CDSAT for Predicate-Sharing Theories: Arrays, Maps, and Vectors with Abstract Domain

Maria Paola Bonacina[1*], Stéphane Graham-Lengrand[2] and Natarajan Shankar[3]

[1*]Dipartimento di Informatica, Università degli Studi di Verona, Strada Le Grazie 15, Verona, 37134, VR, Italy.
[2]Computer Science Laboratory, SRI International, 333 Ravenswood Avenue, Menlo Park, 94025, CA, USA.
[3]Computer Science Laboratory, SRI International, 333 Ravenswood Avenue, Menlo Park, 94025, CA, USA.

*Corresponding author(s). E-mail(s): mariapaola.bonacina@univr.it;
Contributing authors: stephane.graham-lengrand@csl.sri.com; shankar@csl.sri.com;

Abstract

The theory of arrays is a mainstay of reasoning about programs by satisfiability modulo theories (SMT). Arrays are finite in programs, but infinite in the theory. This discrepancy was approached by recurring to quantifiers or quitting arrays for sequences, jeopardizing decidability. We offer a new solution that is quantifier-free, uses arrays, and preserves decidability. We enrich the theory of arrays with the *length* function and a new *admissibility* predicate, whose definition is the *abstract domain* of the array. Thanks to the abstraction, indices and lengths do not have to be integers. Variants of the new theory axiomatize *maps* and *vectors*, providing the first treatment of these *dynamic* data structures in SMT. Since length is a *bridging function*, deciding satisfiability in each of these theories is an instance of *nondisjoint theory combination*. We extend the SMT method CDSAT (*Conflict-Driven Satisfiability*) from disjoint to *predicate-sharing* theories. We equip the new theories with *theory modules* that satisfy the CDSAT requirements for *termination* and *completeness*. We show that the CDSAT framework, the construction of a *global basis* for termination, and the CDSAT *completeness theorem* generalize gracefully from disjoint to predicate-sharing theories.

Keywords: CDSAT, Satisfiability modulo theories, Satisfiability modulo assignment, Theory combination, Bridging functions, Arrays, Maps, Vectors

1 Introduction

Reasoning about programs is a main application of automated reasoning. A key ingredient is the ability to reason in theory combinations including theories of data structures. A stalwart component is the theory of *arrays*. Its simplest presentation [33] describes the interplay of two operations: *selecting* (or *reading*) the value at an index of an array, and *storing* (or *writing*) a value at an index of an array. The theory of arrays with *extensionality* [45], which also defines array equality, is the standard presentation. Due to the fundamental nature of the *read* and *write* operations in computing, the theory of arrays is deemed useful also to model the computer memory (e.g., the heap).

While the theory of arrays is undecidable, its quantifier-free fragment is decidable [14, 15, 45]. Theory combinations including this fragment are handled by reasoning methods such as combination schemes (e.g., [13, 35, 36, 47]), superposition [3, 7, 8], and CDSAT (*Conflict-Driven Satisfiability*) [6, 9, 11]. These approaches to satisfiability modulo theories assume that the theories are *disjoint*. Disjoint theories share only sorts and equality predicates on shared sorts. *Nondisjoint* theories share also other symbols.

In most programming languages an array is *finite*, and its *length* is the number of the values it contains. The indices are non-negative integers, and if the length is n the indices are in the $[0, n)$ interval. In the standard theory of arrays, all arrays have implicitly the same length given by the cardinality of the interpretation of the sort of indices. Thus, arrays can be infinite, and integer-indexed arrays are. Also, the theory does not distinguish whether a store operation is within bounds or not. The *array property fragment* allows one to define the *bounded equality* of integer-indexed arrays, by a limited usage of quantifiers that preserves decidability [14, 15, 23]. Other approaches use theories of finite integer-indexed sequences to model finite integer-indexed arrays [1, 2, 43]. The *array property fragment with concatenation* combines limited quantification with finite integer-indexed sequences [48].

We aim at improving the *theory of arrays itself*, in order to get closer to data structures as they are in programs. By viewing the problem as *theory combination*, we present a new solution with several advantages. The theory of arrays becomes more expressive than any of its existing versions. Our approach is abstract, and hence not limited to integer-indexed arrays, and it also models *maps* and *vectors*. The quantifier-free fragment suffices and it is decidable. Quantifier reasoning is not needed.

The first step is to enrich the signature of the theory of arrays with a *length* function. However, length is a *bridging function* [16, 44], as it leads to write *bridging axioms* [22], that make the theories nondisjoint. For example, the extensionality axiom saying that integer-indexed arrays a and b are equal, if they have the same length n (also an integer) and the same value at all indices in the $[0, n)$ interval

$$\forall a,\ b.\ [\ \mathsf{len}(a) \simeq \mathsf{len}(b) \wedge (\forall i.\ 0 \leq i < \mathsf{len}(a) \rightarrow \mathsf{select}(a, i) \simeq \mathsf{select}(b, i))\ ] \rightarrow a \simeq b$$

forces the theory of arrays to share symbols $\leq$ and $<$ with the theory of integers. This problem arises also in recursive (or absolutely free) data structures (e.g., lists or trees), when defining *length*, *size*, or *depth* in terms of the constructors [16].

The key step is to enrich the signature of the theory of arrays with a new *admissibility* predicate, named Adm, which remains free in the theory of arrays and is defined by another theory $\mathcal{T}$. The $\mathcal{T}$-axiom defining Adm says when an index is *admissible* with respect to a length. The admissibility property is the *abstract domain* of the arrays in the resulting *theory of arrays with abstract domain*, denoted ArrAD. While an array is interpreted as a function, an array with abstract domain is interpreted as a *partial function*, whose (concrete) domain of definition is the set of *admissible indices*.[1] For extensionality, two arrays with abstract domains are equal, if they have the same length and the same values at all admissible indices. Theories ArrAD and $\mathcal{T}$ share the symbol Adm and the sorts of its arguments, indices and lengths.

Admissibility abstracts the concept of index within bounds, making the theory flexible and expressive: the length of an array can be a nonnegative integer, but does not have to be; indices can be integers, but can be interpreted differently; the theory $\mathcal{T}$ defining Adm does not have to be a theory of the integers; and domains of definition are not necessarily linearly ordered intervals. Domain and length of an array may be finite or infinite, depending on Adm's definition. The finiteness of arrays as in programming languages is captured, without renouncing the generality that allows arrays to be infinite.

In theory ArrAD, a store at an inadmissible index leaves the array unchanged: the domain is *static*. We present two variants where the domain, and hence the data structure, is *dynamic*. In the theory MapAD of *maps with abstract domain*, a store at an inadmissible index makes it admissible. In the theory VecAD of *vectors with abstract domain*, a store at an inadmissible index makes that index and all the smaller ones admissible. Similar to ArrAD, theory MapAD shares the Adm predicate, while VecAD shares Adm and an ordering on indices, which does not need to be linear.

In order to reason in ArrAD, MapAD, or VecAD, we *generalize CDSAT from disjoint to predicate-sharing theories*, meaning theories that may share predicate symbols other than equality. CDSAT decides the satisfiability of an input formula modulo a union of theories and possibly an input assignment of values to Boolean or first-order terms. Such problems arise in optimization [19], parallelization [28], interpolation [30], and quantifier reasoning [12]. A solution is a model of the union of the theories that satisfies the input formula and includes the initial assignment. CDSAT orchestrates *theory modules*, one per theory, to perform a *conflict-driven search* to find a solution or report that none exists. A theory module is an abstraction of a theory reasoning procedure. For *soundness*, *termination*, and *completeness*, a theory module is an inference system.

A theory module may need to generate *new* (i.e., non-input) terms, in order to explain theory conflicts. Termination requires that only finitely many new terms can be generated. A *local basis* is a function that maps a given set of terms (e.g., all those occurring in the input) to the *finite* set of terms that the theory module may produce. The *soundness* of CDSAT requires that the theory modules are *sound*. *Termination* requires that there exists a *global basis*, that must be *finite* and yet contain all the terms that can be produced during a CDSAT derivation. We define sound theory modules and a local basis for ArrAD, MapAD, and VecAD, and we generalize the construction of a global basis from the local ones from disjoint [11] to predicate-sharing theories.

[1] This usage is coherent with that of "abstract domain" and "concrete domain" in the literature (e.g., [38]), where a concrete domain is a set of values, and an abstract domain is a set of properties, that is, a set of formulas, so that an abstract domain in our sense is a singleton abstract domain.

The *completeness* of CDSAT employs the concept of a *leading theory*, say $\mathcal{T}_1$, which may be one of the theories in the union or a theory that only exists in principle. $\mathcal{T}_1$ has the information shared by any two component theories. It suffices that each theory agrees with $\mathcal{T}_1$ on the shared information to have an agreement among all theories. Disjoint theories only need to agree on equalities between shared terms and cardinalities of shared sorts. Thus, $\mathcal{T}_1$ has all the sorts in the union and aggregates all the cardinality constraints on shared sorts [9, 11]. In the predicate-sharing case, $\mathcal{T}_1$ has also all the predicate symbols (e.g., Adm) shared by any two theories. The agreement between a component theory and $\mathcal{T}_1$ is guaranteed by the *leading-theory completeness* of the module of the component theory. We prove that the theory modules of ArrAD, MapAD, and VecAD are *leading-theory complete*. Then, we generalize the CDSAT completeness theorem from disjoint [9] to predicate-sharing theories. It follows that for theories ArrAD, MapAD, and VecAD, the quantifier-free fragment is decidable.

This article is organized as follows. After basic definitions and notations (Sect. 2), we present the theories ArrAD, MapAD, and VecAD (Sect. 3). Section 4 updates the CDSAT framework to accommodate shared predicates. In Section 5 we define theory modules and local basis for ArrAD, MapAD, and VecAD, proving that the local basis is finite and the theory modules are leading-theory complete. Sections 6 and 7 are devoted to the generalized global basis construction and the generalized CDSAT completeness theorem, respectively. Sections 8 and 9 contain comparison with related work and final discussion, respectively. Part of this research was presented in preliminary form at a workshop [10], where the temporary name "arrays with abstract length" was used for arrays with abstract domain.

2 Preliminaries

A *signature* Σ is a triple (S, F, ShF), where S is the set of *sorts*, F is the set of *symbols*, and $\mathit{ShF} \subseteq F$ is the subset of *shared symbols*. The set S of every signature includes the sort prop of Booleans. All symbols are *sorted*: for $f \in F$, the notation $f\colon (s_1 \times \cdots \times s_m) \to s$ says that f has arity m, input sorts $s_1, \ldots, s_m$ $(m \geq 0)$ and output sort s. A symbol is a constant if $m = 0$, a predicate if the output sort is prop, and a function otherwise. Equalities are written in infix notation. The set $\simeq_S = \{\simeq_s : s \times s \to \mathsf{prop} \mid s \in S\}$ of the equality symbols for all the sorts in S is a subset of F.

Given a collection $\mathcal{V} = (\mathcal{V}^s)_{s\in S}$ of disjoint sets of *variables*, where $\mathcal{V}^s$ is the set of variables of sort s, the symbol $\mathcal{V}$ also denotes the disjoint union $\biguplus_{s\in S} \mathcal{V}^s$, and $\Sigma[\mathcal{V}]$-terms are defined as usual. We use t and u for $\Sigma[\mathcal{V}]$-terms, l for $\Sigma[\mathcal{V}]$-terms of sort prop (i.e., $\Sigma[\mathcal{V}]$-formulæ), and $\mathit{top}(t)$ for the top symbol of t. We write $t\colon s$ if t has sort s, $u \trianglelefteq t$ if u is a *subterm* of t, and $u \triangleleft t$ if $u \trianglelefteq t$ and $u \neq t$. A $\Sigma[\mathcal{V}]$-*interpretation* $\mathcal{M}$ is defined as usual, with $s^{\mathcal{M}}$, $x^{\mathcal{M}}$, $f^{\mathcal{M}}$, and $\mathcal{M}(t)$, denoting the $\mathcal{M}$-interpretation of a sort s, a variable x, a symbol f, and a $\Sigma[\mathcal{V}]$-term t, respectively. Domain $s^{\mathcal{M}}$ is not empty for all sorts s, and $\mathsf{prop}^{\mathcal{M}} = \{\mathsf{true}, \mathsf{false}\}$. A Σ-*structure* is a $\Sigma[\emptyset]$-interpretation.

A *theory* $\mathcal{T}$ is defined by a signature Σ and an axiomatization $\mathcal{A}$ which is a set of Σ-sentences, the $\mathcal{T}$-*axioms*, that define symbols in Σ. Symbols that appear in $\mathcal{T}$-axioms are *defined* or *interpreted*, those that do not are *free* or *uninterpreted*. The theory $\mathcal{T}$ is also understood as the class of Σ-structures that satisfy $\mathcal{A}$, called models of $\mathcal{T}$ or *$\mathcal{T}$-models.* A $\mathcal{T}$-term is a $\Sigma[\mathcal{V}]$-term if Σ is the signature of theory $\mathcal{T}$. A *$\mathcal{T}[\mathcal{V}]$-model* is a $\Sigma[\mathcal{V}]$-interpretation that is a $\mathcal{T}$-model when the interpretation of variables is ignored. When working with a single theory we may simply write term and model.

Given theories $\mathcal{T}_1, \ldots, \mathcal{T}_n$, with signatures $\Sigma_k = (S_k, F_k, \mathit{ShF}_k)$ and axiomatizations $\mathcal{A}_k$, for $k = 1, \ldots, n$, their union, named $\mathcal{T}_\infty$, is the theory with signature $\Sigma_\infty = (S_\infty, F_\infty, \mathit{ShF}_\infty)$, for $S_\infty = \bigcup_{k=1}^n S_k$, $F_\infty = \bigcup_{k=1}^n F_k$, and $\mathit{ShF}_\infty = \bigcup_{k=1}^n \mathit{ShF}_k$, and axiomatization $\mathcal{A}_\infty = \bigcup_{k=1}^n \mathcal{A}_k$. The above notations can be specialized for any $\mathcal{T}_k$, so that we have $\mathcal{V}_k = (\mathcal{V}^s)_{s \in S_k}$ for variables, $\Sigma_k[\mathcal{V}_k]$-terms, and $\mathcal{T}_k[\mathcal{V}_k]$-models. Similarly, for $\mathcal{T}_\infty$ we have $\mathcal{V}_\infty = (\mathcal{V}^s)_{s \in S_\infty}$ for variables, $\Sigma_\infty[\mathcal{V}_\infty]$-terms, and $\mathcal{T}_\infty[\mathcal{V}_\infty]$-models. We use as much as possible the generic $\mathcal{T}$ (and hence $\mathcal{V}$ and Σ), specifying in the context whether we mean any $\mathcal{T}_k$, or $\mathcal{T}_\infty$, or either one. Unless otherwise stated, variables are variables in $\mathcal{V}_\infty$, terms are $\mathcal{T}_\infty$-terms, and models are $\mathcal{T}_\infty[\mathcal{V}_\infty]$-models.

If Σ is a signature with $F \subset F_\infty$, a symbol in $F_\infty \setminus F$ is a *Σ-foreign* symbol. A subterm u of a term t is a *Σ-foreign* subterm if $\mathit{top}(u)$ is Σ-foreign. The following definition regards $\lhd$-maximal occurrences of non-variable Σ-foreign subterms as variables, without replacing them explicitly with new variables.

Definition 1 (Free Σ-variables)**.** *Let $\Sigma = (S, F)$ be a signature such that $F \subseteq F_\infty$, and t a $\Sigma_\infty[\mathcal{V}_\infty]$-term. For all sorts $s \in S$, the set $\mathsf{fv}^s_\Sigma(t)$ of* free Σ-variables *of sort s occurring in term t is given by*

$$\begin{array}{lll}
\mathsf{fv}^s_\Sigma(x) & = \{x\} & \text{if } x \in \mathcal{V}^s_\infty, \\
\mathsf{fv}^s_\Sigma(x) & = \emptyset & \text{if } x \in \mathcal{V}^r_\infty \text{ for a sort } r \in S,\ r \neq s \\
\mathsf{fv}^s_\Sigma(f(t_1, \ldots, t_m)) & = \bigcup_{i=1}^m \mathsf{fv}^s_\Sigma(t_i) & \text{if } f \in F, \\
\mathsf{fv}^s_\Sigma(f(t_1, \ldots, t_m)) & = \emptyset & \text{if } f \notin F \text{ with output sort } r,\ r \neq s, \text{ and} \\
\mathsf{fv}^s_\Sigma(f(t_1, \ldots, t_m)) & = \{f(t_1, \ldots, t_n)\} & \text{if } f \notin F \text{ with output sort } s,
\end{array}$$

where the last case adds Σ-foreign *terms.*

Then $\mathsf{fv}_\Sigma(t) = \bigcup_{s \in S} \mathsf{fv}^s_\Sigma(t)$, and for a set X of terms $\mathsf{fv}^s_\Sigma(X) = \{u \mid u \in \mathsf{fv}^s_\Sigma(t), t \in X\}$ and $\mathsf{fv}_\Sigma(X) = \{u \mid u \in \mathsf{fv}_\Sigma(t), t \in X\}$. If Σ is Σ_∞, we write $\mathsf{fv}(t)$ and $\mathsf{fv}(X)$: as there are no Σ_∞-foreign symbols, $\mathsf{fv}(t) \subseteq \mathcal{V}_\infty$ and $\mathsf{fv}(X) \subseteq \mathcal{V}_\infty$.

Theories $\mathcal{T}_1, \ldots, \mathcal{T}_n$ are *disjoint* if the only shared symbols are the equality symbols of shared sorts: for all j and k, $1 \leq j \neq k \leq n$, $\mathit{ShF}_j \cap \mathit{ShF}_k = \simeq_Q$ for $Q = S_j \cap S_k$. Otherwise, $\mathcal{T}_1, \ldots, \mathcal{T}_n$ are *nondisjoint.* The union $\mathcal{T}_\infty$ is a *disjoint union*, if $\mathcal{T}_1, \ldots, \mathcal{T}_n$ are disjoint, and a *nondisjoint union* otherwise. A union $\mathcal{T}_\infty$ is a *predicate-sharing union* if either it is a disjoint union, or for some j and k, $1 \leq j \neq k \leq n$, $\mathit{ShF}_j \cap \mathit{ShF}_k = \simeq_Q \uplus \{p \mid p\colon (s_1 \times \cdots \times s_m) \rightarrow \mathsf{prop},\ \forall i, 1 \leq i \leq m, s_i \in Q\}$, where $Q = S_j \cap S_k$. In a predicate-sharing union the theories may share predicate symbols other than equality.

3 Theories of Data Structures with Abstract Domain

The simplest theory of arrays, that we denote Arr_0, is the theory of *arrays without extensionality*, which only has the *select-over-store* axioms:

$$\forall a, v, i.\ \mathsf{select}(\mathsf{store}(a, i, v), i) \simeq v, \tag{1}$$

$$\forall a, v, i, j.\ i \not\simeq j \rightarrow \mathsf{select}(\mathsf{store}(a, i, v), j) \simeq \mathsf{select}(a, j), \tag{2}$$

where a is a variable of sort array, v is a variable of sort value, and i and j are variables of sort index. This theory does not define when two arrays are equal. The theory of *arrays with extensionality*, that we denote Arr, adds to axioms (1)-(2) an *extensionality axiom* saying that two arrays are equal if they have the same values at *all indices*:

$$\forall a, b.\ (\forall i.\ \mathsf{select}(a, i) \simeq \mathsf{select}(b, i)) \rightarrow a \simeq b. \tag{3}$$

In previous work, we presented the theory Arr as featuring a set of basic sorts and an array sort constructor $\Rightarrow$, meaning that $s_1 \Rightarrow s_2$ is the sort of arrays with indices of sort s_1 and values of sort s_2 [9, 11]. The set of sorts of Arr was defined as the free closure of the set of basic sorts with respect to $\Rightarrow$. In this way, the theory may have multiple sorts for indices, values, and arrays. Then, we used $s_1 \overset{s_3}{\Rightarrow} s_2$ for the sort of arrays with indices of sort s_1, values of sort s_2, and lengths of sort s_3 [10]. Here we adopt a simpler approach: there is one sort for indices, one for values, one for lengths, and hence one for arrays. There is no loss of generality, because a theory of arrays with n array sorts ($n > 1$), possibly including arrays of arrays, can be viewed as the union of n theories with one array sort each.

3.1 The Theory of Arrays with Abstract Domain

The theory ArrAD of *arrays with abstract domain* has signature $\Sigma_{\mathsf{ArrAD}} = (S_{\mathsf{ArrAD}}, F_{\mathsf{ArrAD}}, \mathit{ShF}_{\mathsf{ArrAD}})$. The set S_{ArrAD} of sorts contains prop, the sort I of *indices*, the sort V of *values*, the sort L of *lengths*, and the sort A of *arrays* with indices of sort I, values of sort V, and length of sort L, with the proviso that $\mathsf{A} \neq \mathsf{I}$, $\mathsf{A} \neq \mathsf{V}$, $\mathsf{A} \neq \mathsf{L}$, $\mathsf{A} \neq \mathsf{prop}$, and $\mathsf{L} \neq \mathsf{prop}$.

Example 1. *In order to model arrays of arrays with indices of a sort* I*, values of a sort* V*, and lengths of a sort* L*, take the union of two instances* ArrAD_1 *and* ArrAD_2 *of* ArrAD*. Theory* ArrAD_1 *has sorts* $\mathsf{I}_1 = \mathsf{I}$*,* $\mathsf{V}_1 = \mathsf{V}$*, and* $\mathsf{L}_1 = \mathsf{L}$*, so that* A_1 *is the sort of arrays with indices of sort* I*, values of sort* V*, and lengths of sort* L*. Theory* ArrAD_2 *has sorts* $\mathsf{I}_2 = \mathsf{I}$*,* $\mathsf{V}_2 = \mathsf{A}_1$*, and* $\mathsf{L}_2 = \mathsf{L}$*, so that* A_2 *is the desired sort of arrays of arrays.*

The set F_{ArrAD} contains function symbols $\mathsf{select}\colon \mathsf{A} \times \mathsf{I} \rightarrow \mathsf{V}$ for *select* or *read*, $\mathsf{store}\colon \mathsf{A} \times \mathsf{I} \times \mathsf{V} \rightarrow \mathsf{A}$ for *store* or *write*, $\mathsf{len}\colon \mathsf{A} \rightarrow \mathsf{L}$ that maps an array a to its length $\mathsf{len}(a)$, and a new predicate symbol $\mathsf{Adm}\colon \mathsf{I} \times \mathsf{L} \rightarrow \mathsf{prop}$ which is in the $\mathit{ShF}_{\mathsf{ArrAD}}$ subset of shared symbols. If i is a term of sort I and a is a term of sort A, then $\mathsf{Adm}(i, \mathsf{len}(a))$ is true iff index i is *admissible* with respect to $\mathsf{len}(a)$.

Example 2. *The theories* ArrAD_1 *and* ArrAD_2 *of Ex. 1 share the sorts* $\{\mathsf{prop}, \mathsf{I}, \mathsf{L}, \mathsf{A}_1\}$ *and the symbol* Adm*, so that their union is a predicate-sharing union.*

Whenever a union of theories includes ArrAD, it also includes another theory, say $\mathcal{T}$, with signature $\Sigma = (S, F, \mathit{ShF})$, such that $\mathsf{I} \in S$, $\mathsf{L} \in S$, and $\mathsf{Adm} \in \mathit{ShF}$. Theory $\mathcal{T}$ shares with ArrAD the symbol Adm and the sorts I and L (sharing sort V is unnecessary), and it defines admissibility.

Example 3. *Let $\mathcal{T}$ be linear integer arithmetic, or* LIA *for short, interpreting both sorts* I *of indices and* L *of lengths as the set $\mathbb{Z}$ of the integers, and defining* Adm *with the axiom* $\forall i, n.\ \mathsf{Adm}(i, n) \leftrightarrow 0 \leq i < n$. *If $n \leq 0$, no index is admissible. If $n > 0$, the set of admissible indices is the $[0, n)$ interval. The set of admissible indices is* finite.

The interpretation in Example 3 is a popular choice, but the abstraction provided by the notion of admissibility makes other choices possible.

Example 4. *Suppose that theory $\mathcal{T}$ interprets sort* I *as a set D, sort* L *as the powerset $\mathcal{P}(D)$ of D, and defines* Adm *with the axiom* $\forall i, C.\ \mathsf{Adm}(i, C) \leftrightarrow i \in C$. *Here $C \in \mathcal{P}(D)$ is the subset $C \subseteq D$ of admissible indices, indices are not necessarily numbers, and C does not have to be an interval nor even an ordered set. If D is finite, $\mathcal{P}(D)$ and all its elements are finite, so that all sets of admissible indices are finite. If D is infinite, so is $\mathcal{P}(D)$, and $\mathcal{P}(D)$ has both finite and infinite elements, so that there are both finite and infinite sets of admissible indices.*

Let a and b be variables of sort A, v and w be variables of sort V, i and j be variables of sort I, and n and m be variables of sort L. The axiomatization $\mathcal{A}_{\mathsf{ArrAD}}$ of ArrAD features the *congruence* axioms for the symbols in Σ_{ArrAD}:

$$\forall a, b, i, j.\ (a \simeq b \wedge i \simeq j) \rightarrow \mathsf{select}(a, i) \simeq \mathsf{select}(b, j), \quad (4)$$

$$\forall a, b, i, j, w, v.\ (a \simeq b \wedge i \simeq j \wedge w \simeq v) \rightarrow \mathsf{store}(a, i, w) \simeq \mathsf{store}(b, j, v), \quad (5)$$

$$\forall a, b.\ a \simeq b \rightarrow \mathsf{len}(a) \simeq \mathsf{len}(b), \quad (6)$$

$$\forall n, m, i, j.\ (n \simeq m \wedge i \simeq j \wedge \mathsf{Adm}(i, n)) \rightarrow \mathsf{Adm}(j, m). \quad (7)$$

Then $\mathcal{A}_{\mathsf{ArrAD}}$ contains axiom (2) and a new *select-over-store* axiom:

$$\forall a, v, i.\ \mathsf{Adm}(i, \mathsf{len}(a)) \rightarrow \mathsf{select}(\mathsf{store}(a, i, v), i) \simeq v. \quad (8)$$

Axiom (8) differs from axiom (1): if an index is admissible, storing a value at that index and then retrieving it yields the same value; otherwise, axiom (8) provides no guarantee. Then $\mathcal{A}_{\mathsf{ArrAD}}$ has an axiom to say that the length is unaffected by a store:

$$\forall a, i, v.\ \mathsf{len}(\mathsf{store}(a, i, v)) \simeq \mathsf{len}(a). \quad (9)$$

The last axiom of $\mathcal{A}_{\mathsf{ArrAD}}$ is the *extensionality* axiom:

$$\forall a, b.\ [\mathsf{len}(a) \simeq \mathsf{len}(b) \wedge (\forall i.\, \mathsf{Adm}(i, \mathsf{len}(a)) \rightarrow \mathsf{select}(a, i) \simeq \mathsf{select}(b, i))] \rightarrow a \simeq b, \quad (10)$$

whose other direction is omitted as it follows from the congruence axioms. In summary, $\mathcal{A}_{\mathsf{ArrAD}} = \{(4), (5), (6), (7)\} \cup \{(2), (8)\} \cup \{(9), (10)\}$.

