A Rewriting Approach to the Combination of Data Structures with Bridging Theories

Paula Chocron, Pascal Fontaine, Christophe Ringeissen

A Rewriting Approach to the Combination of Data Structures with Bridging Theories

Paula Chocron[1], Pascal Fontaine[2⋆], and Christophe Ringeissen[2]

[1] IIIA-CSIC, Bellaterra, Catalonia, Spain
[2] INRIA, Université de Lorraine and LORIA, Nancy, France

Abstract. We introduce a combination method à la Nelson-Oppen to solve the satisfiability problem modulo a non-disjoint union of theories connected with bridging functions. The combination method is particularly useful to handle verification conditions involving functions defined over inductive data structures. We investigate the problem of determining the data structure theories for which this combination method is sound and complete. Our completeness proof is based on a rewriting approach where the bridging function is defined as a term rewrite system, and the data structure theory is given by a basic congruence relation. Our contribution is to introduce a class of data structure theories that are combinable with a disjoint target theory via an inductively defined bridging function. This class includes the theory of equality, the theory of absolutely free data structures, and all the theories in between. Hence, our non-disjoint combination method applies to many classical data structure theories admitting a rewrite-based satisfiability procedure.

1 Introduction

The modular construction of reasoning engines appears very often in logic and automated deduction, for instance to check whether a property still holds in a union of theories when this property holds in component theories. Working with signature-disjoint theories obviously simplifies the problem, e.g. in the Nelson-Oppen combination method where a satisfiability procedure for $T_1 \cup T_2$ is built from the satisfiability procedures for the two signature-disjoint theories T_1 and T_2. Even in that case, the signature-disjointness of T_1 and T_2 is not sufficient for the combination since additional "semantic" requirements on theories are required to get a complete satisfiability procedure. A first solution by Nelson and Oppen was to require stably infinite theories. This condition can be refined, and several other classes of *kind* theories have been recently investigated: shiny [20], polite [15] and gentle theories [10]. The Nelson-Oppen combination method is

⋆ This work has been partially supported by the project ANR-13-IS02-0001 of the Agence Nationale de la Recherche, by the European Union Seventh Framework Programme under grant agreement no. 295261 (MEALS), and by the STIC AmSud MISMT

now well-understood for disjoint unions of theories, and it is widely adopted to solve Satisfiability Modulo Theories (SMT) problems. It has become the core component of modern SMT solvers. But there is still an increasing demand on non-disjoint combinations. The extension of the Nelson-Oppen combination method to unions of non-disjoint theories has been already investigated during the last decade [11, 19]. This has led to the design of non-disjoint combination methods requiring some strong "semantic" assumptions on theories. However, these assumptions are difficult to meet in practical applications. For this reason, the use of non-disjoint combination methods in SMT solving is currently very limited.

We focus on simple techniques for non-disjoint combinations where the notions of polite and gentle theories [8] initially introduced for the disjoint case are also useful. In this paper, we consider a simple but meaningful non-disjoint case where the two theories T_1 and T_2 are connected by a *bridging* theory, whose axioms can be easily processed for any combined satisfiability problem. In this way, these non-disjoint combinations are reducible to disjoint ones. This avoids the need for complicated non-disjoint combination methods. Practical applications often involve a data structure theory T_1 and an arithmetic theory T_2. This particular union has been extensively studied, especially to combine an equational theorem prover processing (the axioms of) T_1 with an arithmetic solver for T_2 [7, 21]. This problem was first studied for disjoint combinations, but non-disjoint unions naturally arise when considering a bridging theory to relate the data structure theory T_1 to the arithmetic theory T_2, e.g. the length function for the data structure of lists [13, 14, 16]. The Ghilardi non-disjoint combination method [11] has been already applied to handle some connections between theories [3, 13, 14]. In [13, 14], the idea is to use superposition-based satisfiability procedures to process theory extensions of T_2. In that context, it is always a difficult and tedious task to design a new superposition calculus incorporating T_2 as a built-in theory.

In this paper, we develop a lightweight approach which is sufficient to handle the special case of bridging theories. This work is clearly inspired by the locality-based approach presented in [16] to handle bridging functions in local theory extensions. We consider the same problem by introducing a combination-based approach using a slight adaptation of the Nelson-Oppen disjoint combination method. Our approach has been initiated in [9] by studying the case of absolutely free data structures, with a particular focus on the adaptation required by the restriction to standard interpretations [6, 18, 22]. Like a locality-based satisfiability procedure applies to other theories of constructors [17], the combination method is actually applicable beyond the case of absolutely free data structures. In this paper, we investigate the constructor-based theories for which the combination method is sound and complete.

The main contribution of this paper is to identify a class of data structure theories for which our combination method is complete. In this class, theories are many-sorted, with disjoint sorts to denote respectively the data instances and the structure instances. Our combination method solves the satisfiability

problem in a union of a data structure theory plus a target theory and a bridging theory. With this method, the target theory can be arbitrary. Actually, this is due to the fact that we are focusing on data structure theories that fulfill a form of politeness [12, 15]. Hence, our work can be considered as a way to extend the use of polite theories to some simple non-disjoint combinations. The class of data structure theories is clearly of practical interest since it includes well-known theories for which a rewriting approach to satisfiability can be successfully applied [1, 2]. In this class, one can find the theory of equality, the theory of (acyclic) lists, the theory of absolutely free data structures (with or without selectors).

The completeness of our combination method requires the construction of a combined model from the models available in the component theories. For that purpose, we introduce the notion of basic data structure theory, for which a satisfiable input admits a Herbrand model with a very particular *basic* congruence relation E. The originality of our approach is to define a bridging theory as a convergent term rewrite system F, and to analyse the interplay between F and E. The careful study of $F \cup E$ as a convergent rewrite system modulo E leads to the construction of the combined model.

Section 2 recalls the classical notations and concepts used for the equational reasoning and the combination problem. In Section 4, we present the class of *basic* data structure theories. Section 3 introduces a combination procedure for extensions of basic data structure theories with bridging functions. By using a rewriting approach, the completeness of this combination procedure is proved in Section 5. Finally, Section 6 reports directions for future work.

2 Preliminaries: Notations and Combinations

We assume an enumerable set of variables $\mathcal{V}$ and a first-order many-sorted signature Σ given by a set of sorts and sets of function and predicate symbols (equipped with an arity ar). Nullary function symbols are called constant symbols. We assume that, for each sort σ, the equality "$=_\sigma$" is a logical symbol that does not occur in Σ and that is always interpreted as the identity relation over (the interpretation of) σ; moreover, as a notational convention, we omit the subscript for sorts and we simply use the symbol $=$. The notions of Σ-terms, atomic Σ-formulas and first-order Σ-formulas are defined in the usual way. In particular an atomic formula is either an equality, or a predicate symbol applied to the right number of well-sorted terms. Formulas are built from atomic formulas, Boolean connectives ($\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\equiv$), and quantifiers ($\forall$, $\exists$). A literal is an atomic formula or the negation of an atomic formula. A flat equality is either of the form $t_0 = t_1$ or $t_0 = f(t_1, \ldots, t_n)$ where each term $t_0, \ldots, t_n$ is a variable or a constant. A disequality $t_0 \neq t_1$ is flat when each term t_0, t_1 is a variable or a constant. A flat literal is either a flat equality or a flat disequality. An *arrangement* over a finite set of variables V is a maximal satisfiable set of well-sorted equalities and disequalities $x = y$ or $x \neq y$, with $x, y \in V$. Free variables are defined in the usual way, and the set of free variables of a formula φ is

denoted by $Var(\varphi)$. Given a sort σ, $Var_\sigma(\varphi)$ denotes the set of variables of sort σ in $Var(\varphi)$. A formula with no free variables is closed, and a formula without variables is ground. A universal formula is a closed formula $\forall x_1 \ldots \forall x_n.\varphi$ where φ is quantifier-free. A (finite) Σ-theory is a (finite) set of closed Σ-formulas. Two theories are disjoint if no predicate symbol or function symbol appears in both respective signatures.

