Combination of Convex Theories: Modularity, Deduction Completeness, and Explanation

Duc-Khanh Tran, Christophe Ringeissen, Silvio Ranise, Hélène Kirchner

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Combination of Convex Theories: Modularity, Deduction Completeness, and Explanation

Duc-Khanh Tran — Christophe Ringeissen — Silvio Ranise — Hélène Kirchner

N° 6688

Octobre 2008

Thème SYM

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--6688--FR+ENG

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche
NANCY - GRAND EST

Combination of Convex Theories: Modularity, Deduction Completeness, and Explanation

Duc-Khanh Tran[*], Christophe Ringeissen[†], Silvio Ranise[‡] , Hélène Kirchner[§]

Thème SYM — Systèmes symboliques
Équipe-Projet Cassis

Rapport de recherche n° 6688 — Octobre 2008 — 34 pages

Abstract: Decision procedures are key components of theorem provers and constraint satisfaction systems. Their modular combination is of prime interest for building efficient systems, but their effective use is often limited by poor interface capabilities, when such procedures only provide a simple "sat/unsat" answer. In this paper, we develop a rule-based framework to design cooperation schemas between such procedures while maintaining modularity of their interfaces. First, we use the rule-based framework to specify and prove the correctness of classic combination schemas by Nelson-Oppen and Shostak. Second, we introduce the concept of deduction complete satisfiability procedures, we show how to build them for large classes of theories, then we provide a schema to modularly combine them. Third, we consider the problem of modularly constructing explanations for combinations by re-using available proof-producing procedures for the component theories.

Key-words: Decision Procedure, Satisfiability Modulo Theories, Combination, Deduction Completeness, Conflict Set

Preliminary versions of the results in this paper appeared in [32, 24, 25, 33].

[*] Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany. Email: tran@mpi-inf.mpg.de

[†] Email: Christophe.Ringeissen@loria.fr

[‡] Email: Silvio.Ranise@loria.fr

[§] Centre de Recherche INRIA Bordeaux - Sud-Ouest, Bâtiment A29, 351, Cours de la Libération 33405 Talence, France. Email: Helene.Kirchner@inria.fr

Centre de recherche INRIA Nancy – Grand Est
LORIA, Technopôle de Nancy-Brabois, Campus scientifique,
615, rue du Jardin Botanique, BP 101, 54602 Villers-Lès-Nancy
Téléphone : +33 3 83 59 30 00 — Télécopie : +33 3 83 27 83 19

Combinaison de théories convexes: modularité, complétude de déduction et explication

Résumé : Les procédures sont des composants essentiels des prouveurs de théorèmes et des systèmes permettant de décider la satisfiabilité de contraintes. Leur combinaison modulaire est du plus grand intérêt pour construire des systèmes efficaces, mais leur utilisation est souvent limitée par une interface pauvre se limitant à une réponse de la forme “sat/unsat”. Dans ce papier, nous développons un cadre à base de règles pour la conception de schémas de coopération de telles procédures tout en maintenant la modularité de leurs interfaces. Dans un premier temps, nous utilisons ce cadre à base de règles pour spécifier et prouver la correction des schémas de combinaison classiques de Nelson-Oppen et Shostak. Ensuite, nous introduisons le concept de procédures de satisfiabilité complètes vis-à-vis de la déduction, nous montrons comment les construire pour une large classe de théories et nous donnons un schéma pour les combiner de façon modulaire. Finalement, nous considérons le problème de la construction modulaire d'explications en cas d'insatisfiabilité, pour des mélanges de théories, grâce à la réutilisation de procédures engendrant des preuves pour les théories composant le mélange.

Mots-clés : procédure de décision, satisfiabilité, complétude de déduction, explication de l'insatisfiabilité

1 Introduction

Decision procedures and constraint solvers are key components in many systems, such as automated theorem provers, expert systems, and constraint logic programming (CLP) environments. Their interest is to significantly augment the degree of automation of the overall system, thereby reducing user's interaction. Indeed, integrating such reasoning components requires some ingenuity as the problems tackled by complex systems are usually (i) large, (ii) expressed over several domains, and (iii) the computed solutions may require some form of certification (e.g., for safety critical applications). In order to overcome these difficulties, we consider in this paper the following two issues:

1. the combination of decision procedures for signature-disjoint and convex theories, to address (ii),
2. the modularity of the computation of conflict sets or explanations of the results for such decision procedures, to address (i) and (iii).

In this work, we restrict our attention to convex theories, i.e. theories for which it is not necessary to derive disjunctions of equalities to ensure the completeness of the combination schemas.[1] The reason for this choice is twofold. On one hand, restricting to convex theories simplifies the technical developments and allows us to present the key ideas in a straightforward way. On the other hand, the recent trend to develop decision procedures for non-convex theories consists of integrating several complex reasoning modules. This is illustrated by the works of [8] or [21] for the (non-convex) theory of arrays where suitable instantiation strategies are used to reduce the satisfiability problem for arrays to several satisfiability problems in the theory of equality, which is again convex. In the light of this observation, we believe that focusing on convex theories is not too restrictive.

Let us explain in more details our contributions to each issue.

Issue 1: combination. Research on the combination of decision procedures has been independently started in the early 80's by [28] and [37] for unions of theories with disjoint signatures. Each combination schema makes different assumptions on the properties the theories to be combined should satisfy. The former requires *NO* theories to have a satisfiability procedure and to be such that a satisfiable formula in a component theory T is also satisfiable in an infinite model of T (*stable-infiniteness*). The latter assumes that *SH* theories admit procedures for reducing terms to canonical form (*canonizers*) and algorithms for solving equations (*solvers*). A series of papers ([12, 34, 4, 23, 20, 11, 35, 10, 26]) have clarified the subtle issues of combining *SH* theories by studying their relationships with *NO* theories. Some of them use pseudo-code to describe the combination algorithms while others adopt a more abstract rule-based presentation.

The first contribution of this paper (Section 3) is to provide a synthesis of Nelson-Oppen and Shostak approaches to disjoint combination by using a rule-based approach in which many recent results are recast and proved correct in a uniform, rigorous, and simple way. Our rational reconstruction proceeds as

[1] See Section 2.2 for a precise definition of convexity.

follows. First, we recall that *SH* theories are contained in the class of convex *NO* theories. According to this abstract classification, three possible scenarios are to be considered when combining two theories: (a) both are *NO* theories (Section 3.1), (b) both are *SH* theories (Section 3.2), and (c) one is a *SH* and the other is a *NO* theory (Section 3.3). We formalize the combination schema for each scenario as an inference system. The applicability conditions of the inference rules are derived from the properties of the theories being combined. Along the lines of [20, 26, 10], the combination schema for (b) is obtained as a refinement of that for (a). The inference system formalizing the combination schema for (c), already considered in [4], is obtained by modularly reusing those for (a) and (b) in a natural and straightforward way. Our synthesis of combination schemas serves two purposes. First, although the results are not new, we believe that presenting them in a uniform framework can provide a valuable reference for people interested in combination problems, especially for non-experts of the field. Second, it can serve as starting point for further investigations. As an example, a problem of great importance when combining *SH* theories is the lack of modularity for solvers (see [11]): no general method exists to produce a solver for the union of *SH* theories from the solvers of the component theories. Furthermore, as it is well known (see e.g. [15]), to implement the Nelson-Oppen combination method efficiently, the satisfiability procedure for the component theories must be capable of deriving the formulas to exchange with other procedures. This is not obvious for satisfiability procedures in general since they may be incomplete for consequence finding, i.e. there is no guarantee that a formula which is a logical consequence of a set of literals will be eventually derived without resorting to guessing and refutation[2]. The lack of modularity for *SH* theories, together with the observation that the theory of equality (ubiquitous in virtually any application where combinations of decision procedures are needed) is not a *SH* theory, but admits an efficient algorithm to derive entailed equalities, seem to suggest a possible line of investigation. Any *ad hoc* combination schema for scenario (c) constitutes a reasonable trade-off between efficiency and generality: solvers and canonizers for *SH* theories efficiently derive new equalities and cooperate *à la* Nelson-Oppen. By investigating this question in our framework, we propose the concept of *deduction completeness* which constitutes **the second contribution** of this paper (Section 4). Intuitively, a deduction complete satisfiability procedure is a satisfiability procedure defined as an inference-based system with the capability of computing all the entailed elementary equalities with no overhead. We show that deduction complete inference-based satisfiability procedures can be constructed in a modular way (Section 4.2). Another interesting feature is that they can be *efficiently built* by reusing a wealth of existing techniques such as canonizers and solvers for *SH* theories and rewriting techniques, as advocated in [22, 3, 1, 27] for theories which do not admit a solver (Section 4.1).

Issue 2: modular computation of explanations. To efficiently and correctly incorporate decision procedures into deduction systems or constraint programming environments, the capability of explaining the results of the decision procedures is crucial. For example, conflict sets (explanation of unsatisfiability) are useful to prune the search space of Satisfiability Modulo Theories (SMT) solvers (see, e.g., [36]) or to direct backtracking in CLP systems ([9]), whereas

[2] A set of literals S entails a formula ϕ iff $S \cup \{\neg\phi\}$ is unsatisfiable.

explanations can be used to safely import the results of external reasoning modules (e.g., decision procedures for selected theories or unification algorithms) in skeptical proof assistants ([19]). While there has been some work on extending decision procedures for some theories, mainly equality, ([18, 13, 29, 38]), there is no paper, to the best of our knowledge, on the modular construction of conflict sets in unions of theories. Many SMT systems have implemented this capability somehow, but no one has offered a high level description of how this is done. So, implementors, when they want to build such a capability in their own SMT tool, are required to understand the code of other systems (and in many cases the code is not even available), abstract away unimportant implementation details, and finally adapt the ideas to their architecture.

The third contribution of this paper (Section 5) is to provide an abstract account of how to extend the Nelson-Oppen combination schema to build a satisfiability procedure capable of producing conflict sets in the union of theories T_1 and T_2, whenever the satisfiability procedures for T_1 and T_2 provide some interface capabilities. To this end, we first introduce the concept of *explanation graph* (Section 5.2), a data structure which compactly encodes the fact that a certain equality between variables (called elementary equality) is a logical consequence of a set of elementary equalities. Explanation graphs can be easily implemented by using efficient algorithms based on the Union-Find data structure of [39, 16]. Then we show how to derive *explanation engines* from satisfiability procedures that produce conflict sets in the union of the component theories. We also introduce the concept of *quasi-conflict set*, which allows us to precisely characterize a (weak) form of minimality satisfied by the explanations computed by our combination method.

2 Background

2.1 First-order theories

We assume the usual first-order syntactic notions of signature, term, position, and substitution. Let Σ be a first-order signature containing only function symbols with their arity and $\mathcal{X}$ a set of variables. A 0-ary function symbol is called a *constant.* A Σ-*term* is a first-order term built out of the symbols in Σ and the variables in $\mathcal{X}$. We use the standard notion of substitution and denote them by the greek letter σ. We write substitution applications in postfix notation, i.e. $t\sigma$ for a term t and a substitution σ. The set of variables occurring in a term t is denoted by $Var(t)$. If l and r are two Σ-terms, then $l = r$ is a Σ-*equality* and $\neg(l = r)$ (also written as $l \neq r$) is a Σ-*disequality.* If p is an n-ary predicate in Σ and $t_1, \ldots, t_n$ are Σ-terms, then $p(t_1, \ldots, t_n)$ is a Σ-*atom.* A Σ-literal is either a Σ-equality or a Σ-disequality or a Σ-atom or a negation of a Σ-atom. A Σ-*formula* is built in the usual way out of the universal and existential quantifiers, Boolean connectives, and symbols in Σ. A *clause* is a disjunction of literals. A *unit* clause is a clause with only one disjunct, equivalently a literal. The *empty* clause is the clause with no disjunct, equivalently an unsatisfiable formula. If φ is a formula, then $Var(\varphi)$ denotes the set of free variables in φ. We call a formula *ground* if it has no variable, and a *sentence* if it has no free variables. Substitution applications are extended to arbitrary first-order formulas, and are written in postfix notation, i.e. $\varphi\sigma$ for a formula φ and a substitution σ. A term is *flat* if its depth is 0 or 1. For a literal, $depth(l \bowtie r) = depth(l) + depth(r)$, where $\bowtie \in \{=, \neq\}$. A positive literal is *flat* if its depth is 0 or 1. A negative literal is *flat* if its depth is 0. A (dis)equality between two free constants is called *elementary*. A literal which is neither an elementary equality nor an elementary disequality is called *non-elementary*. In the following, φ or Φ denotes an arbitrary set of literals, Ω denotes a set of non-elementary literals, E denotes a set of elementary equalities, and Δ denotes a set of elementary disequalities. E^* is the set of all equalities derived from E by reflexivity, symmetry and transitivity. The set E of elementary equalities is *minimal* iff $E'^* \subset E^*$, for any $E' \subset E$.

We also assume the usual first-order notions of interpretation, satisfiability, validity, logical consequence, and theory, as given for instance in [17]. We write $\models^{\alpha}_{\mathcal{M}} \varphi$ when the Σ-formula φ is true in the Σ-structure $\mathcal{M}$ under the variable assignment α. We also say that α satisfies φ in $\mathcal{M}$. A Σ-formula φ is *valid* in a Σ-structure $\mathcal{M}$, denoted by $\mathcal{M} \models \varphi$, if $\models^{\alpha}_{\mathcal{M}} \varphi$ for any assignment of variables α. A *first-order theory* is a set of first-order sentences. A Σ*-theory* is a theory all of whose sentences have signature Σ. All theories we consider are first-order theories *with equality*, which means that the equality symbol $=$ is always interpreted as the identity relation. A Σ-structure $\mathcal{M}$ is a *model* of a Σ-theory T if every sentence in T is true in $\mathcal{M}$. A theory is *consistent* if it admits a model and *trivial* if the cardinality of each of its models is one. In this paper, we restrict ourselves to non-trivial and consistent theories and will consider a few particular theories: the theory of equality $\mathcal{E}$ whose signature contains a finite set of function and constant symbols, and the equality symbol $=$; the quantifier-free fragment of Linear Rational Arithmetic denoted $\mathcal{LA}^{\leq}$ and its restriction to equalities or disequalities denoted $\mathcal{LA}$. A Σ-formula φ is *valid in* T, denoted by $T \models \varphi$, if it is valid in any model of T. A Σ-formula is T*-satisfiable* if it is satisfiable in a model of T. Two Σ-formulas φ and ψ are *equisatisfiable in* T if for every model $\mathcal{M}$ of T, φ is satisfiable in $\mathcal{M}$ iff ψ is satisfiable in $\mathcal{M}$. The *satisfiability problem* for a theory T amounts to establishing whether any given finite quantifier-free conjunction of literals (or equivalently, any given finite set of literals) is T-satisfiable or not. A *satisfiability procedure* for T is any algorithm that solves the satisfiability problem for T.[3] Note that we can use free constants instead of variables to equivalently redefine the satisfiability problem for T as the problem of establishing the consistency of $T \cup S$ for a finite set S of ground literals.

