Cooperation of Decision Procedures for the Satisfiability Problem

Christophe Ringeissen

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Cooperation of Decision Procedures for the Satisfiability Problem

Christophe Ringeissen

N° 2753

Décembre 1995

PROGRAMME 2

ISSN 0249-6399

Cooperation of Decision Procedures for the Satisfiability Problem

Christophe Ringeissen*

Programme 2 — Calcul symbolique, programmation et génie logiciel
Projet Protheo

Rapport de recherche n°2753 — Décembre 1995 — 19 pages

Abstract: Constraint programming is strongly based on the use of solvers which are able to check satisfiability of constraints. We show in this paper a rule-based algorithm for solving in a modular way the satisfiability problem w.r.t. a class of theories Th. The case where Th is the union of two disjoint theories Th_1 and Th_2 is known for a long time but we study here different cases where function symbols are shared by Th_1 and Th_2. The chosen approach leads to a highly non-deterministic decomposition algorithm but drastically simplifies the understanding of the combination problem. The obtained decomposition algorithm is illustrated by the combination of non-disjoint equational theories.

Key-words: constraint programming, decision procedure, satisfiability, combination problem

(Résumé : tsvp)

*INRIA-Lorraine & CRIN, e-mail: Christophe.Ringeissen@loria.fr

Unité de recherche INRIA Lorraine
Technôpole de Nancy-Brabois, Campus scientifique,
615 rue de Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY (France)
Téléphone : (33) 83 59 30 30 – Télécopie : (33) 83 27 83 19
Antenne de Metz, technopôle de Metz 2000, 4 rue Marconi, 55070 METZ
Téléphone : (33) 87 20 35 00 – Télécopie : (33) 87 76 39 77

Coopération de procédures de décision pour le problème de satisfaisabilité

Résumé : La programmation avec contraintes est basée sur l'utilisation de solveurs capables de vérifier la satisfaisabilité des contraintes. Nous présentons un algorithme à base de règles pour résoudre de façon modulaire le problème de satisfaisabilité par rapport à une classe de théories Th. Le cas où Th est une union de deux théories disjointes Th_1 et Th_2 est connue depuis longtemps mais nous étudions ici des cas où les symboles de fonctions sont partagés par Th_1 et Th_2. L'approche choisie conduit à un algorithme de décomposition fortement non-déterministe qui simplifie pourtant la compréhension du problème de combinaison. L'algorithme de décomposition est illustré par le mélange de théories équationnelles non disjointes.

Mots-clé : programmation avec contraintes, procédure de décision, satisfaisabilité, problème de combinaison

1 Introduction

In recent years, the problem of combining decision procedures became of greatest interest in many fields of computer science, especially for constraint programming and automated deduction. The modular construction of decision procedures has been considered for the first time by Shostak [17, 18] in order to solve heterogeneous formulae involving arithmetic and additional function symbols. Approximately in the same time, Nelson & Oppen [12, 13, 11] proposed an algorithm dedicated to the union of theories axiomatizing reals, arrays, lists and additional function symbols. The aim was to build a validity checker for programming languages in which such formulae frequently appear.

Formally, the *combined* decision problem can be stated as follows: Given two first-order theories Th_1 and Th_2 built respectively on the signatures Σ_1 and Σ_2, how is it possible to build a decision algorithm for the $\Sigma_1 \cup \Sigma_2$-theory $Th_1 \cup Th_2$ thanks to the decision algorithms provided for the Σ_1-theory Th_1 and the Σ_2-theory Th_2?

Until now, the main assumption is the disjointness of signatures Σ_1 and Σ_2. A further assumption needed in the framework of Nelson & Oppen is that single theories must be universal and *stably infinite* which means that a formula φ is satisfiable if and only if there exists a model of the theory with an infinite domain satisfying φ. So, even for the disjoint case, abstract assumptions on theories are needed in order to obtain a completeness result of the decomposition algorithm.

Another modularity problem has been thoroughly studied during the last decade for unification [19, 21, 8, 16, 6, 3] in the union of two equational theories E_1 and E_2, that is for solving equations in the term algebra $\mathcal{T}(\Sigma_1 \cup \Sigma_2, \mathcal{X})/=_{E_1 \cup E_2}$. Then, the combination techniques developed in this context have been extended in two directions: first, to allow other constraints [4, 15, 10] in $\mathcal{T}(\Sigma_1 \cup \Sigma_2, \mathcal{X})/=_{E_1 \cup E_2}$ and second, to permit shared function symbols [14, 7].

In this paper, we present combination techniques for solving the original satisfiability problem in a non-deterministic manner (the simplest one) and in the non-disjoint case. Section 2 recalls the basic definitions of the notions used in the following. In Section 3, we present the rule-based decomposition algorithm which is correct and complete for all unions of theories considered along the paper. Section 4 gives the general construction of a model for the union of theories $Th_1 \cup Th_2$. Section 5 introduces sufficient assumptions on disjoint or non-disjoint theories to be combined. We investigate the case where shared function symbols are interpreted in the trivial way (i.e. syntactically) in both models of the theories. Section 6 presents some applications related to equational theories. In Section 7, we conclude with final remarks and future works.

2 Definitions

We first briefly introduce some basic notations. Let $\Sigma = (\mathcal{F}, \mathcal{P})$ be a mono-sorted first-order signature where $\mathcal{F}$ is a finite set of function symbols and $\mathcal{P}$ is a finite set of predicate symbols. The set $\mathcal{P}$ does not contain $=$ which is always interpreted as the identity relation.

The subset of function symbols in $\mathcal{F}$ (resp. predicate symbols in $\mathcal{P}$) of arity m is denoted by $\mathcal{F}_m$ (resp. $\mathcal{P}_m$). The arity of a function symbol f (resp. a predicate symbol p) is denoted by $ar(f)$ (resp. $ar(p)$). The set of Σ-terms over a set A and of height n is defined recursively as follows:

1. $\mathcal{T}_0(\Sigma, A) = A$,
2. $\mathcal{T}_n(\Sigma, A) = \{f(\vec{a}) \mid f \in \mathcal{F}_m, \vec{a} \in (\mathcal{T}_{n-1}(\Sigma, A))^m\} \cup \mathcal{T}_{n-1}(\Sigma, A)$.

The set of Σ-terms over A is $\mathcal{T}(\Sigma, A) = \cup_{n \geq 0} \mathcal{T}_n(\Sigma, A)$. An equivalence relation $\equiv$ on A can be extended as usual on A^m:

$$(a_1, \ldots, a_m) \equiv (a'_1, \ldots, a'_m) \text{ if } \forall k \in [1, m],\ a_k \equiv a'_k$$

and defines a congruence relation $\equiv$ on $\mathcal{T}(\Sigma, A)$ as follows: $f(\vec{a_1}) \equiv f(\vec{a_2})$ if $\vec{a_1} \equiv \vec{a_2}$ and $f \in \mathcal{F}$. Note that we use vectors to denote tuples. The equivalence class of $a \in A$ w.r.t. $\equiv$ is denoted by $[a]_{\equiv}$.

Let us consider the set of Σ-terms over $\mathcal{X}$ where $\mathcal{X}$ is an infinite denumerable set of variables. The terms $t_{|\omega}$ and $t[\omega \hookleftarrow s]$ denote respectively the subterm of t at the position ω and the replacement in t of $t_{|\omega}$ by s. The symbol of t occurring at the position ω (resp. the top symbol of t) is written $t(\omega)$ (resp. $t(\epsilon)$). The set of variables occurring in a term t is denoted by $\mathcal{V}(t)$. Let $\mathcal{M}$ be a Σ-algebra with A as domain. An assignment α is a mapping from $\mathcal{X}$ to A; it uniquely extends to an homomorphism $\underline{\alpha}$ from $\mathcal{T}(\Sigma, \mathcal{X})$ to $\mathcal{M}$. The restriction of α to a set of variables V is denoted by $\alpha_{|V}$. The range of α is denoted by $Ran(\alpha)$. A $\mathcal{M}$-solution of a quantifier-free Σ-formula φ is an assignment α such that $\underline{\alpha}(\varphi)$ holds in $\mathcal{M}$. The formula φ is valid in $\mathcal{M}$, denoted by $\mathcal{M} \models \varphi$, if any assignment α is a $\mathcal{M}$-solution of φ.

