of the matrix. Similarly, the last execution of line 7 transforms $x = 1 - 3y + 2z$ into $x = 1 + 5z$, corresponding to the transformation of the first row of the matrix due to back-substitution. Notice that the final solution obtained by algorithm *S1* is the same as that obtained from the matrix in reduced row echelon form.

To make the example a little more interesting, suppose a third equation is added: $2x + 8y - 2z = 3$. Transforming the matrix yields:

$$\left(\begin{array}{rrr|r} 1 & 3 & -2 & 1 \\ 1 & -1 & -6 & 1 \\ 2 & 8 & -2 & 3 \end{array}\right) \Rightarrow \left(\begin{array}{rrr|r} 1 & 3 & -2 & 1 \\ 0 & -4 & -4 & 0 \\ 0 & 2 & 2 & 1 \end{array}\right) \Rightarrow \left(\begin{array}{rrr|r} 1 & 3 & -2 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right)$$

At this point, the last row indicates that the system of equations is unsatisfiable. Suppose that the same new equation is processed by algorithm *S1* . Note that rather than restarting the algorithm, the new equation can be placed in Γ and the algorithm can continue from where it left off. This illustrates a very nice property of the algorithm: it is incremental. If a new equation is added to Γ after some of the equations have already been processed, the algorithm can continue without any difficulty. The result is as follows:

Γ	$\mathcal{E}$	$\mathcal{E}(a = b)$	$\mathcal{E}^*$
$2x + 8y - 2z = 3$	$x = 1 + 5z$ $y = -z$	$2(1 + 5z) + 8(-z) - 2z = 3$	false

The solver detects an inconsistency when it tries to solve the equation obtained after applying the substitution from $\mathcal{E}$. The solver indicates this by returning $\{\text{false}\}$, which results in the algorithm returning `FALSE`.

Finally, suppose that instead of the equation $2x+8y-2z = 3$, the disequality $y + x \neq x - z$ is added. This is handled by line 9 of the algorithm:

$$\begin{aligned} \sigma(\mathcal{E}(y + x)) &\equiv \sigma(-z + 1 + 5z) \equiv 1 + 4z \\ \sigma(\mathcal{E}(x - z)) &\equiv \sigma(1 + 5z - z) \quad \equiv 1 + 4z \end{aligned}$$

Since $y + x \neq x - z \in \Delta$ and $\sigma(\mathcal{E}(y + x)) \equiv \sigma(\mathcal{E}(x - z))$, the algorithm returns `FALSE`.

There is no matrix analog to the case which includes the disequality. Algorithm *S1* may, in fact, properly be viewed as a generalization of Gaussian elimination. Not only can it handle disequalities, but it can also introduce new variables or equations when solving. Also, the set of function symbols can be richer than those provided by a vector space. The key requirement is simply that an appropriate canonizer and solver exist.

3.2 Combining Shostak Theories

As noted by Shostak in his original paper [10], it is often possible to combine two Shostak theories to form a new Shostak theory. A canonizer for the combined theory is obtained simply by composing the canonizers from each individual theory. Furthermore, a solver for the combined theory can often be obtained by repeatedly applying the solver for each theory (treating terms in other theories as variables) until a true variable is on the left-hand side of each equation in the solved form. However, as pointed out in [6] and [9], this is not always possible. We do not address this issue here, but mention it as a question which warrants further investigation.

4 The Nelson-Oppen Combination Method

Nelson and Oppen [7] described a method for combining decision procedures for theories which are stably-infinite and have disjoint signatures. A theory $\mathcal{T}_i$ is stably-infinite if any quantifier-free formula is satisfiable in some model of $\mathcal{T}_i$ iff it is satisfiable in an infinite model of $\mathcal{T}_i$. There have been many detailed presentations of the Nelson-Oppen method. A brief overview based on Tinelli and Harandi's approach [11] is given here.

Suppose $\mathcal{T}_1$ and $\mathcal{T}_2$ are such theories with $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$ (the generalization to more than two theories is straightforward). Let Φ be a set of Σ-literals and suppose we wish to determine the satisfiability of $\mathcal{T} \cup \Phi$. A few more definitions are required before presenting the Nelson-Oppen procedure.

