

ANR-08-DEFI-005
DECERT

Task 2: Report on New Combination Methods
for (Fragments of) Arithmetic and Non-stably Infinite Theories

The Decert Consortium

January 5, 2010

Chapter 1

Introduction

The goal of this report is to present the possible extensions of the Nelson-Oppen combination method. We distinguish two main research directions: (1) the extension to non-stably infinite theories and (2) the extension to non-disjoint theories. We present the recent advances that allow us to go beyond the assumptions classically used in the Nelson-Oppen method. These results allow us to get more expressive combined decision procedures that are capable to handle a larger class of theories of interest in verification.

The material presented in this report has been published in [Fon09, NRR09d, NRR09c, NRR09a]. The next chapters reuse the content of these papers.

1.1 Non-stably Infinite Theories

In [Fon09] we consider some of the best-known decidable fragments of first-order logic with equality, including the Löwenheim class (monadic FOL with equality, but without functions), Bernays-Schönfinkel-Ramsey theories (finite sets of formulas of the form $\exists^*\forall^*\varphi$, where φ is a function-free and quantifier-free FOL formula), and the two-variable fragment of FOL. In general, these are not stably infinite, and the Nelson-Oppen scheme cannot be used to integrate them into SMT solvers. Noticing some elementary results about the cardinalities of the models of these theories, we show that they can nevertheless be combined with almost any other decidable theory.

1.2 Non-disjoint Theories

We show how to use a non-disjoint extension of the Nelson-Oppen combination method to obtain decision procedures for theories modelling data structures and arithmetic constraints.

We propose a first solution when the incorporated arithmetic operator allows to express only linear increments, i.e. when the considered constraints have to be interpreted modulo the theory of integer offsets [NRR09d]. We present a superposition calculus dedicated to theories that model some data structures and that share the integer offsets; we show that the calculus is capable to actually decide the existential fragment of these theories and that can be plugged into the non-disjoint extension of the Nelson-Oppen combination method, deriving thus decision procedure for theories modeling more complex data structures.

As a second contribution [NRR09c], we focus on the union of a data-structure and a theory of arithmetic sharing a successor function satisfying the injectivity and the acyclicity axioms. This union allows us to handle more expressive arithmetic constraints and to obtain a combined decision procedure in which the procedures for individual theories can be constructed by using an appropriate superposition calculus for the data-structure and classical solving techniques for the theory of arithmetic (Gauss elimination, Fourier-Motzkin elimination, Groebner bases computation).

To go beyond a shared unary successor symbol, we consider the case of abelian groups [NRR09a]. The possibility of having a shared addition symbol permits us to augment the expressiveness on the arithmetical

part, lifting from linear increment expressed by using the successor symbols, to increment expressed as sums. This allows to handle, e.g., useful counting functions for data structures such as trees. We consider the completeness and the effectiveness of the non-disjoint combination method when the theory of abelian groups is shared. For the completeness, we show that the theory of abelian groups can be embedded into a theory admitting quantifier elimination. For achieving effectiveness, we rely on a superposition calculus modulo abelian groups developed by Godoy and Nieuwenhuis. We consider a many-sorted and constraint-free version of the calculus, in which we use a restricted form of unification in abelian groups with free symbols, and in which only literals are involved.

To be effective in all our papers mentioned above, the non-disjoint extension of the Nelson-Oppen combination method makes use of procedures able to compute the logical consequences over the shared signature.

Chapter 2

Beyond Stably Infinite Theories

2.1 Introduction

Among automated deduction techniques for the verification of computer systems, SMT solvers (Satisfiability Modulo Theories) are nowadays attracting a lot of interest. These solvers are built on top of SAT solvers for propositional logic and include decision procedures for different first-order theories, thus providing more expressive input languages. Usually, SMT solvers implement a combination of a fixed number of theories such as linear arithmetic, uninterpreted symbols, list operators, bit vectors, etc., based on the classical Nelson-Oppen framework [16, 21] for combining decidable theories. This framework covers combinations of disjoint theories provided they are stably infinite: if a set of quantifier-free formulas has a model with respect to a theory, it should also have an infinite model. For instance, a combination of decision procedures for integer linear arithmetic and for the empty theory (equality and uninterpreted symbols) can detect the unsatisfiability of the formula

$$x \leq y \wedge y \leq x + f(x) \wedge P(h(x) - h(y)) \wedge \neg P(0) \wedge f(x) = 0.$$

The Bernays-Schönfinkel-Ramsey (BSR) class [4, 17] is certainly the most well-known decidable class of first-order theories. A BSR theory is a finite set (conjunction) of formulas of the form $\exists^* \forall^* \varphi$, where the first-order formula φ is function-free and quantifier-free. Many verification problems generate formulas in this class (see for instance [11]). The CASC competition [20] for first-order theorem provers has a dedicated division (EPR, Effectively Propositional) for this class. BSR theories are in general not stably infinite. As a trivial example, consider the BSR theory $\forall x \forall y. x = y$ that only accepts models with singleton domains. The Nelson-Oppen framework does not apply to combinations including BSR theories.

A Löwenheim theory with equality is a finite set of closed formulas in a language containing only unary predicates, and no function except constants. This class is also known as first-order relational monadic logic, and it is decidable. The theory $\forall x \forall y. x = y$ also belongs to the Löwenheim class, and hence the Nelson-Oppen framework does not apply to this class.

The last decidable class we study in this paper is the class of finitely axiomatized first-order theories built in a language with equality, only two variables, and no functions (except constants). Again $\forall x \forall y. x = y$ belongs to this class, and the Nelson-Oppen framework is not appropriate.

The objective of the present paper is to lay the ground for incorporating theories from these three well-known classes into SMT solvers.

We are not aware of previous combination results about the full Löwenheim class with equality or the full two-variable fragment with equality. However, it has already been observed [23] that, thanks to its finite model property, a BSR theory can be combined with a theory \mathcal{T} provided the following conditions hold:

- if a set of ground literals L is \mathcal{T} -satisfiable, then the minimal cardinality of \mathcal{T} -models for L can be computed;

- \mathcal{T} only has finite models.

The second requirement is quite strong. In particular, it is not satisfied by combinations including decidable fragments of arithmetic, which admit only infinite models. For example, the combination scheme of [23] cannot be used to decide the satisfiability of the set of literals such as

$$\{a > 0, a < 2, a + b = 2, b > 0, A(f(a)), \neg C(f(b))\}$$

(where $a, b, f(a), f(b)$ are integers and $+, <, >, 0, 2$ have their usual meaning over integers) with respect to the BSR theory

$$\mathcal{T} = \{\forall x [(A(x) \vee B(x)) \equiv (C(x) \vee D(x))]\}.$$

The classical Nelson-Oppen combination scheme and that of [23] introduce rather strong requirements on the theories in the combination, and these requirements ensure that component theories agree on model cardinalities. For instance, the stably infinite requirement ensures that both theories will agree on the cardinality \aleph_0 for their models. But essentially, the combination process is a matter of matching the interpretation of shared symbols (by exchanging disjunction of equalities), and cardinalities of the models of the theories [12, 23, 9].

We observe in this paper that it is possible to compute all the cardinalities of models admitted by a theory in the BSR, Löwenheim, or two-variable classes with equality. The set of cardinalities accepted by such theories even has a very particular structure. In section 2.3 we characterize this structure, and show that any decidable theory that verifies this property can be combined with a decidable theory \mathcal{T} provided \mathcal{T} fulfils very liberal constraints. These constraints are trivially met in most practical cases.

For convenience, the results in this paper are presented in an *unsorted* framework, although most SMT-solvers work in a many-sorted logic framework (see for instance [8]). Our results could be transferred to a many-sorted framework, at the expense of heavier notations.

The chapter is structured as follows: Section 2.2 introduces basic concepts and notations. Section 2.3 presents the general scheme for combining (not necessarily stably infinite) theories, and introduces the required notions for the new combination results with the considered first-order decidable classes. Sections 2.4, 2.5 and 2.6 respectively present essential cardinality results about the Löwenheim, BSR, and two-variables classes. We do not claim that the results in those three sections are original. Some of them can be found in classical Model Theory books [5, 6, 7]. But some of them are less known. This paper thus presents them together, and relates them to the combination scheme. Section 2.7 presents a simple example, and Section 2.8 concludes the chapter.

2.2 Notations

A first-order language is a tuple $\mathcal{L} = \langle \mathcal{V}, \mathcal{F}, \mathcal{P} \rangle$ such that \mathcal{V} is an enumerable set of variables, \mathcal{F} and \mathcal{P} are sets of function and predicate symbols. Every function and predicate symbol is assigned an arity. Nullary predicates are propositions, and nullary functions are constants. Terms and formulas over the language \mathcal{L} are defined in the usual way. A ground term is a term without variables. An atomic formula is either $t = t'$ where t and t' are terms, or a predicate symbol applied to the right number of terms. Formulas are built from atomic formulas, Boolean connectives ($\neg, \wedge, \vee, \Rightarrow, \equiv$), and quantifiers (\forall, \exists). A formula with no free variables is closed. A theory is a set of closed formulas. Two theories are disjoint if no predicate symbol in \mathcal{P} or function symbol in \mathcal{F} appears in both theories. A finite theory or a finitely axiomatized theory is a finite set of formulas.

An interpretation \mathcal{I} for a first-order language provides a domain D , a total function $\mathcal{I}[f] : D^r \rightarrow D$ of appropriate arity for every function symbol f , a predicate $\mathcal{I}[p] : D^r \rightarrow \{\top, \perp\}$ of appropriate arity for every predicate symbol p , and an element $\mathcal{I}[x] \in D$ for every variable x . By extension, an interpretation defines a value in D for every term, and a truth value for every formula. The notation $\mathcal{I}_{x_1/d_1, \dots, x_n/d_n}$ stands for the interpretation that agrees with \mathcal{I} , except that it associates the elements d_i to the variables x_i .

A model of a formula (or a theory) is an interpretation in which the formula (resp., every formula in the theory) evaluates to true. A formula or theory is satisfiable if it has a model, and it is unsatisfiable

otherwise. A formula G is \mathcal{T} -satisfiable if it is satisfiable in the theory \mathcal{T} , that is, if $\mathcal{T} \cup \{G\}$ is satisfiable. A \mathcal{T} -model of G is a model of $\mathcal{T} \cup \{G\}$. A formula G is \mathcal{T} -unsatisfiable if it has no \mathcal{T} -models.

The cardinality of an interpretation is the cardinality of its domain. The restriction of a predicate p on domain D to domain $D' \subseteq D$ is the predicate p' with domain D' such that p and p' have the same truth value for all arguments in D' .

A formula is universal if it is of the form $\forall x_1 \dots \forall x_n. \varphi$ where φ is quantifier-free. A Skolem formula is a formula where all universal quantifiers appear with a positive polarity, and all existential quantifiers appear with a negative polarity. It is always possible to transform a given formula into an equisatisfiable Skolem formula, using Skolemization. We refer to [2] for Skolemization.

2.3 Combination of theories

Assume we want to study the satisfiability of the set of literals

$$L = \{a \leq b, b \leq a + f(a), P(h(a) - h(b)), \neg P(0), f(a) = 0\}$$

in the combination of the integer linear arithmetic theory \mathcal{T}_1 and the empty theory (i.e. the theory of uninterpreted symbols) \mathcal{T}_2 . First, a *separation* is built by introducing fresh uninterpreted constants¹, to produce the equisatisfiable problem

$$\begin{aligned} L_1 &= \{a \leq b, b \leq a + v_1, v_1 = 0, v_2 = v_3 - v_4, v_5 = 0\} \\ L_2 &= \{P(v_2), \neg P(v_5), v_1 = f(a), v_3 = h(a), v_4 = h(b)\}. \end{aligned}$$

The set L_1 only contains arithmetic symbols and uninterpreted constants. The symbols in L_2 are all uninterpreted. The only shared symbols are the uninterpreted constants in the set $S = \{a, b, v_1, v_2, v_3, v_4, v_5\}$. Notice that although L is unsatisfiable in $\mathcal{T}_1 \cup \mathcal{T}_2$, L_1 is \mathcal{T}_1 -satisfiable, and L_2 is \mathcal{T}_2 -satisfiable; it is not sufficient for the decision procedures for \mathcal{T}_1 and \mathcal{T}_2 to only examine the satisfiability of their part of the separation. Indeed, the decision procedures also have to “agree on the common part”. This can be captured using the notion of arrangement:

Definition 2.1. *An arrangement \mathcal{A} for a set of constant symbols S is a maximal satisfiable set of equalities and inequalities $a = b$ or $a \neq b$, with $a, b \in S$.*

The following theorem (other formulations can be found in [22, 23, 12]) then states the completeness of the combination of decision procedures:

Theorem 2.1. *Assume \mathcal{T}_1 and \mathcal{T}_2 are theories over the disjoint languages \mathcal{L}_1 and \mathcal{L}_2 , and L_i ($i = 1, 2$) is a set of literals in \mathcal{L}_i augmented by a finite set of fresh constant symbols S . Then $L_1 \cup L_2$ is $\mathcal{T}_1 \cup \mathcal{T}_2$ -satisfiable if and only if there exist an arrangement \mathcal{A} of S , a cardinality k , and a \mathcal{T}_i -model \mathcal{M}_i of $\mathcal{A} \cup L_i$ with cardinality k for $i = 1, 2$.*

Proof. Assume \mathcal{I} is an interpretation on domain D for a language \mathcal{L} , and \mathcal{L}' is a sub-language of \mathcal{L} , i.e. the set of variable, function, and predicate symbols in \mathcal{L}' are subsets of their counterpart in \mathcal{L} . We say that the interpretation \mathcal{I}' on domain D for language \mathcal{L}' is the restriction of \mathcal{I} if \mathcal{I}' and \mathcal{I} give the same interpretation for the symbols in \mathcal{L}' .

The condition is necessary. Assume \mathcal{M} is a $\mathcal{T}_1 \cup \mathcal{T}_2$ -model for $L_1 \cup L_2$. \mathcal{M} perfectly defines an arrangement \mathcal{A} of S : indeed $a = b \in \mathcal{A}$ with $a, b \in S$ iff $a = b$ is true according to \mathcal{M} . The restriction of \mathcal{M} to \mathcal{L}_i augmented with the constant symbols S is a \mathcal{T}_i -model for $\mathcal{A} \cup L_i$, $i = 1, 2$.

The condition is sufficient. Assume that \mathcal{A} is an arrangement for S , \mathcal{M}_1 on domain D_1 is a \mathcal{T}_1 -model for $\mathcal{A} \cup L_1$, \mathcal{M}_2 on domain D_2 is a \mathcal{T}_2 -model for $\mathcal{A} \cup L_2$, and $|D_1| = |D_2|$. Since both \mathcal{M}_1 and \mathcal{M}_2 are models of \mathcal{A} , there exist two interpretations \mathcal{M}'_1 and \mathcal{M}'_2 on the same domain that are respectively isomorphic to

¹Traditionally combination schemes use variables for this role. Since variables will be used in quantifiers in the following sections, for consistency and clarity we will rather use uninterpreted constants here.

\mathcal{M}_1 and \mathcal{M}_2 and such that $\mathcal{M}'_1[a] = \mathcal{M}'_2[a]$ for every $a \in S$. It is then possible to build an interpretation \mathcal{M} such that its restriction to the language \mathcal{L}_i augmented with S is \mathcal{M}'_i , $i = 1, 2$. \mathcal{M} is a $\mathcal{T}_1 \cup \mathcal{T}_2$ -model of $L_1 \cup L_2$. \square

Checking the existence of a model is the task of the decision procedures for the decidable theories in the combination. The previous theorem however also imposes a restriction on cardinalities: the two decision procedures should exhibit a model with the same cardinality. A theory \mathcal{T} is said to be stably infinite when every \mathcal{T} -satisfiable set of literals has a model with cardinality \aleph_0 . Combining only stably infinite theories is a radical solution to the cardinality requirement in the previous theorem; k can always be \aleph_0 . Since the empty theory and the theory of integer linear arithmetic are both stably infinite, the set of literals L in our example is $\mathcal{T}_1 \cup \mathcal{T}_2$ -satisfiable if and only if there exists an arrangement \mathcal{A} of the seven variables in S such that $\mathcal{A} \cup L_i$ is \mathcal{T}_i -satisfiable for $i = 1$ and $i = 2$. No such arrangements exist, and indeed, L is $\mathcal{T}_1 \cup \mathcal{T}_2$ -unsatisfiable.

The first-order decidable classes considered in this paper contain theories that are not stably infinite. For instance the formula $\forall x (x = a \vee x = b)$ belongs to the BSR, Löwenheim and two variable classes, and it only accepts models with at most two elements. A combination scheme to handle such theories requires to carefully examine cardinalities. The notion of spectrum is helpful for this task:

Definition 2.2. *The spectrum of a theory \mathcal{T} is the set of cardinalities k such that \mathcal{T} is satisfiable in a model of cardinality k .²*

Using this definition and Theorem 2.1, a combination scheme for disjoint theories (not necessarily stably infinite) can thus be easily expressed:

Corollary 2.1. *Given two theories \mathcal{T}_1 and \mathcal{T}_2 over the disjoint languages \mathcal{L}_1 and \mathcal{L}_2 , the $\mathcal{T}_1 \cup \mathcal{T}_2$ -satisfiability problem for sets of literals (written in the union of the languages \mathcal{L}_1 and \mathcal{L}_2) is decidable if, for any sets of literals L_1 and L_2 (respectively written in the languages \mathcal{L}_1 and \mathcal{L}_2 augmented with a finite set of fresh uninterpreted constants) it is possible to compute if the intersection of the spectrums for $\mathcal{T}_1 \cup L_1$ and for $\mathcal{T}_2 \cup L_2$ is non-empty.*

In the case of stably infinite decidable theories, it is guaranteed that, if $\mathcal{T}_1 \cup L_1$ and $\mathcal{T}_2 \cup L_2$ are satisfiable, both spectrums contain cardinality \aleph_0 , and so their intersection is trivially non-empty.

To characterize the spectrum of the decidable classes considered in this paper, we introduce the following property:

Definition 2.3. *A theory \mathcal{T} is gentle if, for every set L of literals in the language of \mathcal{T} (augmented by a finite number of fresh constants), the spectrum of $\mathcal{T} \cup L$ can be computed and is either*

- *a finite set of finite cardinalities*
- *the union of a finite set of finite cardinalities and all the (finite and infinite) cardinalities greater than a computable finite cardinality; it is thus co-finite.*

A gentle theory is decidable. In the following sections, we show that the BSR theories, the Löwenheim theories, and finite theories with only two variables are gentle. The empty theory, as a special case of a BSR theory, is gentle. Shiny theories in general (see [23]) are gentle. We also have the following result:

Theorem 2.2. *The union of disjoint gentle theories is a gentle theory.*

Proof. The case for the union of any number of disjoint gentle theories can be proved by induction, and using the case for two gentle theories.

The intersection of two spectrums of gentle theories is also either a finite set of finite cardinalities, or the union of a finite set of finite cardinalities and all the (finite and infinite) cardinalities greater than a (computable) finite cardinality. The case for two gentle theories is thus a direct consequence of Theorem 2.1. \square

²The spectrum of a theory is usually defined as the set of the *finite* cardinalities of its models. We here slightly extend the definition for convenience.

We point out that a theory \mathcal{T} taking part in a combination of theories has some interesting property about its spectrum. Since the \mathcal{T} -satisfiability problem for sets of literals (written in the language of the theory plus fresh constants) is decidable, it is also possible to assess for any set of literals L if $\mathcal{T} \cup L$ has a model of cardinality greater than a given number k . Indeed it suffices to introduce k new constants a_1, \dots, a_k and check the satisfiability of $\mathcal{T} \cup L \cup \{a_i \neq a_j \mid i \neq j, i, j = 1, \dots, k\}$. Also notice that it is always possible to decide if a finite first-order theory admits a model of a given finite cardinality. Indeed there are only a finite number of interpretations for a finite language, and it takes a finite time to check if a given finite interpretation is a model of the finite theory.

Some widely used theories are not gentle, but in practical cases they can be combined with gentle theories:

Theorem 2.3. *Given a gentle theory \mathcal{T} and another disjoint theory \mathcal{T}' , the $\mathcal{T} \cup \mathcal{T}'$ -satisfiability problem for sets of literals written in the union of their language is decidable if one of the following cases holds:*

- \mathcal{T}' is gentle;
- \mathcal{T}' is a decidable finitely axiomatized first-order theory;
- \mathcal{T}' is a decidable theory that only admits a fixed finite (possibly empty) known set of finite cardinalities for its models, and possibly infinite models.