A *substitution* is an assignment from $\mathcal{X}$ to $\mathcal{T}(\Sigma, \mathcal{X})$ with only finitely many variables not mapped to themselves. A substitution uniquely extends to an endomorphism of $\mathcal{T}(\Sigma, \mathcal{X})$. We use letters $\sigma, \mu, \gamma, \phi, \ldots$ to denote substitutions and do not distinguish σ and $\underline{\sigma}$. Application of substitutions is written out by postfixed juxtaposition. We call *domain* of the substitution σ the (finite) set of variables $\mathcal{D}om(\sigma) = \{x | x \in \mathcal{X} \text{ and } x\sigma \neq x\}$, *range* of σ the set of terms $\mathcal{R}an(\sigma) = \cup_{x \in \mathcal{D}om(\sigma)}\{x\sigma\}$ and *variable range* of σ the set of variables $\mathcal{VR}an(\sigma) = \cup_{x \in \mathcal{D}om(\sigma)}\mathcal{V}(x\sigma)$. A substitution σ is *idempotent* if $\sigma = \sigma\sigma$.

Definition 1 A Σ-theory (resp. universal Σ-theory) is a (possibly infinite) set of first-order Σ-sentences (resp. universally quantified first-order sentences), where sentences are formulaes without free variables. The Σ-theory (resp. universal Σ-theory) of a Σ-structure $\mathcal{M}$ is denoted by $\mathcal{TH}(\mathcal{M})$ (resp. $Th(\mathcal{M})$) and is defined as the set of first-order Σ-sentences (resp. universally quantified first-order Σ-sentences) φ such that $\mathcal{M} \models \varphi$. A Σ-structure $\mathcal{M}$ is a model of a Σ-theory Th, denoted by $\mathcal{M} \models Th$, if $\mathcal{M} \models \varphi$ for any $\varphi \in Th$. A conjunction of atomic Σ-*literals* (i.e. an atomic formula, an equation or its negation) with some universally quantified variables is simply called here a Σ-*formula*. A disjunction of Σ-formulae must be viewed as usual as a sequence of Σ-formulae to consider separately. The set of free variables occurring in a formula φ is denoted by $\mathcal{V}(\varphi)$. A formula φ is *satisfiable* w.r.t. Th if there

exists a model $\mathcal{M}$ of Th and a $\mathcal{M}$-solution of φ (or equivalently $\mathcal{M} \models \exists \mathcal{V}(\varphi) : \varphi$). The *satisfiability value* [1] of a formula φ w.r.t. Th is denoted by $(\varphi)_{Th}$ and is equal to $\top$ (true) if φ is satisfiable w.r.t. Th, else $(\varphi)_{Th} = \bot$ (false).

Let Σ' be a signature such that $\Sigma' \subseteq \Sigma$. If $\mathcal{M}$ is a Σ-structure, then $\mathcal{M}^{\Sigma'}$ denotes the Σ'-structure with the same domain and where function and predicate symbols are interpreted as in $\mathcal{M}$. A position ω of a Σ-term t is an alien position w.r.t. Σ' if $t(\omega)$ is not in Σ' and if for any other position ω' on the path from the root to ω, $t(\omega')$ is in Σ'. The set of alien positions of t w.r.t. Σ' is denoted by $AlienPos_{\Sigma'}(t)$.

3 Decomposition algorithm

We present informally in this section a decomposition algorithm for solving the satisfiability problem w.r.t. a $\Sigma_1 \cup \Sigma_2$-theory $Th_1 \cup Th_2$ where Th_1 is a Σ_1-theory, Th_2 a Σ_2-theory and $\Sigma_1 \cap \Sigma_2$ a set of function symbols $\mathcal{SF}$ (there is no shared predicate). This algorithm transforms any heterogeneous $\Sigma_1 \cup \Sigma_2$-formula into either $\bot$ or $\top$ thanks to decision procedures dedicated respectively to pure Σ_1-formulae and pure Σ_2-formulae. The reader is assumed familiar with algorithms based on transformation rules [9]. Any rule given in this paper transforms a formula φ into φ' in such a way that φ is satisfiable if and only if φ' is satisfiable. So, transformation rules preserve here satisfiability (but not necessarily the set of solutions). Note also that the differents steps are executed sequentially and each of these finitely many steps terminates obviously. Consequently, the whole algorithm terminates and this is one of the greatest advantage of the non-deterministic approach. The normal form of φ with respect to the transformation rules is either $\top$ if φ is satisfiable or $\bot$ if φ is unsatisfiable.

The first step of the decomposition algorithm consists in splitting the input heterogeneous formula into pure equations and formulae. Hence, a heterogeneous $\Sigma_1 \cup \Sigma_2$-formula is transformed into a conjunction $\varphi_1 \wedge \varphi_2$ where φ_i is a Σ_i-formula for $i = 1, 2$. Note that formulae built on the set of shared function symbols $\mathcal{SF} = \mathcal{F}_1 \cap \mathcal{F}_2$ are considered both in φ_1 and in φ_2

In the second and third steps, the same shared equations and disequations are added to each pure formula:

- In Step 2, add equations and disequations of the form $x = t$ and $y \neq t$ where t is a $\mathcal{SF}$-term (over $\mathcal{X}$) of height n such that each variable of t occurs only once and does not appear elsewhere in the current formula. The integer n is somehow a parameter of the decomposition algorithm. We will see later in Section 5 on what relies the choice of n.
- In Step 3, add equations and disequations between variables of the current formula like $x = y$ and $y \neq z$.

In the last step, the solvers related to each theory are called in order to replace each pure formula by true ($\top$) or false ($\bot$).

[1] We will use the boolean operators $\vee, \wedge$ for manipulating satisfiability values.

1. Purification

   Atom

   $$\frac{\varphi \wedge p(t_1, \ldots, t, \ldots, t_n)}{\varphi \wedge p(t_1, \ldots, t[\omega \hookleftarrow x], \ldots, t_n) \wedge x = t_{|\omega}}$$
   $$\text{if } \begin{cases} p \in \mathcal{P}_i, \\ \omega \in AlienPos_{\Sigma_i}(t), \\ x \notin \mathcal{V}(\varphi \wedge p(t_1, \ldots, t, \ldots, t_n)). \end{cases}$$

   Equation

   $$\frac{\varphi \wedge s = t}{\varphi \wedge s[\omega \hookleftarrow x] = t \wedge x = s_{|\omega}} \quad \text{if } \begin{cases} s(\epsilon) \in \Sigma_i, \\ \omega \in AlienPos_{\Sigma_i}(s), \\ x \notin \mathcal{V}(\varphi \wedge s = t). \end{cases}$$

   Disequation

   $$\frac{\varphi \wedge s \neq t}{\varphi \wedge x = s \wedge y = t \wedge x \neq y} \quad \text{if } \begin{cases} s \in \mathcal{T}(\Sigma, \mathcal{X}) \backslash \mathcal{X}, \\ x, y \notin \mathcal{V}(\varphi \wedge s \neq t). \end{cases}$$

   Conflict

   $$\frac{\varphi \wedge t_1 = t_2}{\varphi \wedge x = t_1 \wedge x = t_2} \quad \text{if } \begin{cases} t_i \in \mathcal{T}(\Sigma_i, \mathcal{X}) \backslash \mathcal{T}(\mathcal{SF}, \mathcal{X}) \text{ for } i = 1, 2, \\ x \notin \mathcal{V}(\varphi \wedge t_1 = t_2). \end{cases}$$

2. Instantiation[a]

   $$\frac{\varphi_1 \wedge \varphi_2}{\bigvee_{\rho \in SUBS^{\mathcal{SF}}_{\mathcal{V}(\varphi_1 \wedge \varphi_2), n}} (\varphi_1 \rho \wedge \rho_{\neq}) \wedge (\varphi_2 \rho \wedge \rho_{\neq})}$$

3. Identification[b]

   $$\frac{\varphi_1 \wedge \varphi_2}{\bigvee_{\xi \in ID_{\mathcal{V}(\varphi_1 \wedge \varphi_2)}} (\varphi_1 \xi \wedge \xi_{\neq}) \wedge (\varphi_2 \xi \wedge \xi_{\neq})}$$

4. Decision

   Satisfiability

   $\frac{(\varphi_i)}{\top}$ if φ_i is satisfiable w.r.t. Th_i.

