Combining non-stably infinite, non-first order theories [1]

Pascal Fontaine [2] and Pascal Gribomont [3]

University of Liège (Belgium)

Abstract

A crucial step in the assertional verification of concurrent programs is deciding whether some sets of literals are satisfiable or not. In this context, the Nelson-Oppen combination scheme is often used. This scheme combines decision procedures for two disjoint theories into a decision procedure for the union of these theories. However, the standard version of the Nelson-Oppen technique tackles only one-sorted, stably infinite first-order theories. The scheme has previously been adapted to a many-sorted framework [10], and to handle non-stably infinite theories [9]. Those two enhancements were presented independently. We propose a unifying version in the continuity of both previous ones, which further relaxes the stably infinite requirement. Notably, some non-stably infinite theories can now be combined with the theory of arrays. Also, the combination scheme is presented here using a semantic notion of theory, allowing to handle non-first order theories.

The Nelson-Oppen combination framework [6] aims at creating a decision procedure for the union of two theories on disjoint languages, from the decision procedures for the quantifier-free fragment in each theory. For instance the set of literals

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

contains uninterpreted predicates and functions augmented with linear arithmetic on integer. The combination scheme provides a decision procedure for such sets of literals, from a decision procedure for uninterpreted predicates and functions (i.e., the empty first-order theory) and a decision procedure for linear arithmetic. This is possible because languages are disjoint, the only shared symbols being the equality and variables.

The Nelson-Oppen combination scheme is usually presented as a combination scheme for *first-order theories*. However, adopting a more semantic view

[1] This work was funded by a grant of the "Communauté française de Belgique - Direction de la recherche scientifique - Actions de recherche concertées"

[2] Email: pfontain@montefiore.ulg.ac.be

[3] Email: gribomont@montefiore.ulg.ac.be

This is a preliminary version. The final version will be published in Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

allows to avoid the delicate problem of dealing with, for instance, "theories" admitting only arbitrary large finite models[4] by precisely selecting the models of the "theory" one wants to consider. An *I-theory* is an arbitrary set of interpretations on the considered language. For example, considering linear arithmetic on integers, the I-theory will be the set of all interpretations corresponding to the single structure assigning set $\mathbb{Z}$ to the domain, and their usual meaning to symbols $+$, $-$, $\leq$, 0, 1... We present here a combination scheme for I-theories. As first-order theories can be expressed as I-theories,[5] this can be seen as a generalization of the usual first-order Nelson-Oppen combination scheme. A Shostak's combination scheme with a similar notion of generalized theory has already been presented in [4].

In a verification context, it is natural to express verification conditions in a many-sorted framework [3]. Very recently, the Nelson-Oppen method has been reformulated and proved correct in an order-sorted framework [10]. To simplify the presentation, we will rather work in a basic many-sorted framework, with disjoint sorts (i.e., without subsorts).

Traditionally it is required for the theories in a combination to be *stably infinite*. This means using the combination scheme is problematic with some common theories (notably those having only finite models). In the philosophy of [9], it will be shown that — in a many-sorted framework — non-stably infinite theories combine easily with some very useful theories (notably the empty theory, and the theories of arrays).

1 Preliminaries

A *many-sorted first-order language* is a tuple $\mathcal{L} = \langle \mathcal{S}, \mathcal{V}, \mathcal{F}, \mathcal{P}, r, d \rangle$ such that $\mathcal{S}$ is a countable non-empty set of sorts (or types), $\mathcal{V}$ is the (countable) union of disjoint countable sets $\mathcal{V}_\tau$ of variables of sort τ, $\mathcal{F}$ is a countably infinite set of function symbols, $\mathcal{P}$ is a countably infinite set of predicate symbols, r assigns an arity to each function symbol in $\mathcal{F}$ and each predicate symbol in $\mathcal{P}$, and d assigns a sort in $\mathcal{S}^{r(f)+1}$ to each function symbol $f \in \mathcal{F}$ and a sort in $\mathcal{S}^{r(p)}$ to each predicate symbol $p \in \mathcal{P}$. Nullary predicates are propositions, and nullary functions are constants. A subset of $\mathcal{L}$ is a many-sorted language such that the sets of sorts, variables, function and predicate symbols are subsets of the corresponding sets of $\mathcal{L}$.

It is assumed symbols are not overloaded. That is, a given function or predicate cannot be assigned to different sorts. It is however handy to use the same symbol with arguments of different sorts. In that case it suffices to consider that such symbols are implicitly "decorated" with their sort declaration. One example is the equality symbol "=", which is decorated into one

[4] A first-order theory admitting arbitrary large finite models also has infinite models.

[5] The I-theory corresponding to a first-order theory is just the set of the models of the first-order theory.

symbol for every sort. Terms, atoms (atomic formulas), literals, and formulas of a many-sorted language are defined in the usual way.