Proof. Assume $L \cup L'$ is the separation to check for $\mathcal{T} \cup \mathcal{T}'$ -satisfiability. If an arrangement \mathcal{A} is such that $\mathcal{A} \cup L$ is \mathcal{T} -satisfiable, and $\mathcal{A} \cup L'$ is \mathcal{T}' -satisfiable, then it is possible to compute the spectrum \mathcal{S} of $\mathcal{T} \cup \mathcal{A} \cup L$. Either \mathcal{S} is a finite set of finite cardinalities, or it is a union of a finite set of finite cardinalities and the set of all cardinalities greater than a number k .

If \mathcal{T}' is also gentle, it is possible to compute the spectrum of $\mathcal{T}' \cup \mathcal{A} \cup L'$, and the intersection of the two spectrums can easily be computed.

If \mathcal{T}' is a decidable finite first-order theory, it is possible to check if $\mathcal{T}' \cup \mathcal{A} \cup L'$ admits a cardinality in the finite part of \mathcal{S} , and, if \mathcal{S} is infinite, it is possible to check if $\mathcal{T}' \cup \mathcal{A} \cup L'$ admits a cardinality greater than k .

If \mathcal{T}' is a decidable theory that only admits a fixed finite known set of cardinalities for its models, it suffices to check if one of these cardinalities is in the spectrum \mathcal{S} . The considered theories are first-order, and the Löwenheim-Skolem theorem states that, if a theory has an infinite model, it has models for every infinite cardinality. Infinite cardinalities can thus be understood as one cardinality. \square For instance, the real or integer linear arithmetic theories (or combinations involving real or integer linear arithmetic) fall into the last case, and the usual theories for arrays fall into the second one.

2.4 The Löwenheim class with equality

A Löwenheim theory is a finite set of closed formulas in a language containing only unary predicates, and no functions except constants. This class is also known as first-order relational monadic logic. Usually one distinguishes the Löwenheim class with and without equality. The Löwenheim class has the finite model property (and is thus decidable) even with equality. Full monadic logic *without equality*, i.e. the class of finite theories over a language containing symbols (predicates and functions) of arity at most 1, also has the finite model property. Considering monadic logic with equality, the class of finite theories over a language containing only unary predicates and just two unary functions is already undecidable. With only one unary function however the class remains decidable, but does not have the finite model property anymore. Since the spectrum for this last class is significantly more complicated [14] than for the Löwenheim class we will here only concentrate on the Löwenheim class with equality (only classes with equality are relevant in our context). More can be found about monadic first-order logic in [5, 6]. In particular, the following Theorem can be found in [6]:

Theorem 2.4. *Assume \mathcal{T} is a Löwenheim theory with equality with n distinct unary predicates. Let q be the number of constants plus the maximum number of nested quantifiers in \mathcal{T} . If \mathcal{T} has a model of some cardinality $\geq q2^n$, then \mathcal{T} has models of every cardinality $\geq q2^n$.*

Proof. For simplicity, assume \mathcal{T} is constant-free and is a single formula. Because \mathcal{T} is finite, it is always possible to get back to such a case by taking the conjunction of all formulas in \mathcal{T} , and then quantify existentially over all constants in the formula.

Let p_1, \dots, p_n be the unary predicates used in \mathcal{T} . Given an interpretation \mathcal{I} on domain D for \mathcal{T} , every element $d \in D$ has a color $c(d) = c_1 \dots c_n \in \{\top, \perp\}^n$ where $c_i = \mathcal{I}[p_i](d)$. We denote by $D_c \subseteq D$ the set of elements with color c .

Two interpretations \mathcal{I} (on domain D) and \mathcal{I}' (on domain D') for a formula ψ are *similar* if

- either $D_c = D'_c$ or $|D_c \cap D'_c| \geq q$ for every color $c \in \{\top, \perp\}^n$;
- $D_c \cap D'_{c'} = \emptyset$ for any two distinct colors $c, c' \in \{\top, \perp\}^n$;
- $\mathcal{I}[x] = \mathcal{I}'[x]$ for every variable free in ψ .

We first prove that, given a formula ψ , two similar interpretations for ψ give the same truth value to ψ and to every sub-formula of ψ .

This is proved by induction on the structure of the (sub-)formula ψ . It is obvious if ψ is atomic, since similar interpretations assign the same value to variables, and since ψ is variable-free. If ψ is $\neg\varphi_1$, $\varphi_1 \vee \varphi_2$, $\varphi_1 \wedge \varphi_2$ or $\varphi_1 \Rightarrow \varphi_2$, the result holds if it also holds for φ_1 and φ_2 .

Assume \mathcal{I} makes true the formula $\psi = \exists x \varphi(x)$. Then there exists some $d \in D$ such that $\mathcal{I}_{x/d}$ is a model of $\varphi(x)$. If $d \in D'$, then $\mathcal{I}'_{x/d}$ is similar to $\mathcal{I}_{x/d}$ and, by the induction hypothesis, it is a model of $\varphi(x)$; \mathcal{I}' is thus a model of ψ . If $d \notin D'$, it means that $|D_{c(d)} \cap D'_{c(d)}| \geq q$. Furthermore, since the whole formula contains at most q nested quantifiers, $\varphi(x)$ contains at most $q - 1$ free variables. Let x_1, \dots, x_m be those variables. There exists some $d' \in D_{c(d)} \cap D'_{c(d)}$ such that $d' \neq \mathcal{I}[x_i]$ for every $i \in \{1, \dots, m\}$. By structural induction, it is easy to show that $\mathcal{I}_{x/d}$ and $\mathcal{I}_{x/d'}$ give the same truth value to $\varphi(x)$. Furthermore $\mathcal{I}_{x/d}$ and $\mathcal{I}'_{x/d'}$ are similar. \mathcal{I}' is thus a model of ψ . To summarize, if \mathcal{I} is a model of ψ , \mathcal{I}' is also a model of ψ . By symmetry, if \mathcal{I}' is a model of ψ , \mathcal{I} is also a model of ψ . Thus, if $\psi = \exists x \varphi(x)$, the results hold if it also holds for $\varphi(x)$. The proof for formulas of the form $\forall x \varphi(x)$ is dual.

If \mathcal{M} on domain D is a model for \mathcal{T} with cardinality $\geq q2^n$, then there exists a color c such that $|D_c| \geq q$. For any cardinality $k \geq q2^n$ one can build a model \mathcal{M}' of cardinality k for \mathcal{T} , similar to \mathcal{M} . \square

Corollary 2.2. *The Löwenheim class has the finite model property.*

Proof. Assume \mathcal{T} is a Löwenheim theory, with n distinct unary predicates. Let q be the maximum number of nested quantifiers in \mathcal{T} . Let \mathcal{I} be a model of \mathcal{T} . According to Theorem 2.4, if \mathcal{I} has an infinite cardinality ($\geq q2^n$), \mathcal{T} also has a finite model (e.g. of cardinality $q2^n$). \square

Corollary 2.3. *The satisfiability problem for the Löwenheim class is decidable.*

Proof. It is well-known that any class of finite first-order theories that has the finite model property is also decidable. The decidability of the Löwenheim class can also be easily proved directly. Assume \mathcal{T} is a Löwenheim theory, with n distinct unary predicates. Let q be the maximum number of nested quantifiers in \mathcal{T} . There exist only a finite number of interpretations of a finite theory for a given cardinality. It is thus decidable to check if \mathcal{T} has a model of cardinality $q2^n$. If such a model exists \mathcal{T} is satisfiable. If no such models exist, Theorem 2.4 states that \mathcal{T} has no models of cardinality $\geq q2^n$. It remains to decide if \mathcal{T} has a model of cardinality $< q2^n$, i.e. it remains to examine a finite number of interpretations. \square

Corollary 2.4. *The spectrum of a Löwenheim theory can be computed and expressed either as a finite set of naturals, or as the union of a finite set of naturals with the set of all the (finite or infinite) cardinalities greater than a natural. The Löwenheim theories are gentle.*

2.5 The Bernays-Schönfinkel-Ramsey class

A Bernays-Schönfinkel-Ramsey theory (BSR) is a finite set of formulas of the form $\exists^* \forall^* \varphi$, where φ is a first-order formula which is function-free (but constants are allowed) and quantifier-free. Bernays and Schönfinkel

first proved the decidability of this class without equality; Ramsey later proved that it remains decidable with equality. The results about the spectrum of BSR theories are less known, but were also originally found by Ramsey.

For simplicity, we will assume that existential quantifiers are Skolemized. In the following, a BSR theory is thus a finite closed set of universal function-free first-order formulas.

Theorem 2.5. *Let \mathcal{T} be a BSR theory, and let k_c be the number of constants in \mathcal{T} , or $k_c = 1$ if \mathcal{T} is constant-free. If \mathcal{T} has a model with cardinality $k \geq k_c$, then \mathcal{T} has a model for every cardinality i , with $k \geq i \geq k_c$.*

Proof. Given a model \mathcal{M} for a BSR theory \mathcal{T} with domain D , then any interpretation \mathcal{M}' such that

- the domain of \mathcal{M}' is a non-empty set $D' \subseteq D$ such that $\mathcal{M}'[a] = \mathcal{M}[a] \in D'$ for every constant a in \mathcal{T} , and
- for every predicate p , $\mathcal{M}'[p]$ is the restriction of $\mathcal{M}[p]$ to the domain D'

is also a model of \mathcal{T} . Intuitively, this states that the elements in the domain that are not assigned to ground terms (i.e. the constants) can be eliminated in a model of a BSR theory. Since \mathcal{M} is a model of \mathcal{T} , for each closed formula $\forall x_1 \dots x_n \varphi$ in \mathcal{T} (where φ is function-free and quantifier-free), and for all $d_1, \dots, d_n \in D' \subseteq D$, $\mathcal{M}_{x_1/d_1, \dots, x_n/d_n}$ is a model of φ . This also means that, for all $d_1, \dots, d_n \in D'$, $\mathcal{M}'_{x_1/d_1, \dots, x_n/d_n}$ is a model of φ , and finally that \mathcal{M}' is a model of $\forall x_1 \dots x_n \varphi$. \square

Theorem 2.6. *There exists a computable function f such that, for any BSR theory \mathcal{T} , if \mathcal{T} has a model of some cardinality $\geq f(\mathcal{T})$, then it has a model for every cardinality $\geq f(\mathcal{T})$.*

Proof. The proof is quite long and requires a non trivial theorem on hypergraph coloring. A partial proof can be found in [6], and a full self-contained proof can be found in the full version of the paper [10]. \square

The proofs of the following corollaries are similar to the corresponding proofs for the Löwenheim class.

Corollary 2.5. *The BSR class has the finite model property.*

Corollary 2.6. *The satisfiability problem for the BSR class is decidable.*

Corollary 2.7. *The spectrum of a BSR theory can be computed and expressed either as a finite set of naturals, or as the union of a finite set of naturals with the set of all the (finite or infinite) cardinalities greater than a natural. BSR theories are gentle.*

2.6 First-order logic with two variables

Following [7], we will denote by FO^2 the class of finite theories built over a language with only two variables, and no functions (except constants). The satisfiability problem for FO^2 is known to be decidable with and without equality (see for instance [5, 7, 13]). Again, we will only concentrate here on the language with equality. This class has the finite model property, and also has very nice properties concerning the cardinalities of its models.

The Scott class is a subset of FO^2 : it is the class of finite theories over a language with only two variables, and no functions (except constants) such that every formula in the theory is of the form $\forall x \forall y \varphi(x, y)$ or $\forall x \exists y \varphi(x, y)$ where $\varphi(x, y)$ is quantifier-free. The satisfiability problem for FO^2 (with equality) is traditionally translated into the satisfiability problem for the Scott class, using the following theorem (see [5, 7] for equivalent theorems):

Theorem 2.7. *There exists an algorithm that, for each finite theory \mathcal{T} of FO^2 , constructs a theory \mathcal{T}' in the Scott class such that \mathcal{T} has a model of a given cardinality if and only if \mathcal{T}' has a model of the same cardinality. The size of \mathcal{T}' is linear with respect to the size of \mathcal{T} .*

Proof. First notice that formula $\forall x (R(x) \equiv Qy \varphi(x, y))$ where Q is either \exists or \forall can be rewritten as a set of formulas in the required form:

- $\forall x (R(x) \equiv \forall y \varphi(x, y)) \longleftrightarrow \forall x \forall y (R(x) \Rightarrow \varphi(x, y)) \wedge \forall x \exists y (\varphi(x, y) \Rightarrow R(x))$
- $\forall x (R(x) \equiv \exists y \varphi(x, y)) \longleftrightarrow \forall x \exists y (R(x) \Rightarrow \varphi(x, y)) \wedge \forall x \forall y (\varphi(x, y) \Rightarrow R(x))$

The theory \mathcal{T} can thus be rewritten into a suitable theory \mathcal{T}' by iteratively applying the following step until no more formulas of unsuitable form exist in the theory:

- select a formula ψ in the theory that does not have the required form;
- choose a sub-formula of form $Qy \varphi(x, y)$ of ψ where Q is \exists or \forall and $\varphi(x, y)$ is quantifier-free;
- take a new unary predicate R not used in the theory;
- define the formula ψ' as ψ where $Qy \varphi(x, y)$ has been substituted by $R(x)$;
- remove ψ from the theory, and add ψ' , and the formulas in the required form for $\forall x (R(x) \equiv Qy \varphi(x, y))$.

□

The following theorem is left as an exercise in [7]. For completeness we here give the full proof.

Theorem 2.8. *There exists a computable function f such that, for any Scott theory \mathcal{T} , if \mathcal{T} has a model of some cardinality $\geq f(\mathcal{T})$, then \mathcal{T} has models for every cardinality $\geq f(\mathcal{T})$.*

Proof. We first assume that every formula ψ_i in \mathcal{T} ($i = 1, \dots, m$) of the form $\forall x \exists y \varphi(x, y)$ is such that every model of $\varphi(x, y)$ is a model of $x \neq y$. This assumption is acceptable if $f(\mathcal{T}) \geq 2$ for all Scott theories \mathcal{T} since for all models with at least two elements $\forall x \exists y \varphi(x, y)$ is equivalent to $\forall x \exists y. x \neq y \wedge (\varphi(x, y) \vee \varphi(x, x))$.

For the rest of the proof, we assume that the Scott theory \mathcal{T} has a model \mathcal{M} on domain D . We define the sets $A = \{\mathcal{M}[a] : a \text{ is a constant in } \mathcal{T}\}$ and $B = D \setminus A$. We establish that if B is larger than a computable cardinality $\geq f(\mathcal{T})$, one can build a model for every cardinality $\geq f(\mathcal{T})$.

Given a first-order language \mathcal{L} , a k -table³ $T[x_1, \dots, x_k]$ over the variables x_1, \dots, x_k is a maximal satisfiable set of atomic formulas and negation of atomic formulas using only variables x_1, \dots, x_k . Given an interpretation \mathcal{I} on domain D and k elements d_1, \dots, d_k of D , the k -table of d_1, \dots, d_k (denoted $T_{\mathcal{I}}[d_1, \dots, d_k]$) is the unique k -table $T[x_1, \dots, x_k]$ such that the interpretation $\mathcal{I}_{x_1/d_1, \dots, x_k/d_k}$ is a model of $T[x_1, \dots, x_k]$. Notice that there are only a finite number of k -tables, for a finite language with no functions except constants. In particular if A is the set of constants, a 1-table is determined by at most $b = \sum_p (|A| + 1)^{\text{arity}(p)}$ Boolean values, where the sum ranges over all predicates in the language. Indeed, given a predicate p of arity r , there are at most $(|A| + 1)^r$ terms that can be built with p and $A \cup \{x\}$. Thus the number of different 1-tables is bounded by $C = 2^b$.

For every formula $\psi_i = \forall x \exists y \varphi_i(x, y)$ in \mathcal{T} ($i = 1, \dots, m$), there exists a total function g_i on domain D ranging on D such that $\mathcal{M}[\varphi_i](d, g_i(d))$ is true for every $d \in D$. The set K (commonly referred as the set of kings) is defined as the union of A and of the possibly empty set of all elements of $d \in D$ such that the 1-table of d is unique, i.e. $T_{\mathcal{M}}[d'] \neq T_{\mathcal{M}}[d]$ for every $d' \in D$ such that $d' \neq d$. The set C (commonly referred as the court) is the possibly empty set $C = K \cup \{g_i(d) \mid d \in K, i = 1, \dots, m\}$. The set S is defined as $T_{\mathcal{M}}[D] \setminus T_{\mathcal{M}}[C]$ where $T_{\mathcal{M}}[D]$ is the set of all 1-tables of elements in D (and similarly for $T_{\mathcal{M}}[C]$). We choose a function h on domain S that ranges on D such that $T_{\mathcal{M}}[h(t)] = t$.

The set D' is defined as $C \cup (S \times \{1, \dots, m\} \times \{0, 1, 2\})$. A model \mathcal{M}' on D' for \mathcal{T} is defined such that:

- $T_{\mathcal{M}'}[d_1, \dots, d_k] = T_{\mathcal{M}}[d_1, \dots, d_k]$ for $d_1, \dots, d_k \in C$, $k \in \mathbb{N}$;
- $T_{\mathcal{M}'}[(t, i, j)]$ is t , for every $(t, i, j) \in D' \setminus C$;
- if $g_i(h(t)) \in K$ then $T_{\mathcal{M}'}[(t, i, j), g_i(h(t))] = T_{\mathcal{M}}[h(t), g_i(h(t))]$;

³We here adopt the notation of [5]. The same notion is also called (atomic) k -type, for instance in [13].

- if $g_i(h(t)) \notin K$ then $T_{\mathcal{M}'}[(t, i, j), (T_{\mathcal{M}}(g_i(h(t))), i, (j+1) \bmod 3)]$ is equal to $T_{\mathcal{M}}[h(t), g_i(h(t))]$
- if not yet defined $T_{\mathcal{M}'}[d'_1, d'_2]$ is $T_{\mathcal{M}}[d_1, d_2]$, where d_i is chosen such that $T_{\mathcal{M}}[d_i] = T_{\mathcal{M}'}[d_i]$ ($i = 1, 2$).

The undefined interpretations are not relevant for interpreting the theory and can be arbitrarily defined. The previous assignments are non-conflicting, i.e. 2-tables are never defined twice inconsistently.

Assume $\forall x \forall y \varphi(x, y)$ belongs to \mathcal{T} . Then $\mathcal{M}'_{x/d'_1, y/d'_2} \varphi(x, y) = \top$ since there exists d_1 and d_2 such that $T_{\mathcal{M}}[d_1, d_2] = T_{\mathcal{M}'}[d'_1, d'_2]$. It remains to prove that \mathcal{M}' is a model of every formula $\forall x \exists y \varphi_i(x, y)$ in \mathcal{T} , or equivalently, that for every $d \in D'$, $\mathcal{M}'_{x/d}$ is a model of $\exists y \varphi_i(x, y)$:

- if $d \in K$, $g_i(d) \in C \subseteq D'$, and $\mathcal{M}'_{x/d, y/g_i(d)}$ is a model of $\varphi_i(x, y)$;
- if $d \in C \setminus K$, if $g_i(d) \in C$ then $\mathcal{M}'_{x/d, y/g_i(d)}$ is a model of $\varphi_i(x, y)$;
- if $d \in C \setminus K$, if $g_i(d) \notin C$ then $T_{\mathcal{M}'}[d, (T_{\mathcal{M}}(g_i(d)), i, 0)] = T_{\mathcal{M}}[d, g_i(d)]$, and thus $\mathcal{M}'_{x/d, y/(T_{\mathcal{M}}(g_i(d)), i, 0)}$ is a model of $\varphi_i(x, y)$;
- if $d = (t, i, j) \in D' \setminus C$, if $g_i(h(t)) \in K$ then $T_{\mathcal{M}'}[(t, i, j), g_i(h(t))] = T_{\mathcal{M}}[h(t), g_i(h(t))]$, and thus $\mathcal{M}'_{x/d, y/g_i(h(t))}$ is a model of $\varphi_i(x, y)$;
- if $d = (t, i, j) \in D' \setminus C$, if $g_i(h(t)) \notin K$ then $T_{\mathcal{M}'}[(t, i, j), (T_{\mathcal{M}}(g_i(h(t))), i, (j+1) \bmod 3)] = T_{\mathcal{M}}[h(t), g_i(h(t))]$, and thus $\mathcal{M}'_{x/d, y/(T_{\mathcal{M}}(g_i(h(t))), i, (j+1) \bmod 3)}$ is a model of $\varphi_i(x, y)$.