In theory Arr, axiom (3) says that arrays a and b are equal if they have the same values at *all indices*. By the contrapositive, if a and b are different, they differ at an arbitrary index. In theory ArrAD, axiom (10) says that arrays a and b are equal if they have the same length and the same values at *all admissible indices*. By the contrapositive, if a and b are different, they differ either in length or at an admissible index. Neither theory entails the extensionality axiom of the other.

Example 5. *Picture a model of* Arr*, extended with an interpretation of* len *and* Adm*, where arrays* a *and* b *have the same length, agree at all admissible indices, but disagree at an inadmissible index:* $a \simeq b$ *is false in this model and hence axiom (10) also is false.*

Example 6. *Picture a model of* ArrAD *where arrays* a *and* b *agree at all indices but have different lengths:* $a \simeq b$ *is false in this model and hence axiom (3) also is false.*

The situation of Example 6 can happen even if arrays a and b have the same set of admissible indices.

Example 7. *Consider another theory* $\mathcal{T}$ *that interprets* I *as* $\mathbb{Z}$ *and* L *as the set of pairs of the form* $(addr, n)$*, where* $addr$ *is a binary number representing the starting address of the array in memory, and* n *is a non-negative integer representing the number of admissible indices. The* $\mathcal{T}$*-axiom defining admissibility is* $\forall i, addr, n.\ \mathsf{Adm}(i,(addr,n)) \leftrightarrow 0 \le i < n$*, where the starting address plays no role. With this axiom for admissibility, we can have two distinct arrays* a *and* b *with the same interval of admissible indices, say* $[0,5)$*, but* $\mathsf{len}(a) = (000100,5) \neq (010100,5) = \mathsf{len}(b)$ *because* a *and* b *start at distinct addresses and hence have different lengths.*

Example 7 shows how theory ArrAD models an interpretation of array equality that is common in programming languages: arrays starting at distinct addresses in memory are different. The next example shows how theory ArrAD captures the *bounded equality* of the *array property fragment* (APF) [14, 15].

Example 8. *APF allows one to define a bounded equality predicate* $beq(a,b,l,u)$*, for* a *and* b *integer-indexed arrays, and* l *and* u *integers, by the array property formula* $\forall i.\ l \le i \le u \ \rightarrow\ \mathsf{select}(a,i) \ \simeq\ \mathsf{select}(b,i)$*. Suppose that theory* $\mathcal{T}$ *is* LIA*, sort* I *is interpreted as* $\mathbb{Z}$*, sort* L *is interpreted as the Cartesian product* $\mathbb{Z} \times \mathbb{Z}$*, and* Adm *is defined with the axiom* $\forall i, l, u.\ \mathsf{Adm}(i,(l,u)) \leftrightarrow l \le i \le u$*. Then the extensionality axiom (10) of theory* ArrAD *covers the notion of bounded equality of APF.*

Theory ArrAD is designed having in mind the intuition that a store at an inadmissible index leaves the array unchanged. Therefore, the length is unchanged (axiom (9)), and the value argument of the store is lost, so that axiom (8) requires index i to be admissible. Alternatively, one can have a theory where a store at an inadmissible index *changes the set of admissible indices.* This choice leads to two other theories, namely those of *maps* and *vectors.*

3.2 Theories of Maps and Vectors with Abstract Domain

We begin by replacing axiom (9) (a store does not change the length) with an axiom saying that a store at an admissible index does not change the length:

$$\forall a, i, v.\ \mathsf{Adm}(i, \mathsf{len}(a)) \rightarrow \mathsf{len}(\mathsf{store}(a,i,v)) \simeq \mathsf{len}(a). \tag{11}$$

Then, we restore the select-over-store axioms (1)-(2). In the resulting theory (like in Arr), if $a \simeq \mathsf{store}(a,i,v)$, then by congruence $\mathsf{select}(a,i) \simeq \mathsf{select}(\mathsf{store}(a,i,v),i)$, and by axiom (8) it follows that $\mathsf{select}(a,i) \simeq v$. By the contrapositive, $\mathsf{select}(a,i) \not\simeq v$ implies $a \not\simeq \mathsf{store}(a,i,v)$. Suppose that $a \not\simeq \mathsf{store}(a,i,v)$ and index i is not admissible (i.e.,

$\neg\,\mathsf{Adm}(i, \mathsf{len}(a))$ holds). By the contrapositive of axiom (10) with variable b instantiated to $\mathsf{store}(a, i, v)$, it must be that $\mathsf{len}(a) \not\simeq \mathsf{len}(\mathsf{store}(a, i, v))$. In other words, a store at an inadmissible index changes the length of the structure. A way of specifying the change of length is to impose that index i be admissible in the structure represented by the term $\mathsf{store}(a, i, v)$. This is obtained by adding the axiom

$$\forall a, j, i, v.\ (\mathsf{Adm}(j, \mathsf{len}(a)) \vee j \simeq i) \to \mathsf{Adm}(j, \mathsf{len}(\mathsf{store}(a, i, v))). \tag{12}$$

Models of the resulting theory include data structures such as *maps* and *vectors*, meaning *dynamic arrays*, which satisfy *stronger* versions of axiom (12).

The theory MapAD of *maps with abstract domain* has the same signature as theory ArrAD (i.e., $\Sigma_{\mathsf{MapAD}} = \Sigma_{\mathsf{ArrAD}}$), with the sorts reinterpreted so that A is the sort of *maps* with *indices* of sort I, *values* of sort V, and *lengths* of sort L. The provision that $\mathsf{A} \neq \mathsf{I}$, $\mathsf{A} \neq \mathsf{V}$, $\mathsf{A} \neq \mathsf{L}$, $\mathsf{A} \neq \mathsf{prop}$, and $\mathsf{L} \neq \mathsf{prop}$ is maintained. The key axiom is the double implication version of axiom (12):

$$\forall a, j, i, v.\ (\mathsf{Adm}(j, \mathsf{len}(a)) \vee j \simeq i) \leftrightarrow \mathsf{Adm}(j, \mathsf{len}(\mathsf{store}(a, i, v))). \tag{13}$$

This axiom means that a store at an inadmissible index causes a map to grow exactly by one index: the index argument of the store becomes admissible and it is the only index that becomes admissible as an effect of the store. Therefore, a store operation causes a map to grow by at most one index. In summary, $\mathcal{A}_{\mathsf{MapAD}} = \{(4), (5), (6), (7)\} \cup \{(1), (2)\} \cup \{(10), (11), (13)\}$.

The theory VecAD of *vectors with abstract domain* has signature $\Sigma_{\mathsf{VecAD}} = (S_{\mathsf{VecAD}}, F_{\mathsf{VecAD}}, \mathit{ShF}_{\mathsf{VecAD}})$ with $S_{\mathsf{VecAD}} = S_{\mathsf{ArrAD}}$, $F_{\mathsf{VecAD}} = F_{\mathsf{ArrAD}} \cup \{<\}$, and $\mathit{ShF}_{\mathsf{VecAD}} = \mathit{ShF}_{\mathsf{ArrAD}} \cup \{<\}$. Sort A is interpreted as the sort of *vectors* with *indices* of sort I, *values* of sort V, and length of sort L, with $\mathsf{A} \neq \mathsf{I}$, $\mathsf{A} \neq \mathsf{V}$, $\mathsf{A} \neq \mathsf{L}$, $\mathsf{A} \neq \mathsf{prop}$, and $\mathsf{L} \neq \mathsf{prop}$ as above. The symbol $<\colon \mathsf{I} \times \mathsf{I} \to \mathsf{prop}$ is an ordering on indices. The axiomatization $\mathcal{A}_{\mathsf{VecAD}}$ adds a congruence axiom for the ordering on indices:

$$\forall i_1, j_1, i_2, j_2.\ (i_1 \simeq i_2 \wedge j_1 \simeq j_2 \wedge i_1 < j_1) \to i_2 < j_2. \tag{14}$$

The key axiom replaces $\simeq$ with $\leq$ (meaning $<$ or $\simeq$ as usual) in (13)

$$\forall a, j, i, v.\ (\mathsf{Adm}(j, \mathsf{len}(a)) \vee j \leq i) \leftrightarrow \mathsf{Adm}(j, \mathsf{len}(\mathsf{store}(a, i, v))). \tag{15}$$

Axiom (15) captures the growth of the vector as an effect of a store at an inadmissible index. Unlike a map, a vector can grow by a whole bunch of indices: all those smaller than or equal to the one made admissible by the store operation. Theory VecAD does not impose that the ordering $<$ is linear. In summary, $\mathcal{A}_{\mathsf{VecAD}} = \{(4), (5), (6), (7), (14)\} \cup \{(1), (2)\} \cup \{(10), (11), (15)\}$.

4 CDSAT for Predicate-Sharing Theories

In this section we modify the CDSAT framework [9, 11] to accommodate shared predicates other than equality. We use $\mathcal{T}$ for either any $\mathcal{T}_k$, $1 \leq k \leq n$, or $\mathcal{T}_\infty$.

4.1 Assignment, Theory View, and Endorsement

CDSAT works with *assignments* of values to terms, including assignments of Boolean values to formulæ. Boolean and first-order assignments, initial and generated assignments, are treated as uniformly as possible. The values for a theory $\mathcal{T}$ with signature $\Sigma = (S, F, \mathit{ShF})$ are provided by a *conservative theory extension* $\mathcal{T}^+$ with signature $\Sigma^+ = (S^+, F^+, \mathit{ShF}^+)$, such that $S^+ = S$, $F \subset F^+$ and $\mathit{ShF}^+ = \mathit{ShF}$. Set F^+ extends F with as many constant symbols as needed to name individuals in the domains used to interpret the sorts of $\mathcal{T}$. The added constants are called *$\mathcal{T}$-values*. Terms and values are kept separate, as $\mathcal{T}$-values cannot appear in Σ-terms.

Conservativity means that if a Σ-formula is $\mathcal{T}$-satisfiable then it is also $\mathcal{T}^+$-satisfiable. All extensions add values true and false, that are $\mathcal{T}$-values for all theories $\mathcal{T}$. An extension is *trivial* if it adds only true and false. An extension is *countably infinite* if it adds a countably infinite set of $\mathcal{T}$-values for each sort in $S \setminus \{\mathsf{prop}\}$. The components of the signature of $\mathcal{T}^+_\infty$ are given by the unions of the respective components of the signatures of $\mathcal{T}_1^+, \ldots, \mathcal{T}_n^+$, including $\mathit{ShF}_\infty{}^+ = \bigcup_{k=1}^n \mathit{ShF}_k{}^+ = \bigcup_{k=1}^n \mathit{ShF}_k$. $F^+_\infty = \bigcup_{k=1}^n F_k{}^+$ implies that all values are $\mathcal{T}_\infty$-values. We use $\mathfrak{b}$ for true or false and $\mathfrak{c}$ for generic values of arbitrary sort.

Definition 2 (Assignment (Def. 3 [9]))**.** *A set $J = \{u_1 \leftarrow \mathfrak{c}_1, \ldots, u_m \leftarrow \mathfrak{c}_m\}$ is a $\mathcal{T}$-*assignment *if $\forall i$, $1 \leq i \leq m$, u_i is a $\mathcal{T}_\infty$-term and $\mathfrak{c}_i$ is a $\mathcal{T}$-value of the same sort.*

The set of terms that *occur* in J is denoted $G(J)$, so that $G(J) = \{t \mid t \trianglelefteq u_i, 1 \leq i \leq m\}$, and $G_s(J)$ is the subset of the terms of sort s in $G(J)$. The set of free Σ-variables of J is $\mathsf{fv}_\Sigma(J) = \{u \mid u \in \mathsf{fv}_\Sigma(t), (t \leftarrow \mathfrak{c}) \in J\}$, written $\mathsf{fv}(J)$ when Σ is Σ_∞. If all values in J are Boolean, J is a *Boolean assignment.* If no value in J is Boolean, J is a *first-order assignment.* The *flip* of a Boolean singleton assignment L, written $\overline{L}$, assigns the opposite Boolean value to the same formula. Standard abbreviations are l for $l \leftarrow \mathsf{true}$, $\overline{l}$ for $l \leftarrow \mathsf{false}$, $t \not\simeq u$ for $(t \simeq u) \leftarrow \mathsf{false}$, $\top$ for $(x \simeq_{\mathsf{prop}} x) \leftarrow \mathsf{true}$, and $\bot$ for $x \not\simeq_{\mathsf{prop}} x$, where x is an arbitrary variable of sort prop. We use J for generic assignments, A for generic singletons, L for Boolean singletons, and H or E for $\mathcal{T}_\infty$-assignments. An unqualified assignment is a $\mathcal{T}_\infty$-assignment.

A $\mathcal{T}$-assignment is *plausible* if it does not contain both $l \leftarrow \mathsf{true}$ and $l \leftarrow \mathsf{false}$. Input assignments are assumed to be plausible and CDSAT preserves plausibility. A plausible $\mathcal{T}$-assignment may contain first-order assignments $u \leftarrow \mathfrak{c}_1$ and $u \leftarrow \mathfrak{c}_2$, with $\mathfrak{c}_1 \neq \mathfrak{c}_2$, for a term u of sort s other than prop. From such an assignment CDSAT deduces $u \not\simeq_s u$ and hence $\bot$. Plausibility bars having $l \leftarrow \mathsf{true}$ and $l \leftarrow \mathsf{false}$, because CDSAT is not allowed to deduce $l \not\simeq l$, and hence $\bot$, from $l \leftarrow \mathsf{true}$ and $l \leftarrow \mathsf{false}$. The reason is that CDSAT is allowed to generate equalities $u_1 \simeq_s u_2$ or $u_1 \not\simeq_s u_2$ from assignment to terms u_1 and u_2 of a sort s, $s \neq \mathsf{prop}$, but not arbitrary equalities with Boolean sides, for the sake of termination. Indeed, the possibility of generating arbitrary equalities with Boolean sides allows the construction of an infinite series such as $l_1 = (l \simeq_{\mathsf{prop}} l)$, $l_2 = (l_1 \simeq_{\mathsf{prop}} l_1)$, $l_3 = (l_2 \simeq_{\mathsf{prop}} l_2)$, etc. Such a series can be written because $\simeq_{\mathsf{prop}}$ is a predicate with Boolean arguments.

Different theories may have different *views* of a $\mathcal{T}_\infty$-assignment.

$$\begin{array}{rl}
t_1{\leftarrow}\mathfrak{c}, t_2{\leftarrow}\mathfrak{c} \vdash & t_1 \simeq_s t_2 \text{ if } \mathfrak{c} \text{ is a } \mathcal{T}\text{-value of sort } s \\
t_1{\leftarrow}\mathfrak{c}_1, t_2{\leftarrow}\mathfrak{c}_2 \vdash & t_1 \not\simeq_s t_2 \text{ if } \mathfrak{c}_1 \text{ and } \mathfrak{c}_2 \text{ are distinct } \mathcal{T}\text{-values of sort } s \\
\vdash & t_1 \simeq_s t_1 \text{ (reflexivity)} \\
t_1 \simeq_s t_2 \vdash & t_2 \simeq_s t_1 \text{ (symmetry)} \\
t_1 \simeq_s t_2, t_2 \simeq_s t_3 \vdash & t_1 \simeq_s t_3 \text{ (transitivity)}
\end{array}$$

Fig. 1 The set $\mathcal{I}_\simeq$ of the equality inference rules, where t_1, t_2, and t_3 are terms of sort s

Definition 3 (Theory view (Def. 4 [9])). *The* $\mathcal{T}$-view *of a* $\mathcal{T}_\infty$*-assignment* H *is the* $\mathcal{T}$*-assignment* $H_\mathcal{T}$ *given by the union of the following sets:*

- $\{\ u{\leftarrow}\mathfrak{c} \mid\ u{\leftarrow}\mathfrak{c}$ *is a* $\mathcal{T}$*-assignment in* $H\}$
- $\bigcup_{k=1}^{n}\{\ u_1 \simeq_s u_2 \mid u_1{\leftarrow}\mathfrak{c}, u_2{\leftarrow}\mathfrak{c}$ *are* $\mathcal{T}_k$*-assignments in* H *of sort* $s\}$
- $\bigcup_{k=1}^{n}\{\ u_1 \not\simeq_s u_2 \mid u_1{\leftarrow}\mathfrak{c}_1, u_2{\leftarrow}\mathfrak{c}_2$ *are* $\mathcal{T}_k$*-assignments in* H *of sort* $s,\ \mathfrak{c}_1{\neq}\mathfrak{c}_2\}$

for all $s \in S \setminus \{\mathsf{prop}\}$*, where* S *is the set of sorts of* $\mathcal{T}$*.*

In addition to the $\mathcal{T}$-assignments in H, the $\mathcal{T}$-view $H_\mathcal{T}$ contains all equalities and disequalities that can be gleaned from pairs of first-order $\mathcal{T}_k$-assignments in H, for any $\mathcal{T}_k$ (including $\mathcal{T}$ itself if $\mathcal{T}$ is not $\mathcal{T}_\infty$) sharing sort s with $\mathcal{T}$. Since true and false are $\mathcal{T}$-values for all $\mathcal{T}$, a Boolean assignment is a $\mathcal{T}$-assignment for all $\mathcal{T}$, and hence a Boolean assignment is in $H_\mathcal{T}$ for all $\mathcal{T}$.

Example 9. *Consider theories* ArrAD *and* LIA *as in Example 3 and assignment* $H = \{i{\leftarrow}3,\ i \simeq j,\ \mathsf{len}(a) \simeq n,\ n{\leftarrow}5,\ \mathsf{select}(\mathsf{store}(a,i,v),j) \not\simeq v\}$*. Then* $H_{\mathsf{LIA}} = H \cup \{i \not\simeq n\}$ *and* $H_{\mathsf{ArrAD}} = \{i \simeq j,\ \mathsf{len}(a) \simeq n,\ \mathsf{select}(\mathsf{store}(a,i,v),j) \not\simeq v,\ i \not\simeq n\}$*. Also* H_{ArrAD} *contains* $i \not\simeq n$*, because* ArrAD *shares with* LIA *the sorts of* i *and* n*, namely* I *and* L*, respectively, and* LIA *interprets them as the same sort (the integers).*

For J a $\mathcal{T}$-assignment, and $\mathcal{M}$ a $\mathcal{T}^+[\mathcal{V}]$-model such that $\mathsf{fv}_\Sigma(J) \subseteq \mathcal{V}$, we say that $\mathcal{M}$ *endorses* J, written $\mathcal{M} \models J$, if $\mathcal{M}$ satisfies $u \simeq \mathfrak{c}$ for all pairs $(u{\leftarrow}\mathfrak{c}) \in J$. It follows that if $\{u{\leftarrow}\mathfrak{c}, t{\leftarrow}\mathfrak{c}\} \subseteq J$, then $\mathcal{M}$ also satisfies $u \simeq t$. Endorsing the $\mathcal{T}$-view $J_\mathcal{T}$ of J is generally stronger than endorsing J: if $\mathcal{M} \models J_\mathcal{T}$, then $\mathcal{M}$ also satisfies $u \not\simeq t$, for all pairs $u{\leftarrow}\mathfrak{c}_1$ and $t{\leftarrow}\mathfrak{c}_2$ in J such that $\mathfrak{c}_1 \neq \mathfrak{c}_2$ and the sort of u and t is a sort of $\mathcal{T}$. A $\mathcal{T}$-assignment J is *satisfiable* if there exists a $\mathcal{T}^+[\mathcal{V}]$-model $\mathcal{M}$, with $\mathsf{fv}_\Sigma(J) \subseteq \mathcal{V}$, such that $\mathcal{M} \models J_\mathcal{T}$. Otherwise, J is *unsatisfiable*. For a $\mathcal{T}_\infty$-assignment H, if $\mathcal{M} \models H_{\mathcal{T}_\infty}$ we write $\mathcal{M} \models^G H$ and we say that $\mathcal{M}$ *globally endorses* H. The relation $J \models L$ holds if $\mathcal{M} \models L$ for all $\mathcal{T}^+$-models $\mathcal{M}$ such that $\mathcal{M} \models J_\mathcal{T}$.

4.2 Theory Modules and Local Bases

Every component theory $\mathcal{T}_k$ $(1 \leq k \leq n)$ is equipped with a *theory module* $\mathcal{I}_k$, which is an inference system, whose inferences have the form $J \vdash_{\mathcal{I}_k} L$, abbreviated $J \vdash_k L$, for J a $\mathcal{T}_k$-assignment and L a Boolean assignment. Every theory module includes the *equality inference rules* in Fig. 1: every module knows that equality is an equivalence and can deduce equalities and disequalities from first-order assignments. Only the module for the theory of equality (see Sect. 4.2 in [11]) knows that equality is a congruence. A theory module is *sound* if $J \vdash_k L$ implies $J \models L$.

Theory module inferences can generate *new* (i.e., non-input) terms. In order to avoid generating infinitely many, a theory module is restricted to pick new terms from a *finite basis*. A basis is defined as a function over sets of terms, so that the basis for an $\mathcal{I}_k$-derivation depends on the set of terms occurring in the input. Preliminarily, a set of terms is *closed*, if it is $\trianglelefteq$-*closed* (if t is a member so is every u such that $u \trianglelefteq t$) and *equality-closed* (if non-Boolean terms u and t are members so is $u \simeq t$). The *closure* $\Downarrow X$ of a set X of terms is the smallest closed set containing X. The closure operation is *idempotent* ($\Downarrow(\Downarrow X) = \Downarrow X$) and *monotone* (if $X \subseteq Y$ then $\Downarrow X \subseteq \Downarrow Y$).

Definition 4 (Basis (Def. 9 [9], Def. 2 [11]))**.** *A* basis *for theory $\mathcal{T}$ with signature Σ is a function* basis *from sets of terms to sets of terms, such that for all sets X and Y of terms:*

- $X \subseteq \mathsf{basis}(X)$ *(extensiveness),*
- *If X is finite then* $\mathsf{basis}(X)$ *is finite (finiteness),*
- $\mathsf{basis}(X) = \mathsf{basis}(\Downarrow X) = \Downarrow \mathsf{basis}(X)$ *(closure),*
- *If $X \subseteq Y$ then* $\mathsf{basis}(X) \subseteq \mathsf{basis}(Y)$ *(monotonicity),*
- $\mathsf{basis}(\mathsf{basis}(X)) = \mathsf{basis}(X)$ *(idempotence), and*
- $\mathsf{fv}_\Sigma(\mathsf{basis}(X)) \subseteq \mathsf{fv}_\Sigma(X) \cup \mathcal{V}_\infty$ *(no additional free Σ-variables),*

where the last requirement excludes the introduction of foreign terms.

A basis for a component theory $\mathcal{T}_k$ is called *local basis* and denoted basis_k. Given assignment J, module $\mathcal{I}_k$ can pick new terms from $\mathsf{basis}_k(G(J))$, abbreviated $\mathsf{basis}_k(J)$, where $G(J)$ is the set of terms occurring in J.

4.3 The CDSAT Framework for Predicate-Sharing Theories

CDSAT works with a *trail* Γ which is a sequence of distinct singleton assignments that are either *decisions*, written ${}_?A$ to convey guessing, or *justified assignments*, written ${}_{H\vdash}A$. Decisions can be either Boolean or first-order assignments. The *justification* H in ${}_{H\vdash}A$ is a set of singleton assignments that appear *before* A in the trail. Input assignments are seen as justified assignments with empty justification. All justified assignments are Boolean except for input first-order assignments (whose justification is however empty). Given a trail $\Gamma = A_0, \ldots, A_m$, the *level* of assignment A_i is defined as follows: $\mathsf{level}_\Gamma(A_i) = 1 + \max\{\mathsf{level}_\Gamma(A_j) \mid j < i\}$ if A_i is a decision, and $\mathsf{level}_\Gamma(A_i) = \max\{\mathsf{level}_\Gamma(A) \mid A \in H\}$ if A_i is a justified assignment ${}_{H\vdash}A_i$ (where $\mathsf{level}_\Gamma(A_i) = 0$ if $H = \emptyset$). A decision ${}_?A$ can be made if it is *acceptable* for a $\mathcal{T}$-module $\mathcal{I}$ in the $\mathcal{T}$-view $\Gamma_\mathcal{T}$ of the current trail Γ. Acceptability involves the *relevance* of the term whose value is being decided. The definition of relevance (cf. Def. 6 [9])) needs to be extended to accommodate shared predicates other than equality.