An *interpretation* of a formula in a many-sorted first-order language $\mathcal{L}$ is a pair $\mathcal{I} = \langle D, I \rangle$ where D assigns a non-empty domain D_τ to each sort $\tau \in \mathcal{S}$ and I assigns a meaning to each variable, function, and predicate symbol. As usual, the identity is assigned to the equality symbol. An interpretation $\mathcal{I}$ assigns a value $\mathcal{I}[t]$ in D_τ to every term t of sort τ. Similarly, interpretation $\mathcal{I}$ assigns a value $\mathcal{I}[\varphi]$ in $\{\top, \bot\}$ to every formula φ. An interpretation $\mathcal{I}$ is a model for formula φ if $\mathcal{I}[\varphi] = \top$. It is noted $\mathcal{I} \Vdash \varphi$. A *restriction* of an interpretation $\mathcal{I}$ in language $\mathcal{L}$ to a subset of $\mathcal{L}$ is the interpretation equal to $\mathcal{I}$ for every domain, function symbol, predicate symbol in the subset of $\mathcal{L}$.

A first-order theory is a set of axioms, which defines a set of interpretations, i.e., the models of the first-order theory. A natural extension is to consider I-theories: an I-theory is an arbitrary set of interpretations in a given many-sorted language. The I-theory corresponding to a first-order theory is the set of the models of the first-order theory. An I-theory may leave some predicates and functions uninterpreted. A predicate p of sort $\langle \tau_1, \ldots \tau_n \rangle$ (a function f of sort $\langle \tau_1, \ldots \tau_n, \tau \rangle$) is uninterpreted in an I-theory $\mathcal{T}$ if for every interpretation $\mathcal{I} = \langle D, I \rangle \in \mathcal{T}$, and for every predicate q (resp. function g) of suitable sort there is an interpretation $\mathcal{I}' \in \mathcal{T}$ such that $\mathcal{I}'$ is the same as $\mathcal{I}$ except that $\mathcal{I}'[p] = q$ (resp. $\mathcal{I}'[f] = g$). It is assumed that variables are always left uninterpreted in any I-theory, with a meaning similar to uninterpreted constants. Given an I-theory $\mathcal{T}$, a formula φ is $\mathcal{T}$-satisfiable if it has a model in $\mathcal{T}$. Two I-theories are disjoint if their languages are disjoint. Disjoint languages are languages with disjoint sets of functions (and constants) and predicates (though the equality symbol is shared).

Formulas including predicates and functions of two disjoint I-theories are written in the union of the disjoint languages. The union of the disjoint languages $\mathcal{L}_1 = \langle \mathcal{S}_1, \mathcal{V}_1, \mathcal{F}_1, \mathcal{P}_1, r_1, d_1 \rangle$ and $\mathcal{L}_2 = \langle \mathcal{S}_2, \mathcal{V}_2, \mathcal{F}_2, \mathcal{P}_2, r_2, d_2 \rangle$ is the many-sorted language $\mathcal{L} = \langle \mathcal{S}_1 \cup \mathcal{S}_2, \mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{F}_1 \cup \mathcal{F}_2, \mathcal{P}_1 \cup \mathcal{P}_2, r, d \rangle$, where r is the function equal to r_1 when its argument is in the domain of r_1, otherwise it is equal to r_2. Function d is defined similarly.
An I-theory $\mathcal{T}$ in the union $\mathcal{L}$ of disjoint languages $\mathcal{L}_1$ and $\mathcal{L}_2$ can be seen as the union of two disjoint I-theories: the restrictions $\mathcal{T}_i$ of $\mathcal{T}$ to each language $\mathcal{L}_i$, i.e., the set of all restrictions to $\mathcal{L}_i$ of interpretations in $\mathcal{T}$. However, it is more convenient to define the union of I-theories as:

Definition 1.1 The union $\mathcal{T}$ of I-theories $\mathcal{T}_1$ and $\mathcal{T}_2$ in disjoint languages $\mathcal{L}_1$ and $\mathcal{L}_2$ is the set of all interpretations $\mathcal{I}$ such that interpretations $\mathcal{I}_i = \langle D_i, I_i \rangle \in \mathcal{T}_i$ exist for $i = 1, 2$, with $\mathcal{I}_i$ being the restriction to $\mathcal{L}_i$ of $\mathcal{I}$.

In this definition the restriction to $\mathcal{L}_1$ of $\mathcal{T}$ is not necessarily $\mathcal{T}_1$. Indeed, if for a common sort τ in $\mathcal{L}_1$ and $\mathcal{L}_2$, an interpretation $\mathcal{I}_1$ from $\mathcal{T}_1$ assigns to τ a domain which is never assigned by any interpretation in $\mathcal{T}_2$, then no interpretation in $\mathcal{T}$ will have $\mathcal{I}_1$ as a restriction.

Notice that if $\mathcal{T}_1$ and $\mathcal{T}_2$ are the set of models of first-order theories T_1 and T_2, the union of I-theories $\mathcal{T}_1$ and $\mathcal{T}_2$ is the set of models of the (first-order) theory corresponding to the set of axioms $T_1 \cup T_2$. Our definitions for I-theory and union of I-theories extend the classical definitions of theory and union of theories but are not in contradiction with them.