Finally notice that $D \setminus K$ is necessarily non-empty if $|D| \geq 2^b + 1 + |A|$. In the process of building \mathcal{M}' , any element $(t, i, 0)$ may be duplicated, thus creating models of arbitrary size $\geq 3m 2^b + (m+1)|A|$ where m is the number of formulas of the form $\forall x \exists y \varphi(x, y)$ in \mathcal{T} . \square

Corollary 2.8. *There exists a computable function f such that, for any finite theory \mathcal{T} of FO^2 , if \mathcal{T} has a model of some cardinality $\geq f(\mathcal{T})$, then \mathcal{T} has models for every cardinality $\geq f(\mathcal{T})$.*

Corollary 2.9. *The class of finite theories of FO^2 has the finite model property.*

Corollary 2.10. *The satisfiability problem for finite theories of FO^2 is decidable.*

Corollary 2.11. *The spectrum of a finite theory of FO^2 can be computed and expressed as a finite set of naturals, or as the union of a finite set of naturals with the set of all the (finite or infinite) cardinalities greater than a natural. The finite theories of FO^2 are gentle.*

2.7 An example

Assume that one wants to study the satisfiability of the simple example given in the introduction:

$$\{a > 0, a < 2, a + b = 2, b > 0, A(f(a)), \neg C(f(b))\}$$

in the combination of the theories

$$\mathcal{T}_1 = \forall x [(A(x) \vee B(x)) \equiv (C(x) \vee D(x))]$$

and \mathcal{T}_2 , where \mathcal{T}_2 is itself the combination of the theory of uninterpreted functions and linear arithmetic over the integers. The theory \mathcal{T}_2 is decidable, and a decision procedure can be built using the standard Nelson-Oppen scheme since both components are stably infinite. The domain of the models of \mathcal{T}_2 is always the set of integers, thus all models have cardinality \aleph_0 . The theory \mathcal{T}_1 belongs to the BSR, Löwenheim, and

two-variables classes and is thus gentle.⁴ The third case of Theorem 2.3 is fulfilled. First, a separation is built, to produce the equisatisfiable problem $L_1 \cup L_2$ with

$$\begin{aligned} L_1 &= \{A(t), \neg C(u)\} \\ L_2 &= \{a > 0, a < 2, a + b = 2, b > 0, t = f(a), u = f(b)\}. \end{aligned}$$

The set $L_1 \cup L_2$ is $\mathcal{T}_1 \cup \mathcal{T}_2$ -satisfiable if and only if there exists a \mathcal{T}_1 -model \mathcal{M}_1 for L_1 and a \mathcal{T}_2 -model \mathcal{M}_2 for L_2 , such that \mathcal{M}_1 and \mathcal{M}_2 agree on which shared constant symbols (i.e. t and u) are equal, and agree on cardinalities (Theorem 2.1). The first requirement is fulfilled by checking every arrangement of the variables (here: $t = u$ or $t \neq u$): $\{t \neq u\} \cup L_2$ is \mathcal{T}_2 -unsatisfiable, but $\{t = u\} \cup L_1$ and $\{t = u\} \cup L_2$ are both satisfiable in their respective theory. It remains to check if it is possible for both models to agree on cardinalities. The theory of integer linear arithmetic only accepts models of cardinality \aleph_0 , therefore $L_1 \cup L_2$ is $\mathcal{T}_1 \cup \mathcal{T}_2$ -satisfiable if and only if $\mathcal{T}_1 \cup \{t = u\} \cup L_1$ has a model of cardinality \aleph_0 .

The theory $\mathcal{T}_1 \cup \{t = u\} \cup L_1$ uses only one quantified variable, four predicate symbols (A, B, C, D), and two constants (t, u). Using for instance the fact that this theory is a Löwenheim theory, one can use Theorem 2.4 to check if it has an infinite model. The theory contains two constants, at most one “nested” quantifier, and four unary predicates. If there is a model with cardinality 3×2^4 , then there is an infinite model. It can easily be showed that $\mathcal{T}_1 \cup \{t = u\} \cup L_1$ indeed accepts such a model with cardinality 48. Similar bounds exist for BSR and two-variable theories, but unfortunately they are also large compared to the size of this toy example.

There exists another criteria to check if a BSR theory has an infinite model. Indeed, a BSR theory with n variables has an infinite model if and only if it has a n -repetitive model (see the full version of the paper [10]). Checking if $\mathcal{T}_1 \cup \{t = u\} \cup L_1$ has a 1-repetitive model simply amounts to check if $\mathcal{T}_1 \cup \{t = u\} \cup L_1 \cup \{v \neq t, v \neq u\}$ is satisfiable.

As a final remark, notice that the example used in this section encodes the set of formulas

$$\{a > 0, a < 2, a + b = 2, b > 0, f(a) \in A, f(b) \notin C, A \cup B = C \cup D\}$$

in a language that combines integer linear arithmetic, uninterpreted function symbols, and elementary set-theoretic operations. One motivation for the work reported in this paper is indeed to augment the languages accepted by SMT solvers with certain operators on sets or relations, which can conveniently be represented by BSR theories over their characteristic predicates.

2.8 Conclusion

In this chapter, we observed that one can express completely the spectrum, i.e. the set of the cardinalities of the models, for Löwenheim theories, BSR theories, and finite theories in the two-variables fragment. We characterise those theories as *gentle*. Gentle theories can be combined with almost any decidable theory, including gentle theories, integer or real linear arithmetic, finite first-order theories, and some combinations of these.

It remains to develop algorithmic techniques to make this combination work in practice. The results presented here are prohibitively expensive, the finite cardinalities that guarantee the existence of an infinite model grow very rapidly with the size of the theories. In that sense, the combination scheme presented in this paper is really at the frontiers of combining decision procedures. It is certainly not practical to first extract all cardinalities of the gentle theories in the combination, just like, in the Nelson-Oppen combination scheme, it is not practical to check every arrangement one by one. Rather than guessing arrangements, SMT solvers use, among other techniques, equality propagation. Equality propagation can thus be seen as the negotiation of an arrangement. A practical way to agree on cardinality could also rely on negotiation. This negotiation would often be trivial, for instance if one theory puts very strong constraints on cardinalities, or if most theories are on the contrary very permissive. Another approach to handle the same classes of

⁴ \mathcal{T}_1 is also stably infinite, but we ignore this fact to illustrate the generic approach.

theories can be found in [24]: it consists in reducing each part of the separation to a formula in a common decidable language including Presbruger arithmetics; this approach has the drawback of being much more complex, but as a counterpart the language handled is in some aspects much more expressive.

Usually, SMT solvers implement a combination of a fixed set of theories, which are known *a priori*, and are also known to have the right properties according to cardinalities (typically, being stably infinite). Here, we show that every theory in the major well-known first-order decidable classes can be integrated in a combination. Since it can be shown that assertions over sets or relations and elementary set-theoretic operations like \cup , \cap , etc. just introduce one more BSR theory in the combination, the problem remains decidable even if this theory is not fixed a priori. We mainly target formal methods based on set theory such as B [1] and TLA⁺ [15]. We believe that the results in this paper can help automating the proof of some parts of the verification conditions, which often mix arithmetic symbols, uninterpreted functions, and elementary set theory. Verification conditions generated within those formal methods are usually small and should be most of the time within reach of a decision procedure, even if it is inefficient.

An interesting direction for further research is to investigate how to use the techniques embedded in state-of-the-art first order provers (for instance [3, 18, 19]) to efficiently handle the first-order theories within a combination of decision procedures.

References (of Chapter 2)

- [1] J.-R. Abrial. *The B-Book: Assigning Programs to Meanings*. Cambridge University Press, 1996.
- [2] M. Baaz, U. Egly, and A. Leitsch. Normal form transformations. In J. A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 5, pages 273–333. Elsevier Science B.V., 2001.
- [3] P. Baumgartner, A. Fuchs, and C. Tinelli. Implementing the Model Evolution Calculus. In S. Schulz, G. Sutcliffe, and T. Tammet, editors, *Special Issue of the International Journal of Artificial Intelligence Tools (IJAIT)*, volume 15 of *International Journal of Artificial Intelligence Tools*, 2005.
- [4] P. Bernays and M. Schönfinkel. Zum Entscheidungsproblem der mathematischen Logik. *Math. Annalen*, 99:342–372, 1928.
- [5] E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1997.
- [6] B. Dreben and W. D. Goldfarb. *The Decision Problem: Solvable Classes of Quantificational Formulas*. Addison-Wesley, Reading, Massachusetts, 1979.
- [7] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1995.
- [8] H. B. Enderton. *A Mathematical Introduction to Logic*. Academic Press, Inc., Orlando, Florida, 1972.
- [9] P. Fontaine. Combinations of theories and the Bernays-Schönfinkel-Ramsey class. In B. Beckert, editor, *4th International Verification Workshop - VERIFY'07, Bremen, 15/07/07-16/07/07*, July 2007.
- [10] P. Fontaine. Combinations of theories for decidable fragments of first-order logic, 2009. Available at <http://www.loria.fr/~fontaine/Fontaine12b.pdf>.
- [11] P. Fontaine and E. P. Gribomont. Decidability of invariant validation for parameterized systems. In H. Garavel and J. Hatcliff, editors, *Tools and Algorithms for Construction and Analysis of Systems (TACAS)*, volume 2619 of *Lecture Notes in Computer Science*, pages 97–112. Springer-Verlag, 2003.
- [12] P. Fontaine and E. P. Gribomont. Combining non-stably infinite, non-first order theories. In W. Ahrendt, P. Baumgartner, H. de Nivelle, S. Ranise, and C. Tinelli, editors, *Selected Papers from the Workshops on Disproving and the Second International Workshop on Pragmatics of Decision Procedures (PDPAR 2004)*, volume 125 of *Electronic Notes in Theoretical Computer Science*, pages 37–51, July 2005.
- [13] E. Grädel, P. G. Kolaitis, and M. Y. Vardi. On the decision problem for two-variable first-order logic. *The Bulletin of Symbolic Logic*, 3(1):53–69, 1997.
- [14] Y. Gurevich and S. Shelah. Spectra of monadic second-order formulas with one unary function. In *LICS '03: Proceedings of the 18th Annual IEEE Symposium on Logic in Computer Science*, pages 291–300, Washington, DC, USA, 2003. IEEE Computer Society.

- [15] L. Lamport. *Specifying Systems*. Addison-Wesley, Boston, Mass., 2002.
- [16] G. Nelson and D. C. Oppen. Simplifications by cooperating decision procedures. *ACM Transactions on Programming Languages and Systems*, 1(2):245–257, Oct. 1979.
- [17] F. P. Ramsey. On a Problem of Formal Logic. *Proceedings of the London Mathematical Society*, 30:264–286, 1930.
- [18] A. Riazanov and A. Voronkov. The design and implementation of Vampire. *AI Communications*, 15(2):91–110, 2002.
- [19] S. Schulz. System Abstract: E 0.61. In R. Goré, A. Leitsch, and T. Nipkow, editors, *International Joint Conference on Automated Reasoning (IJCAR)*, number 2083 in Lecture Notes in Artificial Intelligence, pages 370–375. Springer, 2001.
- [20] G. Sutcliffe and C. Suttner. The State of CASC. *AI Communications*, 19(1):35–48, 2006.
- [21] C. Tinelli and M. T. Harandi. A new correctness proof of the Nelson–Oppen combination procedure. In F. Baader and K. U. Schulz, editors, *Frontiers of Combining Systems (FroCoS)*, Applied Logic, pages 103–120. Kluwer Academic Publishers, Mar. 1996.
- [22] C. Tinelli and C. Ringeissen. Unions of non-disjoint theories and combinations of satisfiability procedures. *Theoretical Computer Science*, 290(1):291–353, Jan. 2003.
- [23] C. Tinelli and C. G. Zarba. Combining non-stably infinite theories. *Journal of Automated Reasoning*, 34(3):209–238, 2005.
- [24] T. Wies, R. Piskac, and V. Kuncak. Combining theories with shared set operations. In *Frontiers of Combining Systems (FroCoS)*, 2009. To appear in this volume.

Chapter 3

Non-Disjoint Combination: The case of a shared successor

3.1 Introduction

Satisfiability procedures for fragments of arithmetic and data structures such as lists, arrays, records, trees are at the core of many state-of-the-art verification tools, and their design and correct implementation is a hard task [BM07]. To overcome this difficulty, there is an obvious need for developing general and systematic methods to build and to combine decision procedures. Two important approaches have been investigated, based respectively on combination techniques and rewriting techniques.

The *combination approach* has been initiated in [NO79, Sho84] for the satisfiability problem. The motivation is to combine existing decision procedures known for some component theories in such a way that the combined procedure provides a decision procedure for the union of theories. The combination method introduced by Nelson-Oppen assumes that the component theories must have disjoint signatures, which drastically limits its applicability. A general non-disjoint combination method has been recently proposed in [Ghi04, GNZ08]. In this non-disjoint extension of Nelson-Oppen, the cooperation between the decision procedures relies on their capabilities of computing logical consequences built over the shared signature.

The *rewriting approach* allows us to flexibly build satisfiability procedures [ARR03, ABRS09b] based on a general calculus for automated deduction, namely the superposition calculus [NR01]. Hence, the implementation of a satisfiability procedures becomes easy by using an (almost) off-the-shelf theorem prover implementing the superposition calculus.

These two approaches are complementary for two main reasons. First, combination techniques allow us to incorporate theories which are difficult to handle using rewriting techniques, such as fragments of arithmetic. Second, rewriting techniques are of prime interest to design satisfiability procedures for finitely axiomatized theories modelling the standard data structures. Then, these rewriting-based satisfiability procedures can be efficiently plugged into the disjoint combination framework [KRRT05]. In some particular cases, the rewriting approach is an alternative to the combination approach by allowing us to build superposition-based satisfiability procedures for combinations of finitely axiomatized theories, including the theory of Integer Offsets [ABRS09b, BE08], but these theories must be over *disjoint* signatures.

In this paper, we show how to apply a superposition calculus to build decision procedures that can be plugged into the aforementioned *non-disjoint* combination framework. We focus on two particular shared theories of counter arithmetic, namely the theory of Integer Offsets and the theory of Increment, where a successor function satisfies some equational axioms like the injectivity and the acyclicity. We present a superposition calculus dedicated to both the theories and we show the soundness of this new calculus for several data structures enhanced with counting capabilities. The interest of combining counter arithmetic and uninterpreted functions in verification is advocated in [BLS02], where uninterpreted functions are used for abstracting data and the restricted form of arithmetic is sufficient to express counters and pointers,

thanks to the successor function s and 0 . For instance, we can consider (and combine) several theories of lists:

i) On the one hand, we can use the classical axiomatization of lists à la LISP, using cons , car , cdr operators, augmented with a length function ℓ defined as follows: $\ell(\text{cons}(e, x)) = s(\ell(x))$ and $\ell(\text{nil}) = 0$. In general, lists are over arbitrary elements but we may use also lists over integer elements.

ii) On the other hand, we can consider lists defined as records with two fields, the first one for the list itself, and the second one to store its length. Let us consider the operator rselect_i to access to the i -th field of a record, $\text{rcons}(r, e)$ denotes the record obtained by adding an element e to the list of r , and rnil denotes the record corresponding to the empty list, we have the following axiomatization:

$$\begin{array}{ll} \text{rselect}_1(\text{rcons}(r, e)) = \text{cons}(\text{rselect}_1(r), e) & \text{rselect}_1(\text{rnil}) = \text{nil} \\ \text{rselect}_2(\text{rcons}(r, e)) = s(\text{rselect}_2(r)) & \text{rselect}_2(\text{rnil}) = 0 \end{array}$$

This theory of lists can be seen as a refinement of the first theory in which one has a direct access to its “cardinality”. The combination framework presented in the paper can be applied to decide the satisfiability of ground formulae expressed in the union of these two theories of lists (provided both theories use distinct names for list operators). More generally, it allows us to decide the satisfiability problem in unions of some data structures and some fragments of arithmetic, including the theory of linear arithmetic over the rationals. To be effective, the combination framework relies on procedures able to compute the logical consequences over the shared signature that are exchanged in the main loop of the method. A major contribution of this paper is to build these procedures by using the given superposition calculus for data structures and some classical solving techniques for fragments of arithmetic: Gauss/Fourier-Motzkin elimination and Groebner bases computation.

The chapter is organized as follows. Section 2 introduces the basic definitions and notations. In Section 3, we present several data structures and the two shared theories of counter arithmetic we are interested in. We introduce in Section 4 a superposition calculus modulo both shared theories that can be turned into a decision procedure for the considered data structures. Section 5 describes the non-disjoint combination method that makes use of procedures for computing all the logical consequences over the shared signature to be exchanged. In Section 6, we explain how to compute the required consequences by using the superposition calculus developed for the data structures. In Section 7, we present two fragments of arithmetic. For both fragments of arithmetic, we show how to compute the needed consequences by using respectively Gauss/Fourier-Motzkin elimination and Groebner bases computation. Eventually, in Section 8, we conclude with some final remarks.

3.2 Preliminaries

We consider a many-sorted language. A *signature* Σ is a set of sorts, functions and predicate symbols (each endowed with the corresponding arity and sort). We assume that, for each sort s , the equality “ $=_s$ ” is a logical constant that does not occur in Σ and that is always interpreted as the identity relation over (the interpretation of) s ; moreover, as a notational convention, we will often omit the subscript and the symbol \bowtie will denote either $=$ or \neq . The signature obtained from Σ by adding a set \underline{a} of new constants (i.e., 0-ary function symbols, each of them again equipped with its sort) is denoted by $\Sigma^{\underline{a}}$ and named a *simple expansion* of Σ . Σ -atoms, Σ -literals, Σ -clauses, and Σ -formulae are defined in the usual way. A set of Σ -literals is called a Σ -constraint. Terms, literals, clauses and formulae are called *ground* whenever no variable appears in them; *sentences* are formulae in which free variables do not occur. Given a function symbol f , a f -rooted term is a term whose top-symbol is f .

From the semantic side, we have the standard notion of a Σ -structure \mathcal{M} : it consists of non-empty pairwise disjoint domains M_s for every sort s and a sort- and arity-matching interpretation \mathcal{I} of the function and predicate symbols from Σ . The truth of a Σ -formula in \mathcal{M} is defined in any of the standard ways. If $\Sigma_0 \subseteq \Sigma$ is a subsignature of Σ and if \mathcal{M} is a Σ -structure, the Σ_0 -reduct of \mathcal{M} is the Σ_0 -structure $\mathcal{M}|_{\Sigma_0}$ obtained from \mathcal{M} by forgetting the interpretation of the symbols from $\Sigma \setminus \Sigma_0$.

A collection of Σ -sentences is a Σ -theory, and a Σ -theory T admits *quantifier elimination* iff for every formula $\varphi(\underline{x})$ there is a quantifier-free formula (over the same free variables \underline{x}) $\varphi'(\underline{x})$ such that $T \models \varphi(\underline{x}) \leftrightarrow \varphi'(\underline{x})$. A Σ -theory T is *convex* if, for any set Γ of Σ -literals and any Σ -atoms $\alpha_1, \dots, \alpha_n$, we have that $T \models \Gamma \rightarrow (\alpha_1 \vee \dots \vee \alpha_n)$ implies $T \models \Gamma \rightarrow \alpha_i$ for some i . In the following, all considered theories are convex.

In this paper, we are concerned with the (*constraint*) *satisfiability problem* for a theory T , also called the T -satisfiability problem, which is the problem of deciding whether a Σ -constraint is satisfiable in a model of T (and, if so, we say that the constraint is T -satisfiable). Notice that a constraint may contain variables: since these variables may be equivalently replaced by free constants, we can reformulate the constraint satisfiability problem as the problem of deciding whether a finite conjunction of ground literals in a simply expanded signature Σ^a is true in a Σ^a -structure whose Σ -reduct is a model of T .

3.3 Theories

All the examples of data structures we are interested in involve also a successor function (denoted by s) that satisfies the axioms formalizing the properties of injectivity and acyclicity.