   Unsatisfiability

   $\frac{(\varphi_i)}{\bot}$ if φ_i is unsatisfiable w.r.t. Th_i.

Figure 1: Decomposition algorithm

[a] This step is superfluous for disjoint theories. See Subsection 5.2 for the definition of $SUBS$.

[b] See Subsection 5.1 for the definition of ID.

The decomposition algorithm is described in Figure 1 and looks like one of the most simple we could imagine. It contains for instance only one non-deterministic step if $\mathcal{SF}$ is empty. In the following, we are interested in the assumptions needed on the input theories and formulae for proving the soundness of the decomposition algorithm. We first establish abstract properties which are proved sufficient and necessary.

4 Construction of a combined model

We show how to construct a $\Sigma_1 \cup \Sigma_2$-structure by combining a Σ_1-structure $\mathcal{M}_1$ and a Σ_2-structure $\mathcal{M}_2$. This construction yields a model of a $\Sigma_1 \cup \Sigma_2$-theory $Th_1 \cup Th_2$ when $\mathcal{M}_1$ and $\mathcal{M}_2$ are respectively models of a Σ_1-theory Th_1 and a Σ_2-theory Th_2. We assume that Σ_1 and Σ_2 share only a set of function symbols denoted by $\mathcal{SF}$. In the rest of the paper, φ_i denotes a Σ_i-formula for $i = 1, 2$.

Definition 2 Let $\mathcal{M}_1$ be a Σ_1-structure and $\mathcal{M}_2$ be a Σ_2-structure such that there exists a one-to-one mapping o from the domain A_1 of $\mathcal{M}_1$ to the domain A_2 of $\mathcal{M}_2$ verifying

$$\forall f \in \mathcal{SF}_m,\ \forall \vec{a_1} \in A_1^m,\ \forall \vec{a_2} \in A_2^m,\ \vec{a_1} =_o \vec{a_2} \Rightarrow f_{\mathcal{M}_1}(\vec{a_1}) =_o f_{\mathcal{M}_2}(\vec{a_2})$$

where $=_o$ denotes the equivalence relation on $A_1 \cup A_2$ such that: $a =_o b$ if $a = b$ or $a = o(b)$ or $b = o(a)$. The *combined structure* of $\mathcal{M}_1$ and $\mathcal{M}_2$ is the $\Sigma_1 \cup \Sigma_2$-structure $\mathcal{M}_1 \circledast_o \mathcal{M}_2$ defined as follows:

- its domain is $(A_1 \cup A_2)/ =_o$,
- $\forall g_i \in (\mathcal{F}_i)_m,\ \forall \vec{a_i} \in A_i^m,\ g_{i,\mathcal{M}_1 \circledast_o \mathcal{M}_2}([\vec{a_i}]_{=_o}) = [g_{i,\mathcal{M}_i}(\vec{a_i})]_{=_o}$,
- $\forall p_i \in (\mathcal{P}_i)_m,\ \forall \vec{a_i} \in A_i^m,\ p_{i,\mathcal{M}_1 \circledast_o \mathcal{M}_2}([\vec{a_i}]_{=_o})$ iff $p_{i,\mathcal{M}_i}(\vec{a_i})$.

According to the previous definition, the combined structure $\mathcal{M}_1 \circledast_o \mathcal{M}_2$ exists only if the equivalence relation $=_o$ is preserved after respective application of shared functions. Under this existence assumption, we can choose the combined structure as a model of $Th_1 \cup Th_2$.

Proposition 1 If $\mathcal{M}_1 \circledast_o \mathcal{M}_2$ exists and $\mathcal{M}_1$ is a Σ_1-model of Th_1 and $\mathcal{M}_2$ is a Σ_2-model of Th_2, then $\mathcal{M}_1 \circledast_o \mathcal{M}_2$ is a $\Sigma_1 \cup \Sigma_2$-model of $Th_1 \cup Th_2$.

This combined structure gives us implicitely a satisfiability criterion for the conjunction of a Σ_1-formula and a Σ_2-formula.

Proposition 2 A formula $\varphi_1 \wedge \varphi_2$ is satisfiable w.r.t. $Th_1 \cup Th_2$ if and only if there exist

1. a Σ_i-model $\mathcal{M}_i$ of Th_i and a $\mathcal{M}_i$-solution α_i of φ_i for $i = 1, 2$.
2. a one-to-one mapping o from the domain A_1 of $\mathcal{M}_1$ to the domain A_2 of $\mathcal{M}_2$ such that

$$\forall f \in \mathcal{SF}_m,\ \forall \vec{a_1} \in A_1^m,\ \forall \vec{a_2} \in A_2^m,\ \vec{a_1} =_o \vec{a_2} \Rightarrow f_{\mathcal{M}_1}(\vec{a_1}) =_o f_{\mathcal{M}_2}(\vec{a_2})$$

and $\forall x \in \mathcal{V}(\varphi_1 \wedge \varphi_2),\ \alpha_1(x) =_o \alpha_2(x)$.

Proof: ($\Leftarrow$) The assignment $\alpha : \mathcal{X} \to (A_1 \cup A_2)/=_o$ defined as follows:

$$\forall x \in \mathcal{V}(\varphi_1 \wedge \varphi_2),\ \alpha(x) = [\alpha_1(x)]_{=_o}$$

is obviously a $\mathcal{M}_1 \circledast_o \mathcal{M}_2$-solution of $\varphi_1 \wedge \varphi_2$. We could also choose α_2 instead of α_1 in the definition of α since $[\alpha_1(x)]_{=_o} = [\alpha_2(x)]_{=_o}$.

($\Rightarrow$) If $\mathcal{M}$ is a $\Sigma_1 \cup \Sigma_2$-model of $Th_1 \cup Th_2$, then $\mathcal{M}_i = \mathcal{M}^{\Sigma_i}$ is a Σ_i-model of Th_i. We can obviously choose the identity for o and the $\mathcal{M}$-solution α for the $\mathcal{M}_i$-solution α_i. □

This criterion is not directly usable. We will propose in the next section more practical criteria but restricting also the class of considered theories. However, this proposition can be obviously applied to the case where φ_1 and φ_2 are simply the true formula.

Corollary 1 The $\Sigma_1 \cup \Sigma_2$-theory $Th_1 \cup Th_2$ is consistent if and only if there exist a Σ_i-model $\mathcal{M}_i$ of Th_i for $i = 1, 2$ such that $\mathcal{M}_1^{\Sigma_1 \cap \Sigma_2}$ and $\mathcal{M}_2^{\Sigma_1 \cap \Sigma_2}$ are isomorphic.

Proposition 2 is also obviously applicable when Th_1 and Th_2 are identical modulo a renaming of signatures and if φ_1 and φ_2 are identical modulo the same renaming. Then, we are able to solve the satisfiability problem w.r.t. $Th_1 \cup Th_2$ in a modular way and even if Th_1 and Th_2 share some function symbols.

Example 1 Consider $E_i = \{x +_i (-x) = 0\}$ for $i = 1, 2$ and the formula $\varphi = (x +_1 (-x) = x +_2 (-x))$ which is equivalent to $(\varphi_1 = (x +_1 (-x) = y)) \wedge (\varphi_2 = (x +_2 (-x) = y))$. The formula φ is satisfiable w.r.t. $E_1 \cup E_2$ due to the solution $\alpha = \alpha_1 = \alpha_2 = \{x \mapsto 0, y \mapsto 0\}$ of φ_1 and φ_2.

