Interpolation Results for Arrays with Length and MaxDiff

SILVIO GHILARDI, Università degli Studi di Milano, Italy
ALESSANDRO GIANOLA, Free University of Bozen-Bolzano, Italy
DEEPAK KAPUR, University of New Mexico, USA
CHIARA NASO, Università degli Studi di Milano, Italy

In this paper, we enrich McCarthy's theory of extensional arrays with a length and a maxdiff operation. As is well-known, some diff operation (i.e. some kind of difference function showing where two unequal array differ) is needed to keep interpolants quantifier-free in array theories; our maxdiff operation returns the max index where two arrays differ and so it has a univocally determined semantics. The length function is a natural complement of such a maxdiff operation and is needed to handle real arrays.

Obtaining interpolation results for such a rich theory is a surprisingly hard task. We get such results via a thorough semantic analysis of the models of the theory and of their amalgamation and strong amalgamation properties. The results are modular with respect to the index theory and we show how to convert them into concrete interpolation algorithms via a hierarchical approach realizing a polynomial reduction to interpolation in linear arithmetics endowed with free function symbols.

CCS Concepts: • **Theory of computation → Logic and verification**; **Automated reasoning**; **Verification by model checking**.

Additional Key Words and Phrases: SMT, interpolation, arrays, amalgamation

ACM Reference Format:
Silvio Ghilardi, Alessandro Gianola, Deepak Kapur, and Chiara Naso. 2022. Interpolation Results for Arrays with Length and MaxDiff . *ACM Trans. Comput. Logic* 1, 1 (March 2022), 33 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Craig Interpolation Theorem [13] is a well-known result in first logic that, given an entailment between two logical formulae α and β, establishes the existence of a third formula γ that shares its non-logical symbols with both α and β and such that it is entailed by α and entails β. Studying interpolation has a long-standing tradition also in non-classical logics and in algebraic logic. Nevertheless, interpolation has been obtaining an increasing attention in automated reasoning and formal verification: since McMillan's seminal papers [23, 24], interpolation has been successfully applied in software model checking, also in combination with orthogonal techniques like PDR [36] or k-induction [21]. The reason why interpolation techniques are so attractive is because they allow to discover in a completely *automatic* way new atoms (more precisely, existing predicates with new arguments) that might contribute to the construction of invariants. In fact, software model-checking problems are typically infinite state, so invariant synthesis may require introducing formulae whose

Authors' addresses: Silvio Ghilardi, Università degli Studi di Milano, Milan, Italy, silvio.ghilardi@unimi.it; Alessandro Gianola, Free University of Bozen-Bolzano, Bolzano, Italy, gianola@inf.unibz.it; Deepak Kapur, University of New Mexico, Albuquerque, New Mexico, USA, kapur@cs.unm.edu; Chiara Naso, Università degli Studi di Milano, Milan, Italy, chiaran97@gmail.com.

1529-3785/2022/3-ART $15.00
https://doi.org/XXXXXXX.XXXXXXX

search is not finitely bounded. One way to discover them is to analyze spurious error traces; for instance, if the system under examination (described by a transition formula $Tr(\underline{x}, \underline{x}')$) cannot reach in n-step an error configuration in $U(\underline{x})$ starting from an initial configuration in $In(\underline{x})$, this means that the formula

$$In(\underline{x}_0) \wedge Tr(\underline{x}_0, \underline{x}_1) \wedge \cdots \wedge Tr(\underline{x}_{n-1}, \underline{x}_n) \wedge U(\underline{x}_n)$$

is inconsistent (modulo a suitable theory T). From the inconsistency proof, by computing an interpolant, say at the i-th iteration, one can produce a formula $\phi(\underline{x})$ such that, modulo T, we have

$$In(\underline{x}_0) \wedge \bigwedge_{j=0}^{i} Tr(\underline{x}_{j-1}, \underline{x}_j) \models \phi(\underline{x}_i) \text{ and } \phi(\underline{x}_i) \wedge \bigwedge_{j=i+1}^{n} Tr(\underline{x}_{j-1}, \underline{x}_j) \wedge U(\underline{x}_n) \models \bot. \tag{1}$$

This formula (and the atoms it contains) can contribute to the refinement of the current candidate loop invariant guaranteeing safety. This fact can be exploited in many different ways during invariant search, depending on various techniques employed. It should be noticed however that interpolants are not unique and that different interpolation algorithms may return interpolants of different quality: all interpolants restrict search, but not all of them might be conclusive.

Model-checking applications usually require that such computed interpolants are not arbitrary but present specific shapes so as to guarantee their concrete usability. Since in many cases studied in software verification the underlying theories have a decidable quantifier-free fragment (but are undecidable or have prohibitive complexity outside), the most natural choice is to consider *quantifier-free* interpolants. However, even in case α and β are quantifier-free, Craig's Theorem does not guarantee that an interpolant γ is quantifier-free too. Indeed, this property, called 'quantifier-free interpolation', does not hold in general for arbitrary first order theories. It is then a non-trivial (and, very often, challenging) problem to prove that useful theories admit quantifier-free interpolation.

In this paper, we are interested in studying the problem of quantifier-free interpolation for an expressive datatype theory that strictly extends the well-studied *theory of arrays with extensionality*. Such a theory was introduced by McCarthy in [22]: the main operations considered are the *write* operation (i.e. the array update) and the *read* operation (i.e., the access to the content of an array cell). As such, this theory is suitable to formalize programs over arrays, like standard copying, comparing, searching, sorting, etc. functions; verification problems of this kind are collected in the SV-COMP benchmarks category "ReachSafety-Arrays"[1], where safety verification tasks involving arrays of *finite but unknown length* are considered.

By itself, the theory of arrays with extensionality does not have quantifier free interpolation [19].[2] Moreover, although its quantifier-free fragment is decidable, it is well-known that this theory in its full generality is undecidable as shown in [5]; nonetheless, in the same paper, the authors studied a significant decidable fragment, the so-called 'array property fragment', which strictly extends the quantifier-free one. The array property fragment is expressive enough to formalize several benchmarks; however, as proved in [18], it is not closed under interpolation. Thus, a particularly challenging but interesting problem is that of identifying expressive extensions of the quantifier-free fragment of arrays that are still decidable but also enjoy interpolation: this is what we address in this paper.

A first attempt in this direction is in [7], where a variant of McCarthy's theory was introduced by Skolemizing the axioms of extensionality. This variant turned out to enjoy quantifier-free interpolation [7],[35]. However, this Skolem function `diff` is generic because its semantic interpretation is

[1] https://sv-comp.sosy-lab.org/2020/benchmarks.php

[2] This is the counterexample (due to R. Jhala): the formula $x = wr(y, i, e)$ is inconsistent with the formula $rd(x, j) \neq rd(y, j) \wedge rd(x, k) \neq rd(y, k) \wedge j \neq k$, but all possible interpolants require quantifiers to be written

```
int a[N];
int b[N];
int I = 0;
while I < N do
    b[I] = a[I];
    I++;
end
assert(a = b);
```

- $In(a, b, I) \equiv I = 0 \wedge |a| = N - 1 \wedge |b| = N - 1 \wedge N > 0$
- $Tr(a, b, I, a', b', I') \equiv I < N \wedge I' = I + 1 \wedge a' = a \wedge b' = wr(b, I, rd(a, I))$
- $U(a, b) \equiv a \neq b \wedge I = N$

Fig. 1. `Strcpy` function: code and associated transition system (with program counter missed in the latter for simplicity). Loop invariant: $a = b \vee (N > \mathtt{diff}(a, b) \wedge \mathtt{diff}(a, b) \geq I)$.

undetermined. Moreover, all the array theories mentioned so far allow unlimited out-of-bound write operations and so cannot directly express the notion of array *length*, which is fundamental when formalizing the real behavior of programs. Length is essential for the *faithful* logical formalization of benchmarks coming from software verification, such as C programs included in the SV-COMP competition [4].

These are the main reasons why in [16] the theory was further enriched. There, the semantics of `diff`, called maxdiff, is uniquely determined in the models of the theory and is more informative: it returns *the biggest index* where two different arrays differ. The effectiveness of quantifier-free interpolation in the theory of arrays with maxdiff is exemplified in the simple example of Figure 1: the invariant certifying the assert in line 7 of the `Strcpy` algorithm can be obtained taking a suitable quantifier-free interpolant out of the spurious trace (1) already for $n = 2$.

In the theory considered in [16] a weak notion of length, called 'weak length' from now on, is also introduced. The main contribution of [16] is to show that this enriched theory has quantifier-free interpolants and its quantifier-free fragment is decidable. Still, some expressive limitations (shared with the previous literature) persist: arrays are not forced to be completely defined inside their allocation interval (when an array satisfies this property, we call it 'contiguous'), because they might contain undefined values in some location. Hence, the weak length defined there is is not powerful enough to represent the notion of length used in practice and to adequately formalize real arrays occurring in computer programs. Moreover, although in [16] a complete terminating procedure for computing interpolants is provided, a complexity upper bound is given only in the simple basic case where indexes are just linear orders; for more complex arithmetical theories of indexes, no complexity analysis is carried out and the algorithm becomes quite impractical (it does not have even a primitive recursive bound for termination).

This paper is a substantially revised version and extension of the conference paper [16]: we overcome here all the aforementioned limitations. For that purpose, we introduce the very expressive theory of *contiguous arrays with maxdiff* $\mathcal{CARD}(T_I)$ (parameterized over an index theory T_I), which improves and strictly extends the theory presented in [16] *by requiring arrays to be all contiguous.* This makes the theory more adequate to represent arrays used in common programming languages. Moreover, in contrast to [16] where only amalgamation is shown, we prove here a *strong amalgamation* result, when $\mathcal{CARD}(T_I)$ is enriched with 'constant arrays' of a fixed length with a default value in all their locations. Notably, this not only yields that quantifier-free interpolants exist, but also that interpolation is preserved under disjoint signatures combinations and holds in presence of free function symbols (see the definition of 'general interpolation' below). This result is completely novel and particularly challenging to be proven, since it requires a sophisticated model-theoretic machinery and a careful algebraic analysis of the class of all models.

We also radically re-design the interpolation algorithm, avoiding the use of unbounded loops and of impractical full instantiation routines. Our new 3-Steps algorithm from Section 7 reduces the computation of interpolants of a jointly unsatisfiable pair of constraints to a polynomial size instance of the same problem in the underlying index theory enriched with unary function symbols. As such, the new algorithm becomes part of the hierarchical interpolation algorithms family [31] and in particular somewhat resembles the algorithm presented in [35] for array theory enriched with the basic `diff` symbol. We underline that one aspect making our problems technically more challenging than similar problems investigated in the literature is the fact that we handle a combination with *very expressive* index theories: such a combination is *non-disjoint* because the total orderings on indexes enter into the specification of the maxdiff and length axioms for arrays.

1.1 Plan of the paper

In the following, we call $\mathcal{EUF}$ the theory of equality and uninterpreted symbols. We introduce two novel theories in Section 3: $\mathcal{CARD}(T_I)$, i.e., the theory of contiguous arrays with maxdiff, and $\mathcal{CARDC}(T_I)$, which is an extension of $\mathcal{CARD}(T_I)$ also containing 'constant arrays' of a fixed length with a default value (called '*el*') in all locations. The main technical results are that, for every index theory T_I:

(i) $\mathcal{CARDC}(T_I)$ has general quantifier-free interpolation;
(ii) $\mathcal{CARD}(T_I)$ enjoys quantifier-free interpolation and such interpolants can be computed hierarchically by relying on a black-box interpolation algorithm for the weaker theory $T_I \cup \mathcal{EUF}$ (which has quantifier free interpolation because T_I is assumed to be strongly amalgamable, see Theorem 2.4).

Result (i) is proved semantically, i.e., we show the equivalent strong amalgamation property (see Section 2 for the definitions). The semantic proof requires dedicated constructions (Section 5), relying on some important facts about models and their embeddings (Section 4).

The fact that $\mathcal{CARD}(T_I)$ has interpolants follows from the results in Section 5 (where we prove that this theory is amalgamable). Result (ii) is proved last (Section 7); we first need an investigation on the solvability of the $SMT(\mathcal{CARD}(T_I))$ problem (Section 6).

We supply here some intuitions about our interpolation algorithm from Section 7. The algorithm computes an interpolant out of a pair of (suitably preprocessed) mutually unsatisfiable quantifier-free formulæ A^0, B^0. We call *common* the variables occurring in both A^0 and B^0. The existence of quantifier-free interpolants intuitively means that there are two reasoners, one for A^0 and one for B^0, the first (the second, resp.) of which operates on formulae involving only variables from A_0 (B_0, resp.). The reasoners discover the inconsistency of $A^0 \wedge B^0$ by exchanging information on the common language, i.e., by communicating each other only the entailed quantifier-free formulae over the common variables. The information exchange is hierarchical, i.e., it is limited to $T_I \cup \mathcal{EUF}$-formulæ: literals from the richer language of $\mathcal{CARD}(T_I)$ and outside the language of $T_I \cup \mathcal{EUF}$ can contribute to the information exchange only via *instantiation* of the universal quantifiers in suitable $T_I \cup \mathcal{EUF}$-formulæ given in Section 3: these formulae, as proved in Lemmas 3.3 and 3.5, supply equivalent definitions of such literals. In contrast to [16], instantiations of universal quantifiers is limited to variables and constants for efficiency.

The main problem is to show that the above limited information exchange is sufficient. This is the case thanks to the fact that the the algorithm manipulates *iterated diff operators* [35],[16] (formally defined in Section 3) and it *gives names* to all such operators when applied to common array variables. Both the production of names for iterated diff-terms and the variable instantiations of the universal quantifiers in the equivalent universal $T_I \cup \mathcal{EUF}$-formulæ need in principle to be repeated infinitely many times; what we prove (this is the content of our main Theorem 7.4 below) is that *a pre-determined polynomial size subset of such manipulations is sufficient* for the

$T_I \cup \mathcal{EUF}$-interpolation module to produce the interpolant we are looking for. The reason why we produce such polysize instance is due to the fact that our algorithm has a *linear bound on the iterated* `diff`*-terms* that need to be introduced in Step 1 of our algorithm (see Section 7).[3]

Related work. We already mentioned the related work on first-order theories axiomatizing arrays [7, 16, 19, 22], which our theories of contiguous arrays strictly extend. Since we adopt a hierarchical approach, our method is closely related to hierarchical interpolation, where interpolants are computed by reduction to a base theory treated as black-box. A non-exhaustive summary of this literature is given by the approach in [28, 29] (where in the context of linear arithmetic general interpolation is reduced to constraint solving), by the one based on *local extensions* in [30–33] and by the one based on W-compatibility and finite instantiations of [34, 35].

2 FORMAL PRELIMINARIES

We assume the usual syntactic (e.g., signature, variable, term, atom, literal, formula, and sentence) and semantic (e.g., structure, sub-structure, truth) notions of first-order logic. All the signatures we consider are finite or countable. The equality symbol "=" is in all signatures. Notations like $E(\underline{x})$ mean that the expression (term, literal, formula, etc.) E contains free variables only from the tuple $\underline{x}$. A 'tuple of variables' is a list of variables without repetitions and a 'tuple of terms' is a list of terms (possibly with repetitions). These conventions are useful for substitutions: we use them when denoting with $\phi(\underline{t}/\underline{x})$ (or simply with $\phi(\underline{t})$) the formula obtained from $\phi(\underline{x})$ by simultaneous replacement of the 'tuple of variables' $\underline{x}$ with the 'tuple of terms' $\underline{t}$. A *constraint* is a conjunction of literals. A formula is *universal* (*existential*) iff it is obtained from a quantifier-free formula by prefixing it with a string of universal (existential, resp.) quantifiers.

Theories and satisfiability modulo theory. A *theory* T is a pair (Σ, Ax_T), where Σ is a signature and Ax_T is a set of Σ-sentences, called the *axioms* of T (we shall sometimes write directly T for Ax_T). The *models* of T are those Σ-structures in which all the sentences in Ax_T are true. A Σ-formula ϕ is T*-satisfiable* (or T-consistent) if there exists a model $\mathcal{M}$ of T such that ϕ is true in $\mathcal{M}$ under a suitable assignment a to the free variables of ϕ (in symbols, $(\mathcal{M}, \mathsf{a}) \models \phi$); it is T*-valid* (in symbols, $T \vdash \varphi$) if its negation is T-unsatisfiable or, equivalently, φ is provable from the axioms of T in a complete calculus for first-order logic. A theory $T = (\Sigma, Ax_T)$ is *universal* iff all sentences in Ax_T are universal. A formula φ_1 T*-entails* a formula φ_2 if $\varphi_1 \rightarrow \varphi_2$ is T*-valid* (in symbols, $\varphi_1 \vdash_T \varphi_2$ or simply $\varphi_1 \vdash \varphi_2$ when T is clear from the context). If Γ is a set of formulæ and ϕ a formula, $\Gamma \vdash_T \phi$ means that there are $\gamma_1, \ldots, \gamma_n \in \Gamma$ such that $\gamma_1 \wedge \cdots \wedge \gamma_n \vdash_T \phi$. The *satisfiability modulo the theory* T (SMT(T)) *problem* amounts to establishing the T-satisfiability of quantifier-free Σ-formulæ (equivalently, the T-satisfiability of Σ-constraints). Some theories have special names, which are becoming standard in SMT-literature, we shall recall some of them during the paper. As already mentioned, we shall call $\mathcal{EUF}(\Sigma)$ (or just $\mathcal{EUF}$) the pure equality theory in the signature Σ. A theory T admits *quantifier-elimination* iff for every formula $\phi(\underline{x})$ there is a quantifier-free formula $\phi'(\underline{x})$ such that $T \vdash \phi \leftrightarrow \phi'$.

Embeddings and sub-structures. The support of a structure $\mathcal{M}$ is denoted with $|\mathcal{M}|$. For a (sort, constant, function, relation) symbol σ, we denote as $\sigma^{\mathcal{M}}$ the interpretation of σ in $\mathcal{M}$. Let $\mathcal{M}$ and $\mathcal{N}$ be two Σ-structures; a Σ-embedding (or, simply, an embedding) $\mu : \mathcal{M} \longrightarrow \mathcal{N}$ is an injective function from $|\mathcal{M}|$ into $|\mathcal{N}|$ that preserves and reflects the interpretation of functions and relation symbols (see, e.g., [10] for the formal definition). If such an embedding is a set-theoretical inclusion,

[3] One could reformulate this fact using the W-separability framework from [35]; however, using this framework would not sensibly modify the proof of Theorem 7.4, so we preferred for simplicity to supply proofs within standard direct terminology.

we say that $\mathcal{M}$ is a *substructure* of $\mathcal{N}$ or that $\mathcal{N}$ is a *superstructure* of $\mathcal{M}$. As it is known, the truth of a universal (resp. existential) sentence is preserved through substructures (resp. superstructures).

Given a signature Σ and a Σ-structure $\mathcal{M}$, we indicate with $\Delta_\Sigma(\mathcal{M})$ the *diagram* of $\mathcal{M}$: this is the set of sentences obtained by first expanding Σ with a fresh constant $\bar{a}$ for every element a from $|\mathcal{M}|$ and then taking the set of ground $\Sigma \cup |\mathcal{M}|$-literals which are true in $\mathcal{M}$ (under the natural expanded interpretation mapping $\bar{a}$ to a). An easy but nevertheless important basic result (to be frequently used in our proofs), called *Robinson Diagram Lemma* [10], says that, given any Σ-structure $\mathcal{N}$, there is an embedding $\mu : \mathcal{M} \longrightarrow \mathcal{N}$ iff $\mathcal{N}$ can be expanded to a $\Sigma \cup |\mathcal{M}|$-structure in such a way that it becomes a model of $\Delta_\Sigma(\mathcal{M})$.