Theories of Counter Arithmetic.

$\boxed{T_S}$ denotes the theory of Increment defining the behaviour of the successor function s and the constant 0. T_S has the mono-sorted signature $\Sigma_S := \{0 : \text{NUM}, s : \text{NUM} \rightarrow \text{NUM}\}$, and it is axiomatized as follows:

$$\begin{aligned} \forall x, y \quad s(x) = s(y) &\rightarrow x = y \\ \forall x \quad x \neq s^n(x) &\text{ for all } n \text{ in } \mathbb{N}^+ \end{aligned}$$

$\boxed{T_I}$ denotes the theory of Integer Offsets defined as the union of T_S and $\{\forall x \ s(x) \neq 0\}$.

In what follows, we will generically denote with T_C , a theory for a counting operator, any theory in the set $\{T_S, T_I\}$.

We consider below some theories T corresponding to standard data structures and we focus on the constraint satisfiability problem for $T \cup T_S$ and for $T \cup T_I$.

Lists.

$\boxed{T_{LS}}$ is a theory of lists endowed with length. The many-sorted signature of T_{LS} is Σ_S plus the set of function symbols $\{\text{nil} : \text{LISTS}, \text{car} : \text{LISTS} \rightarrow \text{ELEM}, \text{cdr} : \text{LISTS} \rightarrow \text{LISTS}, \text{cons} : \text{ELEM} \times \text{LISTS} \rightarrow \text{LISTS}, \ell : \text{LISTS} \rightarrow \text{NUM}\}$ and the predicate symbol $\text{atom} : \text{LISTS}$. The axioms¹ of T_{LS} are:

$$\begin{aligned} \text{car}(\text{cons}(x, y)) &= x & \neg \text{atom}(x) &\rightarrow \text{cons}(\text{car}(x), \text{cdr}(x)) = x \\ \text{cdr}(\text{cons}(x, y)) &= y & \neg \text{atom}(\text{cons}(x, y)) & \\ \ell(\text{nil}) &= 0 & \text{atom}(\text{nil}) & \\ \ell(\text{cons}(x, y)) &= s(\ell(y)) & & \end{aligned}$$

The theory T'_{LS} corresponds to a slight variant of T_{LS} where the sort ELEM coincides with the sort NUM . It is important to notice that, by applying some standard reasoning (see, e.g., [NRR09d]), we can substitute T_{LS} (resp. T'_{LS}) with its subset of purely equational axioms, say T_{ELS} (resp. T'_{ELS}), and enrich the set of ground literals G we want to test for satisfiability modulo T_{LS} (resp. T'_{LS}), to a set of literals H in such a way that $T_{LS} \cup G$ is equisatisfiable to $T_{ELS} \cup H$ (resp. $T'_{LS} \cup G$ is equisatisfiable to $T'_{ELS} \cup H$). In this way we can still consider T_{LS} and T'_{LS} as *equational* theories.

¹Here and in the following, all the axioms should be considered as universally quantified.

Records.

T_{RS} denotes a theory of records with increment defined as follows. We consider records in which all the attribute identifiers are associated to the sort NUM or to sorts $ELEM_i$, and suppose we want to be able to increment by a unity every value of sort NUM stored into the record. To formalize this situation, the signature of T_{RS} is Σ_S plus the function symbols defined as follows. Let $Id = \{id_1, id_2, \dots, id_n\}$ be a set of attribute identifiers id_i associated to NUM or $ELEM_i$. Let NI be the set of elements $i \in \{1, \dots, n\}$ such that id_i is associated to NUM, and let \overline{NI} be $\{1, \dots, n\} \setminus NI$. Let us name REC the sort of records; for every attribute identifier id_1, id_2, \dots, id_n we have a couple of functions $rselect_i : REC \rightarrow NUM$ and $rstore_i : REC \times NUM \rightarrow REC$ for $i \in NI$; $rselect_i : REC \rightarrow ELEM_i$ and $rstore_i : REC \times ELEM_i \rightarrow REC$ for $i \in \overline{NI}$. Moreover, there is also an increment function $incr : REC \rightarrow REC$ that increments the elements of sort NUM. The axioms of T_{RS} are:

for every i, j such that $1 \leq i, j \leq n$, $i \neq j$

$$\begin{aligned} rselect_i(rstore_i(x, y)) &= y \\ rselect_j(rstore_i(x, y)) &= rselect_j(x) \\ \bigwedge_{i=1}^n (rselect_i(x) = rselect_i(y)) &\rightarrow x = y \end{aligned} \quad (\text{extensionality})$$

$$\begin{aligned} \text{for any } i \in NI, & \quad rselect_i(incr(x)) = s(rselect_i(x)) \\ \text{for any } i \in \overline{NI}, & \quad rselect_i(incr(x)) = rselect_i(x) \end{aligned}$$

The theory T'_{RS} denotes the particular case where all elements of records are of sort NUM, i.e. $\overline{NI} = \emptyset$. Moreover, following the same argument used in [ABRS09b], it is possible to check the satisfiability forgetting the extensionality axioms, so that, again, the theory of records can be still considered as an *equational* one.

Trees.

T_{BS} corresponds to a theory of binary trees endowed with size functions. The many-sorted signature of T_{BS} is Σ_S plus the set of function symbols $\{\text{bin} : ELEM \times TREES \times TREES \rightarrow TREES, \text{null} : TREES, \text{size}_L : TREES \rightarrow NUM, \text{size}_R : TREES \rightarrow NUM\}$. The axioms of T_{BS} are:

$$\begin{aligned} \text{size}_L(\text{null}) &= 0 & \text{size}_R(\text{null}) &= 0 \\ \text{size}_L(\text{bin}(e, t_1, t_2)) &= s(\text{size}_L(t_1)) & \text{size}_R(\text{bin}(e, t_1, t_2)) &= s(\text{size}_R(t_2)) \end{aligned}$$

The function size_L (resp. size_R) computes the length of the left (resp. right) branch of the input binary tree.

The theory T'_{BS} denotes the particular case where ELEM and NUM coincide.

Let DST be the set of theories $\{T_{LS}, T'_{LS}, T_{RS}, T'_{RS}, T_{BS}, T'_{BS}\}$.

3.4 Superposition Calculus as Decision Procedure

Recent literature has focused on the possibility of using the superposition calculus in order to decide the satisfiability of ground formulae modulo the theory of Integer Offsets and some disjoint extensions [ABRS09b, BE08]. Contrary to those papers, we are interested in a superposition-based calculus in order to cope with non-disjoint extensions of a theory endowed with a counting operator and that constraints the successor symbol by additional axioms.

Let us consider the axiomatization of a theory for a counting operator as formalized in Section 3.3, i.e. let us consider the axioms of T_S or the axioms of T_I . Our aim is to develop a calculus able to take into account those axioms into a framework based on superposition. Thus, let us consider a presentation of the superposition calculus specialized for reasoning over sets of literals, whose rules are described in Figures 3.1

and 3.2, augmented with the four additional rules over ground terms contained in Figure 3.3. The aim of the rules presented in the latter picture is to encode, directly into the calculus, the axioms of the theory for the counting operator. In the side conditions of rules, we use a reduction ordering \succ which is total on ground terms.

Superposition	$\frac{l[u'] = r \quad u = t}{(l[t] = r)\sigma}$	(i), (ii)
Paramodulation	$\frac{l[u'] \neq r \quad u = t}{(l[t] \neq r)\sigma}$	(i), (ii)
Reflection	$\frac{u' \neq u}{\perp}$	

where σ is the most general unifier of u and u' , u' is not a variable in *Superposition* and *Paramodulation*, L is a literal and the following hold:

(i) $u\sigma \not\prec t\sigma$, (ii) $l[u']\sigma \not\prec r\sigma$.

Figure 3.1: Expansion Inference Rules.

Definition 3.1. Let \mathcal{SP}_I be the calculus depicted in Figures 3.1, 3.2 and 3.3. Let \mathcal{SP}_S be the calculus obtained from \mathcal{SP}_I by removing the rule C1. Let T_C be the generic name for a theory chosen between T_I and T_S and, analogously, let \mathcal{SP}_C be the generic name for a calculus chosen between \mathcal{SP}_I and \mathcal{SP}_S .

Let us adapt the standard definition of *derivation* to the calculus we are interested in:

Definition 3.2. A derivation (δ) with respect to \mathcal{SP}_C is a (finite or infinite) sequence of sets of literals $S_1, S_2, S_3, \dots, S_i, \dots$ such that, for every i , it happens that:

- (i) S_{i+1} is obtained from S_i adding a literal obtained by the application of one of the rules in Figures 3.1, 3.2 and 3.3 (Figure 3.3 without C1 in case we are considering \mathcal{SP}_S) to some literals in S_i ;
- (ii) S_{i+1} is obtained from S_i removing a literal according to one of the rules in Figures 3.2 or to the rule R1 or R2.

If we focus on the rules of Simplification, R1 and R2, we notice that the effects of the application of any of these rules involve two steps in the derivation: in the former a new literal is added, and in the latter a literal is deleted.

If S is a set of literals, let GS be the set of all the ground instances of S . A literal L is said to be *redundant* with respect to a set of literals S if, for all the ground instances $L\sigma$ of L , it happens that $\{E \mid E \in GS \ \& \ E < L\sigma\} \models L\sigma$. We notice that in our derivations only redundant literals are deleted:

Fact If in a derivation S_{i+1} is equal to $S_i \setminus \{L\}$, then L is redundant with respect to S_i .

Proof. The claim above is well known if S_{i+1} is obtained from S_i applying one of the rules in Figure 3.2, and it follows immediately in the case we are applying R1 or R2. \square

So, as usual, we label with S_∞ the set of literals generated during a derivation δ (in symbols, $S_\infty = \bigcup_i S_i$), and with S_ω the set of persistent literals of δ : $S_\omega = \bigcup_i \bigcap_{j>i} S_j$. We adopt the standard definition for a rule π of the calculus being *redundant* with respect to a set of clauses S whenever, for every ground instance of the rule $\pi\sigma$, it happens that $\{E \mid E \in GS \ \& \ E < C_m\sigma\} \models D\sigma$, where $C_m\sigma$ is the maximal clause in the antecedent, and $D\sigma$ is the consequent of the rule. According to this definition, a derivation w.r.t. \mathcal{SP}_I is *fair* if, for every literal $L_1, L_2, \dots, L_m \in S_\omega$, every rule that has L_1, \dots, L_m as premises is redundant w.r.t. S_∞ .

Suppose now to take into account a fair derivation δ . We notice that, if a literal L is added at a certain step of the derivation, say S_{i+1} , then L is either a logical consequence of some literals in S_i , or it is a consequence of some literals in S_i and the axioms of the theory T_C .

Subsumption	$\frac{S \cup \{L, L'\}}{S \cup \{L\}}$	if $L\vartheta \equiv L'$ for some substitution ϑ
Simplification	$\frac{S \cup \{L[l'], l = r\}}{S \cup \{L[r\vartheta], l = r\}}$	if $l' \equiv l\vartheta$, $r\vartheta \prec l\vartheta$, and $(l\vartheta = r\vartheta) \prec L[l\vartheta]$
Deletion	$\frac{S \cup \{t = t\}}{S}$	

where L and L' are literals and S is a set of literals.

Figure 3.2: Contraction Inference Rules.

R1	$\frac{S \cup \{s(u) = s(v)\}}{S \cup \{u = v\}}$	if u and v are ground terms
R2	$\frac{S \cup \{s(u) = t, s(v) = t\}}{S \cup \{s(v) = t, u = v\}}$	if u, v and t are ground terms and $s(u) \succ t$, $s(v) \succ t$ and $u \succ v$
C1	$\frac{S \cup \{s(t) = 0\}}{S \cup \{s(t) = 0\} \cup \perp}$	if t is a ground term
C2	$\frac{S \cup \{s^n(t) = t\}}{S \cup \{s^n(t) = t\} \cup \perp}$	if t is a ground term and $n \in \mathbb{N}$

where S is a set of literals and \perp is the symbol for the inconsistency.

Figure 3.3: Ground reduction Inference Rules.

Proposition 3.1. *If the set of persistent literals S_ω contains \perp , then S_ω is unsatisfiable in any model of T_C .*

On the other hand, we notice that the reduction rules we can apply during the derivation satisfy the general requirements about the redundancy.

Proposition 3.2. *If the set of persistent literals S_ω does not contain \perp , then S_ω is satisfiable.*

What remains to show is that this calculus is *refutationally complete* with respect to the models of T_C (namely the structures in which the function s is injective, acyclic and such that 0 does not belong to the image of s in case T_C is T_I).

Remark 3.1. *Since the satisfiability of S_ω is equivalent to the satisfiability of S_∞ , and since the satisfiability of each step S_{i+1} in the derivation implies the satisfiability of S_i , we have in particular that if S_ω is satisfiable, then S_0 is satisfiable. Moreover, it is immediate to check that the unsatisfiability in the models of T_C of S_ω implies the unsatisfiability of S_0 in the same class of structures. So, in case it happens that the calculus described in Figures 3.1, 3.2 and 3.3 (Figure 3.3 minus C1 in case T_C is T_S) is refutationally complete, we can proceed as usual when considering procedures based on saturation methods: an initial set of literals S_0 will be satisfiable (in a model of T_C) if and only if its saturation S_ω does not contain \perp .*

3.4.1 Completeness

From now on, we assume that the ordering we consider when performing any application of \mathcal{SP}_C is *s-good*:

Definition 3.3. *We say that an ordering \succ over terms on a signature containing Σ_I is s-good whenever it satisfies the following requirements:*

- (i) \succ is a simplification ordering that is total on ground terms;
- (ii) 0 is minimal;

(iii) whenever two terms t_1 and t_2 are not \mathfrak{s} -rooted it happens that $\mathfrak{s}^{n_1}(t_1) \succ \mathfrak{s}^{n_2}(t_2)$ iff either $t_1 \succ t_2$ or ($t_1 \equiv t_2$ and n_1 is bigger than n_2).

A few remarks are in order. The requirement (i) is a very standard requirement over the ordering used when considering superposition-based calculi; requirement (iii) plays a key role in the proof of the refutational completeness of \mathcal{SP}_C ; finally, requirement (ii) is not strictly necessary when focusing on the refutational completeness of \mathcal{SP}_S . In any case, it is very easy to obtain an ordering that satisfies (i)-(iii): for example, it is sufficient to choose a LPO with an appropriate precedence on the symbols in the signature.

Proposition 3.3. *Assuming \mathfrak{s} -good ordering \succ over terms, if the set of persistent literals S_ω satisfies the following assumptions:*

- S_ω does not contain \perp ,
- S_ω does not contain equations whose maximal term is a variable of sort NUM, and \mathfrak{s} -rooted terms can be maximal just in ground equations.

then S_ω is satisfiable in a model of T_C .

Proof. By Proposition 3.2 we know that if \perp is not derived, then it is possible to build a model \mathcal{M} that satisfies all the literals contained in the limit of the derivation, S_ω . We can build such a model \mathcal{M} adapting to our case the so called *model-generation* technique [BG94]. By assumption, S_ω contains only literals, so \mathcal{M} will be built over the Herbrand universe relying upon a convergent rewriting system \mathcal{R} defined as follows: suppose that $\mathcal{R}_{\leq D}$ has already been defined for every ground literal D in GS_ω such that $D < L$, and let $\mathcal{R}_{< L} := \bigcup\{\mathcal{R}_{\leq D} \mid D \in GS_\omega \ \& \ D < L\}$. $\mathcal{R}_{\leq L}$ is equal to $\mathcal{R}_{< L} \cup \{l \rightarrow r\}$ if

$$\boxed{\text{i) } C \text{ is } l = r; \quad \text{ii) } l \text{ is in normal form with respect to } \mathcal{R}_{< C}; \quad \text{iii) } l > r.}$$

If any of the above condition is not satisfied, then $\mathcal{R}_{\leq L} := \mathcal{R}_{< L}$.

Thus, given two ground terms t_1 and t_2 , $\mathcal{M} \models t_1 = t_2$ if and only if $t_1 \downarrow_{\mathcal{R}} = t_2 \downarrow_{\mathcal{R}}$.

What remains to show is that the model so obtained is a structure that satisfies also the axioms of T_C .

In the following, we will call OGS_ω the set of all ground literals that are contained in S_ω . Notice that in OGS_ω both the left and the right side of the literals are inter-reduced. Indeed, by contradiction, suppose that $t = s$ is in OGS_ω and that there exists a rule $l \rightarrow r$ in \mathcal{R} that is able to reduce (say) t . $l \rightarrow r$ is a ground instance of some equation in S_ω , that means that the rule Simplification should have been applied, deleting thus $t = r$ in S_ω .

We have to prove now that in \mathcal{M} the followings are true: the axioms for the injectivity of (the interpretation of) \mathfrak{s} , its acyclicity and, in case T_C is T_I , the fact that (the interpretation of) 0 does not belong to the image of \mathfrak{s} .

1) $\forall x, y \ \mathfrak{s}(x) = \mathfrak{s}(y) \rightarrow x = y$

By contradiction, let us suppose that there exist two terms t_1 and t_2 such that $\mathfrak{s}(t_1) \downarrow_{\mathcal{R}} = \mathfrak{s}(t_2) \downarrow_{\mathcal{R}}$ but such that $t_1 \downarrow_{\mathcal{R}} \neq t_2 \downarrow_{\mathcal{R}}$. Without loss of generality, we can choose such a pair minimal with respect to the componentwise order over pairs induced by the ordering over the terms. By minimality and by the fact that \mathcal{R} is convergent, we can suppose that both t_1 and t_2 are irreducible. This latter assumption implies that there exist rules in \mathcal{R} such that $\mathfrak{s}(t_1) \rightarrow r \rightarrow^* z$ and $\mathfrak{s}(t_2) \rightarrow^* z$. Since the rule $\mathfrak{s}(t_1) \rightarrow r$ belongs to \mathcal{R} , the literal $\mathfrak{s}(t_1) = r$ belongs to GS_ω . More precisely, it belongs to OGS_ω , since in S_ω there is no non-ground literal that allows to rewrite terms whose root symbol is \mathfrak{s} . Now two cases are possible:

- either $\mathfrak{s}(t_2)$ is irreducible by \mathcal{R} . Then $\mathfrak{s}(t_2) \equiv z$, and, by the fact that r is irreducible, we obtain that $r \equiv \mathfrak{s}(t_2)$. Therefore, OGS_ω contains the equation $\mathfrak{s}(t_1) = \mathfrak{s}(t_2)$, that is impossible since an application of the rule R1 would have deleted it and replaced with $t_1 = t_2$;
- or there is a term r' and a rule $\mathfrak{s}(t_2) \rightarrow r'$ such that $\mathfrak{s}(t_2) \rightarrow r' \rightarrow^* z$. Again, the equation $\mathfrak{s}(t_2) = r'$ belongs to OGS_ω , implying that r' is irreducible. As a consequence $r \equiv r'$. Again, we have a contradiction because an application of the rule R2 would have been possible, deleting (say) $\mathfrak{s}(t_1) = r$ and substituting it with $t_1 = t_2$.

2) $s^n(t) \neq t$ for all the terms t and for all the naturals $n \in \mathbb{N}$

By contradiction, there exists a ground term t and a natural m such that $s^m(t) \downarrow_{\mathcal{R}} t$. We can choose t as the least ground term with that property; by minimality, we have that t is irreducible. Thus it happens that $s^m(t) \rightarrow r_1 \rightarrow^* t$, where $s^m(t)$ reduces to a term r_1 thanks to an application of a rule of the kind $s^{m_1}(t) \rightarrow r$ that comes from the equation $s^{m_1}(t) = r$ in OGS_ω , because only the equations that are in OGS_ω can reduce terms whose root symbol is s . Since t is irreducible, we must have $m_1 > 0$; moreover r is not s -rooted since, otherwise, R1 would be applied, deleting thus $s^{m_1}(t) = r$. Since r is not s -rooted and by the requirement over \succ , $s^{m_1}(t) \succ r$ implies that $t \succ r$. More in detail, w.l.o.g. we can suppose that $t \equiv s^n(t')$, where t' is not s -rooted. Due to the requirement over \succ and the fact that r is not s -rooted, we have for every k in \mathbb{N} , $s^k(t') \succ r$ iff $t' \succ r$. In particular, $t \equiv s^n(t') \succ r$ implies that $t' \succ r$. Now we know that $s^m(t) \rightarrow s^{m-m_1}(r) \rightarrow^* t$; but then $s^{m-m_1}(r) \succeq t \equiv s^n(t')$. Again, $s^{m-m_1}(r) \succeq s^n(t')$ iff either $r \succ t'$, that cannot be since $t' \succ r$, or $r \equiv t'$ and $m - m_1 \geq n$. But, if $r \equiv t'$, the equation $s^{m_1}(t) = r$ in OGS_ω becomes $s^{m_1+n}(t') = t'$, and, at this point, an application of the rule C2 would have added \perp .