Given an inference system R composed of inference rules, the binary relation $\vdash_{\mathsf{R}}$ is defined on formulas as follows: $\Phi \vdash_{\mathsf{R}} \Phi'$ if Φ' can be derived from Φ by applying a rule in R. A formula Φ' is said T*-equivalent* to Φ if $Var(\Phi') \supseteq Var(\Phi)$ and $T \models (\Phi \Leftrightarrow \exists \tilde{y}.\Phi')$ where $\tilde{y} = Var(\Phi') \backslash Var(\Phi)$. The reflexive and transitive closure of $\vdash_{\mathsf{R}}$, denoted by $\vdash^*_{\mathsf{R}}$, is called the *derivation relation* of R. Also, a *derivation* in R is a sequence $\Phi \vdash_{\mathsf{R}} \Phi' \vdash_{\mathsf{R}} \Phi'' \vdash_{\mathsf{R}} \cdots$. A formula Φ is in *normal form w.r.t.* $\vdash_{\mathsf{R}}$ if there is no derivation in R starting from Φ. The relation $\vdash^*_{\mathsf{R}}$ is *terminating* if there is no infinite derivation.

With such inference systems, it is convenient to identify a conjunctive formula with the set of its conjuncts and to group together specific literals. So

[3] The satisfiability of any quantifier-free formula can be reduced to the satisfiability of sets of literals by converting to disjunctive normal form and then splitting on disjunctions, i.e. checking whether $S_1 \vee S_2$ (where S_1 and S_2 are conjunction of literals) is T-satisfiable reduces to checking the T-satisfiability of either S_1 or S_2.

the inference rules in the following will be applied on so-called *configurations* which are sets of formulas of the form $\Phi; \Phi'$ where Φ and Φ' are unions of literals (identified with their conjunction). Whenever needed, Φ may be written as Γ, Δ in order to emphasize that Γ is a set (a conjunction) of equalities, and Δ is a set (conjunction) of disequalities.

2.2 Combination of theories

Let Σ_1 and Σ_2 be two disjoint signatures (i.e. $\Sigma_1 \cap \Sigma_2 = \emptyset$) and T_i be a Σ_i-theory for $i = 1, 2$. A $\Sigma_1 \cup \Sigma_2$-term t is an *i-term* if it is a variable or it has the form $f(t_1, ..., t_n)$, where f is in Σ_i (for $i = 1, 2$ and $n \geq 0$). Notice that a variable is both a 1-term and a 2-term. A non-variable subterm s of an i-term is *alien* if s is a j-term, and all superterms of s are i-terms, where $i, j \in \{1, 2\}$ and $i \neq j$. An i-term is *i-pure* if it does not contain alien subterms. A literal is i-pure if it contains only i-pure terms. A formula is said to be *pure* if there exists $i \in \{1, 2\}$ such that every term occurring in the formula is i-pure.

In this paper, we consider the problem of solving the satisfiability problem for $T_1 \cup T_2$ (i.e. the problem of checking the $T_1 \cup T_2$-satisfiability of conjunctions of $\Sigma_1 \cup \Sigma_2$-literals) by using the satisfiability procedures for T_1 and T_2. For certain theories, more basic algorithms exist which can be used to build satisfiability procedures, e.g. canonizers and solvers for the class of Shostak theories (see below for a formal definition). When such algorithms exist for either T_1, T_2, or both, we are interested in using them to solve the satisfiability problem for $T_1 \cup T_2$. In order to know which basic algorithms are available for T_1 and T_2 and what are the assumptions on T_1 and T_2, the following notions and results are useful.

Let us remind that a conjunction Φ of Σ-literals is *convex* in a Σ-theory T iff for any disjunction $\bigvee_{i=1}^n x_i = y_i$ (where x_i, y_i are variables and $i = 1, ..., n$) we have: $T \cup \Phi \models \bigvee_{i=1}^n x_i = y_i$ iff $T \cup \Phi \models x_i = y_i$, for some $i \in \{1, ..., n\}$. A Σ-theory T is *convex* iff all conjunctions of Σ-literals are convex.

A Σ-theory T is *stably-infinite* (and called a **SI**-theory, for short) iff for any T-satisfiable Σ-formula φ, there exists a model of T whose domain is infinite and which satisfies φ. A *Nelson-Oppen* theory is a stably-infinite theory which admits a satisfiability algorithm. A **C**-theory is a convex theory. A **CSI**-theory is a convex stably-infinite theory. The class of **C**-theories (resp. **SI**-theories, **CSI**-theories) is denoted by **C** (resp. **SI**, **CSI**).

A *solver* for a Σ-theory T is a function (denoted *solve*) which takes as input a Σ-equality $s = t$ and such that (a) $solve(s = t)$ returns $false$, if $T \models s \neq t$, or (b) $solve(s = t)$ returns a substitution $\sigma = \{x_1 \to t_1, ..., x_n \to t_n\}$ such that (b.1) x_i is a variable occurring in s or t for $i = 1, ..., n$, (b.2) x_i does not occur in any t_j for $i, j = 1, ..., n$, and (b.3) $T \models s = t \Leftrightarrow \exists \tilde{y}. \bigwedge_{i=1}^n x_i = t_i$, where $\tilde{y}$ denotes the "fresh" variables $y_1, ..., y_m$ ($m \geq 0$) such that y_k does not occur in s or t, for all $k = 1, ..., m$. A conjunction of Σ-equalities is in *solved form* iff it has the form $\bigwedge_{i=1}^n x_i = t_i$, which will be denoted by $\widehat{\sigma}$, where $\sigma = \{x_1 \to t_1, ..., x_n \to t_n\}$ is the substitution returned by *solve*.

A *canonizer* for a Σ-theory T is an idempotent function (denoted *canon*) from Σ-terms to Σ-terms such that $T \models a = b$ iff $canon(a) = canon(b)$.

A *Shostak* theory is a convex theory which admits a solver and a canonizer. A **SH**-theory is a stably-infinite Shostak theory. The class of **SH**-theories is denoted by **SH**. Notice that $\mathcal{LA}$ is a **SH**-theory. We assume **SH**-theories to be stably-infinite since this is necessary to combine them with other theories as suggested by many recent papers (see e.g. [26]). This is not too restrictive since, as shown in [4], any convex theory with no trivial models is stably-infinite.

Proposition 1 $\mathbf{SH} \subseteq \mathbf{CSI} \subseteq \mathbf{SI}$.

3 Rational reconstruction of combination schemas

Let T_i be a Σ_i-theory $(i = 1, 2)$ such that $\Sigma_1 \cap \Sigma_2 = \emptyset$. We consider the problem of building a satisfiability procedure for $T_1 \cup T_2$. As a preliminary step, we consider a *purification process* converting any conjunction Φ of $\Sigma_1 \cup \Sigma_2$-literals into a conjunction of pure literals. Such a process is achieved by replacing each alien subterm t by a new variable x and adding the equality $x = t$ to Φ. This mechanism, called *variable abstraction*, is repeatedly applied to Φ until no more alien subterms t can be abstracted away. Obviously, the purification process always terminates yielding $\Phi_1 \wedge \Phi_2$, where Φ_i is a conjunction of Σ_i-literals $(i = 1, 2)$ such that $\Phi_1 \wedge \Phi_2$ and Φ are equisatisfiable in $T_1 \cup T_2$. Without loss of generality, we consider the satisfiability of formulas of the form $\Phi_1 \wedge \Phi_2$ (or, equivalently, of configurations $\Phi_1; \Phi_2$), where Φ_i is a conjunction of i-pure literals.

Our combination schemas are specified by inference systems. To prove that an inference system R yields a satisfiability procedure, we follow a three steps methodology. First, we show that the derivation relation $\vdash_{\mathsf{R}}$ induced by R is terminating. Second, we prove that $\vdash_{\mathsf{R}}$ preserves (un-)satisfiability. Finally, we check that the normal forms defined by $\vdash_{\mathsf{R}}$ (i.e. configurations to which no rule in R can be applied) distinct from $false$ must be satisfiable. The proof of the last step proceeds by contradiction showing that a normal form distinct from $false$ cannot be unsatisfiable by using the following (technical) lemmas from which the proof of correctness of Nelson-Oppen schema in [40] essentially depends.

Let V be a set of variables and E be equivalence relation over V. We define the *arrangements* of V with respect to E, noted $arr(V, E)$, to be the following set of equalities and disequalities: $\{x = y \mid (x, y) \in E\} \cup \{x \neq y \mid (x, y) \in (V \times V) \setminus E\}$.

Lemma 1 *([40]). If T_1 and T_2 are two signature-disjoint theories, then any conjunction $\Phi_1 \wedge \Phi_2$ of pure quantifier-free formulas is $T_1 \cup T_2$-satisfiable if and only if there exists some equivalence relation E over shared variables in $V = Var(\Phi_1) \cap Var(\Phi_2)$ such that $\Phi_i \cup arr(V, E)$ is T_i-satisfiable in a model $\mathcal{M}_i$ (for $i = 1, 2$) and the two models have the same cardinality.*

When the two theories are stably infinite, we get a specialisation of the previous result which is more operational since the requirement on the models cardinality is clearly satisfied.

Lemma 2 *([40]). If T_1 and T_2 are two signature-disjoint stably-infinite theories, then any conjunction $\Phi_1 \wedge \Phi_2$ of pure quantifier-free formulas is $T_1 \cup T_2$-satisfiable if and only if there exists some equivalence relation E over shared variables in $V = Var(\Phi_1) \cap Var(\Phi_2)$ such that $\Phi_i \cup arr(V, E)$ is T_i-satisfiable for $i = 1, 2$.*

Contradiction$_1$

$$\frac{\Phi_1 ; \Phi_2}{false} \textbf{ if } \Phi_1 \text{ is } T_1\text{-unsatisfiable}$$

Deduction$_1$

$$\frac{\Phi_1 ; \Phi_2}{\Phi_1 ; \Phi_2 \cup \{x = y\}}$$

$$\textbf{if} \begin{cases} \Phi_1 \textit{ is } T_1\text{-satisfiable,} \\ \Phi_1 \wedge x \neq y \text{ is } T_1\text{-unsatisfiable,} \\ \Phi_2 \wedge x \neq y \text{ is } T_2\text{-satisfiable, and} \\ x, y \in Var(\Phi_1) \cap Var(\Phi_2) \end{cases}$$

Figure 1: The Inference System NO$_1$

3.1 Combining CSI-theories

We assume that T_1 and T_2 are in **CSI**, which requires satisfiability procedures for both T_1 and T_2. Let us consider the inference system NO obtained as the union of NO$_1$ presented in Figure 1 and NO$_2$ obtained from NO$_1$ by symmetry.[4] NO takes configurations of the form $\Phi_1 ; \Phi_2$ where Φ_i is a set of Σ_i-literals ($i = 1, 2$). Rule Contradiction$_1$ reports the T_1-unsatisfiability of Φ_1 (and hence of $\Phi_1 \wedge \Phi_2$), detected by the available satisfiability procedure. Rule Deduction$_1$ propagates equalities between shared variables detected in T_1 to T_2 (if they are not already known). The problem of checking whether the equality $x = y$ is a logical consequence of $T_1 \cup \Phi_1$ is transformed into the problem of checking the T_1-unsatisfiability of $\Phi_1 \cup \{x \neq y\}$ in order to exploit the available satisfiability procedure.

Theorem 1 *Let T_1, T_2 be two signature-disjoint* **CSI***-theories. Let* NO *be the inference system defined as the union* NO$_1 \cup$ NO$_2$*, where* NO$_1$ *is depicted in Figure 1 and* NO$_2$ *is obtained from* NO$_1$ *by symmetry. The relation $\vdash^*_{\mathsf{NO}}$ is terminating and $\Phi_1 ; \Phi_2 \vdash^*_{\mathsf{NO}} false$ iff $\Phi_1 \wedge \Phi_2$ is $T_1 \cup T_2$-unsatisfiable.*

Proof: Direct consequence of the three following Lemmas.

Lemma 3 (Termination) *The relation $\vdash^*_{\mathsf{NO}}$ is terminating.*

Proof: If the rule Contradiction applies, the procedure terminates. The rule Deduction strictly decreases the number of equivalence classes of shared variables.

Lemma 4 (Soundness) *The relation $\vdash_{\mathsf{NO}}$ preserves equisatisfiability in $T_1 \cup T_2$.*

[4] A symmetric rule for T_2 is obtained from a rule for T_1 by swapping indexes 1 and 2. A symmetric inference system for T_2 is the set of symmetric rules for T_2 obtained from the rules for T_1.

Proof: The soundness of the rule Contradiction is straightforward. Let us consider the rule Deduction. If $\models^{\alpha}_{\mathcal{M}} (\Phi_1 \cup \Phi_2)$, the application conditions guarantee that $T_1 \cup \Phi_1 \models x = y$, implying $\alpha(x) = \alpha(y)$. Thus $\models^{\alpha}_{\mathcal{M}} (\Phi_1 \cup \Phi_2 \cup \{x = y\})$. The converse is trivial

Lemma 5 (Completeness) *If $\Phi_1; \Phi_2$ is a normal form w.r.t. $\vdash_{\mathsf{NO}}$ different from $false$, then $\Phi_1 \wedge \Phi_2$ is $T_1 \cup T_2$-satisfiable.*

Proof: If the procedure terminates without reporting $false$, the final configuration must be of the form $\Phi_1; \Phi_2$ such that:

- Φ_i is T_i-satisfiable for $i = 1, 2$ (otherwise Contradiction applies),
- $\forall x, y \in Var(\Phi_1) \cap Var(\Phi_2)$, $T_1 \models \Phi_1 \Rightarrow x = y$ iff $T_2 \models \Phi_2 \Rightarrow x = y$ (otherwise Deduction applies).