Combinations of theories. A theory T is *stably infinite* iff every T-satisfiable quantifier-free formula (from the signature of T) is satisfiable in an infinite model of T. By compactness, it is possible to show that T is stably infinite iff every model of T embeds into an infinite one (see, e.g., [15]). Let T_i be a stably-infinite theory over the signature Σ_i such that the $SMT(T_i)$ problem is decidable for $i = 1, 2$ and Σ_1 and Σ_2 are disjoint (i.e., the only shared symbol is equality). Under these assumptions, the *Nelson-Oppen combination result* [26] says that the SMT problem for the combination $T_1 \cup T_2$ of the theories T_1 and T_2 is decidable. Nelson-Oppen result trivially extends to many-sorted languages.

Interpolation properties. In the introduction, we roughly stated Craig's interpolation theorem [10]. In this paper, we are interested to specialize this result to the computation of quantifier-free interpolants modulo (combinations of) theories.

Definition 2.1. [Quantifier-free interpolation] A theory T *admits quantifier-free interpolation* iff for every pair of quantifier-free formulae ϕ, ψ such that $\psi \wedge \phi$ is T-unsatisfiable, there exists a quantifier-free formula θ, called an *interpolant*, such that: (i) ψ T-entails θ, (ii) $\theta \wedge \phi$ is T-unsatisfiable, and (iii) only the variables occurring in both ψ and ϕ occur in θ.

In verification, the following extension of the above definition is considered more useful.

Definition 2.2. [General quantifier-free interpolation] Let T be a theory in a signature Σ; we say that T has the *general quantifier-free interpolation property* iff for every signature Σ' (disjoint from Σ) and for every pair of ground $\Sigma \cup \Sigma'$-formulæ ϕ, ψ such that $\phi \wedge \psi$ is T-unsatisfiable,[4] there is a ground formula θ such that: (i) ϕ T-entails θ; (ii) $\theta \wedge \psi$ is T-unsatisfiable; (iv) all relations, constants and function symbols from Σ' occurring in θ also occur in ϕ and ψ.

By replacing free variables with free constants, it is easily seen that the general quantifier-free interpolation property (Definition 2.2) implies the quantifier-free interpolation property (Definition 2.1); the converse implication does not hold, however (a counterexample can be found in this paper too, see Example 5.2 below).

Amalgamation and strong amalgamation. Interpolation can be characterized semantically via amalgamation.

Definition 2.3. A universal theory T has the *amalgamation property* iff, given models $\mathcal{M}_1$ and $\mathcal{M}_2$ of T and a common submodel $\mathcal{A}$ of them, there exists a further model $\mathcal{M}$ of T (called T-amalgam) endowed with embeddings $\mu_1 : \mathcal{M}_1 \longrightarrow \mathcal{M}$ and $\mu_2 : \mathcal{M}_2 \longrightarrow \mathcal{M}$ whose restrictions to $|\mathcal{A}|$ coincide.

A universal theory T has the *strong amalgamation property* [20] if the above embeddings μ_1, μ_2 and the above model $\mathcal{M}$ can be chosen so to satisfy the following additional condition: if for some $m_1 \in |\mathcal{M}_1|, m_2 \in |\mathcal{M}_2|$ we have $\mu_1(m_1) = \mu_2(m_2)$, then there exists an element a in $|\mathcal{A}|$ such that $m_1 = a = m_2$.

[4] By this (and similar notions) we mean that $\phi \wedge \psi$ is unsatisfiable in all $\Sigma' \cup \Sigma$-structures whose Σ-reduct is a model of T.

The first point of the following theorem is an old result due to [3]; the second point is proved in [8] (where it is also suitably reformulated for theories which are not universal):

Theorem 2.4. *Let T be a universal theory. Then*

(i) *T has the amalgamation property iff it admits quantifier-free interpolants;*
(ii) *T has the strong amalgamation property iff it has the general quantifier-free interpolation property.*

We underline that, in presence of stable infiniteness, strong amalgamation is a modular property (in the sense that it transfers to signature-disjoint unions of theories), whereas amalgamation is not (see again [8] for details). As a special case, since $\mathcal{EUF}$ has strong amalgamation and is stably infinite, the following result follows:

Theorem 2.5. *If T is stably infinite and has strong amalgamation, so does $T \cup \mathcal{EUF}$.*

3 ARRAYS WITH MAXDIFF

The *McCarthy theory of arrays* [22] has three sorts ARRAY, ELEM, INDEX (called "array", "element", and "index" sort, respectively) and two function symbols rd ("read") and wr ("write") of appropriate arities; its axioms are:

$$\forall y, i, e. \qquad rd(wr(y, i, e), i) = e$$
$$\forall y, i, j, e. \qquad i \neq j \rightarrow rd(wr(y, i, e), j) = rd(y, j).$$

Arrays *with extensionality* have the further axiom

$$\forall x, y. x \neq y \rightarrow (\exists i.\ rd(x, i) \neq rd(y, i)), \tag{2}$$

called the 'extensionality' axiom. This theory is not universal and does not have quantifier-free interpolants. Here, we want to introduce a variant of this theory where Axiom (2) is skolemized via a function `diff` with a precise semantic interpretation: it returns *the biggest index* where two different arrays differ. We first need the notion of index theory.

Definition 3.1. [16] An *index theory* T_I is a mono-sorted theory (INDEX is its sort) satisfying the following conditions:

- T_I is universal, stably infinite and has the general quantifier-free interpolation property (i.e., it is strongly amalgamable, see Theorem 2.4);
- $SMT(T_I)$ is decidable;
- T_I extends the theory TO of linear orderings with a distinguished element 0.

We recall that TO is the theory whose only proper symbols (beside equality) are a binary predicate $\leq$ and a constant 0 subject to the axioms saying that $\leq$ is reflexive, transitive, antisymmetric and total. Thus, the signature of T_I contains at least the binary relation symbol $\leq$ and the constant 0. In the paper, when we speak of a T_I-term, T_I-atom, T_I-formula, etc. we mean a term, atom, formula in the signature of T_I. Below, we use the abbreviation $i < j$ for $i \leq j \wedge i \neq j$. The constant 0 is used to separate 'positive' indexes - those satisfying $0 \leq i$ - from the remaining 'negative' ones.

Examples of index theories are TO itself, integer difference logic $\mathcal{IDL}$, integer linear arithmetic $\mathcal{LIA}$, and real linear arithmetics $\mathcal{LRA}$. In order to match the requirements of Definition 3.1, one need however to make a careful choice of the language (see [8] for details): most importantly, notice that integer (resp., real) division by all positive integers should be added to the language of $\mathcal{LIA}$ (resp. $\mathcal{LRA}$). For most applications, $\mathcal{IDL}$ (which is the theory of integer numbers with 0, ordering, successor and predecessor) is sufficient as in this theory one can model counters for scanning arrays.

Given an index theory T_I, we can now introduce our *contiguous array theory with maxdiff* $\mathcal{CARD}(T_I)$ (parameterized by T_I) as follows. We still have three sorts ARRAY, ELEM, INDEX; the language includes the symbols of T_I, the read and write operations wr, rd, a binary function diff of type ARRAY $\times$ ARRAY $\longrightarrow$ INDEX, a unary function $|-|$ of type ARRAY $\longrightarrow$ INDEX, as well as constant $\bot, el$ of sort ELEM. The constant $\bot$ models an undefined value; the constant el models an element different from $\bot$ (and so it ensures that the sort ELEM is not 'practically empty', i.e. that it is not reduced to the singleton of $\bot$). The term $\mathtt{diff}(x, y)$ returns the maximum index where x and y differ and returns 0 if x and y are equal. [5] The term $|a|$ indicates the *length* of a, meaning that a is allocated in the interval $[0, |a|]$ and undefined outside. Formally, the axioms of $\mathcal{CARD}(T_I)$ include, besides the axioms of T_I, the following ones:

$$\forall y, i, e.\ |wr(y, i, e)| = |y| \tag{3}$$

$$\forall y, i.\ wr(y, i, \bot) = y \tag{4}$$

$$\forall y, i, e.\ (e \neq \bot \wedge\ 0 \leq i \leq |y|) \longrightarrow rd(wr(y, i, e), i) = e \tag{5}$$

$$\forall y, i, j, e.\ i \neq j \longrightarrow rd(wr(y, i, e), j) = rd(y, j) \tag{6}$$

$$\forall y, i.\ rd(y, i) \neq \bot \leftrightarrow 0 \leq i \leq |y| \tag{7}$$

$$\forall y.\ |y| \geq 0 \tag{8}$$

$$\forall y.\mathtt{diff}(y, y) = 0 \tag{9}$$

$$\forall x, y.\ x \neq y \longrightarrow rd(x, \mathtt{diff}(x, y)) \neq rd(y, \mathtt{diff}(x, y)). \tag{10}$$

$$\forall x, y, i.\ \mathtt{diff}(x, y) < i \longrightarrow rd(x, i) = rd(y, i). \tag{11}$$

$$\bot \neq el. \tag{12}$$

Since an array a is fully allocated only in the interval $[0, |a|]$, any reading or writing attempt outside that interval produces some runtime error in a program; similarly, it is meaningless to overwrite $\bot$ inside that interval. In our declarative context, there is nothing like a 'runtime error', so we assume that such illegal operations *simply do not produce any effect* (this is the combined effect of axioms (3),(4),(7)). However, when applying our theory to produce code annotations, the verification conditions should include that no memory violations like the above ones arise (that is, when, e.g., a term like $rd(b, i)$ occurs in a program, it should be accompanied by the proviso annotation $0 \leq i \leq |a|$, etc.).

As we shall see the above theory enjoys amalgamation (i.e., quantifier-free interpolation) but not strong amalgamation (i.e., it lacks the general quantifier-free interpolation). To restore it, it is sufficient to add some (even limited) support for constant arrays: we call the related theory $\mathcal{CARDC}(T_I)$. The extension is interesting by itself, because it increases the expressivity of the language: in $\mathcal{CARDC}(T_I)$, applying the wr operation to terms $\mathtt{Const}(i)$, one can encode all finite lists (if T_I has a reduct to $\mathcal{IDL}$). Formally, $\mathcal{CARDC}(T_I)$ has an additional unary function $\mathtt{Const} : \mathtt{INDEX} \longrightarrow \mathtt{ARRAY}$, constrained by the following axioms:

$$\forall i.|\mathtt{Const}(i)| = \max(i, 0). \tag{13}$$

$$\forall i, j.\ (0 \leq j \wedge j \leq |\mathtt{Const}(i)| \longrightarrow rd(\mathtt{Const}(i), j) = el). \tag{14}$$

(we assume without loss of generality that max is a symbol of T_I - in fact it is definable in it). What axioms (13)-(14) say is that $\mathtt{Const}(i)$ (in the meaningful case where $i \geq 0$) *represents the array of length* i *having constant value* el.

The following easy facts will be often used in our proofs:

[5] Notice that it might well be the case that $\mathtt{diff}(x, y) = 0$ for different x, y, but in that case 0 is the only index where x, y differ.

Lemma 3.2. *The following formulæ are $\mathcal{CARD}(T_I)$-valid*

$$|a| \neq |b| \rightarrow \mathtt{diff}(a,b) = \max(|a|,|b|) \tag{15}$$

$$\max(\mathtt{diff}(a,b),\mathtt{diff}(b,c)) \geq \mathtt{diff}(a,c)\ . \tag{16}$$

Proof. The first fact (15) trivially follows from (7), (10), (11). We now give the proof of the 'triangular identity' (16). Suppose for instance that we have $\mathtt{diff}(a,b) \geq \mathtt{diff}(b,c)$; for $k > \mathtt{diff}(a,b)$ we have, from (11), $rd(a,k) = rd(b,k) = rd(c,k)$. Thus, since $k > \mathtt{diff}(a,b)$ implies $rd(a,k) = rd(c,k)$, we have $\mathtt{diff}(a,c) \leq \mathtt{diff}(a,b)$ (otherwise if $\mathtt{diff}(a,c) > \mathtt{diff}(a,b)$, then $\mathtt{diff}(a,c)$ would be such a k, implying $a = c$ by axiom (10), hence $0 = \mathtt{diff}(a,c) > \mathtt{diff}(a,b)$, by axiom (11)).[6] □

The next lemma follows from the axioms of $\mathcal{CARD}(T_I)$:

Lemma 3.3. *An atom like $a = b$ is equivalent (modulo $\mathcal{CARD}(T_I)$) to*

$$\mathtt{diff}(a,b) = 0 \wedge rd(a,0) = rd(b,0)\ . \tag{17}$$

An atom like $a = wr(b,i,e)$ is equivalent (modulo $\mathcal{CARD}$) to the conjunction of the following formulae

$$\begin{gathered}(e \neq \bot \wedge 0 \leq i \leq |b|) \rightarrow rd(a,i) = e\\ (i < 0 \vee i > |b| \vee e = \bot) \rightarrow rd(a,i) = rd(b,i)\\ \forall h.\ (h \neq i \rightarrow rd(a,h) = rd(b,h)).\end{gathered} \tag{18}$$

An atom of the kind $|a| = i$ is equivalent to:

$$i \geq 0 \ \wedge\ \forall h.\ (rd(a,h) \neq \bot \leftrightarrow 0 \leq h \leq i). \tag{19}$$

Proof. That $a = b$ implies $\mathtt{diff}(a,b) = 0 \wedge rd(a,0) = rd(b,0)$ follows from axiom (9) and the substitutivity of equality. For the converse, $a \neq b$ and $\mathtt{diff}(a,b) = 0 \wedge rd(a,0) = rd(b,0)$ contradict axiom (10).

Formulae (18) are implied by $a = wr(b,i,e)$ by axioms (3), (4), (5),(6), (7). The converse is true by the extensionality axiom (2) (which holds in the strenghtened form (9) in our theory) and by (4), (5),(6), (7) again.

Formula (19) is trivially equivalent to $|a| = i$ thanks to axioms (7),(8). □

Lemma 3.4. *An atom like $\mathtt{Const}(i) = a$ is equivalent (modulo $\mathcal{CARDC}(T_I)$) to*

$$|a| = \max(i,0) \wedge \forall h.\ (0 \leq h \leq |a| \rightarrow rd(a,h) = el)\ . \tag{20}$$

Proof. The equivalence between Formula (20) and $\mathtt{Const}(i) = a$ trivially follows from axioms (7),(13),(14) and extensionality. □

Similarly to [35] and [16], we now introduce iterated diff operations, that will be used in our interpolation algorithm. In fact, in addition to $\mathtt{diff} := \mathtt{diff}_1$ we need an operator $\mathtt{diff}_2$ that returns the last-but-one index where a, b differ (0 if a, b differ in at most one index), an operator $\mathtt{diff}_3$ that returns the last-but-two index where a, b differ (0 is they differ in at most two indexes), etc. Our language is already sufficiently expressive for that. Indeed, given array variables a, b, we define by mutual recursion the sequence of array terms $b_1, b_2, \ldots$ and of index terms $\mathtt{diff}_1(a,b), \mathtt{diff}_2(a,b), \ldots$:

$$\begin{aligned} b_1 &:= b;\ \mathtt{diff}_1(a,b) := \mathtt{diff}(a,b_1);\\ b_{k+1} &:= wr(b_k, \mathtt{diff}_k(a,b), rd(a,\mathtt{diff}_k(a,b)));\\ &\mathtt{diff}_{k+1}(a,b) := \mathtt{diff}(a,b_{k+1});\end{aligned}$$

[6] Notice that $\mathtt{diff}(a,b)$ cannot be negative by the combination of axioms (7)-(10).

A useful fact is that formulae like $\bigwedge_{j<l} \mathtt{diff}_j(a,b) = k_j$ can be eliminated in favor of universal clauses in a language whose only symbol for array variables is rd. In detail:

Lemma 3.5. *A formula like*

$$\mathtt{diff}_1(a,b) = k_1 \wedge \cdots\cdots \wedge \mathtt{diff}_l(a,b) = k_l \tag{21}$$

is equivalent modulo $\mathcal{CARD}(T_I)$ to the conjunction of the following seven formulae:

$$\begin{aligned}
& k_1 \geq k_2 \wedge \ldots \wedge k_{l-1} \geq k_l \wedge k_l \geq 0 \\
& \bigwedge_{j<l} (k_j > k_{j+1} \rightarrow rd(a,k_j) \neq rd(b,k_j)) \\
& \bigwedge_{j<l} (|a| = |b| \wedge k_j = k_{j+1}) \rightarrow k_j = 0 \\
& \bigwedge_{j\leq l} (rd(a,k_j) = rd(b,k_j) \rightarrow k_j = 0) \\
\forall h.\ & (h > k_l \rightarrow rd(a,h) = rd(b,h) \vee h = k_1 \vee \ldots \vee h = k_{l-1}) \\
& |a| > |b| \rightarrow (k_1 = k_l \wedge k_l = |a|) \\
& |b| > |a| \rightarrow (k_1 = k_l \wedge k_l = |b|).
\end{aligned} \tag{22}$$

4 EMBEDDINGS

In this section we present some useful facts about embeddings that will be crucial in the proofs throughout the paper.

We first introduce the third array theory $\mathcal{AR}_{\text{ext}}(T_I)$, which is weaker than $\mathcal{CARD}(T_I)$, lacks just the `diff` symbol and axioms (9),(10),(11) are replaced by the following extensionality axiom:

$$\forall x, y.\ x \neq y \rightarrow (\exists i,\ rd(x,i) \neq rd(y,i)). \tag{23}$$

Notice that $\mathcal{AR}_{\text{ext}}(T_I) \subseteq \mathcal{CARD}(T_I) \subseteq \mathcal{CARDC}(T_I)$ (the inclusion holds both for signatures and for axioms). To simplify the statements of some lemmas below, let us also introduce the theory $\mathcal{CARC}_{\text{ext}}(T_I)$: this theory is obtained from $\mathcal{AR}_{\text{ext}}(T_I)$ by adding the function symbol `Const` to the signature and the sentences (13),(14) to the axioms.

We now discuss the class of models of $\mathcal{AR}_{\text{ext}}(T_I)$ and we clarify the important features of embeddings between such models. A model $\mathcal{M}$ of $\mathcal{AR}_{\text{ext}}(T_I)$ is *functional* when the following conditions are satisfied:

(i) $\mathtt{ARRAY}^{\mathcal{M}}$ is a subset of the set of all positive-support functions from $\mathtt{INDEX}^{\mathcal{M}}$ to $\mathtt{ELEM}^{\mathcal{M}}$ (a function a is *positive-support* iff there exists an index $|a|$ such that $|a| \geq 0$ and, for every j, $a(j) \neq \perp$ iff $j \in [0,|a|]$);
(ii) rd is function application;
(iii) wr is the point-wise update operation inside the interval $[0,|a|]$ (i.e., function $wr(a,i,e)$ returns the same values as function a, except at the index i and only in case $i \in [0,|a|]$: in this case it returns the element e);
(iv) if $\mathcal{M}$ is also a model of $\mathcal{CARC}_{\text{ext}}(T_I)$, then the set $\mathtt{ARRAY}^{\mathcal{M}}$ contains the positive-support functions with value $el^{\mathcal{M}}$ inside their support.

Because of the extensionality axiom (23), it can be shown that every model of $\mathcal{AR}_{\text{ext}}(T_I)$ or of $\mathcal{CARC}_{\text{ext}}(T_I)$ is *isomorphic to a functional one*. For an array $a \in \mathtt{ARRAY}^{\mathcal{M}}$ in a functional model $\mathcal{M}$ and for $i \in \mathtt{INDEX}^{\mathcal{M}}$, since a is a function, we interchangeably use the notations $a(i)$ and $rd(a,i)$.

Let a, b be elements of $\mathtt{ARRAY}^{\mathcal{M}}$ in a model $\mathcal{M}$. We say that *a and b are cardinality equivalent* iff $|a| = |b|$ and $\{i \in \mathtt{INDEX}^{\mathcal{M}} \mid \mathcal{M} \models rd(a,i) \neq rd(b,i)\}$ is finite. This relation in $\mathcal{M}$ is an equivalence, that we denote as $\sim_{\mathcal{M}}$ or simply as $\sim$. We also write $\mathcal{M} \models a \sim b$ to say that $a \sim_{\mathcal{M}} b$ holds.