As the combining method we propose applies to the general notion of I-theories introduced earlier, it is necessary to encapsulate the "combinable property" in a new notion. Combining two decision procedures for disjoint I-theories will require that, for each common sort, at least one of the I-theories is *flexible*:

Definition 1.2 An I-theory $\mathcal{T}$ in a many-sorted language $\mathcal{L}$ is *flexible* on sort τ if τ is not a sort of $\mathcal{L}$, or if for any interpretation $\mathcal{I} = \langle D, I \rangle$ in $\mathcal{T}$, any interpretation $\mathcal{I}' = \langle D', I' \rangle$ such that

- the sets D_τ and D'_τ have the same cardinality, and $D_{\tau'} = D'_{\tau'}$ for any $\tau' \neq \tau$. Function b defines a bijection from set D_τ to D'_τ, and is the identity on $D_{\tau'}$ for any $\tau' \neq \tau$;
- for any variable x, $\mathcal{I}'[x] = b(\mathcal{I}[x])$;
- for any function symbol $f \in \mathcal{F}$, $\mathcal{I}'[f](b(d_1), \dots b(d_n)) = b\left(\mathcal{I}[f](d_1, \dots d_n)\right)$;
- for any predicate symbol $p \in \mathcal{P}$, $\mathcal{I}'[p](b(d_1), \dots b(d_n)) \equiv \mathcal{I}[p](d_1, \dots d_n)$;

also belongs to $\mathcal{T}$.

Intuitively it means terms are assigned elements in the domain modulo a permutation. It allows to extend the use of the classic combination scheme to I-theories which do not necessarily correspond to first-order theories. As only one I-theory is needed to be flexible, the requirement is directly satisfied if one of the two I-theories corresponds to a first-order theory:

Theorem 1.3 *If I-theory $\mathcal{T}$ is the set of models of a first-order theory in a given language $\mathcal{L}$, then $\mathcal{T}$ is flexible on every sort.*

When combining more than two I-theories, it may seem the "flexible requirement" will impose all but one I-theory in the combination to correspond to first-order theories. Fortunately that is not the case. The following theorem shows that no restriction occurs if any two individual non-flexible I-theories have different sorts:

Theorem 1.4 *Let $\mathcal{T}$ be the union of the disjoint I-theories $\mathcal{T}_1$ and $\mathcal{T}_2$. If $\mathcal{T}_1$ and $\mathcal{T}_2$ are flexible on τ, then $\mathcal{T}$ is flexible on τ.*

The set L of literals in the little example at the beginning of this chapter is unsatisfiable, and so is $L_1 \cup L_2$, with

$$L_1 = \{x \leq y, y \leq x + v_1, v_1 = 0, v_2 = v_3 - v_4, v_5 = 0\}$$
$$L_2 = \{P\left(v_2\right), \neg P\left(v_5\right), v_1 = f\left(x\right), v_3 = h\left(x\right), v_4 = h\left(y\right)\}.$$

But every literal in L_i $(i = 1, 2)$ contains functions (and constants) and predicates from only one I-theory. Every literal in L_1 is in the language of arithmetic, whereas every literal in L_2 is in the language of uninterpreted predicates and functions. The shared symbols are variables only (and equality). The sets L_1 and L_2 are built from L simply by introducing new variables. A pair of sets of literals (L_1, L_2) is a separation of the finite set of literals L in the union of disjoint languages $\mathcal{L}_1$ and $\mathcal{L}_2$, if: first, L is $\mathcal{T}$-satisfiable if and only if $L_1 \cup L_2$ is $\mathcal{T}$-satisfiable; second, L_i is a finite set of literals in language $\mathcal{L}_i$, for $i = 1, 2$. The only shared symbols are variable symbols (and equality). In fact, given two languages and a set of literals in the union of those languages, a separation can always be built, using variable abstraction.

Given a partition $\mathcal{C}$ of the finite set of variables $\{x_1, \ldots x_n\}$, the *arrangement* induced by $\mathcal{C}$ is the set of all equalities between variables of the same sort in the same class of $\mathcal{C}$, and of all inequalities between two variables of the same sort in two distinct classes of $\mathcal{C}$. For instance, the arrangement of the three variables of the same sort x_1, x_2, x_3 induced by partition $\{\{x_1, x_2\}, \{x_3\}\}$ is $\{x_1 = x_2, x_1 \neq x_3, x_2 \neq x_3\}$. In fact the last inequality is not essential: the set of literals $\{x_1 = x_2, x_1 \neq x_3\}$ is logically equivalent to the preceding arrangement.

2 Cooperation of decision procedures

The following theorem provides the abstract basis for the combination procedure. With further requirements on the I-theories, it will allow to combine two decision procedures for disjoint languages into one for the union of languages:

Theorem 2.1 *Let I-theory $\mathcal{T}$ in language $\mathcal{L}$ be the union of I-theories $\mathcal{T}_1$ and $\mathcal{T}_2$ in disjoint languages $\mathcal{L}_1$ and $\mathcal{L}_2$. For each sort, either $\mathcal{T}_1$ or $\mathcal{T}_2$ is flexible. Let L be a set of literals on $\mathcal{L}$, and let (L_1, L_2) be a separation of L according to $\mathcal{L}_1$ and $\mathcal{L}_2$. Let V be the set of common variables in L_1 and L_2.*

Assume[6] *that for every domain τ in $\mathcal{L}_i$ and every term t of sort τ used in L_{2-i} there is an equality $x = t \in L_{2-i}$ where x is a variable in V'.*

The set of literals L is $\mathcal{T}$-satisfiable if and only if there exists an arrangement $\mathcal{A}$ of $V \cup V'$ and $\mathcal{T}_i$-models $\mathcal{I}_i$ of $\mathcal{A} \cup L_i$ for $i = 1, 2$ such that, for any common sort τ in $\mathcal{L}_1$ and $\mathcal{L}_2$, the domains associated to τ in $\mathcal{I}_1$ and $\mathcal{I}_2$ have the same cardinality. In that case $\mathcal{A} \cup L_1 \cup L_2$ is $\mathcal{T}$-satisfiable.