From the semantic side, a *Σ-interpretation* $\mathcal{I}$ comprises non-empty pairwise disjoint domains I_σ for every sort σ, a sort- and arity-matching total function $\mathcal{I}[f]$ for every function symbol f, a sort- and arity-matching predicate $\mathcal{I}[p]$ for every predicate symbol p, and an element $\mathcal{I}[x] \in I_\sigma$ for every variable x of sort σ. By extension, an interpretation defines a value in I_σ for every term of sort σ, and a truth value for every formula. We may write $\mathcal{I} \models \varphi$ whenever $\mathcal{I}[\varphi] = \top$. A Σ-structure is a Σ-interpretation over an empty set of variables. Given a Σ-interpretation $\mathcal{I}$ and signature $\Sigma' \subseteq \Sigma$, the Σ'-reduct of $\mathcal{I}$ is the Σ'-interpretation, denoted by $\mathcal{I}^{\Sigma'}$, obtained from $\mathcal{I}$ by restricting it to interpret only the symbols in Σ'.

A model of a formula (theory) is an interpretation that evaluates the formula (resp. all formulas in the theory) to true. A formula or theory is satisfiable (or consistent) if it has a model; it is unsatisfiable otherwise. A formula G is T-satisfiable if it is satisfiable in the theory T, that is, if $T \cup \{G\}$ is satisfiable. A T-model of G is a model of $T \cup \{G\}$. A formula G is T-unsatisfiable if it has no T-models. In our context, the T-satisfiability problem for any set of literals can be equivalently defined as establishing the consistency of $T \cup \{G\}$ for a set of ground literals G expressed over the signature extended with some fresh constants.

A theory T is *stably infinite* if any T-satisfiable set of literals is satisfiable in a model of T whose domain is infinite. A Σ-theory T can be equivalently defined as a pair $T = (\Sigma, \mathbf{A})$, where $\mathbf{A}$ is a class of Σ-structures, and given a signature $\Sigma' \subseteq \Sigma$, the Σ'-reduct of T is $T^{\Sigma'} = (\Sigma', \{\mathcal{A}^{\Sigma'} \mid \mathcal{A} \in \mathbf{A}\})$. Given a set of Σ-equalities E, the relation $=_E$ denotes the *equational theory of E* which is defined as the smallest relation including E which is closed by reflexivity, symmetry, transitivity, congruence and substitutivity. As usual, the equivalence classes of ground Σ-terms modulo E defines the Σ-structure of ground terms modulo E, denoted by $T(\Sigma)/=_E$. A term rewrite system R is a set of oriented equalities. A convergent term rewrite system R is defined in the usual way [4], and it implies the existence and the unicity of a normal form, denoted by $t\downarrow_R$ for each equivalence class of a term t modulo $=_R$.

Let us now introduce some key notions for the combination problem [15].

Definition 1 (Smoothness). *Let Σ be a signature and $S = \{\sigma_1, \ldots, \sigma_n\}$ a set of sorts in Σ. A Σ-theory T is* smooth *with respect to S if:*

- *for every T-satisfiable quantifier-free Σ-formula φ,*
- *for every T-interpretation $\mathcal{A}$ satisfying φ,*
- *for every cardinal number $\kappa_1, \ldots, \kappa_n$ such that $\kappa_i \geq |A_{\sigma_i}|$, for $i = 1, \ldots, n$,*

there exists a T-interpretation $\mathcal{B}$ satisfying φ such that

$$|B_{\sigma_i}| = \kappa_i \text{ for } i = 1, \ldots, n$$

Definition 2 (Self witnessability). *Let Σ be a signature, S a set of sorts in Σ, and T a Σ-theory. A quantifier-free Σ-formula φ is* S-populated *if $Var_\sigma(\varphi)$ is non-empty for each $\sigma \in S$. A T-satisfiable S-populated quantifier-free Σ-formula φ is* self witnessable in T with respect to S *if there exists a T-interpretation $\mathcal{A}$ satisfying φ such that $A_\sigma = \{\mathcal{A}[v] \mid v \in Var_\sigma(\varphi)\}$ for each $\sigma \in S$. T is* self witnessable *with respect to S if any T-satisfiable S-populated quantifier-free Σ-formula φ is self witnessable in T with respect to S.*

Definition 3 (Perfect politeness). *Let Σ be a signature and S a set of sorts in Σ. A Σ-theory T is* perfectly polite *with respect to S if it is both smooth and self witnessable with respect to S.*

A typical example of a perfectly polite theory is the theory of equality. A perfectly polite theory is a particular polite theory [12,15]. As shown in [12,15], there exists a combination method to decide the satisfiability problem in a union of theories $T_s \cup T_t$ if

- T_s and T_t do not share function symbols but share a set of sorts S;
- T_s is polite with respect to S;
- and the satisfiability problem is decidable in both T_s and T_t.

In this paper, the considered polite theories are perfectly polite.

3 Combination with Bridging Theories

We investigate the satisfiability problem modulo a non-disjoint union $T_s \cup T_f \cup T_t$, where T_s is a data structure theory, e.g., the theory of absolutely free data structures [16]. The source and target theories T_s and T_t are connected with some theory T_f specifying a bridging function f by structural induction over the "constructors" of T_s. A typical example is trees of sort `struct` over elements of sorts in `Elem`, with tree size as bridging function. We now define these theories.