Lemma 4.1. *Let $\mathcal{N}$, $\mathcal{M}$ be models of $\mathcal{AR}_{\text{ext}}(T_I)$ such that $\mathcal{M}$ is a substructure of $\mathcal{N}$. For every $a, b \in \mathtt{ARRAY}^{\mathcal{M}}$, we have that*

$$\mathcal{M} \models a \sim b \quad \text{iff} \quad \mathcal{N} \models a \sim b.$$

Proof. The left-to-right side is trivial because if $\mathcal{M} \models a \sim b$ then a and b have equal length in $\mathcal{M}$ and in $\mathcal{N}$ too because length is preserved; moreover, $\mathcal{M} \models a = wr(b, I, E)$, where $I \equiv i_1, \ldots, i_n$ is a list of costants (naming elements of $\mathcal{M}$) of sort INDEX, $E \equiv e_1, \ldots, e_n$ is a list of costants (naming elements of $\mathcal{M}$) of sort ELEM, and $wr(b, I, E)$ abbreviates the term $wr(wr(\cdots wr(b, i_1, e_1) \cdots), i_n, e_n)$. Thus, also $\mathcal{N} \models a = wr(b, I, E)$ because $\mathcal{M}$ is a substructure of $\mathcal{N}$. Vice versa, suppose that $\mathcal{M} \not\models a \sim b$. This means that either $|a| \neq |b|$ or that there are infinitely many $i \in \mathtt{INDEX}^{\mathcal{M}}$ such that $rd^{\mathcal{M}}(a, i) \neq rd^{\mathcal{M}}(b, i)$. Since $\mathcal{M}$ is a substructure of $\mathcal{N}$, these conditions holds in $\mathcal{N}$ too. □

In a functional model $\mathcal{M}$ of $\mathcal{AR}_{\text{ext}}(T_I)$, we say that $\mathtt{diff}(a, b)$ is *defined* iff there is a maximum index where a, b differ (or if $a = b$). If in the model $\mathcal{M}$ the index sort INDEX is interpreted as the set of the integers, with standard ordering, then for any two positive-support functions a, b, we have that $\mathtt{diff}(a, b)$ is defined. However, this will not be the case if the index sort INDEX is interpreted, e.g., in some non-standard model of the integers. We must take into considerations these models too, since we want to prove amalgamation. For this purpose, we need to build amalgams for *all* models of the theory (only in that case in fact, amalgamation turns out to be equivalent to quantifier-free interpolation). Thus, we are forced to take into consideration below also phenomena that might arise only in non-standard models.

An embedding $\mu : \mathcal{M} \longrightarrow \mathcal{N}$ between $\mathcal{AR}_{\text{ext}}(T_I)$-models (or of $\mathcal{CAR}\mathcal{C}_{\text{ext}}(T_I)$-models) is said to be diff-faithful iff, whenever $\mathtt{diff}(a, b)$ is defined, so is $\mathtt{diff}(\mu(a), \mu(b))$ and it is equal to $\mu(\mathtt{diff}(a, b))$. Since there might not be a maximum index where a, b differ, in principle it is not always possible to expand a functional model of $\mathcal{AR}_{\text{ext}}(T_I)$ to a functional model of $\mathcal{CAR}\mathcal{D}(T_I)$, if the set of indexes remains unchanged. Indeed, in order to do that in a diff-faithful way, one needs to explicitly add to $\mathtt{INDEX}^{\mathcal{M}}$ new indexes including at least the ones representing the missing maximum indexes where two given array differ. This idea leads to Theorem 4.4 below, which is the main result of the current section. We first need a couple of lemmas.

Lemma 4.2. *Let $\mathcal{M}$ be a model of $\mathcal{AR}_{\text{ext}}(T_I)$ and let $a, a', b, b' \in \mathtt{ARRAY}^{\mathcal{M}}$; if $a \sim_{\mathcal{M}} a'$, $b \sim_{\mathcal{M}} b'$ and $\mathtt{diff}_k(a', b')$ is defined for every k, then $\mathtt{diff}(a, b)$ is also defined.*

Proof. Notice first that, from $a \sim a'$ and $b \sim b'$, it follows that $|a| = |a'|$ and $|b| = |b'|$. In case $|a| \neq |b|$, we have that $\mathtt{diff}(a, b) = \max\{|a|, |b|\}$ (see Lemma 3.2), which implies that $\mathtt{diff}(a, b)$ is defined. Hence, the relevant case is when we have $l = |a| = |b| = |a'| = |b'|$ and $a' \nsim b'$ (if $a' \sim b'$, then we have also $a \sim b$ and the maximum of the finitely many indexes where a, b differ is $\mathtt{diff}(a, b)$). Then for the infinitely many indexes $j_k = \mathtt{diff}_k(a', b')$ we have $a'(j_k) \neq b'(j_k)$; for at least one of such j_k we must also have $a(j_k) \neq b(j_k)$ because $a \sim a'$ and $b \sim b'$. Consider now the indexes in $[j_k, l]$: in this interval, the pair of arrays a, b differs on at least one but at most finitely many indices (because a, a' differs on finitely many indices there and so do the pairs b, b' and a', b'), so the biggest one such index will be $\mathtt{diff}(a, b)$. □

Lemma 4.3. *Let $\mathcal{M}$ be a model of $\mathcal{AR}_{\text{ext}}(T_I)$. There exist a model $\mathcal{N}$ of $\mathcal{AR}_{\text{ext}}(T_I)$ and a* diff*-faithful embedding $\mu : \mathcal{M} \to \mathcal{N}$ such that the restriction of μ to the sort* ELEM *is not surjective. In addition, if $\mathcal{M}$ is a model of $\mathcal{CAR}\mathcal{C}_{\text{ext}}(T_I)$, $\mathcal{CAR}\mathcal{D}(T_I)$ or of $\mathcal{CAR}\mathcal{DC}(T_I)$, so is $\mathcal{N}$.*

Proof. To build $\mathcal{N}$ it is sufficient to put:

- $\mathtt{INDEX}^{\mathcal{N}} = \mathtt{INDEX}^{\mathcal{M}}$,
- $\mathtt{ELEM}^{\mathcal{N}} = \mathtt{ELEM}^{\mathcal{M}} \cup \{e\}$ where $e \notin \mathtt{ELEM}^{\mathcal{M}}$,

- $\mathtt{ARRAY}^{\mathcal{N}}$ consists of the positive-support functions $a : \mathtt{INDEX}^{\mathcal{N}} \longrightarrow \mathtt{ELEM}^{\mathcal{N}}$ for which there exist $a' \in \mathtt{ARRAY}^{\mathcal{M}}$ such that $a \sim a'$.

Now notice that if $\mathtt{diff}$ is totally defined in $\mathcal{M}$, so it is in $\mathcal{N}$. In fact, this follows from the definition of $\mathtt{ARRAY}^{\mathcal{N}}$ and Lemma 4.2: if $a \sim a'$, $b \sim b'$ and $\mathtt{diff}_k(a', b')$ is defined for every k, then $\mathtt{diff}(a, b)$ is also defined by the previous lemma. The claim is proved in the same way for all the above mentioned array theories (notice that in case the signature includes the $\mathtt{Const}$ symbol, μ trivially preserves it). □

Theorem 4.4. *For every index theory T_I, every model $\mathcal{M}$ of $\mathcal{AR}_{\mathrm{ext}}(T_I)$ (resp. of $\mathcal{CARC}_{\mathrm{ext}}(T_I)$) has a $\mathtt{diff}$-faithful embedding into a model of $\mathcal{CARD}(T_I)$ (resp. of $\mathcal{CARDC}(T_I)$).*

Proof. It is sufficient to well-order the pairs $a, b \in \mathtt{ARRAY}^{\mathcal{M}}$ such that $\mathtt{diff}(a, b)$ is not defined in $\mathcal{M}$, apply to each pair the construction of the next lemma (taking unions at limit ordinals), and then repeat the whole construction ω times, taking union again. In doing this, we make use of the fact that the models of $\mathcal{AR}_{\mathrm{ext}}(T_I)$ (resp. of $\mathcal{CARC}_{\mathrm{ext}}(T_I)$) are closed under unions of chains, since $\mathcal{AR}_{\mathrm{ext}}(T_I)$ (resp. $\mathcal{CARC}_{\mathrm{ext}}(T_I)$) is a theory comprising only $\forall^*\exists^*$ axioms (see [10] for this and other preservation results).

Formally, consider the set of all pairs (a, b) of arrays in $\mathcal{M}$ such that $\mathtt{diff}(a, b)$ is not defined in $\mathcal{M}$. By Zermelo's Theorem, the set of such pairs (a, b) can be well-ordered: let $\{(a_i, b_i)\}_{i \in I}$ be such a well-ordered set of pairs, where I is some ordinal. By transfinite induction on this well-order, we define $\mathcal{M}_0 := \mathcal{M}$ and, for each $i \in I$, $\mathcal{M}_i := \mathcal{N}$ as an extension of $\bigcup_{j<i} \mathcal{M}_j$ such that (i) $\mathcal{N}$ is a model of $\mathcal{AR}_{\mathrm{ext}}(T_I)$ (of $\mathcal{CARC}_{\mathrm{ext}}(T_I)$ if $\mathcal{M}$ is a model of $\mathcal{CARC}_{\mathrm{ext}}(T_I)$, resp.); (ii) $\bigcup_{j<i} \mathcal{M}_j$ has a $\mathtt{diff}$-faithful embedding into $\mathcal{N}$; and (iii) $\mathtt{diff}^{\mathcal{N}}(a_i, b_i)$ is defined (this $\mathcal{N}$ exists thanks to the next lemma).

Now we take the chain union $\mathcal{M}^1 := \bigcup_{i \in I} \mathcal{M}_i$. Thanks to this construction, we get that, for every pair (a_i, b_i) with $a_i, b_i \in \mathtt{ARRAY}^{\mathcal{M}}$, $\mathtt{diff}(a, b)$ becomes defined in $\mathcal{M}^1$; however, this only guarantees that $\mathtt{diff}$ is defined for every pair (a_i, b_i) such that a_i, b_i are in $\mathtt{ARRAY}^{\mathcal{M}}$, whereas nothing is said for the pairs a, b in $\mathtt{ARRAY}^{\mathcal{M}^1} \setminus \mathtt{ARRAY}^{\mathcal{M}}$. Then, we iteratively repeat the chain construction above for these new (a, b). Indeed, it is possible to construct, by an analogous chain argument, a model $\mathcal{M}^2$ as done above, starting from $\mathcal{M}^1$ instead of $\mathcal{M}$. Clearly, we get $\mathcal{M}_0 := \mathcal{M} \subseteq \mathcal{M}^1 \subseteq \mathcal{M}^2$ by construction.

At this point, we iterate the same argument countably many times, so as to define a new chain of models of $\mathcal{AR}_{\mathrm{ext}}(T_I)$ (of $\mathcal{CARC}_{\mathrm{ext}}(T_I)$):

$$\mathcal{M}_0 := \mathcal{M} \subseteq \mathcal{M}^1 \subseteq \ldots \subseteq \mathcal{M}^n \subseteq \ldots$$

Defining $\mathcal{M}' := \bigcup_n \mathcal{M}^n$, we immediately obtain that $\mathcal{M}'$ is a model of $\mathcal{CARD}(T_I)$ (resp. of $\mathcal{CARDC}(T_I)$). such that $\mathcal{M} \subseteq \mathcal{M}'$. After ω steps we are done, because every pair (a, b) appearing in $\mathcal{M}^i$ occurs after finitely many steps, and its corresponding $\mathtt{diff}(a, b)$ becomes defined in $\mathcal{M}^{i+1}$. □

Lemma 4.5. *Let $\mathcal{M}$ be a model of $\mathcal{AR}_{\mathrm{ext}}(T_I)$ (resp. of $\mathcal{CARC}_{\mathrm{ext}}(T_I)$) and let $a, b \in \mathtt{ARRAY}^{\mathcal{M}}$ be such that $\mathtt{diff}^{\mathcal{M}}(a, b)$ is not defined. Then there are a model $\mathcal{N}$ of $\mathcal{AR}_{\mathrm{ext}}(T_I)$ (resp. of $\mathcal{CARC}_{\mathrm{ext}}(T_I)$) and a $\mathtt{diff}$-faithful embedding $\mu : \mathcal{M} \to \mathcal{N}$ such that $\mathtt{diff}^{\mathcal{N}}(a, b)$ is defined.*

Proof. Thanks to Lemma 4.3, we can assume that $\mathtt{ELEM}^{\mathcal{M}}$ has at least an element e (different from $\bot^{\mathcal{M}}, el^{\mathcal{M}}$). Notice that we must have $|a| = |b|$, otherwise $\mathtt{diff}(a, b)$ is defined and it is $\max(|a|, |b|)$ according to Lemma 3.2.

Let $I = \{i \in \mathtt{INDEX}^{\mathcal{M}} \mid a(i) \neq b(i)\}$ be the set of indices without maximum element (hence infinite) where they differ. Let $\downarrow I := \{j \in \mathtt{INDEX}^{\mathcal{M}} \mid \exists i \in I,\ j \leq i\} \supseteq I$. Notice that the condition

$$(+) \quad \text{“}\exists i \in I\, \forall j \in I\, (j \geq i \rightarrow x(j) = el)\text{”}$$

cannot be satisfied both for $x = a$ and $x = b$: indeed, if this were the case, assuming w.l.o.g. that $i_a \leq i_b$ (where i_a and i_b are the witnesses for the existentially quantified index i in (+) for $x = a$ and $x = b$ respectively), we would have that $a(j) = el^{\mathcal{M}} = b(j)$ for all $j \geq i_b$, $j \in I$, which is a contradiction with the definition of I. In case one of them satisfies it, we assume it is b.

Let Δ be the Robinson diagram of the T_I-reduct of $\mathcal{M}$ and let k_0 be a new constant; let us introduce the set

$$\Delta' := \Delta \cup \{i < k_0 \mid i \in \downarrow I\} \cup \{k_0 < i \mid i \in \mathtt{INDEX}^{\mathcal{M}} \setminus \downarrow I\}.$$

By the compactness theorem for first order logic and since I is infinite, the set Δ' turns out to be consistent. In fact, if Δ' were inconsistent, then there would exist a finite subset of it not admitting a model. However, a finite subset of Δ' can contain constraints only for a finite number of index constants d occurring in Δ, $i \in \downarrow I$, $i' \in \mathtt{INDEX}^{\mathcal{M}} \setminus \downarrow I$ and k_0. Such constraints can be verified inside the T_I-reduct of $\mathcal{M}$ itself: to interpret the additional constant k_0, it is sufficient to use the fact that I contains arbitrarily large indexes and the fact that the definition of $\downarrow I$ implies that

$$\forall i \in \downarrow I,\ \forall j \in \mathtt{INDEX}^{\mathcal{M}} \setminus \downarrow I, \quad i < j.$$

By Robinson Diagram Lemma, there exists a model $\mathcal{A}$ of T_I extending the T_I-reduct of $\mathcal{M}$; such $\mathcal{A}$ contains in its support an element k_0 such that

$$\forall i \in \downarrow I,\ i < k_0,$$

$$\forall i \in \mathtt{INDEX}^{\mathcal{M}} \setminus \downarrow I,\ k_0 < i.$$

We now take $\mathtt{ELEM}^{\mathcal{N}} = \mathtt{ELEM}^{\mathcal{M}}$, $\mathtt{INDEX}^{\mathcal{N}} = \mathtt{INDEX}^{\mathcal{A}}$; we let also $\mathtt{ARRAY}^{\mathcal{N}}$ to be the set of all positive-support functions from $\mathtt{INDEX}^{\mathcal{N}}$ into $\mathtt{ELEM}^{\mathcal{N}}$ (notice that this $\mathcal{N}$ is trivially also a model of $\mathcal{CARC}_{\mathrm{ext}}(T_I)$). We observe that $k_0 < |a|^{\mathcal{M}}$ and recall that $|a|^{\mathcal{M}} = |b|^{\mathcal{M}}$.

Let us now define the embedding $\mu : \mathcal{M} \rightarrow \mathcal{N}$; at the level of the sorts INDEX and ELEM, we use inclusions. For the ARRAY sort, we need to specify the value $\mu(c)(k)$ for $c \in \mathtt{ARRAY}^{\mathcal{M}}$ and $k \in \mathtt{INDEX}^{\mathcal{N}} \setminus \mathtt{INDEX}^{\mathcal{M}}$ (for the other indices we keep the old $\mathcal{M}$-value to preserve the read operation). Our definition for μ must preserve the maxdiff index (whenever already defined in $\mathcal{M}$) and must guarantee that $\mathtt{diff}^{\mathcal{N}}(\mu(a), \mu(b)) = k_0$ (by construction, we have $k_0 > 0$). For a generic array $c \in \mathtt{ARRAY}^{\mathcal{M}}$, we operate as follows:

(1) if $|c|^{\mathcal{M}} < k_0$ we put $\mu(c)(k_0) = \bot^{\mathcal{M}}$, otherwise:
(2) if $|c|^{\mathcal{M}} \geq k_0$ and the condition ($\star$) below holds, we put $\mu(c)(k_0) = e$,
(3) if $|c|^{\mathcal{M}} \geq k_0$ and the condition ($\star$) below does not hold, we put $\mu(c)(k_0) = el^{\mathcal{M}}$.

The condition ($\star$) is specified as follows:

($\star$) there is $i \in I$ such that for all $j \in I$, $j \geq i$ we have $c(j) = a(j)$.

For all the remaining indexes $k \in \mathtt{INDEX}^{\mathcal{N}} \setminus (\mathtt{INDEX}^{\mathcal{M}} \cup \{k_0\})$ we put

$$\mu(c)(k) = \begin{cases} \bot^{\mathcal{M}}, & \text{if } k \notin [0, |c|^{\mathcal{M}}] \\ el^{\mathcal{M}}, & \text{if } k \in [0, |c|^{\mathcal{M}}] \end{cases} \qquad (4)$$

Notice that we have $\mu(a)(k_0) = e \neq el^{\mathcal{M}} = \mu(b)(k_0)$ (the last equality holds because I is infinite and does not have maximum, hence condition ($\star$) holds for a but not for b). In addition:

- for all $i \in \mathtt{INDEX}^{\mathcal{M}}$ such that $k_0 < i$, we have $i \notin \downarrow I$, according to the construction of k_0 and consequently $i \notin I$, that is $a(i) = b(i)$;

- for all $i \in \mathtt{INDEX}^{\mathcal{N}} \setminus (\mathtt{INDEX}^{\mathcal{M}} \cup \{k_0\})$ such that $k_0 < i$, since we have $|a|^{\mathcal{M}} = |b|^{\mathcal{M}}$, we get

$$\mu(a)(i) = \bot^{\mathcal{M}} \text{ iff } \mu(b)(i) = \bot^{\mathcal{M}},$$
$$\mu(a)(i) = el^{\mathcal{M}} \text{ iff } \mu(b)(i) = el^{\mathcal{M}}.$$

Hence, we can conclude that $\mathtt{diff}^{\mathcal{N}}(\mu(a), \mu(b))$ is defined and equal to k_0.

We only need to check that our μ preserves $rd, |-|, wr$, constant arrays and $\mathtt{diff}$ (whenever defined).

The operation rd is preserved because μ acts as an inclusion for indexes and elements and because we have $\mu(c)(k) = c(k)$ if $k \in \mathtt{INDEX}^{\mathcal{M}}$.

Concerning length, we have $|\mu(c)|^{\mathcal{N}} = |c|^{\mathcal{M}}$ because of (4) and because of the above definition of $\mu(c)(k_0)$ (recall that $k_0 > 0$).

Concerning write operation, we prove that for all $c \in \mathtt{ARRAY}^{\mathcal{M}}$, $i \in \mathtt{INDEX}^{\mathcal{M}} \cap [0, |c|]$ and $e' \in \mathtt{ELEM}^{\mathcal{M}} \setminus \{\bot^{\mathcal{M}}\}$ we have

$$\mu(wr(c, i, e')) = wr(\mu(c), i, e').$$

Remember that we have $|wr(c, i, e')| = |c| = |\mu(c)|$.