Assume $\Phi_1 \wedge \Phi_2$ is $T_1 \cup T_2$-unsatisfiable. Consider the equivalence relation E over variables in $Var(\Phi_1) \cap Var(\Phi_2)$ such that $(x, y) \in E$ iff $T_1 \models \Phi_1 \Rightarrow x = y$ and $T_2 \models \Phi_2 \Rightarrow x = y$.

By Lemma 2, if $\Phi_1 \wedge \Phi_2$ is $T_1 \cup T_2$-unsatisfiable, there exists $i \in \{1, 2\}$ such that $\Phi_i \cup arr(V, E)$ is T_i-unsatisfiable. Then two cases must be distinguished:

- $arr(V, E)$ contains at least a disequality. By convexity hypothesis, there exist some i and some disequality $x_k \neq y_k$ such that $\Phi_i \cup E \cup \{x_k \neq y_k\}$ is T_i-unsatisfiable, or equivalently $T_i \cup \Phi_i \cup E \models x_k = y_k$. Therefore (x_k, y_k) is in E, which is a contradiction.
- $arr(V, E)$ contains no disequalities. Then $\Phi_i \cup arr(V, E)$ is $\Phi_i \cup E$ which is T_i-equivalent to Φ_i. Thus Φ_i is T_i-unsatisfiable, which is again a contradiction.

NO specifies only the essence of the Nelson-Oppen schema. Such a schema can be refined to increase efficiency. In the following, we will consider refinements of NO based on the separation of the input set of literals into disequalities and equalities and other literals. The convexity assumption allows us to have a theory-independent handling of disequalities. In addition, considering disequalities is useless to obtain the entailed elementary equalities needed in NO. This property paves the way of incorporating solvers and canonizers for theories in **SH**.

Lemma 6 *Let T be a convex theory, Φ a T-satisfiable set of literals, and Δ a set of elementary disequalities. Then $\Phi \wedge \Delta$ is T-unsatisfiable iff there exists $x \neq y \in \Delta$ such that $T \models \Phi \Rightarrow x = y$.*

Proof: ($\Leftarrow$) Trivial. ($\Rightarrow$) By expressing the convexity definition in terms of unsatisfiability.

Lemma 7 *Let T be a convex theory, Φ a set of T-literals, and Δ a set of T-disequalities. If $\Phi \wedge \Delta$ is T-satisfiable, then $T \models (\Phi \wedge \Delta) \Rightarrow x = y$ iff $T \models \Phi \Rightarrow x = y$.*

Proof: ($\Leftarrow$) Trivial. ($\Rightarrow$) Assume that $T \cup \Phi \cup \Delta \models x = y$. Then $T \cup \Phi \cup (\Delta \cup \{x \neq y\})$ is unsatisfiable. By convexity of T, either $T \cup \Phi \cup \{x \neq y\}$ is unsatisfiable or there exists a disequality $s \neq t \in \Delta$ such that $T \cup \Phi \cup \{s \neq t\}$ is unsatisfiable. In the first case, the Lemma is proved. In the second one, we obtain that $\Phi \wedge \Delta$ is T-unsatisfiable, and so we get a contradiction.

$\mathsf{Solve} - \mathsf{fail}_1$

$$\frac{\Gamma_1, \Delta_1; \Gamma_2, \Delta_2}{false} \ \textbf{if } solve_1(\Gamma_1) = false$$

$\mathsf{Solve} - \mathsf{success}_1$

$$\frac{\Gamma_1, \Delta_1; \Gamma_2, \Delta_2}{\widehat{\sigma_1}, \Delta_1; \Gamma_2, \Delta_2} \ \textbf{if} \left\{ \begin{array}{l} \Gamma_1 \text{ is not in solved form,} \\ \sigma_1 = solve_1(\Gamma_1) \neq false \end{array} \right.$$

$\mathsf{Contradiction}_1$

$$\frac{\widehat{\sigma_1}, \Delta_1 \cup \{s \neq t\}; \Gamma_2, \Delta_2}{false} \ \textbf{if } canon_1(s\sigma_1) = canon_1(t\sigma_1)$$

$\mathsf{Deduction}_1$

$$\frac{\widehat{\sigma_1}, \Delta_1; \widehat{\sigma_2}, \Delta_2}{\widehat{\sigma_1}, \Delta_1; \widehat{\sigma_2} \cup \{x = y\}, \Delta_2}$$

$$\textbf{if} \left\{ \begin{array}{l} canon_1(x\sigma_1) = canon_1(y\sigma_1), \\ canon_2(x\sigma_2) \neq canon_2(y\sigma_2), \\ x, y \in Var(\sigma_1) \cap Var(\sigma_2) \end{array} \right.$$

Figure 2: The Inference System SH_1

3.2 Combining SH-theories

Let us assume that T_1 and T_2 are in **SH**, which requires a canonizer $canon_i$ and a solver $solve_i$ for each theory T_i $(i = 1, 2)$. Preliminary to the combination schema, we extend solvers to handle sets of equalities as follows: $solve(\emptyset)$ returns the identity substitution ϵ; $solve(\Gamma \cup \{s = t\}) = false$, if $solve(s = t) = false$; and $solve(\Gamma \cup \{s = t\}) = \sigma \circ solve(\Gamma\sigma)$, if $solve(s = t) = \sigma$, where $\circ$ denotes composition of substitutions.

Let us consider the inference system SH obtained as the union of SH_1 presented in Figure 2 and SH_2 obtained from SH_1 by symmetry. SH takes configurations of the form $\Gamma_1, \Delta_1; \Gamma_2, \Delta_2$, where Γ_i is a set of Σ_i-equalities and Δ_i is a set of Σ_i-disequalities for $i = 1, 2$. Rule $\mathsf{Solve} - \mathsf{fail}_1$ reports the T_1-unsatisfiability of Γ_1 (and hence of $\Gamma_1 \wedge \Delta_1 \wedge \Gamma_2 \wedge \Delta_2$) detected by $solve_1$. Rule $\mathsf{Solve} - \mathsf{success}_1$ replaces the Σ_1-equalities Γ_1 with their solved form which is obtained again by using $solve_1$. This is important for the next two rules. Dealing with solved forms allows us to simply determine entailed equalities (possibly between shared variables, see $\mathsf{Deduction}_1$) using canonizers. Hence, it is possible to lazily report unsatisfiability as soon as we find a disequality whose corresponding equality is entailed (see $\mathsf{Contradiction}_1$). Indeed, convexity allows us to handle disequalities one by one.

Theorem 2 *Let T_1, T_2 be two signature-disjoint* **SH***-theories. Let* SH *be the inference system defined as the union* $\mathsf{SH}_1 \cup \mathsf{SH}_2$*, where* SH_1 *is depicted in Figure 2 and* SH_2 *is obtained by symmetry. The relation* $\vdash^*_{\mathsf{SH}}$ *is terminating and* $\Gamma_1, \Delta_1; \Gamma_2, \Delta_2 \vdash^*_{\mathsf{SH}} false$ *iff* $\Gamma_1 \wedge \Delta_1 \wedge \Gamma_2 \wedge \Delta_2$ *is $T_1 \cup T_2$-unsatisfiable.*

Proof: Direct consequence of the following Lemmas.

Lemma 8 *Let $\sigma = \{x_1 \to t_1, ..., x_n \to t_n\}$ be a substitution such that $x_i \notin Var(t_j)$ for all i, j. For every theory T and every conjunction Γ of literals we have:*

$$T \cup \Gamma \cup \{x_1 = t_1, ..., x_n = t_n\} \models x = y \textit{ iff } T \cup \Gamma\sigma \models x\sigma = y\sigma$$

Proof: ($\Rightarrow$) Assume that $T \cup \Gamma \cup \{x_1 = t_1, ..., x_n = t_n\} \models x = y$ and let $\mathcal{M}, \alpha$ be such that $\models^{\alpha}_{\mathcal{M}} T \cup \Gamma\sigma$. By the fact that x_i does not appear in $T \cup \Gamma\sigma$, one can redefine α such that $\alpha(x_i) = \alpha(t_i)$, for all i. It is easy to see that α satisfies $T \cup \Gamma \cup \{x_1 = t_1, ..., x_n = t_n\}$ in $\mathcal{M}$. It implies α satisfies $x = y$ in $\mathcal{M}$ too, thus $x\sigma = y\sigma$ evaluates to true in $\mathcal{M}$ by using α.

($\Leftarrow$) Assume that $T \cup \Gamma\sigma \models x\sigma = y\sigma$ and let $\mathcal{M}, \alpha$ be such that $\models^{\alpha}_{\mathcal{M}} T \cup \Gamma \cup \{x_1 = t_1, ..., x_n = t_n\}$. It is straightforward that α satisfies $T \cup \Gamma\sigma$ in $\mathcal{M}$. Hence α satisfies $x\sigma = y\sigma$ in $\mathcal{M}$. In addition, $\alpha(x_i) = \alpha(t_i)$, for all i, which implies $x = y$ evaluates to true in $\mathcal{M}$ by using α.

Lemma 9 (Termination) *The relation $\vdash^*_{\mathsf{SH}}$ is terminating.*

Proof: If rules Solve-fail and Contradiction apply, the procedure terminates. For other rules, we must show that rules Solve-success and Deduction can not be applied infinitely. Indeed, Solve-success can only be applied if Γ_1 is not in solved form. Only Deduction may likely modify a solved form into a non-solved form by integrating an equality between variables. The set of shared variables is finite and Deduction integrates equalities detected in T_1 to T_2 only if they have not been detected in T_2 ($canon_1(x\sigma_1) = canon_1(y\sigma_1)$ and $canon_2(x\sigma_2) \neq canon_2(y\sigma_2)$). This guarantees that Deduction can only be applied finitely many times.

Lemma 10 (Soundness) *The relation $\vdash_{\mathsf{SH}}$ preserves equisatisfiability in $T_1 \cup T_2$.*

Proof: The soundness is straightforward for rules Solve-fail and Contradiction.

- For Solve-success:
  - ($\Rightarrow$) Assume that $\models^{\alpha}_{\mathcal{M}} \Gamma_1 \wedge \Delta_1 \wedge \Gamma_2 \wedge \Delta_2$. Let $\sigma_1 = \{x_1 \to t_1, ..., x_n \to t_n\} = solve(\Gamma_1)$. By definition of $solve$, the equivalence ($\Gamma_1 \Leftrightarrow \exists\tilde{y}. \bigwedge_{i=1}^{n} x_i = t_i$) is $T_1 \cup T_2$-valid. We can extend α to $\tilde{y}$ such that $\alpha(x_i) = \alpha(t_i)$. It is easy to see that $\models^{\alpha}_{\mathcal{M}} \widehat{\sigma_1} \wedge \Delta_1 \wedge \Gamma_2 \wedge \Delta_2$.
  - ($\Leftarrow$) Assume that $\models^{\alpha}_{\mathcal{M}} \widehat{\sigma_1} \wedge \Delta_1 \wedge \Gamma_2 \wedge \Delta_2$, where $\sigma_1 = solve(\Gamma_1)$. Due to the above equivalence, we clearly have $\models^{\alpha}_{\mathcal{M}} \Gamma_1 \wedge \Delta_1 \wedge \Gamma_2 \wedge \Delta_2$.
- For Deduction:
  - ($\Rightarrow$) Assume that $\models^{\alpha}_{\mathcal{M}} \widehat{\sigma_1} \wedge \Delta_1 \wedge \widehat{\sigma_2} \wedge \Delta_2$. We have $\alpha(x) = \alpha(x\sigma_1)$ and $\alpha(y) = \alpha(y\sigma_1)$. By Lemma 8, $T_1 \cup \widehat{\sigma_1} \models x = y$ iff $T_1 \models x\sigma_1 = y\sigma_1$. But then $T_1 \models x\sigma_1 = y\sigma_1$ implies $T_1 \cup T_2 \models x\sigma_1 = y\sigma_1$, thus $\alpha(x\sigma_1) = \alpha(y\sigma_1)$. Hence $\alpha(x) = \alpha(y)$ and $\models^{\alpha}_{\mathcal{M}} \widehat{\sigma_1} \wedge \Delta_1 \wedge \widehat{\sigma_2} \wedge x = y \wedge \Delta_2$.
  - ($\Leftarrow$) Trivial.

Lemma 11 (Completeness) *If* $\Gamma_1, \Delta_1; \Gamma_2, \Delta_2$ *is a normal form w.r.t.* $\vdash_{\mathsf{SH}}$ *different from* $false$*, then* $\Gamma_1 \wedge \Delta_1 \wedge \Gamma_2 \wedge \Delta_2$ *is* $T_1 \cup T_2$*-satisfiable.*

Proof: If the procedure terminates without reporting $false$, the final configuration must be of the form $(\widehat{\sigma_1}, \Delta_1); (\widehat{\sigma_2}, \Delta_2)$ such that:

- $\widehat{\sigma_i} \wedge \Delta_i$ is T_i-satisfiable for $i = 1, 2$ (otherwise Contradiction applies),
- $\forall x, y \in Var(\Phi_1) \cap Var(\Phi_2)$, $canon_1(x\sigma_1) = canon_1(y\sigma_1)$ iff $canon_2(x\sigma_2) = canon_2(y\sigma_2)$ (otherwise Deduction applies).

Since $\widehat{\sigma_i} \wedge \Delta_i$ is T_i-satisfiable we have $T_i \models (\widehat{\sigma_i} \wedge \Delta_i) \Rightarrow x = y$ iff $canon_i(x\sigma_i) = canon_i(y\sigma_i)$ thanks to Lemma 7, Lemma 8, and the definition of *canon*. Then the proof can be continued by contradiction in a way similar to the proof of Lemma 5.