Members of Σ_i, for $i = 1,2$ are called i-symbols. In order to associate all terms with some theory, each variable is also arbitrarily associated with either $\mathcal{T}_1$ or $\mathcal{T}_2$. A variable is called an i-variable if it is associated with $\mathcal{T}_i$ (note that an i-variable is *not* an i-symbol, as it is not a member of Σ_i). A Σ-term t is an i-term if it is an i-variable, a constant i-symbol, or an application of a functional i-symbol. An i-predicate is an application of a predicate i-symbol. An atomic i-formula is an an i-predicate or an equality whose left term is an i-term. An i-literal is an atomic i-formula or the negation of an atomic i-formula. An occurrence of a term t in either a term or a literal is i-alien if it is a j-term, with $i \neq j$ and all of its super-terms (if any) are i-terms. An i-term or i-literal is *pure* if it contains only i-symbols (i.e. its i-alien sub-terms are all variables).

Given an equivalence relation $\sim$, let $dom_\sim$ be the domain of the relation. We define the following sets of formulas induced by $\sim$:

$$\begin{aligned} E_\sim &= \{x = y | x, y \in dom_\sim \text{ and } x \sim y\} \\ D_\sim &= \{x \neq y | x, y \in dom_\sim \text{ and } x \not\sim y\} \\ A_\sim &= E_\sim \cup D_\sim. \end{aligned}$$

If $Ar = A_\sim$ for some equivalence relation $\sim$ with domain $\mathcal{S}$, we call Ar an *arrangement* of $\mathcal{S}$.

The first step in determining the satisfiability of Φ is to transform Φ into an equisatisfiable formula $\Phi_1 \wedge \Phi_2$ where Φ_i consists only of pure i-literals as follows. Let ψ be some i-literal in Φ containing a non-variable i-alien j-term t. Replace all occurrences of t in ψ with a new j-variable z and add the equation $z = t$ to Φ. Repeat until every literal in Φ is pure. The literals can then easily be partitioned into Φ_1 and Φ_2. It is easy to see that Φ is satisfiable if and only if $\Phi_1 \wedge \Phi_2$ is satisfiable.

Now, let $\mathcal{S}$ be the set of all variables which appear in both Φ_1 and Φ_2. A simple version of the Nelson-Oppen procedure simply guesses an equivalence relation $\sim$ on $\mathcal{S}$ nondeterministically, and then checks whether $\mathcal{T}_i \cup \Phi_i \cup A_\sim$ is satisfiable. The correctness of the result is based on the following theorem from [11].

Theorem 2. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two stably-infinite, signature-disjoint theories and let Φ_i be a set of pure i-literals for $i = 1, 2$. Let $\mathcal{S}$ be the set of variables which appear in both Φ_1 and Φ_2. Then $\mathcal{T}_1 \cup \mathcal{T}_2 \cup \Phi_1 \cup \Phi_2$ is satisfiable iff there exists an arrangement Ar of $\mathcal{S}$ such that $\mathcal{T}_i \cup \Phi_i \cup Ar$ is satisfiable for $i = 1, 2$.*

5 Combining Nelson-Oppen and Shostak

5.1 A Variation of the Nelson-Oppen Procedure

The first step in the version of the Nelson-Oppen procedure described above changes the structure and number of literals in Φ. However, it is possible to give a version of the procedure which does not change the literals in Φ. This makes possible the combination of Shostak and Nelson-Oppen described next. First, a few more definitions are needed.

Let v be a mapping such that for $i = 1, 2$, each i-term t is mapped to a *fresh* i-variable $v(t)$. Then, for some formula or term α, define $\gamma_i(\alpha)$ to be the result of replacing all i-alien instances of terms t by $v(t)$. It is easy to see that as a result, $\gamma_i(\alpha)$ is i-pure. Note that since the γ_i operator simply replaces terms with unique place-holders, it is invertible. Also, it distributes over equality and Boolean operators. $\gamma_i(t)$ is a somewhat cumbersome notation, but it allows us to be precise about the notion of treating alien terms as variables which is a key part of Shostak's method for combining theories.