Definition 5 (Predicate-sharing relevance)**.** *Given a theory $\mathcal{T}$ with signature $\Sigma = (S, F, ShF)$ and a $\mathcal{T}$-assignment J, where $G(J)$ is the set of terms that occur in J, a term u is* relevant *to $\mathcal{T}$ in J, if either (i) $u \in G(J)$ and $\mathcal{T}$ has values for the sort of u; or (ii) u is an equality $u_1 \simeq_s u_2$ such that $u_1, u_2 \in G(J)$, $s \in S$, but $\mathcal{T}$ does not have values for sort s; or (iii) u is a Boolean term $p(u_1, \ldots, u_m)$ such that $p \in ShF$ is a* *shared predicate symbol* $p\colon (s_1 \times \cdots \times s_m) \rightarrow \mathsf{prop}$*, and for all* i*,* $1 \leq i \leq m$*,* $u_i \in G(J)$ *and* $s_i \in S$*.*

For Condition (i), it makes sense that a $\mathcal{T}$-module $\mathcal{I}$ may decide a value for a term u if u occurs in the $\mathcal{T}$-view $\Gamma_{\mathcal{T}}$ of the trail and $\mathcal{T}$ has values for the sort of u. For Condition (ii), it also makes sense that $\mathcal{I}$ may decide $u \simeq t$ if u and t occur in $\Gamma_{\mathcal{T}}$, even if $u \simeq t$ does not, provided that $\mathcal{T}$ does not have values for the sort of u and t. If $\mathcal{T}$ has values for the sort of u and t, $\mathcal{I}$ can decide values for u and t, and glean the value of $u \simeq t$ by an equality inference. Condition (iii) extends the treatment of equality to other shared predicates, to which however the previous point does not apply.

Example 10. *Continuing with the theories and the assignment* H *of Example 9, term* $\mathsf{Adm}(i,n)$ *occurs neither in* H_{ArrAD} *nor in* H_{LIA}*, but its arguments do. Thus,* $\mathsf{Adm}(i,n)$ *is relevant to both* LIA *and* ArrAD *by Condition (iii) in Definition 5. Knowing the definition of* Adm*,* LIA *can decide wisely* $\mathsf{Adm}(i,n) \leftarrow \mathsf{true}$*. If* ArrAD *were to venture* $\mathsf{Adm}(i,n) \leftarrow \mathsf{false}$*, then* LIA *would detect a conflict.*

Definition 6 (Acceptability (Def. 7 [9]))**.** *A singleton* $\mathcal{T}$*-assignment* $u \leftarrow \mathfrak{c}$ *is* acceptable *for a* $\mathcal{T}$*-module* $\mathcal{I}$ *in a* $\mathcal{T}$*-assignment* J*, if (i) term* u *is relevant to* $\mathcal{T}$ *in* J*, (ii)* J *does not assign a* $\mathcal{T}$*-value to term* u*, and (iii) if* $u \leftarrow \mathfrak{c}$ *is a first-order assignment, for no* $J' \subseteq J$ *and* $L \in J$ *there exists an* $\mathcal{I}$*-inference* $J' \cup \{u \leftarrow \mathfrak{c}\} \vdash_{\mathcal{I}} \overline{L}$*.*

For Boolean terms and Boolean values, Condition (ii) preserves plausibility, because Boolean values are $\mathcal{T}$-values for all $\mathcal{T}$. Condition (iii) blocks a first-order assignment that triggers an inference contradicting the contents of the trail.

The transition system of CDSAT (see Fig. 2) comprises *trail rules* and *conflict state rules*, and it is parametrized with respect to a *global basis* $\mathcal{B}$. A *conflict* is an unsatisfiable assignment, and a *conflict state* is made of a trail and a conflict. An assignment H *is in* $\mathcal{B}$, if $(t \leftarrow \mathfrak{c}) \in H$ implies $t \in \mathcal{B}$. For the trail rules, Decide expands the trail with an acceptable decision ${}_{?}A$. Deduce expands the trail with a justified assignment ${}_{J\vdash}L$, supported by a theory inference $J \vdash_k L$ for some k, $1 \leq k \leq n$, provided L is in $\mathcal{B}$ and $\overline{L}$ is not on the trail. If $\overline{L}$ is on the trail, $J \cup \{\overline{L}\}$ is a conflict. If the conflict's level is 0, rule Fail reports unsatisfiability. Otherwise, rule ConflictSolve returns the trail produced by the conflict state rules and the search resumes.

In the conflict state rules, $\Gamma^{\leq m}$ is the *restriction* of trail Γ to its elements of level at most m. Rule UndoClear solves the conflict by removing a first-order decision whose level is maximum in the conflict. Rule Resolve unfolds the conflict by replacing a justified assignment A in the conflict with its justification H. Typically Resolve applies until the conflict can be solved by either UndoClear, UndoDecide, or LearnBackjump. However Resolve is blocked if replacing A by H would put in the conflict a first-order decision A' whose level is maximum in the conflict. This provision prevents a loop where UndoClear removes A', Decide puts it back, and Deduce derives A again, yielding the same conflict. Rule UndoDecide solves the conflict by flipping a Boolean justified assignment L that cannot be unfolded by Resolve for the above reason: since a first-order assignment A' does not have a flip, UndoDecide flips a Boolean consequence L of A'. Rule LearnBackjump[2] solves the conflict by learning the *clausal form* of a

[2]Rule LearnBackjump [11] subsumes the Backjump rule of the transition system in [9].

TRAIL RULES (assume $1 \leq k \leq n$)		
Decide	$\Gamma \longrightarrow \Gamma, {}_{?}A$	if A is an acceptable $\mathcal{T}_k$-assignment for $\mathcal{I}_k$ in $\Gamma_{\mathcal{T}_k}$
The next three rules share the conditions: $J \subseteq \Gamma$, $(J \vdash_k L)$, $L \notin \Gamma$, and L is in $\mathcal{B}$.		
Deduce	$\Gamma \longrightarrow \Gamma, {}_{J\vdash}L$	if $\overline{L} \notin \Gamma$
Fail	$\Gamma \longrightarrow \mathsf{unsat}$	if $\overline{L} \in \Gamma$ and $\mathsf{level}_\Gamma(J \cup \{\overline{L}\}) = 0$
ConflictSolve	$\Gamma \longrightarrow \Gamma'$	if $\overline{L} \in \Gamma$, $\mathsf{level}_\Gamma(J \cup \{\overline{L}\}) > 0$, and $\langle \Gamma; J \cup \{\overline{L}\}\rangle \Longrightarrow^* \Gamma'$
CONFLICT STATE RULES (recall that $\uplus$ is disjoint union)		
UndoClear		
	$\langle \Gamma; E \uplus \{A\}\rangle \Longrightarrow \Gamma^{\leq m-1}$	if A is a first-order decision of level $m > \mathsf{level}_\Gamma(E)$
Resolve		
	$\langle \Gamma; E \uplus \{A\}\rangle \Longrightarrow \langle \Gamma; E \cup H\rangle$	if $({}_{H\vdash}A) \in \Gamma$ and for no first-order decision $A' \in H$, $\mathsf{level}_\Gamma(A') = \mathsf{level}_\Gamma(E \uplus \{A\})$
UndoDecide		
	$\langle \Gamma; E \uplus \{L\}\rangle \Longrightarrow \Gamma^{\leq m-1}, {}_{?}\overline{L}$	if $({}_{H\vdash}L) \in \Gamma$ and for a first-order decision $A' \in H$, $m = \mathsf{level}_\Gamma(A')$, and $m = \mathsf{level}_\Gamma(E) = \mathsf{level}_\Gamma(L)$
LearnBackjump		
	$\langle \Gamma; E \uplus H\rangle \Longrightarrow \Gamma^{\leq m}, {}_{E\vdash}L$	if L is a clausal form of H, L is in $\mathcal{B}$, $L \notin \Gamma$, $\overline{L} \notin \Gamma$, and $\mathsf{level}_\Gamma(E) \leq m < \mathsf{level}_\Gamma(H)$

Fig. 2 The CDSAT transition system

Boolean subset H of the conflict and backjumping to a level m. The *clausal form* of $H = \{l_1, \ldots, l_r\}$ is the clause $L = (\neg l_1 \vee \ldots \vee \neg l_r)$. Indeed, if $E \uplus H$ is a conflict, then ${}_{E\vdash}L$ can be asserted. The destination level m is strictly smaller than the level of H, but large enough to keep on the trail the assignments in E, since E is the justification of ${}_{E\vdash}L$. Comments and examples about these rules are available [6, 9, 11].

The requirements for the global basis $\mathcal{B}$ are that the input assignment is in $\mathcal{B}$, and $\mathcal{B}$ is *finite*, *closed* (see Sect. 4.2), and *stable*: for all k, $1 \leq k \leq n$, $\mathsf{basis}_k(\mathcal{B}) \subseteq \mathcal{B}$. Stability ensures that for all sets X of terms, if $X \subseteq \mathcal{B}$ then for all k, $1 \leq k \leq n$, $\mathsf{basis}_k(X) \subseteq \mathcal{B}$. Indeed, from $X \subseteq \mathcal{B}$ we have $\mathsf{basis}_k(X) \subseteq \mathsf{basis}_k(\mathcal{B})$ by monotonicity of basis_k (cf. Def. 4), and then $\mathsf{basis}_k(X) \subseteq \mathcal{B}$ by stability. Thus, for an assignment H, if $G(H) \subseteq \mathcal{B}$, then $\mathsf{basis}_k(G(H)) \subseteq \mathcal{B}$. Stability also implies closure (cf. Def. 4): for all k, $1 \leq k \leq n$, by extensiveness of basis_k, we have $\mathcal{B} \subseteq \mathsf{basis}_k(\mathcal{B})$, which, together with stability, implies $\mathsf{basis}_k(\mathcal{B}) = \mathcal{B}$, so that $\mathcal{B}$ is closed by the closure property of basis_k. In turn, if $\mathcal{B}$ is closed, if H is in $\mathcal{B}$ then $G(H) \subseteq \mathcal{B}$, and since $G(H) \subseteq \mathcal{B}$ trivially implies that H is in $\mathcal{B}$, the closure of $\mathcal{B}$ means that H is in $\mathcal{B}$ iff $G(H) \subseteq \mathcal{B}$.

Definition 7 (Assignment expansion (cf. Def. 10 [9], Def. 3 [11])). *A $\mathcal{T}$-module $\mathcal{I}$ with local basis* basis *can expand a $\mathcal{T}$-assignment J if there exists either (1) a $\mathcal{T}$-assignment A that is acceptable for $\mathcal{I}$ in J, or (2) a Boolean assignment $l \leftarrow \mathfrak{b}$ derived by an $\mathcal{I}$-inference $J' \vdash_{\mathcal{I}} (l \leftarrow \mathfrak{b})$ such that $(l \leftarrow \mathfrak{b}) \notin J$, $l \in \mathsf{basis}(J)$, and $J' \subseteq J$.*

In a Decide transition, a $\mathcal{T}$-module expands the $\mathcal{T}$-view of the trail by Case (1) of Definition 7. In Deduce, Fail, and $\mathsf{ConflictSolve}$ transitions, a $\mathcal{T}$-module expands the $\mathcal{T}$-view of the trail by Case (2).

Definition 8 (One-theory-completeness (cf. Def. 12 [9], Def. 4 [11])). *Given theory $\mathcal{T}$ with signature Σ, a $\mathcal{T}$-module $\mathcal{I}$ is* complete *for $\mathcal{T}$ if whenever $\mathcal{I}$ cannot expand a plausible $\mathcal{T}$-assignment J, there exists a $\mathcal{T}^+[\mathcal{V}]$-model $\mathcal{M}$ such that $\mathcal{M} \models J$ assuming $\mathsf{fv}_\Sigma(J) \subseteq \mathcal{V}$.*

Assume that there exists a *leading theory*, say $\mathcal{T}_1$, whose signature has all the sorts in the union (i.e., $S_1 = S_\infty$) and all the shared symbols: if there exists a predicate symbol p such that $p \in \mathit{ShF}_j \cap \mathit{ShF}_k$ for some j and k, $2 \leq j \neq k \leq n$, then $p \in \mathit{ShF}_1$.

Definition 9 (Predicate-sharing leading-theory-compatibility). *Let $\mathcal{T}_1$ be the leading theory, $\mathcal{T}$ with signature $\Sigma = (S, F, \mathit{ShF})$ stand for $\mathcal{T}_k$ with signature $\Sigma_k = (S_k, F_k, \mathit{ShF}_k)$, $2 \leq k \leq n$, and N be a set of terms. A $\mathcal{T}$-assignment J is* leading-theory-compatible with $\mathcal{T}$ sharing N, *if for all $\mathcal{T}_1^+[\mathcal{V}_1]$-models $\mathcal{M}_1$ such that $\mathcal{M}_1 \models J_{\mathcal{T}_1}$ with $\mathsf{fv}_{\Sigma_1}(J \cup N) \subseteq \mathcal{V}_1$, there exists a $\mathcal{T}^+[\mathcal{V}]$-model $\mathcal{M}$ with $\mathsf{fv}_\Sigma(J \cup N) \subseteq \mathcal{V}$, such that*

(i) $\mathcal{M} \models J$;

(ii) *For all shared predicate symbols $p \in \mathit{ShF} \cap \mathit{ShF}_1$, $p\colon (s_1 \times \cdots \times s_m) \rightarrow \mathsf{prop}$, and for all terms $u_1, \ldots, u_m \in N$ of sorts $s_1, \ldots, s_m$, $\mathcal{M}(p(u_1, \ldots, u_m)) = \mathcal{M}_1(p(u_1, \ldots, u_m))$; and*

(iii) *For all sorts $s \in S$, there exists a bijection $f^s\colon s^{\mathcal{M}} \rightarrow s^{\mathcal{M}_1}$ (so that $|s^{\mathcal{M}}| = |s^{\mathcal{M}_1}|$), such that f^{prop} is identity, and for all $p \in \mathit{ShF} \cap \mathit{ShF}_1$, $p\colon (s_1 \times \cdots \times s_m) \rightarrow \mathsf{prop}$, and for all inhabitants $v_1, \ldots, v_m$ of $s_1^{\mathcal{M}}, \ldots, s_m^{\mathcal{M}}$, $p^{\mathcal{M}}(v_1, \ldots, v_m) = p^{\mathcal{M}_1}(f^{s_1}(v_1), \ldots, f^{s_m}(v_m))$.*

If equality is the only shared predicate, Properties (*ii*) and (*iii*) reduce to their counterparts in the definition for the disjoint case (cf. [9, Def. 13] and [11, Def. 5]). Property (*ii*) reduces to $\mathcal{M}(u_1) = \mathcal{M}(u_2)$ iff $\mathcal{M}_1(u_1) = \mathcal{M}_1(u_2)$ for all sorts $s \in S$ and terms $u_1, u_2 \in N$ of sort s (agreement on equalities between shared terms). Property (*iii*) reduces to $|s^{\mathcal{M}}| = |s^{\mathcal{M}_1}|$ for all $s \in S$ (agreement on cardinalities of shared sorts). The rest of Property (*iii*) is trivial for equality, because all models interpret equality as identity: for all inhabitants v_1, v_2 of $s^{\mathcal{M}}$, $v_1 = v_2$ iff $f^s(v_1) = f^s(v_2)$ is trivial in one direction, because f^s is a function, and it is trivial in the other direction, because f^s is injective.

If equality is not the only shared predicate, Properties (*ii*) and (*iii*) extend the treatment of equality to all shared predicates. For example, for predicate Adm that each of ArrAD, MapAD, and VecAD share with some other $\mathcal{T}$ and hence with $\mathcal{T}_1$, Property (*ii*) says that for all shared terms u_1 of sort I and u_2 of sort L, $\mathcal{M}(\mathsf{Adm}(u_1, u_2)) = \mathcal{M}_1(\mathsf{Adm}(u_1, u_2))$, that is, $\mathcal{M}(u_1)$ is admissible for $\mathcal{M}(u_2)$ iff $\mathcal{M}_1(u_1)$ is admissible for $\mathcal{M}_1(u_2)$ (agreement on shared predicates applied to shared terms). Property (*iii*) says that for all inhabitants v_1 of $\mathsf{I}^{\mathcal{M}}$ and v_2 of $\mathsf{L}^{\mathcal{M}}$, $\mathsf{Adm}^{\mathcal{M}}(v_1, v_2) = \mathsf{Adm}^{\mathcal{M}_1}(f^{\mathsf{I}}(v_1), f^{\mathsf{L}}(v_2))$, that is, an index v_1 is admissible for a length v_2 in $\mathcal{M}$ iff index $f^{\mathsf{I}}(v_1)$ is admissible for length $f^{\mathsf{L}}(v_2)$ in $\mathcal{M}_1$. In other words, the collection of bijections preserves the interpretation of shared predicates.

Definition 10 (Leading-theory-completeness (cf. Def. 14 [9], Def. 6 [11])). *For a nonleading theory $\mathcal{T}$, a $\mathcal{T}$-module $\mathcal{I}$ is* leading-theory-complete*, if whenever $\mathcal{I}$ cannot expand a plausible $\mathcal{T}$-assignment J, then J is leading-theory-compatible with $\mathcal{T}$ sharing $G(J)$.*

Definition 9 has J and not the $\mathcal{T}$-view $J_{\mathcal{T}}$ in Property (i), because Definition 10 applies leading-theory-compatibility to an assignment J that $\mathcal{I}$ cannot expand, which implies $J = J_{\mathcal{T}}$. Indeed, if $J \neq J_{\mathcal{T}}$, by Definition 3, assignment J lacks an equality or disequality that can be gleaned from first-order assignments in J, and then $\mathcal{I}$ can expand J by an equality inference.

5 Three New Theory Modules for CDSAT

In previous work we gave a module $\mathcal{I}_{\mathsf{Arr}}$ for the theory Arr of arrays with extensionality [9, Sect. 4.3] and proved its leading-theory-completeness [11, Thm. 4]. In this section we present modules for the theories of arrays, maps, and vectors with abstract domain, and we show that they are leading-theory-complete.

5.1 Inference Rules for Arrays, Maps, and Vectors

The extensionality axiom (10) can be reduced to the clauses

$$\mathsf{len}(a) \not\simeq \mathsf{len}(b) \vee \mathsf{select}(a, \mathsf{diff}(a,b)) \not\simeq \mathsf{select}(b, \mathsf{diff}(a,b)) \vee a \simeq b$$
$$\mathsf{len}(a) \not\simeq \mathsf{len}(b) \vee \mathsf{Adm}(\mathsf{diff}(a,b), \mathsf{len}(a)) \vee a \simeq b,$$

where $\mathsf{diff}\colon \mathsf{A} \times \mathsf{A} \to \mathsf{I}$ is the Skolem function symbol that maps two arrays to an index, called a *witness*, where they differ. Besides the equality rules of Fig. 1, module $\mathcal{I}_{\mathsf{ArrAD}}$ has inference rules corresponding to the axioms in $\mathcal{A}_{\mathsf{ArrAD}}$ (see Sect. 3.1). The rules for congruence axioms (4)-(7) are:

$$a \simeq b,\ i \simeq j,\ \mathsf{select}(a,i) \not\simeq \mathsf{select}(b,j)\ \vdash\ \bot \tag{16}$$
$$a \simeq b,\ i \simeq j,\ w \simeq v,\ \mathsf{store}(a,i,w) \not\simeq \mathsf{store}(b,j,v)\ \vdash\ \bot \tag{17}$$
$$a \simeq b\ \vdash\ \mathsf{len}(a) \simeq \mathsf{len}(b) \tag{18}$$
$$n \simeq m,\ i \simeq j,\ \mathsf{Adm}(i,n),\ \neg\mathsf{Adm}(j,m)\ \vdash\ \bot \tag{19}$$
$$a \simeq c,\ b \simeq d,\ \mathsf{diff}(a,b) \not\simeq \mathsf{diff}(c,d)\ \vdash\ \bot, \tag{20}$$

plus rule (20) for the congruence axiom for diff (c and d are additional variables of sort A), which gets added to $\mathcal{A}_{\mathsf{ArrAD}}$ once diff is added to Σ_{ArrAD}. The rules for select-over-store axioms (2)-(8) are:

$$i \not\simeq j,\ k \simeq j,\ b \simeq \mathsf{store}(a,i,v),\ a \simeq c,\ \mathsf{select}(b,k) \not\simeq \mathsf{select}(c,j)\ \vdash\ \bot \tag{21}$$

$$i \simeq j,\ \mathsf{len}(a) \simeq n,\ \mathsf{Adm}(i,n),\ b \simeq \mathsf{store}(a,i,v),\ \mathsf{select}(b,j) \not\simeq v\ \vdash\ \bot, \tag{22}$$

where the premises have been flattened and linearized (in the sense of avoiding repeated occurrences of variables), by introducing new variables, such as k of sort I. Consider rule (22). If the trail contains $\mathsf{Adm}(i, \mathsf{len}(a))$ and $\mathsf{select}(\mathsf{store}(a,i,v),j) \not\simeq v$, rule (22) can fire, because flattening and linearizing do not require inferences. On the other hand, a rule with $\mathsf{Adm}(i, \mathsf{len}(a))$ and $\mathsf{select}(\mathsf{store}(a,i,v),j) \not\simeq v$ as premises could not fire when the trail contains terms as in the premise of rule (22), because the equality rules of Fig. 1 do not include a rule for replacement of equals by equals,[3] and hence could not deduce $\mathsf{Adm}(i, \mathsf{len}(a))$ and $\mathsf{select}(\mathsf{store}(a,i,v),j) \not\simeq v$ from the terms on the trail. Then there are a rule for axiom (9) (store does not change length)

$$b \simeq \mathsf{store}(a,i,v),\ \mathsf{len}(b) \not\simeq \mathsf{len}(a)\ \vdash\ \bot \tag{23}$$

and two rules for the clauses from extensionality axiom (10)

$$a \not\simeq b,\ \mathsf{len}(a) \simeq \mathsf{len}(b)\ \vdash\ \mathsf{select}(a, \mathsf{diff}(a,b)) \not\simeq \mathsf{select}(b, \mathsf{diff}(a,b)) \tag{24}$$

$$a \not\simeq b,\ \mathsf{len}(a) \simeq \mathsf{len}(b)\ \vdash\ \mathsf{Adm}(\mathsf{diff}(a,b), \mathsf{len}(a)). \tag{25}$$

In summary, $\mathcal{I}_{\mathsf{ArrAD}} = \mathcal{I}_{\simeq} \cup \{(16),(17),(18),(19),(20)\} \cup \{(21),(22)\} \cup \{(23),(24),(25)\}$.

Module $\mathcal{I}_{\mathsf{MapAD}}$ for the theory of maps with abstract domain includes the equality rules of Fig. 1 and the same congruence rules of $\mathcal{I}_{\mathsf{ArrAD}}$ (cf. rules (16)-(20)). As axiom (1) differs from axiom (8), $\mathcal{I}_{\mathsf{MapAD}}$ differs from $\mathcal{I}_{\mathsf{ArrAD}}$ in the second select-over-store rule:

$$i \simeq j,\ b \simeq \mathsf{store}(a,i,v),\ \mathsf{select}(b,j) \not\simeq v\ \vdash\ \bot, \tag{26}$$

whereas the first select-over-store rule remains rule (21). Since axiom (11) about store and len differs from axiom (9), $\mathcal{I}_{\mathsf{MapAD}}$ differs from $\mathcal{I}_{\mathsf{ArrAD}}$ also in the rule for axiom (11):

$$\mathsf{len}(a) \simeq n,\ \mathsf{Adm}(i,n),\ b \simeq \mathsf{store}(a,i,v),\ \mathsf{len}(b) \not\simeq \mathsf{len}(a)\ \vdash\ \bot, \tag{27}$$

whereas the rules for extensionality remain rules (24)-(25). By transforming axiom (13) into clauses, we get two clauses for the $\rightarrow$ direction:

$$\neg\,\mathsf{Adm}(j, \mathsf{len}(a)) \vee \mathsf{Adm}(j, \mathsf{len}(\mathsf{store}(a,i,v)))$$

$$j \not\simeq i \vee \mathsf{Adm}(j, \mathsf{len}(\mathsf{store}(a,i,v))).$$

These clauses yield the rules

$$\begin{pmatrix} m \simeq \mathsf{len}(a),\ b \simeq \mathsf{store}(a,i,v),\ n \simeq \mathsf{len}(b) \\ k \simeq j,\ \mathsf{Adm}(k,m),\ \neg\,\mathsf{Adm}(j,n) \end{pmatrix} \vdash\ \bot \tag{28}$$

$$b \simeq \mathsf{store}(a,i,v),\ n \simeq \mathsf{len}(b),\ j \simeq i,\ \neg\,\mathsf{Adm}(j,n) \vdash\ \bot. \tag{29}$$

[3]Replacement of equals by equals is reasoning about congruence, which belongs to the module for the theory of equality, as already mentioned in Sect. 4.2.

Rule (28) ensures that every admissible index k of map a remains admissible in map $\mathsf{store}(a,i,v)$. Rule (29) ensures that the index i where the store occurs is admissible in map $\mathsf{store}(a,i,v)$. The one clause for the $\leftarrow$ direction of axiom (13)

$$\neg\,\mathsf{Adm}(j,\mathsf{len}(\mathsf{store}(a,i,v))) \vee \mathsf{Adm}(j,\mathsf{len}(a)) \vee j \simeq i$$

yields one more rule:

$$\begin{pmatrix} m \simeq \mathsf{len}(a),\ b \simeq \mathsf{store}(a,i,v),\ n \simeq \mathsf{len}(b) \\ k \simeq j,\ \neg\,\mathsf{Adm}(k,m),\ \mathsf{Adm}(j,n),\ j \not\simeq i \end{pmatrix} \vdash \bot \tag{30}$$

which ensures that every admissible index j of map $\mathsf{store}(a,i,v)$ such that $j \not\simeq i$ is admissible in map a. In summary, $\mathcal{I}_{\mathsf{MapAD}} = \mathcal{I}_{\simeq} \cup \{(16),(17),(18),(19),(20)\} \cup \{(21),(26)\} \cup \{(24),(25),(27),(28),(29),(30)\}$.