In the example above, one could remark that $(\varphi_1 \wedge \varphi_2)_{Th_1 \cup Th_2} = (\varphi_i)_{Th_i}$ and so there is no need here to combine decision procedures. The reader should note however that the problem of solving a conjunction of two renamed formulae is no more obvious in the context of unification [7].

5 Assumptions on theories

We study now three cases of disjoint and non-disjoint unions of theories for which we give sufficient assumptions for satisfying the second point of Proposition 2.

5.1 Stably infinite disjoint theories

When the signatures of theories are disjoint, then it is sufficient to assume that each theory admits, for each satisfiable quantifier-free formula, a model having an infinite domain and which satisfies the formula. If this assumption holds, then it is easy to prove that there exists also a model with an infinite *denumerable* domain. Moreover, there exists obviously a one-to-one mapping between two infinite denumerable sets.

Definition 3 [11] A Σ-structure is *infinite* if its domain is infinite. A universal Σ-theory Th is *stably infinite* if for any quantifier-free Σ-formula φ satisfiable w.r.t. Th, there exists an infinite Σ-structure $\mathcal{M}$ verifying $\mathcal{M} \models Th$ and $\mathcal{M} \models \exists \mathcal{V}(\varphi) : \varphi$.

The following proposition provides interesting examples of stably infinite theories.

Proposition 3 If $\mathcal{M}$ is an infinite structure, then $Th(\mathcal{M})$ is stably infinite and any quantifier-free formula is satisfiable w.r.t. $Th(\mathcal{M})$ if and only if it is satisfiable in $\mathcal{M}$.

Proof: If a quantifier-free formula φ is satisfiable in $\mathcal{M}$, then φ is satisfiable w.r.t. $Th(\mathcal{M})$ since $\mathcal{M}$ is obviously a model of $Th(\mathcal{M})$. Conversely, if φ is unsatisfiable in $\mathcal{M}$, then $\mathcal{M} \models \neg\varphi$ and so $Th(\mathcal{M}) \models \neg\varphi$. Therefore, φ is unsatisfiable w.r.t. $Th(\mathcal{M})$. □

Lemma 1 A quantifier-free formula satisfiable w.r.t a stably infinite theory Th is satisfiable in a model of Th with an infinite *denumerable* domain.

Proposition 4 If $\mathcal{M}_1$ and $\mathcal{M}_2$ are infinite structures with denumerable domains built on disjoint signatures, then there is a one-to-one mapping o such that $\mathcal{M}_1 \circledast_o \mathcal{M}_2$ exists and is an infinite structure with a denumerable domain.

It remains to verify the assumption which states that solutions α_1 and α_2 of respectively φ_1 and φ_2 are equivalent modulo $=_o$. Hence, two variables must have the same solution (or must have different solutions) simultaneously in both models. This explains why we need to consider all possible identifications of variables occurring in $\varphi_1 \wedge \varphi_2$. Formally, the set of identifications of variables in V is $ID_V = \{\xi \mid \mathcal{D}om(\xi) \subseteq V,\ \mathcal{R}an(\xi) \subseteq V, \xi \text{ is idempotent}\}$. Given an identification $\xi \in ID_V$, $\xi_{\neq}$ denotes the equational formula

$$\bigwedge_{x,y \in V \setminus \mathcal{D}om(\xi)} x \neq y.$$

Proposition 5 Let Th_1 and Th_2 be two disjoint stably infinite theories. Then

$$(\varphi_1 \wedge \varphi_2)_{Th_1 \cup Th_2} = \bigvee_{\xi \in ID_{\mathcal{V}(\varphi_1 \wedge \varphi_2)}} (\varphi_1 \xi \wedge \xi_{\neq})_{Th_1} \wedge (\varphi_2 \xi \wedge \xi_{\neq})_{Th_2}$$

where φ_i is a Σ_i-formula for $i = 1, 2$.

Proof: ($\Rightarrow$) If $(\varphi_1 \wedge \varphi_2)_{Th_1 \cup Th_2}$ is true, then there exists a $\mathcal{M}$-solution α of $\varphi_1 \wedge \varphi_2$, where $\mathcal{M}$ is a model $Th_1 \cup Th_2$. Thus, the disjunct corresponding to an identification ξ such that $\forall x, y \in \mathcal{V}(\varphi_1 \wedge \varphi_2),\ x\xi = y\xi \Leftrightarrow \alpha(x) = \alpha(y)$ is true.

($\Leftarrow$) If one disjunct is true, then there exists a $\mathcal{M}_i$-solution α_i of φ_i for $i = 1, 2$ such that $\forall x, y \in \mathcal{V}(\varphi_1 \wedge \varphi_2),\ \alpha_1(x) = \alpha_1(y) \Leftrightarrow \alpha_2(x) = \alpha_2(y)$. So, we are able to choose the one-to-one mapping o such that

$$\forall x \in \mathcal{V}(\varphi_1 \wedge \varphi_2),\ \alpha_2(x) = o(\alpha_1(x)).$$

Then, the assignment α from $\mathcal{X}$ to $(A_1 \cup A_2)/=_o$ defined by:

$$\forall x \in \mathcal{V}(\varphi_1 \wedge \varphi_2),\ \alpha(x) = [\alpha_1(x)]_{=_o}$$

is a $\mathcal{M}_1 \circledast_o \mathcal{M}_2$-solution of $\varphi_1 \wedge \varphi_2$. □

Corollary 2 If Th_1 and Th_2 are two disjoint stably infinite theories, then $Th_1 \cup Th_2$ is stably infinite.

Corollary 3 The decomposition algorithm described in Figure 1 (without Step 2) solves the satisfiability problem of quantifier-free formulae w.r.t. the union of two disjoint stably infinite theories.

Theorem 1 *Satisfiability of quantifier-free $\Sigma_1 \cup \Sigma_2$-formulae is decidable w.r.t. $Th_1 \cup Th_2$ if*

- $\Sigma_1 \cap \Sigma_2 = \emptyset$,
- *satisfiability of quantifier-free Σ_i-formulae is decidable w.r.t. Th_i,*
- *Th_i is a stably infinite theory (for $i = 1, 2$).*

We thus get the result due to Nelson & Oppen [13, 11] with a different proof. It is important to note that even for the disjoint case, we have (like them) additional assumptions concerning the individual theories. This result holds however for some non-universal theories since we can also consider arbitrary theories Th_i, provided each quantifier-free Σ_i-formula is satisfiable w.r.t. Th_i iff it is satisfiable in a model of Th_i with an infinite *denumerable* domain. In the following, we prefer to use this new assumption since more theories are taken into account.

5.2 Theories stably generated by shared terms

When the signatures of theories share function symbols, we have now to ensure that the application of the corresponding interpretations of function symbols preserves the one-to-one mapping between the domains of models of Th_1 and Th_2. The idea followed here is simply to assume that shared functions are interpreted syntactically in both models.

Definition 4 A Σ-structure $\mathcal{M}$ is *generated by $\mathcal{SF}$-terms* (over A) if

- its domain is $\mathcal{T}(\mathcal{SF}, A)$ where A is an infinite denumerable set,
- $\forall f \in \mathcal{SF}_m, \forall \vec{a} \in \mathcal{T}(\mathcal{SF}, A)^m,\ f_{\mathcal{M}}(\vec{a}) = f(\vec{a})$.