Our variation on the Nelson-Oppen procedure works as follows. Given a set of literals, Φ, first partition Φ into two sets Φ_1 and Φ_2, where Φ_i is exactly the set of i-literals in Φ. Let $\mathcal{S}$ be the set of all terms which are i-alien (for some i) in some literal in Φ or in some sub-term of some literal in Φ. $\mathcal{S}$ will also be referred to as the set of *shared* terms. As before, an equivalence relation $\sim$ on $\mathcal{S}$ is guessed. If $\mathcal{T}_i \cup \gamma_i(\Phi_i \cup A_\sim)$ is satisfiable for each i, then $\mathcal{T} \cup \Phi$ is satisfiable, as shown by the following theorem.

Theorem 3. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two stably-infinite, signature-disjoint theories and let Φ be a set of literals in the combined signature Σ. If Φ_i is the set of all i-literals in Φ and $\mathcal{S}$ is the set of shared terms in Φ, then $\mathcal{T}_1 \cup \mathcal{T}_2 \cup \Phi$ is satisfiable iff there exists an equivalence relation $\sim$ on $\mathcal{S}$ such that for $i = 1, 2$, $\mathcal{T}_i \cup \gamma_i(\Phi_i \cup A_\sim)$ is satisfiable.*

Proof.
$\Rightarrow$: Suppose $M \models_\rho \mathcal{T} \cup \Phi$. Let $a \sim b$ iff $a, b \in \mathcal{S}$ and $M \models_\rho a = b$. Then clearly for $i = 1, 2$, $M \models_\rho \mathcal{T}_i \cup \Phi_i \cup A_\sim$. It is then easy to see that $\mathcal{T}_i \cup \gamma_i(\Phi_i \cup A_\sim)$ is satisfiable by choosing a variable assignment which assigns to each variable $v(t)$

the corresponding value of the term t which it replaces.
$\Leftarrow$: Suppose that for each i, $T_i \cup \gamma_i(\Phi_i \cup A_\sim)$ is satisfiable. Consider $i = 1$. Let Θ_1 be the set of all equations $v(t) = t$, where $t \in \mathcal{S}$ is a 1-term. Consider $\gamma_1(\Theta_1)$. Since γ_1 never replaces 1-terms and each $v(t)$ is a new variable, it follows that $\gamma_1(\Theta_1)$ is in solved form, and its solitary variables are exactly the variables which are used to replace 1-terms. Thus, by Corollary 1, $\mathcal{T}_1 \cup \gamma_1(\Theta_1)$ is satisfiable. Furthermore, since none of the solitary variables of $\gamma_1(\Theta_1)$ appear in $\gamma_1(\Phi_1 \cup A_\sim)$, a satisfiable assignment for $\mathcal{T}_1 \cup \gamma_1(\Theta_1)$ can be constructed from the satisfying assignment for $\mathcal{T}_1 \cup \gamma_1(\Phi_1 \cup A_\sim)$ (which exists by hypothesis) so that the resulting assignment satisfies $\mathcal{T}_1 \cup \gamma_1(\Phi_1 \cup A_\sim \cup \Theta_1)$. Now, note that the equations in Θ_1 are exactly those used if γ_2 is applied to a set of expressions which are 1-pure. Thus $\mathcal{T}_1 \cup \gamma_1(\Phi_1 \cup A_\sim \cup \Theta_1)$ is equisatisfiable with $\mathcal{T}_1 \cup \gamma_1(\Phi_1 \cup \Theta_1) \cup \gamma_2(\gamma_1(A_\sim))$. Applying the same argument with i=2, we conclude that $\mathcal{T}_2 \cup \gamma_2(\Phi_2 \cup \Theta_2) \cup \gamma_1(\gamma_2(A_\sim))$ is satisfiable. But for each i, $\gamma_i(\Phi_i \cup \Theta_i)$ is a set of i-literals. Furthermore, $\gamma_2(\gamma_1(A_\sim))$ is equivalent to $\gamma_1(\gamma_2(A_\sim))$ and is an arrangement of the variables shared by these two sets, so Theorem 2 can be applied to conclude that $\mathcal{T} \cup \Phi \cup \Theta_1 \cup \Theta_2$, and thus $\mathcal{T} \cup \Phi$, is satisfiable. □

5.2 Convexity and Stable-Infiniteness

In order to generalize Shostak's algorithm we use the following result which relates convexity and stable-infiniteness.