Besides the equality rules of Fig. 1, module $\mathcal{I}_{\mathsf{VecAD}}$ adds to congruence rules (16)-(20) a rule for congruence axiom (14):

$$i_1 \simeq i_2,\ j_1 \simeq j_2,\ i_1 < j_1,\ i_2 \not< j_2\ \vdash\ \bot. \tag{31}$$

The rules for the select-over-store axioms, for axiom (11), and for extensionality are the same as in $\mathcal{I}_{\mathsf{MapAD}}$. Axiom (15) differs from axiom (13) in that it has atom $j \leq i$ where axiom (13) has atom $j \simeq i$. Thus, the rules for axiom (15) about vectors follow those for axiom (13) about maps (i.e., rules (28)-(30)), with the difference that while rule (28) remains unchanged, $j \leq i$ replaces $j \simeq i$ in the other two rules:

$$b \simeq \mathsf{store}(a,i,v),\ n \simeq \mathsf{len}(b),\ j \leq i,\ \neg\,\mathsf{Adm}(j,n) \vdash \bot \tag{32}$$

$$\begin{pmatrix} m \simeq \mathsf{len}(a),\ b \simeq \mathsf{store}(a,i,v),\ n \simeq \mathsf{len}(b) \\ k \simeq j,\ \neg\,\mathsf{Adm}(k,m),\ \mathsf{Adm}(j,n),\ j \not\leq i \end{pmatrix} \vdash \bot. \tag{33}$$

In summary, $\mathcal{I}_{\mathsf{VecAD}} = \mathcal{I}_{\simeq} \cup \{(16),(17),(18),(19),(20),(31)\} \cup \{(21),(26)\} \cup \{(24),(25),(27),(28),(32),(33)\}$. The inference rules of $\mathcal{I}_{\mathsf{ArrAD}}$, $\mathcal{I}_{\mathsf{MapAD}}$, and $\mathcal{I}_{\mathsf{VecAD}}$ are *sound*.

5.2 Local Basis for Arrays, Maps, and Vectors

For $\mathcal{T} \in \{\mathsf{ArrAD},\mathsf{MapAD},\mathsf{VecAD}\}$, most of the inference rules in $\mathcal{I}_{\mathcal{T}}$ are *lazy*: they fire when the trail violates an axiom, generating $\bot$ to signal a conflict. Lazy rules are trivial for the definition of the local basis, since it suffices that it contains $\top$ whose flip is $\bot$. However, for completeness, the inference rules of a module must be powerful enough to ensure that when they cannot expand the assignment on the trail, there exists a model. In turn, this means that the inference rules must be able to put on the trail useful terms for specifying a model. This is why $\mathcal{I}_{\mathcal{T}}$ contains also rules that are not lazy, namely rules (18), (24) and (25). The design of a CDSAT module demands to balance completeness, which may require the generation of new terms, with finiteness of the local basis, which suggests to minimize the generation of new terms.

Definition 11. *For $\mathcal{T} \in \{\mathsf{ArrAD}, \mathsf{MapAD}, \mathsf{VecAD}\}$, given a set X of terms,* $\mathsf{basis}_{\mathcal{T}}(X)$ *is the smallest closed set Y such that $X \subseteq Y$, $\top \in Y$, and:*

1. $(l_1 \simeq_{\mathsf{prop}} l_2) \in Y$ *for all terms* $l_1, l_2 \in \mathcal{L}(Y)$ *for* $\mathcal{L}(Y) = \{l \mid l\colon \mathsf{prop},\ l \trianglelefteq u,\ u \in Y,\ \mathit{top}(u) = \mathsf{select} \vee \mathit{top}(u) = \mathsf{diff}\} \cup \{l \mid l\colon \mathsf{prop},\ l \triangleleft u,\ u \in Y,\ \mathit{top}(u) = \mathsf{store} \vee \mathit{top}(u) = \mathsf{len}\}$*;*
2. $\mathsf{len}(t) \in Y$*, for all terms* $t \in Y$ *of sort* A*; and*
3. $\mathsf{select}(t, \mathsf{diff}(t,u)) \in Y$*,* $\mathsf{select}(u, \mathsf{diff}(t,u)) \in Y$*,* $\mathsf{Adm}(\mathsf{diff}(t,u), \mathsf{len}(t)) \in Y$*, and* $\mathsf{Adm}(\mathsf{diff}(t,u), \mathsf{len}(u)) \in Y$*, for all pairs of terms* $t, u \in Y$ *of sort* A*.*

Clause (1) adds equalities between Boolean terms that may be needed and whose presence is not guaranteed by equality-closure that applies only to non-Boolean terms. As indices or values may be Boolean, equalities between Boolean terms that represent values or indices may be needed. Thus, Clause (1) considers non-strict subterms of terms whose top symbol is select or diff, because select returns a value and diff returns an index. Since $\mathsf{A} \neq \mathsf{prop}$ and $\mathsf{L} \neq \mathsf{prop}$ (see Section 3), Clause (1) considers only strict subterms of terms whose top symbol is store or len. It does not consider subterms of terms whose top symbol is Adm. Indeed, first there is no need of equalities involving Adm-terms. Second, a shared symbol should not contribute to make a local basis larger precisely because a local basis is local to a theory and its module. Clause (2) adds the terms that may be generated by rule (18). Clause (3) adds the terms that may be generated by rules (24)-(25). We prove finiteness next, while it is plain to see that the other requirements in Definition 4 are met.

Lemma 1. *For $\mathcal{T} \in \{\mathsf{ArrAD}, \mathsf{MapAD}, \mathsf{VecAD}\}$, for all finite sets X of terms,* $\mathsf{basis}_{\mathcal{T}}(X)$ *is a finite set of terms.*

Proof. For Clause (1), let $\mathrm{Sat}^1(X) = X \cup \{l_1 \simeq_{\mathsf{prop}} l_2 \mid l_1, l_2 \in \mathcal{L}(X)\}$. For Clause (2), let $\mathrm{Sat}^2(X) = X \cup \{\mathsf{len}(t) \mid t \in X,\ t\colon \mathsf{A}\}$. For Clause (3), let $\mathrm{Sat}^3(X)$ be the union of X and the set containing all and only the terms of the form $\mathsf{select}(t, \mathsf{diff}(t,u))$, $\mathsf{select}(u, \mathsf{diff}(t,u))$, $\mathsf{Adm}(\mathsf{diff}(t,u), \mathsf{len}(t))$, and $\mathsf{Adm}(\mathsf{diff}(t,u), \mathsf{len}(u))$ for all pairs of terms $t, u \in X$ of sort A. Then by Definition 11, $\mathsf{basis}_{\mathcal{T}}(X)$ can be obtained either as $\{\top\} \cup \mathrm{Sat}^1(\Downarrow \mathrm{Sat}^3(\Downarrow \mathrm{Sat}^2(\Downarrow X)))$ or as $\{\top\} \cup \mathrm{Sat}^1(\Downarrow \mathrm{Sat}^2(\Downarrow \mathrm{Sat}^3(\Downarrow X)))$. Indeed, saturation by Clause (2) and saturation by Clause (3) are independent of each other, whereas saturation by Clause (1) must be applied last, because Clauses (2) and (3) may add new terms with top symbol len or select, to which Clause (1) may apply. Consider $\{\top\} \cup \mathrm{Sat}^1(\Downarrow \mathrm{Sat}^2(\Downarrow \mathrm{Sat}^3(\Downarrow X)))$. This set is finite, because each saturation step is finite and none can reignite a previous one. Indeed, in inside-out order, $\triangleleft$-closure is finite because the proper subterm ordering is well-founded, and equality-closure is finite because it produces equalities but it does not apply to Boolean terms. Sat^3 adds terms of sorts V and prop from terms of sort A: its application is finite as $\mathsf{A} \neq \mathsf{V}$ and $\mathsf{A} \neq \mathsf{prop}$. The closure on top of Sat^3 adds equalities and proper subterms of new terms added by Sat^3; it cannot reignite Sat^3, because it does not add new terms of sort A. Sat^2 adds terms of sort L from terms of sort A: its application is finite as $\mathsf{A} \neq \mathsf{L}$ and it cannot reignite Sat^3 for the same reason. The closure on top of Sat^2 adds only equalities (the proper subterms of the terms of sort L added by Sat^2 are already present), and hence it can reignite neither Sat^3 nor Sat^2. The application of Sat^1 is finite, because it adds equalities from non-equational terms, and Sat^1 cannot reignite either Sat^3 or Sat^2 because $\mathsf{A} \neq \mathsf{prop}$. □

5.3 Suitable Leading Theories for Arrays, Maps, and Vectors

A model of Arr interprets an array as a *function from indices to values.* Let $\mathcal{U}$ and $\mathcal{V}$ be generic sets, and $\mathcal{V}^{\mathcal{U}}$ denote the set of the functions from $\mathcal{U}$ to $\mathcal{V}$. A subset $\mathcal{W} \subseteq \mathcal{V}^{\mathcal{U}}$ is an *updatable function set from $\mathcal{U}$ to $\mathcal{V}$*, if every function that differs from a function in $\mathcal{W}$ on finitely many elements of $\mathcal{U}$ is also in $\mathcal{W}$. The intuition is that if an array is interpreted as a function in $\mathcal{W}$, the array resulting from a finite number of updates by store operations is interpreted as a function which is also in $\mathcal{W}$. Therefore, a model $\mathcal{M}$ of Arr interprets the array sort A as an updatable function set from $\mathsf{I}^{\mathcal{M}}$ to $\mathsf{V}^{\mathcal{M}}$.

A model $\mathcal{M}$ of ArrAD interprets an array with abstract domain as a *function from admissible indices to values.* Since the admissibility of an index depends on the length of the array, model $\mathcal{M}$ interprets an array of length n as a function from the set of admissible indices for length n to $\mathsf{V}^{\mathcal{M}}$. For model $\mathcal{M}$, the set of admissible indices for length n is given by $I_n = \{i \mid i \in \mathsf{I}^{\mathcal{M}} \wedge \mathsf{Adm}^{\mathcal{M}}(i,n)\}$. Let X_n be the updatable function set from I_n to $\mathsf{V}^{\mathcal{M}}$, whose elements are used to interpret the arrays of length n. Then sort A is interpreted as the disjoint union of the sets X_n for all $n \in \mathsf{L}^{\mathcal{M}}$.

What is most important in the interpretation of a sort is its cardinality, because the theories need to agree on the cardinality of a shared sort. In CDSAT, it is the leading theory $\mathcal{T}_1$ that aggregates the requirements on cardinalities coming from the theories in the union, and global agreement is ensured by having each theory agree with $\mathcal{T}_1$. Therefore, $\mathcal{T}_1$ has all the sorts of ArrAD, and it is up to every $\mathcal{T}_1$-model $\mathcal{M}_1$ to ensure that the cardinality of sort A can be determined, based on the interpretation of Adm and of the other sorts (L, I, and V). The next definition collects these requirements.

Definition 12 (ArrAD-suitability)**.** *A leading theory $\mathcal{T}_1$ with signature $\Sigma_1 = (S_1, F_1, \mathit{ShF}_1)$ is* suitable for ArrAD*, or* ArrAD-suitable*, if $S_{\mathsf{ArrAD}} \subseteq S_1$, $\mathit{ShF}_{\mathsf{ArrAD}} \cap \mathit{ShF}_1 = \simeq_{S_{\mathsf{ArrAD}}} \cup \{\mathsf{Adm}\}$, and for all $\mathcal{T}_1$-models $\mathcal{M}_1$ there exists a length-indexed collection $(X_n)_{n \in \mathsf{L}^{\mathcal{M}_1}}$ of nonempty updatable function sets from $I_n = \{i \mid i \in \mathsf{I}^{\mathcal{M}_1} \wedge \mathsf{Adm}^{\mathcal{M}_1}(i,n)\}$ to $\mathsf{V}^{\mathcal{M}_1}$ such that $|\mathsf{A}^{\mathcal{M}_1}| = |\biguplus_{n \in \mathsf{L}^{\mathcal{M}_1}} X_n|$.*

Since ArrAD-suitability only formalizes sensible requirements on the cardinality of sort A, it does not restrict the realm of theories with which ArrAD can be combined.

Example 11. *Resuming Example 3, suppose that ArrAD interprets also V as $\mathbb{Z}$. A leading theory that interprets L, I, and Adm as dictated by LIA, and V as dictated by ArrAD is ArrAD-suitable. For all $n \in \mathbb{Z}$, the set I_n of admissible indices is $\{i \mid i \in \mathbb{Z} \wedge 0 \leq i < n\}$ and it is finite. For all $n \in \mathbb{Z}$, $n > 0$, the set X_n is the updatable function set of all functions from I_n to $\mathbb{Z}$, and hence it is countably infinite. Therefore, the cardinality of the interpretation of A is countably infinite. A leading theory interpreting A as being finite would not be ArrAD-suitable.*

Example 12. *For a different continuation of Example 3, suppose that ArrAD interprets V as a finite set D of cardinality m ($m > 0$). A leading theory that interprets L, I, and Adm as stipulated by LIA, and V as stipulated by ArrAD is ArrAD-suitable. For all $n \in \mathbb{Z}$, $n > 0$, the set X_n is the updatable function set of all functions from I_n*

to D, and hence its cardinality is m^n. Therefore, the cardinality of the interpretation of A *is countably infinite. Again, a leading theory interpreting* A *as being finite would not be* ArrAD*-suitable.*

Example 13. *Consider the union of* ArrAD *and the theory* BV *of bitvectors, where* $\mathsf{BV}[n]$ *is the set of bitvectors of length n. Assume that* BV *interprets* I *as* $\mathsf{BV}[1]$, L *as* $\mathsf{BV}[2]$*, and* Adm *as* true *everywhere except for the pairs $(0, 00)$, $(1, 00)$, and $(1, 01)$. Suppose that the two theories share also* V *and that* BV *interprets it as* $\mathsf{BV}[1]$*. A leading theory that interprets* I, L, Adm*, and* V *as stipulated by* BV *is* ArrAD*-suitable. The sets of admissible indices are $I_{00} = \emptyset$, $I_{01} = \{0\}$, and $I_{10} = I_{11} = \{0, 1\}$. For all $n \in$* $\mathsf{BV}[2]$*, X_n is the updatable function set of the functions from I_n to* $\mathsf{BV}[1]$*. The cardinalities of the X_n's are $|X_{00}| = 2^0 = 1$, $|X_{01}| = 2^1 = 2$, and $|X_{10}| = |X_{11}| = 2^2 = 4$. The cardinality of the interpretation of* A *is their sum, that is, $1+2+4+4 = 11$. A leading theory interpreting* A *as being countably infinite would not be* ArrAD*-suitable.*

Also a model $\mathcal{M}$ of MapAD interprets a map with abstract domain as a *function from admissible indices to values*, but for a map an update can change the domain. Therefore, the notion of updatable function set does not suffice. The appropriate closure property must be defined for the entire collection of function sets. A collection $(X_n)_n$ is an *updatable collection* if a function that differs from a function in some X_n on *finitely many* indices belongs to some X_m, where m and n may be equal or different. The following definition strenghtens this closure property to capture the fact that a store operation causes a map to grow by at most one index.

Definition 13 (Incrementally updatable collection)**.** *Let $\mathcal{U}$, $\mathcal{V}$, and $\mathcal{L}$ be nonempty sets, and $R \subseteq \mathcal{U} \times \mathcal{L}$ a binary relation. An $\mathcal{L}$-indexed collection $(X_n)_{n\in\mathcal{L}}$ is* incrementally updatable with respect to $\mathcal{U}$, $\mathcal{V}$, and R *if:*

- *For all $n \in \mathcal{L}$, X_n is a nonempty set of functions $f \colon I_n \to \mathcal{V}$ for $I_n = \{i \mid i \in \mathcal{U} \wedge R(i, n)\}$, and*
- *For all $i \in \mathcal{U}$, $v \in \mathcal{V}$, $n \in \mathcal{L}$, and $f \in X_n$, if f^i_v is the function that maps i to v and is identical to f otherwise, there exists an $m \in \mathcal{L}$ such that $f^i_v \in X_m$, and*
  - *$m = n$ if $i \in I_n$, and*
  - *$I_m = I_n \cup \{i\}$ otherwise.*

This kind of collection is used to interpret the sort A of maps in theory MapAD, and to define the MapAD-suitability of a leading theory, by replacing the generic $\mathcal{U}$, $\mathcal{V}$, $\mathcal{L}$, and R of Definition 13 with the interpretations of I, V, L, and Adm in a model of the leading theory.

Definition 14 (MapAD-suitability)**.** *A leading theory $\mathcal{T}_1$ with signature $\Sigma_1 = (S_1, F_1, ShF_1)$ is* suitable for MapAD*, or* MapAD-suitable*, if $S_{\mathsf{MapAD}} \subseteq S_1$, $ShF_{\mathsf{MapAD}} \cap ShF_1 = \simeq_{S_{\mathsf{MapAD}}} \cup \{\mathsf{Adm}\}$, and for all $\mathcal{T}_1$-models $\mathcal{M}_1$ there exists a length-indexed collection $(X_n)_{n\in\mathsf{L}^{\mathcal{M}_1}}$ such that $|\mathsf{A}^{\mathcal{M}_1}| = |\biguplus_{n\in\mathsf{L}^{\mathcal{M}_1}} X_n|$ and $(X_n)_{n\in\mathsf{L}^{\mathcal{M}_1}}$ is incrementally updatable with respect to $\mathsf{I}^{\mathcal{M}_1}$, $\mathsf{V}^{\mathcal{M}_1}$, and $\mathsf{Adm}^{\mathcal{M}_1}$.*

For vectors, a store at an inadmissible index makes also all smaller indices admissible. Therefore, Definition 13 is modified by adding an ordering and replacing equality with the smaller than or equal to relation.

Definition 15 (Extensibly updatable collection). *Let $\mathcal{U}$, $\mathcal{V}$, and $\mathcal{L}$ be nonempty sets, $R \subseteq \mathcal{U} \times \mathcal{V}$ a binary relation, and $<_u$ an ordering on $\mathcal{U}$. An $\mathcal{L}$-indexed collection $(X_n)_{n\in\mathcal{L}}$ is* extensibly updatable with respect to $\mathcal{U}$, $\mathcal{V}$, R, and $<_u$ *if:*

- *For all $n \in \mathcal{L}$, X_n is a nonempty set of functions $f\colon I_n \to \mathcal{V}$ for $I_n = \{i \mid i \in \mathcal{U} \wedge R(i,n)\}$, and*
- *For all $i \in \mathcal{U}$, $v \in \mathcal{V}$, $n \in \mathcal{L}$, and $f \in X_n$, if f^i_v is the function that maps i to v and is identical to f otherwise, there exists an $m \in \mathcal{L}$ such that $f^i_v \in X_m$, where*
  - *$m = n$ if $i \in I_n$, and*
  - *$I_m = I_n \cup \{j \mid j \leq_u i\}$ otherwise.*

This kind of collection is used to interpret the sort A of vectors in theory VecAD, and to define the VecAD-suitability of a leading theory, by replacing the generic $\mathcal{U}$, $\mathcal{V}$, $\mathcal{L}$, R, and $<_u$ of Definition 15 with the interpretations of I, V, L, Adm, and $<\in F_{\mathsf{VecAD}}$ in a model of the leading theory.

Definition 16 (VecAD-suitability). *A leading theory $\mathcal{T}_1$ with signature $\Sigma_1 = (S_1, F_1, ShF_1)$ is* suitable for VecAD*, or* VecAD-suitable*, if $S_{\mathsf{VecAD}} \subseteq S_1$, $ShF_{\mathsf{VecAD}} \cap ShF_1 = \simeq_{S_{\mathsf{VecAD}}} \cup \{\mathsf{Adm}, <\}$, and for all $\mathcal{T}_1$-models $\mathcal{M}_1$ there exists a length-indexed collection $(X_n)_{n\in\mathsf{L}^{\mathcal{M}_1}}$ such that $|\mathsf{A}^{\mathcal{M}_1}| = |\biguplus_{n\in\mathsf{L}^{\mathcal{M}_1}} X_n|$ and $(X_n)_{n\in\mathsf{L}^{\mathcal{M}_1}}$ is extensibly updatable with respect to $\mathsf{I}^{\mathcal{M}_1}$, $\mathsf{V}^{\mathcal{M}_1}$, $\mathsf{Adm}^{\mathcal{M}_1}$, and $<^{\mathcal{M}_1}$.*

A model interprets function symbols as total functions. Therefore, while the functions in a set X_n, member of a collection, are defined on I_n and not on $\mathsf{I}^{\mathcal{M}_1}$, the interpretation of select in an $\mathcal{T}$-model $\mathcal{M}$, for $\mathcal{T} \in \{\mathsf{ArrAD}, \mathsf{MapAD}, \mathsf{VecAD}\}$, will be defined on the entire $\mathsf{I}^{\mathcal{M}}$, so that every term $\mathsf{select}(a, i)$ gets interpreted.

5.4 Leading-Theory Completeness

For $\mathcal{T} \in \{\mathsf{ArrAD}, \mathsf{MapAD}, \mathsf{VecAD}\}$, the extension $\mathcal{T}^+$ may be either trivial or countably infinite (see Sect. 4.1). In this section we prove that $\mathcal{I}_{\mathcal{T}}$ is leading-theory-complete in the second case. The other case will follow as a simple variant. We begin with three auxiliary lemmas. Given a $\mathcal{T}$-assignment J, with $G_s(J)$ the set of terms of sort s occurring in J, the binary relation $\simeq^J_s \subseteq G_s(J) \times G_s(J)$ is defined by $t_1 \simeq^J_s t_2$ iff $(t_1 \simeq_s t_2) \in J$.

Lemma 2 (Lemma 1 [11]). *If $\mathcal{T}$-module $\mathcal{I}$ cannot expand a plausible $\mathcal{T}$-assignment J, then:*

1. *For all sorts $s \in S \setminus \{\mathsf{prop}\}$, the relation $\simeq^J_s$ is an equivalence relation, and if $\{t_1 \leftarrow \mathfrak{c}_1,\ t_2 \leftarrow \mathfrak{c}_2\} \subseteq J$, then $\mathfrak{c}_1$ and $\mathfrak{c}_2$ are identical if and only if $t_1 \simeq^J_s t_2$;*
2. *Assignment J gives a value to every formula that is relevant to $\mathcal{T}$ in J;*
3. *Assignment J gives a value to every term t of sort $s \in S \setminus \{\mathsf{prop}\}$ that is relevant to $\mathcal{T}$ in J, provided that (i) there exists a $\mathcal{T}$-value of sort s that J does not use, and*

(ii) the only $\mathcal{I}$-inferences involving first-order assignments of sort s are equality inferences.

Hypothesis (i) in Claim (3) ensures that a value for a decision is available. Hypothesis (ii) makes the analysis of acceptability of decisions module-independent. For $\mathcal{T} \in \{\mathsf{ArrAD}, \mathsf{MapAD}, \mathsf{VecAD}\}$, if $\mathcal{T}^+$ is countably infinite, Hypothesis (i) is satisfied a priori. If $\mathcal{T}^+$ is trivial, Claim (3) is vacuously true, as by Definition 5 no term t of a sort $s \in S \setminus \{\mathsf{prop}\}$ is relevant to $\mathcal{T}$.

Lemma 3. *For all theories $\mathcal{T}$ with signature Σ, if $\mathcal{T}$-module $\mathcal{I}$ cannot expand a plausible $\mathcal{T}$-assignment J, extension $\mathcal{T}^+$ is countably infinite, and the only $\mathcal{I}$-inferences involving first-order assignments are equality inferences, then (1) J assigns values to all terms in $G(J)$, and hence (2) $\mathsf{fv}_\Sigma(J \cup G(J)) = \mathsf{fv}_\Sigma(G(J)) = \mathsf{fv}_\Sigma(J)$.*

Proof. For all $l \in G_{\mathsf{prop}}(J)$, formula l is relevant to $\mathcal{T}$ by Condition (i) of Definition 5, and hence J assigns a value to l by Claim (2) of Lemma 2. For all sorts $s \in S \setminus \{\mathsf{prop}\}$ and all terms $u \in G_s(J)$, we reason as follows: first, u is relevant to $\mathcal{T}$ because Condition (i) of Definition 5 is satisfied by the hypotheses; second, J assigns a value to u because the conditions of Claim (3) of Lemma 2 are satisfied by the hypotheses. For Claim (2), the first equality is trivial, and the second one follows from Claim (1). Note that the second equality in Claim (2) is not trivial, because there could be two Σ-foreign terms $u, t \in G(J)$ such that $u \lhd t$, so that $u \in \mathsf{fv}_\Sigma(G(J))$, but $u \notin \mathsf{fv}_\Sigma(J)$. $\square$

Lemma 4. *For $\mathcal{T} \in \{\mathsf{ArrAD}, \mathsf{MapAD}, \mathsf{VecAD}\}$, if J is a plausible $\mathcal{T}$-assignment that $\mathcal{I}_\mathcal{T}$ cannot expand, for all terms $t \in G_{\mathsf{A}}(J)$, $\mathsf{len}(t) \in G_{\mathsf{L}}(J)$.*

Proof. Since $\mathcal{I}_\mathcal{T}$ cannot expand J, for all terms $t \in G_{\mathsf{A}}(J)$, we have $(t \simeq t) \in J$ by the equality rule for reflexivity (see Fig. 1), and hence $(\mathsf{len}(t) \simeq \mathsf{len}(t)) \in J$ by rule (18), so that $\mathsf{len}(t) \in G_{\mathsf{L}}(J)$. $\square$

Lemma 4 and the form of rule (18) contradict the notion that module $\mathcal{I}_\mathcal{T}$ only needs to be concerned with the lengths of arrays (maps, or vectors) that differ. In theories ArrAD, MapAD, and VecAD, the length is an essential feature of an array, a map, and a vector, respectively. The model construction in the proof of leading-theory-completeness of $\mathcal{I}_\mathcal{T}$ will define a length function as a step towards the functional interpretation of arrays (maps, and vectors).