Proof. The proof can be found in appendix A. □

The classical way of solving the cardinality constraints like those in Theorem 6 is to consider only *stably infinite* theories. In a many-sorted context, it means both theories should be stably infinite with respect to the set of common sorts.

Definition 2.2 An I-theory $\mathcal{T}$ in a many-sorted language $\mathcal{L}$ is *stably infinite*

[6] Every separation can be transformed into a separation verifying this condition.

with respect to a set of sorts S if every $\mathcal{T}$-satisfiable ground set of literals L on $\mathcal{L}$ has a $\mathcal{T}$-model assigning a domain of cardinality $\aleph_0$ for every sort $\tau \in S$ used in $\mathcal{L}$.

Many useful I-theories are stably infinite: the I-theory of linear arithmetic on integer, I-theories corresponding to the first-order theories of arrays [8] or lists [7], etc. Also observe that, when talking about I-theories which are sets of models of first-order theories, the Löwenheim-Skolem Theorem[7] allows to relax the condition on domain cardinality in this definition: the cardinalities should be greater or equal to $\aleph_0$. But this remains a very strong requirement. Notably it prevents to combine I-theories with only finite domains.

We will rather adopt a pragmatic view, and study two I-theories in particular. *Compatible I-theories* are I-theories requiring no "cardinality clause" in Theorem 6:

Definition 2.3 Two disjoint I-theories $\mathcal{T}_1$ and $\mathcal{T}_2$ in many-sorted languages $\mathcal{L}_1$ and $\mathcal{L}_2$ are *compatible* if for every separation (L_1, L_2) and arrangement $\mathcal{A}$ such that

- V is the set of common symbols in L_1 and L_2;
- for every sort τ in $\mathcal{L}_i$ and every term t of sort τ used in L_{2-i} there is an equality $x = t \in L_{2-i}$ where x is a variable in V';
- $\mathcal{A}$ is an arrangement of $V \cup V'$;
- $\mathcal{A} \cup L_i$ is $\mathcal{T}_i$-satisfiable for $i = 1, 2$,

then there are $\mathcal{T}_i$-models $\mathcal{I}_i$ of $\mathcal{A} \cup L_i$ for $i = 1, 2$ such that $\mathcal{I}_1$ and $\mathcal{I}_2$ assign sets with the same cardinality for every common sort.

If two I-theories are stably infinite with respect to the set of common sorts then they are compatible. But this is not a necessary condition: many I-theories are not stably infinite, but are compatible with some other I-theories.

Assume that $\mathcal{T}_1$ and $\mathcal{T}_2$ are compatible disjoint I-theories such that, for each sort, at least one of them is flexible. Then there is a satisfiability decision procedure for sets of literals in the language of their union if there are satisfiability decision procedures for sets of literals in the language of each I-theory. Indeed, $L_1 \cup L_2$ is unsatisfiable in the union if, for every arrangement $\mathcal{A}$, $\mathcal{A} \cup L_i$ is unsatisfiable for $i = 1$ or $i = 2$. In theory, the decision procedure can be, first, to compute the separation, and second, to check every arrangement of the suitable set of variables together with each part of the separation. The original set is unsatisfiable, if and only if no arrangement has been found to be satisfiable with both parts of the separation.

Given a set of variables V, there are as many arrangements as partitions for V (at least in a one-sorted logic). The number of partitions, also known as Bell numbers (see for example [5]), grows exponentially with respect to the

[7] A formulation of Löwenheim-Skolem Theorem for many-sorted first-order logic may be found in [10].

size of $|V|$. In practice, checking every arrangement is not feasible, even for a small number of variables.[8] A more practical way for decision procedures to cooperate is by exchanging (disjunctions of) equalities.[9]

In Theorem 6, the set of variables on which the arrangement is built is larger than the set of *shared* variables, usually used in the Nelson-Oppen method. This is necessary for the combination with non-stably infinite I-theories, for instance, I-theories having only finite domains. A similar solution was also used in [11], to combine the theory of sets with a theory for the elements, even if this last theory is not stably infinite.

Combining decision procedures for a finite number of I-theories not only requires those I-theories to have a decision procedure. Some I-theories in the combination should be flexible on some sorts in such a way that for any two I-theories in the combination, at least one is flexible for each (common) sort. This requirement is not very restricting, as first-order I-theories are flexible on every sort (Theorem 1.3). The requirement on domain cardinality is more restricting: the I-theories that are to be combined together should be compatible. Unfortunately, there is no simple and practical criteria (other than stable infiniteness) to find if I-theories are compatible. However, some common I-theories have nice properties regarding compatibility.