Definition 4. *Consider a set of sorts* `Elem`*, and a sort* `struct` $\notin$ `Elem`*. Let Σ be a signature whose set of sorts is* $\{\texttt{struct}\} \cup \texttt{Elem}$ *and whose function symbols $c \in \Sigma$ (called* constructors*) have arities of the form:*

$$c : \sigma_1 \times \cdots \times \sigma_m \times \texttt{struct} \times \cdots \times \texttt{struct} \to \texttt{struct}$$

where $\sigma_1, \ldots, \sigma_m \in$ `Elem`. To each n-ary constructor c, we associate the selectors *$s_1^c, \ldots, s_n^c$ that are disjoint from Σ and such that $s_i^c = s_j^d$ iff $i = j$ and $c = d$. Let $\Sigma^+ = \Sigma \cup \{s_i^c \mid c \in \Sigma,\ i = 1, \ldots, ar(c)\}$. Consider the following axioms (where upper case letters denote implicitly universally quantified variables)*

$$\begin{cases} (Inj_c) & c(X_1, \ldots, X_n) = c(Y_1, \ldots, Y_n) \Rightarrow \bigwedge_{i=1}^n X_i = Y_i \\ (Dis_{c,d}) & c(X_1, \ldots, X_n) \neq d(Y_1, \ldots, Y_m) \\ (Acyc_\Sigma) & X \neq t[X] \text{ if } t \text{ is a non-variable } \Sigma\text{-term} \\ (Proj_{c,i}) & s_i^c(c(X_1, \ldots, X_n)) = X_i \end{cases}$$

The Σ-theory of Absolutely Free Data Structures is

$$AFDS_\Sigma = \big(\bigcup_{c\in\Sigma} Inj_c\big) \cup \big(\bigcup_{c,d\in\Sigma, c\neq d} Dis_{c,d}\big) \cup Acyc_\Sigma$$

and the Σ^+-theory of Absolutely Free Data Structures with selectors is

$$AFDS^+_\Sigma = AFDS_\Sigma \cup \bigcup_{c\in\Sigma} \big(\bigcup_{i=1}^{ar(c)} Proj_{c,i}\big)$$

The class of Data Structure Theories $\mathbf{DST}^+$ *consists of all theories T_s such that T_s is any union of axioms among Inj_c, $Dis_{c,d}$, $Acyc_\Sigma$, and $Proj_{c,i}$. The subclass* $\mathbf{DST}$ *of* $\mathbf{DST}^+$ *consists of all theories without axioms $Proj_{c,i}$.*

$\mathbf{DST}^+$ includes inductive data structures (with selectors) such as lists and trees, but also, e.g., the theory of equality or injective functions alone.

Given a tuple $\boldsymbol{e}$ of terms of sorts in `Elem` and a tuple $\boldsymbol{t}$ of terms of sort `struct`, the tuple $\boldsymbol{e}, \boldsymbol{t}$ may be written $\boldsymbol{e}; \boldsymbol{t}$ to distinguish terms of sort `struct` from the other ones. Hence, a Σ-term is denoted by $c(\boldsymbol{e}; \boldsymbol{t})$.

Definition 5. *Consider two signatures Σ and Σ_t possibly sharing sorts except* `struct` *but no function symbols, where Σ complies with Definition 4. Let f be a new function symbol f with arity* `struct` $\rightarrow$ `t`*, where* `t` *is a sort in Σ_t. A* bridging *theory T_f associated to f has the form:*

$$T_f = \bigcup_{c\in\Sigma} \Big\{ \ \forall \boldsymbol{e} \forall t_1, \ldots, t_n \ . \ f(c(\boldsymbol{e}; t_1, \ldots, t_n)) = f_c(\boldsymbol{e}; f(t_1), \ldots, f(t_n)) \ \Big\}$$

where $f_c(\boldsymbol{x}; \boldsymbol{y})$ denotes a Σ_t-term.

Notice that the notation $f_c(\boldsymbol{x}; \boldsymbol{y})$ does not enforce all elements of $\boldsymbol{x}; \boldsymbol{y}$ to occur in the term $f_c(\boldsymbol{x}; \boldsymbol{y})$: in particular only elements in $\boldsymbol{x}$ of sort in Σ_t are allowed in $f_c(\boldsymbol{x}; \boldsymbol{y})$. Throughout the paper, we assume that for any constant c in Σ, f_c denotes a constant in Σ_t, and the equality $f(c) = f_c$ occurs in T_f. For instance, in the case of length of lists, $\ell(nil) = \ell_{nil} = 0$.

For the rest of this section, let $T = T_s \cup T_f \cup T_t$ where

- T_s is a Σ_s-theory in $\mathbf{DST}^+$;
- T_t is a stably infinite Σ_t-theory such that Σ_s and Σ_t do not share function symbols, and `struct` does not occur in Σ_t;
- T_f is a bridging theory.

We describe below a decision procedure for checking the T-satisfiability of sets of literals. As usual, the input set of literals is first purified to get a separate form.

Definition 6. *A set of literals φ is in* separate form *if $\varphi = \varphi_{struct} \cup \varphi_{elem} \cup \varphi_t \cup \varphi_f$ where:*

- φ_{struct} *contains only flat* Σ_s*-literals of sort* `struct`
- φ_{elem} *contains only flat* Σ_s*-literals of sort in* `Elem` *that are not* Σ_t*-literals*
- φ_t *contains only* Σ_t*-literals*
- φ_f *contains only flat equalities of the form* $f_x = f(x)$*, where* f_x *denotes a variable associated to* $f(x)$*, such that* f_x *and* $f(x)$ *occur once in* φ_f *and each variable of sort* `struct` *in* φ_{struct} *occurs in* φ_f*.*

It is easy to convert any set of literals into an equisatisfiable separate form by introducing fresh variables to denote impure terms.

Unlike classical disjoint combination methods, guessing only one arrangement on the shared variables is not sufficient to get a modular decision procedure. An additional arrangement on variables of sort `struct` is considered and the resulting set of flat Σ-equalities E is translated to a set of Σ_t-literals CP_E.

Definition 7. *Given a bridging theory* T_f*, the* target encoding *of a set of flat* Σ*-equalities* E *is the set of* Σ_t*-literals*

$$\begin{aligned} CP_E = \quad & \{f_{x'} = f_c(\boldsymbol{e}; f_{x_1}, \ldots, f_{x_n}) \mid c(\boldsymbol{e}; x_1, \ldots, x_n) = x' : \texttt{struct} \in E\} \\ & \cup \{f_{x'} = f_x \mid x = x' : \texttt{struct} \in E\} \end{aligned}$$

The combination procedure below is presented in [9] for the particular case of absolutely free data structures.

Lemma 1. *Let* $\varphi = \varphi_{struct} \cup \varphi_{elem} \cup \varphi_t \cup \varphi_f$ *be a set of literals in separate form,* $V = Var(\varphi_{struct} \cup \varphi_{elem}) \cup Var_{\texttt{struct}}(\varphi_f)$*, and* $V_t = Var(\varphi_t) \cup Var_{\texttt{t}}(\varphi_f)$*. The formula* φ *is* T*-satisfiable if and only if there exist an arrangement* Arr_t *over* $V \cap V_t$ *and an arrangement* $Arr_{\texttt{struct}}$ *over the set of variables of sort* `struct` *in* V*, such that*

- $\varphi_{struct} \cup \varphi_{elem} \cup Arr_t \cup Arr_{\texttt{struct}}$ *is* T_s*-satisfiable,*
- $\varphi_t \cup Arr_t \cup CP_E$ *is* T_t*-satisfiable,*

where E *is the set of equalities in* $\varphi_{struct} \cup Arr_{\texttt{struct}}$*.*

Proof. (Only if direction: soundness) Straightforward, since $T_f \cup T_t \cup \varphi \models CP_E$. (If direction: completeness) See Lemma 2 in Section 5.

□

For the sake of simplicity in Lemma 1, we have chosen to use arrangements to fix E. In practice however another solution would be to use a saturation-based T_s-satisfiability procedure with the capability to deduce E, like the one introduced in Section 4.