Given a set $V \subseteq \mathcal{X}$, the conjunction of disequations

$$\bigwedge_{x \in V, f \in \mathcal{SF}} \forall x_{f,1}, \ldots, x_{f,ar(f)} : x \neq f(x_{f,1}, \ldots, x_{f,ar(f)})$$

is denoted by $basic_V$. A formula ϕ of the form $\varphi \wedge basic_V$, where φ is quantifier-free and $V \subseteq \mathcal{V}(\varphi)$, is said *partly basic*. The set of basic variables in ϕ is $\mathcal{BV}(\phi) = V$. A partly basic formula ϕ is *basic* if $\mathcal{BV}(\phi) = \mathcal{V}(\phi)$. A Σ-theory Th is *stably generated by $\mathcal{SF}$-terms* if for any basic Σ-formula φ satisfiable w.r.t. Th, there exists a Σ-structure $\mathcal{M}$ generated by $\mathcal{SF}$-terms verifying $\mathcal{M} \models Th$ and $\mathcal{M} \models \exists \mathcal{V}(\varphi) : \varphi$.

If $\mathcal{M}$ is a structure generated by $\mathcal{SF}$-terms (over A), then an assignment $\alpha : \mathcal{X} \rightarrow \mathcal{T}(\mathcal{SF}, A)$ is a $\mathcal{M}$-solution of $basic_V$ if and only if $Ran(\alpha_{|V}) \subseteq A$. Quantifier-free formulae are partly basic formulae φ with $\mathcal{BV}(\varphi) = \emptyset$. Note also that a stably infinite theory is stably generated by $\mathcal{SF}$-terms if $\mathcal{SF}$ is empty.

Similarly to Proposition 3, we can use a unique structure generated by $\mathcal{SF}$-terms to built a theory stably generated by $\mathcal{SF}$-terms.

Proposition 6 If $\mathcal{M}$ is a structure generated by $\mathcal{SF}$-terms, then $\mathcal{TH}(\mathcal{M})$ is stably generated by $\mathcal{SF}$-terms, and any basic formula is satisfiable w.r.t. $\mathcal{TH}(\mathcal{M})$ if and only if it is satisfiable in $\mathcal{M}$.

Proof: The key point is to prove that a basic formula which is unsatisfiable in $\mathcal{M}$ is also unsatisfiable w.r.t. $\mathcal{TH}(\mathcal{M})$. Let $\phi = \varphi \wedge basic_{\mathcal{V}(\varphi)}$ be a basic formula unsatisfiable in $\mathcal{M}$. The first-order sentence $\bar{\phi}$ defined as follows:

$$\forall \mathcal{V}(\varphi) : (\varphi \Rightarrow \bigvee_{x \in \mathcal{V}(\varphi), f \in \mathcal{SF}} \exists x_{f,1}, \ldots, x_{f,ar(f)} : x = f(x_{f,1}, \ldots, x_{f,ar(f)}))$$

is the negation of ϕ and so $\mathcal{M} \models \bar{\phi}$. Then, $\mathcal{TH}(\mathcal{M}) \models \bar{\phi}$ and ϕ is unsatisfiable w.r.t. $\mathcal{TH}(\mathcal{M})$. □

Proposition 7 If $\mathcal{M}_1$ and $\mathcal{M}_2$ are structures generated by $\mathcal{SF}$-terms, then there exists a one-to-one mapping o such that $\mathcal{M}_1 \circledast_o \mathcal{M}_2$ is a structure generated by $\mathcal{SF}$-terms.

Proof: By assumption, there exists a one-to-one mapping o from A_1 to A_2 which can be used to define a congruence relation on $\mathcal{T}(\mathcal{SF}, A_1 \cup A_2)$:

$$\forall f \in \mathcal{SF},\ f(\vec{a_1}) =_o f(\vec{a_2}) \text{ if } \vec{a_1} =_o \vec{a_2}.$$

The $\Sigma_1 \cup \Sigma_2$-structure $\mathcal{M}_1 \circledast_o \mathcal{M}_2$ is as follows:

- its domain is $\mathcal{T}(\mathcal{SF}, (A_1 \cup A_2)/ =_o)$
- $\forall f \in \mathcal{SF}_m,\ \forall \vec{a} \in (\mathcal{T}(\mathcal{SF}, (A_1 \cup A_2)/ =_o))^m,\ f_{\mathcal{M}_1 \circledast_o \mathcal{M}_2}(\vec{a}) = f(\vec{a})$

Therefore, $\mathcal{M}_1 \circledast_o \mathcal{M}_2$ is still a structure generated by $\mathcal{SF}$-terms. □

We are now faced with the fact that respective pure solutions must be equivalent modulo $=_o$ in order to construct a combined solution in $\mathcal{M}_1 \circledast_o \mathcal{M}_2$. An idea for dealing with this construction is to apply simultaneously in each theory the same kind of shared instance. Hence, partly basic formulae are transformed into basic ones.

Definition 5 A formula φ is n-satisfiable in a Σ-structure $\mathcal{M}$ generated by $\mathcal{SF}$-terms if there exists a $\mathcal{M}$-solution of φ such that its range is included in $\mathcal{T}_n(\mathcal{SF}, A)$. A formula φ is n-satisfiable w.r.t. a theory Th if φ is n-satisfiable in a model of Th generated by $\mathcal{SF}$-terms. If φ is n-satisfiable w.r.t. Th, then we define $(\varphi)_{Th,n} = \top$, else $(\varphi)_{Th,n} = \bot$.

This restricted form of satisfiability is interesting in practice since we are not looking for any solution but only for a "short" solution (it depends on the choice of the height n). However, our non-deterministic approach leads us to enumerate systematically all possible solutions of height less than n. This explains the use of a **finite** set $SUBS^{\mathcal{SF}}_{V,n}$ of substitutions such that their domains and ranges are respectively included in V and $\mathcal{T}_n(\mathcal{SF}, \mathcal{X})$. The variables which are not substituted or those introduced by a substitution $\rho \in SUBS^{\mathcal{SF}}_{V,n}$ must have basic values as solution (i.e. in A_i). This property can be encoded by the conjunction of disequations $\rho_{\neq}$ defined as follows: $\rho_{\neq} = basic_{(V \setminus \mathcal{D}om(\rho)) \cup \mathcal{VR}an(\rho)}$. We assume also that each variable in $\mathcal{VR}an(\rho)$ occurs once in $\mathcal{R}an(\rho)$ and does not appear elsewhere in the input formula.

Proposition 8 If Th_1 and Th_2 are two theories stably generated by $\mathcal{SF}$-terms, then

$$(\varphi_1 \wedge \varphi_2)_{Th_1 \cup Th_2, n} =$$

$$\bigvee_{\rho \in SUBS^{\mathcal{SF}}_{\mathcal{V}(\varphi_1 \wedge \varphi_2), n}} \bigvee_{\xi \in ID_{\mathcal{V}((\varphi_1 \wedge \varphi_2)\rho \wedge \rho_{\neq})}} ((\varphi_1 \rho \wedge \rho_{\neq})\xi \wedge \xi_{\neq})_{Th_1} \wedge ((\varphi_2 \rho \wedge \rho_{\neq})\xi \wedge \xi_{\neq})_{Th_2}$$

Proof: ($\Rightarrow$) If $(\varphi_1 \wedge \varphi_2)_{Th_1 \cup Th_2, n}$ is true, then $\varphi_1 \wedge \varphi_2$ is satisfiable in a model of $Th_1 \cup Th_2$ generated by $\mathcal{SF}$-terms (over A), say $\mathcal{M}$, and there exists a $\mathcal{M}$-solution α of $\varphi_1 \wedge \varphi_2$ such that $Ran(\alpha) \subseteq \mathcal{T}_n(\mathcal{SF}, A)$. The structure $\mathcal{M}^{\Sigma_i}$ is obviously a model of Th_i for $i = 1, 2$. We can always choose $\rho \in SUBS_{\mathcal{V}(\varphi_1 \wedge \varphi_2), n}$ such that there exists an assignment $\alpha_i : \mathcal{X} \to \mathcal{T}(\mathcal{SF}, A)$ verifying $\forall x \in \mathcal{V}(\varphi_1 \wedge \varphi_2)$, $\alpha(x) = \underline{\alpha_i}(x\rho)$ with $Ran(\alpha_i) \subseteq A$. Then, we define ξ as follows: $x\xi = y\xi$ iff $\alpha_i(x) = \alpha_i(y)$. The assignment α_i is a $\mathcal{M}^{\Sigma_i}$-solution α_i of $(\varphi_i \rho \wedge \rho_{\neq})\xi \wedge \xi_{\neq}$ for $i = 1, 2$ and so one disjunct is true.