Theorem 5. *For $\mathcal{T} \in \{\mathsf{ArrAD}, \mathsf{MapAD}, \mathsf{VecAD}\}$, module $\mathcal{I}_\mathcal{T}$ is leading-theory-complete for all $\mathcal{T}$-suitable leading theories.*

Proof. Let J be a plausible $\mathcal{T}$-assignment that $\mathcal{I}_\mathcal{T}$ cannot expand. We show that J is leading-theory-compatible with $\mathcal{T}$ sharing $G(J)$. By Lemma 3, J assigns values to all terms in $G(J)$ (†) and $\mathsf{fv}_\Sigma(J \cup G(J)) = \mathsf{fv}_\Sigma(G(J)) = \mathsf{fv}_\Sigma(J)$, where $\Sigma = (S, F, ShF)$ is the signature of $\mathcal{T}$. Let $\mathcal{T}_1$ be a $\mathcal{T}$-suitable leading theory, Σ_1 its signature, and $\mathcal{T}_1^+$ its extension. Let $\mathcal{M}_1$ be a $\mathcal{T}_1^+[\mathcal{V}_1]$-model such that $\mathsf{fv}_{\Sigma_1}(J \cup G(J)) = \mathsf{fv}_{\Sigma_1}(G(J)) \subseteq \mathcal{V}_1$, $\mathcal{M}_1 \models J_{\mathcal{T}_1}$, and $|\mathsf{A}^{\mathcal{M}_1}| = |\biguplus_{n \in \mathsf{L}^{\mathcal{M}_1}} X_n|$, where $(X_n)_{n \in L^{\mathcal{M}_1}}$ is

- The collection of updatable function sets for $\mathcal{M}_1$ according to Definition 12 for $\mathcal{T} = \mathsf{ArrAD}$;
- The incrementally updatable collection of function sets for $\mathcal{M}_1$ according to Definition 14 for $\mathcal{T} = \mathsf{MapAD}$;
- The extensibly updatable collection of function sets for $\mathcal{M}_1$ according to Definition 16 for $\mathcal{T} = \mathsf{VecAD}$.

We construct a $\mathcal{T}^+[\mathcal{V}]$-model $\mathcal{M}$ with $\mathsf{fv}_\Sigma(J) \subseteq \mathcal{V}$. We start with:

1. $s^{\mathcal{M}} = s^{\mathcal{M}_1}$ for all sorts $s \in S$ (towards (iii) in Def. 9),
2. $\mathcal{M}(t) = \mathcal{M}_1(t)$ for all variables $t \in \mathsf{fv}_\Sigma(J)$,
3. $\mathsf{Adm}^{\mathcal{M}} = \mathsf{Adm}^{\mathcal{M}_1}$ and, for $\mathcal{T} = \mathsf{VecAD}$, $<^{\mathcal{M}} = <^{\mathcal{M}_1}$ (towards (ii) and (iii) in Def. 9),
4. For all $\mathcal{T}$-values $\mathfrak{c}$, $\mathcal{M}(\mathfrak{c}) = \mathcal{M}_1(t)$ if $(t \leftarrow \mathfrak{c}) \in J$ and $\mathcal{M}(\mathfrak{c})$ is chosen arbitrarily otherwise.

The bulk of the construction is for the definition of the $\mathcal{M}$-interpretation of len, store, select, and diff. To this end, we will define functions

- $\mathit{len}\colon \mathsf{A}^{\mathcal{M}} \to \mathsf{L}^{\mathcal{M}}$ mapping arrays (maps, or vectors) to lengths;
- $\psi\colon \mathsf{A}^{\mathcal{M}} \to \biguplus_{n \in \mathsf{L}^{\mathcal{M}}} X_n$ such that $\psi(a) \in X_n$ for $n = \mathit{len}(a)$ and $\psi(a)\colon I_n \to \mathsf{V}^{\mathcal{M}}$ is a function from admissible indices to values;
- ϕ from $\mathsf{A}^{\mathcal{M}}$ to a function set from $\mathsf{I}^{\mathcal{M}}$ to $\mathsf{V}^{\mathcal{M}}$, so that $\phi(a)\colon \mathsf{I}^{\mathcal{M}} \to \mathsf{V}^{\mathcal{M}}$ is a function from indices to values that agrees with $\psi(a)$ on admissible indices;
- $\mathit{diff}\colon \mathsf{A}^{\mathcal{M}} \times \mathsf{A}^{\mathcal{M}} \to \mathsf{I}^{\mathcal{M}}$ mapping pairs of arrays (maps, or vectors) to indices.

The functions len, ψ, ϕ, and diff will be used to define the $\mathcal{M}$-interpretation of symbols len, store, select, and diff, respectively, in such a way that:

- $\mathcal{M} \models \mathcal{A}_{\mathcal{T}}$ so that $\mathcal{M}$ is a $\mathcal{T}^+$-model,
- $\mathcal{M} \models J$ in order to satisfy Part (i) of Definition 9, and
- The sort cardinality constraint $|\mathsf{A}^{\mathcal{M}}| = |\biguplus_{n \in \mathsf{L}^{\mathcal{M}}} X_n|$ conveyed from $\mathcal{M}_1$ to $\mathcal{M}$ by Point (1) above is respected, so that $\psi\colon \mathsf{A}^{\mathcal{M}} \to \biguplus_{n \in \mathsf{L}^{\mathcal{M}}} X_n$ will be a bijection.

The essence of the following construction is that for all inhabitants a of $\mathsf{A}^{\mathcal{M}}$, we give a *functional interpretation* mapping indices in $\mathsf{I}^{\mathcal{M}}$ to values in $\mathsf{V}^{\mathcal{M}}$. To this end, for all n in $\mathsf{L}^{\mathcal{M}}$ we pick from X_n a function $f_n\colon I_n \to \mathsf{V}^{\mathcal{M}}$ and we complete it into a function $g_n\colon \mathsf{I}^{\mathcal{M}} \to \mathsf{V}^{\mathcal{M}}$. These functions will be used as defaults in the construction, which is subdivided in four steps. In the first step we consider those inhabitants of $\mathsf{A}^{\mathcal{M}}$ that are used by $\mathcal{M}_1$ to interpret terms in $G_{\mathsf{A}}(J)$. Let Y be the finite subset of $\mathsf{A}^{\mathcal{M}}$ consisting of those elements a such that $\mathcal{M}_1(t) = a$ for some term $t \in G_{\mathsf{A}}(J)$. We define len_Y, ψ_Y, and ϕ_Y as the cores of len, ψ, and ϕ, respectively, that are defined only on Y.

I. *Definition of len_Y, ϕ_Y, and ψ_Y:*
   Let a be an element of Y with $a = \mathcal{M}_1(t)$ for term $t \in G_{\mathsf{A}}(J)$. By Lemma 4, $\mathsf{len}(t) \in G_{\mathsf{L}}(J)$. Model $\mathcal{M}_1$ sees $\mathsf{len}(t)$ as a variable in $\mathsf{fv}_{\Sigma_1}(J)$, because len is a Σ_1-foreign symbol. We define $\mathit{len}_Y\colon Y \to \mathsf{L}^{\mathcal{M}}$ by $\mathit{len}_Y(a) = \mathcal{M}_1(\mathsf{len}(t))$. Let $\mathcal{R}_a \subseteq \mathsf{I}^{\mathcal{M}} \times \mathsf{V}^{\mathcal{M}}$ be the set of index-value pairs dictated by J. Formally, $R_a = R_a^1 \cup R_a^2 \cup R_a^3$ for $\mathcal{T} = \mathsf{ArrAD}$, and $R_a = R_a^1 \cup R_a^2 \cup R_a^3 \cup R_a^4$ for $\mathcal{T} = \mathsf{MapAD}$ or $\mathcal{T} = \mathsf{VecAD}$, where:

   - $R_a^1 = \{(\mathcal{M}_1(i), \mathcal{M}_1(\mathsf{select}(t,i))) \mid \mathsf{select}(t,i) \in G_{\mathsf{V}}(J),\ \mathcal{M}_1(t) = a\}$,

- $R_a^2 = \{(\mathcal{M}_1(i), \mathcal{M}_1(\mathsf{select}(t,i))) \mid \mathsf{select}(t,i) \in G_{\mathsf{V}}(J),\ \mathsf{store}(t,j,v) \in G_{\mathsf{A}}(J),$ $\mathcal{M}_1(\mathsf{store}(t,j,v)) = a,\ \mathcal{M}_1(i) \neq \mathcal{M}_1(j)\}$,
- $R_a^3 = \{(\mathcal{M}_1(i), \mathcal{M}_1(v)) \mid \mathsf{store}(t,i,v) \in G_{\mathsf{A}}(J),\ \mathcal{M}_1(\mathsf{store}(t,i,v)) = a,$ $\mathcal{M}_1(t) = b,\ len_Y(b) = n,\ \mathcal{M}_1(i) \in I_n\}$, and
- $R_a^4 = \{(\mathcal{M}_1(i), \mathcal{M}_1(v)) \mid \mathsf{store}(t,i,v) \in G_{\mathsf{A}}(J),\ \mathcal{M}_1(\mathsf{store}(t,i,v)) = a,$ $\mathcal{M}_1(t) = b,\ len_Y(b) = n,\ \mathcal{M}_1(i) \notin I_n\}$.

Set R_a^1 contains the index-value pairs dictated by select-terms where select is applied to a term t that $\mathcal{M}_1$ interprets as a, so that $len_Y(a) = \mathcal{M}_1(\mathsf{len}(t))$. Sets R_a^2, R_a^3, and R_a^4 contain the index-value pairs dictated by a store-term that $\mathcal{M}_1$ interprets as a, so that $len_Y(a) = \mathcal{M}_1(\mathsf{len}(\mathsf{store}(t,i,v)))$ and $len_Y(b) = \mathcal{M}_1(\mathsf{len}(t)) = n$.
Set R_a^2 follows select-over-store axiom (2), which is common to all three theories. Set R_a^3 covers store-terms where the index is admissible, following select-over-store axiom (8) if $\mathcal{T} = \mathsf{ArrAD}$ and select-over-store axiom (1) if $\mathcal{T} = \mathsf{MapAD}$ or $\mathcal{T} = \mathsf{VecAD}$. Set R_a^4 covers store-terms where the index is inadmissible, following select-over-store axiom (1) for $\mathcal{T} = \mathsf{MapAD}$ or $\mathcal{T} = \mathsf{VecAD}$.
For each pair in R_a^3 (store at an admissible index), $len_Y(a) = len_Y(b) = n$, because otherwise $\mathcal{I}_{\mathcal{T}}$ could expand J by rule (23) if $\mathcal{T} = \mathsf{ArrAD}$, and by rule (27) if $\mathcal{T} = \mathsf{MapAD}$ or $\mathcal{T} = \mathsf{VecAD}$. For each pair in R_a^4 (store at an inadmissible index) and $\mathcal{T} = \mathsf{MapAD}$, $\mathcal{M}_1(i) \in I_m$ and $I_m = I_n \cup \{\mathcal{M}_1(i)\}$ for $len_Y(a) = m$, because otherwise $\mathcal{I}_{\mathsf{MapAD}}$ could expand J by rule (29). For each pair in R_a^4 (store at an inadmissible index) and $\mathcal{T} = \mathsf{VecAD}$, $\mathcal{M}_1(j) \in I_m$ for all indices $\mathcal{M}_1(j)$ such that $\mathcal{M}_1(j) \leq^{\mathcal{M}_1} \mathcal{M}_1(i)$, where $len_Y(a) = m$, because otherwise $\mathcal{I}_{\mathsf{VecAD}}$ could expand J by rule (32). Since $G(J)$ is finite, $\mathcal{R}_a$ is finite. Also, $\mathcal{R}_a$ is a functional relation, because otherwise $\mathcal{I}_{\mathcal{T}}$ could expand J by select-over-store rules (21)-(22) if $\mathcal{T} = \mathsf{ArrAD}$, and by select-over-store rules (21)-(26) if $\mathcal{T} = \mathsf{MapAD}$ or $\mathcal{T} = \mathsf{VecAD}$. The completion of this step is distinct for each of the three theories.

- If $\mathcal{T} = \mathsf{ArrAD}$: let $\phi_Y(a)\colon \mathsf{I}^{\mathcal{M}} \to \mathsf{V}^{\mathcal{M}}$ be the function that is identical to $\mathcal{R}_a = R_a^1 \cup R_a^2 \cup R_a^3$ where $\mathcal{R}_a$ is defined and to g_n for $n = len_Y(a)$ elsewhere. Let $\psi_Y(a)\colon I_n \to \mathsf{V}^{\mathcal{M}}$ be the restriction of $\phi_Y(a)$ to I_n. Since $\mathcal{R}_a$ is finite, $\phi_Y(a)$ differs from g_n at finitely many indices. Hence $\psi_Y(a)$ differs from f_n at finitely many indices, so that $\psi_Y(a) \in X_n$.
- If $\mathcal{T} = \mathsf{MapAD}$: since $\mathcal{R}_a = R_a^1 \cup R_a^2 \cup R_a^3 \cup R_a^4$ is finite, the incrementally updatable collection for $\mathcal{M}_1$ contains an X_h such that I_h contains all indices $\mathcal{M}_1(i)$ that appear as first element of a pair in $\mathcal{R}_a$. Let $\phi_Y(a)\colon \mathsf{I}^{\mathcal{M}} \to \mathsf{V}^{\mathcal{M}}$ be the function that is identical to $\mathcal{R}_a$ where $\mathcal{R}_a$ is defined and to g_h elsewhere. Let $\psi_Y(a)\colon I_h \to \mathsf{V}^{\mathcal{M}}$ be the restriction of $\phi_Y(a)$ to I_h. By the finiteness of $\mathcal{R}_a$ and Definition 13 of incrementally updatable collection, $\psi_Y(a) \in X_h$.
- If $\mathcal{T} = \mathsf{VecAD}$: since $\mathcal{R}_a = R_a^1 \cup R_a^2 \cup R_a^3 \cup R_a^4$ is finite, the extensibly updatable collection for $\mathcal{M}_1$ contains an X_h such that I_h contains (1) all indices $\mathcal{M}_1(i)$ that appear as first element of a pair in $\mathcal{R}_a$, and (2) all indices $\mathcal{M}_1(j)$ such that $\mathcal{M}_1(j) \leq^{\mathcal{M}_1} \mathcal{M}_1(i)$ for some $\mathcal{M}_1(i)$ that appears as first element of a pair in $\mathcal{R}_a^4$. Let $\phi_Y(a)\colon \mathsf{I}^{\mathcal{M}} \to \mathsf{V}^{\mathcal{M}}$ be the function that is identical to $\mathcal{R}_a$ where $\mathcal{R}_a$ is defined and to g_h elsewhere. Let $\psi_Y(a)\colon I_h \to \mathsf{V}^{\mathcal{M}}$ be the restriction of $\phi_Y(a)$ to I_h. By Definition 15 of extensibly updatable collection, $\psi_Y(a) \in X_h$.

II. *Injectivity of $\psi_Y : Y \to \biguplus_{n\in\mathsf{L}^{\mathcal{M}}} X_n$ and definition of $\mathit{diff}_Y : Y \times Y \to \mathsf{I}^{\mathcal{M}}$:*
By way of contradiction, suppose that there are elements $a, b \in Y$ such that $a \neq b$ and $\psi_Y(a) = \psi_Y(b)$. Since $\psi_Y(a)$ is a function in X_n for $n = \mathit{len}_Y(a)$ and $\psi_Y(b)$ is a function in X_m for $m = \mathit{len}_Y(b)$, the equality $\psi_Y(a) = \psi_Y(b)$ means that $X_n \cap X_m \neq \emptyset$. If $n \neq m$, then $I_n \neq I_m$ and hence $X_n \cap X_m = \emptyset$. Thus, $X_n \cap X_m \neq \emptyset$ implies $n = m$. By definition of Y, the assumption that $a, b \in Y$ means that there exist terms $t, u \in G_{\mathsf{A}}(J)$ such that $a = \mathcal{M}_1(t)$ and $b = \mathcal{M}_1(u)$, and the assumption that $a \neq b$ means that $\mathcal{M}_1 \models t \not\simeq u$. By (†) J assigns values to t and u, and therefore it also assigns a truth value $\mathfrak{b}$ to $t \simeq u$, because otherwise $\mathcal{I}_{\mathcal{T}}$ could expand J by an equality inference rule. Also, $((t \simeq u) \leftarrow \mathfrak{b}) \in J_{\mathcal{T}_1}$ because a Boolean assignment belongs to the theory view of every theory (see Def. 3). Since $\mathcal{M}_1 \models t \not\simeq u$ and $\mathcal{M}_1 \models J_{\mathcal{T}_1}$, the value $\mathfrak{b}$ must be false, that is, $(t \not\simeq u) \in J$.
By Lemma 4, $\mathsf{len}(t)$ and $\mathsf{len}(u)$ are in $G_{\mathsf{L}}(J)$, and J assigns them values by (†). Thus, J assigns a truth value $\mathfrak{b}'$ to $\mathsf{len}(t) \simeq \mathsf{len}(u)$ and so does $J_{\mathcal{T}_1}$. Since $\mathit{len}_Y(a) = \mathit{len}_Y(b)$, by definition of len_Y we have $\mathcal{M}_1(\mathsf{len}(t)) = \mathcal{M}_1(\mathsf{len}(u))$. Since $\mathcal{M}_1 \models J_{\mathcal{T}_1}$, the truth value $\mathfrak{b}'$ must be true (i.e., $(\mathsf{len}(t) \simeq \mathsf{len}(u)) \in J$). Also, $\mathsf{select}(t, \mathsf{diff}(t, u)) \not\simeq \mathsf{select}(u, \mathsf{diff}(t, u))$ is in J (*) and hence in $J_{\mathcal{T}_1}$, because otherwise $\mathcal{I}_{\mathcal{T}}$ could expand J by rule (24). From $\mathcal{M}_1 \models J_{\mathcal{T}_1}$ we have $\mathcal{M}_1(\mathsf{select}(t, \mathsf{diff}(t, u))) \neq \mathcal{M}_1(\mathsf{select}(u, \mathsf{diff}(t, u)))$.
Now we define diff_Y. By (*) $\mathsf{diff}(t, u) \in G_{\mathsf{I}}(J)$. Model $\mathcal{M}_1$ sees $\mathsf{diff}(t, u)$ as a variable in $\mathsf{fv}_{\Sigma_1}(J)$ because diff is a Σ_1-foreign symbol. For all $a, b \in Y$, let $\mathit{diff}_Y(a, b) = \mathcal{M}_1(\mathsf{diff}(t, u))$, if $a \neq b$ and $\mathit{len}_Y(a) = \mathit{len}_Y(b)$, and let $\mathit{diff}_Y(a, b)$ be arbitrary otherwise.
We resume the proof of the injectivity of ψ_Y. Also $\mathsf{Adm}(\mathsf{diff}(t, u), \mathsf{len}(t))$ is in J and hence in $J_{\mathcal{T}_1}$, because otherwise $\mathcal{I}_{\mathcal{T}}$ could expand J by rule (25). Since $\mathcal{M}_1 \models J_{\mathcal{T}_1}$, it follows that $\mathcal{M}_1 \models \mathsf{Adm}(\mathsf{diff}(t, u), \mathsf{len}(t))$. Thus, $\mathcal{M}_1((\mathsf{diff}(t, u)))$ is an admissible index (i.e., it is in I_n for $n = \mathcal{M}_1(\mathsf{len}(t))$). By definition of $\psi_Y(a) : I_n \to \mathsf{V}^{\mathcal{M}}$ (based on $\mathcal{R}_a$) for a generic a, we have:

$$\psi_Y(a)(\mathcal{M}_1(\mathsf{diff}(t, u))) = \mathcal{M}_1(\mathsf{select}(t, \mathsf{diff}(t, u)))$$
$$\psi_Y(b)(\mathcal{M}_1(\mathsf{diff}(t, u))) = \mathcal{M}_1(\mathsf{select}(u, \mathsf{diff}(t, u))).$$

Since the two right hand sides are different, the two left hand sides are also different, so that $\psi_Y(a) \neq \psi_Y(b)$, which contradicts the assumption that $\psi_Y(a) = \psi_Y(b)$. Therefore, ψ_Y is injective. The injectivity of ψ_Y allows us to define ψ, len, ϕ, and diff as extensions of ψ_Y, len_Y, ϕ_Y, and diff_Y.

III. *Definition of $\psi : \mathsf{A}^{\mathcal{M}} \to \biguplus_{n\in\mathsf{L}^{\mathcal{M}}} X_n$, $\mathit{len} : \mathsf{A}^{\mathcal{M}} \to \mathsf{L}^{\mathcal{M}}$, ϕ, and $\mathit{diff} : \mathsf{A}^{\mathcal{M}} \times \mathsf{A}^{\mathcal{M}} \to \mathsf{I}^{\mathcal{M}}$:*

- Since $\psi_Y : Y \to \biguplus_{n\in\mathsf{L}^{\mathcal{M}}} X_n$ is injective, we can extend it to a bijection ψ from $\mathsf{A}^{\mathcal{M}}$ to $\biguplus_{n\in\mathsf{L}^{\mathcal{M}}} X_n$ by taking as pre-images of the elements of $\biguplus_{n\in\mathsf{L}^{\mathcal{M}}} X_n$ that are not images of elements of Y other elements of $\mathsf{A}^{\mathcal{M}}$ and there are enough distinct such elements as $|\mathsf{A}^{\mathcal{M}}| = |\biguplus_{n\in\mathsf{L}^{\mathcal{M}}} X_n|$.
- For all $a \in \mathsf{A}^{\mathcal{M}}$ let $\mathit{len}(a)$ be the unique n in $\mathsf{L}^{\mathcal{M}}$ such that $\psi(a)$ is in X_n. Note that for $a \in Y$ we have $\mathit{len}(a) = \mathit{len}_Y(a)$.
- For all $a \in \mathsf{A}^{\mathcal{M}}$, if $a \in Y$ let $\phi(a) = \phi_Y(a)$; otherwise, let $\phi(a)$ be the function that agrees with $\psi(a)$ on I_n for $n = \mathit{len}(a)$ and with g_n everywhere else.

- For all $a, b \in \mathsf{A}^{\mathcal{M}}$, if $a, b \in Y$ let $\mathit{diff}(a,b) = \mathit{diff}_Y(a,b)$; otherwise, if $a = b$ or $a \neq b$ and $\mathit{len}(a) \neq \mathit{len}(b)$, let $\mathit{diff}(a,b)$ be arbitrary; otherwise (i.e., $a \neq b$ and $\mathit{len}(a) = \mathit{len}(b)$), let $\mathit{diff}(a,b) = j$ for any index $j \in I_n$, where $n = \mathit{len}(a)$, such that $\psi(a)(j) \neq \psi(b)(j)$. At least one such j exists, because $a \neq b$ implies $\psi(a) \neq \psi(b)$ by injectivity of ψ.

IV. *The $\mathcal{M}$-interpretation of symbols* len, diff, select, *and* store:

- For all $a \in \mathsf{A}^{\mathcal{M}}$ let $\mathsf{len}^{\mathcal{M}}(a) = \mathit{len}(a) \in \mathsf{L}^{\mathcal{M}}$;
- For all $a, b \in \mathsf{A}^{\mathcal{M}}$ let $\mathsf{diff}^{\mathcal{M}}(a,b) = \mathit{diff}(a,b) \in \mathsf{I}^{\mathcal{M}}$;
- For all pairs $(a,e) \in \mathsf{A}^{\mathcal{M}} \times \mathsf{I}^{\mathcal{M}}$ let $\mathsf{select}^{\mathcal{M}}(a,e) = \phi(a)(e) \in \mathsf{V}^{\mathcal{M}}$;
- For all triples $(a,e,v) \in \mathsf{A}^{\mathcal{M}} \times \mathsf{I}^{\mathcal{M}} \times \mathsf{V}^{\mathcal{M}}$ let $\mathsf{store}^{\mathcal{M}}(a,e,v)$ be defined assuming $\mathit{len}(a) = n$ and distinguishing the following cases:
  - If $e \in I_n$, then let $f\colon I_n \to \mathsf{V}^{\mathcal{M}}$ be the function such that $f(e) = v$ and for all $j \in I_n$, $j \neq e$, $f(j) = \psi(a)(j) \in \mathsf{V}^{\mathcal{M}}$. If $\mathcal{T} = \mathsf{ArrAD}$, function f is in X_n as it differs from $\psi(a) \in X_n$ at one index. If $\mathcal{T} = \mathsf{MapAD}$ or $\mathcal{T} = \mathsf{VecAD}$, function f is in X_n by Definition 13 or by Definition 15, respectively. Since ψ is a bijection, we take $\psi^{-1}(f)$ which is in $\mathsf{A}^{\mathcal{M}}$ and define $\mathsf{store}^{\mathcal{M}}(a,e,v) = \psi^{-1}(f)$.
  - If $e \notin I_n$ and $\mathcal{T} = \mathsf{ArrAD}$, then let $\mathsf{store}^{\mathcal{M}}(a,e,v) = a$.
  - If $e \notin I_n$ and $\mathcal{T} = \mathsf{MapAD}$ or $\mathcal{T} = \mathsf{VecAD}$, then by Definition 13 for MapAD, there exists an $m \in \mathsf{L}^{\mathcal{M}}$ such that $I_m = I_n \cup \{e\}$, and by Definition 15 for VecAD, there exists an $m \in \mathsf{L}^{\mathcal{M}}$ such that $I_m = I_n \cup \{j \mid j \in \mathsf{I}^{\mathcal{M}},\ j \leq^{\mathcal{M}} e\}$. Let $f\colon I_m \to \mathsf{V}^{\mathcal{M}}$ be the function such that $f(e) = v$ and for all $j \in I_m$, $j \neq e$, $f(j) = \psi(a)(j) \in \mathsf{V}^{\mathcal{M}}$. Function f is in X_m by Definition 13 for MapAD, and by Definition 15 for VecAD. Since ψ is a bijection, we take $\psi^{-1}(f)$ which is in $\mathsf{A}^{\mathcal{M}}$ and define $\mathsf{store}^{\mathcal{M}}(a,e,v) = \psi^{-1}(f)$.

