Deciding Extensions of the Theories of Vectors and Bags

Patrick Maier

Laboratory for Foundations of Computer Science
School of Informatics, The University of Edinburgh, Scotland
Patrick.Maier@ed.ac.uk

Abstract. Vectors and bags are basic collection data structures, which are used frequently in programs and specifications. Reasoning about these data structures is supported by established algorithms for deciding ground satisfiability in the theories of arrays (for vectors) and multisets (for bags), respectively. Yet, these decision procedures are only able to reason about vectors and bags in isolation, not about their combination.
This paper presents a decision procedure for the combination of the theories of vectors and bags, even when extended with a function bagof bridging between vectors and bags. The function bagof converts vectors into the bags of their elements, thus admitting vector/bag comparisons. Moreover, for certain syntactically restricted classes of ground formulae decidability is retained if the theory of vectors is extended further with a map function which applies uninterpreted functions to all elements of a vector.

1 Introduction

Vectors and bags are basic collection data structures, which are used frequently in programs and specifications. Reasoning about these data structures is supported by decision procedures for deciding the satisfiability of quantifier-free formulae in the theories of arrays (for vectors) and multisets (for bags), respectively. However, known decision procedures are essentially only able to reason about vectors and bags in isolation, whereas practical software verification problems often require non-trivial combinations.

Let us illustrate this problem with an example. Figure 1 shows a Java method `sendBulk` taking a message text `msg`, a group of recipients `group` (represented as an array of phone numbers) and a *resource manager* `mgr` holding (symbolic representations of) the resources required to send text messages to the recipients. As the cost of sending text messages may vary depending on the recipient, the state of a resource manager cannot be simply the number of messages that may be sent; instead it should be a multiset of resources, representing exactly how many messages may be sent to whom. In order to enforce the resource limit, at least at run-time, actual use of resources must be preceded by a call to the resource manager's `use` method, which checks whether the required resource is present and if so, deduces it, otherwise aborts the program. This is what's happening in the body of method `sendBulk`, which iterates over `group`, sending `msg` to each member by calling `SMS.send`, but only after checking for and using up the associated resource by calling `mgr.use`. This approach to run-time monitoring of resources via explicit resource managers has been described in [1], for example.

```
void sendBulk(String msg, PhoneNum[] group, ResourceMgr mgr) {
    for (int i=0; i < group.length; i++) {
        mgr.use(MessageResource(group[i]));
        SMS.send(msg, group[i]);
    }
}
```

$$
\begin{aligned}
\mathit{PreCond} &\equiv \mathrm{bagof}(\mathrm{map}_{\texttt{MessageResource}}(\texttt{group})) \subseteq \texttt{mgr} \\
\mathit{PostCond} &\equiv \texttt{\textbackslash old(mgr)} = \texttt{mgr} \uplus \mathrm{bagof}(\mathrm{map}_{\texttt{MessageResource}}(\texttt{group})) \\
\mathit{LoopInv} &\equiv 0 \leq \texttt{i} \leq \texttt{group.length} \wedge \\
&\quad \mathrm{bagof}(\mathrm{map}_{\texttt{MessageResource}}(\texttt{group[i:group.length]})) \subseteq \texttt{mgr} \wedge \\
&\quad \texttt{\textbackslash old(mgr)} = \texttt{mgr} \uplus \mathrm{bagof}(\mathrm{map}_{\texttt{MessageResource}}(\texttt{group[0:i]})) \\
\mathit{VC} &\equiv \mathit{LoopInv} \wedge \neg \mathit{LoopInv}[\texttt{i} + 1/\texttt{i}, \texttt{mgr}'/\texttt{mgr}] \wedge \texttt{i} < \texttt{group.length} \wedge \\
&\quad \mathrm{count}(\texttt{mgr}, \texttt{MessageResource(group[i])}) > 0 \wedge \\
&\quad \texttt{mgr} = \texttt{mgr}' \uplus [\![\texttt{MessageResource(group[i])}]\!]^{(1)}
\end{aligned}
$$

Fig. 1. Java bulk messaging example: code and specification of send loop.