It is easy to see that a strategy applying rules $\mathsf{Solve} - \mathsf{fail}_1$, $\mathsf{Solve} - \mathsf{success}_1$, and $\mathsf{Contradiction}_1$ in SH to a configuration $\Gamma_1, \Delta_1; \Gamma_2, \Delta_2$ yields the same result as that of applying rule $\mathsf{Contradiction}_1$ in NO to $\Gamma_1 \cup \Delta_1; \Gamma_2 \cup \Delta_2$. Similarly, the application of rules $\mathsf{Solve} - \mathsf{success}_1$ and $\mathsf{Deduction}_1$ in SH simulates the application of $\mathsf{Deduction}_1$ in NO; showing that equalities between shared variables can be derived by invoking a solver (and a canonizer) rather than resorting to guessing as for NO when applying the rule $\mathsf{Deduction_i}$ $(i = 1, 2)$. This is one of the key insights underlying Shostak's schema.

3.3 Combining a CSI-theory with a SH-theory

Without loss of generality, let us assume that T_1 is in **CSI** and that T_2 is in **SH**. This situation frequently arises in practical verification problem, e.g. the union of a theory in **SH** and $\mathcal{E}$ (which is *not* in **SH**). We consider the inference system NS obtained as the union of NO_1 in Figure 1 and SH_2, the symmetric of SH_1 in Figure 2. NS takes configurations of the form $\Phi_1; \Gamma_2, \Delta_2$ where Φ_1 is a set of Σ_1-literals, Γ_2 is a set of Σ_2-equalities, and Δ_2 is a set of Σ_2-disequalities. We furtherly assume that when a rule of NO is applied, $\Phi_1; \Gamma_2, \Delta_2$ stands for $\Phi_1; \Gamma_2 \cup \Delta_2$ and when a rule of SH is applied, $\Phi_1; \Gamma_2, \Delta_2$ is considered as $\Gamma_1, \Delta_1; \Gamma_2 \cup \Delta_2$, where $\Phi_1 = \Gamma_1 \cup \Delta_1$ and Γ_1 (Δ_1) is a set of Σ_1-equalities (-disequalities, respectively). NS can be seen as an abstract version of the one proposed in [4].

Theorem 3 *Let* T_1, T_2 *be two signature-disjoint theories such that* T_1 *is in* **CSI** *and* T_2 *is in* **SH**. *Let* NS *be the inference system defined as the union* $\mathsf{NO}_1 \cup \mathsf{SH}_2$*, where* NO_1 *is in Figure 1 and* SH_2 *is obtained from* SH_1 *in Figure 2 by symmetry. The relation* $\vdash^*_{\mathsf{NS}}$ *is terminating and* $\Phi_1; \Gamma_2, \Delta_2 \vdash^*_{\mathsf{NS}} false$ *iff* $\Phi_1 \wedge \Gamma_2 \wedge \Delta_2$ *is* $T_1 \cup T_2$*-unsatisfiable.*

Let $T_1, ..., T_k$ and $T_{k+1}, ..., T_{k+n}$ be k theories in **CSI** and n theories in **SH**, respectively, and such that $\Sigma_i \cap \Sigma_j \neq \emptyset$ for $i, j = 1, \ldots, k+n$, $i \neq j$, and $n, k \geq 1$. It is possible to modularly build a satisfiability procedure for $T = \bigcup_{j=1}^{k+n} T_j$ as follows. Repeatedly use NO to obtain a satisfiability procedure for $U_0 = \bigcup_{j=1}^{k} T_j$, then repeatedly use NS to build satisfiability procedures for $U_1 = U_0 \cup T_{k+1}, \ldots, U_n = U_{n-1} \cup T_{k+n}$, where U_n is T. An alternative would be to repeatedly use SH to construct satisfiability procedures for unions of two theories in **SH**, followed by a repeated use of NO on the resulting theories.

Also, let us mention still another possibility to combine k theories in **CSI** and n theories in **SH**. It is possible to slightly modify our inference rules to take into account $k+n$ theories and configurations of the form

$$\Phi_1; \ldots; \Phi_k; \Gamma_{k+1}, \Delta_{k+1}; \ldots; \Gamma_{k+n}, \Delta_{k+n}.$$

The rule $\mathsf{Deduction}$ would propagate an equality between shared variables, deduced in one theory, to the other $(k+n)-1$ theories. At this point, it is not difficult to modify the proof of correctness for NS to show that the resulting rules (taken from $\mathsf{NO}_1, \ldots, \mathsf{NO}_k, \mathsf{SH}_{k+1}, \ldots, \mathsf{SH}_{k+n}$) yield a satisfiability procedure for T. The resulting proof would be a bit more involved because of the more complex notation.

3.4 Combining Non Stably Infinite Convex Theories

So far, we have only considered stably infinite convex theories. However we can sometimes drop the requirement of stable infiniteness and replace it with a somewhat more natural requirement, i.e. the decidability for the (convex) theory to admit a trivial model.

The following simple fact will be useful for the completeness of the satisfiability problem in combination of (convex) non stably infinite theories.

Proposition 2 *Let T be a $\mathbf{C}$-theory. If T has a non trivial model, then T has an infinite model.*

Proof: If T does not have infinite models, then T must entail the formula $\bigvee_{1 \leq j \neq k \leq n} x_j = x_k$ for some positive integer n. Then, T entails $x_k = x_{k'}$ since T is convex. Therefore, T has only trivial models, which leads to a contradiction.

It turns out that by studying the existence of trivial models for (convex) theories, we are able to decide the satisfiability in the combination of non stably infinite theories.

Proposition 3 *Let T_1 and T_2 be decidable $\mathbf{C}$-theories such that we know whether T_i has a trivial model, for $i = 1, 2$. If the signatures of T_1 and T_2 are disjoint, then $T_1 \cup T_2$ is a decidable $\mathbf{C}$-theory.*

Proof: Let Σ_1 (resp. Σ_2) be the signature of T_1 (resp. T_2). Let Φ be a set of quantifier free $\Sigma_1 \cup \Sigma_2$-literals. By variable abstraction, Φ is $T_1 \cup T_2$-equisatisfiable to $\Phi_1 \cup \Phi_2$ such that Φ_i contains only Σ_i-literals ($i = 1, 2$). Thanks to Lemma 1, $\Phi_1 \cup \Phi_2$ is $T_1 \cup T_2$-satisfiable iff there exists an arrangement Φ_0 such that $\Phi_1 \cup \Phi_0$ is satisfiable in a model $\mathcal{M}_1$ of T_1 and $\Phi_2 \cup \Phi_0$ is satisfiable in a model $\mathcal{M}_2$ of T_2 and the domains of $\mathcal{M}_1$ and $\mathcal{M}_2$ have the same cardinality. We show that such models exist if we know whether T_i (for $i = 1, 2$) has a trivial model. We have to consider the following cases:

- $\Phi_1 \cup \Phi_0$ is T_1-satisfiable in a model $\mathcal{M}_1$ and $\Phi_2 \cup \Phi_0$ is T_2-satisfiable in a model $\mathcal{M}_2$ such that $\mathcal{M}_1$ and $\mathcal{M}_2$ are both non trivial. This can be checked by deciding for $i = 1, 2$, if $\Phi_i \cup \Phi_0 \cup \{x \neq y\}$ is T_i-satisfiable, where x, y are new variables. By Proposition 2, we know that there exists a model $\mathcal{M}'_1$ of T_1 satisfying $\Phi_1 \cup \Phi_0$ and a model $\mathcal{M}'_2$ of T_2 satisfying $\Phi_2 \cup \Phi_0$ such that $\mathcal{M}'_1$ and $\mathcal{M}'_2$ have the same infinite cardinality. As a consequence, $\Phi_1 \cup \Phi_2$ is $T_1 \cup T_2$-satisfiable.

- $\Phi_i \cup \Phi_0$ is T_i-satisfiable but only in a trivial model, for $i = 1, 2$. In this case, it is sufficient to check that for $i = 1, 2$, if $\Phi_i \cup \Phi_0 \cup \{x \neq y\}$ is T_i-unsatisfiable, where x, y are new variables. Thus $\Phi_1 \cup \Phi_2$ is $T_1 \cup T_2$-satisfiable in a trivial model.

- $\Phi_1 \cup \Phi_0$ is T_1-satisfiable only in a trivial model and $\Phi_2 \cup \Phi_0$ is T_2-satisfiable in a non trivial model. This can be checked by testing if $\Phi_1 \cup \Phi_0 \cup \{x \neq y\}$ is T_1-unsatisfiable, and $\Phi_2 \cup \Phi_0 \cup \{x \neq y\}$ is T_2-satisfiable, where x, y are new variables. Now, we only need to check if $\Phi_2 \cup \Phi_0$ is satisfiable in a trivial model of T_2. To this end, it is sufficient to verify whether $\Phi_2 \cup \Phi_0$ contains a disequality. Indeed, $\Phi_2 \cup \Phi_0$ is satisfiable in a trivial model of T_2 iff $\Phi_2 \cup \Phi_0$ does not contain any disequalities and T_2 has a trivial model.

4 Combining Deduction Complete Theories

In order to find a suitable trade-off between modularity (of the Nelson-Oppen approach) and efficiency (by using canonizers and solvers), we introduce in this Section the concepts of deduction completeness and inference-based satisfiability procedure. Such procedures can be flexibly built by adapting the rewriting approach of [1] (cf. Section 4.1) and modularly combined by reusing the ideas underlying both Nelson-Oppen and Shostak approaches (cf. Section 4.2).

Informally, an *inference-based* procedure is a satisfiability procedure defined via an inference system whose inference rules have fixed arity and perform validity preserving transformations on sets of clauses. We also assume that the (fair) application of the inference rules is terminating for any input, and that the final set of clauses contains $false$ iff the input is T-unsatisfiable. When it is T-satisfiable, the final set of clauses does not contain $false$ but the so-called *deduction complete* set of elementary equalities representing all entailed elementary equalities.

Definition 1 *Let T be a convex theory and Φ be a T-satisfiable set of literals. A set of elementary equalities E is* deduction complete (for Φ modulo T) *if*

$$\forall x, y \in Var(\Phi),\ T \models \Phi \Rightarrow x = y \text{ iff } E \models x = y$$

A deduction complete T-satisfiability procedure *is a T-satisfiability procedure, denoted by DC_T, such that if Φ is T-satisfiable, then it returns $false$, otherwise it returns true$\{E\}$ where E is deduction complete for Φ modulo T.*

Before being able to define the notion of inference system which is at the core of a deduction complete satisfiability procedure, we need to introduce some technical notions and notations. The set of elementary equalities occurring in a set S of clauses is denoted by $E(S)$ and $\overline{E(S)}$ is $S \backslash E(S)$. If E is a finite set of elementary equalities over totally ordered constants, then $E_\downarrow$ denotes the *canonical form* of E defined as the finite set of elementary equalities $c = c\!\downarrow$ such that $c\!\downarrow$ is the smallest constant such that $c =_E c\!\downarrow$ and $c\!\downarrow \neq c$. $S\!\downarrow_E$ denotes the normal form of S with respect to the (canonical) rewrite system defined by orienting from left to right the equalities in $E_\downarrow$. E is said in *canonical form* if $E = E_\downarrow$.

Definition 2 *A* basic inference system $\mathcal{I}$ for a theory T *is an inference system working on sets of clauses such that each inference rule in $\mathcal{I}$ has a given arity k and transforms k clauses into a T-equivalent set of clauses. An $\mathcal{I}$-transition is defined as follows: $S \rightarrow_{\mathcal{I}} S'$ if S' is obtained by applying a rule in $\mathcal{I}$ on $\overline{E(S)}$. A ξ-transition is defined as follows: $S \rightarrow_{\xi} S'$ if $S' = E(S)_{\downarrow} \cup \overline{E(S)}\downarrow_{E(S)}$. An $\mathcal{I}$-derivation is a chain of $\mathcal{I}$-transitions and ξ-transitions, denoted by $\overset{*}{\rightarrow}_{\xi\cup\mathcal{I}}$, where ξ-transitions are applied eagerly. An $\mathcal{I}$-normal form of a set of clauses S is a set of clauses S' such that $S \overset{*}{\rightarrow}_{\xi\cup\mathcal{I}} S'$ and there is no S'' satisfying $S' \rightarrow_{\mathcal{I}} S''$ or $S' \rightarrow_{\xi} S''$.*

We are now ready to identify some interesting properties of inference-based satisfiability procedures.

Definition 3 *Let T be a convex theory and $\mathcal{I}$ a basic inference system for T. Given a finite set of T-valid clauses Ax:*

- $(\mathcal{I}, Ax)$ *is* refutation complete *if for any T-unsatisfiable set of ground equalities Γ, any $\mathcal{I}$-derivation starting from $Ax \cup \Gamma$ is finite and $false$ occurs in any $\mathcal{I}$-normal form of $Ax \cup \Gamma$.*
- $(\mathcal{I}, Ax)$ *is* terminating *if for any set of ground equalities Γ, any $\mathcal{I}$-derivation starting from $Ax \cup \Gamma$ is finite.*
- $(\mathcal{I}, Ax)$ *is* deduction complete *if $(\mathcal{I}, Ax)$ is refutation complete and terminating, and for any T-satisfiable set of ground equalities Γ and any $\mathcal{I}$-normal form S of $Ax \cup \Gamma$, $E(S)$ is deduction complete for Γ modulo T.*

The set of clauses Ax is omitted whenever it is clear from the context and the inference system $\mathcal{I}$ is said refutation complete (resp. terminating, deduction complete). A theory T is deduction complete *if there exist a basic inference system $\mathcal{I}$ and a set of clauses Ax such that $(\mathcal{I}, Ax)$ is deduction complete. The class of deduction complete (resp. deduction complete and stably infinite) convex theories is denoted by* **DCC** *(resp.* **DCCSI***).*

Some remarks are in order. First, if $(\mathcal{I}, Ax)$ is *deduction complete* for T, then $\mathcal{I}$ provides a deduction complete satisfiability procedure. Second, by definition **DCC** $\subseteq$ **C** and **DCCSI** $\subseteq$ **CSI**. Third, the assumptions on basic inference systems prevent us to use guessing and a T-satisfiability procedure to derive entailed elementary equalities, since the rules in a deduction complete inference system may only perform "local" changes. Finally, it is possible to build a deduction complete satisfiability procedure by using the solver and the canonizer of a **SH**-theory.