3 The empty theory

The empty first-order theory in a language $\mathcal{L}$ defines the I-theory in which every constant, predicate, and function is uninterpreted, and no constraints are put on the domains. This I-theory will be referred to as the uninterpreted I-theory. It is certainly the simplest one, but it is nonetheless important as it allows uninterpreted predicates and functions to be added to the language of a decidable I-theory.

The uninterpreted I-theory has an optimal behaviour in a combination framework. Classically, only its stable infiniteness is exploited as all theories considered are assumed to be stably infinite. However, it was also noticed in [9] that it has the required properties to be combined not only with stably infinite theories but also with every first-order theory. Here, we show that it can be combined with *any* I-theory, corresponding to a first-order theory or not. The deep reason for this is encapsulated in this lemma:

Lemma 3.1 *Let $\mathcal{T}$ be the uninterpreted I-theory in a many-sorted language $\mathcal{L}$, and L be a set of literals on $\mathcal{L}$. If $\mathcal{I} = \langle D, I \rangle \in \mathcal{T}$ is a model of L, then there is a model $\mathcal{I}' = \langle D', I' \rangle \in \mathcal{T}$ of L such that, for any sort τ and any cardinality $\kappa \geq |D_\tau|$, $|D'_\tau| = \kappa$ and $|D'_{\tau'}| = |D_{\tau'}|$ for every $\tau' \neq \tau$.*

A corollary of preceding Lemma is the stable infiniteness of the uninterpreted I-theory:

[8] There are more than four million arrangements of 12 variables.

[9] The justification is similar to the one presented in [2].

Theorem 3.2 *The uninterpreted I-theory in a language $\mathcal{L}$ is stably infinite on every set of sorts of $\mathcal{L}$.*

Proof. Assume L is a satisfiable finite set of literals in $\mathcal{L}$. There is an interpretation $\mathcal{I} = \langle D, I \rangle$ of L such that D_τ is finite for every sort τ of $\mathcal{L}$. Using Lemma 3.1 on each sort successively, it is possible to build from $\mathcal{I}$ a model $\mathcal{I}'$ assigning to every sort a domain of cardinality $\aleph_0$. $\square$

As a consequence, the uninterpreted I-theory is compatible with every stably infinite I-theory. But a much stronger result can be deduced from Lemma 3.1:

Theorem 3.3 *The uninterpreted I-theory in a language $\mathcal{L}$ is compatible with every I-theory.*

Proof. Assume $\mathcal{T}_1$ is the uninterpreted I-theory in language $\mathcal{L}_1$ and $\mathcal{T}_2$ is any I-theory in language $\mathcal{L}_2$ disjoint from $\mathcal{L}_1$. Let (L_1, L_2) and $\mathcal{A}$ be a separation and an arrangement verifying the conditions of Definition 2.3.

For every common sort τ of $\mathcal{L}_1$ and $\mathcal{L}_2$, the arrangement $\mathcal{A}$ defines a partition of variables of sort τ: two variables of sort τ are in the same class if and only if they are assigned the same value by a model of $\mathcal{A}$. This partition contains a given number k_τ of classes. The models $\mathcal{I}_i$ of $\mathcal{A} \cup L_i$ attribute to each common sort τ a domain $D_{i,\tau}$ such that $|D_{i,\tau}| \geq k_\tau$. Furthermore, if $\mathcal{A} \cup L_1$ is satisfiable, then it has a model $\mathcal{I}_1$ which attributes to each common sort τ a domain $D_{1,\tau}$ such that $|D_{1,\tau}| = k_\tau$.

Using Lemma 3.1 on each common sort successively, it is possible to build from $\mathcal{I}_1$ a model $\mathcal{I}_1'$ assigning to every sort a domain of the same cardinality as interpretation $\mathcal{I}_2$. $\square$

Given any decidable I-theory $\mathcal{T}$ in language $\mathcal{L}$, adding any finite number of uninterpreted functions, predicates, or constants keeps the I-theory decidable, for sets of literals in the resulting language.

4 The theory of arrays

The first-order theory of arrays is one of the most popular theories used in combination schemes. A survey of results and history about it can be found in [5,8]. The satisfiability problem for sets of literals in the language of a theory of arrays is decidable. In [8] the occurrences of "write" are eliminated and every atom read$(a, \cdot)$ is translated to an application of an uninterpreted function associated to a, and finally congruence closure is used. In [1] it is proven that the superposition calculus is a satisfiability procedure for the (one-sorted) theory of arrays.

The theory contains "read-over-write" axioms:

$$\forall a \forall i \forall e \left[\mathrm{read}\left(\mathrm{write}\left(a, i, e\right), i\right) = e\right]$$

$$\forall a \forall i \forall j \forall e \left[i \neq j \Rightarrow \mathrm{read}\left(\mathrm{write}\left(a, i, e\right), j\right) = \mathrm{read}\left(a, j\right)\right]$$

and an extensionality axiom:

$$\forall a \forall b \left[\forall i \, \text{read}\,(a,i) = \text{read}\,(b,i) \Rightarrow a = b\right].$$

There are obviously three possible sorts: indices (i, j), arrays (a, b), and elements (e). But some sorts may be merged. It is natural and conservative to consider that the sort of arrays is disjoint from the sorts of elements and indices.[10] However, in numerous practical cases, it may be interesting to merge sorts for indices and elements. Two theories should be thus considered: the two-sorted theory, and the three-sorted theory.