Run-time monitoring provides dynamic guarantees of *resource safety*, as abuse of resources will be trapped. However, aborting a program midway is not always a desirable solution; it would be better if we could guarantee statically that a program will never even *attempt* to abuse resources. This is done in [2], which presents a type system for proving static resource safety in a programming language with explicit resource managers. When proving resource safety of a method like `sendBulk`, whether it is done via a type system as in [2] or in the more traditional way by generating verification conditions, the hard part is reasoning about constraints between the program variables. Ideally, we'd like to have fully automated theorem provers for this task.

Let us take a look at the constraints required to express invariants and pre- and postconditions for `sendBulk`, see the bottom half of Figure 1. Informally, the precondition states that `mgr` is a super-multiset of the vector `group`, when the latter is viewed as a multiset of resources. To express this view, we first need to convert `group` into a vector of resources (by applying the map function) and then into a multiset of resources (by applying the bagof function). The postcondition states that the old `mgr` splits into two multisets: the new `mgr` and the multiset of resources corresponding to the vector `group`. The loop invariant essentially combines pre- and postcondition, but for different slices of the vector `group`. The first conjunct bounds the loop variable `i`, the second is the precondition for the remainder of the loop, i. e., for the subvector from index `i` to the end, and the third is the effect of the loop so far, i. e., the postcondition for the subvector from index 0 up to (but excluding) `i`. The (negated) verification condition conjoins the loop invariant before, the negated loop invariant after the execution of the loop (arising by substituting the variables `i` and `mgr`), the loop condition, and the precondition (`mgr` has some resources corresponding to number `group[i]`) and effect (`mgr`$'$ holds one unit of resource less than `mgr`) of the loop body. Hence, to verify the loop invariant of an example even this simple we must prove unsatisfiability of

constraints about bags, vectors, subvectors, the map function for transforming vectors pointwise, and the bagof function for transforming vectors into multisets.

Decision procedures for vectors (or arrays) exist for quite some time; early work goes back to the late 1970s [6, 10]. Recently, [4] and [3] found expressive yet decidable extensions of the theory arrays by injectivity predicates and by restricted quantification over array indices, respectively. Decision procedures for bags (or multisets) have been published recently in [12] and [7, 8], where the latter supports a cardinality operator. However, decision procedures combining vectors and bags and linking them via the bagof function (or something similar) do not exist.

The main contribution of this paper is a decision procedure for ground satisfiability in the combination of the theories of vectors and bags extended with the function bagof. For certain syntactically restricted classes of ground formulae decidability is retained if the theory of vectors is extended further with a map_f function for transforming vectors pointwise by applying the uninterpreted function f. The decision procedure reduces formulae containing $\mathrm{bagof}(\cdot)$ to formulae without by instantiating universally quantified variables in the axiomatisation of the bagof function, eventually reducing the problem to the theories of vectors and bags. It relies on a decision procedure for the Array Property Fragment described in [3] and on a decision procedure for multisets with cardinality described in [7, 8].

Plan. Section 2 introduces some basic notation. Section 3 presents the theories of bags, vectors, map and bagof functions. Section 4 utilises known results to construct a decision procedure for the combination of the theories of bags and vectors (including map). Section 5 presents our main result: an extension of the decision procedure (and its proof of correctness) to cope with bagof.

2 Preliminaries

We work in the framework of *many-sorted first-order logic with equality*, assuming familiarity with the basic syntactic and semantic concepts. Below we fix some notation.

Throughout the paper, we fix three countably infinite and pairwise disjoint universes: a set $\mathcal{S}$ of *sorts*, a set $\mathcal{F}$ of *function symbols* and a set $\mathcal{X}$ of *variable symbols*. By S^+ we denote the set of non-empty words over a set S.

Signatures. A *decorated variable* x_s is a pair consisting of a variable $x \in \mathcal{X}$ and a sort $s \in \mathcal{S}$. A *decorated function symbol* f_w is a pair consisting of a function symbol $f \in \mathcal{F}$ and an *arity* $w \in \mathcal{S}^+$. A decorated function symbol c_s of arity $s \in \mathcal{S}$ is called a *decorated constant*. For the sake of readability, we may write decorated constants and function symbols in the form $c : s$ and $f : s_1 \times \ldots \times s_n \to s_0$ instead of c_s and $f_{s_0 s_1 \ldots s_n}$, respectively. We may drop decorations entirely if they are clear from the context.