Proposition 4 **SH** $\subseteq$ **DCCSI***, i.e. each theory in* **SH** *admits a deduction complete decision procedure.*

It is straightforward to prove this proposition by using the basic inference system $\mathcal{SH}$ depicted in Figure 3 and noticing that $(\mathcal{SH}, \emptyset)$ is deduction complete for any **SH**-theory.

4.1 Deduction completeness by superposition

We show in what follows how a Superposition Calculus ($\mathcal{SP}$) can be used to build deduction complete decision procedures. This can be seen as a generalization of

Solve − fail

$$\frac{s = t}{false} \quad \textbf{if} \begin{cases} s = t \text{ is not in solved form,} \\ solve(s = t) = false \end{cases}$$

Solve − success

$$\frac{s = t}{\widehat{\sigma}} \quad \textbf{if} \begin{cases} s = t \text{ is not in solved form,} \\ \sigma = solve(s = t) \neq false \end{cases}$$

Replace

$$\frac{u = v \wedge x = t}{(u = v)\{x \mapsto t\} \wedge x = t} \quad \textbf{if} \begin{cases} x = t \text{ is in solved form,} \\ x \text{ occurs in } u = v \end{cases}$$

Canon

$$\frac{x = t}{x = canon(t)} \quad \textbf{if} \begin{cases} x \text{ does not occur in } t \\ t \neq canon(t) \end{cases}$$

Deduction

$$\frac{x = t \wedge y = t}{x = t \wedge y = t \wedge x = y} \quad \textbf{if } x, y \text{ do not occur in } t$$

Figure 3: The basic inference system $\mathcal{SH}$

the rewriting approach to flexibly build satisfiability procedures proposed in [1]. We explore this problem for the subclass of Horn theories axiomatized by a finite set of Horn clauses.[5]

The Superposition Calculus $\mathcal{SP}$. A fundamental feature of $\mathcal{SP}$ is the usage of a *reduction ordering* $\succ$ which is total on ground terms, for example the lexicographic path ordering ([14]). We also assume that if a term t is not a variable or constant, then for any constant c we have that $t \succ c$. The ordering $\succ$ is extended to positive literals by considering them as multisets of terms, and then to the clauses by considering them as multisets of positive literals. The inference system $\mathcal{SP}$ uses a selection function *sel* such that for each clause C, $sel(C)$ contains a negative literal in C if C contains one, otherwise all maximal literals in C w.r.t. $\succ$. $\mathcal{SP}$ contains two kind of rules: Expansion Rules and Contraction Rules. Expansion Rules are necessary for refutation completeness of $\mathcal{SP}$. Contraction Rules are crucial for efficiency, and they are required to be performed eagerly whenever possible.

A clause C is *redundant* with respect to a set S of clauses if either $C \in S$ or S can be obtained from $S \cup \{C\}$ by a sequence of application of the contraction rules of Figure 5. An inference is *redundant* with respect to a set S of clauses if its conclusion is redundant with respect to S. A set S of clauses is *saturated* with respect to $\mathcal{SP}$ if every inference of $\mathcal{SP}$ with a premise in S is redundant with respect to S. A derivation in this inference system is a sequence

[5] Horn theories are known to be convex.

Right paramodulation	$\dfrac{\Gamma \Rightarrow \Delta, l[u'] = r \quad \Pi \Rightarrow \Sigma, u = t}{\sigma(\Gamma, \Pi \Rightarrow \Delta, \Sigma, l[t] = r)}$	$(i - iv)$
Left paramodulation	$\dfrac{\Gamma, l[u'] = r \Rightarrow \Delta \quad \Pi \Rightarrow \Sigma, u = t}{\sigma(l[t] = r, \Gamma, \Pi \Rightarrow \Delta, \Sigma)}$	$(i - iv)$
Reflection	$\dfrac{\Gamma, u' = u \Rightarrow \Delta}{\sigma(\Gamma \Rightarrow \Delta)}$	(v)
Eq. Factoring	$\dfrac{\Gamma \Rightarrow \Delta, u = t, u' = t'}{\sigma(\Gamma, t = t' \Rightarrow \Delta, u = t')}$	$(i), (vi)$

where a clause $\neg A_1 \vee \cdots \vee \neg A_n \vee B_1 \vee \cdots \vee B_n$ is written in sequent style as $\{A_1, \ldots, A_n\} \Rightarrow \{B_1, \ldots, B_m\}$ (where the A_i's and B_j's are equalities), equality is the only predicate symbol, σ is the most general unifier of u and u', u' is not a variable in *Left paramodulation* and *Right paramodulation*, L is a literal, and:

(i) $\sigma(u) \not\preceq \sigma(t)$, ***(ii)*** $u = t$ is selected in its clause, ***(iii)*** $\sigma(l[u']) \not\preceq \sigma(r)$, ***(iv)*** $l = r$ is selected in its clause, ***(v)*** $u' = u$ is selected in its clause, ***(vi)*** $u = t$ is selected in its clause and $\sigma(t) \not\preceq \sigma(t')$ and $\sigma(u') \not\preceq \sigma(t')$.

Figure 4: Expansion Rules of $\mathcal{SP}$.

Orientation	$\dfrac{S \cup \{c = c'\}}{S[c \rightarrow c']}$	if $c \succ c'$
Simplification	$\dfrac{S \cup \{C[l'], l = r\}}{S \cup \{C[\theta(r)], l = r\}}$	if $l' \equiv \theta(l)$, $\theta(l) \succ \theta(r)$, and $l = r$ is selected in its clause
Subsumption	$\dfrac{S \cup \{C, C'\}}{S \cup \{C\}}$	if for some subs. θ, $\theta(C) \subseteq C'$
Deletion	$\dfrac{S \cup \{\Gamma \Rightarrow \Delta, t = t\}}{S}$	

where C and C' are clauses and S is a set of clauses.

Figure 5: Contraction Rules of $\mathcal{SP}$.

$S_0, S_1, \ldots, S_i, \ldots$ of sets of clauses where at each step an inference of $\mathcal{SP}$ is applied to generate and add a clause (cf. expansion rules in Figure 4) or to delete or reduce a clause (cf. contraction rules in Figure 5). A derivation is characterized by its *limit*, defined as the set of persistent clauses $S_\infty = \bigcup_{j \geq 0} \bigcap_{i > j} S_i$. A derivation $S_0, S_1, ..., S_i, ...$ with limit S_∞ is *fair* with respect to $\mathcal{SP}$ if for every inference in $\mathcal{SP}$ with premises in S_∞, there is some $j \geq 0$ such that the inference is redundant in S_j.

Theorem 4 ([30]) *If $S_0, S_1, \ldots$ is a fair derivation of $\mathcal{SP}$, then (*i*) its limit S_∞ is saturated with respect to $\mathcal{SP}$, (*ii*) S_0 is unsatisfiable iff the empty clause is in S_j for some j, and (*iii*) if such a fair derivation is finite, i.e. it is of the form $S_0, \ldots, S_n$, then S_n is saturated and logically equivalent to S_0.*

The saturation-based methodology [1] for T-satisfiability consists of two phases:

1. *Flattening:* all ground literals are flattened by introducing new constants, yielding an equisatisfiable *flat* problem.

2. *Ordering selection and termination:* any fair derivation of $\mathcal{SP}$ is shown to be finite when applied to a flat problem together with the axioms of T, provided that $\succ$ satisfies a few properties depending on T.

Variable inactivity and combination. In [2], the notion of variable inactive theory has been identified as the key (sufficient) condition to obtain the modularity of termination of the fair and exhaustive application of the rules of $\mathcal{SP}$ on the union of (presentations of) theories. A theory T axiomatized by $Ax(T)$ is *variable inactive* when any fair derivation of $\mathcal{SP}$ on $Ax(T) \cup S$, for some set of clauses S, does not contain a maximal literal (w.r.t. the ordering $\succ$) of the form $X = t$ and $X \notin Var(t)$. The absence of such (maximal) literals ensures that no inference on variables may involve clauses derived from different axiomatizations of theories and so termination is preserved. It turns out that the variable inactivity property of a theory is a key requirement also for deduction completeness, as stated in the following theorem.

Theorem 5 *Let T be a theory axiomatized by a finite set $Ax(T)$ of Horn clauses. Assume that for every set S of ground flat literals, any saturation S' of $Ax(T) \cup S$ by $\mathcal{SP}$ is finite and contains no equality of the form $X = t$, where $X \notin Var(t)$. Then T is stably infinite and $(\mathcal{SP}, Ax(T))$ is deduction complete for T or, equivalently, T is a* **DCCSI***-theory.*

Proof: Stable infiniteness of T follows from the variable inactivity as proved in [25]. Consider now the deduction completeness problem. Let E be the set of elementary equalities in S'. Assume that there is some equality $c = c'$ between constants such that $T \cup S \models c = c'$. By refutation, $T \cup S \models c = c'$ iff $S \wedge c \neq c'$ is T-unsatisfiable. Hence, it must be possible to derive the empty clause by applying $\mathcal{SP}$ to the set $S' \cup \{c \neq c'\}$. Since S' is T-satisfiable and saturated, only inferences involving both clauses from (or inferred from) S' and $c \neq c'$ can yield the empty clause. If there is an inference between $c \neq c'$ and C' in S', then C' must be an equality between constants or variables. This is because the ordering used in $\mathcal{SP}$ is defined in such a way that a disequality is always bigger than an equality and hence an equality is maximal in a clause only if the latter contains no disequalities. If C' contains a variable, then C' must have the form $X = t$, where $X \notin Var(t)$. That would contradict the assumption of the lemma. If C' only contains constants, then C' is an equality between constants and the clause inferred from $c \neq c'$ and C' must be a disequality between constants. This means that an inference between $c \neq c'$ and a clause in S' is possible only if the latter is an equality between constants and derives a disequality between constants. Therefore, $E \cup \{c \neq c'\}$ suffices to infer the empty clause. Or, equivalently, $E \models c = c'$.

Let us consider a few theories in which this approach can be successfully applied.

Example 1 *Consider the theory of equality $\mathcal{E}$, axiomatized by an empty set of axioms. Let Φ be a set of ground flat literals. It is easy to see that the saturation of Φ w.r.t. $\mathcal{SP}$ contains only ground and flat literals (see, e.g., [1]). By Theorem 5, $\mathcal{E}$ is stably infinite and $(\mathcal{SP}, \emptyset)$ is deduction complete for $\mathcal{E}$.*

Example 2 *Let $\mathcal{L}$ be the theory of lists, axiomatized by the following set $Ax(\mathcal{L})$ of axioms:*

$$car(cons(X,Y)) = X \tag{1}$$

$$cdr(cons(X,Y)) = Y \tag{2}$$

$$cons(car(X), cdr(X)) = X \tag{3}$$

By induction on the length of the derivation w.r.t. $\mathcal{SP}$, we can show (see [1]) that for any set of ground flat literals S, the saturation of $Ax(\mathcal{L}) \cup S$ w.r.t. $\mathcal{SP}$ contains only literals of the following forms: (a) the empty clause, (b) the axioms in $Ax(\mathcal{L})$, (c) ground flat literals, and (d) equalities of the form $cons(b, cdr(a)) = a$ or $cons(car(a), b) = a$, where a, b are constants. Again by Theorem 5, $\mathcal{L}$ is stably infinite and $(\mathcal{SP}, Ax(\mathcal{L}))$ is deduction complete for $\mathcal{L}$.

Example 3 *Let $\mathcal{SC}$ be the theory finitely presented by the following set $Ax(\mathcal{SC})$ of axioms:*

$$\begin{array}{lr}
(Sel): & s_1(c(X_1,\ldots,X_n)) = X_1 \\
 & \ldots \\
 & s_n(c(X_1,\ldots,X_n)) = X_n \\
(Inj): & c(X_1,\ldots,X_n) = c(Y_1,\ldots,Y_n) \Rightarrow X_1 = Y_1 \\
 & \ldots \\
 & c(X_1,\ldots,X_n) = c(Y_1,\ldots,Y_n) \Rightarrow X_n = Y_n
\end{array}$$

By induction on the length of the derivation w.r.t. $\mathcal{SP}$, it is easy to show that for any set of ground flat literals S, the saturation of $Ax(\mathcal{L}) \cup S$ w.r.t. $\mathcal{SP}$ contains only clauses of the following forms: (a) the empty clause, (b) the axioms in $Ax(\mathcal{SC})$, (c) ground flat literals, (d) clauses of the form $a = b \Rightarrow a' = b'$, where a, b, a', b' are constants, and clauses of the forms: $(a_1 = c(Y_1,\ldots,Y_n) \Rightarrow a'_1 = Y_1), \ldots, (a_n = c(Y_1,\ldots,Y_n) \Rightarrow a'_n = Y_n)$ where $a_1, a'_1 \ldots, a_n, a'_n$ are constants. By Theorem 5, $\mathcal{SC}$ is stably infinite and $(\mathcal{SP}, Ax(\mathcal{SC}))$ is deduction complete for $\mathcal{SC}$.

4.2 Combining DCCSI theories

So far, we have introduced the concept of deduction complete satisfiability procedures and shown that such procedures can be flexibly built for an interesting class of theories. We are left with the problem of showing that deduction complete procedures can be modularly combined, taking thereby advantage of both Shostak (derivation of entailed equalities) and Nelson-Oppen (modularity) schemas. The combination algorithm for deduction complete satisfiability procedures is depicted in Figure 6, where $DC_{T_i}(\Phi)$ denotes the deduction complete satisfiability procedure in the theory T_i applied to the set of literals Φ. Its correctness is stated in the next theorem.