Example 1. Let T_t be the theory of integers and a theory of binary trees over integers, with `Elem` $= \{\texttt{int}\}$, constructors $\Sigma = \{nil : \texttt{struct}, cons : \texttt{int} \times \texttt{struct} \times \texttt{struct} \rightarrow \texttt{struct}\}$, and selectors $val, left, right$, formally defined by $T_s = \{val(cons(I, X, Y)) = I, left(cons(I, X, Y)) = X, right(cons(I, X, Y)) = Y\}$. The bridging theory for the function sum : `struct` $\rightarrow$ `int` is $T_{sum} = \{sum(nil) = 0, sum(cons(I, X, Y)) = I + sum(X) + sum(Y)\}$.

Consider the T-satisfiability of

$$\varphi = \{a = cons(e,b,c), d = cons(0, left(a), right(a)), a \neq d,\\ sum(a) \leq sum(left(a)) + sum(right(a)), e \geq 0\}$$

or in separate form $\varphi_{struct} \cup \varphi_t \cup \varphi_{sum}$ with

$$\varphi_{struct} = \{a = cons(e,b,c), d = cons(e', a_1, a_2),\\ a_1 = left(a), a_2 = right(a), a \neq d\}$$

$$\varphi_t = \{sum_a \leq sum_{a_1} + sum_{a_2}, e' = 0, e \geq 0\}$$

$$\varphi_{sum} = \{sum_x = sum(x) \mid x \in \{a, a_1, a_2, b, c, d\}\}$$

Let us compute the arrangements used in Lemma 1. First, Arr_{struct} relates a, b, c, d, a_1 and a_2; notice that $a_1 = left(a) = left(cons(e,b,c)) = b$ and similarly $a_2 = c$, so these equalities should belong to Arr_{struct}, as well as $a \neq d$ (from φ_{struct}). Second, Arr_t should be $\{e \neq e'\}$, since otherwise $a = cons(e,b,c) = cons(e',b,c) = cons(e',a_1,a_2) = d$ holds, in contradiction with Arr_{struct}.

The target encoding CP_E will contain $sum_a = e + sum_b + sum_c$, as well as $sum_b = sum_{a_1}$, $sum_c = sum_{a_2}$, derived from the equalities in Arr_{struct}. From $sum_a \leq sum_{a_1} + sum_{a_2}$ in φ_t, we get $e = 0$, contradicting Arr_t since $e' = 0$. ■

Example 2. Assume that T_{sum} and T_t are defined as in Example 1. The formula

$$\varphi = \{a = cons(e,b,c), a = cons(e', a_1, a_2), sum(a) \leq sum(a_1) + sum(a_2), e > 0\}$$

can easily be shown unsatisfiable modulo $AFDS_\Sigma \cup T_{sum} \cup T_t$. However, in the combination $EQ_\Sigma \cup T_{sum} \cup T_t$ where EQ_Σ is the theory of equality over Σ, Arr_{struct} can be such that a, a_1, a_2, b, c are all different, and $Arr_t = \{e \neq e'\}$:

- $\varphi_{struct} \cup \varphi_{elem} \cup Arr_t \cup Arr_{\texttt{struct}}$ is trivially EQ_Σ-satisfiable
- $\varphi_t \cup Arr_t \cup CP_E$ is satisfiable in the theory of integers, e.g. with $e' = 0$.

Consequently, φ is satisfiable modulo $EQ_\Sigma \cup T_{sum} \cup T_t$. ■

The combination method requires only few restrictions on the target theory to be sound and complete (cf. Section 5). Actually, T_t could be also a data structure theory obtained from a previous application of the combination method. Consider the case $T = T_{tree} \cup T_{sz} \cup T_t$ where T_{tree} denotes a theory of trees and T_{sz} denotes the bridging theory defining the tree size thanks to a target theory $T_t = T_{list} \cup T_\ell \cup T_\mathbb{Z}$ corresponding to a theory of lists extended with a bridging function ℓ computing the list length. Applying twice the combination method is a way to build a T-satisfiability procedure where T corresponds to the union of two disjoint data structure theories extended with their respective bridging functions to $T_\mathbb{Z}$: $T = (T_{tree} \cup T_{sz} \cup T_\mathbb{Z}) \cup (T_{list} \cup T_\ell \cup T_\mathbb{Z})$. In the same vein, the combination method applied twice yields a satisfiability procedure for a theory of lists of trees extended with tree size sz and list length ℓ. The above examples illustrate the generality of our combination method.

4 Basic Data Structure Theories

The class $\mathbf{DST}^+$ introduced above includes theories of practical interest worth considering for non-disjoint combinations with bridging functions. It contains the theory of Absolutely Free Data Structures, but also the theory of equality and other theories for which a rewriting approach to satisfiability can be successfully applied [2]. It appears that those theories satisfy a nice model-theoretic property, instrumental to prove the completeness of the above combination procedure. They admit some particular Herbrand models similar to the ones we can build for the theory of equality. Hence, it is another way to consider data structure theories that can be "reduced" to the theory of equality. In the same vein, one could use the locality approach [17] to get a "reduction" to the theory of equality through a finite instantiation of axioms. Our model-based approach eases the construction of a model for data structures extended with bridging functions.

Rather than handling a set of literals and a theory, we will consider the theory extension including a set of (ground) literals.

Definition 8. *Consider a finite constant expansion $\Sigma_s \cup C$ of a signature Σ_s such that C_σ is non-empty for each sort σ in Σ_s, and a Σ_s-theory T_s. A* ground flat T_s-extension *is a $\Sigma_s \cup C$-theory defined as the union $\mathfrak{T}_s = T_s \cup G$ such that G is a finite set of ground flat $\Sigma_s \cup C$-literals. $\mathfrak{T}_s = T_s \cup G$ is said to be* subpopulated *if C contains for each sort a constant not occurring in G.*

The consistency of a ground flat T_s-extension $\mathfrak{T}_s$ corresponds to a T_s-satisfiability problem of a set of flat literals. We focus on theories admitting models defined as structures of terms generated by some constructors and (a superset of) the free constants occurring in $\mathfrak{T}_s$. We will see in the proof of Proposition 1 that the unused constant generators in subpopulated $\mathfrak{T}_s$ are required to build the models in the presence of selectors.