Theorem 4. *Every convex first-order theory with no trivial models is stably-infinite.*

Proof. Suppose $\mathcal{U}$ is a first-order theory which is not stably-infinite. Then there exists some quantifier-free set of literals Φ which is satisfiable in a finite model of $\mathcal{U}$, but not in an infinite model of $\mathcal{U}$. Let $\exists\overline{x}\Phi$ be the existential closure of Φ. Then $\exists\overline{x}.\Phi$ is true in some finite model, but not in any infinite model, of $\mathcal{U}$. It follows that $\mathcal{U} \cup \{\exists\overline{x}.\Phi\}$ is a theory with no infinite models. By first-order compactness, there must be some finite cardinality n such that there is a model of $\mathcal{U} \cup \{\exists\overline{x}.\Phi\}$ of cardinality n, but none of cardinality larger than n. Clearly, $\mathcal{U} \cup \Phi$ is satisfiable in some model of size n, but not in any models larger than n. It follows by the pigeonhole principle that if $y_i, 0 \le i \le n$ are fresh variables, then $\mathcal{U} \cup \Phi \models \bigvee_{i \ne j} y_i = y_j$, but because $\mathcal{U}$ has no trivial models (i.e. models of size 1), $\mathcal{U} \cup \Phi \not\models y_i = y_j$ for any i, j with $i \ne j$. Thus, $\mathcal{U}$ is not convex. □

5.3 Combining the methods

Suppose now that $\mathcal{T}_1$ is a Shostak theory and $\mathcal{T}_2$ is a convex theory, neither of which admit trivial models (typically, theories of interest do not admit trivial models, or can be easily modified so that this is the case). The above theorem implies that both theories are also stably-infinite. As a result, Theorem 3 can be applied to decide a combination of the two theories as follows.

Suppose Φ is a set of Σ-literals. As in the previous section, divide Φ into Φ_1 and Φ_2 where Φ_i contains exactly the i-literals of Φ. Let $\mathcal{S}$ be the set of shared terms. By Theorem 3, $\mathcal{T} \cup \Phi_1 \cup \Phi_2$ is satisfiable iff there exists an equivalence relation $\sim$ such that for $i = 1, 2$, $\mathcal{T}_i \cup \gamma_i(\Phi_i \cup A_\sim)$ is satisfiable.

Let Γ be the set of all equalities in Φ_1 and Δ the set of disequalities in Φ_1. Furthermore, let Sat_2 be a decision procedure for satisfiability of literals in $\mathcal{T}_2$:

$$Sat_2(\Phi) = \texttt{TRUE} \textit{ iff } \mathcal{T}_2 \cup \gamma_2(\Phi) \not\models \text{false}.$$

Algorithm *S2* is a modification of algorithm *S1* which accommodates the additional theory $\mathcal{T}_2$. Essentially, the algorithm is identical except for the addition of lines 3 through 5 which check whether Φ_2 is consistent in theory $\mathcal{T}_2$ with an arrangement $A_\sim$. The equivalence relation $\sim$ on $\mathcal{S}$ is defined in such a way that $\mathcal{E}$ is consistent with $A_\sim$ by definition:

$$a \sim b \textit{ iff } a, b \in \mathcal{S} \wedge \sigma(\mathcal{E}(\gamma_1(a))) \equiv \sigma(\mathcal{E}(\gamma_1(b)))$$

Thus, when the algorithm returns TRUE, both Φ_1 and Φ_2 are known to be consistent with the arrangement $A_\sim$.