($\Leftarrow$) Conversely, if some $\bigwedge_{i=1}^{2}((\varphi_i \rho \wedge \rho_{\neq})\xi \wedge \xi_{\neq})_{Th_i}$ is true for an instantiation ρ and an identification ξ, then there exists a model $\mathcal{M}_i$ of Th_i generated by $\mathcal{SF}$-terms (over A_i) for $i = 1, 2$. The combined structure $\mathcal{M}_1 \circledast_o \mathcal{M}_2$ is a model of $Th_1 \cup Th_2$ generated by $\mathcal{SF}$-terms (over $(A_1 \cup A_2)/=_o$). A $\mathcal{M}_1 \circledast_o \mathcal{M}_2$-solution of $\varphi_1 \wedge \varphi_2$ can be constructed thanks to the $\mathcal{M}_i$-solution of $(\varphi_i \rho \wedge \rho_{\neq})\xi \wedge \xi_{\neq}$ for $i = 1, 2$. First, due to the identification of variables, we can assume without loss of generality that $\forall y \in \mathcal{V}((\varphi_1 \wedge \varphi_2)\rho)$, $\alpha_1(y) =_o \alpha_2(y)$. Then, the assignment $\alpha : \mathcal{X} \to \mathcal{T}_n(\mathcal{SF}, (A_1 \cup A_2)/=_o)$ such that

$$\forall x \in \mathcal{V}(\varphi_1 \wedge \varphi_2),\ \alpha(x) = ([\underline{\alpha_1}(x\rho)]_{=_o})\ \ (\text{or } [\underline{\alpha_2}(x\rho)]_{=_o})$$

is a $\mathcal{M}_1 \circledast_o \mathcal{M}_2$-solution of $\varphi_1 \wedge \varphi_2$ and so $(\varphi_1 \wedge \varphi_2)_{Th_1 \cup Th_2, n}$ is true. □

We are now able to present a new algorithm which looks like the one developed for the disjoint case. The difference is that some shared formulae must be taken into account by both theories thanks to an additional non-deterministic step:

Instantiation

$$\frac{\varphi_1 \wedge \varphi_2}{\bigvee_{\rho \in SUBS^{\mathcal{SF}}_{\mathcal{V}(\varphi_1 \wedge \varphi_2),n}} (\varphi_1\rho \wedge \rho_{\neq}) \wedge (\varphi_2\rho \wedge \rho_{\neq})}$$

This new step must be performed after Purification and before Identification. The corresponding algorithm can be applied to quantifier-free $\Sigma_1 \cup \Sigma_2$-formulae but also more generally for partly basic $\Sigma_1 \cup \Sigma_2$-formulae.

Corollary 4 The decomposition algorithm described in Figure 1 solves the n-satisfiability problem of partly basic formulae w.r.t. the union of two theories stably generated by $\mathcal{SF}$-terms.

Proposition 9 n-satisfiability of partly basic $\Sigma_1 \cup \Sigma_2$-formulae is decidable w.r.t. $Th_1 \cup Th_2$ if

- $\Sigma_1 \cap \Sigma_2 = \mathcal{SF}$,
- satisfiability of basic Σ_i-formulae is decidable w.r.t. Th_i,
- Th_i is stably generated by $\mathcal{SF}$-terms (for $i = 1, 2$).

Proposition 8 and Proposition 9 are also applicable to basic $\Sigma_1 \cup \Sigma_2$-formulae. In the particular case of basic formulae, 0-satisfiability coincides with satisfiability.

Corollary 5 If Th_1 and Th_2 are stably generated by $\mathcal{SF}$-terms, then $Th_1 \cup Th_2$ is stably generated by $\mathcal{SF}$-terms.

Theorem 2 *Satisfiability of basic $\Sigma_1 \cup \Sigma_2$-formulae is decidable w.r.t. $Th_1 \cup Th_2$ if*

- $\Sigma_1 \cap \Sigma_2 = \mathcal{SF}$,
- *satisfiability of basic Σ_i-formulae is decidable w.r.t. Th_i,*
- *Th_i is stably generated by $\mathcal{SF}$-terms (for $i = 1, 2$).*

From the decomposition algorithm, we can easily derive a semi-decision procedure for the satisfiability problem in a model of $Th_1 \cup Th_2$ generated by $\mathcal{SF}$-terms. This procedure works as follows: the conjunctions of pure formulae are enumerated according to a breadth-first strategy from smallest shared instances to larger ones and are solved in parallel. If a quantifier-free formula φ is satisfiable, then there exists an integer n such that φ is n-satisfiable w.r.t. $Th_1 \cup Th_2$ and so there is after a finite time a conjunction of pure formulae for which each elementary solver terminates and returns $\top$. In this case, we can halt the process and return $\top$ as result.

Theorem 3 *Satisfiability of partly basic $\Sigma_1 \cup \Sigma_2$-formulae in a model of $Th_1 \cup Th_2$ generated by $\mathcal{SF}$-terms is semi-decidable if*

- $\Sigma_1 \cap \Sigma_2 = \mathcal{SF}$,
- *satisfiability of basic* Σ_i*-formulae is semi-decidable w.r.t.* Th_i,
- Th_i *is stably generated by* $\mathcal{SF}$*-terms (for* $i = 1, 2$*).*

5.3 Theories stably generated by bounded shared terms

In this section, we study how the previous semi-decision procedure can be turned into a decision algorithm for the satisfiability problem w.r.t. $Th_1 \cup Th_2$. This is possible if the definition of a structure generated by $\mathcal{SF}$-terms can be restricted to a bounded set of shared terms. The most trivial example appears when $\mathcal{SF}$ is simply a set of shared constants. Then, $\mathcal{SF}$-terms are necessarily of height 1.

Example 2 If Th is a stably infinite Σ-theory such that

$$\forall c, c' \in \mathcal{SF}_0,\ c \neq c' \Rightarrow Th \models c \neq c',$$

then Th is stably generated by $\mathcal{SF}_0$-terms (of height 1).

In the particular case of shared constants, 1-satisfiability corresponds to satisfiability and there is no universally quantified variables in the formula $basic_{|V}$ introduced in Definition 4. According to Proposition 9, we obtain the following result:

Theorem 4 *Satisfiability of quantifier-free* $\Sigma_1 \cup \Sigma_2$*-formulae w.r.t.* $Th_1 \cup Th_2$ *is decidable if*

- $\Sigma_1 \cap \Sigma_2 = \mathcal{SF}_0$ *(a set of constants),*
- *satisfiability of quantifier-free* Σ_i*-formulae is decidable w.r.t.* Th_i,
- Th_i *is a stably infinite theory (for* $i = 1, 2$*) such that*

$$\forall c, c' \in \mathcal{SF}_0,\ c \neq c' \Rightarrow Th_i \models c \neq c'.$$

We could also introduce more complicated (and more artificial) structures generated by $\mathcal{SF}$-terms where only a set of terms of height n is sufficient to define non-shared functions and predicates.

Example 3 Consider the convergent rewrite system $R_i = \{h_i(f(x)) \to h_i(a_i)\}$ built over the signature $\Sigma_i = \{f, h_i, a_i\}$. The Σ_i-algebra $\mathcal{T}(\Sigma_i, \mathcal{X}) \downarrow_{R_i}$ of normalized terms w.r.t. R_i may be seen as a structure generated by $\{f\}$-terms over the set A_i of normalized terms without f as top-symbol. All terms in $\mathcal{T}(\mathcal{SF}, A_i)$ behave like a_i with respect to the non-shared function h_i. The same remark holds also for the following convergent rewrite system $R_i = \{x *_i -(y) \to 0, (-y) *_i x \to 0, x *_i 0 \to 0, 0 *_i x \to 0\}$ where all $\{-, 0\}$-terms are mapped onto 0 by $*_i$.