Theorem 6 *Let T_1 and T_2 be two signature-disjoint convex and stably infinite theories such that for each $i = 1, 2$, a deduction complete T_i-satisfiability procedure is available. Let Ω_i be a set of non-elementary T_i-literals for $i = 1, 2$, let E be a set of elementary equalities and let Δ be a set of elementary disequalities. Let $\varphi = (\Omega_1 \cup \Omega_2 \cup \Delta \cup E)$ and S' be a final configuration obtained by the repeated*

$\mathsf{Unsat}_{=1}$

$$\frac{\Omega_1; \Delta; E; \Omega_2}{false} \ \textbf{if } DC_{T_1}(\Omega_1 \cup E) = false$$

$\mathsf{Unsat}_{\neq}$

$$\frac{\Omega_1; \Delta; E; \Omega_2}{false} \ \textbf{if } x \neq y \in \Delta \text{ and } (x, y) \in E^*$$

$\mathsf{Deduction}_1$

$$\frac{\Omega_1; \Delta; E; \Omega_2}{\Omega_1; \Delta; E'; \Omega_2} \ \textbf{if } \left\{ \begin{array}{l} DC_{T_1}(\Omega_1 \cup E) = true\{E'\} \\ E'^* \neq E^* \end{array} \right.$$

Figure 6: Combination of deduction complete satisfiability procedures

application of the rules of Figure 6 on the initial configuration $S = \Omega_1; \Delta; E; \Omega_2$. Then,

- *if S' is $false$, then φ is $T_1 \cup T_2$-unsatisfiable;*
- *otherwise, S' is of the form $\Omega_1; \Delta; E'; \Omega_2$ and φ is $T_1 \cup T_2$-satisfiable. Furthermore, E' is deduction complete for φ modulo $T_1 \cup T_2$.*

Proof: Similar to the proof of Theorem 1. Let us consider the rules of Figure 6.

Termination. The rules return $false$ or strictly decrease the number of pairs of shared variables which are not equal modulo E.

Soundness. The rules are clearly sound.

Completeness. Similarly to Lemma 5, we proceed by contradiction. Consider a final configuration S' of the form $(\Omega_1; \Delta; E'; \Omega_2)$. According to the rules, $\Phi_i = (\Omega_i \cup \Delta \cup E)$ is necessarily T_i-satisfiable for $i = 1, 2$, otherwise either $\mathsf{Unsat}_{=1}$ or $\mathsf{Unsat}_{\neq}$ would apply according to Lemma 6. Assuming $\Phi_1 \cup \Phi_2$ is $T_1 \cup T_2$-unsatisfiable, we can apply Lemma 1 to exhibit a contradiction with the fact that S' is a final configuration.

Deduction Completeness. It remains to show that, if $S' = (\Omega_1; \Delta; E'; \Omega_2)$ is a final configuration, E' is deduction complete. By contradiction, assume that E' is not deduction complete. There exists $x = y$ such that $T_1 \cup T_2 \cup S' \models x = y$ and $E' \not\models x = y$. Therefore $S'' = (\Omega_1; \Delta \cup \{x \neq y\}; E'; \Omega_2)$ is unsatisfiable, and so S'' is reducible with combination rules of Figure 6. But if S'' is reducible, then S' is also reducible, which contradicts that S' is a final configuration.

Corollary 1 DCCSI *is closed under disjoint union.*

Proof: Let $(\mathcal{I}_k, Ax_k)$ be a deduction complete inference system for T_k $(k = 1, 2)$. Consider the basic inference system $\mathcal{I}$ for $T_1 \cup T_2$ defined as the union of $\mathcal{I}_1$

and $\mathcal{I}_2$ plus the classical rules to purify heterogeneous equalities. We assume that rules in $\mathcal{I}_k$ are only applied to pure non-elementary equalities. Clauses in $Ax = Ax_1 \cup Ax_2$ are $T_1 \cup T_2$-valid. We show below that $(\mathcal{I}, Ax)$ is terminating, refutation complete, and deduction complete for $T_1 \cup T_2$.

Termination. We assume that a set of ground equalities Γ is split into a set Γ_H of impure equalities, a set E of elementary equalities, and a set Γ_i of i-pure equalities (for $i = 1, 2$). The termination of $\mathcal{I}$ can be proved by using a lexicographic combination of complexity measures defined and ordered as follows:

(H) the number of alien positions occurring in Γ_H,

(0) the number of E-equivalence classes,

(1) the number of steps to reach a $\mathcal{I}_1$-normal form of $Ax_1 \cup \Gamma_1$, and

(2) the number of steps to reach a $\mathcal{I}_2$-normal form of $Ax_2 \cup \Gamma_2$.

Termination follows from the facts that any $\mathcal{I}$-transition strictly decreases the complexity measure defined above, and any ξ-transition does not increase **(H)** and strictly decreases **(0)**.

Soundness. When $false$ is derived by $\mathcal{I}$, this result is correct since any $\mathcal{I}$-transition transforms a set of clauses into a $T_1 \cup T_2$-equivalent one.

Completeness. Consider a $\mathcal{I}$-normal form S of Γ such that $false \notin S$. Let $E' = E(S)$. S is also a $\mathcal{I}$-normal form of $\Gamma_1 \cup \Gamma_2 \cup E$, where Γ_1, Γ_2, E are obtained from Γ by applying the purification rules. The configuration $\Gamma_1; \emptyset; E'; \Gamma_2$ is necessarily a final configuration w.r.t. the combination rules depicted in Figure 6, otherwise this would contradict the assumption that $(\mathcal{I}_k, Ax_k)$ is deduction complete for T_k $(k = 1, 2)$. According to Theorem 6, $\Gamma_1 \cup \Gamma_2 \cup E'$ is $T_1 \cup T_2$-satisfiable (and so is Γ). Moreover, E' is deduction complete for $\Gamma_1 \cup \Gamma_2 \cup E'$, and so for Γ. Consequently, $(\mathcal{I}, Ax)$ is deduction complete for $T_1 \cup T_2$.

5 Explaining Combination

Although combining decision procedures is already a crucial activity for their effective use in large systems, this is seldom sufficient: procedures should also provide the capability of explaining their results in order to be correctly integrated. In this Section, we study how to augment the interface capabilities of available decision procedures to compute such explanations and to modularly combine them. The decision procedures that we want to combine can conclude either unsatisfiability or satisfiability and can produce entailed elementary equalities in case of satisfiability. So explaining these results involves both the production of an unsatisfiability witness, called conflict set, given by a small set of unsatisfiable literals, and the production of explanations for entailed elementary equalities. So a first step in Section 5.1 is to precisely define conflict sets, explanations, and the minimality property in each case. Then in order to build the deductive part of the justification in case of satisfiability, the notion of explanation graph is proposed in Section 5.2. Such graphs record the successive justifications used in the construction of entailment proofs. For decision procedures we are interested in, the notion of quasi-conflict sets is proposed in Section 5.3 to represent proofs of unsatisfiability integrating both conflict sets and explanation graphs. Then, in Section 5.4, we propose to use explanation engines instead of decision procedures. Such engines return either an explanation graph in case of satisfiability or a quasi-conflict set in case of unsatisfiability. Eventually it is proved that explanation engines can be built in a modular way for signature-disjoint combination of convex and stably infinite theories.

5.1 Conflict sets and explanations

The starting point of our development is the notion of conflict set, commonly used in the context of unsatisfiability. Informally, a conflict set is a subset of an unsatisfiable set of literals which is already unsatisfiable. For efficiency, a suitable notion of minimality for conflict sets is introduced. A T-conflict set CS of literals is *minimal* if there is no $CS' \subset CS$ such that CS' is a T-conflict set. In the context of satisfiability, the dual notion of conflict set is the concept of explanation. A T-*explanation* of an equality e is a T-satisfiable set φ of literals such that $T \models \varphi \Rightarrow e$. A T-explanation of e is *minimal* if there is no $\varphi' \subset \varphi$ such that $T \models \varphi' \Rightarrow e$. We omit the theory T when it is clear from the context. The next proposition states the duality between the two notions and follows directly from the definitions.

Proposition 5 *A T-satisfiable set of literals φ is a minimal T-explanation for an equality e iff $\varphi \cup \{\neg e\}$ is a minimal T-conflict set.*

The convexity assumption allows us to further characterize conflict sets.

Proposition 6 *If T is a convex theory, then any minimal conflict set contains at most one disequality. If T is a convex theory axiomatized by a set of equalities, then any minimal conflict set contains exactly one disequality.*

For example, $\mathcal{E}$ is a convex theory such that any minimal conflict set contains exactly one disequality. Notice that $\mathcal{LA}$, and hence also $\mathcal{LA}^{\leq}$, does not satisfy this property (e.g, $\{x = 3, x = 2\}$ is a minimal $\mathcal{LA}$-conflict set). We now explain how to store explanations by using undirected and acyclic graphs.

5.2 Explanation Graphs

We use some standard notions as undirected graph, acyclic graph, subgraph, connected graph, path, simple path, elementary path, and connected components. In the rest of the paper, we only consider acyclic undirected graphs, often called graphs for the sake of simplicity. An undirected graph G is a pair (V, E) where V (also written as $Vertex(G)$) is a finite set of vertices and E (also written as $Edge(G)$) is a set of unordered pairs written as (v, w) for v, w in V. $G_{\emptyset}^{V}$ denotes the graph whose vertices in V are connected by no edge, i.e. $G_{\emptyset}^{V} = (V, \emptyset)$. The subgraph relation is denoted by $\subseteq$. Let $G = (V, E)$ be an acyclic undirected graph. The set $ElemPath(G, x, y)$ denotes the set of edges in an elementary path between x and y in G, i.e. if $v_0, ..., v_n$ is an elementary path where v_0 is x and v_n is y, then $ElemPath(G, x, y)$ is the set of edges $(v_{i-1}, v_i) \in Edge(G)$, for $i = 1, ..., n$. Given two distinct vertices x and y, $ElemPath(G, x, y)$ is empty iff

x and y are not in the same connected component of G. The set of *pairs of connected vertices in* G is $CP(G) = \{(x, y) \mid x, y \in V \text{ and } ElemPath(G, x, y) \neq \emptyset\}$. Notice that $E^* = CP(G) \cup \{(x, x) \mid x \in V\}$.

Two preliminary remarks about the relationship between an acyclic undirected graph $G = (V, E)$ and a set of elementary equalities are useful. First, let us observe that an elementary equality can be regarded as an unordered pair and edges of G are unordered pairs. So, we write $(x, y) \in E$ as $x = y$ and define the *set of elementary equalities of the graph* G as $Eq(G) = \bigcup_{(x,y)\in E}\{x = y\}$. Second, it is easy to see that a set of elementary equalities E is minimal iff there exists an acyclic undirected graph G such that $Eq(G) = E$.

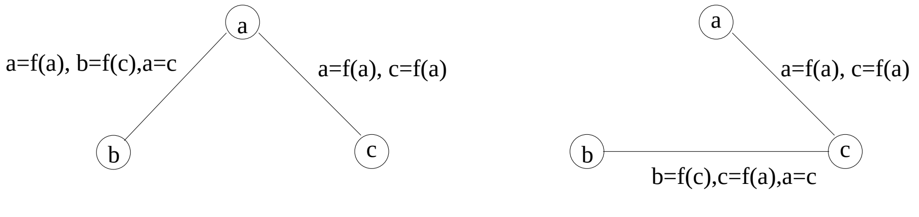
Definition 4 *Let T be a theory, φ be a set of T-literals, and $G = (V, E)$ be an acyclic undirected graph such that E is a set totally ordered by some ordering $<_E$. G is an* explanation graph *of φ if (i) V is the set of constants occurring in φ, (ii) there exists a labelling function $\mathcal{L}_G$ with domain E and co-domain $\mathbf{2}^{\varphi\cup CP(G)}$[6], (iii) the following properties are satisfied for any $v_1 = v_2 \in E$:*

(iii.a) $\mathcal{L}_G(v_1 = v_2)$ is T-satisfiable and $T \models \mathcal{L}_G(v_1 = v_2) \Rightarrow v_1 = v_2$,

(iii.b) for each $v_1' = v_2'$ in $\mathcal{L}_G(v_1 = v_2)\backslash\varphi$ we have that $e <_E (v_1 = v_2)$, for any e in $ElemPath(G, v_1', v_2')$.

The set of literals of φ in G *is $Lit(G) = \varphi \cap (\bigcup_{e\in E}\mathcal{L}_G(e))$. An edge $v_1 = v_2 \in E$ is* minimally explained *if $\mathcal{L}_G(v_1 = v_2)$ is a minimal T-explanation for $v_1 = v_2$. A explanation graph is* minimally explained *if all its edges are minimally explained. An explanation graph G' is* smaller *than an explanation graph G, denoted by $G' \sqsubseteq G$, if $Edge(G') \subseteq Edge(G)$ and $\forall e \in Edge(G')$, $\mathcal{L}_{G'}(e) \subseteq \mathcal{L}_G(e)$. An explanation graph G is* minimal *for E if $E \subseteq CP(G)$ and there is no explanation graph G' such that $G' \sqsubset G$ and $E \subseteq CP(G')$. An explanation graph G of a T-satisfiable set φ of literals is* deduction complete *(modulo T) if $Eq(G)$ is deduction complete for φ (modulo T).*

Example 4 *Consider the theory of equality $\mathcal{E}$ and the set of literals $\varphi = \{a = f(a), b = f(c), c = f(a)\}$. Two explanation graphs of φ are depicted below:*



In both cases, $a = c$ is the smallest edge of the graph, and is used in the explanation of another edge.

In the definition above, edges are ordered to express the fact that explanation graphs are built dynamically. The ordering $<_E$ on edges corresponds to the order of insertion of edges in the graph. Adding an edge $x = y$ to the explanation graph $G = (V, E)$ of the set φ of literals is defined as follows: if x and y are two distinct vertices in V such that $x = y \notin CP(G)$, and L is a set of T-literals in $\varphi \cup CP(G)$ such that L is T-satisfiable and $T \models L \Rightarrow x = y$,

[6] Let X be a set, $\mathbf{2}^X$ denotes the power-set of X.

then $Insert(G, x = y, L)$ denotes the explanation graph $G' = (V, E')$, where $E' = E \cup \{x = y\}$, $\mathcal{L}_{G'}$ is such that $\mathcal{L}_{G'}(x = y) = L$, $\forall e \in E, \mathcal{L}_{G'}(e) = \mathcal{L}_G(e)$, and $<_{E'}$ is the smallest ordering containing $<_E$ such that $\forall e \in E, e <_{E'} x = y$. From now on, we assume that $<_{E'}$ and $<_E$ coincide on elements of E whenever $G = (V, E)$ and $G' = (V', E')$ are two explanation graphs such that $E \subseteq E'$.