The model-theoretic properties of $\mathbf{DST}^+$ theories are essential for combinations: models can be generated from some of their symbols (i.e., the constructors). The following definition captures these properties:

Definition 9. *Consider a set of sorts* `Elem`*, and a sort* `struct` $\notin$ `Elem`*. Let Σ_s be a signature whose set of sorts is* $\{$`struct`$\} \cup$ `Elem`*. Let $\Sigma \subseteq \Sigma_s$ be a signature containing only function symbols whose codomain sort is* `struct`*. Let $\mathfrak{T}_s = T_s \cup G$ be a ground flat T_s-extension whose signature is $\Sigma_s \cup C$. A* Σ-basic Herbrand model *of $\mathfrak{T}_s$ is a model $\mathcal{H}$ of $\mathfrak{T}_s$ such that $\mathcal{H}^{\Sigma \cup C}$ is $T(\Sigma \cup C)/=_E$, where E is a finite set of ground flat $\Sigma \cup C$-equalities defined as the set of $\Sigma \cup C$-equalities in G plus some additional equalities between constants of C occurring in G.*

A consistent Σ_s-theory T_s is a Σ-basic *(resp.,* perfect Σ-basic*)* data structure theory *if any subpopulated (resp., arbitrary) consistent ground flat T_s-extension admits a Σ-basic Herbrand model.*

A Σ-basic Herbrand model is constructed on a subsignature Σ of T_s. This introduces a natural distinction between *constructors* in Σ and defined symbols in $\Sigma_s \setminus \Sigma$. The constructors are used to build the domain of the basic Herbrand

model whilst the defined symbols are interpreted as operators on this domain. A classical example is $AFDS^+_\Sigma$ where the selectors are the defined symbols. From now on, we assume that function symbols $c \in \Sigma$ have arities as in Definition 4:

$$c : \sigma_1 \times \cdots \times \sigma_m \times \mathtt{struct} \times \cdots \times \mathtt{struct} \to \mathtt{struct}.$$

Notice also that the above definition is suitable for a deductive approach in contrast to a guessing approach. In a guessing approach, the set E of equalities would be maximal (obtained from an arrangement) and in that case no additional equality would be needed.

We now prove that all the source theories T_s considered in Section 3, ranging from the theory of equality to $AFDS^+_\Sigma$, are Σ-basic data structure theories. For any of these source theories T_s, a saturation-based calculus (see Figure 1) provides a T_s-satisfiability procedure. As a side effect, the saturated set computed by this calculus yields a Σ-basic Herbrand model.

Proposition 1. *Theories in* $\mathbf{DST}^+$ *are Σ-basic data structure theories, and theories in* **DST** *are perfect Σ-basic data structure theories.*

Proof. Consider any finite set of ground flat $\Sigma_s \cup C$-literals G and $\mathfrak{T}_s = T_s \cup G$. To check the consistency of $\mathfrak{T}_s$, we can use a (simplified) superposition calculus. It can be viewed as an abstract congruence closure procedure for the theory of equality extended with additional simplification rules on ground clauses to take into account the axioms listed above. In Figure 1, we provide a version of this calculus instantiated for the case of $AFDS^+_\Sigma$. One may remark that there is a one to one correspondence between the axioms of $AFDS^+_\Sigma$ and inference rules of this calculus. If we want to omit an axiom of $AFDS^+_\Sigma$, we just have to remove the corresponding inference rule, to get a satisfiability procedure. Hence, if we omit $\mathbf{Inj}_c$, $\mathbf{Dis}_{c,d}$, $\mathbf{Acyc}_\Sigma$ and $\mathbf{Proj}_{c,i}$, we retrieve the inference system for the satisfiability problem in the theory of equality. This inference system is parametrised by an ordering on constants.

For any considered T_s, the calculus terminates by computing a finite saturation. If this finite saturation does not contain the empty clause, $\mathfrak{T}_s$ is consistent. Moreover, it is possible to construct a model using the model-generation technique introduced by Bachmair and Ganzinger [5]: the set of equalities in the finite saturation leads to a convergent term rewrite system R such that the structure of R-normal forms $T(\Sigma_s \cup C)\downarrow_R$ is a model of $\mathfrak{T}_s$. Let E be the set of equalities corresponding to ground flat rules in R. Then, we must distinguish two cases:

- If T_s does not include the *Projection* axiom, R only consists of ground flat rules. In that case, we can take $\Sigma = \Sigma_s$, and $T(\Sigma_s \cup C)\downarrow_R$ is isomorphic to $T(\Sigma \cup C)/=_E$.
- Otherwise, we consider the signature Σ obtained from Σ_s by removing selectors, and a structure whose domain is $T(\Sigma \cup C)\downarrow_R$. By assumption on the constants used in G, there is a constant $u_{\mathtt{s}} \in C$ not occurring in $\mathfrak{T}_s$, for each $\mathtt{s}$ in Σ_s. On this domain, the selector s^c_i with $\mathtt{s}$ as codomain sort is interpreted as follows:

- For any normal form which is a constant x, $s_i^c(x) = x'$ if $s_i^c(x)\downarrow_R = x' \in C$, otherwise $s_i^c(x) = u_{\mathtt{s}}$.
- For any normal form which is a term $t = c(t_1, \ldots, t_n)$, $s_i^c(t) = t_i$
- For any other normal form t, $s_i^c(t) = u_{\mathtt{s}}$.

Using this interpretation, we get a structure of the desired form that is still a model of $\mathfrak{T}_s$, when $\mathfrak{T}_s$ includes the $Proj_{c,i}$ axiom. □

$$
\begin{array}{ll}
\textbf{Sup}: & x = x', x = y \vdash x' = y \quad \text{if } x > x', x > y \\
\textbf{Cong1}: & x_j = x'_j, x = f(\ldots, x_j, \ldots) \vdash x = f(\ldots, x'_j, \ldots) \quad \text{if } x_j > x'_j \\
\textbf{Cong2}: & x = f(x_1, \ldots, x_n), x' = f(x_1, \ldots, x_n) \vdash x = x' \\
\textbf{Param}: & x = x', x \neq y \vdash x' \neq y \quad \text{if } x > x', x > y \\
\textbf{Ref}: & x \neq x \vdash \square \\
\textbf{Inj}_c: & x = c(x_1, \ldots, x_n), x = c(x'_1, \ldots, x'_n) \vdash x_1 = x'_1 \ldots x_n = x'_n \quad \text{if } c \in \Sigma \\
\textbf{Dis}_{c,d}: & x = c(x_1, \ldots, x_n), x = d(y_1, \ldots, y_m) \vdash \square \quad \text{if } c, d \in \Sigma, c \neq d \\
\textbf{Acyc}_{\Sigma}: & x = t_1[x_1], \ldots, x_{n-1} = t_n[x] \vdash \square \quad \text{if } t_1, \ldots, t_n \text{ are } \Sigma\text{-terms of depth 1} \\
\textbf{Proj}_{c,i}: & x = c(x_1, \ldots, x_n) \vdash x_i = s_i^c(x)
\end{array}
$$

Fig. 1. T_s-satisfiability procedure

Proposition 2. *Theories in* **DST** *are perfectly polite with respect to* `Elem`*.*

Proof. Self witnessability directly follows from the definition of a perfect Σ-basic data structure theory. The smoothness is a consequence of the fact that sorts in `Elem` are only inhabited by constants in C. Thus, the set of generators C can be extended to any set of generators whose cardinality is larger than the cardinality of C, and the related term-generated structure remains a model. □

5 Completeness Proof

We study the satisfiability problem modulo $T = T_s \cup T_f \cup T_t$ where T_f is a bridging theory between a source theory T_s and a target theory T_t fulfilling the following assumption:

Assumption 1 (Theories) *The Σ_s-theory T_s and the Σ_t-theory T_t share no function symbol. The set of sorts in Σ_s is* $\mathtt{Elem} \cup \{\mathtt{struct}\}$*, and* `struct` *does not occur in Σ_t. One of the following three cases hold:*