A *(many-sorted) signature* Σ is a pair $\Sigma = \langle S, F \rangle$ where $S \subseteq \mathcal{S}$ is a non-empty finite set of sorts and $F \subseteq \mathcal{F} \times S^+$ is a set of decorated function symbols. We may write Σ^{S} and Σ^{F} to refer to S and F, respectively. If Σ_1 and Σ_2 are signatures then the *union* $\Sigma_1 \cup \Sigma_2 = \langle \Sigma_1^{\mathrm{S}} \cup \Sigma_2^{\mathrm{S}}, \Sigma_1^{\mathrm{F}} \cup \Sigma_2^{\mathrm{F}} \rangle$ and *intersection* $\Sigma_1 \cap \Sigma_2 = \langle \Sigma_1^{\mathrm{S}} \cap \Sigma_2^{\mathrm{S}}, \Sigma_1^{\mathrm{F}} \cap \Sigma_2^{\mathrm{F}} \rangle$ are signatures, too. Two signatures Σ_1 and Σ_2 are *disjoint* if $\Sigma_1^{\mathrm{F}} \cap \Sigma_2^{\mathrm{F}} = \emptyset$, i. e., disjoint signatures do not share decorated function symbols but may share sorts.

Union and intersection induce a lattice structure on signatures. We denote the induced partial order by $\supseteq$, where $\Sigma_2 \supseteq \Sigma_1$ (in words: Σ_2 *extends* Σ_1) if $\Sigma_2^{\mathrm{S}} \supseteq \Sigma_1^{\mathrm{S}}$ and $\Sigma_2^{\mathrm{F}} \supseteq \Sigma_1^{\mathrm{F}}$. The *constant expansion* of Σ, denoted by $\hat{\Sigma}$, is the greatest signature extending Σ such that $\hat{\Sigma}^{\mathrm{S}} = \Sigma^{\mathrm{S}}$ and all function symbols in $\hat{\Sigma}^{\mathrm{F}} \setminus \Sigma^{\mathrm{F}}$ are constants, i. e., $\hat{\Sigma}$ provides infinitely many constants per sort.

Terms and formulae. Let Σ be a signature. *Σ-terms* are well-sorted terms constructed from decorated function symbols in Σ^{F} and decorated variables in $\mathcal{X} \times \Sigma^{\mathrm{S}}$. A *ground* Σ-term is a variable-free Σ-term. If Σ is clear from the context, we may drop the prefix and write "term" instead of "Σ-term". We may refer to terms of sort $s \in \Sigma^{\mathrm{S}}$ as *s-terms*.

A Σ-atom is an equality[1] $t = t'$, where t and t' are Σ-terms of the same sort. A Σ-literal is a Σ-atom $t = t'$ or its negation $\neg(t = t')$, often written as $t \neq t'$. If we want to stress that the sort of left- and right-hand sides of a Σ-atom (resp.-literal) is s, we may refer to the atom (resp. literal) as s-atom (resp. s-literal). Σ-formulae are formed from Σ-atoms by the usual connectives ($\neg$, $\wedge$, $\vee$, $\Rightarrow$) and quantifiers ($\forall$, $\exists$) of first-order logic, inducing the usual notion of bound and free variables. A *Σ-sentence* is a Σ-formula without free variables, and a *Σ-theory* is a set of Σ-sentences. Note that a Σ-theory $\mathcal{T}$ is also a Σ'-theory, for all Σ' extending Σ. A *ground* Σ-formula is a quantifier-free Σ-sentence.

Algebras and satisfiability. Let $\Sigma = \langle S, F \rangle$ be a signature. A *Σ-algebra* $\mathcal{A}$ is a pair $\langle S^{\mathcal{A}}, F^{\mathcal{A}} \rangle$ where $S^{\mathcal{A}}$ is a S-indexed family of carrier sets and $F^{\mathcal{A}}$ is a F-indexed family of functions on the carrier sets. More formally, $S^{\mathcal{A}} = \{s^{\mathcal{A}} | s \in \Sigma^{\mathrm{S}}\}$ is a family of non-empty and pairwise disjoint sets $s^{\mathcal{A}}$, and $F^{\mathcal{A}} = \{f^{\mathcal{A}}_{s_0 s_1 \ldots s_n} | f_{s_0 s_1 \ldots s_n} \in F\}$ is a family of functions $f^{\mathcal{A}}_{s_0 s_1 \ldots s_n}$ from $s_1^{\mathcal{A}} \times \cdots \times s_n^{\mathcal{A}}$ to $s_0^{\mathcal{A}}$. We extend the interpretation of function symbols in a Σ-algebra $\mathcal{A}$ homomorphically to ground Σ-terms t in the usual way, denoting the resulting element of the algebra by $t^{\mathcal{A}}$. Note that for all Σ' extending Σ, a Σ'-algebra $\mathcal{A}$ can also be viewed as a Σ-algebra.