- For $k \neq i$ in $\mathtt{INDEX}^{\mathcal{M}}$

$$\mu(wr(c, i, e'))(k) = wr(c, i, e')(k) = c(k)$$
$$wr(\mu(c), i, e')(k) = \mu(c)(k) = c(k);$$

- For $k = i$

$$\mu(wr(c, i, e'))(i) = wr(c, i, e')(i) = e'$$
$$wr(\mu(c), i, e')(i) = e';$$

- For $k = k_0 > |c|$ the claim follows immediately from the definition;
- For $k = k_0 < |c|$, $(\star)$ holds for c iff it holds for $wr(c, i, e')$, because I is infinite. Hence we have

$$\mu(wr(c, i, e'))(k_0) = e \text{ iff } \mu(c)(k_0) = e.$$

- For $k \in \mathtt{INDEX}^{\mathcal{N}} \setminus (\mathtt{INDEX}^{\mathcal{M}} \cup \{k_0\})$, the claim is clear from the definition and from the fact that wr preserves length.

For constant arrays, we must show only that $\mu(\mathtt{Const}(i))(k_0) = el^{\mathcal{M}}$ in case $k_0 < i$: this is clear, because a does not satisfy (+), hence $(\star)$ does not hold for $\mathtt{Const}(i)$.

Let us now finally consider the $\mathtt{diff}$ operation and let us prove that if $\mathtt{diff}^{\mathcal{M}}(c_1, c_2)$ is defined, then $\mathtt{diff}^{\mathcal{N}}(\mu(c_1), \mu(c_2))$ is also defined and equal to it. Assume that $\mathtt{diff}^{\mathcal{M}}(c_1, c_2)$ is defined; since μ preserve length, the only relevant case, in view of Lemma 3.2, is when we have $|c_1| = |c_2|$; since the values of c_1, c_2 on indexes from $\mathcal{M}$ are preserved, taking in mind (4) (in particular, taking in mind that, for $k \in \mathtt{INDEX}^{\mathcal{N}} \setminus (\mathtt{INDEX}^{\mathcal{M}} \cup \{k_0\})$, we have $\mu(c_1)(k) = \mu(c_2)(k)$), we only have to exclude that we have

$$\mathtt{diff}^{\mathcal{M}}(c_1, c_2) < k_0 \text{ and } \mu(c_1)(k_0) \neq \mu(c_2)(k_0)$$

If this is the case, we have, e.g., $\mu(c_1)(k_0) = e \neq el^{\mathcal{M}} = \mu(c_2)(k_0)$ ($k_0 < |c_1| = |c_2|$), which implies that $(\star)$ holds for c_1 but not for c_2. However, it cannot be that $(\star)$ holds for only one among c_1, c_2. The reason for this is as follows. Indeed, if $i \in I$ is the index that witnesses $(\star)$ for c_1, then $c_1(j) = a(j)$ for all indexes $j \in I$ such that $j \geq i$ (which are infinitely many). Since I is infinite and without maximum and since $\mathtt{diff}^{\mathcal{M}}(c_1, c_2) < k_0$, we must have $\mathtt{diff}^{\mathcal{M}}(c_1, c_2) \in \downarrow I$ by the definition of Δ, so there must be infinitely many indices in I bigger than $\mathtt{diff}^{\mathcal{M}}(c_1, c_2)$ and

arbitrarily large, which means in particular that there exists an index $i' \in I$ such that $i' \geq i$ and $i' > \mathtt{diff}(c_1, c_2)$. This i' witnesses ($\star$) for c_2, as wanted. This concludes the proof. □

5 STRONG AMALGAMATION FOR $\mathcal{CARDC}(T_I)$

In this section, we prove that the most expressive theory of the paper $\mathcal{CARDC}(T_I)$ has strong amalgamation. However, we also show that this is not the case for $\mathcal{CARD}(T_I)$ (even if it is amalgamable). We recall that strong amalgamation holds for models of T_I (see Definition 3.1): this observation is crucial for the following.

Strong amalgamation of $\mathcal{CARDC}(T_I)$ will be proved in two steps. First, we provide the amalgam construction for $\mathcal{CARDC}(T_I)$, where we also notice that the same arguments can be used to prove that $\mathcal{CARD}(T_I)$ has amalgamation too. Then, after exhibiting a counterexample showing that the strong amalgamation fails for $\mathcal{CARD}(T_I)$, we check that the amalgam construction for $\mathcal{CARDC}(T_I)$ satisfies the condition for being a $\mathcal{CARDC}(T_I)$-strong amalgam.

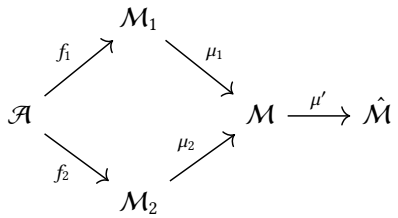
5.1 Amalgam constructions

Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two models of $\mathcal{CARDC}(T_I)$ (resp. of $\mathcal{CARD}(T_I)$); we want to amalgamate them over their common substructure $\mathcal{A}$ and let f_i be the embedding of $\mathcal{A}$ into $\mathcal{M}_i$ (we assume that f_i is just inclusion for the INDEX and ELEM components). We can assume w.l.o.g. that our models are all functional and, by applying renaming, that

$$(\mathtt{INDEX}^{\mathcal{M}_1} \setminus \mathtt{INDEX}^{\mathcal{A}}) \cap (\mathtt{INDEX}^{\mathcal{M}_2} \setminus \mathtt{INDEX}^{\mathcal{A}}) = \emptyset$$

$$(\mathtt{ELEM}^{\mathcal{M}_1} \setminus \mathtt{ELEM}^{\mathcal{A}}) \cap (\mathtt{ELEM}^{\mathcal{M}_2} \setminus \mathtt{ELEM}^{\mathcal{A}}) = \emptyset.$$

We build the amalgamated model in two steps. We first embed $\mathcal{M}_1$ and $\mathcal{M}_2$, via the embeddings μ_i (i=1,2), into a model $\mathcal{M}$ of $\mathcal{CARC}_{\text{ext}}(T_I)$ (resp., of $\mathcal{AR}_{\text{ext}}(T_I)$) in a diff-faithful way. Then $\mathcal{M}$ is embedded, via another diff-faithful embedding μ' into a model $\hat{\mathcal{M}}$ of $\mathcal{CARDC}(T_I)$ (resp., of $\mathcal{CARD}(T_I)$): μ' is guaranteed to exist by Theorem 4.4.



Construction of μ_i

We build the model $\mathcal{M}$ and the two diff-faithful embeddings $\mu_i : \mathcal{M}_i \to \mathcal{M}$ such that $\mu_1 \circ f_1 = \mu_2 \circ f_2$. We let $\mathtt{INDEX}^{\mathcal{M}}$ be a strong amalgam of $\mathtt{INDEX}^{\mathcal{M}_1}$ and $\mathtt{INDEX}^{\mathcal{M}_2}$ (T_I enjoys strong amalgamation), whereas we let $\mathtt{ELEM}^{\mathcal{M}} = \mathtt{ELEM}^{\mathcal{M}_1} \cup \mathtt{ELEM}^{\mathcal{M}_2}$. Let $\mathtt{ARRAY}^{\mathcal{M}}$ be the set of all positive-support functions from $\mathtt{INDEX}^{\mathcal{M}}$ into $\mathtt{ELEM}^{\mathcal{M}}$.

The INDEX and ELEM components of the embeddings μ_i will be inclusions. The definition of the value of $\mu_i(a)(k)$, for $a \in \mathtt{ARRAY}^{\mathcal{M}_i}$ and $k \in \mathtt{INDEX}^{\mathcal{M}}$, is given by cases as follows:

- if $k \in \mathtt{INDEX}^{\mathcal{M}_i}$, we put $\mu_i(a)(k) = a(k)$;

- if $k \in \mathtt{INDEX}^{\mathcal{M}_{3-i}} \setminus \mathtt{INDEX}^{\mathcal{A}}$: let (2$\star$) be the relation[7]

$$''\text{there exist } c \in \mathtt{ARRAY}^{\mathcal{A}}, b \in \mathtt{ARRAY}^{\mathcal{M}_i} \text{ s.t. } b \sim^{\mathcal{M}_i} a,\ k > \mathtt{diff}^{\mathcal{M}_i}(b, f_i(c))''$$

we put

$$\mu_i(a)(k) = \begin{cases} f_{3-i}(c)(k), & \text{if (2}\star\text{) holds} \\ \bot^{\mathcal{M}}, & \text{if (2}\star\text{) does not hold \& } k \notin [0, |a|^{\mathcal{M}_i}] \\ el^{\mathcal{M}}, & \text{if (2}\star\text{) does not hold \& } k \in [0, |a|^{\mathcal{M}_i}] \end{cases}$$

- if $k \notin \mathtt{INDEX}^{\mathcal{M}_i} \cup \mathtt{INDEX}^{\mathcal{M}_{3-i}}$, we put

$$\mu_i(a)(k) = \begin{cases} \bot^{\mathcal{M}}, & \text{if } k \notin [0, |a|^{\mathcal{M}_i}] \\ el^{\mathcal{M}}, & \text{if } k \in [0, |a|^{\mathcal{M}_i}]. \end{cases}$$

Requirements check for the amalgamated model

The model $\mathcal{M}$ introduced above is in fact a $\mathcal{CARDC}(T_I)$-amalgam (and also a $\mathcal{CARD}(T_I)$-amalgam) for the models $\mathcal{M}_1$ and $\mathcal{M}_2$ with the common substructure $\mathcal{A}$:

Theorem 5.1. *$\mathcal{CARD}(T_I)$ and $\mathcal{CARDC}(T_I)$ enjoy the amalgamation property.*

Proof. We need to prove that the functions μ_i: (i) are well-defined, (ii) are injective, (iii) preserve $|-|$, (iv) preserve rd and wr, (v) preserve $\mathtt{diff}$, (vi) satisfy the condition $\mu_1 \circ f_1 = \mu_2 \circ f_2$, (vii) preserve constant arrays (for the statement about $\mathcal{CARDC}(T_I)$).

(i) Since $\mathtt{INDEX}^{\mathcal{M}}$ is a strong amalgam of $\mathtt{INDEX}^{\mathcal{M}_1}$ and $\mathtt{INDEX}^{\mathcal{M}_2}$, the case distinctions we made for defining $\mu_i(a)(k)$ are non-overlapping and exhaustive.
We now show that if, for $i = 1, 2$ and $k \in \mathtt{INDEX}^{\mathcal{M}_{3-i}} \setminus \mathtt{INDEX}^{\mathcal{A}}$ and $a \in \mathtt{ARRAY}^{\mathcal{M}_i}$, the relation (2$\star$) holds relatively to two different pairs of arrays (c_1, b_1), (c_2, b_2) from $\mathtt{ARRAY}^{\mathcal{A}} \times \mathtt{ARRAY}^{\mathcal{M}_i}$, then we nevertheless have $f_{3-i}(c_1)(k) = f_{3-i}(c_2)(k)$ (this proves the consistency of the definition). For symmetry, let us consider only the case $i = 1$. Since the index ordering is total, let us suppose that we have for instance

$$k > \mathtt{diff}^{\mathcal{M}_1}(b_1, f_1(c_1)) \geq \mathtt{diff}^{\mathcal{M}_1}(b_2, f_1(c_2)). \tag{24}$$

By the transitivity of $\sim^{\mathcal{M}_1}$, b_1 and b_2 differ on finitely many indices, hence we can consider the finite sets

$$J := \{j \in \mathtt{INDEX}^{\mathcal{A}} \mid b_1(j) \neq b_2(j), j > \mathtt{diff}^{\mathcal{M}_1}(b_1, f_1(c_1))\}$$

$$E := \{b_1(j) \mid j \in J\} \subseteq \mathtt{ELEM}^{\mathcal{A}}.$$

Let now pick $c := wr(c_2, J, E)$; then $c \sim^{\mathcal{A}} c_2$. Since f_2 is an embedding, we have $f_2(c)(k) = f_2(c_2)(k)$ for all $k \in \mathtt{INDEX}^{\mathcal{M}_2} \setminus \mathtt{INDEX}^{\mathcal{A}}$. Suppose we have also

$$\mathtt{diff}^{\mathcal{A}}(c_1, c) \leq \mathtt{diff}^{\mathcal{M}_1}(f_1(c_1), b_1). \tag{25}$$

Then

$$\mathtt{diff}^{\mathcal{A}}(c_1, c) \leq \mathtt{diff}^{\mathcal{M}_1}(f_1(c_1), b_1) < k$$

and consequently also the desired equality

$$f_2(c_1)(k) = f_2(c)(k) = f_2(c_2)(k)$$

follows.

[7]When we write $k > \mathtt{diff}^{\mathcal{M}_i}(b, f_i(c))$ we mean in fact that $k > \mu_i(\mathtt{diff}^{\mathcal{M}_i}(b, f_i(c)))$ (this relation is meant to hold in $\mathcal{M}$). Our simplified notation is justified by the fact that μ_i is inclusion for $\mathtt{INDEX}$ sort. The fact that the choice of the c satisfying (2$\star$) is immaterial is shown in the proof of Theorem 5.1, see the item (i) below.

In order to prove (25), we consider $j \in \mathtt{INDEX}^{\mathcal{A}}$ with $j > \mathtt{diff}^{\mathcal{M}_1}(f_1(c_1), b_1)$ and show that we have $c(j) = c_1(j)$ (in fact, if this is true, then (25) cannot fail because $\mathtt{diff}^{\mathcal{A}}(c_1, c)$ would be such a j, absurd). Suppose not, i.e. that $c(j) \neq c_1(j)$; then we cannot have $j \in J$ otherwise, by definition of c and E, we would have $c(j) = b_1(j) = f_1(c_1)(j) = c_1(j)$ (f_1 is inclusion for indexes and $J \subseteq \mathtt{INDEX}^{\mathcal{A}}$), contradiction. Hence, we have $j \notin J$, so $c(j) = c_2(j)$ and $b_1(j) = b_2(j)$. Now remember that

$$j > \mathtt{diff}^{\mathcal{M}_1}(b_1, f_1(c_1)) \geq \mathtt{diff}^{\mathcal{M}_1}(b_2, f_1(c_2)),$$

hence

$$c_1(j) = b_1(j),\ c_2(j) = b_2(j)$$

$$c(j) = c_2(j) = b_2(j) = b_1(j) = c_1(j)$$

thus getting an absurdity.

(ii) Injectivity of μ_1 and μ_2 is immediate.

(iii) In order to prove that $|-|$ is preserved, it is sufficient to show that for every $a \in \mathtt{ARRAY}^{\mathcal{M}_1}$ and for all $k \in \mathtt{INDEX}^{\mathcal{M}}$, we have

$$\mu_1(a)(k) \neq \bot \leftrightarrow 0 \leq k \leq |a|^{\mathcal{M}_1}.$$

The only relevant case is when $k \in \mathtt{INDEX}^{\mathcal{M}_2} \setminus \mathtt{INDEX}^{\mathcal{A}}$ and (2$\star$) holds. In such a case, we have two possibilities:

$|b|^{\mathcal{M}_1} = |c|^{\mathcal{A}}$: in this case, since $b \sim^{\mathcal{M}_1} a$, we have $|a|^{\mathcal{M}_1} = |b|^{\mathcal{M}_1} = |c|^{\mathcal{A}} = |f_2(c)|^{\mathcal{M}_2}$ (f_2 is an embedding), thus getting what we need for k;

$|b|^{\mathcal{M}_1} \neq |c|^{\mathcal{A}}$: in this case $k > \mathtt{diff}^{\mathcal{M}_1}(b, f_1(c)) = \max\{|b|^{\mathcal{M}_1}, |c|^{\mathcal{A}}\}$ by Lemma 3.2. Since $a \sim b$ implies $|a| = |b|$, the definition of μ_1 produces $\mu_1(a)(k) = f_2(c)(k) = \bot$ (the last identity holds because $k > |c|^{\mathcal{A}}$), which is as desired because $k > |b|^{\mathcal{M}_1} = |a|^{\mathcal{M}_1}$ too.

(iv) The fact that rd and wr operations are preserved is easy (notice in particular that, if (2$\star$) holds for a via the pair (c, b), then the same pair guarantees (2$\star$) for arrays of the kind $wr(a, i, e)$).

(v) Again we limit to the case of μ_1 for symmetry. We need to show that for every $a_1, a_2 \in \mathtt{ARRAY}^{\mathcal{M}_1}$, we have $\mu_1(\mathtt{diff}^{\mathcal{M}_1}(a_1, a_2)) = \mathtt{diff}^{\mathcal{M}}(\mu_1(a_1), \mu_1(a_2))$. Notice that, if we call j the index $\mathtt{diff}(a_1, a_2) \in \mathtt{INDEX}^{\mathcal{M}_1}$, by definition of μ_1 on array applied to indexes in $\mathtt{INDEX}^{\mathcal{M}_1}$, we have that $\mu_1(a_1)(j) = a_1(j) \neq a_2(j) = \mu_1(a_2)(j)$. Hence, in order to conclude, it is sufficient to show that, given $k \in \mathtt{INDEX}^{\mathcal{M}}$ such that $k > \mathtt{diff}^{\mathcal{M}_1}(a_1, a_2)$, the equality $\mu_1(a_1)(k) = \mu_1(a_2)(k)$ holds. Notice first that we can always reduce to one of the following three cases

(a) $|a_1| < |a_2| = \mathtt{diff}^{\mathcal{M}_1}(a_1, a_2)$;
(b) $\mathtt{diff}^{\mathcal{M}_1}(a_1, a_2) < |a_1| = |a_2|$;
(c) $|a_1| = |a_2| = \mathtt{diff}^{\mathcal{M}_1}(a_1, a_2)$.

We now show that $\mu_1(a_1)(k) = \mu_1(a_2)(k)$.

- If $k \in \mathtt{INDEX}^{\mathcal{M}_1}$:

$$\mu_1(a_1)(k) = a_1(k) = a_2(k) = \mu_1(a_2)(k).$$

- If $k \notin \mathtt{INDEX}^{\mathcal{M}_1} \cup \mathtt{INDEX}^{\mathcal{M}_2}$, we analyze the three cases separately:

Case(a) : then $k \notin [0, |a_i|]$ for $i = 1, 2$. We have $\mu_1(a_1)(k) = \bot = \mu_1(a_2)(k)$;

Case (b) : we have

$$\mu_1(a_1)(k) = \bot \Leftrightarrow \mu_1(a_2)(k) = \bot$$

$$\mu_1(a_1)(k) = el \Leftrightarrow \mu_1(a_2)(k) = el;$$

Case (c) : similarly to (a), we have $k \notin [0, |a_i|]$ for $i = 1, 2$.