At this point, the proof is concluded in case we consider \mathcal{SP}_S ; in case we consider \mathcal{SP}_I there is a last axiom to verify:

3) $\forall x s(x) \neq 0$

By contradiction again, let us suppose that there exists a ground term $s(t)$ such that $s(t) \downarrow_{\mathcal{R}} 0$. Again, we can choose such as t the least ground term that satisfies that property; that implies that t is irreducible. By the ordering over terms we have that 0 is irreducible, so the relation $s(t) \downarrow_{\mathcal{R}} 0$ can be rewritten as $s(t) \rightarrow r \rightarrow^* 0$ for some ground term r . The rule $s(t) \rightarrow r$ comes from the equation $s(t) = r$ that belongs to OGS_ω thus, since r is irreducible, $r \equiv 0$. But, if the equation $s(t) = 0$ had been in OGS_ω , then the application of the rule C1 would have added \perp .

□

Collecting all the results obtained so far, we can conclude that:

Theorem 3.1. *Let T be a Σ -theory presented as a finite set of unit clauses such that $\Sigma \supseteq \Sigma_S$, and assume to put an ordering over terms that is s -good. \mathcal{SP}_C induces a decision procedure for the constraint satisfiability problem w.r.t. $T \cup T_C$ if, for any set G of ground literals:*

- *the saturation of $Ax(T) \cup G$ w.r.t. \mathcal{SP}_C is finite,*
- *the saturation of $Ax(T) \cup G$ w.r.t. \mathcal{SP}_C does not contain non-ground equations whose maximal term is s -rooted, or equations whose maximal term is a variable of sort NUM.*

In [NRR09d, NRR09c], we have shown that conditions of Theorem 3.1 are satisfied for all theories in DST .

Corollary 3.1. *For any $T \in DST$, \mathcal{SP}_S induces a decision procedure for the constraint satisfiability problem w.r.t. $T \cup T_S$ and, analogously, for any $T \in DST$, \mathcal{SP}_I induces a decision procedure for the constraint satisfiability problem w.r.t. $T \cup T_I$*

3.5 Non-Disjoint Combination of Theories

We have presented some theories modeling data structures, and we have shown how an appropriate superposition calculus can provide a flexible decision procedure for the constraint satisfiability problem w.r.t. these theories. It would be interesting to reuse such procedures in order to obtain an algorithm able to cope with the constraint satisfiability problem w.r.t. the *union* of the theories in DST , and so, to this aim, we will rely on a general method for the combination of satisfiability procedures for unions of non-disjoint theories. This method extends the Nelson–Oppen combination method known for unions of signature-disjoint theories, and leads to the following result:

Theorem 3.2. [GNZ08] Consider two theories T_1, T_2 in signatures Σ_1, Σ_2 and suppose that:

1. both T_1, T_2 have decidable constraint satisfiability problem;
2. there exists some universal theory T_0 in the signature $\Sigma_1 \cap \Sigma_2$ such that:
 - T_1, T_2 are both T_0 -compatible;
 - T_0 is Noetherian;
 - T_1, T_2 are both effectively Noetherian extensions of T_0 .

Then the $(\Sigma_1 \cup \Sigma_2)$ -theory $T_1 \cup T_2$ also has decidable constraint satisfiability problem.

Let us motivate the requirements of Theorem 3.2. The disjointness assumption used by Nelson-Oppen is replaced by the condition that the component theories are both compatible with a common sub-theory. The requirement of T_0 -compatibility on T_1 and T_2 is the key condition in order to ensure the completeness of the combination procedure. The requirement of Noetherianity of T_0 is a sufficient hypothesis for the termination of the combination procedure. Finally, asking the theories to be “effectively Noetherian extensions” is a sufficient condition for designing a combination procedure that works à la Nelson-Oppen by exchanging logical consequences on the shared signature $\Sigma_1 \cup \Sigma_2$ until a fixpoint is reached. Let us enter now more into details.

Definition 3.4 (T_0 -compatibility). Let T be a theory in the signature Σ and let T_0 be a universal theory in a subsignature $\Sigma_0 \subseteq \Sigma$. We say that T is T_0 -compatible iff $T_0 \subseteq T$ and there is a Σ_0 -theory T_0^* such that

- (i) $T_0 \subseteq T_0^*$;
- (ii) T_0^* has quantifier elimination;
- (iii) every Σ_0 -constraint which is satisfiable in a model of T_0 is satisfiable also in a model of T_0^* ;
- (iv) every Σ -constraint which is satisfiable in a model of T is satisfiable also in a model of $T_0^* \cup T$.

These requirements are a generalization of the stable infiniteness requirement of the Nelson-Oppen combination procedure: in fact, if T_0 is the empty theory in the empty signature, T_0^* is the theory axiomatizing an infinite domain, so that (iii) holds trivially and (iv) is precisely stable infiniteness.

Example 3.1. T_I is a universal theory such that a needed T_I^* exists: indeed, if we add to T_I the axiom $\forall x(x \neq 0 \rightarrow \exists y x = s(y))$, we obtain a theory T_I^* that admits quantifier elimination (see, e.g. [End72]) and such that every constraint that is satisfiable in a model of T_I is satisfiable also in a model of T_I^* . To justify the last claim, it is sufficient to observe that each model of T_I can be extended to a model of T_I^* simply by adding recursively to each element different from (the interpretation of) 0 a “predecessor”. Since this operation does not affect the truth of any constraint, we obtain that the condition (iii) is satisfied.

Now, for any theory $T \supseteq T_I$ over a signature $\Sigma \supseteq \Sigma_I$, the T_I -compatibility requirement simply reduces to the following condition: every constraint Γ that is satisfiable in a model of T must be satisfiable also in a model of $T \cup \forall x(x \neq 0 \rightarrow \exists y x = s(y))$.

Example 3.2. Naturally, also T_S is a universal theory; moreover, adapting the quantifier elimination procedure useful in the case of the previous Example 3.1, it is easy to show that the theory $T_S^* := T_S \cup (\forall x \exists y x = s(y))$ admits quantifier elimination (req. (ii)), and satisfies again the requirement (iii): indeed, every constraint that is satisfiable in a model of T_S is satisfiable also in a model of T_S^* (again it is sufficient to endow each element in a model of T_S of a “predecessor”). Now, for any theory $T \supseteq T_S$ over a signature $\Sigma \supseteq \Sigma_S$, the requirement (iii) simply means that the satisfiability problem has the same answer in the models of T and in the models of $T \cup \{\forall x \exists y x = s(y)\}$.

Our combination method makes use of satisfiability procedures having the capability of deducing logical consequences over the shared signature. In order to ensure the termination when deducing those logical consequences, we rely on Noetherian theories. Intuitively, a theory is Noetherian if there exists only a finite number of atoms that are not redundant when reasoning modulo T_0 .

Definition 3.5 (Noetherian Theory). *A Σ_0 -theory T_0 is Noetherian if and only if for every finite set of free constants \underline{a} , every infinite ascending chain*

$$\Theta_1 \subseteq \Theta_2 \subseteq \dots \subseteq \Theta_n \subseteq \dots$$

of sets of ground $\Sigma_0^{\underline{a}}$ -atoms is eventually constant modulo T_0 , i.e. there is an n such that $T_0 \cup \Theta_n \models A$, for every natural number m and atom $A \in \Theta_m$.

Example 3.3. *Many examples of Noetherian theories come from the formalization of algebraic structures, but an interesting class of Noetherian theories consists in all the theories whose signature contains only constants and one unary function symbol [GNRZ07, Zuc08]. Thus, both the theories of Integer Offsets T_I and of Increment T_S enjoy this property.*

Let us consider now a theory $T \supseteq T_0$ with signatures $\Sigma \supseteq \Sigma_0$, and suppose we want to discover, given an arbitrary set of ground clauses Θ over Σ , a “complete set” of logical positive consequences of Θ over Σ_0 , formalized by the notion of T_0 -basis.

Definition 3.6 (T_0 -basis). *Given a finite set Θ of ground clauses (built out of symbols from Σ and possibly further free constants) and a finite set of free constants \underline{a} , a T_0 -basis modulo T for Θ w.r.t. \underline{a} is a set Δ of positive ground $\Sigma_0^{\underline{a}}$ -clauses such that*

- (i) $T \cup \Theta \models C$, for all $C \in \Delta$ and
- (ii) if $T \cup \Theta \models C$ then $T_0 \cup \Delta \models C$, for every positive ground $\Sigma_0^{\underline{a}}$ -clause C .

Notice that in the definition of a basis we are interested only in positive ground clauses: the exchange of positive information is sufficient to ensure the completeness of the resulting procedure. The interest in Noetherian theories lies in the fact that, for every set of Σ -clauses Θ and for every set \underline{a} of constants, a finite T_0 -basis for Θ w.r.t. \underline{a} always exists. Unfortunately, a basis for a Noetherian theory needs not to be computable; this motivates the following definition corresponding to the last hypothesis of Theorem 3.2:

Definition 3.7. *A theory T is an effectively Noetherian extension of T_0 if and only if T_0 is Noetherian and a T_0 -basis modulo T is computable for every set of literals and every finite set \underline{a} of free constants.*

In the following we will show how to discover theories that are effectively Noetherian extensions of the theory of Integer Offsets T_I and of the theory of Increment T_S . More in detail, we will see that the calculus \mathcal{SP}_C can compute T_C -bases in interesting cases.

3.6 Combining Data Structures

3.6.1 T_C -Compatibility

Being for a theory $T \supseteq T_I$ a T_I -compatible theory means that every constraint that is satisfiable w.r.t. T is satisfiable also in a model in which the axiom $\forall x(x \neq 0 \rightarrow \exists y x = s(y))$ holds. To see that actually it is the case for all the theories considered in Section 3.3, it is sufficient to check that any model of that theories can always be extended, if needed, by adding recursively to each element that is different from (the interpretation of) 0 its predecessor and, in case it is needed, modifying accordingly the remaining part of the structure; and to check that this enlargement does not affect the validity both of the constraints that are verified in the structure and of the axioms of the theory. More details can be found in [NRR09d]. On the other hand, being for a theory $T \supseteq T_S$ a T_S -compatible theory means that every constraint that is satisfiable w.r.t. T is satisfiable also in a model in which the axiom $\forall x \exists y x = s(y)$ holds. Again, it is easy to see that the T_S -compatibility requirements holds again for all the theories considered in Section 3.3, because the potential adjunction of predecessors to elements in the (interpretation of the) sort NUM does not affect the satisfiability of constraints.

3.6.2 Computing T_C -Bases for Data Structures

In this section we show that the superposition calculus \mathcal{SP}_C allows us to build T_C -bases modulo theories that are axiomatized by unit clauses.

Assume that $G(\underline{a}, \underline{b})$ is a set of ground literals over an expansion of Σ with the finite sets of fresh constants $\underline{a}, \underline{b}$. The theory $T \cup T_C$ is convex because it is a Horn theory. At this point, Proposition 3.4 shows how \mathcal{SP}_C can be used in order to derive T_C -bases.

Proposition 3.4. *Let S_ω be a finite saturation of $T \cup G(\underline{a}, \underline{b})$ w.r.t \mathcal{SP}_C using a \mathfrak{s} -good ordering over the terms in the signature $\Sigma \cup \{\underline{a}, \underline{b}\}$ such that (i) every term over the subsignature $\Sigma_{\mathfrak{s}}^{\underline{a}}$ is smaller than any term that contains a symbol in $(\Sigma \setminus \Sigma_{\mathfrak{s}}) \cup \{\underline{b}\}$, (ii) not containing \perp , and such that (iii) \mathfrak{s} -rooted terms can be maximal just in ground equations in S_ω and (iv) variables of sort NUM are never the maximal term in the equations. The set $\Delta(\underline{a})$ of all the ground equations over $\Sigma_{\mathfrak{s}}^{\underline{a}}$ in S_ω is a T_C -basis for T .*

Proof. Let us consider only the case of T_S . Suppose that $T \cup T_S \cup G(\underline{a}, \underline{b}) \models l = r$, being $l = r$ a ground equation over $\Sigma_{\mathfrak{s}}^{\underline{a}}$. We want to show that already $T_S \cup \Delta(\underline{a}) \models l = r$.

A saturation of $Ax(T) \cup G(\underline{a}, \underline{b}) \cup \{l \neq r\}$ under \mathcal{SP}_S is equal to a saturation of $S_\omega \cup \{l \neq r\}$. Since S_ω contains neither \perp , nor non-ground equations whose maximal term is \mathfrak{s} -rooted, nor equations whose maximal term is a variable of sort NUM, the only way to derive \perp is by reducing $l \neq r$ via equations from $\Delta(\underline{a})$: indeed, $l \neq r$ is defined on the signature $\mathfrak{s} \cup 0 \cup \underline{a}$ and, at this point, recalling also our choice of the reduction ordering, no equation in S_ω containing a symbol different from $\mathfrak{s}, 0, \underline{a}$, i.e. no equation out of $\Delta(\underline{a})$, can be used to rewrite a term on signature $\mathfrak{s}, 0, \underline{a}$.

Thus it follows that the saturation of $S_\omega \cup \{l \neq r\}$ will add only ground literals to S_ω , or \perp . In any case, the saturation still satisfies all the requirements in order to apply Theorem 3.1, and so we have the following chain of implications: $T \cup T_S \cup G(\underline{a}, \underline{b}) \models l = r$ iff the saturation of $Ax(T) \cup G(\underline{a}, \underline{b}) \cup \{l \neq r\}$ under \mathcal{SP}_S contains \perp , iff saturation of $\Delta(\underline{a}) \cup \{l \neq r\}$ under \mathcal{SP}_S contains \perp , iff $T_S \cup \Delta(\underline{a}) \models l = r$. The hypothesis that S_ω is finite guarantees that also $\Delta(\underline{a})$ is finite, i.e. $\Delta(\underline{a})$ is really a T_S -basis for T . The case for T_I follows from the same argument. \square

Corollary 3.2. *For any $T \in DST$, \mathcal{SP}_S is able to compute T_S -bases for $T \cup T_S$. Analogously, for any $T \in DST$, \mathcal{SP}_I is able to compute T_I -bases for $T \cup T_I$.*

3.6.3 Applying the Combination Method

In Section 3.3, we have pointed out that both the theory of Increment T_S and the theory of Integer Offsets are Noetherian, in Examples 3.1 and 3.2 and in Section 3.6.1 we have also shown that all the theories for the data structures we have introduced in Section 3.3 are T_I and T_S -compatible; Section 3.6.2 describes how to compute T_I and T_S -bases modulo the theories for the considered data structures. Hence, all the hypotheses of Theorem 3.2 are satisfied. As a consequence, we have the following decidability result.

Theorem 3.3. *For any Σ_1 -theory $T_1 \in DST$ and any Σ_2 -theory $T_2 \in DST$ such that $\Sigma_1 \cap \Sigma_2 = \Sigma_S$, $T_1 \cup T_I \cup T_2$ and $T_1 \cup T_S \cup T_2$ have decidable constraint satisfiability problems.*

3.7 Combining Fragments of Arithmetic

In the previous sections we have shown how to integrate a limited form of reasoning about arithmetic constraints when dealing with theories formalizing data structures. Nonetheless, it would be useful also to enlarge the expressiveness of the arithmetic in order to be able to reason also about, e.g., the comparison between two quantities ($\ell(a) < \ell(b)$), or non-linear relations ($\text{size}(t_1) \cdot \text{size}(t_2) = \text{size}(t_3)$). A very natural solution could be the cooperation between the procedures developed in this paper with the algorithms already available in the literature (and in the current practice) that are able to reason about fragments of arithmetic. Formally, we are required to deal with satisfiability problems in the union of theories. Moreover, since the formalization of the data structures involve some arithmetic symbols, namely 0 and \mathfrak{s} , we will have

to deal with a non-disjoint union. In the following, we will restrict our attention to two *convex* fragments of arithmetic: the linear rational arithmetic and the theory of \mathbb{Q} -algebras. In these theories, the interpretation of the successor function symbol \mathbf{s} coincides with the translation of one unit. For that reason, the axioms in the theory T_I that requires the 0 to be not the successor of anything doesn't make sense. Thus, from now on, we will consider, as theory for a counting operator, only T_S .

3.7.1 Theory of Linear Rational Arithmetic

A very natural extension of the theory of Increment is the linear arithmetic over the rationals. In more detail, let us fix the signature over the sort $\text{NUM } \Sigma_{\mathbb{Q}} := \{0, 1, +, -, \{f_q\}_{q \in \mathbb{Q}}, \mathbf{s}, <\}$, where $0, 1$ are constants, $-, f_q, \mathbf{s}$ are unary function symbols, $+$ is a binary one and $<$ is a binary predicate symbol. Let $T_{\mathbb{Q}}$ be the set of all the $\Sigma_{\mathbb{Q}}$ -sentences that are true in \mathbb{Q} considered as an ordered \mathbb{Q} -vector space, under the obvious convention that $0, 1, -, +, <$ are interpreted in their intended meaning, \mathbf{s} is the function that to each rational q associates the rational $q + 1$, and the f_q 's represent the external product of the \mathbb{Q} -vector spaces.

We can observe that in all the models of $T_{\mathbb{Q}}$ the function for the successor function symbol \mathbf{s} has an explicit definition using only the symbol 1 and $+$, since $T_{\mathbb{Q}} \models \forall x, y (y = \mathbf{s}(x) \leftrightarrow y = x + 1)$. This observation can be useful in order to rewrite all the formulae over $\Sigma_{\mathbb{Q}}$ discarding the symbol \mathbf{s} .

Decision Procedure

To build a $T_{\mathbb{Q}}$ -satisfiability procedure, a possible solution is to transform equalities and disequalities into inequalities and then to apply the Fourier-Motzkin elimination procedure for checking the satisfiability of the resulting set of inequalities. But for efficiency reasons, it is more convenient to keep the initial form of literals. Moreover, we are interested in a decision procedure enhanced with the capability of computing some particular entailed equalities. To this aim, we use the notions of solver and canonizer introduced by Shostak [Sho84]. A solver (*solve*) for $T_{\mathbb{Q}}$ computes a solved form (an idempotent substitution) of a set of equalities given by the Gauss elimination procedure, and a canonizer (*canon*) for $T_{\mathbb{Q}}$ is the classical normalization of arithmetic expressions (assuming an ordering over free constants). Any set of literals denoted by Γ is partitioned into a set of equalities $\Gamma^=$, a set of disequalities Γ^{\neq} and a set of inequalities Γ^{\leq} . Disequalities are processed in an easy way by using the fact that $T_{\mathbb{Q}}$ is convex. To handle inequalities, we use Fourier-Motzkin elimination to derive (1) unsatisfiable inequalities $q \leq 0$ where q is a strictly positive rational and (2) implicit equalities. The use of Fourier-Motzkin is justified by results stated in [LM92a, LM92b] for the case of the reals. These results hold also when the rationals are considered:

- An inequality $s \leq t$ in Γ^{\leq} is an implicit equality, which means $T_{\mathbb{Q}} \models \Gamma^{\leq} \rightarrow s = t$, iff it appears in a derivation computed by Fourier-Motzkin leading to the inequality $0 \leq 0$.
- If an equality is entailed by Γ^{\leq} , then it is entailed by the implicit equalities of Γ^{\leq} .

A $T_{\mathbb{Q}}$ -satisfiability procedure can be obtained by using the following architecture:

GE (Gauss) The solver is applied to compute a solved form γ for the set of equalities. The substitution γ is applied to disequalities and inequalities.

DH (Disequalities Handler) If there exists some disequality $s \neq t$ such that $\text{canon}(s) = \text{canon}(t)$, then the unsatisfiability is reported.