- *sorts in Σ_s and Σ_t are disjoint, T_s is a Σ-basic data structure theory and T_t is arbitrary*
- *Σ_s and Σ_t share sorts, and either*
  - *$T_s \in$* **DST** *and T_t is arbitrary*
  - *$T_s \in$* $\mathbf{DST}^+ \backslash \mathbf{DST}$ *and T_t is stably infinite.*

The combination procedure described in Section 3 is sound and complete also for the cases listed above. We prove the completeness of the combination procedure thanks to a combined model constructed using rewriting techniques. Given a bridging function $f : \texttt{struct} \to \texttt{t}$ where $\texttt{t}$ is a sort from the target theory, we define a bridging theory via a convergent term rewrite system F such that for any term s of sort $\texttt{struct}$, its normal form $f(s)\downarrow_F$ corresponds to a term that can be interpreted in a model of the target theory. To solve this problem, we must carefully study the interplay between the equational theory E related to a Σ-basic Herbrand model and the term rewrite system F.

For convenience, we will consider theory extensions including the sets of (ground) literals rather than handling literals and theories separately.

Assumption 2 (Input formulas) *Let T_s and T_t be theories as in Assumption 1. The signatures $\Sigma_s \cup C$ and $\Sigma_t \cup C_t$ are finite constant expansions of Σ_s and Σ_t, respectively.*

1. *$\mathfrak{T}_s$ is a consistent $\Sigma_s \cup C$-theory defined as a subpopulated ground flat extension of T_s. It admits a Σ-basic Herbrand model $\mathcal{H}$ such that $\mathcal{H}^{\Sigma \cup C}$ is $T(\Sigma \cup C)/=_E$. The set of $C \cap C_t$-literals occurring in $\mathfrak{T}_s$ is an arrangement denoted by Arr_t.*
2. *$\mathfrak{T}_t$ is a $\Sigma_t \cup C_t$-theory defined as the union of T_t and some finite set of ground $\Sigma_t \cup C_t$-literals, such that $\mathfrak{T}_t \cup Arr_t$ is consistent.*

From now on, we assume a context where Assumption 1 and Assumption 2 hold.

A bridging theory T_f (from Definition 5 above) is an equational theory. It happens that it can naturally be oriented as a term rewrite system F.

Proposition 3. *Let T_f be a bridging theory as introduced in Definition 5, and let $\mathfrak{T}_F = T_f \cup \{f(x) = f_x \mid x : \texttt{struct} \in C\}$. The term rewrite system $F = \{f(l) \to r \mid f(l) = r \in \mathfrak{T}_F\}$ is convergent and satisfies the following properties:*

- *$f(c(\boldsymbol{e}; t_1, \ldots, t_n))\downarrow_F = f_c(\boldsymbol{e}; f(t_1)\downarrow_F, \ldots, f(t_n)\downarrow_F)$ for any non-constant constructor $c \in \Sigma$;*
- *$f(c)\downarrow_F = f_c$ for any constant c in Σ, where f_c is a constant in Σ_t;*
- *$f(x)\downarrow_F = f_x$ for any constant $(x : \texttt{struct}) \in C$, where $(f_x : \texttt{t}) \in C_t$.*

Example 3. Consider the length function ℓ from lists to integers, and let $\mathfrak{T}_s = \{a = cons(e, b), b = cons(e', c), c = nil, a \neq c\}$. The set of constants of sort $\texttt{struct}$ in $\mathfrak{T}_s$ is $\{a, b, c\}$ and the related term rewrite system F is given by $\{\ell(cons(W, X)) \to 1 + \ell(X), \ell(nil) \to 0\} \cup \{\ell(a) \to \ell_a, \ell(b) \to \ell_b, \ell(c) \to \ell_c\}$. ■

We focus on the problem of checking the $\mathfrak{T}_s \cup \mathfrak{T}_F \cup \mathfrak{T}_t$-consistency. To get a well-defined interpretation for $f : \texttt{struct} \to \texttt{t}$, we need a $\mathfrak{T}_t$-model in which f returns the same value of sort $\texttt{t}$ for all E-equal input terms of sort $\texttt{struct}$. This motivates the following definition of *E-compatibility*. Below, a $\texttt{struct}$-*term* denotes a Σ-term in which constants of sort $\texttt{struct}$ are in C.

Definition 10. *F is E-*compatible *in a model $\mathcal{A}$ of $\mathfrak{T}_t$ if for any $\texttt{struct}$-terms s and t, $s =_E t \Rightarrow \mathcal{A}[f(s)\downarrow_F] = \mathcal{A}[f(t)\downarrow_F]$.*

Proposition 4. *If F is E-compatible in a model of $\mathfrak{T}_t$, then $\mathfrak{T}_s \cup \mathfrak{T}_F \cup \mathfrak{T}_t$ is consistent.*

Proof. Consider the set of sorts S shared by Σ_s and Σ_t. Let us first assume $S = \emptyset$. We know that F is E-compatible in a model $\mathcal{A}$ of $\mathfrak{T}_t$, and there exists a model $\mathcal{H}$ of $\mathfrak{T}_s$ such that $\mathcal{H}^{\Sigma \cup C}$ is $T(\Sigma \cup C)/=_E$. Given $\mathcal{A}$ and $\mathcal{H}$, let us define an interpretation $\mathcal{M}$ as follows. The domains of $\mathcal{M}$ are:

- $M_{\mathtt{t}} = A_{\mathtt{t}}$ for any sort $\mathtt{t}$ in Σ_t
- $M_{\mathtt{s}} = H_{\mathtt{s}}$ for any sort $\mathtt{s}$ in Σ_s

The function symbols are interpreted as follows[3]:

- For any g in $\Sigma_t \cup C_t$, $\mathcal{M}[g] = \mathcal{A}[g]$
- For any g in $\Sigma_s \cup C$, $\mathcal{M}[g] = \mathcal{H}[g]$
- For any $\mathtt{struct}$-term t, $\mathcal{M}[f]([\![t]\!]) = \mathcal{A}[f(t)\downarrow_F]$

$\mathcal{M}$ is well-defined due to the assumption that F is E-compatible in $\mathcal{A}$. Let us check that $\mathcal{M}$ is a model of $\mathfrak{T}_s \cup \mathfrak{T}_F \cup \mathfrak{T}_t$.

- $\mathcal{M}^{\Sigma_s \cup C} = \mathcal{H}$, which is a model of $\mathfrak{T}_s$ by assumption.
- $\mathcal{M}^{\Sigma_t \cup C_t} = \mathcal{A}$, which is a model of $\mathfrak{T}_t$ by assumption.
- For any $\mathtt{struct}$-term t, we have that

$$\mathcal{M}[f(t)] = \mathcal{M}[f]([\![t]\!]) = \mathcal{A}[f(t)\downarrow_F] = \mathcal{M}[f(t)\downarrow_F]$$

by definition of $\mathcal{M}$. Therefore $\mathcal{M}$ is a model of $\mathfrak{T}_F$.