The previous example can be generalized and we could imagine a Σ-structure $\mathcal{M}$ generated by $\mathcal{SF}$-terms over A such that there exist an integer n and an infinite set $B \subseteq \mathcal{T}_n(\mathcal{SF}, A)$ verifying:

- For any $g \in \mathcal{F} \backslash \mathcal{SF}$, the range of $g_{\mathcal{M}}$ is included in $\mathcal{T}_n(\mathcal{SF}, A)$,
- $\forall p \in \mathcal{P}_m,\ \forall g \in (\mathcal{F} \backslash \mathcal{SF})_m,\ \forall \vec{a} \in \mathcal{T}(\mathcal{SF}, A)^m,\ \forall k \in [1, m]$,
  $a_k \notin \mathcal{T}_n(\mathcal{SF}, A) \Rightarrow (\forall b \in B,\ g_{\mathcal{M}}(\vec{a}) = g_{\mathcal{M}}(a_1, \ldots, a_{k-1}, b, a_{k+1}, \ldots, a_m)$
  and $p_{\mathcal{M}}(\vec{a})$ iff $p_{\mathcal{M}}(a_1, \ldots, a_{k-1}, b, a_{k+1}, \ldots, a_m))$.

These assumptions are rather technical and not easy to fulfill. For sake of simplicity, we do not consider in details theories with such structures as models. However, the reader can verify that a quantifier-free formula φ is satisfiable in $\mathcal{M}$ if and only if there exists a $\mathcal{M}$-solution of φ such that its range is included in $\mathcal{T}_n(\mathcal{SF}, A)$, provided φ does not contain any shared function symbol at the top of its equations and disequations. Indeed, the equation $x = t$, where $t \in \mathcal{T}_{n+1}(\mathcal{SF}, \mathcal{X})$ and x does not occur in t, cannot be satisfied thanks to this kind of solutions. So, we would need first to transform such formulae having shared symbols at the top of equations and disequations. In this context, the following transformation rules (used for unification) should be added:

Decompose

$$\frac{\varphi \wedge f(s_1, \ldots, s_m) = f(t_1, \ldots, t_m)}{\varphi \wedge \bigwedge_{k=1}^{m} s_k = t_k} \quad \text{if } f \in \mathcal{SF}_m$$

Fail

$$\frac{\varphi \wedge f(\vec{s}) = f'(\vec{t})}{\bot} \quad \text{if } \{f, f'\} \subseteq \mathcal{SF}, f \neq f'$$

Replace

$$\frac{\varphi \wedge x = f(\vec{s})}{\varphi\{x \mapsto f(\vec{s})\}} \quad \text{if } f \in \mathcal{SF}, x \notin \mathcal{V}(f(\vec{s}))$$

These rules preserve satisfiability since the structure $\mathcal{M}$ is generated by $\mathcal{SF}$-terms.

6 Applications

The decomposition algorithm can be applied for solving the satisfiability problem w.r.t. $Th(\mathcal{A}_1) \cup Th(\mathcal{A}_2)$ where $\mathcal{A}_1$ and $\mathcal{A}_2$ are specific structures like term algebras.

6.1 Combining equational theories

Most of the decision algorithms related to a set of equational axioms E are in fact developed for decision problems in the particular term algebra $\mathcal{T}(\Sigma, \mathcal{X})/=_E$ like for instance:

- E-unifiability, the satisfiability of a conjunction of equations in $\mathcal{T}(\Sigma, \mathcal{X})/=_E$,

- E-equality, the validity of an equation in $\mathcal{T}(\Sigma, \mathcal{X})/=_E$.

The well-known result of G. Birkhoff [5] states a connection between E-equality and validity w.r.t. E but this is unfortunately not sufficient for an equivalence between satisfiability in $\mathcal{T}(\Sigma, \mathcal{X})/=_E$ and satisfiability w.r.t. E. According to Proposition 3, the theory Th_E of interest is here the theory $Th(\mathcal{T}(\Sigma, \mathcal{X})/=_E)$. We assume that the cardinality of $\mathcal{T}(\Sigma, \mathcal{X})/=_E$ is strictly greater than 1 or equivalently there is no E-equality between two different variables in $\mathcal{X}$. Since $\mathcal{X}$ is an infinite denumerable set, the domain of $\mathcal{T}(\Sigma, \mathcal{X})/=_E$ is still infinite and so Th_E is stably infinite. Due the equivalence between satisfiability w.r.t Th_E and satisfiability in $\mathcal{T}(\Sigma, \mathcal{X})/=_E$, a E-unification algorithm can be used for computing solutions of equations which are then propagated to disequations. Finally, we simply have to check that left-hand sides and right-hand sides of these resulting disequations are not E-equal.

Proposition 10 Satisfiability of quantifier-free formulae w.r.t. Th_E is decidable if a finitary E-unification algorithm is known and if E-equality is decidable.

Proof: Let $\varphi = ((\bigwedge_{k \in K} s_k = t_k) \wedge (\bigwedge_{k' \in K'} s'_k \neq t'_k))$. Then

$$(\varphi)_{Th_E} = \bigvee_{\sigma \in CSU_E(\bigwedge_{k \in K} s_k = t_k)} \bigwedge_{k' \in K'} (s'_k\sigma \neq t'_k\sigma)_{Th_E}$$

where $(s \neq t)_{Th_E} = \bot$ if $s =_E t$, else $\top$. CSU_E denotes a complete set of E-unifiers [9]. □

The decomposition algorithm provides under the adequate assumptions a decision algorithm for satisfiability w.r.t. $Th_{E_1} \cup Th_{E_2}$. But the reader must be aware that this problem is neither satisfiability w.r.t. $Th_{E_1 \cup E_2}$ nor satisfiability in $\mathcal{T}(\Sigma_1 \cup \Sigma_2, \mathcal{X})/=_{E_1 \cup E_2}$.

We could also suppose that E is a convergent term rewriting system R such that function symbols in $\mathcal{SF}$ are **constructors**. Then $\mathcal{T}(\Sigma, \mathcal{X})/=_E$ is isomorphic to $\mathcal{T}(\Sigma, \mathcal{X}) \downarrow_R =$ $\mathcal{T}(\mathcal{SF}, A)$, where A denotes the set of normalized terms with a top-symbol not in $\mathcal{SF}$, and so $\mathcal{TH}(\mathcal{T}(\Sigma, \mathcal{X})/=_E)$ is a theory stably generated by $\mathcal{SF}$-terms over A. Shared function symbols were already assumed to be constructors in the framework developed in [7] and it was of greatest interest for the modular construction of $E_1 \cup E_2$-unification algorithms.

6.2 Combining equational theories of non-disjoint finite Post algebras

In this section, the only shared function symbols are constants. The problem of combining Post algebras has been already studied in [14], but the corresponding cooperation algorithm is much more complicated since it needs elementary solvers based on unification w.r.t. linear constant restriction [3].

Definition 6 Let $B_i = \{b_0^i, \ldots, b_{n_i-1}^i\}$ be a finite totally ordered set such that b_0^i denotes the minimal element of B_i and $b_{n_i-1}^i$ denotes the maximal element of B_i and let $\Sigma_{P_i} =$

$\{+^i, *^i, C^i_{b_0}, \ldots, C^i_{b_{n_i-1}}, b^i_0, \ldots, b^i_{n_i-1}\}$ be a set of function symbols. The *Post* Σ_{P_i}-algebra $\mathcal{B}_i$ is the set B_i together with the following operators:

- $\forall b, b' \in B_i,\ b +^i_{\mathcal{B}_i} b' = \max(b, b')$
- $\forall b, b' \in B_i,\ b *^i_{\mathcal{B}_i} b' = \min(b, b')$
- $\forall k \in B_i,\ \forall b \in B_i,\ C^i_{k\,\mathcal{B}_i}(b) = b^i_{n_i-1}$ if $k = b$, else b^i_0
- $\forall b \in B_i,\ b_{\mathcal{B}_i} = b$

Proposition 11 Let $\mathcal{B}_i$ be a finite Post Σ_{P_i}-algebra. Satisfiability of quantifier-free formulae w.r.t. $Th_{\mathcal{B}_i} = Th(\mathcal{T}(\Sigma_{P_i}, \mathcal{X})/=_{\mathcal{B}_i})$ is decidable.