FME (Fourier-Motzkin) Provided that Gauss does not apply, Fourier-Motzkin is used to derive unsatisfiable (ground) inequalities or implicit equalities. Fourier-Motzkin eliminates successively the variables occurring in the inequalities. Eventually, if it derives an inequality $q \leq 0$ such that q is a strictly positive rational, then the unsatisfiability is reported. If it derives an inequality $0 \leq 0$, then the implicit equalities used in the derivation of $0 \leq 0$ are sent to **GE**.

This procedure is terminating because neither **GE** nor **FME** introduces new variables and **GE** strictly decreases the number of unsolved variables.

3.7.2 Theory of \mathbb{Q} -Algebras

We can consider now another extension of the theory of Increment, namely we can see T_S as subtheory of the theory of (non-degenerate) \mathbb{Q} -algebras. More in detail, we fix as a signature $\Sigma_{\mathbb{Q}\text{-alg}}$ the set consisting of the constants $0, 1$, the two binary function symbols $+, \times$, the unary function symbols $-$ and the \mathbb{Q} -indexed family of unary function symbols f_q . As a notational convention, of course we use the infix notation for $+$ and write qv, v_1v_2 for $f_q(v), \times(v_1, v_2)$, respectively. The theory of \mathbb{Q} -algebras, denoted by $T_{\mathbb{Q}\text{-alg}}$, is described using the axioms of abelian groups for $+$ (stating the associativity, the commutativity of $+$, the existence of the inverse $-v$ for each v and the fact that 0 is the unity of $+$), the axioms of abelian monoids for \times (asserting the associativity and the commutativity of \times , and that 1 is the unity of \times), the fact that 0 is different from 1 and the other six axioms relating the behaviour of $+$ and \times

for every q, q_1 and q_2 in \mathbb{Q}

$$\forall x, y, z (x + y)z = xz + yz \quad (3.1)$$

$$\forall x, y q(x + y) = qx + qy \quad (3.2)$$

$$\forall x (q_1 \oplus q_2)x = q_1x + q_2x \quad (3.3)$$

$$\forall x (q_1 \cdot q_2)x = q_1(q_2x) \quad (3.4)$$

$$\forall x 1_{\mathbb{Q}}x = x \quad (3.5)$$

$$\forall x, y q(xy) = x(qy) \quad (3.6)$$

where \oplus and \cdot are respectively the sum and multiplication operation in \mathbb{Q} , and $1_{\mathbb{Q}}$ is the multiplicative unit of \mathbb{Q} .

Again, the symbol s admits in $T_{\mathbb{Q}\text{-alg}}$ the explicit definition as in the previous example: we have $T_{\mathbb{Q}\text{-alg}} \models \forall x, y (y = s(x) \leftrightarrow y = x + 1)$. Injectivity of s is guaranteed by the group structure (i.e., it holds $T_{\mathbb{Q}\text{-alg}} \models \forall x, y (x + 1 = y + 1 \leftrightarrow x = y)$), and the acyclicity of s is guaranteed by the fact that $1 \neq 0$ and by the axiom (3.4).

Decision Procedure

Given a set \underline{a} of n fresh constants, the ground atoms over $\Sigma_{\mathbb{Q}\text{-alg}}^{\underline{a}}$ are polynomials in at most n indeterminates whose normalized representation is of the kind $p(\underline{a}) = 0$. Given the convexity of $T_{\mathbb{Q}\text{-alg}}$, the constraint satisfiability problem in $T_{\mathbb{Q}\text{-alg}}$ is just the problem of deciding whether an equation $p(\underline{a}) = 0$ is a logical consequence of a finite number of equations $\{p_1(\underline{a}) = 0, \dots, p_m(\underline{a}) = 0\}$. Since the polynomial ring $\mathbb{Q}[a_1, \dots, a_n]$ is the free \mathbb{Q} -algebra over n generators, this problem is equivalent to the membership of the polynomial p to the ideal $\langle p_1, \dots, p_m \rangle$ generated by the polynomials p_1, \dots, p_m . The Buchberger algorithm solves the problem by computing the Groebner basis associated to the ideal $\langle p_1, \dots, p_m \rangle$ [Buc76].

3.7.3 Computing T_S -bases for Fragments of Arithmetic

In this section we will show how to derive T_S -bases when we consider the theory $T_{\mathbb{Q}}$ and the theory $T_{\mathbb{Q}\text{-alg}}$. First of all, we recall that both $T_{\mathbb{Q}}$ and $T_{\mathbb{Q}\text{-alg}}$ are convex theories and we will see that, in both the cases, given a set of atoms, the respective decision procedures are able to derive a “representative set” of the linear equalities, i.e. equalities in the shape $q_1x_1 + \dots + q_nx_n = 0$, $q_i \in \mathbb{Q}$, that are implied.

Our aim is, at that point, to describe a procedure that, given a generic constraint over $\Sigma_{\mathbb{Q}}$ (resp. $\Sigma_{\mathbb{Q}\text{-alg}}$), say Γ , is able to derive a set of ground atoms over an expansion $\Sigma_S^{\underline{a}}$, say Δ , such that $T_{\mathbb{Q}} \cup \Gamma \models \Delta$ (resp. $T_{\mathbb{Q}\text{-alg}} \cup \Gamma \models \Delta$), and such that, for every $\Sigma_S^{\underline{a}}$ -atom e it holds that $T_{\mathbb{Q}} \cup \Gamma \models e$ iff $T_S \cup \Delta \models e$ (resp. $T_{\mathbb{Q}\text{-alg}} \cup \Gamma \models e$ iff $T_S \cup \Delta \models e$).

We start by recalling that all the literals in Γ that are not atoms, i.e. that are the negation of some atoms, are irrelevant in order to compute the set Δ .

Lemma 3.1. *Let T be a convex theory, let P be a set of atoms, let N be a set of negative literals, i.e. a set consisting only of negations of atoms, and let α be an atom. If $P \wedge N$ is T -satisfiable, it holds $T \models (P \wedge N) \rightarrow \alpha$ iff $T \models P \rightarrow \alpha$.*

Let us now introduce $T_{\mathbb{Q}}^=$, the theory of the (non-degenerate) \mathbb{Q} -vector spaces. This theory is a subtheory of both $T_{\mathbb{Q}}$ and $T_{\mathbb{Q}\text{-alg}}$, it is built on the signature $\Sigma_{\mathbb{Q}^=} := \{0, 1, +, -, \{f_q\}_{q \in \mathbb{Q}}, \mathbf{s}\}$, and it is ruled by the axioms of abelian groups over $+$, the requirement that $1 \neq 0$ and the axioms (3.3) – (3.6) in Section 3.7.2. Again, we require the relationship $\forall x, y (y = \mathbf{s}(x) \leftrightarrow y = x + 1)$ to hold in all the structures that are models of $T_{\mathbb{Q}}^=$.

Lemma 3.2. *Let $\underline{a}, \underline{b}$ be two sets of free constants such that $\underline{a} \subseteq \underline{b}$. Given a $T_{\mathbb{Q}}^=$ -satisfiable set of linear equalities P over the signature $\Sigma_{\mathbb{Q}^=}^{\underline{b}}$, it is possible to derive a T_S -basis modulo $T_{\mathbb{Q}}^=$ for P w.r.t. \underline{a} .*

Proof. Any $\Sigma_S^{\underline{a}}$ -equation is of the form $\mathbf{s}^{n_1}(a_1) = \mathbf{s}^{n_2}(a_2)$ for some n_1, n_2 in \mathbb{N} and for some a_1, a_2 in $\underline{a} \cup \{0\}$. Due to the injectivity axiom for the \mathbf{s} function symbol, any equation can be equivalently rewritten in the form $a_1 = \mathbf{s}^{n_2 - n_1}(a_2)$ whenever $n_2 \geq n_1$, or in the form $\mathbf{s}^{n_1 - n_2}(a_1) = a_2$ whenever $n_1 > n_2$. Thus, for any couple of constants a_1, a_2 in \underline{a} , it is sufficient to detect if $T_{\mathbb{Q}}^= \cup P \models a_1 = a_2 + n$ for some $n \in \mathbb{N}$, or if $T_{\mathbb{Q}}^= \cup P \models a_2 = a_1 + n$ (for some $n \in \mathbb{N}$, again). While running the Gauss elimination procedure on P and computing $\sigma = \text{solve}(P)$, we obtain:

$$T_{\mathbb{Q}}^= \cup P \models a_1 = a_2 + n \text{ iff } \text{canon}(a_1\sigma - a_2\sigma) = n$$

Let Δ be the set of $\Sigma_S^{\underline{a}}$ -equations obtained by collecting all the equations of the form $a_1 = \mathbf{s}^n(a_2)$ for which $\text{canon}(a_1\sigma - a_2\sigma) = n$. The properties (i) and (ii) of Definition 3.6 for T_S -bases are straightforward. \square

Example 3.4. *Consider $P = \{a_1 - 1 = a_3 + 1, 2b_2 + a_3 = b_2 + 2b_1 + b_2, a_2 - 1 = 2a_3 - 2b_1\}$. A solved form for P is given by $\sigma = \{a_1 \mapsto 2b_1 + 2, a_2 \mapsto 2b_1 + 1, a_3 \mapsto 2b_1\}$. By using the method given in the proof of Lemma 3.2, we can derive that $a_1 = \mathbf{s}^2(a_3), a_2 = \mathbf{s}(a_3)$ and these equalities define a T_S -basis modulo $T_{\mathbb{Q}}^=$ for P w.r.t. $\{a_1, a_2, a_3\}$.*

The $T_{\mathbb{Q}}$ case.

While running over a constraint Γ the procedure presented in Section 3.7.1, we have already pointed out that, if Γ is satisfiable, the procedure halts returning a conjunction of the form $\hat{\sigma} \wedge \Phi^{\neq} \wedge \Phi^{\leq}$, where $\hat{\sigma}$ is a set of linear equalities that, thanks to the results in [LM92a, LM92b] and Lemma 3.1, satisfies the following two properties:

1. $T_{\mathbb{Q}} \cup \Gamma \models \hat{\sigma}$;
2. if e is a linear equality such that $T_{\mathbb{Q}} \cup \Gamma \models e$, then $T_{\mathbb{Q}}^= \cup \hat{\sigma} \models e$.

The $T_{\mathbb{Q}\text{-alg}}$ case.

In Section 3.7.2, we have recalled that the satisfiability problem modulo $T_{\mathbb{Q}\text{-alg}}$ can be solved by running the Buchberger algorithm for computing the Groebner basis associated to a set Γ of polynomials. Actually, the Groebner basis computation can be considered as a way to obtain a confluent and terminating rewriting system for deciding the universal fragment of the theory of \mathbb{Q} -algebras. In [Nic07], it is shown how a little tuning on the ordering of the rules in the term rewriting system is able to produce in the final Groebner basis associated to Γ a set, say P , of linear polynomials such that:

1. $T_{\mathbb{Q}\text{-alg}} \cup \Gamma \models P$;
2. if e is a linear polynomial such that $T_{\mathbb{Q}\text{-alg}} \cup \Gamma \models e$, then $T_{\mathbb{Q}}^= \cup P \models e$.

Proposition 3.5. *Let $\underline{a}, \underline{b}$ be two sets of free constants such that $\underline{a} \subseteq \underline{b}$. Given a constraint Γ over the signature $\Sigma_{\mathbb{Q}}^{\underline{b}}$ (resp. $\Sigma_{\mathbb{Q}-alg}^{\underline{b}}, (\Sigma_{\mathbb{Q}} \cup \Sigma_{\mathbb{Q}-alg})^{\underline{b}}$), it is possible to compute a T_S -basis modulo $T_{\mathbb{Q}}$ (resp. $T_{\mathbb{Q}-alg}, T_{\mathbb{Q}} \cup T_{\mathbb{Q}-alg}$) for Γ w.r.t. \underline{a} .*

Proof. $T_{\mathbb{Q}}$ Let us run the decision procedure for testing the satisfiability of Γ w.r.t. $T_{\mathbb{Q}}$. If it reports unsatisfiability, then the T_S -basis is simply $\{\perp\}$. Otherwise collect all the equalities (say $\hat{\sigma}$) as described in Section 3.7.1, and apply on $\hat{\sigma}$ the procedure described in Lemma 3.2. Thanks to the properties 1. and 2. recalled in the paragraph above about the $T_{\mathbb{Q}}$ case, the set Δ is a T_S -basis. Indeed, since $T_{\mathbb{Q}} \cup \Gamma \models \hat{\sigma}$ and $T_{\mathbb{Q}}^{\equiv} \cup \hat{\sigma} \models \Delta$, it follows (i) $T_{\mathbb{Q}} \cup \Gamma \models \Delta$ (recall that $T_{\mathbb{Q}}^{\equiv} \subset T_{\mathbb{Q}}$); moreover it holds the following chain of implications: for any e s.t. $T_{\mathbb{Q}} \cup \Gamma \models e$, then the set of equalities $\hat{\sigma}$ derived using Fourier-Motzkin and Gauss elimination procedures is such that $T_{\mathbb{Q}}^{\equiv} \cup \hat{\sigma} \models e$, and thus, by Lemma 3.2, also (ii) $T_S \cup \Delta \models e$.

$T_{\mathbb{Q}-alg}$ The case to compute a T_S -basis for Γ is analogous, taking into account the fact that the set P of representative linear polynomials is given by running the Buchberger algorithm as described in [Nic07], and again the properties 1. and 2. in the paragraph above about the $T_{\mathbb{Q}-alg}$ case.

$T_{\mathbb{Q}} \cup T_{\mathbb{Q}-alg}$ The proofs of Lemma 3.2 and the two cases above make clear that, once we are able to guarantee the derivation of a set of linear equalities P that satisfy the properties of the kind 1. and 2., we are also able to compute T_S -bases. Since it is possible to isolate such a set w.r.t. $T_{\mathbb{Q}}$ and $T_{\mathbb{Q}-alg}$, it is possible to apply Theorem 1.3.12 in [Zuc08] to derive, given a set of literals Γ over $(\Sigma_{\mathbb{Q}} \cup \Sigma_{\mathbb{Q}-alg})^{\underline{b}}$, a set P' of linear equalities such that, again,

1. $T_{\mathbb{Q}} \cup T_{\mathbb{Q}-alg} \cup \Gamma \models P'$;
2. if e is a linear equality such that $T_{\mathbb{Q}} \cup T_{\mathbb{Q}-alg} \cup \Gamma \models e$, then $T_{\mathbb{Q}}^{\equiv} \cup P' \models e$.

At this point, it is immediate to apply again Lemma 3.2 to compute a T_S -basis modulo $T_{\mathbb{Q}} \cup T_{\mathbb{Q}-alg}$. \square

3.7.4 Applying the Combination Method

At the beginning of Section 3.7, we have pointed out that the theory of Increment T_S is Noetherian and that it can be “enlarged” to T_S^* , which admits quantifier elimination and behaves the same w.r.t. the satisfiability of constraints; moreover we have also shown that $T_{\mathbb{Q}}, T_{\mathbb{Q}-alg}$ and all the theories for the data structures we have introduced in Section 3.3 are T_S -compatible. Since the T_S -compatibility is a modular property (cf. Proposition 4.4 in [Ghi04]), also $T_{\mathbb{Q}} \cup T_{\mathbb{Q}-alg}$ ² is T_S -compatible. Moreover, in Section 3.6.2 we have shown how to compute T_S -bases modulo the theories for the considered data structures, and in Section 3.7.3 we have shown how to compute T_S -bases modulo the three fragments of arithmetic we are taking into account. Hence, all the hypotheses of Theorem 3.2 are satisfied.

Theorem 3.4. *For any Σ_1 -theory $T_1 \in DST$ and any Σ_2 -theory $T_2 \in \{T_{\mathbb{Q}}, T_{\mathbb{Q}-alg}, T_{\mathbb{Q}} \cup T_{\mathbb{Q}-alg}\}$ such that $\Sigma_1 \cap \Sigma_2 = \Sigma_S$, $T_1 \cup T_S \cup T_2$ has a decidable constraint satisfiability problem.*

3.8 Conclusion

We have shown how to instantiate the non-disjoint extension of the Nelson-Oppen method in order to combine various data structures with some fragments of arithmetic. Our approach allows us to consider arbitrary arithmetic constraints even if the shared signature is restricted to the successor function. We have focused on fragments over the rationals. The fragments over the integers are more problematic, since first of all

²The satisfiability problem w.r.t. $T_{\mathbb{Q}} \cup T_{\mathbb{Q}-alg}$ can be decided through an appropriate application of Theorem 3.2: for the details we refer to [Nic07].

the convexity is lost, and secondly it is not so clear how to extract from the existing decision procedures the sets that are representative of the logical consequences involving only the successor function symbol. This is a problem left for future work. Another interesting issue is to study how to handle more complex connecting axioms between the data structure and the arithmetic, and to try to enlarge the shared signature. In [NRR09d], the shared theory is a more precise approximation of the theory of integers, but on the other hand there is no integration of standard techniques for reasoning about arithmetic. In [NRR09a], we show how to combine data structures sharing the theory of abelian groups. In a similar way to what is investigated here, it would be interesting to study the combination of a data structure with some fragments of arithmetic when the shared theory is the one of abelian groups. More generally, an important issue is the capability of dealing with a non-convex data structure such as the arrays. To go beyond theories axiomatized by unit clauses, we would need to adapt our superposition calculus.

Chapter 4

Non-Disjoint Combination: The case of a shared addition

4.1 Introduction

Decision procedures are the basic engines of the verification tools used to check the satisfiability of formulae modulo background theories, which may include axiomatizations of standard data-types such lists, arrays, bit-vectors, etc. Nowadays, there is a growing interest in applying theorem provers to construct decision procedures for theories of interest in verification [ARR03, ABR09a, dMB08, BE07]. The problem of incorporating some reasoning modulo arithmetic properties inside theorem provers is particularly challenging. Many works are concerned with the problem of building-in certain equational axioms, starting from the seminal contributions by Plotkin [Pl72] and by Peterson and Stickel [PS81]. The case of Associativity-Commutativity has been extensively investigated since it appears in many equational theories, and among them, the theory of abelian groups is a very good candidate as fragment of arithmetic. Recently, the standard superposition calculus [NR01] has been extended to a superposition calculus modulo the built-in theory of abelian groups [GN04]. This work paves the way for the application of a superposition calculus modulo a fragment of arithmetic to build decision procedures of practical interest in verification. However, practical problems are often expressed in a combination of theories where the fragment of arithmetic is shared by all the other theories involved. In this case the classical Nelson-Oppen combination method cannot be applied since the theories share some arithmetic operators. An extension of the Nelson-Oppen combination method to the non-disjoint case has been proposed in [GNZ08]. This non-disjoint combination framework has been recently applied to the theory of Integer Offsets [NRR09d]. In this paper, our aim is to consider a more expressive fragment by studying the case of abelian groups.

The contributions of the paper are twofold. First, we show that abelian groups satisfy all the properties required to prove the completeness, the termination and the effectiveness of the non-disjoint extension of the Nelson-Oppen combination method. To prove the completeness, we show the existence of an extension of the theory of abelian groups having quantifier elimination and that behaves the same w.r.t. the satisfiability of constraints. Second, we identify a class of theories that extend the theory of abelian groups and for which a simplified constraint-free (but many-sorted) version of the superposition calculus introduced in [GN04] is proved to be complete. This superposition calculus allows us to obtain effective decision procedures that can be plugged into the non-disjoint extension of the Nelson-Oppen combination method.

This chapter is organized as follows. In Section 2, we show some very useful properties in order to use the theory of abelian groups, namely AG , in the non-disjoint combination framework, especially we prove the quantifier elimination of a theory that is an extension of AG . In Section 3, we present a calculus modulo AG . In Section 4, we show its refutational completeness and we study how this calculus may lead to combinable decision procedures. Examples are given in Section 5. We conclude with some final remarks in Section 6. Most of the proofs can be found in [NRR09b].

4.2 The Theory of Abelian Groups

In this section we focus on some properties that are particularly useful when trying to apply Theorem 3.2 to a combination of theories sharing AG .