Consider now the case $S \neq \emptyset$. By Assumption 1, T_s is smooth with respect to $\mathtt{Elem}$, and more precisely there exists also a larger model $\mathcal{H}$ of $\mathfrak{T}_s$ such that $\mathcal{H}^{\Sigma \cup C}$ is $T(\Sigma \cup C \cup D)/=_E$, where

- D is a set of elements of sort in $S \subseteq \mathtt{Elem}$,
- $H_\sigma = A_\sigma$ for each sort $\sigma \in S$.

Then the construction of $\mathcal{M}$ follows directly from the case $S = \emptyset$. In particular, $\mathcal{M}$ is well-defined on $C \cap C_t$ due to the arrangement Arr_t. □

The missing piece of the method is to provide a way to check E-compatibility of F in a model of $\mathfrak{T}_t$. In the following, we show that this property can be reduced to a $\mathfrak{T}_t$-satisfiability problem.

Proposition 5. *F is E-compatible in a model of $\mathfrak{T}_t$ if the theory $\mathfrak{T}_t \cup Arr_t \cup CP_E$ is consistent, where CP_E is the target encoding of E (Definition 7).*

Proof. Let $\mathcal{A}$ be a model of $\mathfrak{T}_t \cup Arr_t \cup CP_E$. Let R be the convergent term rewrite system associated to E. Since $\mathcal{A}$ satisfies Arr_t, we have that $\mathcal{A}[e\downarrow_R] = \mathcal{A}[e]$ for any constant e of sort in $\Sigma_s \cap \Sigma_t$. We first prove by structural induction that for any $\mathtt{struct}$-term u, $\mathcal{A}[f(u\downarrow_R)\downarrow_F] = \mathcal{A}[f(u)\downarrow_F]$.

(Inductive case) Assume $u = c(\boldsymbol{e}; u_1, \ldots, u_n)$.

[3] For any $\mathtt{struct}$-term t, $[\![t]\!]$ is the equivalence class of t modulo $=_E$.

– If $c(\boldsymbol{e}; u_1, \ldots, u_n)\downarrow_R = c(\boldsymbol{e}\downarrow_R; u_1\downarrow_R, \ldots, u_n\downarrow_R)$, then we have:

$$\begin{aligned}
&\mathcal{A}[f(c(\boldsymbol{e}; u_1, \ldots, u_n)\downarrow_R)\downarrow_F] \\
&= \mathcal{A}[f(c(\boldsymbol{e}\downarrow_R; u_1\downarrow_R, \ldots, u_n\downarrow_R))\downarrow_F] \\
&= \mathcal{A}[f_c(\boldsymbol{e}\downarrow_R; f(u_1\downarrow_R)\downarrow_F, \ldots, f(u_n\downarrow_R)\downarrow_F)] \\
&= f_c(\mathcal{A}[\boldsymbol{e}\downarrow_R]; \mathcal{A}[f(u_1\downarrow_R)\downarrow_F], \ldots, \mathcal{A}[f(u_n\downarrow_R)\downarrow_F)] \\
&= f_c(\mathcal{A}[\boldsymbol{e}]; \mathcal{A}[f(u_1\downarrow_R)\downarrow_F], \ldots, \mathcal{A}[f(u_n\downarrow_R)\downarrow_F)] \\
&= f_c(\mathcal{A}[\boldsymbol{e}]; \mathcal{A}[f(u_1)\downarrow_F], \ldots, \mathcal{A}[f(u_n)\downarrow_F]) \\
&= \mathcal{A}[f_c(\boldsymbol{e}; f(u_1)\downarrow_F, \ldots, f(u_n)\downarrow_F)] \\
&= \mathcal{A}[f(c(\boldsymbol{e}; u_1, \ldots, u_n))\downarrow_F]
\end{aligned}$$

– Otherwise, $c(\boldsymbol{e}; u_1, \ldots, u_n)\downarrow_R$ is necessarily a constant x', and $u_1, \ldots, u_n$ are constants $x_1, \ldots, x_n$. Then, by assumption on $\mathcal{A}$, we have

$$\mathcal{A}[f(x')\downarrow_F] = \mathcal{A}[f_{x'}] = \mathcal{A}[f_c(\boldsymbol{e}; f_{x_1}, \ldots, f_{x_n})] = \mathcal{A}[f(c(\boldsymbol{e}; x_1, \ldots, x_n))\downarrow_F]$$

(Base case) Assume u is a constant x. If $x\downarrow_R = x$, then we have $f(x\downarrow_R)\downarrow_F = f(x)\downarrow_F$, and so $\mathcal{A}[f(x\downarrow_R)\downarrow_F] = \mathcal{A}[f(x)\downarrow_F]$. Otherwise, we have $x\downarrow_R = x'$. Then, by assumption on $\mathcal{A}$, we have $\mathcal{A}[f(x')\downarrow_F] = \mathcal{A}[f_{x'}] = \mathcal{A}[f_x] = \mathcal{A}[f(x)\downarrow_R]$.

To conclude the proof, let s and t be any `struct`-terms. If $s =_E t$, then $s\downarrow_R = t\downarrow_R$ and $\mathcal{A}[f(s)\downarrow_F] = \mathcal{A}[f(s\downarrow_R)\downarrow_F] = \mathcal{A}[f(t\downarrow_R)\downarrow_F] = \mathcal{A}[f(t)\downarrow_F]$. This means F is E-compatible in the model $\mathcal{A}$ of $\mathfrak{T}_t$. □

Example 4. (Example 3 continued). Let $\mathfrak{T}_t$ be the theory of integers. We have $E = \{a = cons(e, b), b = cons(e', c), c = nil\}$ and so $CP_E = \{\ell_a = 1 + \ell_b, \ell_b = 1 + \ell_c, \ell_c = 0\}$. Since $\mathfrak{T}_t \cup CP_E$ is consistent, we get the consistency of $\mathfrak{T}_s \cup \mathfrak{T}_F \cup \mathfrak{T}_t$ by applying Proposition 5 and then Proposition 4. ■

As a side remark, in the trivial case of $F = \{f(x_k) \to f_{x_k}\}_{k\in K}$, the combination becomes disjoint, and the consistency of $\mathfrak{T}_s \cup \mathfrak{T}_F \cup \mathfrak{T}_t$ corresponds to the consistency of the union of three disjoint theories, including the theory of equality for f.