By construction, $\mathcal{M} \models \mathcal{A}_{\mathcal{T}}$ and $\mathcal{M}$ fulfills Parts (i), (ii), and (iii) of Def. 9 (the instance of (ii) for equality follows by induction on the term structure).

$\square$

For $\mathcal{T} \in \{\mathsf{ArrAD}, \mathsf{MapAD}, \mathsf{VecAD}\}$, if $\mathcal{T}^+$ is trivial, Theorem 5 still holds with a similar proof, except that non-Boolean terms are not relevant to $\mathcal{T}$ and hence they are not assigned $\mathcal{T}$-values.

6 Global Basis for Predicate-Sharing Unions

In prior work [11, Sect. 5], we showed how to construct a finite and stable (hence closed, see Sect. 4.3) global basis $\mathcal{B}$ from local bases $\mathsf{basis}_1, \ldots, \mathsf{basis}_n$ for disjoint theories $\mathcal{T}_1, \ldots, \mathcal{T}_n$. In this section we generalize the construction to predicate-sharing theories.

The availability of $\mathsf{basis}_1, \ldots, \mathsf{basis}_n$ does not necessarily imply the existence of a finite $\mathcal{B}$, because a circular behavior may occur. Let H be the input assignment and let X_0 be equal to the set $G(H)$ of the terms occurring in H. Suppose that module $\mathcal{I}_k$ introduces a new term u_0 in $Y_0 = \mathsf{basis}_k(X_0)$, which causes another module $\mathcal{I}_j$ ($j \neq k$) to generate a new term t_1 in $X_1 = \mathsf{basis}_j(Y_0)$, which in turn leads $\mathcal{I}_k$ to create a new term u_1 in $Y_1 = \mathsf{basis}_k(X_1)$, and so on. For $Y_m = \mathsf{basis}_k(X_m)$ and $X_{m+1} = \mathsf{basis}_j(Y_m)$, the set $\bigcup_{m \geq 0} X_m$ will be infinite, even if for all m the sets Y_m and X_m are finite.

In order to avoid this kind of behavior, we define a *well-founded ordering on the theories* (and hence on their theory modules and local bases). A well-founded ordering is acyclic. Since there are finitely many theories in the union, being cyclic is the only way a theory ordering may be non-well-founded.

The aim of the theory ordering is to capture the dependencies between theories in the following sense. Intuitively, a theory $\mathcal{T}_j$ *depends* on a theory $\mathcal{T}_k$ if there exists a sort s such that s is shared by $\mathcal{T}_j$ and $\mathcal{T}_k$, module $\mathcal{I}_k$ produces new terms of sort s, and module $\mathcal{I}_j$ consumes terms of sort s, according to the notions of *production and consumption of a sort* defined below. These notions are defined for local bases, because the new terms produced by a theory module must be in its local basis. In words, a basis *produces* a sort s if its application to a closed set X generates some new term of sort s. A basis *consumes* a sort s if its application to a set $X \uplus \{t\}$, where X is closed and t is a term of sort s, generates some new term u that would not arise if the basis were applied to X only. In the disjoint case the term t can be either a variable or an equality (cf. [11, Def. 7]). In the predicate-sharing case the term t can be either a variable or a Boolean term whose top symbol is a shared predicate.

Definition 17 (Predicate-sharing production/consumption of a sort)**.** *Let* basis *be a basis for theory* $\mathcal{T}$ *with signature* $\Sigma = (S, F, ShF)$*. For all sorts* $s \in S$*, (i)* basis produces sort s *if for some term* $t\colon s$ *and some closed set* X *of terms,* $t \in \mathsf{basis}(X) \setminus X$*; and (ii)* basis consumes sort s *if there exist a closed set* X *of terms and a term* $t\colon s$*, which is either a free* Σ*-variable or a Boolean term* $p(u_1, \ldots, u_m)$ *with* $p \in ShF$ *and* $u_i \in X$ *for all* i*,* $1 \leq i \leq m$*, such that* $\mathsf{basis}(X \uplus \{t\}) \not\subseteq \Downarrow(\mathsf{basis}(X) \uplus \{t\})$*.*

What term t can be depends on what suffices for the forthcoming Lemma 6.

Example 14. *For* $\mathcal{T} \in \{\mathsf{ArrAD}, \mathsf{MapAD}, \mathsf{VecAD}\}$ *and* Σ *the signature of* $\mathcal{T}$*, the basis* $\mathsf{basis}_{\mathcal{T}}$ *(cf. Def. 11) produces sort* prop *by adding* $\top$ *and the equalities in Clause (1). Furthermore, Clause (1) does not consume any sort, because the terms it applies to have either* select *or* diff *or* store *or* len *as top symbol, and hence they are neither free* Σ*-variables nor Boolean terms with a shared predicate symbol. Clause (2) in Def. 11 produces sort* L *and consumes sort* A*, because given any variable* t *of sort* A*, it yields term* $\mathsf{len}(t)$ *of sort* L*. Clause (3) in Def. 11 produces sorts* I*,* V*,* L*, and* prop*, and consumes sort* A*, because given any variables* t *and* u *of sort* A*, it introduces terms* $\mathsf{diff}(t, u)$ *of sort* I*, terms* $\mathsf{select}(t, \mathsf{diff}(t, u))$ *and* $\mathsf{select}(u, \mathsf{diff}(t, u))$ *of sort* V*, terms* $\mathsf{len}(t)$ *and* $\mathsf{len}(u)$ *of sort* L*, and terms* $\mathsf{Adm}(\mathsf{diff}(t, u), \mathsf{len}(t))$ *and* $\mathsf{Adm}(\mathsf{diff}(t, u), \mathsf{len}(u))$ *of sort* prop*. In summary,* $\mathsf{basis}_{\mathcal{T}}$ *consumes sort* A *and produces sorts* prop*,* L*,* I*, and* V*.*

Definition 18 (Dependency ordering)**.** *Given theories* $\mathcal{T}_1, \ldots, \mathcal{T}_n$*, with sets of sorts* $S_1, \ldots, S_n$*, and local bases* $\mathsf{basis}_1, \ldots, \mathsf{basis}_n$*, the* dependency ordering $\prec$ *on* $\mathcal{T}_1, \ldots, \mathcal{T}_n$ *is defined as follows: for all* k *and all* j*,* $1 \leq k \neq j \leq n$*,* $\mathcal{T}_k \prec \mathcal{T}_j$ *if there exists a sort* $s \in S_k \cap S_j$ *that* basis_k *produces and* basis_j *consumes.*

Example 15. *Consider theories* ArrAD_1 *and* ArrAD_2 *as in Ex. 1 and 2. Let* basis_1 *and* basis_2 *be their respective local bases. As seen in Ex. 14 above,* basis_1 *consumes sort* $\mathsf{A}_1 = (\mathsf{I} \overset{\mathsf{L}}{\Rightarrow} \mathsf{V})$ *and produces sorts* prop*,* I*,* L*, and* $\mathsf{V}_1 = \mathsf{V}$*;* basis_2 *consumes sort*

$\mathsf{A}_2 = (\mathsf{I} \overset{\mathsf{L}}{\Rightarrow} (\mathsf{I} \overset{\mathsf{L}}{\Rightarrow} \mathsf{V}))$ *and produces sorts* prop, L, I, *and* $\mathsf{V}_2 = \mathsf{A}_1$. *The two theories share sorts* $\{\mathsf{prop}, \mathsf{I}, \mathsf{L}, \mathsf{A}_1\}$. *The relation* $\mathsf{ArrAD}_2 \prec \mathsf{ArrAD}_1$ *holds, because there exists a shared sort, namely* A_1, *such that* basis_2 *produces* A_1 *(as* V_2*) and* basis_1 *consumes* A_1. *On the other hand,* $\mathsf{ArrAD}_1 \prec \mathsf{ArrAD}_2$ *does not hold, because for no sort* $s \in \{\mathsf{prop}, \mathsf{I}, \mathsf{L}, \mathsf{A}_1\}$ *it is the case that* basis_1 *produces* s *and* basis_2 *consumes* s. *Indeed, for* $s \in \{\mathsf{prop}, \mathsf{I}, \mathsf{L}\}$, basis_1 *produces* s, *but* basis_2 *does not consume it. For* $s = \mathsf{A}_1$, basis_1 *does not produce* s. *In general, consider a predicate-sharing union with* m *(*$m \geq 2$*) instances* $\mathsf{ArrAD}_1, \ldots, \mathsf{ArrAD}_m$ *of* ArrAD, *where for all* j, $1 \leq j \leq m$, *there are* j *occurrences of the array sort constructor* $\Rightarrow$ *in sort* A_j. *We have* $\mathsf{ArrAD}_{j+1} \prec \mathsf{ArrAD}_j$ *for all* j, $1 \leq j \leq m$, *as* basis_{j+1} *produces as values arrays that* basis_j *consumes. By transitivity, we have* $\mathsf{ArrAD}_k \prec \mathsf{ArrAD}_j$ *for all* j *and* k, $1 \leq j < k \leq m$. *In other words, the inverse of the ordering on the number of occurrences of* $\Rightarrow$ *in the array sorts yields an ordering on the array sorts, and hence on the theories, since each instance of* ArrAD *has one array sort.*

If the dependency ordering $\prec$ is acyclic, the theories in the union can be numbered according to the ordering: if $\mathcal{T}_k \prec \mathcal{T}_j$ then $k < j$. Also, the numbering can be used to convey the ordering by stipulating that for all k and all j, $1 \leq k \neq j \leq n$, $k < j$ iff $\mathcal{T}_k \prec \mathcal{T}_j$. For the rest of this section, we assume that the theories are numbered according to this convention.[4]

Definition 19 (Dependency-induced global basis)**.** *Let* $\mathcal{T}_1, \ldots, \mathcal{T}_n$ *be theories ordered by an acyclic dependency ordering* $\prec$ *and numbered accordingly. Let* $\mathsf{basis}_1, \ldots, \mathsf{basis}_n$ *be their local bases. Then the* dependency-induced global basis *is defined by* $\mathsf{basis}_\infty(X) = \mathsf{basis}_n(\ldots \mathsf{basis}_1(X))$ *for all sets* X *of terms.*

We show that $\mathsf{basis}_\infty(X)$ is stable and finite for all sets X of terms. We begin with a lemma that establishes the needed permutability property.

Lemma 6. *If* $\mathcal{T}_1, \ldots, \mathcal{T}_n$ *are predicate-sharing theories ordered by an acyclic dependency ordering* $\prec$ *and numbered accordingly, then* $\forall k, \forall j, 1 \leq k < j \leq n$, *and for all finite closed sets* X *of terms, the following claims hold:*

1. *For all* $\trianglelefteq$*-closed sets* Y *of terms such that* $X \subseteq Y \subseteq \mathsf{basis}_j(X)$, *it holds that* $\mathsf{basis}_k(Y) \subseteq \Downarrow(\mathsf{basis}_k(X) \cup Y)$*; and*
2. $\mathsf{basis}_k(\mathsf{basis}_j(X)) \subseteq \mathsf{basis}_j(\mathsf{basis}_k(X))$.

Proof. By acyclicity of the ordering, $k < j$ (i.e., $\mathcal{T}_k \prec \mathcal{T}_j$) implies $\mathcal{T}_j \not\prec \mathcal{T}_k$, so that no sort $s \in S_k \cap S_j$ is produced by basis_j and consumed by basis_k $(*)$.

1. The proof of Claim (1) is by induction on the cardinality $|Y \setminus X|$ of $Y \setminus X$, which is finite, because X is finite by hypothesis, $\mathsf{basis}_j(X)$ is finite by finiteness of basis_j, and $Y \subseteq \mathsf{basis}_j(X)$ by hypothesis.
   Base case: if $|Y \setminus X| = 0$, $Y = X$, $\mathsf{basis}_k(X) \cup X = \mathsf{basis}_k(X)$ by extensiveness of basis_k, and $\mathsf{basis}_k(X) \subseteq \Downarrow(\mathsf{basis}_k(X))$ by definition of closure.
   Induction hypothesis: Claim (1) holds for all sets Y such that $|Y \setminus X| = q \geq 0$.
   Induction step: suppose $|Y \setminus X| = q+1$. Let t be a term of largest size (symbol count)

[4]This convention clashes with what we observed for the theories in Ex. 15, but in general it goes well with the intuition that $\mathcal{T}_k \prec \mathcal{T}_j$ means that $\mathcal{T}_j$ depends on $\mathcal{T}_k$.

in $Y \setminus X$ $(**)$ and such that the sort s of t is in $S_k \cap S_j$. By hypothesis, $t \in \mathsf{basis}_j(X)$, and hence $t \in (\mathsf{basis}_j(X) \setminus X)$. Thus, basis_j produces sort s by (i) in Def. 17. By $(*)$, basis_k does not consume sort s. By the contrapositive of (ii) in Def. 17, basis_k does not consume sort s implies that for all closed sets X of terms and for all terms $t\colon s$, that are either free Σ_k-variables or Boolean terms $p(u_1, \ldots, u_m)$ with $p \in \mathit{ShF}_k$ and $u_i \in X$ for all i, $1 \leq i \leq m$, we have $\mathsf{basis}(X \uplus \{t\}) \subseteq \Downarrow(\mathsf{basis}(X) \uplus \{t\})$ (c-ii). We want to apply (c-ii) to basis_k and the closed set $\Downarrow(Y \setminus \{t\})$. In order to do it, we need to show that t is either a free Σ_k-variable or a Boolean term as described in (c-ii). Since the theories are predicate-sharing, symbol $\mathit{top}(t)$ is either (a) unknown to $\mathcal{T}_k$ (i.e., $\mathit{top}(t)$ is Σ_k-foreign), or (b) unknown to $\mathcal{T}_j$ (i.e., $\mathit{top}(t)$ is Σ_j-foreign), or (c) known to both (i.e., $\mathit{top}(t) \in \mathit{ShF}_j \cap \mathit{ShF}_k$).

(a) If $\mathit{top}(t)$ is Σ_k-foreign, t is a free Σ_k-variable.

(b) If $\mathit{top}(t)$ is Σ_j-foreign, t is a Σ_j-foreign term, and since $t \in \mathsf{basis}_j(X)$, we have $t \in \mathsf{fv}_{\Sigma_j}(\mathsf{basis}_j(X))$. By the "no introduction of foreign terms" property of a local basis (cf. Def. 4), $\mathsf{fv}_{\Sigma_j}(\mathsf{basis}_j(X)) \subseteq \mathsf{fv}_{\Sigma_j}(X) \cup \mathcal{V}_\infty$. Term t cannot be in $\mathsf{fv}_{\Sigma_j}(X)$, because if it were, since X is closed, and hence $\trianglelefteq$-closed, t would also be in X, which is not the case by $(**)$. Thus, t must be in $\mathcal{V}_\infty$, which implies that t is a free Σ_k-variable.

(c) If $\mathit{top}(t) \in \mathit{ShF}_j \cap \mathit{ShF}_k$, term t is a Boolean term $p(u_1, \ldots, u_m)$ with $p \in \mathit{ShF}_k$. Furthermore, by $\trianglelefteq$-closure of Y, all strict subterms of t are in $Y \setminus \{t\}$, and hence in $\Downarrow(Y \setminus \{t\})$.

Therefore, we can apply (c-ii) to basis_k and the closed set $\Downarrow(Y \setminus \{t\})$, getting

$$\mathsf{basis}_k(\Downarrow(Y \setminus \{t\}) \cup \{t\}) \quad \subseteq \quad \Downarrow(\mathsf{basis}_k(\Downarrow(Y \setminus \{t\})) \cup \{t\}) \ (\dagger).$$

Also, we have $X \subseteq (Y \setminus \{t\})$, because $X \subseteq Y$ and $t \in Y \setminus X$. Next, we show that $Y \setminus \{t\}$ is $\trianglelefteq$-closed. By way of contradiction, suppose that $Y \setminus \{t\}$ is not $\trianglelefteq$-closed. Since Y is $\trianglelefteq$-closed, by the hypothesis that $Y \setminus \{t\}$ is not, we have that there exists a term $u \in Y$ such that $t \triangleleft u$. But then, either $u \in X$ or $u \in Y \setminus X$. If $u \in X$, then $t \in X$, because X is closed, and $t \in X$ contradicts $(**)$. If $u \in Y \setminus X$, then t is not a term of largest size in $Y \setminus X$ which contradicts $(**)$. Therefore, we can apply the induction hypothesis to $Y \setminus \{t\}$ and get

$$\mathsf{basis}_k(Y \setminus \{t\}) \ \subseteq \Downarrow(\mathsf{basis}_k(X) \cup (Y \setminus \{t\})) \ (\ddagger).$$

Then Claim (1) is established as follows:

$$\begin{array}{rll}
\mathsf{basis}_k(Y) & = \mathsf{basis}_k((Y \setminus \{t\}) \cup \{t\}) & \\
& \subseteq \mathsf{basis}_k(\Downarrow(Y \setminus \{t\}) \cup \{t\}) & \text{by monotonicity of } \mathsf{basis}_k \\
& \subseteq \Downarrow(\mathsf{basis}_k(\Downarrow(Y \setminus \{t\})) \cup \{t\}) & \text{by } (\dagger) \\
& = \Downarrow(\mathsf{basis}_k(Y \setminus \{t\}) \cup \{t\}) & \text{by closure of } \mathsf{basis}_k \\
& \subseteq \Downarrow(\Downarrow(\mathsf{basis}_k(X) \cup (Y \setminus \{t\})) \cup \{t\}) & \text{by } (\ddagger) \\
& \subseteq \Downarrow\Downarrow(\mathsf{basis}_k(X) \cup (Y \setminus \{t\}) \cup \{t\}) & \\
& = \Downarrow(\mathsf{basis}_k(X) \cup Y) & \text{by idempotence of } \Downarrow.
\end{array}$$

2. Claim (2) is derived as follows:

$$\begin{array}{lll}
\mathsf{basis}_k(\mathsf{basis}_j(X)) & \subseteq \Downarrow(\mathsf{basis}_k(X) \cup \mathsf{basis}_j(X)) & \text{by Claim (1)} \\
& \subseteq \Downarrow(\mathsf{basis}_j(\mathsf{basis}_k(X)) \cup \mathsf{basis}_j(X)) & \text{by extensiveness} \\
& = \Downarrow(\mathsf{basis}_j(\mathsf{basis}_k(X))) & \text{by } X \subseteq \mathsf{basis}_k(X) \\
& \quad \text{and the monotonicity of } \mathsf{basis}_j & \\
& = \mathsf{basis}_j(\mathsf{basis}_k(X)) & \text{by closure of bases.}
\end{array}$$

□

Now we can use the premutability property (Claim (2) in Lemma 6) to show that $\mathsf{basis}_\infty(X)$ is stable.

Lemma 7. *Let $\mathcal{T}_1, \ldots, \mathcal{T}_n$ be predicate-sharing theories ordered by an acyclic dependency ordering $\prec$ and numbered accordingly. Let $\mathsf{basis}_1, \ldots, \mathsf{basis}_n$ be their local bases and basis_∞ be the dependency-induced global basis. Then for all finite sets X of terms, $\forall k, 1 \le k \le n, \mathsf{basis}_k(\mathsf{basis}_\infty(X)) = \mathsf{basis}_\infty(X)$.*

Proof. We prove a more general claim, namely that for all k, for all j, $1 \le k \le j \le n$, $\mathsf{basis}_k(\mathsf{basis}_j(\ldots \mathsf{basis}_1(X))) = \mathsf{basis}_j(\ldots \mathsf{basis}_1(X))$. The $\supseteq$-direction holds by extensiveness of basis_k. The proof of the $\subseteq$-direction is by induction on j.
Base case: if $j = k$, the claim holds by idempotence of basis_j.
Induction hypothesis: the claim is true for j.
Induction step: we prove the claim for $j + 1$. Let Z stand for the expression $\mathsf{basis}_j(\ldots \mathsf{basis}_1(X))$. Since $k \le j$ it is $k < j + 1$. Since Z is finite and closed by finiteness and closure of the bases, by Claim (2) in Lemma 6 we get $\mathsf{basis}_k(\mathsf{basis}_{j+1}(Z)) \subseteq \mathsf{basis}_{j+1}(\mathsf{basis}_k(Z))$ $(*)$. Then, $\mathsf{basis}_k(Z) \subseteq Z$ holds by the induction hypothesis, and $\mathsf{basis}_{j+1}(\mathsf{basis}_k(Z)) \subseteq \mathsf{basis}_{j+1}(Z)$ $(**)$ follows by monotonicity of basis_{j+1}. By chaining $(*)$ and $(**)$ we get $\mathsf{basis}_k(\mathsf{basis}_{j+1}(Z)) \subseteq \mathsf{basis}_{j+1}(Z)$ as desired. □

Theorem 8. *Let $\mathcal{T}_\infty$ be the predicate-sharing union of theories $\mathcal{T}_1, \ldots, \mathcal{T}_n$ ordered by an acyclic dependency ordering $\prec$ and numbered accordingly. Let basis_∞ be the dependency-induced global basis. Then for all input assignments H the set $\mathcal{B} = \mathsf{basis}_\infty(G(H))$ is a finite stable global basis.*

Proof. Function basis_∞ is a basis for $\mathcal{T}_\infty$ according to Def. 4, as it inherits the properties of local bases. Thus, $\mathcal{B}$ is finite, as $G(H)$ is finite. Also, $\mathcal{B}$ is stable by Lemma 7. □

7 CDSAT is Complete in the Predicate-Sharing Case

CDSAT is *sound* if the theory modules are [9, Thm. 1]. CDSAT is *terminating* if there exists a *finite* and *closed* global basis $\mathcal{B}$, such that the input assignment is in $\mathcal{B}$ [9, Thm. 2]. Recall that an assignment H *is in* $\mathcal{B}$ if $G(H) \subseteq \mathcal{B}$, where $G(H)$ is the set of all terms occurring in H (cf. Sect. 4.1, 4.3). The extension from disjoint to predicate-sharing unions does not affect soundness and termination, whereas the completeness result needs to be generalized. In this section we show that CDSAT is *complete* for a predicate-sharing union $\mathcal{T}_\infty$ of theories $\mathcal{T}_1, \dots, \mathcal{T}_n$, where $\mathcal{T}_1$ is the leading theory. We begin with two preliminary definitions.

Definition 20 (Shared terms (Def. 18 [9]))**.** *The set of* shared terms *for an assignment H, denoted $\mathcal{V}_{\mathsf{sh}}(H)$, is the smallest set N closed under the following rules*

$$\frac{(t{\leftarrow}\mathfrak{c}) \in H}{t \in N} \qquad \frac{u, u' \in N,\ t \in \mathsf{fv}_{\Sigma_i}(u) \cap \mathsf{fv}_{\Sigma_j}(u'),\ i{\neq}j}{t \in N} \qquad \frac{u \in N,\ t \in \mathsf{fv}_{\Sigma_k}(u) \setminus \mathcal{V}_\infty}{t \in N}$$

where $1 \leq i, j, k \leq n$; also, $\mathcal{V}^s_{\mathsf{sh}}(H)$ is the set of shared terms of sort s, $s \in S_\infty$.

The inductive rules add shared free variables and foreign terms, with shared foreign terms added by both rules.

Example 16. *Assume that theories* LIA *and* ArrAD *share sorts* I*,* V*, and* L*, all three interpreted as the set $\mathbb{Z}$ of the integers. For $H = \{i{\leftarrow}3,\ i \simeq j,\ \mathsf{len}(a) \simeq n,\ n{\leftarrow}5,\ \mathsf{select}(\mathsf{store}(a, i, v), j) \not\simeq v,\ \mathsf{Adm}(i, n)\}$, the base of the inductive construction of $\mathcal{V}_{\mathsf{sh}}(H)$ is that i, $i \simeq j$, $\mathsf{len}(a) \simeq n$, n, $\mathsf{select}(\mathsf{store}(a, i, v), j) \not\simeq v$ and $\mathsf{Adm}(i, n)$ are shared. The third rule adds $\mathsf{select}(\mathsf{store}(a, i, v), j)$ and $\mathsf{len}(a)$ as they are Σ_{LIA}-foreign terms. The second rule adds i, j, n, and v as they are shared free variables. In contrast, $\mathsf{store}(a, i, v)$ and a are not shared: they are seen only by* ArrAD*. If sort* V *is not shared and it is interpreted by* ArrAD *as something else (e.g., colors), the construction of $\mathcal{V}_{\mathsf{sh}}(H)$ is the same, except that v is no longer shared and it is seen only by* ArrAD*.*

The following definition instantiates the generic $\mathcal{T}$-assignment J and the generic term set N of Definition 9 (leading-theory-compatibility) with the $\mathcal{T}_k$-view $H_{\mathcal{T}_k}$ of a $\mathcal{T}_\infty$-assignment H and the set $\mathcal{V}_{\mathsf{sh}}(H)$ of shared terms.