We now consider the case of an explanation graph G obtained by adding a set of elementary equalities in a given order. (This will be important for our combination schema described in Section 5.4 below.) For the sake of conciseness, we write $UF(E)$ to abbreviate the graph obtained by the sequence of insertions above.[7] If V is a set of variables, $UF^V(E)$ is the explanation graph obtained by adding V to the set of vertices of $UF(E)$. It is not difficult to see that for any set E of elementary equalities, $UF(E)$ is a minimally explained explanation graph of E such that $Eq(UF(E))$ is a minimal set of elementary equalities (included in E). Moreover, $UF(E)$ is a deduction complete explanation graph of E. More generally, it is possible to construct (deduction complete) explanation graphs when a (deduction complete) satisfiability procedure is known. For a (deduction complete) inference-based satisfiability procedure, an explanation graph of a satisfiable formula can be constructed by collecting the literals used in each rule application of the derivation. For the particular case of the theory of equality $\mathcal{E}$, a congruence closure with explanation leads to a minimally explained and deduction complete explanation graph for any set of flat equalities, as shown in [33]. In that case, the explanation of an edge $x = y$ is either $\{x = y\}$ or $\{x = f(x_1, \dots, x_n), y = f(y_1, \dots, y_n), x_1 = y_1, \dots, x_n = y_n\}$, and so is minimal.

5.3 Quasi-conflict sets

Let us now concentrate on unsatisfiability proofs. The notion of conflict set is not sufficiently structured to take into account the deduction steps in the proof and their explanation. This is why we consider a richer structure called quasi-conflict sets.

Definition 5 (Quasi-conflict sets) *Let φ be an unsatisfiable set of literals, ψ a subset of φ, G an explanation graph of φ, and E a set of equalities. The triplet (ψ, E, G) is a* quasi-conflict set *of φ if $E \subseteq CP(G)$, $\psi \cup E$ is unsatisfiable, and $E \neq \emptyset$ implies that ψ is satisfiable.*

A quasi-conflict set (ψ', E', G') is smaller *than a conflict set (ψ, E, G), denoted by $(\psi', E', G') \preceq (\psi, E, G)$, if $\psi' \subseteq \psi$, $E' \subseteq E$ and $G' \sqsubseteq G$. A quasi-conflict set (ψ, E, G) is* minimal *if there is no quasi-conflict (ψ', E', G') such that $(\psi', E', G') \prec (\psi, E, G)$.*

Notice that if (ψ, E, G) is a quasi-conflict set, then $E \neq \emptyset$ iff ψ is satisfiable. Also, if φ is a conflict set, then $(\varphi, \emptyset, G_\emptyset^{Var(\varphi)})$ is a quasi-conflict set.

Example 5 *Let us consider the theory $\mathcal{LA}$ and the set of literals $\varphi = \{z = x + y, x - y = 1, x = y + u, u = 0\}$. Let G denote the explanation graph $(\{x, y\}, \{x = y\})$ such that $\mathcal{L}_G(x = y) = \{x = y + u, u = 0\}$. Then $(\{x - y = 1\}, \{x = y\}, G)$ is a quasi-conflict set of φ.*

[7] UF abbreviates Union-Find since the sequence of insertions is typically implemented using this data structure (see [18, 13, 29]).

Proposition 7 *If (ψ, E, G) is a quasi-conflict set of φ, then $\psi \cup Lit(G)$ is a conflict set of φ.*

Proof: If (ψ, E, G) is a quasi-conflict set of φ, then $\psi \cup E$ is unsatisfiable. By definition, $Lit(G) \subseteq \varphi$ and $Lit(G)$ entails E. Consequently, $\psi \cup Lit(G)$ is unsatisfiable and $\psi \cup Lit(G) \subseteq \varphi$. So $\psi \cup Lit(G)$ is a conflict set of φ.

Given a quasi-conflict set (ψ, E, G) of φ, $\psi \cup Lit(G)$ is called the *conflict set associated to* (ψ, E, G).

Example 6 *(Example 5 continued). The conflict set associated to $(\{x - y = 1\}, \{x = y\}, G)$ is $\{x - y = 1, x = y + u, u = 0\}$.*

The set $Lit(G)$ provides an explanation of equalities in E, but it is a superset of what we need: it is sufficient to consider the subgraph of G obtained by focusing only on the paths in G "connecting" the equalities in E.

Definition 6 *Let G be an explanation graph of φ, $x = y \in CP(G)$ and $E \subseteq CP(G)$. The set of* explanation edges of $x = y$ in G *is the subset of $Edge(G)$ defined as follows:*

$$Exe(G, x = y) = ElemPath(G, x, y) \ \cup \ (\bigcup_{e \in ElemPath(G,x,y)} \ \bigcup_{e' \in \mathcal{L}_G(e) \setminus \varphi} Exe(G, e')).$$

The set of explanation edges of E *in G is $ExE(G, E) = \bigcup_{e \in E} Exe(G, e)$.*

The restriction of G to E *is the subgraph $G_{|E}$ of G such that $Edge(G_{|E}) = ExE(G, E)$ and $\forall e \in Edge(G_{|E})$, $\mathcal{L}_{G_{|E}}(e) = \mathcal{L}_G(e)$.*

Proposition 8 *$G_{|E}$ is minimal for E iff $G_{|E}$ is minimally explained.*

Proof: ($\Leftarrow$) Let us assume that $G_{|E}$ is not minimal for E. There exists $G' \sqsubset G_{|E}$ such that $E \subseteq CP(G')$. If $G' \subset G_{|E}$, then this would contradict the fact that $E \subseteq CP(G')$. Therefore, $Edge(G') = Edge(G)$ and there exists an edge e of G' such that $\mathcal{L}_{G'}(e) \subset \mathcal{L}_{G_{|E}}(e)$. So $G_{|E}$ is not minimally explained.
($\Rightarrow$) If $G_{|E}$ is not minimally explained, then it is possible to construct a graph $G' \sqsubset G_{|E}$ such that $E \subseteq CP(G')$. Consequently, $G_{|E}$ is not minimal for E.

We are now ready to give a characterization of minimal quasi-conflict sets.

Theorem 7 *A quasi-conflict set (ψ, E, G) is minimal iff $\psi \cup E$ is a minimal conflict set and G is minimal for E.*

Proof: ($\Leftarrow$) By contradiction, assuming that there exists a quasi-conflict set (ψ', E', G') such that $(\psi', E', G') \prec (\psi, E, G)$. If $\psi' \subset \psi$ or $E' \subset E$, then this would contradict that $\psi \cup E$ is a minimal conflict set. Therefore $\psi' = \psi$, $E' = E$ and $G' \sqsubset G$, where $E \subseteq CP(G')$. This contradicts that G is minimal for E.
($\Rightarrow$) If $\psi \cup E$ is not a minimal conflict set or G is not minimal for E, then it is possible to construct a quasi-conflict set (ψ', E', G') such that $(\psi', E', G') \prec (\psi, E, G)$.

Example 7 *(Example 5 continued). The quasi-conflict set $(\{x - y = 1\}, \{x = y\}, G)$ is minimal since $\{x - y = 1, x = y\}$ is a minimal conflict set and G is minimal for $\{x = y\}$.*

5.4 Explanation Engines and Their Combination

We now adapt the combination algorithm depicted in Figure 6 for deduction complete satisfiability procedures in order to generate $T_1 \cup T_2$-conflict sets. More generally, the goal is to built an explanation for each truth value returned by a satisfiability procedure. To develop our main combination result, we introduce the notion of *explanation engine*: this is a component capable of computing (1) an explanation graph in case of satisfiability and (2) a quasi-conflict set in case of unsatisfiability.

Definition 7 (Explanation engine) *Let T be a theory and $\mathcal{L}$ be a set of (finite) sets of non-elementary T-literals closed under union. A* T-explanation engine *for $\mathcal{L}$ is a T-satisfiability procedure, denoted by μEX_T, such that, for any $\Omega \in \mathcal{L}$ and any minimal set of elementary equalities E:*

1. *If $\Omega \cup E$ is T-satisfiable, then μEX_T returns $true\{G\}$ where G is deduction complete for $\Omega \cup E$.*
2. *If $\Omega \cup E$ is T-unsatisfiable, then μEX_T returns $false\{(\Omega', E', G)\}$ where (Ω', E', G) is a quasi-conflict set of $\Omega \cup E$.*

If μEX_T computes minimal quasi-conflict sets and minimally explained explanation graphs, then μEX_T is said minimal.

Remark 1 *As shown in [33], the Gauss elimination algorithm can be adapted to build minimal $\mathcal{LA}$-conflict sets and minimal $\mathcal{LA}$-explanations of elementary equalities, and so minimal quasi-conflict sets and minimally explained explanation graphs. This allows us to obtain a minimal $\mathcal{LA}$-explanation engine. As another example, a congruence closure algorithm computing a minimally explained explanation graph for any given set of flat equalities (see end of Section 5.2) provides a minimal $\mathcal{E}$-explanation engine for all sets of flat equalities.*

Figure 7 presents a variant of the Nelson-Oppen combination method for the union of two arbitrary signature-disjoint, stably infinite, and convex theories where explanation engines are used in place of satisfiability procedures, and where explanation graphs are used to encode the entailment of elementary equalities together with their explanations.

Theorem 8 *Let T_1 and T_2 be two signature-disjoint convex and stably infinite theories such that for each $i = 1, 2$, a T_i-explanation engine for $\mathcal{L}_i$, E be a set of elementary equalities and Δ a set of elementary disequalities. Let $\Omega_i \in \mathcal{L}_i$ for $i = 1, 2$, $\varphi = m(\Omega_1 \cup \Omega_2 \cup \Delta \cup E)$ and S' be a final configuration obtained by the repeated application of the rules of Figure 7 on the initial configuration $S = \Omega_1; \Delta; UF^{Var(\varphi)}(E); \Omega_2$.*

- *If S' is of the form $false\{(\Omega', E', G)\}$, then φ is $T_1 \cup T_2$-unsatisfiable and (Ω', E', G) is a quasi-conflict set of φ.*
- *Otherwise, S' is of the form $\Omega_1; \Delta; G; \Omega_2$ and φ is $T_1 \cup T_2$-satisfiable. Furthermore, G is deduction complete for φ modulo $T_1 \cup T_2$.*

Proof: It follows directly from Theorem 6. The rules depicted in Figure 7 can be seen as a refinement of rules in Figure 6, where an explanation graph

Unsat$_{=1}$

$$\frac{\Omega_1; \Delta; G; \Omega_2}{false\{(\Omega'_1, E'_1, G'_{|E'_1})\}}$$

$$\textbf{if} \begin{cases} \mu EX_{T_1}(\Omega_1, Eq(G)) = false\{(\Omega'_1, E'_1, G_1)\} \\ G' = Merge(G, G_1) \end{cases}$$

Unsat$_{\neq}$

$$\frac{\Omega_1; \Delta; G; \Omega_2}{false\{(\{x \neq y\}, \{x = y\}, G)\}}$$

if $(x, y) \in CP(G)$ and $x \neq y \in \Delta$

Deduction$_1$

$$\frac{\Omega_1; \Delta; G; \Omega_2}{\Omega_1; \Delta; G'; \Omega_2}$$

$$\textbf{if} \begin{cases} \mu EX_{T_1}(\Omega_1, Eq(G)) = true\{G_1\} \\ G' = Merge(G, G_1) \\ G' \neq G \end{cases}$$

Legenda: Ω_i is a set of non-elementary T_i-literals, for $i = 1, 2$. Δ is a set of elementary disequalities. Symmetric rules are not depicted here for the sake of conciseness and they can be obtained by changing the subscript 1 into 2 in the rules above. The function $Merge$ is defined as follows:

```
Function $Merge(G, G')$
  $G'' := G$
    Foreach $(x, y) \in Edge(G') \backslash Edge(G)$
      $G'' := Insert(G'', x = y, \mathcal{L}_{G'}(x = y))$
    EndForeach
  Return $G''$
```

Figure 7: Combination of explanation engines

G represents the set of elementary equalities E and explanation engines μEX_T correspond to deduction complete satisfiability procedures DC_T. Assuming $E = Eq(G)$, there is an obvious correspondence between the two sets of rules:

- In Unsat$_{=i}$, $\mu EX_{T_i}(\Omega_i, E) = false\{\dots\}$ iff $DC_{T_i}(\Omega_i \cup E) = false$,
- In Unsat$_{\neq}$, $(x, y) \in CP(G)$ iff $(x, y) \in E^*$,
- In Deduction$_i$, given $E' = Eq(G')$, we have $G' \neq G$ iff $E'^* \neq E^*$.

Theorem 8 has an interesting consequence. If the previous combination algorithm is applied with an empty set of elementary disequalities Δ, it provides a $T_1 \cup T_2$-explanation engine. Given two explanation engines respectively for $\mathcal{L}_1$ and $\mathcal{L}_2$, as defined in Definition 7, we get an explanation engine for $\mathcal{L}_1 \cup \mathcal{L}_2$, where $\mathcal{L}_1 \cup \mathcal{L}_2$ denotes the smallest set closed under union including $\mathcal{L}_1$ and $\mathcal{L}_2$.

Corollary 2 (Modular construction of explanation engines) *Let T_1 and T_2 be two signature-disjoint, convex, and stably infinite theories such that a T_1-explanation engine for $\mathcal{L}_1$ and a T_2-explanation engine for $\mathcal{L}_2$ are known. The combination rules depicted in Figure 7 provide a $T_1 \cup T_2$-explanation engine for $\mathcal{L}_1 \cup \mathcal{L}_2$. Moreover, this $T_1 \cup T_2$-explanation engine is minimal if the T_1-explanation engine and the T_2-explanation engine are minimal.*

Proof: The preservation of minimality follows from Proposition 8 and Theorem 7.

6 Conclusion and Related Work

Rational reconstruction. Our presentation of combination schemas for disjoint unions of theories in various classes highlights the key ideas underlying each combination and allows proofs of correctness which are easy to grasp.