- If $k \in \mathtt{INDEX}^{\mathcal{M}_2} \setminus \mathtt{INDEX}^{\mathcal{A}}$ and (2$\star$) does not hold neither for a_1 nor for a_2, the argument is the same as in the previous case.
  Otherwise, suppose that (2$\star$) holds for, say, a_1 as witnessed by the pair (c_1, b_1). Then we get $\mu_1(a_1)(k) = f_2(c_1)(k)$. We prove that (2$\star$) holds for a_2 too and that we have $\mu_1(a_1)(k) = f_2(c_1)(k) = \mu_1(a_2)(k)$.
  Since b_1 and a_1 differ on finitely many indices inside $\mathcal{M}_1$, we can consider the finite sets
  $$I := \{i \in \mathtt{INDEX}^{\mathcal{M}_1} \mid \ b_1(i) \neq a_1(i),\ i > \mathtt{diff}^{\mathcal{M}_1}(a_1, a_2)\},$$
  $$E := \{b_1(i) \mid \ i \in I\} \subseteq \mathtt{ELEM}^{\mathcal{M}_1}$$
  and the array $\hat{b} := wr(a_2, I, E)$; for this array, we obviously have $a_2 \sim^{\mathcal{M}_1} \hat{b}$. If we also have
  $$\mathtt{diff}^{\mathcal{M}_1}(b_1, \hat{b}) \leq \mathtt{diff}^{\mathcal{M}_1}(a_1, a_2) \tag{26}$$
  then we get: $k > \mathtt{diff}^{\mathcal{M}_1}(a_1, a_2)$ e $k > \mathtt{diff}^{\mathcal{M}_1}(b_1, f_1(c_1))$ (the latter is from (2$\star$)). Hence:
  $$k > \max\{\mathtt{diff}^{\mathcal{M}_1}(a_1, a_2), \mathtt{diff}^{\mathcal{M}_1}(b_1, f_1(c_1))\} \geq$$
  $$\geq \max\{\mathtt{diff}^{\mathcal{M}_1}(b_1, \hat{b}), \mathtt{diff}^{\mathcal{M}_1}(b_1, f_1(c_1))\} \geq$$
  $$\geq \mathtt{diff}(\hat{b}, f_1(c_1))$$
  (the last disequality holds because of the 'triangular disequality' (16) of Lemma 3.2). Hence we obtain (2$\star$) for a_2 via the pair given by $c := c_1$ and $b := \hat{b}$ (because we have $a_2 \sim^{\mathcal{M}_1} \hat{b}$ and $k > \mathtt{diff}(\hat{b}, f_1(c_1))$) and consequently $\mu_1(a_2)(k) = f_2(c_1)(k)$.
  It remains to prove (26); to this aim, let us pick $j \in \mathtt{INDEX}^{\mathcal{M}_1}$ such that $j > \mathtt{diff}(a_1, a_2)$ and let us show that $b_1(j) = \hat{b}(j)$. If this is not the case, i.e., if $b_1(j) \neq \hat{b}(j)$, then according to the definition of $\hat{b}$ we have $\hat{b}(j) = a_2(j)$ and $j \notin I$. Hence $\hat{b}(j) = a_2(j) = a_1(j) = b_1(j)$ (the last identity holds because $j \notin I$ and $j > \mathtt{diff}(a_1, a_2)$), absurd.

(vi) In order to prove $\mu_1 \circ f_1 = \mu_2 \circ f_2$, let us consider $c \in \mathtt{ARRAY}^{\mathcal{A}}$; let us put $a_i = f_i(c)$ for $i = 1, 2$ and let us check that
$$\mu_1(a_1)(k) = \mu_2(a_2)(k)$$
holds for all $k \in \mathtt{INDEX}^{\mathcal{M}}$.

- Case $k \in \mathtt{INDEX}^{\mathcal{A}}$: we have
  $$\mu_1(a_1)(k) = a_1(k) = f_1(c)(k) = c(k)$$
  $$\mu_2(a_2)(k) = a_2(k) = f_2(c)(k) = c(k).$$
- Case $k \in \mathtt{INDEX}^{\mathcal{M}_1} \setminus \mathtt{INDEX}^{\mathcal{A}}$: clearly (2$\star$) holds for a_2 with $c := c$ and $b := a_2$, consequently
  $$\mu_1(a_1)(k) = a_1(k) = f_1(c)(k)$$
  $$\mu_2(a_2)(k) = f_1(c)(k).$$
- Case $k \in \mathtt{INDEX}^{\mathcal{M}_2} \setminus \mathtt{INDEX}^{\mathcal{A}}$: clearly (2$\star$) holds for a_1 with $c := c$ and $b := a_1$, consequently
  $$\mu_1(a_1)(k) = f_2(c)(k)$$
  $$\mu_2(a_2)(k) = a_2(k) = f_2(c)(k).$$
- Case $k \notin (\mathtt{INDEX}^{\mathcal{M}_1} \cup \mathtt{INDEX}^{\mathcal{M}_2})$ and $k \in [0, |c|]$: we have
  $$\mu_1(a_1)(k) = el = \mu_2(a_2)(k).$$
- se $k \notin (\mathtt{INDEX}^{\mathcal{M}_1} \cup \mathtt{INDEX}^{\mathcal{M}_2})$ and $k \notin [0, |c|]$: we have
  $$\mu_1(a_1)(k) = \bot = \mu_2(a_2)(k).$$

This completes our case analysis.

(vii) Here we assume that $\mathcal{M}_1, \mathcal{M}_2$ are models of $\mathcal{CARDC}(T_I)$; we need to show, e.g., that $\mu_1(\mathtt{Const}^{\mathcal{M}_1}(i))(k) = el$ for every k such that $0 \leq k \leq i$. Now, if $i \in \mathtt{INDEX}^{\mathcal{A}}$ this is obvious, because $\mathtt{Const}^{\mathcal{M}_1}(i) = f_1(\mathtt{Const}^{\mathcal{A}}(i))$. Hence suppose that $i \in \mathtt{INDEX}^{\mathcal{M}_1} \setminus \mathtt{INDEX}^{\mathcal{A}}$; the only possibly problematic case is when $k \in \mathtt{INDEX}^{\mathcal{M}_2} \setminus \mathtt{INDEX}^{\mathcal{A}}$ and (2$\star$) applies, as witnessed by a pair (b, c) for $\mathtt{Const}^{\mathcal{M}_1}(i)$. But we have $|b| = |\mathtt{Const}^{\mathcal{M}_1}(i)| = i$ and $|f_1(c)| \neq i$ (because $|f_1(c)| = |c| \in \mathtt{INDEX}^{\mathcal{A}}$). Then, according to (2$\star$) and recalling Lemma 3.2, we have $k > \mathtt{diff}^{\mathcal{M}_i}(b, f_i(c)) = \max\{|b|, |f_1(c)|\} = \max\{i, |f_1(c)|\} \geq i$, contradicting the choice of k.

□

5.2 The $\mathcal{CARDC}(T_I)$-amalgam is strong

We now prove the main result of the section, i.e., strong amalgamation for $\mathcal{CARDC}(T_I)$. Unfortunately, this property does not hold for $\mathcal{CARD}(T_I)$: as it is shown in the example below, we need constant arrays in the language (recall that strong amalgamation is equivalent to general quantifier-free interpolation, see Theorem 2.4(ii)).

Example 5.2. Consider the following two formulae (where P is a free predicate symbol):

$$(A) \quad |a| = 0 \wedge rd(a, 0) = e \wedge P(a)$$
$$(B) \quad |b| = 0 \wedge rd(b, 0) = e \wedge \neg P(b).$$

The conjunction $(A) \wedge (B)$ is inconsistent because a and b are in fact the same array (because of Axioms 7 and 10). However, the only common variable is e; to get the interpolant, we can use $P(wr(\mathtt{Const}(0), 0, e))$, but then it is clear that the language lacking constant arrays does not suffice.

To prove *strong* amalgamation for $\mathcal{CARDC}(T_I)$ we need a couple of lemmas.

Lemma 5.3. *Every model $\mathcal{M}$ of $\mathcal{CARDC}(T_I)$ can be embedded into a model $\mathcal{N}$ such that $\mathtt{INDEX}^{\mathcal{N}}$ is infinite.*

Proof. This is basically due to the fact that T_I is stably infinite. So let us first embed the T_I-reduct of $\mathcal{M}$ into an infinite model $\mathcal{A}$ of T_I. We define $\mathcal{N}$ as follows. We let $\mathtt{ELEM}^{\mathcal{N}}$ be equal to $\mathtt{ELEM}^{\mathcal{M}}$ and the T_I-reduct of $\mathtt{INDEX}^{\mathcal{N}}$ be equal to $\mathcal{A}$. We let $\mathtt{ARRAY}^{\mathcal{N}}$ be the set of positive support functions from $\mathtt{INDEX}^{\mathcal{N}}$ to $\mathtt{ELEM}^{\mathcal{N}}$ (the model so built will then be embedded into a full model of $\mathcal{CARDC}(T_I)$ using Theorem 4.4). We only need to define the embedding $\mu : \mathcal{M} \longrightarrow \mathcal{N}$. This embedding will be the identity for INDEX and ELEM sorts; for arrays, we let $\mu(a)(k)$ be equal to $a(k)$ for $k \in \mathtt{INDEX}^{\mathcal{M}}$ and for $k \notin \mathtt{INDEX}^{\mathcal{M}}$, we put $\mu(a)(k)$ equal to $el^{\mathcal{M}}$ or $\perp^{\mathcal{M}}$ depending whether we have $k \in [0, |a|]$ or not. The proof that μ preserves all operations is easy. □

Let us call an element $i \in \mathtt{INDEX}^{\mathcal{M}}$ of a model $\mathcal{M}$ of $\mathcal{CARDC}(T_I)$ *finite* iff the set $\{j \in \mathtt{INDEX}^{\mathcal{M}} \mid 0 \leq j \leq i\}$ is finite. $Fin(\mathcal{M})$ denotes the set of finite elements of $\mathcal{M}$.

Lemma 5.4. *Let $\mathcal{A}, \mathcal{M}$ be models of $\mathcal{CARDC}(T_I)$ and let $f : \mathcal{A} \longrightarrow \mathcal{M}$ be an embedding. Then there exist a third model $\mathcal{N}$ of $\mathcal{CARDC}(T_I)$ and an embedding $\nu : \mathcal{M} \to \mathcal{N}$, such that for every $a \in \mathtt{ARRAY}^{\mathcal{N}}$ with $|a|^{\mathcal{N}} \in \nu(f(\mathtt{INDEX}^{\mathcal{A}}))$ one of the following conditions hold:*

(1) there exists $c \in \mathtt{ARRAY}^{\mathcal{A}}$ such that $\nu(f(c)) \sim^{\mathcal{N}} a$;
(2) there exists $k_a \in \mathtt{INDEX}^{\mathcal{N}} \setminus \nu(\mathtt{INDEX}^{\mathcal{M}})$ with $a(k_a) \notin \nu(f(\mathtt{ELEM}^{\mathcal{A}}))$ such that for every $c \in \mathtt{ARRAY}^{\mathcal{A}}$ we have $a(k_a) \neq \nu(f(c))(k_a)$.

Proof. Because of Lemma 5.3, we can freely assume that $\mathtt{INDEX}^{\mathcal{M}}$ is infinite. If for all $a \in \mathtt{ARRAY}^{\mathcal{M}}$ whose length comes from $\mathcal{A}$, there is $c \in \mathtt{ARRAY}^{\mathcal{A}}$ such that $f(c) \sim^{\mathcal{M}} a$ then it is sufficient to take $\mathcal{N} = \mathcal{M}$ and the identity as ν. Otherwise, one take a well ordering of the arrays, apply the

construction below by tranfinite induction and repeat it ω-times. The union of the chain so built will have the required properties.

Let $a \in \mathtt{ARRAY}^{\mathcal{M}}$ be such that $|a|$ is from $\mathcal{A}$ (i.e. such that $|a| = f(i)$ for some $i \in \mathtt{INDEX}^{\mathcal{A}}$) and such that there does not exist $c \in \mathtt{ARRAY}^{\mathcal{A}}$ such that $f(c) \sim^{\mathcal{M}} a$. Then $|a|$ is not finite, because otherwise we would have that $a \sim^{\mathcal{M}} f(\mathtt{Const}(i))$.[8] Consider the diagram Δ of the T_I-reduct of $\mathcal{M}$ and let k_a a fresh constant; the set

$$\begin{aligned} \Delta' := \Delta \cup \{i < k_a \mid i \in Fin(\mathcal{M})\} \cup \\ \cup \{i > k_a \mid i \in \mathtt{INDEX}^{\mathcal{M}} \setminus Fin(\mathcal{M})\}. \end{aligned} \tag{27}$$

is consistent. Suppose that Δ' is inconsistent. By compactness, we would have that there exists a finite subset Δ'_0 of Δ' which is inconsistent too. Δ'_0 would involve finitely many finite indexes $i_1 < \cdots < i_n$ and finitely many infinite indexes $j_1 < \cdots < j_m$ and a finite subset Δ_0 of Δ.

If $m = 0$, since $\mathtt{INDEX}^{\mathcal{M}}$ is infinite, there exist an element $i' \in \mathtt{INDEX}^{\mathcal{M}}$ large enough, so as to get $i_1 < \cdots < i_n < i'$ in $\mathcal{M}$. If $m > 0$, then already in $\mathtt{INDEX}^{\mathcal{M}}$ there exists an element i' such that $i_1 < \cdots < i_n < i'$ and such that $i' < j_1 < \cdots < j_m$ hold in $\mathcal{M}$, otherwise j_1 would be a finite index. This element i' can interpret the constant k_a. In both cases, we conclude that Δ'_0 would be consistent, which is a contradiction.

By Robinson Diagram Lemma, Δ has a model $\mathcal{B}$ extending the T_I-reduct of $\mathcal{M}$.

We let now $\mathtt{ELEM}^{\mathcal{N}} = \mathtt{ELEM}^{\mathcal{M}}$, $\mathtt{INDEX}^{\mathcal{N}} = \mathtt{INDEX}^{\mathcal{B}}$ and we let $\mathtt{ARRAY}^{\mathcal{N}}$ to be the set of positive-support functions from $\mathtt{INDEX}^{\mathcal{N}}$ into $\mathtt{ELEM}^{\mathcal{N}}$ (then, in view of Theorem 4.4, $\mathcal{N}$ can be embedded into a full model of $\mathcal{CARDC}(T_I)$). This model contains an element k_a such that for all $i \in \mathtt{INDEX}^{\mathcal{M}}$ we have that $k_a \neq i$ and $i < k_a$ iff $i \in Fin(\mathcal{M})$. In particular, $k_a < |a|$ (because $|a|$ is infinite).

Thanks to Lemma 4.3, we can freely suppose that $\mathtt{ELEM}^{\mathcal{N}} = \mathtt{ELEM}^{\mathcal{M}}$ has an element e not belonging to $f(\mathtt{ELEM}^{\mathcal{A}})$ (in particular $e \neq f(el^{\mathcal{A}}) = el^{\mathcal{N}} = el^{\mathcal{M}}$). We build μ as required by the statement of the lemma (more precisely, the ν required by the lemma will be a chain unions of the μ's built at each transfinite step as shown below). The INDEX- and ELEM-components of μ will be inclusions. We define $\mu(b)(k)$ for all $b \in \mathtt{ARRAY}^{\mathcal{M}}$. If $k \in \mathtt{INDEX}^{\mathcal{M}}$, we obviously put $\mu(b)(k) = b(k)$; in the other cases, the definition is as follows:

(1) if $b \nsim^{\mathcal{M}} a$, then $\mu(b)(k) = el^{\mathcal{N}}$ or $\mu(b)(k) = \bot^{\mathcal{N}}$, depending on whether $k \in [0, |b|]$ or not;
(2) if $b \sim^{\mathcal{M}} a$ and $k \neq k_a$, then again $\mu(b)(k) = el^{\mathcal{N}}$ or $\mu(b)(k) = \bot^{\mathcal{M}}$, depending on whether $k \in [0, |b|]$ or not;
(3) if $b \sim^{\mathcal{M}} a$ and $k = k_a$, then $\mu(b)(k) = e$.

We need to show that μ preserves rd, wr, $|-|$, constant arrays and $\mathtt{diff}$. Preservation of rd, wr, $|-|$ are easy; constant arrays are preserved, because we cannot have $\mathtt{Const}^{\mathcal{M}}(i) \sim^{\mathcal{M}} a$, otherwise $|a| = i$, which cannot be because $|a|$ is an element from $\mathtt{INDEX}^{\mathcal{A}}$ by hypothesis, so that we would have $f(\mathtt{Const}^{\mathcal{A}}(i)) = \mathtt{Const}^{\mathcal{M}}(i) \sim^{\mathcal{M}} a$, contradiction. For preservation of $\mathtt{diff}$, the problematic case would be the case in which we have $k_a > \mathtt{diff}(b_1, b_2)$, $b_1 \sim^{\mathcal{M}} a$ and $b_2 \nsim^{\mathcal{M}} a$. However, this is impossible because $\mathtt{diff}(b_1, b_2) \in \mathtt{INDEX}^{\mathcal{M}}$ and the fact that we have $k_a > \mathtt{diff}(b_1, b_2)$ implies that $\mathtt{diff}(b_1, b_2)$ is finite, which would entail either $b_1 \sim^{\mathcal{M}} b_2$ or $|b_1| \neq |b_2|$: in the former case, we would have $b_2 \sim^{\mathcal{M}} a$ and in the latter $k_a > \mathtt{diff}(b_1, b_2) = \max(|b_1|, |b_2|) = \max(|a|, |b_2|) \geq |a|$.

We finally notice that k_a satisfies the requirements of the lemma. First, $k_a \notin \mu(\mathtt{INDEX}^{\mathcal{M}})$ and $\mu(a)(k_a) = e \notin \mu(f(\mathtt{ELEM}^{\mathcal{A}}))$ hold by construction. Moreover, since for every $c \in \mathtt{ARRAY}^{\mathcal{A}}$ we have $\mu(f(c))(k_a) = el^{\mathcal{N}}$ or $\mu(f(c))(k_a) = \bot^{\mathcal{N}}$ (depending whether $k_a \in [0, |c|]$ holds or not), in any case we see that $\mu(f(c))(k_a) \neq \mu(a)(k_a)$. □

[8] Notice that this is the only argument in the whole strong amalgamation proof requiring the fact that we have Const in the language.

Theorem 5.5. *$\mathcal{CARDC}(T_I)$ enjoys the strong amalgamation property.*

Proof. We keep the same notation and construction as in the proof of Theorem 5.1. However, thanks to Lemma 5.4 we can now suppose (for $i = 1, 2$) that all arrays $a \in \mathtt{ARRAY}^{\mathcal{M}_i}$ whose length belongs to $\mathtt{INDEX}^{\mathcal{A}}$ are such that one of the following two conditions are satisfied:

(1) there exists $c \in \mathtt{ARRAY}^{\mathcal{A}}$ with $f_i(c) \sim^{\mathcal{M}_i} a$;
(2) there exists $k_a \in \mathtt{INDEX}^{\mathcal{M}_i} \setminus \mathtt{INDEX}^{\mathcal{A}}$ such that $a(k_a)$ is an element from $\mathtt{ELEM}^{\mathcal{M}_i} \setminus \mathtt{ELEM}^{\mathcal{A}}$ different from all the $f_i(c)(k_a)$, varying $c \in \mathtt{ARRAY}^{\mathcal{A}}$.

Let $a_i \in \mathtt{ARRAY}^{\mathcal{M}_i}$ $(i = 1, 2)$ be such that $\forall k \in \mathtt{INDEX}^{\mathcal{M}}$ we have

$$\mu_1(a_1)(k) = \mu_2(a_2)(k) \tag{28}$$

Notice that, since T_I has the strong amalgamation property and the μ_i preserve length, this can only happen if $|a_1| = |a_2|$ belongs to $\mathtt{INDEX}^{\mathcal{A}}$. We look for some $c \in \mathtt{ARRAY}^{\mathcal{A}}$ such that $a_1 = f_1(c)$; since μ_2 is injective this would entail $a_2 = f_2(c)$ because

$$\mu_2(a_2) = \mu_1(a_1) = \mu_1(f_1(c)) = \mu_2(f_2(c)),$$

implying that $\hat{\mathcal{M}}$ is a strong amalgam, as requested.

We separate two cases: (i) one of the arrays a_1, a_2 satisfy the above condition 2; (ii) both arrays a_1, a_2 satisfy the above condition 1.