Definition 21 (Model-describing assignment (Def. 19 [9]))**.** *An assignment H is* model-describing *if there exists a $\mathcal{T}_1^+[\mathcal{V}]$-model $\mathcal{M}_1$ such that $\mathcal{M}_1 \models H_{\mathcal{T}_1}$ (assuming $\mathsf{fv}_{\Sigma_1}(H_{\mathcal{T}_1}) \subseteq \mathcal{V}$), and for all k, $2 \leq k \leq n$, $H_{\mathcal{T}_k}$ is leading-theory-compatible with $\mathcal{T}_k$ sharing $\mathcal{V}_{\mathsf{sh}}(H)$.*

A collection of theory modules $\mathcal{I}_1, \dots, \mathcal{I}_n$ for $\mathcal{T}_1, \dots, \mathcal{T}_n$ is *complete* for their union $\mathcal{T}_\infty$, if module $\mathcal{I}_1$ is complete for $\mathcal{T}_1$, and modules $\mathcal{I}_k$'s, $2 \leq k \leq n$, are leading-theory-complete. This assumption is used in the next theorem, which generalizes Theorem 3 [9] from disjoint to predicate-sharing unions.

Theorem 9. *In a predicate-sharing union of theories equipped with a complete collection of theory modules and a stable global basis $\mathcal{B}$, for all input assignments H in $\mathcal{B}$, whenever a CDSAT derivation from H halts in a state Γ other than* unsat*, Γ is model-describing.*

Proof. The proof is the same as that of [9, Thm. 3], because the CDSAT transition system is unchanged, except for replacing Backjump [9] with LearnBackjump [11], and it was already shown in [11, Sect. 3.3] that this change does not affect completeness (as well as soundness and termination) of CDSAT. □

The next lemma will be used in the following theorem.

Lemma 10 (Lemma 7 [9]). *For all assignments H, $\mathsf{fv}(H) \subseteq \bigcup_{k=1}^{n} \mathsf{fv}_{\Sigma_k}(\mathcal{V}_{\mathsf{sh}}(H))$.*

Furthermore, for all k, $1 \leq k \leq n$, $\mathsf{fv}_{\Sigma_k}(H_{\mathcal{T}_k}) = \mathsf{fv}_{\Sigma_k}(H) \subseteq \mathsf{fv}_{\Sigma_k}(\mathcal{V}_{\mathsf{sh}}(H))$ $(*)$, so that $\mathsf{fv}_{\Sigma_k}(H_{\mathcal{T}_k} \cup \mathcal{V}_{\mathsf{sh}}(H)) = \mathsf{fv}_{\Sigma_k}(\mathcal{V}_{\mathsf{sh}}(H))$ $(**)$.

The core of the completeness proof is to show that a model-describing assignment is globally endorsed, meaning that its global view (i.e., $\mathcal{T}_\infty$-view) is satisfied by a $\mathcal{T}_\infty^+$-model (cf. Sect. 4.1). The next theorem generalizes this result from disjoint [9, Thm. 4] to predicate-sharing unions.

Theorem 11. *In a predicate-sharing union of theories, if an assignment H is model-describing, there exists a $\mathcal{T}_\infty^+[\mathsf{fv}(H)]$-model $\mathcal{M}$ such that $\mathcal{M} \models^G H$.*

Proof. Similar to that of [9, Thm. 4], the proof is structured in eight steps.

1. *Existence of a leading-theory model $\mathcal{M}_1$*: by the hypothesis that H is model-describing, there exists a $\mathcal{T}_1^+[\mathcal{V}_1]$-model $\mathcal{M}'_1$, with $\mathsf{fv}_{\Sigma_1}(H_{\mathcal{T}_1}) \subseteq \mathcal{V}_1$, such that $\mathcal{M}'_1 \models H_{\mathcal{T}_1}$. Since $\mathsf{fv}_{\Sigma_1}(H_{\mathcal{T}_1}) \subseteq \mathsf{fv}_{\Sigma_1}(\mathcal{V}_{\mathsf{sh}}(H))$ by $(*)$, we pick arbitrary elements in the domains of $\mathcal{M}'_1$ to interpret terms in $\mathsf{fv}_{\Sigma_1}(\mathcal{V}_{\mathsf{sh}}(H)) \setminus \mathcal{V}_1$, if any, and we extend $\mathcal{M}'_1$ into a $\mathcal{T}_1^+[\mathsf{fv}_{\Sigma_1}(\mathcal{V}_{\mathsf{sh}}(H))]$-model $\mathcal{M}_1$ such that $\mathcal{M}_1 \models H_{\mathcal{T}_1}$.
2. *Existence of the other $\mathcal{T}_k$-models $\mathcal{M}_k$*: by the hypothesis that H is model-describing, for all k, $2 \leq k \leq n$, there exists a $\mathcal{T}_k^+[\mathcal{V}_k]$-model $\mathcal{M}_k$ with $\mathsf{fv}_{\Sigma_k}(H_{\mathcal{T}_k} \cup \mathcal{V}_{\mathsf{sh}}(H)) \subseteq \mathcal{V}_k$, and hence $\mathsf{fv}_{\Sigma_k}(\mathcal{V}_{\mathsf{sh}}(H)) \subseteq \mathcal{V}_k$ by $(**)$, such that: (i) $\mathcal{M}_k \models H_{\mathcal{T}_k}$; (ii) for all shared predicate symbols $p \in \mathit{ShF}_k \cap \mathit{ShF}_1$, $p\colon (s_1 \times \cdots \times s_m) \to \mathsf{prop}$, and for all terms $u_1, \ldots, u_m \in \mathcal{V}_{\mathsf{sh}}(H)$ of sorts $s_1, \ldots, s_m$, $\mathcal{M}_k(p(u_1, \ldots, u_m)) = \mathcal{M}_1(p(u_1, \ldots, u_m))$; and (iii) for all sorts $s \in S_k$, there exists a bijection $f_k^s\colon s^{\mathcal{M}_k} \to s^{\mathcal{M}_1}$, such that f_k^{prop} is identity, and for all $p \in \mathit{ShF}_k \cap \mathit{ShF}_1$, $p\colon (s_1 \times \cdots \times s_m) \to \mathsf{prop}$, and for all inhabitants $v_1, \ldots, v_m$ of $s_1^{\mathcal{M}_k}, \ldots, s_m^{\mathcal{M}_k}$, $p^{\mathcal{M}_k}(v_1, \ldots, v_m) = p^{\mathcal{M}_1}(f_k^{s_1}(v_1), \ldots, f_k^{s_m}(v_m))$.
3. *Bijection between any $\mathcal{M}_k$ and $\mathcal{M}_1$*: for all k, $1 \leq k \leq n$, we construct a sort-indexed collection $(\phi_k^s)_{s \in S_k}$ of bijections $\phi_k^s\colon s^{\mathcal{M}_k} \to s^{\mathcal{M}_1}$, such that ϕ_1^s is identity for all sorts, and for all k, $2 \leq k \leq n$, the $(\phi_k^s)_{s \in S_k}$ collection satisfies the same properties as the $(f_k^s)_{s \in S_k}$ collection in Step (2), but also satisfies the additional property $\phi_k^s(\mathcal{M}_k(t)) = \mathcal{M}_1(t)$ for all shared terms $t \in \mathcal{V}_{\mathsf{sh}}^s(H)$ and all sorts $s \in S_k$.

   For a collection $(f_k^s)_{s \in S_k}$ of bijections as in (iii) in Step (2), let $\Psi(f_k^s)_{s \in S_k}$ be the (finite) number of terms t in $\mathcal{V}_{\mathsf{sh}}(H)$ such that $f_k^s(\mathcal{M}_k(t)) \neq \mathcal{M}_1(t)$ where s is the sort of t. We aim at producing a collection $(\phi_k^s)_{s \in S_k}$ such that $\Psi(\phi_k^s)_{s \in S_k} = 0$.

   We give a transformer Φ such that $\Psi(f_k^s)_{s \in S_k} > 0$ implies $\Psi(\Phi(f_k^s)_{s \in S_k}) < \Psi(f_k^s)_{s \in S_k}$. Assume $f_k^s(\mathcal{M}_k(t)) \neq \mathcal{M}_1(t)$. Let $v_1 = f_k^s(\mathcal{M}_k(t))$, where $v_1 \in s^{\mathcal{M}_1}$, and let $v_k = (f_k^s)^{-1}(\mathcal{M}_1(t))$, where $v_k \in s^{\mathcal{M}_k}$. Let $\Phi(f_k^s)_{s \in S_k}$ be the collection that differs from $(f_k^s)_{s \in S_k}$ in that f_k^s is replaced by g_k^s defined as follows: $g_k^s(v_k) = v_1$, $g_k^s(\mathcal{M}_k(t)) = \mathcal{M}_1(t)$, and for all other $v \in s^{\mathcal{M}_k}$, $g_k^s(v) = f_k^s(v)$. Hence, $\Psi(\Phi(f_k^s)_{s \in S_k}) < \Psi(f_k^s)_{s \in S_k}$. Also note that $\Phi(f_s^k)_{s \in S_k}$ satisfies (iii) in Step (2) as $(f_k^s)_{s \in S_k}$ does.

   We keep applying Φ to the collection $(f_k^s)_{s \in S_k}$ from (iii) in Step (2), until we obtain a collection $(\phi_k^s)_{s \in S_k}$ that also satisfies the additional property $\phi_k^s(\mathcal{M}_k(t)) = \mathcal{M}_1(t)$ for all shared terms $t \in \mathcal{V}_{\mathsf{sh}}^s(H)$ of sort s.

4. *Construction of a $\mathcal{T}_\infty^+[\mathsf{fv}(H)]$-model $\mathcal{M}$*: first, $\mathcal{M}$ adopts the domains of $\mathcal{M}_1$ and interprets all sorts, shared variables, and shared predicate symbols as $\mathcal{M}_1$ does:
   (a) For all sorts $s \in S_\infty$: $s^{\mathcal{M}} = s^{\mathcal{M}_1}$;
   (b) For all variables $x \in \mathsf{fv}(H)$ such that $x \in \mathcal{V}_{\mathsf{sh}}(H)$: $x^{\mathcal{M}} = x^{\mathcal{M}_1}$.
   (c) For all shared predicate symbols $p \in ShF_1$: $p^{\mathcal{M}} = p^{\mathcal{M}_1}$.
   Second, $\mathcal{M}$ interprets everything else as in the appropriate $\mathcal{M}_k$, using the bijection ϕ_k^s or its inverse $(\phi_k^s)^{-1}$ to reach elements in its domains:
   (d) For a $\mathcal{T}_k^+$-value $\mathfrak{c}$ of sort s: $\mathfrak{c}^{\mathcal{M}} = \phi_k^s(\mathfrak{c}^{\mathcal{M}_k})$;
   (e) For all variables $x \in \mathsf{fv}^s(H)$ such that $x \notin \mathcal{V}_{\mathsf{sh}}(H)$: $x^{\mathcal{M}} = \phi_k^s(x^{\mathcal{M}_k})$, for the unique k, $1 \le k \le n$, for which $x \in \mathsf{fv}_{\Sigma_k}(\mathcal{V}_{\mathsf{sh}}(H))$; such a k exists by Lemma 10, and it is unique, otherwise $x \in \mathcal{V}_{\mathsf{sh}}(H)$ by Definition 20;
   (f) For all non-shared symbols $f\colon (s_1 \times \cdots \times s_m) \to s$ $(m \ge 0)$, $f \in (F_k \setminus ShF_k)$, where k is unique as f is not shared: for all inhabitants $v_1, \ldots, v_m$ of $s_1^{\mathcal{M}}, \ldots, s_m^{\mathcal{M}}$,

$$f^{\mathcal{M}}(v_1, \ldots, v_m) = \phi_k^s(f^{\mathcal{M}_k}((\phi_k^{s_1})^{-1}(v_1), \ldots, (\phi_k^{s_m})^{-1}(v_m))).$$

5. *$\mathcal{M}$ and $\mathcal{M}_k$ agree on terms, if they agree on their shared free Σ_k-variables*: $\forall k$, $1 \le k \le n$, if t is a term of sort $s \in S_k$ such that:

(h1) Its free Σ_k-variables occur in shared terms: $\mathsf{fv}_{\Sigma_k}(t) \subseteq \mathsf{fv}_{\Sigma_k}(\mathcal{V}_{\mathsf{sh}}(H))$, and
(h2) $\mathcal{M}$ and $\mathcal{M}_k$ agree on the shared free Σ_k-variables of t: for all sorts $r \in S_k$ and all terms $u \in \mathsf{fv}_{\Sigma_k}^r(t) \cap \mathcal{V}_{\mathsf{sh}}^r(H)$ it holds that $\mathcal{M}(u) = \phi_k^r(\mathcal{M}_k(u))$,

then $\mathcal{M}(t) = \phi_k^s(\mathcal{M}_k(t))$. The proof is by structural induction.

- If $t \in \mathsf{fv}_{\Sigma_k}(t) \cap \mathcal{V}_{\mathsf{sh}}(H)$ (t is a shared free Σ_k-variable), the claim holds by (h2).
- If $t \in \mathsf{fv}_{\Sigma_k}(t) \setminus \mathcal{V}_{\mathsf{sh}}(H)$ (t is a non-shared free Σ_k-variable), then by (h1) we get $t \in \mathsf{fv}_{\Sigma_k}(\mathcal{V}_{\mathsf{sh}}(H))$, and $t \in \mathcal{V}_\infty$ must hold, otherwise from $t \in \mathsf{fv}_{\Sigma_k}(\mathcal{V}_{\mathsf{sh}}(H))$, the third rule of Definition 20 would conclude $t \in \mathcal{V}_{\mathsf{sh}}(H)$. Then from $t \in \mathsf{fv}_{\Sigma_k}(\mathcal{V}_{\mathsf{sh}}(H)))$ and $t \in \mathcal{V}_\infty$, we have $t \in \mathsf{fv}(\mathcal{V}_{\mathsf{sh}}(H)) = \mathsf{fv}(H)$, and by Item (e) in the construction of $\mathcal{M}$ we have $\mathcal{M}(t) = \phi_k^s(\mathcal{M}_k(t))$.
- If t is a term $f(t_1, \ldots, t_m)$ with $f\colon (s_1 \times \cdots \times s_m) \to s$ $(m \ge 0)$ and $f \in (F_k \setminus ShF_k)$, for some k, $1 \le k \le n$, then $\mathcal{M}(f(t_1, \ldots, t_m)) = f^{\mathcal{M}}(\mathcal{M}(t_1), \ldots, \mathcal{M}(t_m))$, and by Item (f) in the construction of $\mathcal{M}$ we get

$$\mathcal{M}(f(t_1, \ldots, t_m)) = \phi_k^s(f^{\mathcal{M}_k}((\phi_k^{s_1})^{-1}(\mathcal{M}(t_1)), \ldots, (\phi_k^{s_m})^{-1}(\mathcal{M}(t_m)))).$$

  The induction hypothesis is that $\forall i$, $1 \le i \le m$, $\mathcal{M}(t_i) = \phi_k^{s_i}(\mathcal{M}_k(t_i))$, so that $(\phi_k^{s_i})^{-1}(\mathcal{M}(t_i)) = \mathcal{M}_k(t_i)$. It follows that

$$\mathcal{M}(f(t_1, \ldots, t_m)) = \phi_k^s(f^{\mathcal{M}_k}(\mathcal{M}_k(t_1), \ldots, \mathcal{M}_k(t_m))) = \phi_k^s(\mathcal{M}_k(t)).$$

- If t is a term $p(t_1, \ldots, t_m)$ with $p\colon (s_1 \times \cdots \times s_m) \to \mathsf{prop}$ $(m \ge 0)$ and $p \in ShF_k$, for some k, $1 \le k \le n$, then $p \in ShF_1$, and Item (c) in the construction of $\mathcal{M}$ gives

$$\mathcal{M}(p(t_1, \ldots, t_m)) = p^{\mathcal{M}_1}(\mathcal{M}(t_1), \ldots, \mathcal{M}(t_m)).$$

By the induction hypothesis whereby $\forall i,\ 1 \le i \le m$, $\mathcal{M}(t_i) = \phi_k^{s_i}(\mathcal{M}_k(t_i))$, we get

$$\mathcal{M}(p(t_1,\ldots,t_m)) = p^{\mathcal{M}_1}(\phi_k^{s_1}(\mathcal{M}_k(t_1)),\ldots,\phi_k^{s_m}(\mathcal{M}_k(t_m))).$$

Then, since $\mathcal{M}_k(t_1),\ldots,\mathcal{M}_k(t_m)$ are inhabitants of $s_1^{\mathcal{M}_k},\ldots,s_m^{\mathcal{M}_k}$, and the $(\phi_k^s)_{s\in S_k}$ collection of Step (3) satisfies (iii) in Step (2) we get the first equality in

$$\mathcal{M}(p(t_1,\ldots,t_m)) = p^{\mathcal{M}_k}(\mathcal{M}_k(t_1),\ldots,\mathcal{M}_k(t_m)) = \mathcal{M}_k(t) = \phi_k^{\mathsf{prop}}(\mathcal{M}_k(t))$$

where the last equality holds because ϕ_k^{prop} is identity for all $k,\ 1 \le k \le n$.

6. *$\mathcal{M}$ and $\mathcal{M}_1$ agree on shared terms*: for all $t \in \mathcal{V}_{\mathsf{sh}}(H)$ it holds that $\mathcal{M}(t) = \mathcal{M}_1(t)$. The proof is by structural induction. If $t \in \mathcal{V}_\infty$ then by Item (b) in the construction of $\mathcal{M}$ we have $\mathcal{M}(t) = \mathcal{M}_1(t)$. For the induction step, assume t is a term such that $top(t) \in F_k$, for some $k,\ 1 \le k \le n$, with arity m ($m \ge 0$) and output sort s. We prove that t satisfies (h1) and (h2). (h1) follows from $t \in \mathcal{V}_{\mathsf{sh}}(H)$. For (h2), consider a $u \in \mathsf{fv}^r_{\Sigma_k}(t) \cap \mathcal{V}^r_{\mathsf{sh}}(H)$ for any $r \in S_k$. Since $f \in F_k$, it is $u \lhd t$. Since $u \lhd t$, by induction hypothesis $\mathcal{M}(u) = \mathcal{M}_1(u)$. From $u \in \mathcal{V}^r_{\mathsf{sh}}(H)$ it follows by the additional property of the $(\phi_k^s)_{s\in S_k}$ collection of Step (3) that $\mathcal{M}_1(u) = \phi_k^r(\mathcal{M}_k(u))$. By transitivity, $\mathcal{M}(u) = \phi_k^r(\mathcal{M}_k(u))$, so that also (h2) is satisfied. Then, by Step (5) of this proof, $\mathcal{M}(t) = \phi_k^s(\mathcal{M}_k(t))$. Since $t \in \mathcal{V}^s_{\mathsf{sh}}(H)$, by the additional property of the $(\phi_k^s)_{s\in S_k}$ collection in Step (3) we have $\mathcal{M}_1(t) = \phi_k^s(\mathcal{M}_k(t))$, hence $\mathcal{M}(t) = \mathcal{M}_1(t)$ by transitivity.
7. *$\mathcal{M}$ and $\mathcal{M}_k$ agree on terms in extended signatures*: $\forall k,\ 1 \le k \le n$, for all $\Sigma_k^+[\mathsf{fv}_{\Sigma_k}(\mathcal{V}_{\mathsf{sh}}(H))]$-terms t of sort $s \in S_k$, it holds that $\mathcal{M}(t) = \phi_k^s(\mathcal{M}_k(t))$. As before, the proof is by structural induction.
   - If t is a free Σ_k^+-variable in $\mathsf{fv}_{\Sigma_k}(\mathcal{V}_{\mathsf{sh}}(H))$, the result follows from Step (5) of this proof provided (h1) and (h2) hold for t. (h1) follows from $\mathsf{fv}_{\Sigma_k}(t) = \{t\} \subseteq \mathsf{fv}_{\Sigma_k}(\mathcal{V}_{\mathsf{sh}}(H))$. For (h2), by applying Step (6) of this proof to any term $u \in \mathsf{fv}^r_{\Sigma_k}(t) \cap \mathcal{V}^r_{\mathsf{sh}}(H)$ with $r \in S_k$, we get $\mathcal{M}(u) = \mathcal{M}_1(u)$. Then, by applying the additional property of the $(\phi_k^s)_{s\in S_k}$ collection in Step (3), we get $\mathcal{M}_1(u) = \phi_k^r(\mathcal{M}_k(u))$, so that by transitivity $\mathcal{M}(u) = \phi_k^r(\mathcal{M}_k(u))$.
   - If t is a $\mathcal{T}_k^+$-value, then by Item (d) in the construction of $\mathcal{M}$, we have $\mathcal{M}(t) = \phi_k^s(\mathcal{M}_k(t))$.
   - If t is a term $f(t_1,\ldots,t_m)$ with $f\colon (s_1\times\cdots\times s_m)\to s$ ($m \ge 0$) and $f \in (F_k \setminus ShF_k)$, for some $k,\ 1 \le k \le n$, the proof is identical to that of this case in Step (5).
   - If t is a term $p(t_1,\ldots,t_m)$ with $p\colon (s_1\times\cdots\times s_m)\to\mathsf{prop}$ ($m \ge 0$) and $p \in ShF_k$, for some $k,\ 1 \le k \le n$, the proof is identical to that of this case in Step (5).
8. *Global endorsement*: we show that $\mathcal{M} \models^G H$ by showing that for all $(u \leftarrow \mathfrak{c}) \in H_{\mathcal{T}_\infty}$ it holds that $\mathcal{M}(u) = \mathfrak{c}^{\mathcal{M}}$. For all $(u \leftarrow \mathfrak{c}) \in H_{\mathcal{T}_\infty}$, either $(u \leftarrow \mathfrak{c}) \in H$, or u is a gleaned equality $t_1 \simeq_s t_2$ and $\mathfrak{c}$ is a Boolean value $\mathfrak{b}$. If $(u \leftarrow \mathfrak{c}) \in H$, then $\mathfrak{c}$ is a $\mathcal{T}_k^+$-value for some $k,\ 1 \le k \le n$, and $(u \leftarrow \mathfrak{c}) \in H_{\mathcal{T}_k}$: the assigned value determines to which $\mathcal{T}_k$-view $u \leftarrow \mathfrak{c}$ belongs. If u is an equality $t_1 \simeq_s t_2$, the sort s of the equality determines to which $\mathcal{T}_k$-view $u \leftarrow \mathfrak{b}$ belongs. Sort s must belong to at least one of the signatures: say that $s \in S_k$ for some $k,\ 1 \le k \le n$; then $(u \leftarrow \mathfrak{b}) \in H_{\mathcal{T}_k}$. Either

way, by Step (1) of this proof, if $k = 1$, or by Step (2) of this proof, if $2 \le k \le n$, we have $\mathcal{M}_k \models H_{\mathcal{T}_k}$, that is, (i) $\mathcal{M}_k(u) = \mathfrak{c}^{\mathcal{M}_k}$. Let r be the sort of term u. By Step (7) of this proof, we have (ii) $\mathcal{M}(u) = \phi_k^r(\mathcal{M}_k(u))$. By Item (d) of Step (4) of this proof, it is (iii) $\phi_k^r(\mathfrak{c}^{\mathcal{M}_k}) = \mathfrak{c}^{\mathcal{M}}$. Thus, by chaining (ii), (i), and (iii), one gets $\mathcal{M}(u) = \phi_k^r(\mathcal{M}_k(u)) = \phi_k^r(\mathfrak{c}^{\mathcal{M}_k}) = \mathfrak{c}^{\mathcal{M}}$, which means that $\mathcal{M}$ endorses the assignment.

□

Theorems 9 and 11 directly entail the completeness of CDSAT for predicate-sharing unions, which subsumes the completeness property for disjoint unions [9, Thm. 5].

Theorem 12 (Completeness)**.** *In a predicate-sharing union of theories equipped with a complete collection of theory modules and a stable global basis $\mathcal{B}$, for all input assignments H in $\mathcal{B}$, whenever a CDSAT derivation from H halts in a state Γ other than* unsat*, there exists a $\mathcal{T}_\infty^+[\mathsf{fv}(\Gamma)]$-model $\mathcal{M}$ such that $\mathcal{M} \models^G \Gamma$ and hence $\mathcal{M} \models^G H$ (input assignments never quit the trail).*

8 Comparison with Related Work

Given the breadth of topics, this section is organized in subsections.

8.1 Array Property Fragments and Theories of Sequences

The problem of modeling finite integer-indexed arrays was approached in three ways: (1) using quantifiers [14, 15, 23], (2) using sequences [1, 2, 43], and (3) using quantifiers and sequences [48].

The *array property fragment* (APF) [14, 15, 23] allows one to write *array property* formulas with guarded universal quantification of index variables. An example is the bounded equality covered in Ex. 8. APF is decidable, because it suffices to instantiate the universally quantified variables with terms from a finite set to get a quantifier-free problem in the disjoint union of the theories of arrays, values, and indices (e.g., LIA for integer indices). APF does not add a *length* function and does not allow *index shifting* (i.e., formulas with terms of the form $a[i]$ and $a[i+n]$, for a an integer-indexed array, i a universally quantified index variable, and n a constant), so that the definition of *concatenation* is not included in the fragment. Dropping this restriction causes undecidability [15, 48], except in some cases [23], identified as *tangle-free formulas* [48].