The I-theory corresponding to the first-order theory of arrays is not compatible with every I-theory like the uninterpreted I-theory is. However, very positive results hold about the compatibility with the I-theory of arrays, thanks to the following lemma:

Lemma 4.1 *Let $\mathcal{T}$ be a two-sorted or three-sorted I-theory of arrays in language $\mathcal{L}$ and L be a set of literals on $\mathcal{L}$. If $\mathcal{I} = \langle D, I\rangle \in \mathcal{T}$ is a model of L, then there is a model $\mathcal{I}' = \langle D', I'\rangle \in \mathcal{T}$ of L such that, for sort τ of values (or indices) and any cardinality $\kappa \geq |D_\tau|$, $|D'_\tau| = \kappa$. If $\mathcal{T}$ is a three-sorted I-theory and τ' is the sort of indices (resp. values), then $|D'_{\tau'}| = |D_{\tau'}|$.*

Proof. The proof can be found in appendix B. □

The stable-infiniteness is a direct consequence of preceding lemma:

Theorem 4.2 *The two-sorted (three-sorted) I-theory of arrays in language $\mathcal{L}$ is stably infinite on every set of sorts.*

It is often required to use the I-theory of arrays with a finite domain for elements or indices. Its stable-infiniteness is of no help in those cases. To guarantee that it is still possible to combine the I-theory of arrays with non-stably infinite theories, it is further required that no inequality is used on the array sort. This can be achieved by replacing, in the separation part in the language of arrays, every inequality $t_1 \neq t_2$ where t_1 and t_2 are array terms by $\text{read}(t_1, i) \neq \text{read}(t_2, i)$ where i is a new index variable. As a consequence, new variables may have to be introduced in the set V' to verify conditions of Theorem 6.

Theorem 4.3 *The two-sorted and three-sorted I-theories of arrays are compatible with any I-theory in a language $\mathcal{L}$, if the sort of arrays is not a sort of $\mathcal{L}$, as long as no inequality is used on the array sort.*

Proof. Assume $\mathcal{T}_1$ is the two-sorted I-theory of arrays in language $\mathcal{L}_1$ and $\mathcal{T}_2$ is any I-theory in language $\mathcal{L}_2$ disjoint from $\mathcal{L}_1$. Let (L_1, L_2) and $\mathcal{A}$ be a separation and an arrangement verifying the conditions of Definition 2.3.

For every common sort τ of $\mathcal{L}_1$ and $\mathcal{L}_2$, the arrangement $\mathcal{A}$ defines a partition of variables of sort τ: two variables of sort τ are in the same class if

[10] Nevertheless elements in a theory of arrays can be arrays. But their language should be disjoint. For instance, read_1, write_1 and read_2, write_2.

and only if they are assigned the same value by a model of $\mathcal{A}$. This partition contains a given number k_τ of classes. The models $\mathcal{I}_i$ of $\mathcal{A} \cup L_i$ attribute to each common sort τ a domain $D_{i,\tau}$ such that $|D_{i,\tau}| \geq k_\tau$. As opposed to the empty theory, it is not direct to guarantee that, if $\mathcal{A} \cup L_1$ is satisfiable, then it has a model $\mathcal{I}_1$ (in the array I-theory) which attributes to each common sort τ a domain $D_{1,\tau}$ such that $|D_{1,\tau}| = k_\tau$.

Consider the two-sorted theory. Every term of the common sort τ used in L_1 is equal to a term used in $\mathcal{A}$. Assume $d \in D_{1,\tau}$ is associated to no term used in $\mathcal{A}$. Let $\mathcal{I}'_1$ be the same as $\mathcal{I}_1$ except that its domain for sort τ, $D'_{1,\tau}$ is $D_{1,\tau} \setminus \{d\}$, and $\mathcal{I}'_1[\mathrm{read}(a,i)] = \mathcal{I}_1[\mathrm{read}(a,i)]$ if $\mathcal{I}_1[\mathrm{read}(a,i)] \neq d$, or d' otherwise, where d' is an element of $D'_{1,\tau}$ chosen once and for all. Interpretation $\mathcal{I}'_1$ is still a model of every literal in L_1, and of both "read-over-write" axioms, as those axioms are only universally quantified. If formula $\forall i\, \mathrm{read}\,(a,i) = \mathrm{read}\,(b,i) \Rightarrow a = b$ is made false by $\mathcal{I}'_1$ for elements a and b of the array domain in $\mathcal{I}'_1$, then $\mathrm{read}\,(a,i) = \mathrm{read}\,(b,i)$ for every i. It is possible to build $\mathcal{I}''_1$ from $\mathcal{I}'_1$ such that only one representative of all such elements is kept. Interpretation $\mathcal{I}''_1$ still makes true both "read-over-write" axioms, and also the extensionality axiom. Assuming no inequality is used on the array sort in L_1, $\mathcal{I}''_1$ is also a model of every literal in L_1.