S2(Γ, Δ, σ, ω, Φ_2, Sat_2)
1. $\mathcal{E} := \emptyset$;
2. WHILE $\Gamma \neq \emptyset$ OR $\neg Sat_2(\Phi_2 \cup A_\sim)$ DO BEGIN
3.   IF $\neg Sat_2(\Phi_2 \cup A_\sim)$ THEN BEGIN
4.   IF $\neg Sat_2(\Phi_2 \cup E_\sim)$ THEN RETURN FALSE;
5.   ELSE choose $a \neq b \in D_\sim$ such that $\neg Sat_2(\Phi_2 \cup E_\sim \cup \{a \neq b\})$;
6.   END ELSE Remove some equality $a = b$ from Γ;
7. $a^* := \mathcal{E}(\gamma_1(a))$; $b^* := \mathcal{E}(\gamma_1(b))$;
8. $\mathcal{E}^* := \omega(a^* = b^*)$;
9.   IF $\mathcal{E}^* = \{\texttt{false}\}$ THEN RETURN FALSE;
10. $\mathcal{E} := \mathcal{E}^*(\mathcal{E}) \cup \mathcal{E}^*$;
11. END
12. IF $a \sim b$ for some $a \neq b \in \Delta$ THEN RETURN FALSE;
13. RETURN TRUE;

Fig. 2. Algorithm S2: a generalization of Shostak's algorithm.

Line 5 requires a little explanation. If the algorithm reaches line 5, it means that $\Phi_2 \cup E_\sim \cup D_\sim$ is not satisfiable in $\mathcal{T}_2$, but $\Phi_2 \cup E_\sim$ is. It follows from convexity of $\mathcal{T}_2$ that there must be a disequality $a \neq b$ in $D_\sim$ such that $\Phi_2 \cup E_\sim \cup \{a \neq b\}$ is not satisfiable in $\mathcal{T}_2$.

It is not hard to see that algorithm *S2* must terminate. This is because each step terminates and in each iteration either the size of Γ is reduced by one or two equivalence classes in $\sim$ are merged. As before, the correctness proof requires a couple of preparatory lemmas.

Lemma 8. *Suppose T_1 is a Shostak theory with signature Σ_1, $\mathcal{E}$ is a set of Σ_1-formulas in solved form, $\mathcal{S}$ is a set of terms, and $\sim$ is defined as above. If $\approx$ is an equivalence relation on $\mathcal{S}$ such that $\mathcal{T}_1 \cup \gamma_1(A_\approx) \cup \mathcal{E}$ is satisfiable, then $E_\sim \subseteq A_\approx$.*

Proof. Consider an arbitrary equation $a = b$ between terms in $\mathcal{S}$. $a = b \in E_\sim$ iff $\sigma(\mathcal{E}(\gamma_1(a))) \equiv \sigma(\mathcal{E}(\gamma_1(b)))$ iff (by Lemma 2) $\mathcal{T}_1 \cup \mathcal{E} \models \gamma_1(a = b)$. So $\gamma_1(a = b)$ must be true in every model and assignment satisfying $\mathcal{T}_1 \cup \mathcal{E}$. In particular, if $\mathcal{T}_1 \cup \gamma_1(A_\approx) \cup \mathcal{E}$ is satisfiable, the corresponding model and assignment must also satisfy $\gamma_1(a = b)$. Since either the equation $a = b$ or the disequation $a \neq b$ must be in $A_\approx$, it must be the case that $a = b \in A_\approx$. Thus, $E_\sim \subseteq A_\approx$. □

Lemma 9. *Let Γ_n and $\mathcal{E}_n$ be the values of Γ and $\mathcal{E}$ after the loop in algorithm S2 has been executed n times. Then for each n, the following invariant holds: $\mathcal{T} \cup \Phi$ is satisfiable iff there exists an equivalence relation $\approx$ on $\mathcal{S}$ such that*

(1) $\mathcal{T}_1 \cup \gamma_1(\Gamma_n \cup \Delta \cup A_\approx) \cup \mathcal{E}_n$ is satisfiable, and
(2) $\mathcal{T}_2 \cup \gamma_2(\Phi_2 \cup A_\approx)$ is satisfiable.

Proof. The proof is by induction on n. For the base case, notice that by Theorem 3, $\mathcal{T} \cup \Phi$ is satisfiable iff there exists an equivalence relation $\approx$ such that (1) and (2) hold with $n = 0$.