(i) We show that this case is impossible. Suppose, e.g., that a_1 satisfies condition 2 in $\mathcal{M}_1$. Then there exists an index k_{a_1} in $\mathtt{INDEX}^{\mathcal{M}_1} \setminus \mathtt{INDEX}^{\mathcal{A}}$ such that $a(k_{a_1})$ is an element from $\mathtt{ELEM}^{\mathcal{M}_1} \setminus \mathtt{ELEM}^{\mathcal{A}}$ which is different from all the $f_1(c)(k_{a_1})$, varying $c \in \mathtt{ARRAY}^{\mathcal{A}}$. Since we must have $\mu_1(a_1)(k_{a_1}) = \mu_2(a_2)(k_{a_1})$ and $\mu_1(a_1)(k_{a_1})$ does not belong to $\mathtt{ELEM}^{\mathcal{M}_2}$ (recall that $\mathtt{ELEM}^{\mathcal{M}_1} \cap \mathtt{ELEM}^{\mathcal{M}_2} = \mathtt{ELEM}^{\mathcal{A}}$), the value of a_2 for the index k_{a_1} $(0 < k_{a_1} < |a_1| = |a_2|)$ is built according to the rule (2$\star$), because otherwise $\mu_2(a_2)(k_{a_1})$ would be equal to some element in $\mathtt{ELEM}^{\mathcal{M}_2}$. Let (c, b) the pair such that

$$c \in \mathtt{ARRAY}^{\mathcal{A}}, b \in \mathtt{ARRAY}^{\mathcal{M}_2},\ b \sim^{\mathcal{M}_2} a_2,\ k_{a_1} > \mathtt{diff}^{\mathcal{M}_2}(b, f_2(c))$$
$$\mu_2(a_2)(k_{a_1}) = f_1(c)(k_{a_1}).$$

Then we have

$$\mu_1(a_1)(k_{a_1}) = a_1(k_{a_1}) \neq f_1(c)(k_{a_1}) = \mu_2(a_2)(k_{a_1})$$

contradiction.

(ii) Hence we can have $\mu_1(a_1) = \mu_2(a_2)$ only when both a_1, a_2 satisfy condition 1 above. Let us call $c_i \in \mathtt{ARRAY}^{\mathcal{A}}$ $(i = 1, 2)$ the arrays such that $f_i(c_i) \sim^{\mathcal{M}_i} a_i$. Then, the pair $(c_1, f_1(c_1))$ witnesses (2$\star$) for a_1 and for every positive [9] index $k \in \mathtt{INDEX}^{\mathcal{M}_1} \setminus \mathtt{INDEX}^{\mathcal{A}}$ (and similarly for a_2). We look for $c \in \mathtt{ARRAY}^{\mathcal{A}}$ such that $f_1(c) = a_1$. Let us consider the following relations coming from (28) and from the definition of μ_i:

$$\begin{aligned} \forall k \in \mathtt{INDEX}^{\mathcal{A}},\ a_1(k) = a_2(k) \\ \forall k \in \mathtt{INDEX}^{\mathcal{M}_1} \setminus \mathtt{INDEX}^{\mathcal{A}},\ a_1(k) = f_1(c_2)(k) \\ \forall k \in \mathtt{INDEX}^{\mathcal{M}_2} \setminus \mathtt{INDEX}^{\mathcal{A}},\ f_2(c_1)(k) = a_2(k). \end{aligned} \tag{29}$$

where we used that $\mu_2(a_2)(k) = f_1(c_2)(k)$ if $k \in \mathtt{INDEX}^{\mathcal{M}_1} \setminus \mathtt{INDEX}^{\mathcal{A}}$, and $\mu_1(a_1)(k) = f_2(c_1)(k)$ if $k \in \mathtt{INDEX}^{\mathcal{M}_2} \setminus \mathtt{INDEX}^{\mathcal{A}}$.

Let us now consider the sets (they are finite because $f_2(c_2) \sim^{\mathcal{M}_2} a_2$)

$$J = \{j \in \mathtt{INDEX}^{\mathcal{A}} \mid c_2(j) \neq a_2(j)\} \subseteq \mathtt{INDEX}^{\mathcal{A}}$$

[9] Notice that if $k \in \mathtt{INDEX}^{\mathcal{M}_1} \setminus \mathtt{INDEX}^{\mathcal{A}}$ is positive, then $k > \mathtt{diff}(f_1(c), f_1(c)) = 0$.

$$E = \{a_2(j) \mid \ j \in J\} \subseteq \mathtt{ELEM}^{\mathcal{A}},$$

and let us put $c = wr(c_2, J, E)$; we check that c is such that $f_1(c) = a_1$.

- If $k \in \mathtt{INDEX}^{\mathcal{A}}$:

$$f_1(c)(k) = c(k) = wr(c_2, J, E)(k) = a_2(k) = a_1(k)$$

because f_1 preserves rd, by the definition of J and because of the equalities (29);

- If $k \in \mathtt{INDEX}^{\mathcal{M}_1} \setminus \mathtt{INDEX}^{\mathcal{A}}$:

$$f_1(c)(k) = f_1(wr(c_2, J, E))(k) = wr(f_1(c_2), J, E)(k) = f_1(c_2)(k) = a_1(k)$$

by the definition of c, the fact that f_1 is an embedding, because $J \subseteq \mathtt{INDEX}^{\mathcal{A}}$ (hence $k \notin J$) and because of the equalities (29).

□

Strong amalgamation corresponds to general quantifier-free interpolation (Theorem 2.4), hence we obtain that:

Corollary 5.6. *The theory $\mathcal{CARDC}(T_I)$ has the general quantifier-free interpolation property.*

6 SATISFIABILITY

We now address the problem of checking satisfiability of quantifier-free formulæ in $\mathcal{CARD}(T_I)$ and $\mathcal{CARDC}(T_I)$. Decidability of the $SMT(\mathcal{CARD}(T_I))$- and $SMT(\mathcal{CARDC}(T_I))$-problems, at least in the relevant case where T_I is any fragment of Presburger arithmetics, can be solved by reduction to the satisfiability problem for the so-called 'array-property fragment' of [5]: the reduction can be obtained by eliminating the wr, $|-|$, $\mathtt{diff}$ and $\mathtt{Const}$ symbols in favor of universally quantified formulæ belonging to that fragment (see Lemmas 3.3,3.4,3.5). However, we now supply a direct decision procedure for quantifier-free formulæ, since this will be useful for the interpolation algorithm in Section 7.

A *flat* literal L is a formula of the kind $x_0 = f(x_1, \ldots, x_n)$ or $x_1 \neq x_2$ or $R(x_1, \ldots, x_n)$ or $\neg R(x_1, \ldots, x_n)$, where the x_i are variables, R is a relation symbol, and f is a function symbol. If $\mathcal{I}$ is a set of T_I-terms, an *$\mathcal{I}$-instance* of a universal formula of the kind $\forall i\, \phi$ is a formula $\phi(t/i)$ for some $t \in \mathcal{I}$.

Definition 6.1. A pair of sets of quantifier-free $\mathcal{CARD}(T_I)$-formulæ $\Phi = (\Phi_1, \Phi_2)$ is a *separated pair* iff

(1) Φ_1 contains equalities of the form $|a| = i, \mathtt{diff}_k(a, b) = i$ and $a = wr(b, i, e)$; moreover if it contains the equality $\mathtt{diff}_k(a, b) = i$, it must also contain an equality of the form $\mathtt{diff}_l(a, b) = j$ for every $l < k$; finally, if $\Phi_1 \cup \Phi_2$ contains occurrences of an array variable a, Φ_1 must contain also an equality of the form $|a| = i$;

(2) Φ_2 contains Boolean combinations of T_I-atoms and of atoms of the forms:

$$rd(a, i) = rd(b, j),\ rd(a, i) = e,\ e_1 = e_2, \tag{30}$$

where a, b, i, j, e, e_1, e_2 are variables or constants of the appropriate sorts.

Φ is said to be finite iff Φ_1 and Φ_2 are both finite.

Example 6.2. The pair $\Phi = (\Phi_1, \Phi_2)$ given by

$$\Phi_1 = \{c_1 = wr(c_2, i, e), |c_1| = l_{c_1}, |c_2| = l_{c_2}\}, \qquad \Phi_2 = \{l_{c_1} \neq l_{c_2}\} \tag{31}$$

fulfills the conditions for being a separated pair.

Notably, in a separated pair $\Phi = (\Phi_1, \Phi_2)$, if we introduce a unary function symbol f_a for every array variable a and rewrite $rd(a, i)$ as $f_a(i)$, it turns out that *the formulæ from Φ_2 can be seen as formulæ of the combined theory $T_I \cup \mathcal{EUF}$*. $T_I \cup \mathcal{EUF}$ is decidable in its quantifier-free fragment and admits quantifier-free interpolation because T_I is an index theory (see Nelson-Oppen results [26] and Theorems 2.4,2.5): we adopt a hierarchical approach (similarly to [31, 33]) and *we rely on satisfiability and interpolation algorithms for such a theory as black boxes.*

In order to be able to apply such hierarchical approach to satisfiability problems, we first reduce to satisfiability problems for separated pairs (Lemma 6.4 below). Then, given a separated pair $\Phi = (\Phi_1, \Phi_2)$, we transfer to Φ_2 some information that is hidden in Φ_1: this is the information stored in the universally quantified formulae (18),(19),(22). In principle, one should instantiate the universally quantified variable appearing in those formulae with all possible ground terms that can be built up using the index variables occurring in the current constraint. Unfortunately, there are infinitely many such terms; in the interpolation algorithm of our previous paper [16], we devised incremental instantiations: first we instantiate with terms of complexity 0, then with terms of complexity 1, then with terms of complexity 2, etc. (the complexity of a term can be defined as the maximum nestings of function symbols occurring in it). We completely avoid these incremental instantiations in the current paper, not only in the satisfiability algorithm, but also in the interpolation algorithm of next Section (this is the substantial improvement over [16] from the computational point of view). We first define our instantiations with respect to an arbitrary set of terms I:

Definition 6.3. Let $\mathcal{I}$ be a set of T_I-terms and let $\Phi = (\Phi_1, \Phi_2)$ be a separated pair; $\Phi(\mathcal{I}) = (\Phi_1(\mathcal{I}), \Phi_2(\mathcal{I}))$ is the smallest separated pair satisfying the following conditions:

- $\Phi_1(\mathcal{I})$ is equal to Φ_1[10] and $\Phi_2(\mathcal{I})$ contains Φ_2;
- if Φ_1 contains the atom $a = wr(b, i, e)$ then $\Phi_2(\mathcal{I})$ contains *all the $\mathcal{I}$-instances of the formulae* (18) (with the terms $|a|, |b|$ replaced by the index constants i, j such that $|a| = i, |b| = j \in \Phi_1$, respectively);
- if Φ_1 contains the atom $|a| = i$, then $\Phi_2(\mathcal{I})$ contains all the $\mathcal{I}$-instances of the formulae (19);
- if Φ_1 contains the conjunction $\bigwedge_{i=1}^{l} \mathtt{diff}_i(a, b) = k_l$, then $\Phi_2(\mathcal{I})$ contains the formulae (22) (with the terms $|a|, |b|$ replaced by the index constants i, j such that $|a| = i, |b| = j \in \Phi_1$, respectively).

A separated pair Φ is *0-instantiated* iff $\Phi = \Phi(\mathcal{I})$, where $\mathcal{I}$ is the set of index variables occurring in Φ.

We say that a separated pair $\Phi = (\Phi_1, \Phi_2)$ is is $\mathcal{CARD}(T_I)$-satisfiable iff so it is the formula $\bigwedge \Phi_1 \wedge \bigwedge \Phi_2$.[11]

Lemma 6.4. *Let ϕ be a quantifier-free formula; then it is possible to compute in linear time a finite separation pair $\Phi = (\Phi_1, \Phi_2)$ such that ϕ is $\mathcal{CARD}(T_I)$-satisfiable iff Φ is satisfiable.*

Proof. We first flatten all atoms from ϕ by repeatedly abstracting out subterms (to abstract out a subterm t, we introduce a fresh variable x and update ϕ to $x = t \wedge \phi(x/t)$); then we remove all atoms of the kind $a = b$ occurring in ϕ by replacing them by the equivalent formula (17), namely

$$\mathtt{diff}(a, b) = 0 \wedge rd(a, 0) = rd(b, 0)\ .$$

Then we abstract out all terms of the kind $wr(b, i, e), \mathtt{diff}(a, b)$ and $|a|$, so that ϕ has now the form $\Phi_1 \wedge \Phi_2$, where Φ_2 does not contain $wr, \mathtt{diff}, |-|$-symbols and Φ_1 is a conjunction of atoms of the

[10] This is because only Φ_2 needs to be instantiated.

[11] In case Φ_1, Φ_2 are not finite, this means that all formulae in $\Phi_1 \cup \Phi_2$ are simultaneously satisfiable. Notice however that in all our concrete applications, Φ_1 and Φ_2 are always finite.

form $a = wr(b, i, e), i = \mathtt{diff}(a, b), j = |a|$. Finally, we add to Φ_1 the missing atoms of the kind $|a| = i$ required by Definition 6.1.[12] □

Example 6.5. If we apply the procedure of Lemma 6.4 to the formula

$$c_1 = wr(c_2, i, e) \wedge |c_1| \neq |c_2|$$

we obtain the separated pair (31) of Example 6.2.

Next we show that 0-instantiations suffice:

Lemma 6.6. *The following conditions are equivalent for a finite 0-instantiated separated pair* $\Phi = (\Phi_1, \Phi_2)$*:*

(i) Φ *is* $\mathcal{CARD}(T_I)$*-satisfiable;*

(ii) $\bigwedge \Phi_2$ *is* $T_I \cup \mathcal{EUF}$*-satisfiable.*

Proof. The meaning of the lemma is that the role of Φ_1 is just that of contributing instances to 0-instantiation (such a role is exhausted when 0-instantiation is done).

$(i) \Rightarrow (ii)$ is clear.

To prove $(ii) \Rightarrow (i)$, let $\mathcal{A}$ be the model witnessing the satisfiability of $\bigwedge \Phi_2$ in $T_I \cup \mathcal{EUF}$ and let $\mathcal{I}$ be the set of all index variables occurring in $\Phi_1 \cup \Phi_2$. According to the definition of a separated pair, for every array variable a occurring in $\Phi_1 \cup \Phi_2$ there is an index variable l_a such that:

$$\mathcal{A} \models f_a(i) \neq \bot \leftrightarrow 0 \leq i \leq l_a.$$

for all $i \in \mathcal{I}$ (here f_a is the unary function symbol replacing a in $T_I \cup \mathcal{EUF}$, recall that formulae (19) have been instantiated with index variables according to Definition 6.3).

The *standardization* $\mathcal{A}'$ of $\mathcal{A}$ is the $T_I \cup \mathcal{EUF}$-model obtained from $\mathcal{A}$ by modifying the values $a(k)$ (for all array variables a occurring in Φ_2 and for all indexes $k \in \mathtt{INDEX}^{\mathcal{A}}$ different from the elements assigned in $\mathcal{A}$ to the variables in $\mathcal{I}$) in such a way that we have

$$\mathcal{A}' \models f_a(k) = \bot \leftrightarrow (l_a < k \vee k < 0),$$

$$\mathcal{A}' \models f_a(k) = el \leftrightarrow 0 \leq k \leq l_a\ .$$

The standardization $\mathcal{A}'$ of $\mathcal{A}$ is still a model of $\bigwedge \Phi_2$. For instance, suppose that $\bigwedge \Phi_2$ contains a literal of the type $rd(a, i) = e$: since k is different from the elements assigned in $\mathcal{A}$ to the variables in $\mathcal{I}$, and i is in $\mathcal{I}$, the changes above in the definition of f_a introduced in $\mathcal{A}'$ do not interfere with $rd(a, i) = e$, hence it is still valid in $\mathcal{A}'$. The other cases are analogous.

However, in $\mathcal{A}'$ now also the formulae (19), (22) and (18) hold: this is because the $\mathcal{I}$-instantiations of the universal index quantifiers occurring in such formulae were taken care in $\mathcal{A}$ and their truth value is not modified passing to $\mathcal{A}'$, whereas the construction of $\mathcal{A}'$ takes care of the instantiations outside $\mathcal{I}$.

Let us now define an $\mathcal{CARD}(T_I)$-model $\mathcal{M}$ satisfying Φ. We first build a structure $\mathcal{N}$ where $\mathtt{diff}$ may not be totally defined. We let $\mathtt{INDEX}^{\mathcal{N}} = \mathtt{INDEX}^{\mathcal{A}'}$ and $\mathtt{ELEM}^{\mathcal{N}} = \mathtt{ELEM}^{\mathcal{A}'}$; we take as $\mathtt{ARRAY}^{\mathcal{N}}$ the set of all positive-support functions from $\mathtt{INDEX}^{\mathcal{N}}$ into $\mathtt{ELEM}^{\mathcal{N}}$: this includes all functions of the form f_a. In addition, if $\bigwedge_{n=1}^{l} \mathtt{diff}_n(a_1, a_2) = k_n \in \Phi_1$, then the related iterated maxdiff's are defined in $\mathcal{N}$ and we have $\mathcal{N} \models \bigwedge_{n=1}^{l} \mathtt{diff}_n(a_1, a_2) = k_n$ by the above construction. Thus Φ holds in $\mathcal{N}$ and in order to obtain our final $\mathcal{M}$ we only need to apply Theorem 4.4. □

From Lemmas 6.4 and 6.6, we get the following result:

Theorem 6.7. *The SMT*$(\mathcal{CARD}(T_I))$ *problem is decidable for every index theory* T_I*.*

[12]The transformation of Lemma 6.4 does not introduce in Φ_1 any formula of the kind $\mathtt{diff}_n(a, b) = k_n$ (for $n > 1$). These formulæ will however be introduced by the Step 1 of the interpolation algorithm of Section 7.

Example 6.8. Let us show that

$$c_1 = wr(c_2, i, e) \wedge |c_1| \neq |c_2| \tag{32}$$

is not $\mathcal{CARD}(T_I)$-satisfiable. This is obvious if one considers our axiom (3), however we want to obtain this result from the above algorithm making a reduction to $T_I \cup \mathcal{EUF}$-satisfiability. As we know from Example 6.5, a separated pair $\Phi = (\Phi_1, \Phi_2)$ equisatisfiable to (32) is the separated pair (31) of Example 6.2, namely:

$$\Phi_1 = \{c_1 = wr(c_2, i, e), |c_1| = l_{c_1}, |c_2| = l_{c_2}\} \quad \Phi_2 = \{l_{c_1} \neq l_{c_2}\}\ .$$

For 0-instantiations, we have to consider 3 index variables (namely i, l_{c_1}, l_{c_2}). Thus, instantiating (18) gives

$$\begin{aligned}
(e \neq \bot \wedge 0 \leq i \leq l_{c_2}) &\rightarrow rd(c_1, i) = e \\
(i < 0 \vee i > l_{c_2} \vee e = \bot) &\rightarrow rd(c_1, i) = rd(c_2, i) \\
i \neq i &\rightarrow rd(c_1, i) = rd(c_2, i) \\
l_{c_1} \neq i &\rightarrow rd(c_1, l_{c_1}) = rd(c_2, l_{c_1}) \\
l_{c_2} \neq i &\rightarrow rd(c_1, l_{c_2}) = rd(c_2, l_{c_2})
\end{aligned} \tag{33}$$

whereas instantiating (19) gives

$$\begin{aligned}
l_{c_1} \geq 0 \ \wedge\ (rd(c_1, i) \neq \bot \leftrightarrow 0 \leq i \leq l_{c_1}) \\
l_{c_1} \geq 0 \ \wedge\ (rd(c_1, l_{c_1}) \neq \bot \leftrightarrow 0 \leq l_{c_1} \leq l_{c_1}) \\
l_{c_1} \geq 0 \ \wedge\ (rd(c_1, l_{c_2}) \neq \bot \leftrightarrow 0 \leq l_{c_2} \leq l_{c_1}) \\
l_{c_2} \geq 0 \ \wedge\ (rd(c_2, i) \neq \bot \leftrightarrow 0 \leq i \leq l_{c_2}) \\
l_{c_2} \geq 0 \ \wedge\ (rd(c_2, l_{c_1}) \neq \bot \leftrightarrow 0 \leq l_{c_1} \leq l_{c_2}) \\
l_{c_2} \geq 0 \ \wedge\ (rd(c_2, l_{c_2}) \neq \bot \leftrightarrow 0 \leq l_{c_2} \leq l_{c_2})
\end{aligned} \tag{34}$$

We can now see that formulae (33)-(34) are indeed $T_I \cup \mathcal{EUF}$-inconsistent with $l_{c_1} \neq l_{c_2}$ (a direct check is slightly laborious, but an SMT-solver discharges instantaneously this proof obligation).