The same argument can be applied in the three-sorted case, for both elements and indices. Thus, if $\mathcal{A} \cup L_1$ is $\mathcal{T}_1$-satisfiable, it has a model $\mathcal{I}_1 \in \mathcal{T}_1$ which attributes to each common sort τ a domain $D_{1,\tau}$ such that $|D_{1,\tau}| = k_\tau$. Using Lemma 4.1 it is thus possible to build a model $\mathcal{I}_1$ assigning to every common sort a domain of the same cardinality as interpretation $\mathcal{I}_2$. □

Notice that previous theorem does not mean the I-theory of arrays is not compatible if the array sort is a common sort. It only states that if the array sort is a common sort, then the other I-theory must have a nice property to guarantee they are compatible. For instance, the array sort can be the sort of elements for another array I-theory. Also, the I-theory of arrays is compatible with the uninterpreted I-theory, even if both languages contain the sort of arrays.

5 Conclusion

This new presentation of the Nelson-Oppen combination scheme enhances the classical scheme in two aspects. First, it handles I-theories, i.e., arbitrary sets of interpretations, not necessarily corresponding to first-order theories. This improvement allows to deal directly, for instance, with the linear arithmetic I-theories for $\mathbb{N}$, $\mathbb{Z}$, with the arithmetic I-theory on $\mathbb{R}$, or with I-theories having only finite domains of unbounded size, used, for example in proof obligations issued in the context of verification of parameterized systems. Any quantifier-free decidable first-order language with interpreted functions and predicates corresponds to an I-theory which perfectly describes it.

An I-theory having only finite domains of unbounded size is essentially non-stably infinite. The second improvement in this presentation is the combination of non-stably infinite I-theories, following [9]. The many-sorted framework allows to combine in practice non-stably infinite I-theories with the I-theory corresponding to the empty theory, and with the (I-)theory of arrays.

As for the classical Nelson-Oppen framework, this framework for combining two I-theories can be applied to any number of I-theories, as long as they can be added one by one in the combination. A combination provides a decision procedure for the union of two I-theories, which is itself an I-theory. For instance, to deal with arrays containing lists of elements in $\mathbb{N}$ with some linear arithmetic on the sort of elements, the I-theory of lists can be combined with the I-theory for linear arithmetic on $\mathbb{N}$, thanks to the stable-infiniteness of both I-theories, and the obtained I-theory can be combined with the I-theory of arrays, using the good behaviour of the I-theory of arrays for combinations. Also, a three-sorted I-theory of arrays with functions read_1 and write_1 may have elements being arrays from another disjoint two-sorted I-theory of arrays with functions read_2 and write_2, if the only common sort is the sort of elements of for the first I-theory, i.e., the sort of arrays of the second one.

A limitation of this combination scheme is connected to the strict many-sorted framework. For instance, it does not allow to consider lists of elements which are either naturals *or* lists of naturals. Also, it remains to identify all cases where V' can be safely ignored in Theorem 6. Identifying more theories with nice compatibility properties is another issue for future research.

We would like to thank the anonymous reviewers for their comments.

References

[1] A. Armando, S. Ranise, and M. Rusinowitch. A rewriting approach to satisfiability procedures. *Information and Computation*, 183(2):140–164, 2003.

[2] S. Conchon and S. Krstić. Strategies for combining decision procedures. 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 537–553. Springer-Verlag, Apr. 2003.

[3] 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.

[4] H. Ganzinger. Shostak light. In A. Voronkov, editor, *Proc. Conference on Automated Deduction (CADE)*, volume 2392 of *Lecture Notes in Computer Science*, pages 332–346. Springer-Verlag, July 27-30 2002.

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

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

[7] D. C. Oppen. Reasoning about recursively defined data structures. *Journal of the ACM*, 27(3):403–411, July 1980.

[8] A. Stump, C. W. Barrett, D. L. Dill, and J. R. Levitt. A decision procedure for an extensional theory of arrays. In *Logic In Computer Science (LICS)*, pages 29–37. IEEE Computer Society, June 2001.

[9] C. Tinelli and C. G. Zarba. Combining non-stably infinite theories. In I. Dahn and L. Vigneron, editors, *Workshop on First Order Theorem Proving, (FTP)*, volume 86 of *Electronic Notes in Theoretical Computer Science*, Valencia, Spain, 2003. Elsevier Science Publishers.

[10] C. Tinelli and C. G. Zarba. Combining decision procedures for theories in sorted logics. Technical Report 04-01, Department of Computer Science, The University of Iowa, Feb. 2004.

[11] C. G. Zarba. Combining sets with elements. In N. Dershowitz, editor, *Verification: Theory and Practice*, volume 2772 of *Lecture Notes in Computer Science*, pages 762–782. Springer, 2004.

A Proof of Theorem 6

The condition is necessary. Let $\mathcal{I} \in \mathcal{T}$ be a model of L. By definition of separation, there is a model $\mathcal{I}' \in \mathcal{T}$ of $L_1 \cup L_2$. This interpretation $\mathcal{I}'$ perfectly defines an arrangement $\mathcal{A}$ of $V \cup V'$, and $\mathcal{I}'$ is a model of $\mathcal{A} \cup L_1 \cup L_2$, and so of $\mathcal{A} \cup L_i$.