Subsequently, theories of *finite integer-indexed sequences* were proposed for modeling finite integer-indexed arrays [1, 2, 43]. A theory of sequences developed on top of prior work on *strings* of characters pre-existed [5]. Sequences are a generalization of strings: strings are sequences of elements from a finite alphabet, whereas the sort of the elements of sequences is generic and can be countably infinite [29]. Thus, arrays and sequences are more flexible than strings especially in theory combination, where the sort of elements may be shared and interpreted as countably infinite. The basic symbols for the theory of sequences [5] are a constant symbol for the *empty sequence* and a binary *concatenation* operation. Since concatenation is associative and the empty sequence is its identity, sequences form a monoid. The signature also includes a unary constructor that wraps any element into a singleton sequence, an *extract* or *slice* function that extracts from a given sequence the contiguous subsequence between two integer positions, an *access* function that corresponds to select, and a *length* function from sequences to integers, whose application is denoted $|x|$ for x a sequence.

The theory of sequences was revisited in order to model finite integer-index arrays in [43], where finite integer-index arrays are called *vectors*, and in [1, 2]. Positions are renamed indices, and the indices of a sequence x form the $[0, |x|)$ [43] or the $[n, n+|x|)$ [1, 2] interval. For this reason, the sequences of [1, 2] are called n-sequences. The sort of elements remains generic, provided it is countably infinite [43]. A function *update* corresponding to store is added. The binary concatenation operator is made varyadic [43]. The choice of not imposing that the first index is 0 leads to adding a *relocate* function [1, 2]. The axiomatization of sequences is extended with two axioms [43]. The first one is the instance of extensionality axiom (10) (see Sect. 3.1) in the special case where the admissible indices for x are those in the $[0, |x|)$ interval. The second one characterizes the effects of an update: the length does not change (cf. axiom (9)), and only an update at an index in the $[0, |x|)$ interval modifies the element (cf. axiom (8)).

Two sound inference systems for sequences, one based on string reasoning [4, 31], and one based on array reasoning, were defined [43], extended to n-sequences that can be relocated [1, 2], and applied to array benchmarks [1, 2, 43]. Termination and completeness are not guaranteed, as the decidability of the quantifier-free fragment of these theories of sequences is not known. However, if the derivation terminates with a "satisfiable" answer, a model can be extracted from the produced saturated set [43].

A hybrid approach [48] defines the *APF with Concatenation* (APFC) of a theory of integer-indexed arrays interpreted as finite integer-indexed sequences. The result resembles more sequences than arrays. For example, there is neither store nor *update*. The signature features a *repeat* function that produces the sequence e^n for element e and length n, and the effect of an update is obtained by concatenating a slice, e^1, and another slice. APFC is more expressive than APF, because APFC allows *index shifting*, so that *concatenation* can be defined. It follows that APFC is undecidable, but a decision procedure detects whether the input is *tangle-free*, and if yes, decides its validity, by finite instantiation and reduction to a base theory [48]. Tangle-freeness guarantees that finitely many instances suffice for the universally quantified variables [23, 48].

Sequences are interesting in their own right, but the notion that sequences would be naturally finite, whereas arrays would be necessarily infinite, is counterintuitive. Our approach uses neither quantifiers nor sequences, and it preserves decidability. We extended the theory of arrays with extensionality to the theory ArrAD of arrays with abstract domain, where arrays can be *finite*, regardless of the interpretation of the sort of indices. For us, vectors are *dynamic arrays*: the theory VecAD of vectors with abstract domain is obtained by modifying ArrAD so as to axiomatize precisely the dynamic nature of vectors. The quantifier-free fragments of the theories ArrAD, MapAD, and VecAD introduced in this article are decidable, as a consequence of our results: we equipped these theories with CDSAT theory modules, we proved that they are leading-theory complete, and we generalized the termination and completeness of CDSAT to predicate-sharing unions.

8.2 Other Theories of Arrays

Two enrichments of the theory of arrays with extensionality were considered for the purpose of quantifier-free interpolation [25, 27]. The theory of *arrays with MaxDiff* [25], denoted $\mathcal{ARD}(T_I)$, is parametrized with respect to a theory T_I of indices, which is required to extend the theory of *linear orderings* with a 0 element. LIA, LRA (linear rational arithmetic), and the theory IDL of *integer difference logic* (i.e., the theory with 0, successor, predecessor, and the ordering), satisfy this requirement. The signature of $\mathcal{ARD}(T_I)$ features a symbol $\perp$ for the undefined value and a symbol ϵ for the array whose value is $\perp$ at all indices. The axioms impose that an array has value $\perp$ at all indices smaller than 0, and that $\mathsf{diff}(a, b)$ is the largest index where a and b differ and 0 otherwise. The signature does not include a length function, and the length of an array a is captured indirectly as $\mathsf{diff}(a, \epsilon)$.

The theory $\mathcal{CARD}(T_I)$ of *contiguous arrays with length and MaxDiff* [27] drops the ϵ symbol and adds a length function, denoted $|y|$ for array y. Arrays are required to be *contiguous*, meaning that array y has a value other than $\perp$ at all indices in the interval $[0, |y|]$ and has value $\perp$ everywhere else. Similar to $\mathcal{ARD}(T_I)$, the length of an array is the largest index where the value is not $\perp$. Thus, array y has $|y| + 1$ values other than $\perp$, whereas in programming languages (cf. Ex. 3) array y has $|y|$ values.

A similarity between $\mathcal{ARD}(T_I)$ and $\mathcal{CARD}(T_I)$ on one hand and ArrAD on the other is that in all three theories an out-of-bounds store leaves the array unchanged. While both $\mathcal{ARD}(T_I)$ and $\mathcal{CARD}(T_I)$ share 0 and the linear ordering with the theory T_I of indices, nondisjointness is not an issue in [25, 27], because the objective is quantifier-free interpolation and the problem is not viewed as theory combination.

Our approach is general, because it abandons the assumption that the indices of an array form an interval in a linearly ordered set. We make no special assumptions on the interpretation of the sorts of indices, values, and lengths. The flexibility and expressivity of our data structure theories are made possible by the abstract notion of *admissibility* of indices, and the choice of viewing the problem as a nondisjoint theory combination, where the admissibility predicate is defined by another theory. While ArrAD only needs to share the admissibility predicate with another theory, theory VecAD also shares an ordering on indices, but this ordering does not have to be linear, as in $\mathcal{ARD}(T_I)$, $\mathcal{CARD}(T_I)$, and all theories of integer-indexed data structures.

8.3 Theories of Maps

In [15] the theory of maps is the same as the theory of arrays without the assumption that indices are integers. However, as a consequence, bounded equality is not defined for maps. Also, the maps of [15] are renamed arrays in [14, Sect. 11.1].

A theory of *finite maps* was presented towards building a library in the HOL theorem prover [17]. The signature comprises *apply* and *update* symbols, corresponding to select and store, a constant symbol *empty* for the empty map, and a *Domain* predicate, which corresponds to our Adm in the sense that for all maps f and indices x, $\mathit{Domain}(f, x)$ iff $\mathsf{Adm}(x, \mathsf{len}(f))$. A length function is not included. The axiomatization has the select-over-store axioms (1-2), an axiom corresponding to axiom (13), an axiom saying that two consecutive stores at distinct indices commute, and an axiom saying that the effect of two consecutive stores at the same index is the effect of the second (i.e., outermost) store. The latter two axioms are theorems of the Arr theory as shown in [3]. The axiomatization in [17] does not contain an extensionality axiom, but it includes a second-order induction principle, which allows one to derive an extensionality theorem corresponding to axiom (10). The second-order induction principle characterizes these HOL maps as *finite*, because every map is the result of a finite number of updates to the empty map.

Theories ArrAD, MapAD, and VecAD do not impose finiteness: the data structure can be infinite, but with a finite set of admissible indices, and hence a finite length. The length of a structure is not restricted to be finite either: since the length is viewed as what defines the set of admissible indices, an infinite structure can have infinite set of admissible indices and infinite length. The axiomatizations of ArrAD, MapAD, and VecAD are first-order and suitable for SMT approaches such as CDSAT.

8.4 Nondisjoint Theories and Bridging Functions

The archetype of combination schemes is the equality-sharing or Nelson-Oppen scheme [35, 36] (see [14, Ch. 10], [13, Sect. 3], and [6, Sect. 3] for recent presentations), which requires the component theories $\mathcal{T}_k$ $(1 \leq k \leq n)$ to be *disjoint* and *stably infinite* (every $\mathcal{T}_k$-satisfiable quantifier-free formula has a $\mathcal{T}_k$-model where all sorts except prop are interpreted as countably infinite sets). The *combination framework* [24, 26] identifies two properties, named *noetherianity* and *compatibility*, that are more general than disjointness and stable infiniteness, respectively, and are sufficient for termination and completeness. Noetherianity says that given a finite set of variables there is no infinite ascending chain of atoms (Boolean terms) made of shared symbols, modulo logical consequence in the theory. Thus, only finitely many atoms made of shared symbols can be exchanged among the theories in the union.

In the context of the superposition-based approach [3], ideas from the combination framework were used to allow *convex* theories (whenever a set of $\mathcal{T}$-literals $\mathcal{T}$-entails a disjunction of equalities, one of the equalities is also $\mathcal{T}$-entailed) to share a subtheory of counter arithmetic with 0 and successor [37, 42]. This approach was applied to non-empty *lists* with *length*, *records* with an *increment* operator to increase integer values of record's attributes, and *binary trees* with a *size* function [37]. In a theory of non-empty lists without length and in a theory of records without increment, the theory of counter arithmetic can be used to count the elements in a list or the number of store operations over a record, that are alternative ways to compute the length [42].

The disjointness requirement was lifted for combination schemes in the case of *lists* and *trees* with bridging functions, or, in general, *recursive* (or *absolutely free*) data structures (AFDS) with bridging functions [16]. These theories are also convex [16]. Arrays are not AFDS, and no theory of arrays is convex, not even Arr_0 (cf. Sect. 3).

To the best of our knowledge none of the previous work on lifting disjointness applies to arrays. We lifted the disjointness requirement for CDSAT by generalizing relevance (cf. Def. 5), leading theory compatibility (cf. Def. 9), global basis construction (see Sect. 6), and completeness (see Sect. 6) to predicate-sharing theories. This generalization applies to theories ArrAD, MapAD, and VecAD.

9 Discussion

The theory of *arrays* is of fundamental importance for reasoning about programs, and hence in satisfiability modulo theories (SMT). In most approaches array indices are interpreted as the (non-negative) integers. This choice, together with the absence of a *length* function in the theory signature, means that arrays are regarded as being infinite. Another reason for the view of arrays as infinite structures in SMT is that combination schemes require the component theories to be stably infinite. A third difficulty is that length is a *bridging function*, as the extensionality axiom for arrays with length mixes symbols from the theory of arrays and the theory of indices (e.g., integer arithmetic). The resulting combination of theories is not disjoint, whereas most SMT methods require the component theories to be disjoint. In summary, there is a hyatus between the typical treatment of arrays in SMT and the desiderata from program verification, because in programming languages arrays are *finite* data structures whose *length* is defined. In this article we solved all these intertwined issues by providing:

1. A new theory ArrAD of *arrays with abstract domain*, equipped with an abstract notion of *admissibility*, that allows ArrAD to model arrays as they are in programming languages, while sharing only the admissibility predicate with another theory that provides its definition;
2. Variants of ArrAD that present *maps with abstract domain* (MapAD) and *vectors with abstract domain* (VecAD), modeling for the first time vectors as *dynamic arrays*, which is how they are conceived in programming languages;
3. An extension of the CDSAT combination method from disjoint to *predicate-sharing* theories;
4. CDSAT *modules* for arrays, maps, and vectors with abstract domain that meet the requirements for *termination* and *completeness* of CDSAT;
5. Generalizations of the *global basis construction* for termination and of the CDSAT *completeness theorem* from the disjoint to the *predicate-sharing* case.

There are many avenues for future work. The data structures of this article may be implemented in the Yices 2 state-of-the-art SMT solver.[5] Other theories and bridging functions may be considered, defining appropriate shared predicates and CDSAT modules. We may investigate the integration of CDSAT within the QSMA algorithm for *quantified satisfiability* modulo a complete theory and an assignment [12], by having CDSAT as the quantifier-free solver underlying QSMA. Such an integration would simultaneously endow CDSAT with quantifier reasoning and extend the applicability of QSMA beyond a single complete theory.

In order to further augment expressivity, we may seek to add a *concatenation* operator to the theories considered in this article, so as to subsume the theory of sequences. In general, concatenation may jeopardize decidability. A source of difficulty is that concatenation is associative, but not commutative. For example, the uniform word problem for semigroups, namely the problem of deciding $\{s_i \simeq t_i\}_{i=1}^{n} \models s_{n+1} \simeq t_{n+1}$, where $\forall i$, $1 \le i \le n+1$, s_i and t_i are terms made of constants and a binary associative concatenation operator, is undecidable [20] (which refers to [46] and [18, Page 292]).

[5]See https://yices.csl.sri.com/ for Yices 2.

The theory of string (or word) equations (i.e., sentences made of string equations and logical connectives) is undecidable [21] (which refers to [41]). On the other hand, there exist decision procedures for the solvability of quantifier-free equations in a free monoid, that is, word equations [32, 39, 40]. The decidability of the theory of strings with concatenation and length is open [21]. The decidability of the theory of sequences with concatenation and length is Turing-equivalent to that of strings [29].

If the sort of elements is interpreted as triples of integers, it is possible to write in APFC a *quantified* formula that encodes the halting problem of a two-register machine [48, Remark 1], which is undecidable [34, Thm. 14.1-1]. The choice of the two-register machine is not essential, as it would be possible to encode the halting problem of a more complex machine by using longer tuples of integers. Concatenation does not appear in the formula that encodes the halting problem, but both the formula defining concatenation and the formula capturing the halting problem are *entangled*, that is, they are not tangle-free [48].

To the best of our knowledge, the decidability of the *quantifier-free* fragment of a theory of arrays with both length and concatenation is open. In future work, we may study what happens if theories ArrAD, MapAD, and VecAD are enriched with concatenation, and investigate what CDSAT may offer in this case.

Acknowledgements. This work was started while the first author was visiting the Computer Science Laboratory of SRI International, whose support is greatly appreciated. This research was supported in part by an Amazon Research Award (Fall 2022 Call) to the first author and by NSF grants 1816936, 1817204, and 2016597 to the second and third authors. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of Amazon or NSF. The first author thanks EU COST Action CA20111 for support in disseminating some of the results in this article.

References

[1] Ait-El-Hara, H.: Theory of sequences tailored for program verification. PhD thesis, Université Paris-Saclay (October 2025)

[2] Ait-El-Hara, H., Bobot, F., Bury, G.: Reasoning over n-indexed theories of sequences in SMT. Acta Informatica **62.3**, 1–33 (2025) https://doi.org/10.1007/s00236-025-00496-w

[3] Armando, A., Bonacina, M.P., Ranise, S., Schulz, S.: New results on rewrite-based satisfiability procedures. ACM Trans. Comput. Log. **10**(1), 129–179 (2009) https://doi.org/10.1145/1459010.1459014

[4] Berzish, M., Ganesh, V., Zheng, Y.: Z3str3: a string solver with theory-aware heuristics. In: Stewart, D. (ed.) Proc. FMCAD-17, pp. 55–59. FMCAD Inc., Austin (2017). https://doi.org/10.5555/3168451.3168468

[5] Bjørner, N., Ganesh, V., Michel, R., Veanes, M.: SMT-LIB sequences and regular expressions. In: Fontaine, P., Goel, A. (eds.) Proc. of SMT-10. EPiC

Series in Computing, vol. 20, pp. 77–87. EasyChair, Manchester (2012). https://doi.org/10.29007/w5m5

[6] Bonacina, M.P.: The CDSAT method for satisfiability modulo theories and assignments: an exposition. In: Beckmann, A., Oitavem, I., Manea, F. (eds.) Proc. of CiE-21. LNCS, vol. 15764, pp. 1–16. Springer, Cham (2025). https://doi.org/10.1007/978-3-031-95908-0_1

[7] Bonacina, M.P., Echenim, M.: On variable-inactivity and polynomial $\mathcal{T}$-satisfiability procedures. J. Log. Comput. **18**(1), 77–96 (2008) https://doi.org/10.1093/logcom/exm055

[8] Bonacina, M.P., Echenim, M.: Theory decision by decomposition. J. Symb. Comput. **45**(2), 229–260 (2010) https://doi.org/10.1016/j.jsc.2008.10.008

[9] Bonacina, M.P., Graham-Lengrand, S., Shankar, N.: Conflict-driven satisfiability for theory combination: transition system and completeness. J. Autom. Reason. **64**(3), 579–609 (2020) https://doi.org/10.1007/s10817-018-09510-y

[10] Bonacina, M.P., Graham-Lengrand, S., Shankar, N.: CDSAT for nondisjoint theories with shared predicates: arrays with abstract length. In: Hyvärinen, A., Déharbe, D. (eds.) Proc. SMT-20. CEUR Proceedings, vol. 3185, pp. 18–37. CEUR WS-org, Aachen (2022). https://ceur-ws.org/Vol-3185/

[11] Bonacina, M.P., Graham-Lengrand, S., Shankar, N.: Conflict-driven satisfiability for theory combination: lemmas, modules, and proofs. J. Autom. Reason. **66**(1), 43–91 (2022) https://doi.org/10.1007/s10817-021-09606-y

[12] Bonacina, M.P., Graham-Lengrand, S., Vauthier, C.: The QSMA algorithm for quantifiers in SMT. J. Autom. Reason. **69**(2), 1–40 (2025) https://doi.org/10.1007/s10817-025-09727-8 . Article n. 13

[13] Bonacina, M.P., Fontaine, P., Ringeissen, C., Tinelli, C.: Theory combination: beyond equality sharing. In: Lutz, C., Sattler, U., Tinelli, C., Turhan, A.-Y. (eds.) Description Logic, Theory Combination, and All That: Essays Dedicated to Franz Baader. LNCS, vol. 11560, pp. 57–89. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-22102-7_3

[14] Bradley, A.R., Manna, Z.: The Calculus of Computation - Decision Procedures with Applications to Verification. Springer, Berlin (2007)

[15] Bradley, A.R., Manna, Z., Sipma, H.B.: What's decidable about arrays? In: Emerson, E.A., Namjoshi, K.S. (eds.) Proc. of VMCAI-7. LNCS, vol. 3855, pp. 427–442. Springer, Heidelberg (2006). https://doi.org/10.1007/11609773_28

[16] Chocron, P., Fontaine, P., Ringeissen, C.: Politeness and combination methods for theories with bridging functions. J. Autom. Reason. **64**, 97–134 (2020)

https://doi.org/10.1007/s10817-019-09512-4

[17] Collins, G., Syme, D.: A theory of finite maps. In: Schubert, T.E., Windley, P.J., Alves-Foss, J. (eds.) Proc. of TPHOLs. LNCS, vol. 971, pp. 122–137. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-60275-5_61

[18] Davis, M.: The Undecidable. Raven Press, New York (1965)

[19] de Moura, L., Passmore, G.O.: Exact global optimization on demand (presentation only). In: Ghilardi, S., Sofronie-Stokkermans, V., Tiwari, A. (eds.) Proc. of ADDCT-3, pp. 50–50 (2013)

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

[21] Ganesh, V., Minnes, M., Solar-Lezama, A., Rinard, M.: Word equations with length constraints: what's decidable? In: Biere, A., Nahir, A., Vos, T. (eds.) Proc. of HVC-8. LNCS, vol. 7857, pp. 209–226. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-39611-3_21

[22] Ganzinger, H., Rueß, H., Shankar, N.: Modularity and refinement in inference systems. Technical Report CSL-SRI-04-02, CSL, SRI International, Menlo Park, CA, USA (2004)

[23] Ge, Y., de Moura, L.: Complete instantiation for quantified formulas in satisfiability modulo theories. In: Proc. CAV-21. LNCS, vol. 5643, pp. 306–320. Springer, Berlin (2009). https://doi.org/10.1007/978-3-642-02658-4_25

[24] Ghilardi, S.: Model-theoretic methods in combined constraint satisfiability. J. Autom. Reason. **33**, 221–249 (2004) https://doi.org/10.1007/s10817-004-6241-5

[25] Ghilardi, S., Gianola, A., Kapur, D.: Interpolation and amalgamation for arrays with MaxDiff. In: Kiefer, S., Tasson, C. (eds.) Proc. of FoSSaCS-24. LNCS, vol. 12650, pp. 268–288. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-71995-1_14

[26] Ghilardi, S., Nicolini, E., Zucchelli, D.: A comprehensive combination framework. ACM Trans. Comput. Log. **9**(2), 1–54 (2008) https://doi.org/10.1145/1342991.1342992

[27] Ghilardi, S., Gianola, A., Kapur, D., Naso, C.: Interpolation results for arrays with length and MaxDiff. ACM Trans. Comput. Log. **24**(4), 28–12833 (2023) https://doi.org/10.1145/3587161

[28] Hyvärinen, A.E.J., Wintersteiger, C.: Parallel satisfiability modulo theories. In: Hamadi, Y., Sais, L. (eds.) Handbook of Parallel Constraint Reasoning, pp. 141–178. Springer, Cham (2018). Chap. 5. https://doi.org/10.1007/978-3-319-63516-3_5

[29] Jeż, A., Lin, A.W., Markgraf, O., Rümmer, P.: Decision procedures for sequence theories. In: Enea, C., Lal, A. (eds.) Proc. CAV-35. LNCS, vol. 13965, pp. 18–40. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-37703-7_2

[30] Jovanović, D., Dutertre, B.: Interpolation and model checking for nonlinear arithmetic. In: Silva, A., Leino, K.R.M. (eds.) Proc. CAV-33. LNCS, vol. 12760, pp. 266–288. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-81688-9_13

[31] Liang, T., Reynolds, A., Tinelli, C., Barrett, C., Deters, M.: A DPLL(T) theory solver for a theory of strings and regular expressions. In: Biere, A., Bloem, R. (eds.) Proc. CAV-26. LNCS, vol. 8559, pp. 646–662. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-319-08867-9_43

[32] Makanin, G.S.: The problem of solvability of equations in a free semigroup. Math. Sbornik **103**, 147–236 (1977). English transl. in Math. USSR Sbornik 32 (1977)

[33] McCarthy, J.W.: Towards a mathematical science of computation. In: Colburn, T.R. (ed.) Program Verification. COGS, vol. 14, pp. 35–56. Springer, Heidelberg (1993). https://doi.org/10.1007/978-94-011-1793-7_2

[34] Minsky, M.L.: Computation: Finite and Infinite Machines. Prentice-Hall, Hoboken (1967)

[35] Nelson, G.: Combining satisfiability procedures by equality sharing. In: Bledsoe, W.W., Loveland, D.W. (eds.) Automatic Theorem Proving: After 25 Years, pp. 201–211. AMS, Providence (1983)

[36] Nelson, G., Oppen, D.C.: Simplification by cooperating decision procedures. ACM Trans. Prog. Lang. Syst. **1**(2), 245–257 (1979) https://doi.org/10.1145/357073.357079

[37] Nicolini, E., Ringeissen, C., Rusinowitch, M.: Combining satisfiability procedures for unions of theories with a shared counting operator. Fundam. Inform. **105**(1-2), 163–187 (2010) https://doi.org/10.3233/FI-2010-362

[38] Pellau, M., Miné, A., Truchet, C., Benhamou, F.: A constraint solver based on abstract domains. In: Giacobazzi, R., Berdine, J., Mastroeni, I. (eds.) Proc. VMCAI-14. LNCS, vol. 7737, pp. 434–454. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-35873-9_26

[39] Plandowski, W.: Satisfiability of word equations with constants is in PSPACE. J. ACM **51**(3), 483–496 (2004) https://doi.org/10.1145/990308.990312

[40] Plandowski, W.: An efficient algorithm for solving word equations. In: Proc. of STOC-38, pp. 467–476. ACM, New York (2006). https://doi.org/10.1145/1132516.1132584

[41] Quine, W.V.O.: Concatenation as a basis for arithmetic. J. Symb. Log. **11**(4), 105–114 (1946)

[42] Ringeissen, C., Senni, V.: Modular termination and combinability for superposition modulo counter arithmetic. In: Tinelli, C., Sofronie-Stokkermans, V. (eds.) Proc. of FroCoS-8. LNAI, vol. 6989, pp. 211–226. Springer, Berlin (2011). https://doi.org/10.1007/978-3-642-24364-6_15

[43] Sheng, Y., Nötzli, A., Reynolds, A., Zohar, Y., Dill, D., Grieskamp, W., Park, J., Qaader, S., Barrett, C., Tinelli, C.: Reasoning about vectors: satisfiability modulo a theory of sequences. J. Autom. Reason. **67**, 32–13232 (2023) https://doi.org/10.1007/s10817-023-09682-2

[44] Sofronie-Stokkermans, V.: Locality results for certain extensions of theories with bridging functions. In: Schmidt, R.A. (ed.) Proc. of CADE-22. LNAI, vol. 5663, pp. 67–83. Springer, Berlin (2009). https://doi.org/10.1007/978-3-642-02959-2_5

[45] Stump, A., Barrett, C., Dill, D.L., Levitt, J.: A decision procedure for an extensional theory of arrays. In: Halpern, J. (ed.) Proc. of LICS-16. IEEE Computer Society Press, Los Alamitos (2001)

[46] Tarski, A., Mostowski, A., Robinson, R.M.: Undecidable Theories. North Holland, Amsterdam (1953)

[47] Toledo, G.V., Przybocki, B., Zohar, Y.: Being polite is not enough (and other limits of theory combination). In: Barrett, C., Waldmann, U. (eds.) Proc. of CADE-30. LNAI, vol. 15943, pp. 17–34. Springer, Cham (2025). https://doi.org/10.1007/978-3-031-99984-0_2

[48] Wang, Q., Appel, A.W.: A solver for arrays with concatenation. J. Autom. Reason. **67**, 4–1431 (2023) https://doi.org/10.1007/s10817-022-09654-y