Before doing the induction case, we first show that for some fixed equivalence relation $\approx$, (1) and (2) hold when $n = k$ iff (1) and (2) hold when $n = k + 1$. Notice that (2) is independent of n, so it is only necessary to consider (1). There are two cases to consider.

First, suppose that the condition of line 3 is true and line 5 is executed. We first show that (1) holds when $n = k$ iff the following holds:

(3) $\mathcal{T}_1 \cup \gamma_1(\Gamma_{k+1} \cup \Delta \cup A_\approx \cup \{a = b\}) \cup \mathcal{E}_k$ is satisfiable.

Since line 6 is not executed, $\Gamma_{k+1} = \Gamma_k$. The if direction is then trivial since the formula in (1) is a subset of the formula in (3). To show the only if direction, first note that it follows from line 5 that $\mathcal{T}_2 \cup \gamma_2(\Phi_2 \cup E_\sim) \models \gamma_2(a = b)$. But by Lemma 8, $E_\sim \subseteq A_\approx$, so it follows that $\mathcal{T}_2 \cup \gamma_2(\Phi_2 \cup A_\approx) \models \gamma_2(a = b)$. Since either $a = b \in A_\approx$ or $a \neq b \in A_\approx$, it must be the case that $a = b \in A_\approx$ and thus (3) follows trivially from (1). Now, by Lemma 5 (where ϕ is false), (3) holds iff

(4) $\mathcal{T}_1 \cup \gamma_1(\Gamma_{k+1} \cup \Delta \cup A_\approx) \cup \mathcal{E}^*(\mathcal{E}_k) \cup \mathcal{E}^*$ is satisfiable,

where $\mathcal{E}^* = \omega(\mathcal{E}(\gamma_1(a = b)))$. But $\mathcal{E}_{k+1} = \mathcal{E}^*(\mathcal{E}_k) \cup \mathcal{E}^*$, so (4) is equivalent to (1) with $n = k + 1$.

In the other case, line 6 is executed (so that $\Gamma_{k+1} = \Gamma_k - \{a = b\}$). Thus, (1) holds with $n = k$ iff $\mathcal{T}_1 \cup \gamma_1(\Gamma_{k+1} \cup \Delta \cup \{a = b\} \cup A_\approx) \cup \mathcal{E}_k$ is satisfiable, which is equivalent to (3). As in the previous case, it then follows from Lemma 5 that (1) holds at k iff (1) holds at $k + 1$.

Thus, given an equivalence relation, (1) and (2) hold at $k + 1$ exactly when they hold at k. It follows easily that if an equivalence relation exists which satisfies (1) and (2) at k, then there exists an equivalence relation satisfying (1) and (2) at $k + 1$ and vise-versa. Finally, the induction case assumes that that $\mathcal{T} \cup \Phi$ is satisfiable iff there exists an equivalence relation $\approx$ such that (1) and (2) hold at k. It follows from the above argument that $\mathcal{T} \cup \Phi$ is satisfiable iff there exists an equivalence relation $\approx$ such that (1) and (2) hold at $k + 1$. □

Theorem 5. *Suppose $\mathcal{T}_1$ is a Shostak theory with signature Σ_1, canonizer σ, and solver ω, and $\mathcal{T}_2$ is a convex theory with signature Σ_2 disjoint from Σ_1 and satisfiability procedure Sat_2. Suppose also that neither $\mathcal{T}_1$ nor $\mathcal{T}_2$ admit trivial models, and let $T = \mathcal{T}_1 \cup \mathcal{T}_2$ and $\Sigma = \Sigma_1 \cup \Sigma_2$. Suppose Φ is a set of Σ-literals. Let Γ be the subset of Φ which consists of 1-equalities, Δ the subset of Φ which consists of 1-disequalities, and Φ_2 the remainder of the literals in Φ. $\mathcal{T} \cup \Phi$ is satisfiable iff* $\mathtt{S2}(\Gamma, \Delta, \sigma, \omega, \Phi_2, Sat_2) = \mathtt{TRUE}$.

Proof. First note that by the same argument used in Lemma 6, $\mathcal{E}$ is always in solved form.