AG rules the behaviour of the binary function symbol $+$, of the unary function symbol $-$ and of the constant 0 . More precisely, $\Sigma_{AG} := \{0 : AG, - : AG \rightarrow AG, + : AG \times AG \rightarrow AG\}$, and AG is axiomatized as follows:

$$\begin{array}{ll} \forall x, y, z \quad (x + y) + z = x + (y + z) & \forall x, y \quad x + y = y + x \\ \forall x \quad x + 0 = x & \forall x \quad x + (-x) = 0 \end{array}$$

From now on, given an expansion of Σ_{AG} , a generic term of sort AG will be written as $n_1 t_1 + \dots + n_k t_k$, where t_i is a term whose root symbol is different both from $+$ and $-$, $t_1 - t_2$ is a shortening for $t_1 + (-t_2)$, and $n_i t_i$ is a shortening for $t_i + \dots + t_i$ (n_i)-times if n_i is a positive integer, or $-t_i - \dots - t_i$ ($-n_i$)-times if n_i is negative.

4.2.1 Unification in Abelian Groups

We will consider a superposition calculus using unification in AG with free symbols, which is known to be finitary [BJSS90]. In the following, we restrict ourselves to particular AG -unification problems with free symbols in which no variables of sort AG occur. By using a straightforward many-sorted extension of the Baader-Schulz combination procedure [BS96], one can show that an AG -equality checker is sufficient to construct a complete set of unifiers for these particular AG -unification problems with free symbols. Moreover, the following holds:

Lemma 4.1. *Let Γ be a AG -unification problem with free symbols in which no variable of sort AG occurs, and let $CSU_{AG}(\Gamma)$ be a complete set of AG -unifiers of Γ . For any $\mu \in CSU_{AG}(\Gamma)$, we have that 1.) $VRan(\mu) \subseteq Var(\Gamma)$, and that, 2.) for any AG -unifier σ of Γ such that $Dom(\sigma) = Var(\Gamma)$, there exists $\mu \in CSU_{AG}(\Gamma)$ such that $\sigma =_{AG} \mu(\sigma|_{VRan(\mu)})$.*

4.2.2 Noetherianity of Abelian Groups

Let us start by proving the Noetherianity of AG ; the problem of discovering effective Noetherian extensions of AG will be addressed in Section 4.4.1, after the introduction of an appropriate superposition calculus (Section 4.3).

Proposition 4.1. *AG is Noetherian.*

Proof. Note that any equation is AG -equivalent to $(\#) \sum_{i=1}^k n_i a_i = \sum_{j=1}^h m_j b_j$, where a_i, b_j are free constants in $\underline{a} \cup \underline{b}$ and n_i, m_j are positive integers, so we can restrict ourselves to chains of sets of equations of the kind $(\#)$. Theorem 3.11 in [Che86] shows that AC is Noetherian, where AC is the theory of an associative and commutative $+$ (thus $\Sigma_{AC} = \{+\}$). From the definition of Noetherianity it follows that, if T is a Noetherian Σ -theory, any other Σ -theory T' such that $T \subseteq T'$ is Noetherian, too. Clearly, the set of sentences over Σ_{AC} implied by AG extends AC ; hence any ascending chain of sets of equations of the kind $(\#)$ is eventually constant modulo AG , too. □

In order to apply Theorem 3.2 to a combination of theories that share AG , we need to find an extension of AG that admits quantifier elimination and such that any constraint is satisfiable w.r.t. such an extension iff it is already satisfiable w.r.t. AG . A first, natural candidate would be AG itself. Unfortunately it is not the case: more precisely, it is known that AG cannot admit quantifier elimination (Theorem A.1.4 in [Hod93]). On the other hand, it is possible to find an extension AG^* with the required properties: AG^* is the theory of divisible abelian groups with infinitely many elements of each finite order.

4.2.3 An Extension of Abelian Groups having Quantifier Elimination

Let $D_n := \forall x \exists y ny = x$, let $O_n(x) := nx = 0$ and let $L_{m,n} := \exists y_1, y_2, \dots, y_m \bigwedge_{i \neq j} y_i \neq y_j \wedge \bigwedge_{i=1}^m O_n(y_i)$, for $n, m \in \mathbb{N}$. D_n expresses the fact that each element is divisible by n , $O_n(x)$ expresses that the element x is of order n , and $L_{m,n}$ expresses the fact that there exist at least m elements of order n . The theory AG^* of divisible abelian groups with infinitely many elements of each finite order can be thus axiomatized by $AG \cup \{D_n\}_{n>1} \cup \{L_{m,n}\}_{m>0, n>1}$.

Now, instead of showing directly that AG^* admits quantifier elimination and satisfies exactly the same constraints that are satisfiable w.r.t. AG , we rely on a different approach. Let us start by introducing some more notions about structures and their properties. Given a Σ -structure $\mathcal{M} = (M, \mathcal{I})$, let Σ^M be the signature where we add to Σ constant symbols m for each element of M . The *diagram* $\Delta(\mathcal{M})$ of \mathcal{M} is the set of all the ground Σ^M -literals that are true in \mathcal{M} . Given two Σ -structures $\mathcal{M} = (M, \mathcal{I})$ and $\mathcal{N} = (N, \mathcal{J})$, a Σ -*embedding* (or, simply, an embedding) between \mathcal{M} and \mathcal{N} is a mapping $\mu : M \rightarrow N$ among the corresponding support sets satisfying, for all the Σ^M -atoms ψ , the condition $\mathcal{M} \models \psi$ iff $\mathcal{N} \models \psi$ (here \mathcal{M} is regarded as a Σ^M -structure, by interpreting each additional constant $a \in M$ into itself, and \mathcal{N} is regarded as a Σ^M -structure by interpreting each additional constant $a \in M$ into $\mu(a)$). If $M \subseteq N$ and if the embedding $\mu : M \rightarrow N$ is just the identity inclusion $M \subseteq N$, we say that \mathcal{M} is a *substructure* of \mathcal{N} . If it happens that, given three models of T : $\mathcal{A}, \mathcal{M}, \mathcal{N}$ and two embeddings $f : \mathcal{A} \rightarrow \mathcal{M}$ and $g : \mathcal{A} \rightarrow \mathcal{N}$, there always exists another model of T , \mathcal{H} , and two embeddings $h : \mathcal{M} \rightarrow \mathcal{H}$ and $k : \mathcal{N} \rightarrow \mathcal{H}$ such that the composition $f \circ h = g \circ k$, we say that T has the *amalgamation property*. Finally if, given a Σ -theory T and a model \mathcal{M} for T , it happens that, for each Σ -sentence ψ , $\mathcal{M} \models \psi$ if and only if $T \models \psi$, then we say that T is a *complete* theory.

Now, in [Hod93], Exercise 8 page 380, it is stated that AG^* is the so-called *model companion* of the theory AG , meaning that (i) for each model \mathcal{M} of AG^* the theory $AG^* \cup \Delta(\mathcal{M})$ is a complete theory, (ii) every constraint that is satisfiable in a model of AG is satisfiable in a model of AG^* and (iii) every constraint that is satisfiable in a model of AG^* is satisfiable in a model of AG (of course, since $AG \subset AG^*$, condition (iii) gets trivial, but we report here for sake of completeness). At this point, since the behaviour of AG and AG^* is the same w.r.t. the satisfiability of constraints, the only condition that remains to be verified is that AG^* admits quantifier elimination. But:

Theorem 4.1. *AG has the amalgamation property.*

Corollary 4.1. *AG* admits quantifier elimination.*

Proof. In [ES71] it is shown that, if T is a universal theory and T^* is a model-companion of T , then the following are equivalent:

- (i) T^* has quantifier elimination;
- (ii) T has the amalgamation property.

Since AG has the amalgamation property, and AG^* is the model-companion of AG , we have that AG^* has quantifier elimination. \square

4.3 A Calculus for Abelian Groups

In [GN04] the authors give a superposition calculus in which the reasoning about elements of an abelian group is completely built-in. Our aim is to elaborate that calculus so that it provides a decision procedure for the satisfiability problem modulo theories modelling interesting data structures and extending AG . More precisely, we want to produce a calculus able to check the satisfiability, in the models of AG , of clauses in the shape $Ax(T) \cup G$, where $Ax(T)$ is a set of unit clauses, not necessarily ground, formalizing the behaviour of some data structure, and G is a set of ground literals. To that purpose, we eliminate the constraints from the calculus and we use a many-sorted language that extends the signature of the theory of abelian groups

Σ_{AG} by additional function symbols. Moreover, we will adopt from now on the following assumption: we will consider only

unit clauses with no occurrence of variables of sort AG. (*)

Let us start to see more in detail the notations and the concepts used in the rules of the calculus.

First of all, we will reason over terms modulo an *AG*-rewriting system: quoting [GN04], the system R_{AG} consists of the rules (i) $x+0 \rightarrow 0$, (ii) $-x+x \rightarrow 0$, (iii) $-(-x) \rightarrow 0$, (iv) $-0 \rightarrow 0$, (v) $-(x+y) \rightarrow (-x)+(-y)$. Moreover, rewriting w.r.t. R_{AG} is considered *modulo AC*, namely the associativity and the commutativity of the $+$, thus, when rewriting $\rightarrow_{R_{AG}}$, we mean the relation $=_{AC} \rightarrow_{R_{AG}} =_{AC}$. The normal form of a term t w.r.t. R_{AG} will be often written as *AG-nf*(t), and two terms t_1 and t_2 are equal modulo *AG* iff $AG\text{-nf}(t_1) =_{AC} AG\text{-nf}(t_2)$. Accordingly, we say that a substitution σ is in *AG*-normal form whenever all the terms occurring in the codomain of σ are in *AG*-normal form.

Moreover, we will consider an order \succ over terms that is total, well-founded, strict on ground terms and such that 1. \succ is *AC*-compatible, meaning that $s' =_{AC} s \succ t =_{AC} t'$ implies $s' \succ t'$, 2. \succ orients all the rules of R_{AG} , meaning that $l\sigma \succ r\sigma$ for every rule $l \rightarrow r$ of R_{AG} and all the grounding substitutions σ ; 3. \succ is monotonic on ground terms, meaning that for all ground terms $s, t, u, u[s]_p \succ u[t]_p$ whenever $s \succ t$. An ordering satisfying all the requirements above can be easily obtained considering an RPO ordering with a total precedence \succ_Σ on the symbols of the signature Σ such that $f \succ_\Sigma - \succ_\Sigma + \succ_\Sigma 0$ for all symbols f in Σ and such that all the symbols have a lexicographic status, except $+$, whose status is multiset (see [Der82], where, in order to compare two terms, the arity of $+$ is considered variable, but always greater than 1).

As a last convention, with a little abuse of notation, we will call *summand* any term whose root symbol is different from both $+$ and $-$, notwithstanding its sort. In this way a *generic* term can be written in the shape $n_1t_1 + \dots + n_kt_k$ (if it is of sort different from AG, it simply boils down to t_1).

Now, we are ready to describe the calculus. We will rely basically on three rules, *Direct AG-superposition*, *Inverse AG-superposition* and *Reflection*, and, as in [GN04], we will apply the rules only in case the premises satisfy certain conditions as explained in the following. Moreover, from now on we assume that all the literals will be eagerly maintained in *AG*-normal form, meaning that they will be maintained as (dis)equations between terms in *AG*-normal form.

Orientation for the left premises of direct *AG*-superposition Let $l = r$ be an equation; if it is on the sort *AG*, then it can be equivalently rewritten into $e = 0$. Thus the term e is a term of the form $n_1t_1 + n_2t_2 + \dots + n_pt_p$, where the t_i are non variable distinct summands, and the n_i 's are non zero integers. By splitting the summands into two disjoint sets, the equation $e = 0$ can be rewritten as $n_1t_1 + \dots + n_kt_k = -n_{k+1}t_{k+1} - \dots - n_pt_p$. In the following, we will call any equation over AG in that form an *orientation* for $e = 0$. If $l = r$ is an equation over a sort different from AG, then an *orientation* of $l = r$ will be either $l = r$ or $r = l$.

Orientation for the left premises of inverse *AG*-superposition Let $e = 0$ be an equation over the sort AG. If e or $-e$ is a term of the form $s + e'$, where s is a summand that occurs positively and e' is a generic term, then $-s = e'$ is an *inverse orientation* for $e = 0$.

Splitting of the right premises for direct *AG*-superposition Let t be a non-variable subterm of either r or s in the literal $r \bowtie s$; moreover, if s is of sort AG, we can freely assume that s is 0. If t is of sort AG, we ask that t is not immediately under $+$ nor under $-$, and that the root of t is different from $-$. Thus, we can imagine that t is of the kind $n_1s_1 + \dots + n_ps_p + t'$, where all s_i are distinct summands, all n_i are positive integers and t' contains only negative summands. In this case, $t_1 + t_2$ is a *splitting* for t if t_1 is a term of the form $k_1s_1 + \dots + k_ps_p$, where $0 \leq k_i \leq n_i$, and t_2 is $(n_1 - k_1)s_1 + \dots + (n_p - k_p)s_p + t'$. If t is not over the sort AG, then the only splitting admissible for t is t itself.

Splitting of the right premises for inverse AG -superposition Let t be a non variable subterm of either r or s in the literal $r \bowtie s$; moreover, if s is of sort AG , we can freely assume that s is 0. Let t be of sort AG , and let t be not immediately below $+$ nor $-$. If t is of the form $-s + t'$, where s is a summand, then $t_1 + t_2$ is an *inverse splitting* for t if t_1 is $-s$ and t_2 is t' .

AG -superposition rules In the left premise $l = r$ of the direct AG -superposition rule, it is assumed that $l = r$ is an orientation of the literal. Similarly, in the right premise, $D[t_1 + t_2]_p$ denotes that $D|_p$ is a non-variable term that is not immediately below $+$ or $-$ with a splitting $t_1 + t_2$. Similarly, in the inverse AG -superposition rule, $l = r$ and $D|_p$ denote inverse orientation and splitting, respectively. The inference system, denoted by \mathcal{SP}_{AG} , is made of the following rules:

Direct AG -superposition	$\frac{l = r \quad D[t_1 + t_2]_p}{(D[r + t_2]_p)\mu_i}$	(i)
Inverse AG -superposition	$\frac{l = r \quad D[t_1 + t_2]_p}{(D[r + t_2]_p)\mu_i}$	(ii)
Reflection	$\frac{u' \neq u}{\square}$	(iii)

The condition (i) is that μ_i is a most general solution of the AG -unification problem $l =_{AG} t_1$; moreover the inference has to be performed whenever there is a ground instantiation of μ_i, θ , s.t., if $nu = s$ is the AG -normal form of $(l = r)\mu_i\theta$ and $D'[nu]_q$ is the AG -normal form of $(D[t_1 + t_2]_p)\mu_i\theta$ in which, in position q , nu appears as subterm, then (a) $u \succ s$, (b) nu appears as subterm of the maximal term in D' .

The condition (ii) is that μ_i is a most general solution of the AG -unification problem $l =_{AG} t_1$; moreover the inference has to be performed whenever there is a ground instantiation of μ_i, θ , s.t., if $-u = s$ is the AG -normal form of $(l = r)\mu_i\theta$ and $D'[-u]_q$ is the AG -normal form of $(D[t_1 + t_2]_p)\mu_i\theta$ in which, in position q , $-u$ appears as subterm, then (a) either u is the maximal summand in s or $u \succ s$, (b) $-u$ appears as subterm of the maximal term in D' .

The condition (iii) is that the AG -unification problem $u =_{AG} u'$ has a solution (and \square is the syntactic convention for the empty clause).

Moreover, we assume that, after each inference step, the newly-derived literal is normalized modulo AG .

We point out that, thanks to Lemma 4.1(1.) and to our assumption (*), at any step of a saturation no variable of sort AG is introduced, thus the resulting saturated set will consist of literals in which no variable of sort AG occurs. Moreover, we can note that the conditions on the inferences are, in general, far from being obvious to check. However, for our purposes, we will often perform inferences involving at least one ground literal. In that case, verifying all the conditions becomes easier.

4.4 Refutational Completeness of \mathcal{SP}_{AG}

In order to prove the refutational completeness of the calculus presented above, we will adapt the model generation technique presented in [GN04]. The idea behind this technique consists in associating to any saturated set of literals that does not contain the empty clause a model of terms identified modulo a rewriting system, the latter being built according to some of the equations in the saturated set. Even if in our calculus no constrained literal will appear, in order to build the model of terms we will rely only on ground instances of the literals in the saturation that are *irreducible*. Moving from [GN04] and extending to the many-sorted case, we say that:

Definition 4.1. *An equation $s = t$ is in one-sided form whenever, (a) if s and t are of sort AG , the equation is in the form $e = 0$, and e is in AG -normal form; (b) if s and t are not of sort AG , both s and t are in AG -normal form.*

Whereas an equation over a sort different from AG has a unique one-sided form, an equation over the sort AG has two AG -equivalent one-sided forms, but in what follows it does not matter which of the two will be

considered. Thus, from now on, when we will refer to equations, we will always assume that the equations are in one-sided form.

Definition 4.2. Let s be a term, σ be a grounding substitution such that both σ and s are in AG-normal form. Moreover, let R be a ground term rewriting system. We will say that the $\text{maxred}_R(s\sigma)$ is

- 0, if $\text{AG-nf}(s\sigma)$ is R -irreducible;
- $\max PS$, where PS is the following set of terms (ordered w.r.t. \succ):
 $PS := \{u \text{ is a summand} \mid \text{for some term } v \text{ and some } n \text{ in } \mathbb{Z}, \text{AG-nf}(s\sigma) \text{ is of the form } nu + v \text{ and } nu \text{ is } R\text{-reducible}\}.$

Definition 4.3.¹ Let s be a term in which no variable of sort AG occurs, let σ be a grounding substitution such that both s and σ are in AG-normal form, and let R be a ground TRS. The pair (s, σ) is irreducible w.r.t. R whenever:

- $\text{AG-nf}(s\sigma)$ is R -irreducible, or
- if $\text{AG-nf}(s\sigma)$ is R -reducible, let u be the $\text{maxred}_R(s\sigma)$. Then, (s, σ) is irreducible if s is not a variable and, for each term of the form $t = f(t_1, \dots, t_n)$ such that s is of the form $t + v$ or $-t + v$ or t and such that $u \succeq \text{AG-nf}(t\sigma)$, each (t_i, σ) is irreducible.

If L is a literal, the pair (L, σ) is irreducible w.r.t. R :

- if L is an (dis)equation whose one-sided form is of the form $e \bowtie 0$, then (e, σ) is irreducible w.r.t. R ;
- if L is an (dis)equation whose one-sided form is of the form $s \bowtie t$, both (s, σ) and (t, σ) are irreducible w.r.t. R .

Before going on with the description of all the ingredients that are needed in order to show the completeness of the calculus, we want to point out a property that will be useful in the following.

Proposition 4.2. Let s be a term in which no variable of sort AG occurs, let σ be a grounding substitution such that both s and σ are in AG-normal form, and let R be a ground TRS such that (s, σ) is irreducible w.r.t. R . Moreover, let $\sigma =_{\text{AG}} \mu\pi$, where π is another grounding substitution in AG-normal form and μ is a substitution that does not have variables of sort AG in its range. Then $(s\mu, \pi)$ is still irreducible w.r.t. R .

To extract, from a given set of ground literals, a term rewriting system, we first of all transform all the equations in reductive normal form (see [GN04]):

Definition 4.4. A ground literal $s \bowtie t$ in AG-normal form is in reductive form whenever s is of the form nu , t is the form $n_1v_1 + \dots + n_kv_k$ and $n > 0$, n_i are non-zero integers, u and v_i are summands with $u \succ v_i$.

Of course, if s and t are of sort different from AG, the definition above simply says that $s \succ t$; moreover, it is always possible, given an equation, to obtain an equivalent one in reductive normal form. Now, a term rewriting system is obtained as follows:

Definition 4.5. Let S be a set of literals, let L be an equation with a ground instance $L\sigma$, let G be the reductive form of $L\sigma$: $G \equiv nu = r$. Then G generates the rule $nu \rightarrow r$ if the following conditions are satisfied:

- (i) $(R_G \cup \text{AG}) \not\models G$;
- (ii) $u \succ r$;
- (iii) nu is R_G -irreducible;

¹Here we are adapting, in case of absence of variables of sort AG, the definition of recursive irreducibility of [GN04], but in our context the two notions of recursive irreducibility and irreducibility are collapsing.

(iv) (L, σ) is irreducible w.r.t. R_G .

where R_G is the set of rules generated by the reductive forms of the ground instances of S that are smaller than G w.r.t. \succ . Moreover, if $n > 1$, then also the rule $-u \rightarrow (n-1)u - r$ is generated.