Assume now interpretation $\mathcal{I}_i \in \mathcal{T}_i$ makes true $\mathcal{A} \cup L_i$ for $i = 1, 2$. Interpretation $\mathcal{I}_i$ defines a partition $\mathcal{C}_i$ of all terms used in $\mathcal{A} \cup L_i$ such that two terms a and b are in the same class if and only if $\mathcal{I}_i \Vdash a = b$. Consider $c_1 \in \mathcal{C}_1$ and $c_2 \in \mathcal{C}_2$, then $c_1 \cap c_2 \subset V \cup V'$. But $\mathcal{A}$ perfectly defines which elements of $V \cup V'$ are in the same class. It is thus easy to make partitions $\mathcal{C}_1$ and $\mathcal{C}_2$ coincide on classes containing a same element. Let $\mathcal{C}$ be a partition of terms in partitions $\mathcal{C}_1$ and $\mathcal{C}_2$ such that

- if $c_1 \in \mathcal{C}_1$ and $c_2 \in \mathcal{C}_2$ then $c_1 \cup c_2 \in \mathcal{C}$ if and only if $c_1 \cap c_2$ is non empty;
- for every $c_i \in \mathcal{C}_i$ there exists $c \in \mathcal{C}$ such that $c_i \subset c$. In other words, every class of $\mathcal{C}_i$ is fully contained in one class of $\mathcal{C}$;
- if $c_i \neq c'_i \in \mathcal{C}_i$, and if $c, c' \in \mathcal{C}$ are such that $c_i \subset c$ and $c'_i \subset c'$ then $c \neq c'$.

Partition $\mathcal{C}$ will allow to build an interpretation $\mathcal{I}$ from $\mathcal{I}_1$ and $\mathcal{I}_2$ such that $\mathcal{I} \Vdash \mathcal{A} \cup L_1 \cup L_2$.

First, let's define the domains for $\mathcal{I}$. For every sort τ in language $\mathcal{L}$, at least $\mathcal{T}_1$ or $\mathcal{T}_2$ is flexible. It can thus be assumed $\mathcal{I}_1$ and $\mathcal{I}_2$ assign the same domain to every common sort. $\mathcal{I}$ naturally assigns this domain for every common sort, whereas it assigns the domain assigned by $\mathcal{I}_i$ for every sort which is only in language $\mathcal{L}_i$.

Second, as for every sort τ in language $\mathcal{L}$, at least $\mathcal{T}_1$ or $\mathcal{T}_2$ is flexible, $\mathcal{I}_1$ and $\mathcal{I}_2$ can be assumed to associate the same element to every variable in $V \cap V'$.

Every class c of $\mathcal{C}$ contains elements of the same sort. If this is a common sort to both I-theories then $c \cap (V \cup V') \neq \emptyset$, and for every term t in c, $\mathcal{I}[t] = \mathcal{I}_1[t] = \mathcal{I}_2[t]$. If it is not, then every term $t \in c$ belongs to one language $\mathcal{L}_i$, and $\mathcal{I}[t] = \mathcal{I}_i[t]$. □

B Proof of Lemma 4.1

First assume $\mathcal{T}$ is the three-sorted I-theory and τ is the sort of values; τ' is the sort of indices and σ the sort of arrays. Let Δ be a set such that $|\Delta \cup D_\tau| = \kappa$, and such that $\Delta \cap D_{\tau'} = \Delta \cap D_\sigma = \emptyset$. The interpretation $\mathcal{I}' = \langle D', I' \rangle$ is built from interpretation $\mathcal{I}$:

- $D'_\tau = D_\tau \cup \Delta$, $D'_{\tau'} = D_{\tau'}$;
- for any constant a, $I'[a] = I[a]$;
- for any variable x, $I'[x] = I[x]$;
- if $v \in D_\tau$, $i \in D_{\tau'}$ and $a \in D_\sigma$ then $I'[\text{read}](a, i) = I[\text{read}](a, i)$ and

$I'[\text{write}](a, i, v) = I[\text{write}](a, i, v)$.

This suffices to make $\mathcal{I}'$ a model of L, but it is not yet perfectly defined, and not yet a model of $\mathcal{T}$. For every element a in D_σ, there is an element a' in $D'_\sigma \setminus D_\sigma$ corresponding to an array similar to a but such that a finite number of elements have been replaced by elements of Δ. More formally:

$$\{\text{write}(a, i, v), \text{write}(\text{write}(a, i, v), i', v'), \ldots\} \in D'_\sigma \setminus D_\sigma$$

for $\{v, v', \ldots\} \subset D'_\tau \setminus D_\tau$ and $\{i, i', \ldots\} \subset D_{\tau'}$. Special care must be taken to verify the extensionality and read-over-write axioms. Functions "read" and "write" are extended in the natural way.

The two-sorted case, and the three-sorted case when τ is the sort of indices are similar. Also notice that, if $\kappa \leq \aleph_0$, $|D_\tau| \leq \aleph_0$, $|D_{\tau'}| \leq \aleph_0$, and $|D_\sigma| \leq \aleph_0$, then $|D'_\sigma| \leq \aleph_0$. $\square$