Regarding complexity for the $SMT(\mathcal{CARD}(T_I))$ problem, notice that the satisfiability of the quantifier-free fragment of common index theories (like $\mathcal{IDL}$, $\mathcal{LIA}$, $\mathcal{LRA}$) is decidable in NP; hence, for such index theories, an NP bound for our $SMT(\mathcal{CARD}(T_I))$-problems is easily obtained, because 0-instantiation is clearly finite and polynomial (all strings of universal quantifiers to be instantiated have length one).

The above decidability and complexity results apply also to $\mathcal{CARDC}(T_I)$: one only simply has to allow the Φ_1-component of a separation pair to contain also atoms of the form $\mathtt{Const}(i) = a$ and Definition 6.3 to require that $\Phi_2(\mathcal{I})$ contains all the $\mathcal{I}$-instances of the formulae (20) in case $Const(i) = a \in \Phi_1$.

7 THE INTERPOLATION ALGORITHM

Since amalgamation is equivalent to quantifier-free interpolation for universal theories such as $\mathcal{CARD}(T_I)$ and $\mathcal{CARDC}(T_I)$ (thanks to Theorem 2.4), Theorem 5.1 guarantees that $\mathcal{CARD}(T_I)$ and $\mathcal{CARDC}(T_I)$ admit quantifier-free interpolation. However, the proof of Theorem 5.1 is not constructive: hence, in order to compute an interpolant for an unsatisfiable conjunction like $\psi(\underline{x}, \underline{y}) \wedge \phi(\underline{y}, \underline{z})$, one needs in principle to enumerate all quantifier-free formulæ $\theta(\underline{y})$ that are consequences of ϕ and are inconsistent with ψ. Since the quantifier-free fragments of $\mathcal{CARD}(T_I)$ and $\mathcal{CARDC}(T_I)$ are decidable, this is an effective procedure and, considering the fact that interpolants of jointly unsatisfiable pairs of formulæ exist, it also terminates. However, this type of algorithm is not practical. In this section, we provide a better and more practical algorithm that

relies on a hierarchical reduction to $T_I \cup \mathcal{EUF}$. Our algorithm works for $\mathcal{CARD}(T_I)$ only; for $\mathcal{CARDC}(T_I)$, we make some comments in Section 8.

Our problem is the following: given two quantifier-free formulae A^0 and B^0 such that $A^0 \wedge B^0$ is not satisfiable (modulo $\mathcal{CARD}(T_I)$), to compute a quantifier-free formula C such that

(i) $\mathcal{CARD}(T_I) \models A^0 \rightarrow C$;
(ii) $\mathcal{CARD}(T_I) \models C \wedge B^0 \rightarrow \bot$;
(iii) C contains only the variables (of sort INDEX, ARRAY, ELEM) which occur both in A^0 and in B^0.

Below, we work with ground formulae over signatures expanded with free constants instead of quantifier-free formulae. We use letters $A, B, \ldots$ for finite sets of ground formulae; the logical reading of a set of formulae is the conjunction of its elements. For a signature Σ and a set A of formulae, Σ^A denotes the signature Σ expanded with the free constants occurring in A. Let A and B be two finite sets of ground formulae in the signatures Σ^A and Σ^B, resp., and $\Sigma^C := \Sigma^A \cap \Sigma^B$. We 'color' a term, a literal, or a formula φ by calling it:

- *AB-common* iff it is defined over Σ^C;
- *A-local* (resp. *B-local*) if it is defined over Σ^A (resp. Σ^B);
- *A-strict* (resp. *B-strict*) iff it is A-local (resp. B-local) but not AB-common;
- *strict* if it is either A-strict or B-strict.

There are a number of manipulations that can be freely applied to a jointly unsatisfiable pair A, B without compromising the possibility of extracting an interpolant out of them. A list of such manipulations (called 'metarules') is supplied in [7], [8]. Here we need to introduce only some of them:

(i) we can add to A an A-local quantifier-free formula entailed by A (similarly we can add to B a B-local quantifier-free formula entailed by B): the interpolant computed after such a transformation is trivially an interpolant for the original pair too;
(ii) we can pick an A-local term t and a fresh constant x (to be considered A-strict from now on) and add to A the equality $x = t$: again, the interpolant computed after such a transformation is trivially an interpolant for the original pair too (the same observation extends to B);
(iii) we can pick an AB-common term t and a fresh constant x (to be considered AB-common from now on) and add to both A and B the equality $x = t$: in this case, if θ is the interpolant computed after such a transformation, then $\theta(t/x)$ is an interpolant for the original pair.

We shall often apply the above metarules (i)-(ii)-(iii) in the sequel.

7.1 The Algorithm

We reduce the problem of finding an interpolant of an unsatifiable pair (A^0, B^0) to an analogous polynomial size problem in the weaker theory $T_I \cup \mathcal{EUF}$.

Our unsatisfiable pair (A^0, B^0) needs to be *preprocessed*. Using the procedure in the proof of Lemma 6.4, we can suppose that both A^0 and B^0 are given in the form of finite separated pairs. In fact, the procedure of Lemma 6.4 just introduces constants in order to explicitly name terms, so that it fits within the above explained remarks (see metarules (ii)-(iii)). The newly introduced constants are colored A-strict, B-strict or AB-common depending on the color of the terms they name. Notice that because of this preprocessing, for every A-strict (resp. B-strict, AB-common) array constant a, in A^0 (resp. B_0, $A^0 \cap B^0$) there is an atom of the kind $|a| = l_a$.

To sum up, A^0 is of the form $\bigwedge A^0_1 \wedge \bigwedge A^0_2$ and B^0 is of the form $\bigwedge B^0_1 \wedge \bigwedge B^0_2$, for separated pairs (A^0_1, A^0_2) and (B^0_1, B^0_2).

Our interpolation algorithm consists of *three transformation steps* (all of them fit our metarules (i)-(ii)-(iii)). We let N_A (resp. N_B) be the number of A-local (resp. B-local) index constants occurring within a wr symbol in A^0 (resp. B^0); we let also N be equal to $1 + max(N_A, N_B)$.

Step 1. This transformation must be applied for every pair of distinct AB-common ARRAY-constants c_1, c_2. The transformation picks fresh INDEX constants $k_1, \ldots, k_N$ (to be colored AB-common) and adds the atoms $\mathtt{diff}_n(c_1, c_2) = k_n$ (for all $n = 1, \ldots, N$) to both sets A_1 and B_1. This transformation fits metarule (iii).

Step 2. We apply 0-instantiation, that is we replace A with $A(\mathcal{I}_A)$ and B with $B(\mathcal{I}_B)$, where $\mathcal{I}_A$ is the set of A-local index constants and $\mathcal{B}$ is the set of B-local index constants (see Definition 6.3). This transformation fits metarule (i).

Step 3. As proved in Theorem 7.4 below, at this step $A_2 \wedge B_2$ is $T_I \cup \mathcal{EUF}$-inconsistent; since $T_I \cup \mathcal{EUF}$ has quantifier-free interpolation by Theorem 2.5, we can compute an interpolant θ of the jointly unsatisfiable pair A_2, B_2. To get our desired $\mathcal{CARD}(T_I)$-interpolant, we only have to replace back in it the fresh AB-common constants introduced by our trasformations by the AB-common terms they name.

Example 7.1. This is the classical example (due to R. Jhala) showing that $\mathcal{AR}_{\text{ext}}$ does not have quantifier-free interpolation (one needs diff in the signature to recover it). Let A^0 be $\{c_1 = wr(c_2, i, e)\}$ and B^0 be $\{i_1 \neq i_2,\ rd(c_1, i_1) \neq rd(c_2, i_1),\ rd(c_1, i_2) \neq rd(c_2, i_2)\}$. Preprocessing adds the AB-common literals $|c_1| = l_{c_1}, |c_2| = l_{c_2}$ to both A_1^0 and B_1^0. Step 1 introduces the AB-common atoms

$$\mathtt{diff}_1(c_1, c_2) = k_1,\ \mathtt{diff}_2(c_1, c_2) = k_2$$

again to be added to both A_1^0 and B_1^0. We now examine the 0-instantiations $A(\mathcal{I}_A)$ and $B(\mathcal{I}_B)$ required by Step 2. Such instantiations are finitely many, but their number is rather large, so we limit ourselves to indicate a set of instantiations that is sufficient to produce an inconsistency in Step 3.

Considering $A(\mathcal{I}_A)$, we add to the formulae (33),(34) from Example 6.8, and also the following further instances of (18)

$$\begin{aligned} k_1 \neq i \rightarrow rd(c_1, k_1) = rd(c_2, k_1) \\ k_2 \neq i \rightarrow rd(c_1, k_2) = rd(c_2, k_2) \end{aligned} \tag{35}$$

Considering $B(\mathcal{I}_B)$, we need the following instances of (19)

$$\begin{aligned} l_{c_1} \geq 0 \ \wedge\ (rd(c_1, i_1) \neq \bot \leftrightarrow 0 \leq i_1 \leq l_{c_1}) \\ l_{c_2} \geq 0 \ \wedge\ (rd(c_2, i_1) \neq \bot \leftrightarrow 0 \leq i_1 \leq l_{c_2}) \\ l_{c_1} \geq 0 \ \wedge\ (rd(c_1, i_2) \neq \bot \leftrightarrow 0 \leq i_2 \leq l_{c_1}) \\ l_{c_2} \geq 0 \ \wedge\ (rd(c_2, i_2) \neq \bot \leftrightarrow 0 \leq i_2 \leq l_{c_2}) \end{aligned} \tag{36}$$

and also the following 6 instances of (22) (actually the first 4 formulas below can be added to $A(\mathcal{I}_A)$ too):

$$\begin{aligned} k_1 \geq k_2 \geq 0 \\ rd(c_1, k_1) = rd(c_2, k_1) \rightarrow k_1 = 0 \\ rd(c_1, k_2) = rd(c_2, k_2) \rightarrow k_2 = 0 \\ (l_{c_1} = l_{c_2} \wedge k_1 = k_2) \rightarrow k_1 = 0 \\ i_1 > k_2 \rightarrow (rd(c_1, i_1) = rd(c_2, i_1) \vee i_1 = k_1) \\ i_2 > k_2 \rightarrow (rd(c_1, i_2) = rd(c_2, i_2) \vee i_2 = k_1). \end{aligned} \tag{37}$$

The $T_I \cup \mathcal{EUF}$-inconsistency of (33),(34),(35),(36),(37) with B_0 is an easy problem for an SMT-solver (but can also be checked manually with some effort).

Since we got the desired inconsistency, we can rely in Step 3 on a black-box interpolation algorithm for $T_I \cup \mathcal{EUF}$. Such algorithm for instance produces the formula

$$k_2 = 0 \ \wedge \ (k_1 = k_2 \vee rd(c_1, k_2) = rd(c_2, k_2)) \tag{38}$$

which is $T_I \cup \mathcal{EUF}$-implied by $A(\mathcal{I}_A)$ and $T_I \cup \mathcal{EUF}$-inconsistent with $B(\mathcal{I}_B)$. To get an $\mathcal{CARD}(T_I)$-interpolant, it is enough to replace k_1, k_2 by $\mathtt{diff}_1(c_1, c_2), \mathtt{diff}_2(c_1, c_2)$ respectively in (38).

Example 7.2. We let

$$A^0 \equiv \{\mathtt{diff}(a_1, a_2) = j, \mathtt{diff}(a_1, c_1) = j_1, \mathtt{diff}(a_2, c_2) = j_2\}$$
$$B^0 \equiv \{j < l,\ j_1 < l,\ j_2 < l,\ rd(c_1, l) \neq rd(c_2, l)\}$$

In the preprocessing step, we must add the atoms $|a_1| = l_{a_1}, |a_2| = l_{a_2}$ to A_1^0 and $|c_1| = l_{c_1}, |c_2| = l_{c_2}$ to both A_1^0 and B_1^0. Since $N_A = N_B = 0$, we have $N = 1$; Step 1 adds the AB-common atom $\mathtt{diff}(c_1, c_2) = k_1$ with fresh index variable k_1. Step 2 makes the required 0-instantiations producing a 0-instantiated separated pair, let us call it (A, B). From such instantiations, we get in Step 3 the $T_I \cup \mathcal{EUF}$-interpolant

$$k_1 \leq \max(j_1, j_2, j). \tag{39}$$

This formula is in fact $T_I \cup \mathcal{EUF}$-implied by A_1 and $T_I \cup \mathcal{EUF}$-inconsistent with B_1 (notice that B_1 contains, in addition to $j < l,\ j_1 < l,\ j_2 < l, rd(c_1, l) \neq rd(c_2, l)$, also

$$k_1 < l \rightarrow rd(c_1, l) = rd(c_2, l)$$

by (22)). Using the recover instruction of meta-rule (iii), we get form (39) the $\mathcal{CARD}(T_I)$-interpolant $\mathtt{diff}(c_1, c_2) \leq \max(j_1, j_2, j)$.

Example 7.3. Let A^0 be

$$\{\mathtt{diff}(a, c_1) = i_1, \mathtt{diff}(b, c_2) = i_1, a_1 = wr(b, i_1, e_1), |a| = k,$$
$$a = wr(a_1, i_3, e_3), |a_1| = k,\ |b_1| = k,\ |c_1| = k,\ |c_2| = k\}$$

and let B^0 be $\{rd(c_1, i_1) \neq rd(c_2, i_2),\ i_1 < i_2,\ i_2 < i_3,\ i_3 < k,\ |c_1| = k,\ |c_2| = k\}$.

We do not need any preprocessing here; since $N = 3$,[13] Step 1 adds the AB-common atoms

$$\mathtt{diff}_1(c_1, c_2) = k_1,\ \mathtt{diff}_2(c_1, c_2) = k_2,\ \mathtt{diff}_3(c_1, c_2) = k_3\ .$$

Step 2 produces a separated pair (A, B) such that $A_2 \wedge B_2$ is $T_I \cup \mathcal{EUF}$-inconsistent (inconsistency can be tested via an SMT-solver like z3 [14] or MathSat [6]). The related $T_I \cup \mathcal{EUF}$-interpolant (once k_1, k_2 and k_3 are replaced by $\mathtt{diff}_1(c_1, c_2), \mathtt{diff}_2(c_1, c_2)$ and $\mathtt{diff}_3(c_1, c_2)$, respectively) gives our $\mathcal{CARD}(T_I)$-interpolant.

Theorem 7.4. *The above 3-steps algorithm computes a quantifier-free interpolant for every $\mathcal{CARD}(T_I)$-mutually unsatifiable pair A^0, B^0 of quantifier-free formulae.*

Proof. We only need to prove that Step 3 really applies.

Suppose not; let $A = (A_1, A_2)$ and $B = (B_1, B_2)$ be the separated pairs obtained after applications of Steps 1 and 2. If Step 3 does not apply, then $A_2 \wedge B_2$ is $T_I \cup \mathcal{EUF}$-consistent. We claim that (A, B) is $\mathcal{CARD}(T_I)$-consistent (contradicting that $(A^0, B^0) \subseteq (A, B)$ was $\mathcal{CARD}(T_I)$-inconsistent).

Let $\mathcal{M}$ be a $T_I \cup \mathcal{EUF}$-model of $A_2 \wedge B_2$. $\mathcal{M}$ is a two-sorted structure (the sorts are INDEX and ELEM) endowed for every array constant d occurring in $A \cup B$ of a function $d^{\mathcal{M}} : \mathtt{INDEX}^{\mathcal{M}} \longrightarrow \mathtt{ELEM}^{\mathcal{M}}$. In addition, $\mathtt{INDEX}^{\mathcal{M}}$ is a model of T_I. We list the properties of $\mathcal{M}$ that comes from the fact that our

[13] Recall that N is defined as $1 + max(N_A, N_B)$, where N_A (resp. N_B) be the number of A-local (resp. B-local) index constants occurring within a wr symbol in A^0 (resp. B^0).

Steps 1-2 have been applied (below, we denote by $k^{\mathcal{M}}$ the element of $\mathtt{INDEX}^{\mathcal{M}}$ assigned to an index constant k):[14]

(a) we have that $\mathcal{M} \models \bigwedge A_1(\mathcal{I}_A)$ (where $\mathcal{I}_A$ is the set of A-local constants) and $\mathcal{M} \models \bigwedge B_1(\mathcal{I}_B)$ (where $\mathcal{I}_B$ is the set of B-local constants): this is because (A_1, A_2) and (B_1, B_2) are 0-instantiated by Step 2;[15]

(b) for AB-common array variables c_1, c_2, we have that $A_1 \cap B_1$ contains a literal of the kind $\mathtt{diff}_n(c_1, c_2) = k_n$ for $n \leq N$; suppose that $\mathcal{M} \models l_{c_1} = l_{c_2}$ [16] and that k is an index constant such that $\mathcal{M} \models k \neq l$ for all AB-common index constant l; then, we can have $\mathcal{M} \models c_1(k) \neq c_2(k)$ only when $\mathcal{M} \models k < k_N$: this is because Step 1 has been applied and because of (a).

We expand $\mathcal{M}$ to an $\mathcal{AR}_{\text{ext}}(T_I)$-structure $\mathcal{N}$ and endow it with an assignment to our A-local and B-local variables, in such a way that all $\mathtt{diff}$ operators mentioned in A_1, B_1 are defined and all formulae in A, B are true. In view of Theorem 4.4, this structure can be expanded to the desired full model of $\mathcal{CARD}(T_I)$. We take $\mathtt{INDEX}^{\mathcal{N}}$ and $\mathtt{ELEM}^{\mathcal{N}}$ to be equal to $\mathtt{INDEX}^{\mathcal{M}}$ and $\mathtt{ELEM}^{\mathcal{M}}$; the T_I-reduct of $\mathcal{N}$ will be equal to the T_I-reduct of $\mathcal{M}$ and we let $x^{\mathcal{N}} = x^{\mathcal{M}}$ for all index and element constants occurring in $A \cup B$. $\mathtt{ARRAY}^{\mathcal{N}}$ is the set of all positive support functions from $\mathtt{INDEX}^{\mathcal{N}}$ into $\mathtt{ELEM}^{\mathcal{N}}$. The interpretation of A-local and B-local constants of sort $\mathtt{ARRAY}$ is more subtle. We need a détour to explain it.

Let k be an index constant such that $\mathcal{M} \models k \neq l$ for all A-local index constants l; we introduce an equivalence relation $\equiv_k$ on the set of A-local array variables as follows: $\equiv_k$ is the smallest equivalence relation that contains all pairs (a_1, a_2) such that $\mathcal{M} \models l_{a_1} = l_{a_2}$ and moreover an atom of one of the following two kinds belongs to A_1^0: (I) $a_1 = wr(a_2, i, e)$; (II) $\mathtt{diff}(a_1, a_2) = l$, for an l such that $\mathcal{M} \models l < k$.

Claim1: for every A-local constants a_1, a_2 such that $a_1 \equiv_k a_2$, the number of the A-local constants j such that $\mathcal{M} \models k < j$ and $a_1^{\mathcal{M}}(j^{\mathcal{M}}) \neq a_2^{\mathcal{M}}(j^{\mathcal{M}})$ is less or equal to $N_A < N$.

Proof of Claim1. This is easily shown by induction on the length of the finite sequence witnessing $a_1 \equiv_k a_2$. According to the definition of reflexive-symmetric-transitive closure, if $a_1 \equiv_k a_2$ holds then there are $d_0, \ldots, d_n$ such that $d_0 = a_1, d_n = a_2$ and for each $j < n$, we have that either (d_j, d_{j+1}) or (d_{j+1}, d_j) satisfies the above requirements: the induction is on such n. Notice that the statement is not entirely obvious because the number of the A-local index constants is much bigger than N_A (for instance, it includes the AB-common constants introduced in Step 1). However induction is easy: it goes through the atoms occurring in the input set A_1^0 and uses (a). The required observations are the following: if $d_j = wr(d_{j+1}, i, e) \in A_1^0$, then the only A-local constant where $d_j^{\mathcal{M}}$ and $d_{j+1}^{\mathcal{M}}$ can differ is i; if $\mathtt{diff}(d_j, d_{j+1}) = l \in A_1^0$ and $\mathcal{M} \models l < k$, then $d_j^{\mathcal{M}}$ and $d_{j+1}^{\mathcal{M}}$ cannot differ on any A-local constant above $k^{\mathcal{M}}$. Iterating these observations during induction, the claim is clear: we can collect at most the set of the A-local constants occurring in A_1^0 within a wr symbol.