Similarly to [20], the abstract schema presented in Section 3.2 for combining **SH** theories seems to emphasize the importance of the solver w.r.t. the canonizer. In fact, if the solved form returned by the solver is also canonical, the canonizer can be trivially implemented as the identity function. Nonetheless, we believe that the concept of canonizer is quite important mainly for two reasons. First, it offers the entry point to refinements of the proposed schema to increase efficiency. In fact, solving a set of equalities in “one-shot”, as done when applying rule $\mathsf{Solve} - \mathsf{success}_1$, may not be as efficient as solving equalities incrementally, as e.g. in [34, 23]. This can be incorporated in our schema by refining the inference system SH along the lines described in [10] so that the solver is applied to only one equality at a time and the canonizer needs to return a canonical form for arbitrary terms. The second reason is that the concept of canonizer is a useful basic building block, together with solvers, for constructing what we have called deduction complete satisfiability procedures (cf. Section 4).

The particular case of combining a **SI**-theory (e.g., the theory of equality $\mathcal{E}$) and one in **SH** considered in Section 3.3 has been extensively studied by many researchers following [37]. It is possible to derive the correctness of such combination schemas in our framework by using the following observations: (i) NO, SH, NS are correct, (ii) the class of theories **CSI** is closed under disjoint union, and (iii) the class **SH** is contained in **CSI**. Similar results are given in [26]. As an additional remark, we mention the possibility of refining the abstract inference systems presented here with strategies as done in [10], so to get a more fine-grained rule-based implementation which mimics a Shostak procedure as described in [35]. We have not done this here, since we were more focused on modularity rather than efficiency.

Non-stably infinite theories. In [4], the authors have shown that all convex theories with no trivial models are stably infinite, and therefore they can be combined using the NO combination schema. In Section 3.4, we go a step further by showing that arbitrary convex theories can be combined provided that we know whether component theories admit a trivial model or not. In practice this requirement is not restrictive as almost all useful theories enjoy this property. Our result is along the line of [5] in which the class of $\exists^n$-decidable and $\exists^\infty$-decidable theories is defined and shown to be modular. Basically $\exists^n$-decidability (resp. $\exists^\infty$-decidability) allows us to check the existence of a model of finite (resp. infinite) cardinality. The combination method proposed in Section 3.4 exploits the hypothesis of convexity and the fact that we know whether or not component theories have a trivial model. It can be seen as an instance with a more operational flavour of the non-deterministic method of [5].

Deduction completeness. We have introduced the notion of deduction complete satisfiability procedure and defined the class of theories **DCCSI**. Basically they have satisfiability procedures defined as inference-based systems with the capability of computing all entailed elementary equalities with no overhead. We have shown that the class of **DCCSI** theories is closed under disjoint union. The concept of deduction complete inference-based satisfiability procedures offers an interesting trade-off between modularity and efficiency for the problem of solving satisfiability in disjoint combinations of theories under a common interface.

There were some attempts in the development of SMT solving to extend decision procedures with the capability of deriving entailed facts while checking for satisfiability (see, e.g., [36] for an overview on this and related issues). However, in such a line of work, deduced facts are used to prune the search space of a Boolean solver rather than to combine procedures and completeness is regarded as detrimental to performances. In this paper, we restrict our interest to inference-based satisfiability procedures for convex theories built over the equality predicate only. An interesting direction to explore is to consider inference-based satisfiability procedures for arbitrary convex theories involving other predicates than equality and generalize the deduction completeness approach in that case.

Modularity of conflict sets. We have also proposed a method to modularly build conflict sets in unions of theories by refining the Nelson-Oppen combination schema. The key concept of explanation graph allows us to encode the fact that a certain elementary equality is a logical consequence of a set of elementary equalities. Explanation engines formalize proof-producing procedures capable of computing explanation graphs. We have shown how to re-use efficient proof-producing procedures available in the literature to build explanation engines. Furthermore, explanation engines for unions of several theories can be obtained as a by-product of our combination method. A suitable notion of minimality related to quasi-conflict sets in unions of theories was also investigated.

An alternative approach to producing conflict sets in combinations of theories has been proposed in [7], which does not require the direct combination of the solvers for the component theories. While the technique of [7] may yield better performances for SMT problems, we believe our combination method could become a key ingredient in the certification of the results produced by solvers to be integrated in skeptical proof assistants (see, e.g., [19]). In a slightly different context, our techniques could also be used to build *equational reasoners* having the capability of computing a (small) witness of unsatisfiability for equational problems such as unification, matching, and word problems. For equational theories, there are satisfiability procedures with the property of deriving elementary equalities (like unification or matching algorithms) and deductive combination methods based on the propagation of elementary equalities [6, 31]. Applying the techniques developed here to more general equational reasoners appears to be a promising line of research.

References

[1] A. Armando, S. Ranise, and M. Rusinowitch. A Rewriting Approach to Satisfiability Procedures. *Journal of Information and Computation*, 183(2):140–164, June 2003.

[2] Alessandro Armando, Maria Paola Bonacina, Silvio Ranise, and Stephan Schulz. New results on rewrite-based satisfiability procedures. *ACM Transactions on Computational Logic*, 2008. . To appear.

[3] L. Bachmair, A. Tiwari, and L. Vigneron. Abstract Congruence Closure. *Journal of Automated Reasoning*, 31(2):129–168, 2003.

[4] C. W. Barrett, D. L. Dill, and A. Stump. A generalization of Shostak's method for combining decision procedures. In A. Armando, editor, *Proc. of the 4th International Workshop on Frontiers of Combining Systems, FroCoS'02 (Santa Margherita Ligure, Italy)*, volume 2309 of *LNCS*, pages 132–147. Springer, April 2002.

[5] M. P. Bonacina, S. Ghilardi, E. Nicolini, S. Ranise, and D. Zucchelli. Decidability and undecidability results for nelson-oppen and rewrite-based decision procedures. In *Proc. of the 3rd Int. Conference on Automated Reasoning (IJCAR'06), Seattle, WA, USA*, number 4130 in LNAI, pages 513–527. Springer, August 2006.

[6] Alexandre Boudet. Combining unification algorithms. *Journal of Symbolic Computation*, 16(6):597–626, December 1993.

[7] Marco Bozzano, Roberto Bruttomesso, Alessandro Cimatti, Tommi Junttila, Peter van Rossum, Silvio Ranise, and Roberto Sebastiani. Efficient theory combination via boolean search. *Information and Computation*, 204(10):1493–1525, 2006.

[8] A. R. Bradley, Z. Manna, and H. B. Sipma. What's decidable about arrays? In *Proc. of the 7th Int. Conf. on Verification, Model Checking, and Abstract Interpretation (VMCAI'06)*, volume 3855 of *LNCS*, pages 427–442. Springer, 2006.

[9] J. Burg, S.-D. Lang, and C. E. Hughes. Intelligent Backtracking in CLP(R). *Ann. Math. Artif. Intell.*, 17(3-4):189–211, 1996.

[10] S. Conchon and S. Krstić. Strategies for combining decision procedures. In *Proc. of the 9th Int. Conference on Tools and Algorithms for the Construction and Analysis of Systems, (TACAS'03)*, volume 2619 of *LNCS*, pages 537–553. Springer, April 2003.

[11] S. Conchon and Sava Krstić. Canonization for disjoint unions of theories. In F. Baader, editor, *Proc. of the 19th International Conference on Automated Deduction (CADE'03)*, volume 2741 of *LNCS*, pages 197–211, Miami Beach, FL, USA, July 2003. Springer.

[12] D. Cyrluk, P. Lincoln, and N. Shankar. On Shostak's decision procedure for combinations of theories. In M. A. McRobbie and J.K. Slaney, editors, *Proc. of the 13th International Conference on Automated Deduction, (CADE'96), New Brunswick, NJ*, volume 1104 of *LNAI*, pages 463–477. Springer, 1996.

[13] L. de Moura, H. Rueß, and N. Shankar. Justifying equality. In C. Tinelli and S. Ranise, editors, *Selected Papers from the Workshops on Disproving and the Second International Workshop on Pragmatics of Decision Procedures (PDPAR 2004)*, volume 125, Issue 3 of *Electronic Notes in Theoretical Computer Science (ENTCS)*, pages 69–85, 2005.

[14] N. Dershowitz and J.-P. Jouannaud. *Handbook of Theoretical Computer Science*, volume B: Formal Models and Semantics, chapter 6: Rewrite Systems, pages 244–320. Elsevier and MIT Press, 1990.

[15] David Detlefs, Greg Nelson, and James B. Saxe. Simplify: a theorem prover for program checking. *Journal of the ACM (JACM)*, 52(3):365–473, 2005.

[16] P. J. Downey, R. Sethi, and R. E. Tarjan. Variations on the common subexpression problem. *J. ACM*, 27(4):758–771, 1980.

[17] H. B. Enderton. *A Mathematical Introduction to Logic.* Ac. Press, Inc., 1972.

[18] P. Fontaine. *Techniques for Verification of Concurrent Systems with Invariants.* PhD thesis, Université de Liège, 2004.

[19] Pascal Fontaine, Jean-Yves Marion, Stephan Merz, Leonor Prensa Nieto, and Alwen Tiu. Expressiveness + Automation + Soundness: Towards Combining SMT Solvers and Interactive Proof Assistants. In *12th Intl. Conf. Tools and Algorithms for the Construction and Analysis of Systems (2006)*, volume 3920 of *LNCS*, pages 167–181, 2006.

[20] H. Ganzinger. Shostak light. In A. Voronkov, editor, *Proc. of the 18th International Conference on Automated Deduction, (CADE'02)*, volume 2392 of *LNCS*, pages 332–346. Springer, July 2002.

[21] Silvio Ghilardi, Enrica Nicolini, Silvio Ranise, and Daniele Zucchelli. Decision procedures for extensions of the theory of arrays. *Annals of Mathematics and Artificial Intelligence*, 50(3-4):231–254, 2007.

[22] D. Kapur. Shostak's congruence closure as completion. In *Proc. of Rewriting Techniques and Applications, 8th International Conference, (RTA'97), Sitges, Spain, June 2-5*, volume 1232 of *LNCS*, pages 23–37. Springer, 1997.

[23] D. Kapur. A rewrite rule based framework for combining decision procedures. In *Proc. of the 4th Int. Workshop on Frontiers of Combining Systems, (FroCos'02)*, volume 2309 of *LNCS*, pages 87–102. Springer, 2002.

[24] H. Kirchner, S. Ranise, C. Ringeissen, and D.-K. Tran. On superposition-based satisfiability procedures and their combination. In *Proc. of Second International Colloquium on Theoretical Aspects of Computing, (ICTAC'05), Hanoi, Vietnam*, volume 3722 of *LNCS*, pages 594–608. Springer, October 2005.

[25] H. Kirchner, S. Ranise, C. Ringeissen, and D.-K. Tran. Automatic Combinability of Rewriting-Based Satisfiability Procedures. In *Proc. of the 13th Int. Conference on Logic for Programming, Artificial Intelligence, and Reasoning, (LPAR'06)*, volume 4246 of *LNAI*, pages 542–556, Phnom Penh, Cambodia, November 2006. Springer.

[26] Z. Manna and C. G. Zarba. Combining decision procedures. In *Formal Methods at the Cross Roads: From Panacea to Foundational Support*, volume 2757 of *LNCS*, pages 381–422. Springer, 2003.

[27] C. Marché. Normalized rewriting: An alternative to rewriting modulo a set of equations. *Journal of Symbolic Computation*, 21(3):253–288, 1996.

[28] G. Nelson and D. C. Oppen. Simplification by cooperating decision procedures. *ACM Transations on Programming Languages and Systems*, 1(2):245–257, October 1979.

[29] R. Nieuwenhuis and A. Oliveras. Proof-Producing Congruence Closure. In *Proc. of the 16th International Conference on Rewriting Techniques and Applications, (RTA'05), Nara, Japan*, volume 3467 of *LNCS*, pages 453–468. Springer, 2005.

[30] R. Nieuwenhuis and A. Rubio. Paramodulation-based theorem proving. In A. Robinson and A. Voronkov, editors, *Hand. of Automated Reasoning*, pages 371–443. Elsevier and MIT Press, 2001.

[31] Tobias Nipkow. Combining matching algorithms: The regular case. *Journal of Symbolic Computation*, 12:633–653, 1991.

[32] S. Ranise, C. Ringeissen, and D.-K. Tran. Nelson-Oppen, Shostak and the Extended Canonizer : A Family Picture with a Newborn. In *Proc. of First International Colloquium on Theoretical Aspects of Computing, (ICTAC'04), Guiyang, China*, volume 3407 of *LNCS*, pages 372–386. Springer, September 2004.

[33] S. Ranise, C. Ringeissen, and D.-K. Tran. Combining Proof-Producing Decision Procedures. In *Proc. of the 6th Int. Symposium on Frontiers of Combining Systems, (FroCos'07), Liverpool, UK*, volume 4720 of *LNAI*, pages 237–251, September 2007.

[34] H. Rueß and N. Shankar. Deconstructing Shostak. In *Proc. of the 16th Annual IEEE Symposium on Logic in Computer Science, (LICS'01), Boston, Massachusetts, USA*, pages 19–28. IEEE Computer Society, June 2001.

[35] H. Rueß and N. Shankar. Combining shostak theories. In S. Tison, editor, *Proc. of the 13th International Conference on Rewriting Techniques and Applications (Copenhagen, Denmark)*, volume 2378 of *LNCS*, pages 1–18. Springer, July 2002.

[36] Roberto Sebastiani. Lazy Satisfiability Modulo Theories. *Journal on Satisfiability, Boolean Modeling and Computation (JSAT)*, 3:141–224, 2007.

[37] R. E. Shostak. Deciding combinations of theories. *Journal of the ACM*, 31:1–12, 1984.

[38] A. Stump and L.-Y. Tang. The Algebra of Equality Proofs. In *Proc. of the 16th International Conference on Rewriting Techniques and Applications (RTA)*, volume 3467 of *LNCS*, pages 469–483. Springer, 2005.

[39] R. E. Tarjan. Efficiency of a good but not linear set union algorithm. *Journal of the ACM*, 22(2):215–225, April 1975.

[40] C. Tinelli and C. Ringeissen. Unions of non-disjoint theories and combinations of satisfiability procedures. *Theoretical Computer Science*, 290(1):291–353, January 2003.

Centre de recherche INRIA Nancy – Grand Est
LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399