Now, exactly as in [GN04], we associate to a generic set of literals saturated under the rules of our calculus and that does not contain the empty clause, S , a structure I that is an AG -model for S . I is the equality Herbrand interpretation defined as the congruence on ground terms generated by $R_S \cup AG$, where R_S is the set of rules generated by S according to Definition 4.5. Since we are in a many-sorted context, the domain of I consists of different sets, one for each sort; since the rewriting rules in $R_S \cup AG$ are sort-preserving, the congruence on the ground terms is well-defined. Applying the same kind of arguments used to prove Lemma 10 in [GN04], we have that $R_S \cup AG$ is terminating and confluent, and it still holds that $I \models s = t$ iff $s \rightarrow_{R_S \cup R_{AG}}^* \tau \leftarrow_{R_S \cup R_{AG}}^* t$ for some term τ . To show that I is really an AG -model for S , we can prove the following lemma:

Lemma 4.2. *Let S be the closure under the calculus of a set of literals S_0 , and let us assume that the empty clause does not belong to S . Let I be the model of terms derived from S as described above, and let $Ir_{R_S}(S)$ be the set of ground instances $L\sigma$ of L in S such that (L, σ) is irreducible w.r.t. R_S . Then (1) $I \models Ir_{R_S}(S)$ implies that $I \models S$, and (2) $I \models Ir_{R_S}(S)$.*

From the lemma above, it follows immediately:

Theorem 4.2. *The calculus \mathcal{SP}_{AG} is refutational complete for any set of literals that do not contain variables of sort AG .*

4.4.1 Computing AG -bases

Let us go back, for the moment, to Theorem 3.2, and especially to the condition that states that, in order to apply a combination procedure à la Nelson-Oppen to a pair of theories T_1 and T_2 sharing AG , we have to ensure that T_1 and T_2 are effectively Noetherian extensions of AG , i.e. we have to ensure the capability of computing AG -bases for T_1 and T_2 . Let us suppose that T_i is a Σ_i -theory (for $i = 1, 2$) whose set of axioms is described by a finite number of unit clauses.

Now, for $i = 1, 2$, let Γ_i be a set of ground literals over an expansion of $\Sigma_i \supseteq \Sigma_{AG}$ with the finite sets of fresh constants $\underline{a}, \underline{b}_i$, and suppose to perform a saturation w.r.t. \mathcal{SP}_{AG} adopting an RPO ordering in which the precedence is $f \succ a \succ - \succ + \succ 0$ for every function symbol f in $\Sigma_i^{\underline{b}_i}$ different from $+, -, 0$, every constant a in \underline{a} and that all the symbols have a lexicographic status, except $+$, whose status is multiset. Relying on the refutational completeness of \mathcal{SP}_{AG} , Proposition 4.3 shows how \mathcal{SP}_{AG} can be used in order to ensure that T_1 and T_2 are effectively Noetherian extensions of AG :

Proposition 4.3. *Let S be a finite saturation of $T_i \cup \Gamma_i$ w.r.t. \mathcal{SP}_{AG} not containing the empty clause and suppose that, in every equation $e = 0$ containing at least one of the constants a in \underline{a} as summand, the maximal summand is not unifiable with any other summand in e . Then the set Δ_i of all the ground equations over $\Sigma_{AG}^{\underline{a}}$ in S is an AG -basis for T_i w.r.t. \underline{a} ($i = 1, 2$).*

4.5 Some Examples

Theorem 4.2 guarantees that \mathcal{SP}_{AG} is refutational complete, thus, if we want to turn it into a decision procedure for the constraint satisfiability problem w.r.t. a theory of the kind $T \cup AG$, it is sufficient to prove that any saturation under the rules of \mathcal{SP}_{AG} of a set of ground literals and the axioms of T is finite. Let us show some examples in which this is actually the case.

Lists with Length The theory of lists with length can be seen as the union of the theories $T_L \cup T_\ell \cup AG$, with T_L being the theory of lists and T_ℓ being the theory that axiomatizes the behaviour of the function for the length; more formally:

T_L has the many-sorted signature of the theory of lists: Σ_L is the set of function symbols $\{\text{nil} : \text{LISTS}, \text{car} : \text{LISTS} \rightarrow \text{ELEM}, \text{cdr} : \text{LISTS} \rightarrow \text{LISTS}, \text{cons} : \text{ELEM} \times \text{LISTS} \rightarrow \text{LISTS}\}$ plus the predicate symbol $\text{atom} : \text{LISTS}$, and it is axiomatized as follows:

$$\begin{array}{ll} \forall x, y \text{ car}(\text{cons}(x, y)) = x & \forall x \neg \text{atom}(x) \Rightarrow \text{cons}(\text{car}(x), \text{cdr}(x)) = x \\ \forall x, y \text{ cdr}(\text{cons}(x, y)) = y & \forall x, y \neg \text{atom}(\text{cons}(x, y)) \\ & \text{atom}(\text{nil}) \end{array}$$

T_ℓ is the theory that gives the axioms for the function $\ell : \text{LISTS} \rightarrow \text{AG}$ and the constant $(1 : \text{AG})$: $\ell(\text{nil}) = 0; \forall x, y \ell(\text{cons}(x, y)) = \ell(y) + 1; 1 \neq 0$

Applying some standard reasoning (see, e.g. [NRR09d]), we can substitute T_L with the set of the purely equational axioms of T_L , say $T_{L'}$, and enrich a bit the set of literals G to a set of literals G' in such a way $T_L \cup T_\ell \cup AG \cup G$ is equisatisfiable to $T_{L'} \cup T_\ell \cup AG \cup G'$. Let us choose as ordering an RPO with a total precedence \succ such that all the symbols have a lexicographic status, except $+$, whose status is multiset, and such that it respects the following requirements: (a) $\text{cons} \succ \text{cdr} \succ \text{car} \succ c \succ e \succ \ell$ for every constant c of sort LISTS and every constant e of sort ELEM ; (b) $\ell \succ g \succ - \succ + \succ 0$ for every constant g of sort AG .

Proposition 4.4. *For any set G of ground literals, any saturation of $T_{L'} \cup T_\ell \cup G'$ w.r.t. \mathcal{SP}_{AG} is finite.*

Trees with Size Let us reason about trees and their size. We can propose a formalization in which we need to reason about a theory of the kind $T_T \cup T_{\text{size}} \cup AG$, where T_T rules the behaviour of the trees and T_{size} constraints the behaviour of a function that returns the number of nodes of a tree. Thus we have:

T_T has the mono-sorted signature $\Sigma_T := \{\mathcal{E} : \text{TREES}, \text{binL} : \text{TREES} \rightarrow \text{TREES}, \text{binR} : \text{TREES} \rightarrow \text{TREES}, \text{bin} : \text{TREES} \times \text{TREES} \rightarrow \text{TREES}\}$, and it is axiomatized as follows:

$$\begin{array}{ll} \forall x, y \text{ binL}(\text{bin}(x, y)) = x & \forall x, y \text{ binR}(\text{bin}(x, y)) = y \\ \forall x \text{ bin}(\text{binL}(x), \text{binR}(x)) = x & \end{array}$$

T_{size} is the theory that gives the axioms for the function $\text{size} : \text{TREES} \rightarrow \text{AG}$: $\text{size}(\mathcal{E}) = 0; \forall x, y \text{ size}(\text{bin}(x, y)) = \text{size}(x) + \text{size}(y)$

Let us now put as ordering an RPO with a total precedence \succ on the symbols of the signature such that all the symbols have a lexicographic status, except $+$, whose status is multiset, and such that it respects the following requirements: (a) $\text{bin} \succ \text{binR} \succ \text{binL} \succ c \succ \text{size}$ for every constant c of sort TREES ; (b) $\text{size} \succ g \succ - \succ + \succ 0$ for every constant g of sort AG .

Proposition 4.5. *For any set G of ground literals, any saturation of $T_T \cup T_{\text{size}} \cup G$ w.r.t. \mathcal{SP}_{AG} is finite.*

Application (Algorithm 2.8 in [Zha06]: *Left-Rotation of trees*) Using the procedure induced by the calculus \mathcal{SP}_{AG} , it is possible to verify, e.g. that the input tree x and the output tree y have the same size:

1. $t := x$; 2. $y := \text{binR}(t)$; 3. $\text{binR}(t) := \text{binL}(y)$; 4. $\text{binL}(y) := t$; 5. Return y

In order to check that the size of x is exactly the one of y , we check for unsatisfiability modulo $T_T \cup T_{\text{size}} \cup AG$ the following constraint (see, again [Zha06]):

$$\begin{array}{l} \text{binR}(t') = \text{binL}(\text{binR}(x')) \wedge \text{binL}(t') = \text{binL}(x') \wedge \text{binL}(y') = t' \\ \wedge \text{binR}(y') = \text{binR}(\text{binR}(x')) \wedge \text{size}(x') \neq \text{size}(y') \end{array}$$

where x', y' and t' are fresh constants that identify the trees on which the algorithm applies.

4.5.1 Applying the Combination Framework

In the section above we have shown some examples of theories that extend the theory of abelian groups and whose constraint satisfiability problem is decidable. We have proved that AG can be enlarged to AG^* and AG and AG^* behave the same w.r.t. the satisfiability of constraints; moreover we have checked that AG is a Noetherian theory. To guarantee now that the theories that have been studied can be combined together it is sufficient to show that they fully satisfy the requirement of being AG -compatible and effectively Noetherian extension of AG (cf. Theorem 3.2). The AG -compatibility both of lists with length and trees with size is easily ensured observing that a constraint is satisfied w.r.t. $T_L \cup T_\ell \cup AG$ iff it is satisfied w.r.t. $T_L \cup T_\ell \cup AG^*$ and, analogously, any constraint is satisfiable w.r.t. $T_T \cup T_{size} \cup AG$ iff it is w.r.t. $T_T \cup T_{size} \cup AG^*$.

Moreover, checking the shape of the saturations produced, it is immediate to see that all the hypotheses required by Proposition 4.3 are satisfied when considering both the cases of lists with length and trees with size, turning \mathcal{SP}_{AG} not only into a decision procedure for the constraint satisfiability problem, but also into an effective method for deriving complete sets of logical consequences over the signature of abelian groups (namely, the AG -bases). This implies that also the requirement of being effectively Noetherian extensions of abelian groups is fulfilled for both lists with length and trees with size. To sum up, we have proved that the theories presented so far can be combined preserving the decidability of the constraint satisfiability problem.

4.6 Conclusion

The problem of integrating a reasoning modulo arithmetic properties into the superposition calculus has been variously studied, and different solutions have been proposed, both giving the possibility of reasoning modulo the linear rational arithmetic ([KV07]) and relying on an over-approximation of arithmetic via abelian groups ([GN04, Stu98]) or divisible abelian groups ([Wal01, Wal02]).

We have focused on the second kind of approach, giving an original solution to the satisfiability problem in combinations of theories sharing the theory of abelian groups. We have shown that in this case all the requirements to apply the non-disjoint combination method are satisfied, and we have considered an appropriate superposition calculus modulo abelian groups in order to derive satisfiability procedures. This calculus relies on a non trivial adaptation the one proposed in [GN04]: We consider a many-sorted and constraint-free version of the calculus, in which we use a restricted form of unification in abelian groups with free symbols, and in which only literals are involved. Under these assumptions we have proved that the calculus is refutationally complete, but, as a side remark, we notice that the same kind of proof works also in case the rules are extended to deal with Horn clauses and also, exactly as it happens in [GN04], after the introduction of an appropriate rule for the Factoring, to deal with general clauses. Our focus on the unit clause case is justified by our interest in the application to particular theories whose formalization is actually through axioms of that form.

It is worth noticing that two combination methods are involved in our approach: the method for unification problems [BS96] and the non-disjoint extension of Nelson-Oppen for satisfiability problems [GNZ08].

The framework for the non-disjoint combination used here cannot be applied, as it is, to the case where we consider a combination of theories sharing the Presburger arithmetic, because the latter is not Noetherian. Another framework, able to guarantee the termination of the resulting procedure on all the inputs, should be designed for that case.

As future work, we would like to relax current restrictions on theories and saturation types to apply effectively the calculus in the non-disjoint combination method. At the moment, since the presence of variables of sort AG in the clauses is not allowed, the results in [NRR09d] are not subsumed by the present paper. That restriction is justified by technical reasons: an important issue would be to discard it, enlarging in this way the applicability of our results.

References (of Chapters 3 - 4)

- [ABRS09a] Alessandro Armando, Maria P. Bonacina, Silvio Ranise, and Stephan Schulz. New results on rewrite-based satisfiability procedures. *ACM Transactions on Computational Logic*, 10(1), 2009.
- [ABRS09b] Alessandro Armando, Maria-Paola Bonacina, Silvio Ranise, and Stephan Schulz. New results on rewrite-based satisfiability procedures. *ACM Trans. on Computational Logic*, 10(1), 2009.
- [ARR03] Alessandro Armando, Silvio Ranise, and Michaël Rusinowitch. A rewriting approach to satisfiability procedures. *Information and Computation*, 183(2):140–164, 2003.
- [BE07] Maria Paola Bonacina and Mnacho Echenim. T-decision by decomposition. In *Proc. of CADE'07*, volume 4603 of *LNCS*, pages 199–214. Springer, July 2007.
- [BE08] Maria P. Bonacina and Mnacho Echenim. On variable-inactivity and polynomial T -satisfiability procedures. *Journal of Logic and Computation*, 18(1):77–96, 2008.
- [BG94] Leo Bachmair and Harald Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 4(3):217–247, 1994.
- [BJS90] 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.
- [BLS02] Randal E. Bryant, Shuvendu K. Lahiri, and Sanjit A. Seshia. Modeling and verifying systems using a logic of counter arithmetic with lambda expressions and uninterpreted functions. In E. Brinksma and K. G. Larsen, editors, *Proc. of CAV'02*, volume 2404 of *LNCS*, pages 78–92, Copenhagen (Denmark), 2002. Springer-Verlag.
- [BM07] Aaron Bradley and Zohar Manna. *The Calculus of Computation — Decision Procedures with Applications to Verification*. Springer, 2007.
- [BS96] Franz Baader and Klaus U. Schulz. Unification in the union of disjoint equational theories: Combining decision procedures. *Journal of Symbolic Computation*, 21(2):211–243, 1996.
- [Buc76] B. Buchberger. A theoretical basis for the reduction of polynomials to canonical forms. *ACM SIGSAM Bull.*, 10(3):19–29, 1976.
- [Che86] Philippe Le Chenadec. *Canonical Forms in Finitely Presented Algebras*. Research Notes in Theoretical Computer Science. Pitman-Wiley, 1986.
- [Der82] Nachum Dershowitz. Orderings for term-rewriting systems. *Theoretical Computer Science*, 17(3):279–301, 1982.
- [dMB08] Leonardo Mendonça de Moura and Nikolaj Bjørner. Engineering dpll(t) + saturation. In *Proc. of IJCAR'08*, volume 5195 of *LNCS*, pages 475–490. Springer, 2008.
- [End72] Herbert B. Enderton. *A Mathematical Introduction to Logic*. Academic Press, New York-London, 1972.

- [ES71] P. C. Eklof and G. Sabbagh. Model-completions and modules. *Annals of Mathematical Logic*, 2:251–295, 1971.
- [Fon09] Pascal Fontaine. Combinations of theories for decidable fragments of first-order logic. In *Proc. of FroCoS’09*, volume 5749 of *LNAI*, pages 263–278. Springer, 2009. Also as INRIA Report RR-6963.
- [Ghi04] Silvio Ghilardi. Model theoretic methods in combined constraint satisfiability. *Journal of Automated Reasoning*, 33(3-4):221–249, 2004.
- [GN04] Guillem Godoy and Robert Nieuwenhuis. Superposition with completely built-in abelian groups. *Journal of Symbolic Computation*, 37(1):1–33, 2004.
- [GNRZ07] Silvio Ghilardi, Enrica Nicolini, Silvio Ranise, and Daniele Zucchelli. Noetherianity and combination problems. In B. Konev and F. Wolter, editors, *Proc. of FroCoS 2007*, volume 4720 of *LNCS*, pages 206–220, Liverpool (UK), 2007. Springer-Verlag. Extended version available at <http://homes.dsi.unimi.it/~zucchelli/publications/conference/GhiNiRaZu-FroCoS-07.pdf>.
- [GNZ08] Silvio Ghilardi, Enrica Nicolini, and Daniele Zucchelli. A comprehensive combination framework. *ACM Transactions on Computational Logic*, 9(2):1–54, 2008.
- [Hod93] Wilfrid Hodges. *Model Theory*. Number 42 in Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1993.
- [KRRTO5] Hélène Kirchner, Silvio Ranise, Christophe Ringeissen, and Duc-Khanh Tran. On superposition-based satisfiability procedures and their combination. In D. Van Hung and M. Wirsing, editors, *Proc. of ICTAC 2005*, volume 3722 of *LNCS*, pages 594–608, Hanoi (Vietnam), 2005. Springer-Verlag.
- [KV07] Konstantin Korovin and Andrei Voronkov. Integrating linear arithmetic into superposition calculus. In *Proc. of CSL’07*, volume 4646 of *LNCS*, pages 223–237. Springer, 2007.
- [LM92a] Jean-Louis Lassez and Michael J. Maher. On Fourier’s algorithm for linear arithmetic constraints. *Journal of Automated Reasoning*, 9(3):373–379, 1992.
- [LM92b] Jean-Louis Lassez and Ken McAloon. A canonical form for generalized linear constraints. *Journal of Symbolic Computation*, 13(1):1–24, 1992.
- [Nic07] Enrica Nicolini. *Combined decision procedures for constraint satisfiability*. PhD thesis, Dipartimento di Matematica, Università degli Studi di Milano, Milano (Italy), 2007.
- [NO79] Greg Nelson and Derek C. Oppen. Simplification by cooperating decision procedures. *ACM Transaction on Programming Languages and Systems*, 1(2):245–257, 1979.
- [NR01] R. Nieuwenhuis and A. Rubio. Paramodulation-based theorem proving. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 7, pages 371–443. Elsevier Science, 2001.
- [NRR09a] Enrica Nicolini, Christophe Ringeissen, and Michaël Rusinowitch. Combinable extensions of abelian groups. In R. Schmidt, editor, *Proc. of CADE’09*, volume 5663 of *LNAI*, pages 51–66, Montreal (Canada), 2009. Springer.
- [NRR09b] Enrica Nicolini, Christophe Ringeissen, and Michael Rusinowitch. Combinable Extensions of Abelian Groups. Research Report, INRIA, 2009. RR-6920.
- [NRR09c] Enrica Nicolini, Christophe Ringeissen, and Michaël Rusinowitch. Data structures with arithmetic constraints: a non-disjoint combination. In *Proc. of FroCoS’09*, volume 5749 of *LNAI*, pages 335–350. Springer, 2009. Also as INRIA Report RR-6963.

- [NRR09d] Enrica Nicolini, Christophe Ringeissen, and Michaël Rusinowitch. Satisfiability procedures for combination of theories sharing integer offsets. In *Proc. of TACAS'09*, volume 5505 of *LNCS*, pages 428–442. Springer, 2009.
- [Plo72] Gordon Plotkin. Building-in equational theories. *Machine Intelligence*, 7:73–90, 1972.
- [PS81] Gerald E. Peterson and Mark E. Stickel. Complete sets of reductions for some equational theories. *J. ACM*, 28(2):233–264, 1981.
- [Sho84] Robert E. Shostak. Deciding combinations of theories. *J. of the ACM*, 31:1–12, 1984.
- [Stu98] Jürgen Stuber. Superposition theorem proving for abelian groups represented as integer modules. *Theoretical Computer Science*, 208(1-2):149–177, 1998.
- [Wal01] Uwe Waldmann. Superposition and chaining for totally ordered divisible abelian groups. In *Proc. of IJCAR'01*, volume 2083 of *LNCS*, pages 226–241. Springer, 2001.
- [Wal02] Uwe Waldmann. Cancellative abelian monoids and related structures in refutational theorem proving (Part I,II). *Journal of Symbolic Computation*, 33(6):777–829, 2002.
- [Zha06] Ting Zhang. *Arithmetic integration of decision procedures*. PhD thesis, Department of Computer Science, Stanford University, Stanford (U.S.), 2006.
- [Zuc08] Daniele Zucchelli. *Combination Methods for Software Verification*. PhD thesis, Università degli Studi di Milano and Université Henri Poincaré - Nancy 1, 2008.