Proposition 4 and Proposition 5 are instrumental to prove the completeness of the combination procedure. We thus get this result, subsuming Lemma 1:

Lemma 2. *Let $T = T_s \cup T_f \cup T_t$, where T_s, T_t follow Assumption 1 and T_f is a bridging theory according to Definition 5. The combination procedure introduced in Lemma 1 is sound and complete for T-satisfiability.*

Proof. The soundness is straightforward just like in Lemma 1. Let us focus on the completeness. Consider the separate form φ and the sets of variables V and V_t given in Lemma 1. By viewing φ as a set of ground literals in a constant expansion of $\Sigma_s \cup \Sigma_f \cup \Sigma_t$, we can introduce the same theories $\mathfrak{T}_s$, $\mathfrak{T}_t$ and $\mathfrak{T}_F$ as in Assumption 2 and Proposition 3:

– the $\Sigma_s \cup C$-theory $\mathfrak{T}_s$ is $T_s \cup \varphi_{struct} \cup \varphi_{elem} \cup Arr_t \cup Arr_{\texttt{struct}}$,
– the $\Sigma_t \cup C_t$-theory $\mathfrak{T}_t$ is $T_t \cup \varphi_t$,
– $\mathfrak{T}_F = T_f \cup \varphi_f \cup \bigcup_{x:\texttt{struct}\in C\setminus V}\{f(x) = f_x\}$,

where C and C_t are as follows:

- $C = V$ when $T_s \in \mathbf{DST}$. Otherwise, C is equal to V plus one fresh constant for each sort in T_s.
- $C_t = V_t \cup \bigcup_{x:\mathtt{struct} \in C \setminus V} \{f_x\}$.

Assume $\varphi_{struct} \cup \varphi_{elem} \cup Arr_t \cup Arr_{\mathtt{struct}}$ is T_s-satisfiable and $\varphi_t \cup Arr_t \cup CP_E$ is T_t-satisfiable. Equivalently, $\mathfrak{T}_s$ and $\mathfrak{T}_t$ are consistent. By applying Proposition 4 and Proposition 5, we get that $\mathfrak{T}_s \cup \mathfrak{T}_F \cup \mathfrak{T}_t$ is consistent, and so $T_s \cup T_f \cup T_t \cup \varphi$ is consistent, or equivalently, φ is T-satisfiable. □

6 Conclusion

In this paper, we present a combination method to solve the satisfiability problem in some particular non-disjoint union of three theories including a source, a target and a bridging theory from the source to the target. The combination method is sound and complete for a large class of source data structure theories, ranging from the theory of equality to the theory of absolutely free data structures. For all these axiomatized theories, the satisfiability problem can be solved by using an off-the-shelf equational theorem prover.

We envision several further investigations. First, we would like to consider the case of non-absolutely free constructors, e.g., associative-commutative constructors, to allow a more general congruence relation E in the definition of a data structure theory. Second, it would be interesting to allow non-convex data structure theories, such as the theory of possibly empty lists [1]. Third, to go beyond the considered bridging axioms, a natural continuation is to identify other "simple" connecting axioms that could be compiled into a combination method à la Nelson-Oppen.

References

1. A. Armando, M. P. Bonacina, S. Ranise, and S. Schulz. New results on rewrite-based satisfiability procedures. *ACM Trans. Comput. Log.*, 10(1), 2009.
2. A. Armando, S. Ranise, and M. Rusinowitch. A rewriting approach to satisfiability procedures. *Inf. Comput.*, 183(2):140–164, 2003.
3. F. Baader and S. Ghilardi. Connecting many-sorted theories. *J. Symb. Log.*, 72(2):535–583, 2007.
4. F. Baader and T. Nipkow. *Term rewriting and all that.* Cambridge University Press, 1998.
5. L. Bachmair and H. Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *J. Log. Comput.*, 4(3):217–247, 1994.
6. C. Barrett, I. Shikanian, and C. Tinelli. An abstract decision procedure for a theory of inductive data types. *JSAT*, 3(1-2):21–46, 2007.
7. P. Baumgartner and U. Waldmann. Hierarchic superposition with weak abstraction. In *Automated Deduction - CADE-24 - 24th International Conference on Automated Deduction, Lake Placid, NY, USA*, volume 7898 of *LNCS*, pages 39–57. Springer, 2013.

8. P. Chocron, P. Fontaine, and C. Ringeissen. A Gentle Non-Disjoint Combination of Satisfiability Procedures. In S. Demri, D. Kapur, and C. Weidenbach, editors, *Proc. of the 7th International Joint Conference on Automated Reasoning, IJCAR*, volume 8562 of *LNCS*, pages 122–136. Springer, 2014.
9. P. Chocron, P. Fontaine, and C. Ringeissen. A Polite Non-Disjoint Combination Method: Theories with Bridging Functions Revisited. In A. P. Felty and A. Middeldorp, editors, *Proc. Conference on Automated Deduction (CADE)*, volume 9195 of *LNCS*, pages 419–433. Springer, 2015.
10. P. Fontaine. Combinations of theories for decidable fragments of first-order logic. In S. Ghilardi and R. Sebastiani, editors, *Frontiers of Combining Systems (FroCoS)*, volume 5749 of *LNCS*, pages 263–278. Springer, 2009.
11. S. Ghilardi. Model-theoretic methods in combined constraint satisfiability. *Journal of Automated Reasoning*, 33(3-4):221–249, 2004.
12. D. Jovanovic and C. Barrett. Polite theories revisited. In C. Fermueller and A. Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'10)*, volume 6397 of *LNCS*, pages 402–416. Springer, 2010.
13. E. Nicolini, C. Ringeissen, and M. Rusinowitch. Combinable extensions of Abelian groups. In R. A. Schmidt, editor, *Proc. Conference on Automated Deduction (CADE)*, volume 5663 of *LNCS*, pages 51–66. Springer, 2009.
14. E. Nicolini, C. Ringeissen, and M. Rusinowitch. Combining satisfiability procedures for unions of theories with a shared counting operator. *Fundam. Inform.*, 105(1-2):163–187, 2010.
15. S. Ranise, C. Ringeissen, and C. G. Zarba. Combining data structures with non-stably infinite theories using many-sorted logic. In B. Gramlich, editor, *Frontiers of Combining Systems (FroCoS)*, volume 3717 of *LNCS*, pages 48–64. Springer, 2005.
16. V. Sofronie-Stokkermans. Locality results for certain extensions of theories with bridging functions. In R. A. Schmidt, editor, *Proc. Conference on Automated Deduction (CADE)*, volume 5663 of *LNCS*, pages 67–83. Springer, 2009.
17. V. Sofronie-Stokkermans. Automated reasoning in extensions of theories of constructors with recursively defined functions and homomorphisms. In T. Ball, J. Giesl, R. Hähnle, and T. Nipkow, editors, *Interaction versus Automation: The two Faces of Deduction*, number 09411 in Dagstuhl Seminar Proceedings. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2010.
18. P. Suter, M. Dotta, and V. Kuncak. Decision procedures for algebraic data types with abstractions. In M. V. Hermenegildo and J. Palsberg, editors, *Principles of Programming Languages (POPL)*, pages 199–210. ACM, 2010.
19. C. Tinelli and C. Ringeissen. Unions of non-disjoint theories and combinations of satisfiability procedures. *Theoretical Comput. Sci.*, 290(1):291–353, Jan. 2003.
20. C. Tinelli and C. G. Zarba. Combining non-stably infinite theories. *Journal of Automated Reasoning*, 34(3):209–238, Apr. 2005.
21. D. Tran, C. Ringeissen, S. Ranise, and H. Kirchner. Combination of convex theories: Modularity, deduction completeness, and explanation. *J. Symb. Comput.*, 45(2):261–286, 2010.
22. T. Zhang, H. B. Sipma, and Z. Manna. Decision procedures for term algebras with integer constraints. *Inf. Comput.*, 204(10):1526–1574, 2006.