Suppose $\mathtt{S2}(\Gamma, \Delta, \sigma, \omega, \Phi_2, Sat_2) = \mathtt{FALSE}$. If the algorithm terminates at line 9 or 12, then the proof that Φ is unsatisfiable is the same as that for algorithm *S1* above. If it stops at line 4, then suppose there is an equivalence relation $\approx$ satisfying condition (1) of Lemma 9. It follows from Lemma 8 that $E_\sim \subseteq A_\approx$. But since the algorithm terminates at line 4, $\mathcal{T}_2 \cup \gamma_2(\Phi_2 \cup A_\approx)$ must be unsatisfiable. Thus condition (2) of Lemma 9 cannot hold. Thus, by Lemma 9, $\mathcal{T} \cup \Phi$ is unsatisfiable.

Suppose on the other hand that $\mathtt{S2}(\Gamma, \Delta, \sigma, \omega, \Phi_2, Sat_2) = \mathtt{TRUE}$. By the definition of $\sim$ and Lemma 2, $a = b \in A_\sim$ iff $\mathcal{T}_1 \cup \mathcal{E} \models \gamma_1(a = b)$. It follows from the convexity of $\mathcal{T}_1$ and Corollary 1 that $\mathcal{T}_1 \cup \mathcal{E} \cup \gamma_1(A_\sim)$ is satisfiable. It then follows from the fact that *S2* does not terminate at line 12 (as well as convexity again) that $\mathcal{T}_1 \cup \mathcal{E} \cup \gamma_1(\Delta \cup A_\sim)$ is satisfiable. This is condition (1) of Lemma 9. Condition (2) must hold because the while loop terminates. Thus, by Lemma 9, $\mathcal{T} \cup \Phi$ is satisfiable. □

6 Comparison with Shostak's Original Method

There are two main ways in which this work differs from Shostak's original method, which, as was mentioned, is best represented by Ruess and Shankar in [9]. The first is in the set of requirements a theory must fulfill. The second is in the level of abstraction at which the algorithm is presented.

6.1 Requirements on the Theory

Recall that the definition of a Shostak theory gave four requirements which must be met. The first of these is simply that the theory contain no predicate symbols. This is a minor point that is included simply because Shostak's method does not give any guidance on what to do if the theory includes predicate symbols. One possible solution is to encode predicates as functions, but this only works if the resulting encoding maintains the properties of canonizability and solvability.

The second requirement is that the theory be convex. This may seem overly restrictive since Shostak claims that non-convex theories can be handled [10]. Consider, however, a theory with exactly two elements in its domain and the set of formulas $\{x \neq y, y \neq z, x \neq z\}$. Clearly, this set of equations is unsatisfiable, but even if a canonizer and solver exist for the theory, Shostak's algorithm will fail to detect the inconsistency. Ruess and Shankar avoid this difficulty by restricting their attention to the problem of whether $\mathcal{T} \cup \Gamma \models a = b$ for some set of equalities Γ. However, the ability to solve this problem does not lead to a self-contained decision procedure unless the theory is convex.

The third requirement on the theory is that a canonizer exist. Shostak gave five properties that must be satisfied by the canonizer. We have retained only one of these. In fact, this is the only one that is needed at the level of abstraction of our algorithms. However, efficient implementions typically require the additional properties.

A similar situation arises with the requirements on the solver: only a subset of the original requirements are needed. Note that although we require the set of equalities returned by the solver to be equisatisfiable in *every* model of $\mathcal{T}$, whereas Ruess and Shankar require only that it be equisatisfiable in every σ-model, it is not difficult to show that their requirements on the canonizer imply that every model of $\mathcal{T}$ must be a σ-model.

6.2 Level of Abstraction

Algorithm *S2* looks very different from Shostak's original published algorithm as well as most other published versions, though these are, in fact, closely related to instances of *S2*. The most obvious difference is that while we leave $\mathcal{T}_2$ unspecified, in other work, $\mathcal{T}_2$ is always the theory of pure equality with uninterpreted functions.