Claim2: *if c_1, c_2 are AB-common and $c_1 \equiv_k c_2$, then $c_1^{\mathcal{M}}(k^{\mathcal{M}}) = c_2^{\mathcal{M}}(k^{\mathcal{M}})$.*

Proof of Claim2. We apply Claim1 to the AB-common array variables c_1, c_2 and let us consider the atoms $\mathtt{diff}_1(c_1, c_2) = k_1, \ldots, \mathtt{diff}_N(c_1, c_2) = k_N \in A_1$. Now $k_1, \ldots, k_N$ are all AB-common (hence also A-local) constants, moreover $N > N_A$ and the number of the A-local constants above $k^{\mathcal{M}}$ where c_1, c_2 differ is at most N_A. According to (b) above, if for absurdity $c_1^{\mathcal{M}}(k^{\mathcal{M}}) = c_2^{\mathcal{M}}(k^{\mathcal{M}})$ does not hold, then we have $\mathcal{M} \models k < k_N$. Since $\mathcal{M} \models k \geq 0$ (otherwise $c_1^{\mathcal{M}}(k^{\mathcal{M}}) = c_2^{\mathcal{M}}(k^{\mathcal{M}})$ follows), this means by Lemma 3.5 that we have $\mathcal{M} \models k_1 > \cdots > k_N > 0$. However $k_1, \ldots, k_N$ are all A-local

[14] Thus if e.g. k, l are index constants, $\mathcal{M} \models k = l$ is the same as $k^{\mathcal{M}} = l^{\mathcal{M}}$.

[15] The sets $A_1(\mathcal{I}_A), B_1(\mathcal{I}_B)$ are introduced in Definition 6.3.

[16] Recall that l_{c_1}, l_{c_2} are the AB-common constants such that the literals $|c_1| = l_{c_1}, |c_2| = l_{c_2}$ belongs to $A_1 \cap B_1$.

constants above k, their number is bigger than N_A, hence we must have $\mathcal{M} \models c_1^{\mathcal{M}}(k_i^{\mathcal{M}}) = c_2^{\mathcal{M}}(k_i^{\mathcal{M}})$ for some $i = 1, \ldots, N$; the latter implies $\mathcal{M} \models k_i = \cdots = k_N = 0$ by Lemma 3.5, absurd. ⊣

In order to interpret A-local constants of sort ARRAY, we assign to an A-local constant a of sort ARRAY the function $a^{\mathcal{N}}$ defined as follows for every $i \in \mathtt{INDEX}^{\mathcal{N}}$:

(†-i) if i is equal to $k^{\mathcal{M}}$, where k is an A-local constant, then $a^{\mathcal{N}}(i) := a^{\mathcal{M}}(k^{\mathcal{M}})$;

(†-ii) if i is different from $k^{\mathcal{M}}$ for every A-local constant k, but nevertheless i is equal to $k^{\mathcal{M}}$ for some (necessarily B-strict) index constant k and there is an AB-common array variable c such that $c \equiv_k a$, then $a^{\mathcal{N}}(i)$ is equal to $c^{\mathcal{M}}(k^{\mathcal{M}})$;

(†-iii) in the remaining cases, $a^{\mathcal{N}}(i)$ is equal to $el^{\mathcal{M}}$ or $\bot^{\mathcal{M}}$ depending whether $\mathcal{M} \models 0 \leq i \wedge i \leq l_a$ holds or not.

Notice that $a^{\mathcal{N}}(i)$ is univocally specified in case (†-ii) because of the above Claim2. We now show that

(*) *all $a^{\mathcal{N}}$ are positive-support functions and all formulæ from $A_1 \cup A_2$ are true in $\mathcal{N}$.*

Recall in fact that formulæ in A_2 are Boolean combinations of A-local atoms of the kind (30): these are $T_I \cup \mathcal{EUF}$-atoms and, due to their shape, each of them is true in $\mathcal{M}$ iff it is true in $\mathcal{A}$ (this is because rd functions are applied only to A-local index constants, so that the modifications we introduced for passing from $a^{\mathcal{M}}$ to $a^{\mathcal{N}}$ does not affect truth of these atoms). Concerning formulæ in A_1, these are all wr, diff and $|-|$-atoms.[17] The reason why they are true in $\mathcal{N}$ is the 0-instantiation performed by Step 2 (see (a) above). For example, consider an atom of the kind $|a| = l_a$ appearing in A_1: since formulae (19) have been instantiated with all the A-local index constants via Step 2, for all A-local index constant h, we have $\mathcal{M} \models rd(a, h) \neq \bot$ iff $\mathcal{M} \models 0 \leq h \leq i$. Now, thanks to definition (†), for all the elements $h \in \mathtt{INDEX}^{\mathcal{N}}$ we have an analogous result: in case h is equal to $k^{\mathcal{M}}$ for some A-local constant k, we employ definition (†-i), otherwise we use (†-ii) or (†-iii) (for (†-ii), notice that $a \equiv_k c$ implies that $\mathcal{M} \models l_a = l_c$ and 0-instantiation guarantees that $c^{\mathcal{M}}(k^{\mathcal{M}})$ is equal to $\bot^{\mathcal{M}}$ or to $el^{\mathcal{M}}$ depending whether $\mathcal{M} \models 0 \leq k \wedge k \leq l_c$ holds or not). Hence we get that $\mathcal{A}$ satisfies formula (19), since the universal quantifier has been instantiated in all possible ways. Thus $\mathcal{A} \models |a| = l_a$, by Lemma 3.3. The other cases are similar: notice in particular that if $a = wr(a', i, e) \in A_1$ then $a \equiv_k a'$. If $\mathtt{diff}(a, a') = i \in A_1$, the relevant case is when $\mathcal{M} \models i < k$ and $\mathcal{M} \models l_{a_1} = l_{a_2}$, but in this case we have $a \equiv_k a'$ too.

The assignments to the B-local array variables b are defined analogously, so that

(*) *all $b^{\mathcal{N}}$ are positive-support functions and all formulæ from $B_1 \cup B_2$ are true in $\mathcal{N}$.*

There is however one important point to notice: for all AB-common constants c of sort ARRAY, our specification of $c^{\mathcal{M}}$ *does not depend* on the fact that we use the above definition for A-local or for B-local array variables: to see this, we only have to notice that $c \equiv_k c$ holds in case (†-ii) is applied. This remark concludes our proof. □

It is not difficult to see that the quantifier-free $T_I \cup \mathcal{EUF}$-interpolation problem generated by our algorithm is of polynomial size, thus a polysize reduction obtains.

8 FURTHER RELATED WORK AND CONCLUSIONS

We introduced two theories of arrays, namely the theory $\mathcal{CARD}(T_I)$ of contiguous arrays with maxdiff and its extension $\mathcal{CARDC}(T_I)$ that also supports 'constant' arrays'. These theories are strictly more expressive than McCarthy's theory and the other variants studied in the literature: notably, the length of arrays is available, and inside it arrays are fully defined in every memory

[17] In addition, we have $\mathtt{diff}_n$-atoms for $n > 1$, but these are all AB-common atoms that were not part of the initial pair A^0, B^0. In fact they are also true in $\mathcal{N}$, but strictly speaking we do not need to check this fact to get the absurdity that $A^0 \wedge B^0$ is $\mathcal{CARD}(T_I)$-consistent.

location. We proved that $\mathcal{CARDC}(T_I)$ admits general interpolation by showing that its models are strongly amalgamable; the existence of amalgams also implies that $\mathcal{CARD}(T_I)$ has interpolants. We also studied the SMT problem for $\mathcal{CARD}(T_I)$ and showed through instantiations techniques that it is decidable. Finally, we provided a general algorithm for computing $\mathcal{CARD}(T_I)$ quantifier-free interpolants that relies on a polynomial reduction to the problem of computing general interpolants for the index theory. Differently from the previous algorithm in [16], this procedure avoids full instantiation of terms.

One future research direction regards the implementation of this procedure, which is still missing. In the last decade, some implemented approaches have been introduced to compute interpolants for different theories, by relying on different techniques. For complex theories, in [25] McMillan proposed an interpolating proof calculus to compute interpolants via refutational proofs obtained from the z3 SMT-solver. It is worth mentioning his approach because it takes advantages from the flexibility of the z3 solver to deal with several theories and their combination: it makes use of a secondary interpolation engine in order to 'fill the gaps' of refutational proofs introduced by *theory lemmas*, which are specific formulæ derived by the satellites theories encoded in z3. This secondary engine only needs an interpolation algorithm for `QF_UFLIA`. This approach can be used to compute interpolants for array theories, but since here theory lemmas use quantified formulæ, the method can generate quantified formulæ.

Concerning in particular array theories, another notable approach for computing interpolants is due to the authors of [17], which exploited the proof tree preserving interpolation scheme from [12] to construct interpolants via a resolution proof. This method supports the use of the `diff` operation between arrays in order to compute quantifier-free interpolants, but the semantic interpretation of `diff` is undetermined as in [7].

In [9], the authors presented AXDInterpolator [1], an implementation of the interpolation algorithm from [16], which allows the user to choose z3, MathSAT, or SMTInterpol ([11]) as the underlying interpolation engines. In order to show its feasibility, it was tested against a benchmark based on C programs from the ReachSafety-Arrays and MemSafety-Arrays tracks of SV-COMP [4]. Since many C programs from [4] require the usage of the proper array length, we plan to develop a tool that implements the new algorithm for contiguous arrays presented in this paper.

There is still a question concerning our interpolation algorithm that needs to be investigated: extending the algorithm to the theory $\mathcal{CARDC}(T_I)$ (with constant arrays in the language). In order to handle constant arrays, the construction of Theorem 7.4 is still appropriate, except for the fact that condition (†-ii) should not be applied to define $\mathtt{Const}(i)^{\mathcal{N}}$ when i is an A-strict constant such that $i^{\mathcal{M}}$ is equal to $j^{\mathcal{M}}$ for some AB-common j. To avoid this, one could introduce right after Step 1 some form of guessing for equalities between index constants: however, such a guessing (based on colorings) would create branches and consequently would not produce a polynomial instance of a $T_I \cup \mathcal{EUF}$-interpolation problem in Step 3. This issue needs further analysis.

Finally, although quite challenging, it would be interesting to extend our interpolation results also to array theories combined with cardinality constraints, similar to those introduced, e.g., in [2],[27].

REFERENCES

[1] 2021. AXDInterpolator. https://github.com/typesAreSpaces/AXDInterpolator Accessed: 2021-10-12.

[2] Francesco Alberti, Silvio Ghilardi, and Elena Pagani. 2017. Cardinality constraints for arrays (decidability results and applications). *Formal Methods Syst. Des.* 51, 3 (2017), 545–574. https://doi.org/10.1007/s10703-017-0279-6

[3] Paul D. Bacsich. 1975. Amalgamation properties and interpolation theorems for equational theories. *Algebra Universalis* 5 (1975), 45–55.

[4] Dirk Beyer. 2021. Software Verification: 10th Comparative Evaluation (SV-COMP 2021). In *Proc. of TACAS 2021 (LNCS, Vol. 12652)*. Springer, Berlin, Heidelberg, 401–422. https://doi.org/10.1007/978-3-030-72013-1_24

[5] Aaron R. Bradley, Zohar Manna, and Henny B. Sipma. 2006. What's Decidable About Arrays?. In *Proc. of VMCAI 2006 (LNCS, Vol. 3855)*. Springer, Berlin, Heidelberg, 427–442. https://doi.org/10.1007/11609773_28

[6] Roberto Bruttomesso, Alessandro Cimatti, Anders Franzén, Alberto Griggio, and Roberto Sebastiani. 2008. The MathSAT 4 SMT Solver. In *Proc. of CAV 2008 (LNCS, Vol. 5123)*. Springer, Berlin, Heidelberg, 299–303. https://doi.org/10.1007/978-3-540-70545-1_28

[7] Roberto Bruttomesso, Silvio Ghilardi, and Silvio Ranise. 2012. Quantifier-Free Interpolation of a Theory of Arrays. *Log. Methods Comput. Sci.* 8, 2 (2012). https://doi.org/10.2168/LMCS-8(2:4)2012

[8] Roberto Bruttomesso, Silvio Ghilardi, and Silvio Ranise. 2014. Quantifier-free interpolation in combinations of equality interpolating theories. *ACM Trans. Comput. Log.* 15, 1 (2014), 5:1–5:34. https://doi.org/10.1145/2490253

[9] José Abel Castellanos Joo, Silvio Ghilardi, Alessandro Gianola, and Deepak Kapur. 2021. AXDInterpolator: A Tool for Computing Interpolants for Arrays with MaxDiff. In *Proc. of SMT 2021*, Vol. 2908. CEUR Workshop Proceedings, 40–52. http://ceur-ws.org/Vol-2908/paper15.pdf

[10] C. C. Chang and H. Jerome Keisler. 1990. *Model Theory* (third ed.). North-Holland, Amsterdam-London.

[11] Jürgen Christ, Jochen Hoenicke, and Alexander Nutz. 2012. SMTInterpol: An Interpolating SMT Solver. In *Proc. of SPIN 2012 (LNCS, Vol. 7385)*. Springer, Berlin, Heidelberg, 248–254.

[12] Jürgen Christ, Jochen Hoenicke, and Alexander Nutz. 2013. Proof Tree Preserving Interpolation. In *Proc. of TACAS 2013 (LNCS, Vol. 7795)*. Springer, Berlin, Heidelberg, 124–138. https://doi.org/10.1007/978-3-642-36742-7_9

[13] William Craig. 1957. Three uses of the Herbrand-Gentzen theorem in relating model theory and proof theory. *J. Symbolic Logic* 22 (1957), 269–285.

[14] Leonardo De Moura and Nikolaj Bjørner. 2008. Z3: An Efficient SMT Solver. In *Proc. of the TACAS 2008 (LNCS, Vol. 4963)*. Springer, Berlin, Heidelberg, 337–340. https://doi.org/10.1007/978-3-540-78800-3_24

[15] Silvio Ghilardi. 2004. Model Theoretic Methods in Combined Constraint Satisfiability. *J. Autom. Reasoning* 33, 3-4 (2004), 221–249.

[16] Silvio Ghilardi, Alessandro Gianola, and Deepak Kapur. 2021. Interpolation and Amalgamation for Arrays with MaxDiff. In *Proc. of FOSSACS 2021 (LNCS, Vol. 12650)*. Springer, Berlin, Heidelberg, 268–288. https://doi.org/10.1007/978-3-030-71995-1_14

[17] Jochen Hoenicke and Tanja Schindler. 2018. Efficient Interpolation for the Theory of Arrays. In *Proc. of IJCAR 2018 (LNCS, Vol. 10900)*. Springer, Berlin, Heidelberg, 549–565. https://doi.org/10.1007/978-3-319-94205-6_36

[18] Jochen Hoenicke and Tanja Schindler. 2019. Interpolation and the Array Property Fragment. *CoRR* abs/1904.11381 (2019). arXiv:1904.11381 http://arxiv.org/abs/1904.11381

[19] Deepak Kapur, Rupak Majumdar, and Calogero G. Zarba. 2006. Interpolation for Data Structures. In *Proc. of SIGSOFT/FSE 2006*. ACM, 105–116.

[20] Emil W. Kiss, László Márki, Péter Pröhle, and Walter Tholen. 1982. Categorical algebraic properties. A compendium on amalgamation, congruence extension, epimorphisms, residual smallness, and injectivity. *Studia Sci. Math. Hungar.* 18, 1 (1982), 79–140.

[21] Hari Govind Vediramana Krishnan, Yakir Vizel, Vijay Ganesh, and Arie Gurfinkel. 2019. Interpolating Strong Induction. In *Computer Aided Verification - 31st International Conference, CAV 2019, New York City, NY, USA, July 15-18, 2019, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 11562)*, Isil Dillig and Serdar Tasiran (Eds.). Springer, 367–385. https://doi.org/10.1007/978-3-030-25543-5_21

[22] John McCarthy. 1962. Towards a Mathematical Science of Computation. In *Proc. of IFIP Congress 1962*. North-Holland, 21–28.

[23] Kenneth L. McMillan. 2003. Interpolation and SAT-Based Model Checking. In *Proc. of CAV 2003 (LNCS, Vol. 2725)*. Springer, Berlin, Heidelberg, 1–13. https://doi.org/10.1007/978-3-540-45069-6_1

[24] Kenneth L. McMillan. 2006. Lazy Abstraction with Interpolants. In *Proc. of CAV 2006 (LNCS, Vol. 4144)*. Springer, Berlin, Heidelberg, 123–136.

[25] Kenneth L. McMillan. 2011. Interpolants from Z3 proofs. In *Proc. of FMCAD 2011*. FMCAD Inc., 19–27.

[26] Greg Nelson and Derek C. Oppen. 1979. Simplification by Cooperating Decision Procedures. *ACM Trans. Program. Lang. Syst.* 1, 2 (1979), 245–257.

[27] Rodrigo Raya and Viktor Kunkak. 2022. NP Satisfiability for Arrays as Powers. In *Proc. of VMCAI 2022 (LNCS, Vol. 13182)*. Springer, Berlin, Heidelberg, 301–318. https://doi.org/10.1007/978-3-030-94583-1_15

[28] Andrey Rybalchenko and Viorica Sofronie-Stokkermans. 2007. Constraint Solving for Interpolation. In *Proc. of VMCAI 2007 (LNCS, Vol. 4349)*. Springer, Berlin, Heidelberg, 346–362. https://doi.org/10.1007/978-3-540-69738-1_25

[29] Andrey Rybalchenko and Viorica Sofronie-Stokkermans. 2010. Constraint solving for interpolation. *J. Symb. Comput.* 45, 11 (2010), 1212–1233. https://doi.org/10.1016/j.jsc.2010.06.005

[30] Viorica Sofronie-Stokkermans. 2006. Interpolation in Local Theory Extensions. In *Proc. of IJCAR 2006 (LNCS, Vol. 4130)*. Springer, Berlin, Heidelberg, 235–250. https://doi.org/10.1007/11814771_21

[31] Viorica Sofronie-Stokkermans. 2008. Interpolation in Local Theory Extensions. *Log. Methods Comput. Sci.* 4, 4 (2008). https://doi.org/10.2168/LMCS-4(4:1)2008

[32] Viorica Sofronie-Stokkermans. 2016. On Interpolation and Symbol Elimination in Theory Extensions. In *Proc. of IJCAR 2016 (LNCS, Vol. 9706)*. Springer, Berlin, Heidelberg, 273–289. https://doi.org/10.1007/978-3-319-40229-1_19

[33] Viorica Sofronie-Stokkermans. 2018. On Interpolation and Symbol Elimination in Theory Extensions. *Log. Methods Comput. Sci.* 14, 3 (2018). https://doi.org/10.23638/LMCS-14(3:23)2018

[34] Nishant Totla and Thomas Wies. 2013. Complete instantiation-based interpolation. In *Proc. of POPL 2013*. ACM, 537–548. https://doi.org/10.1145/2429069.2429132

[35] Nishant Totla and Thomas Wies. 2016. Complete Instantiation-Based Interpolation. *J. Autom. Reasoning* 57, 1 (2016), 37–65. https://doi.org/10.1007/s10817-016-9371-7

[36] Yakir Vizel and Arie Gurfinkel. 2014. Interpolating Property Directed Reachability. In *Computer Aided Verification - 26th International Conference, CAV 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 18-22, 2014. Proceedings*. 260–276. https://doi.org/10.1007/978-3-319-08867-9_17