A solver for the satisfiability problem can be designed thanks to a $\mathcal{B}_i$-unification algorithm and a decision algorithm for $\mathcal{B}_i$-equality, as previously.

Proposition 12 Let $\mathcal{B}_1$ and $\mathcal{B}_2$ be respectively a finite Post Σ_{P_1}-algebra and a finite Post Σ_{P_2}-algebra such that $B_1 \cap B_2$ is non-necessarily disjoint.
Satisfiability of quantifier-free formulae w.r.t. $Th_{\mathcal{B}_1} \cup Th_{\mathcal{B}_2}$ is decidable.

The next example shows how the computation of solved forms helps to eliminate the non-determinism inherent to the decomposition algorithm.

Example 4 Let $\mathcal{B}_1$ be the boolean $\{|, \&, \bar{}, 0, 1\}$-algebra (where $|$ stands for (or), $\&$ for (and)) and $\mathcal{B}_2$ be the Post $\{+, *, C_0, C_1, C_2, 0, 1, 2\}$-algebra. The formula $\varphi = (\bar{x} = 0 \ \wedge\ 1 + (x|y) = 1 \ \wedge\ C_1(y) = 2)$ is equivalent to $(\varphi_1 = (\bar{x} = 0 \ \wedge\ X = x|y)) \wedge (\varphi_2 = (C_1(y) = 2 \ \wedge\ 1 + X = 1))$. Since $\bar{x} = 0$ is equivalent to the solved form $x = 1$, φ_1 is equivalent to $x = 1 \ \wedge\ X = 1|y$. In the same way, φ_2 is equivalent to $y = 1 \ \wedge\ 1 + X = 1$. Then, we can propagate $y = 1$ into φ_1. Then, we have that $\varphi_1 \wedge y = 1$ is equivalent to $x = 1 \ \wedge\ y = 1 \ \wedge\ X = 1$. Again, we can propagate $X = 1$ into φ_2. We get that $\varphi_2 \wedge X = 1$ is equivalent to $y = 1 \ \wedge\ X = 1$. Finally, φ is satisfiable and a solution is $\{x \mapsto 1, y \mapsto 1\}$.

7 Conclusion

We have presented a non-deterministic decomposition algorithm for solving in a modular way the satisfiability problem w.r.t. a non-disjoint union of theories. The non-determinism is very helpful for the understanding of the algorithm based only on four steps namely purification, instantiation, identification and decision. In the disjoint case, our decomposition algorithm is identical to the one developed recently in [20]. But, for the moment, the algorithm cannot be efficiently implemented: it was not our aim in this paper. Further work is still necessary to make it more deterministic. One idea could be to propagate as much as possible in one theory the identifications and instantiations by shared terms deduced in the other theory. In that case, algorithms for solving the satisfiability problem should be replaced by more

powerful algorithms devoted to the efficient deduction of the "largest shared conclusion" of a formula. We expect also that this strategy can capture more theories than those considered in this paper. In the same way, it would be helpful for the combination process if pure formulae were transformed to some kind of solved forms according to elementary solvers.

Finally, our combined structure has to be compared with the amalgamation product described in [1, 2]. Even if the contexts are completely different, these two constructions have some similarities and the respective decomposition algorithms are closed.

Acknowledgements: I would like to thank Hélène Kirchner for many helpful comments, Franz Baader, Klaus Schulz and the anonymous referees for a lot of pertinent and constructive remarks.

References

[1] F. Baader and K. Schulz. Combination of constraint solving techniques: An algebraic point of view. In *Proceedings 6th Conference on Rewriting Techniques and Applications, Kaiserslautern (Germany)*, volume 914 of *Lecture Notes in Computer Science*, pages 352–366. Springer-Verlag, 1995.

[2] F. Baader and K. Schulz. On the combination of symbolic constraints, solution domains, and constraint solvers. In *Proceedings of the first International Conference on Principles and Practice of Constraint Programming - CP'95, Cassis (France)*, volume 976 of *Lecture Notes in Computer Science*, pages 380–397. Springer-Verlag, 1995.

[3] F. Baader and K. Schulz. Unification in the union of disjoint equational theories: Combining decision procedures. In Deepak Kapur, editor, *11th International Conference on Automated Deduction, Saratoga Springs (USA)*, volume 607 of *Lecture Notes in Artificial Intelligence*, pages 50–65. Springer-Verlag, June 15–18, 1992.

[4] F. Baader and K. Schulz. Combination techniques and decision problems for disunification. *Theoretical Computer Science*, 142:229–255, 1995.

[5] G. Birkhoff. On the structure of abstract algebras. *Proceedings Cambridge Phil. Soc.*, 31:433–454, 1935.

[6] A. Boudet, J.-P. Jouannaud, and M. Schmidt-Schauß. Unification in boolean rings and abelian groups. In C. Kirchner, editor, *Unification*, pages 267–296. Academic Press, London, 1990.

[7] E. Domenjoud, F. Klay, and Ch. Ringeissen. Combination techniques for non-disjoint equational theories. In Alan Bundy, editor, *Proceedings 12th International Conference on Automated Deduction, Nancy (France)*, volume 814 of *Lecture Notes in Artificial Intelligence*, pages 267–281. Springer-Verlag, June/July 1994.

[8] A. Herold. *Combination of Unification Algorithms in Equational Theories*. PhD thesis, Universität Kaiserslautern (Germany), 1987.

[9] J.-P. Jouannaud and C. Kirchner. Solving equations in abstract algebras: a rule-based survey of unification. In J.-L. Lassez and G. Plotkin, editors, *Computational Logic. Essays in honor of Alan Robinson*, chapter 8, pages 257–321. The MIT press, Cambridge (MA, USA), 1991.

[10] H. Kirchner and Ch. Ringeissen. Combining symbolic constraint solvers on algebraic domains. *Journal of Symbolic Computation*, 18(2):113–155, 1994.

[11] G. Nelson. Techniques for program verification. Technical Report CS-81-10, Xerox Palo Research Center California USA, 1981.

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

[13] D. C. Oppen. Complexity, convexity and combinations of theories. *Theoretical Computer Science*, 12:291–302, 1980.

[14] Ch. Ringeissen. Unification in a combination of equational theories with shared constants and its application to primal algebras. In *Proceedings of the 1st International Conference on Logic Programming and Automated Reasoning, St. Petersburg (Russia)*, volume 624 of *Lecture Notes in Artificial Intelligence*, pages 261–272. Springer-Verlag, 1992.

[15] Ch. Ringeissen. *Combinaison de Résolutions de Contraintes*. Thèse de Doctorat d'Université, Université de Nancy 1, December 1993.

[16] M. Schmidt-Schauß. Combination of unification algorithms. *Journal of Symbolic Computation*, 8(1 & 2):51–100, 1989. Special issue on unification. Part two.

[17] R. Shostak. A practical decision procedure for arithmetic with function symbols. *Journal of the ACM*, 26(2):351–360, 1979.

[18] R. Shostak. Deciding combination of theories. *Journal of the ACM*, 31(1):1–12, 1984.

[19] E. Tidèn. *First-order unification in combinations of equational theories*. PhD thesis, The Royal Institute of Technology, Stockholm, 1986.

[20] C. Tinelli. Extending the CLP scheme to unions of constraint theories. Master's thesis, Department of Computer Science, University of Illinois, Urbana-Champaign, Illinois, October 1995.

[21] K. Yelick. Unification in combinations of collapse-free regular theories. *Journal of Symbolic Computation*, 3(1 & 2):153–182, April 1987.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399