Additionally, Shostak incorporates several optimizations, which our algorithms would also benefit from. First of all, notice that Sat_2 need only be called if $A_\sim$ changes. This can easily be tracked by maintaining a mapping from each term $\mathtt{t} \in \mathcal{S}$ to $\sigma(\mathcal{E}(\gamma_1(\mathtt{t})))$.

Another optimization is to attempt to reduce the number of shared terms, thus reducing the size of the arrangement $A_\sim$. Rather than precomputing the set, it can be computed incrementally as follows. Initially the set of shared terms $\mathcal{S}$ is the empty set. Then, before an equation is processed in line 7, each sub-term t in the equation is considered. If it can be replaced with a term u already in $\mathcal{S}$ which is known to be equivalent, then all instances of t are replaced with u. Otherwise if t is i-alien for some i, it is added to $\mathcal{S}$. It is clear that in an actual implementation, it is desirable to have such optimizations. However, such details naturally complicate the presentation and proof.

7 Conclusions and Future Work

We have presented a simplified explanation of Shostak's algorithm, omitting uninterpreted functions. It was then shown that by using the same reasoning as that used to justify the Nelson-Oppen combination procedure, any convex theory which satisfies the criteria for the Nelson-Oppen procedure can be combined with the simple Shostak algorithm.

It is our hope that the insights presented in this paper will serve as a foundation for greater understanding and application of cooperating decision procedures.

Acknowledgments

This work was partially supported by the National Science Foundation Grant CCR-9806889, and ARPA/AirForce contract number F33615-00-C-1693.

References

1. Clark Barrett, David Dill, and Jeremy Levitt. Validity Checking for Combinations of Theories with Equality. In M. Srivas and A. Camilleri, editors, *Formal Methods in Computer-Aided Design*, volume 1166 of *Lecture Notes in Computer Science*, pages 187–201. Springer-Verlag, 1996.
2. Clark W. Barrett, David L. Dill, and Aaron Stump. A Framework for Cooperating Decision Procedures. In *17th International Conference on Automated Deduction*, Lecture Notes in Computer Science. Springer-Verlag, 2000.
3. N. Bjorner. *Integrating Decision Procedures for Temporal Verification*. PhD thesis, Stanford University, 1999.
4. D. Cyrluk, P. Lincoln, and N. Shankar. On Shostak's Decision Procedure for Combinations of Theories. In M. McRobbie and J. Slaney, editors, *13th International Conference on Computer Aided Deduction*, volume 1104 of *Lecture Notes in Computer Science*, pages 463–477. Springer-Verlag, 1996.
5. Z. Manna et al. STeP: Deductive-Algorithmic Verification of Reactive and Real-time Systems. In *8th International Conference on Computer-Aided Verification*, volume 1102 of *Lecture Notes in Computer Science*, pages 415–418. Springer-Verlag, 1996.
6. J. Levitt. *Formal Verification Techniques for Digital Systems*. PhD thesis, Stanford University, 1999.
7. G. Nelson and D. Oppen. Simplification by Cooperating Decision Procedures. *ACM Transactions on Programming Languages and Systems*, 1(2):245–57, 1979.
8. S. Owre, J. Rushby, and N. Shankar. PVS: A Prototype Verification System. In D. Kapur, editor, *11th International Conference on Automated Deduction*, volume 607 of *Lecture Notes in Artificial Intelligence*, pages 748–752. Springer-Verlag, 1992.
9. H. Ruess and N. Shankar. Deconstructing Shostak. In *16th Annual IEEE Symposium on Logic in Computer Science*, pages 19–28, June 2001.
10. R. Shostak. Deciding Combinations of Theories. *Journal of the Association for Computing Machinery*, 31(1):1–12, 1984.
11. C. Tinelli and M. Harandi. A New Correctness Proof of the Nelson-Oppen Combination Procedure. In F. Baader and K. Schulz, editors, *1st International Workshop on Frontiers of Combining Systems (FroCoS'96)*, volume 3 of *Applied Logic Series*. Kluwer Academic Publishers, 1996.
12. A. Tiwari. *Decision Procedures in Automated Deduction*. PhD thesis, State University of New York at Stony Brook, 2000.