The truth of a Σ-sentence ϕ in a Σ-algebra $\mathcal{A}$, denoted by $\mathcal{A} \models \phi$, is defined in the usual way. $\mathcal{A}$ is a *model* of a Σ-theory $\mathcal{T}$, also denoted by $\mathcal{A} \models \mathcal{T}$, if $\mathcal{A} \models \phi$ for all $\phi \in \mathcal{T}$. Given a Σ-algebra $\mathcal{A}$, the theory $\mathcal{T}(\mathcal{A})$ is the greatest Σ-theory which has $\mathcal{A}$ as a model. Given a class Δ of Σ-algebras, $\mathcal{T}(\Delta) = \bigcap_{\mathcal{A} \in \Delta} \mathcal{T}(\mathcal{A})$ is the greatest Σ-theory which has all algebras $\mathcal{A} \in \Delta$ as models.

Let $\mathcal{T}$ be a Σ-theory. A Σ-algebra $\mathcal{A}$ is a $\mathcal{T}$-model if $\mathcal{A} \models \mathcal{T}$. A $\hat{\Sigma}$-sentence ϕ is *$\mathcal{T}$-satisfiable* if there is a $\mathcal{T}$-model $\mathcal{A}$ which is a model of ϕ; note that $\mathcal{A}$ must be a $\hat{\Sigma}$-algebra. Two $\hat{\Sigma}$-sentences ϕ and ψ are *$\mathcal{T}$-equisatisfiable* if both are $\mathcal{T}$-satisfiable or neither is.

Given a subset $S' \subseteq \Sigma^{\mathrm{S}}$ of sorts, a Σ-theory $\mathcal{T}$ is *stably infinite w. r. t. S'* if every $\mathcal{T}$-satisfiable ground $\hat{\Sigma}$-formula ϕ has a $\mathcal{T}$-model $\mathcal{A}$ such that $s^{\mathcal{A}}$ is infinite for all $s \in S'$. $\mathcal{T}$ is *stably infinite* if it is stably infinite w. r. t. the set of all sorts Σ^{S}.

3 Theories

We introduce the signatures and theories used throughout this paper, see also Figure 2.

[1] We consider equality the only predicate symbol of the logic. Other predicates can be encoded as functions with a non-trivial codomain.

Σ_{E}-theory $\mathcal{T}_{\mathrm{E}}$ of elements

Σ_{E} arbitrary signature disjoint from all signatures below,
$\mathcal{T}_{\mathrm{E}}$ arbitrary stably infinite theory with decidable ground $\mathcal{T}_{\mathrm{E}}$-satisfiability problem.

Σ_{INT}-theory $\mathcal{T}_{\mathrm{INT}}$ of Presburger arithmetic

$$\Sigma^{\mathrm{S}}_{\mathrm{INT}} = \{\mathrm{INT}\}$$
$$\Sigma^{\mathrm{F}}_{\mathrm{INT}} = \big\{0, 1 : \mathrm{INT},\ +, -, \min, \max : \mathrm{INT} \times \mathrm{INT} \to \mathrm{INT}\big\}$$
$$\mathcal{T}_{\mathrm{INT}} = \mathcal{T}(\mathcal{A}_{\mathrm{INT}}) \text{ where } \mathcal{A}_{\mathrm{INT}} \text{ is the standard } \Sigma_{\mathrm{INT}}\text{-algebra.}$$

Σ_{BAG}-theory $\mathcal{T}_{\mathrm{BAG}}$ of multisets with cardinality

$$\Sigma^{\mathrm{S}}_{\mathrm{BAG}} = \Sigma^{\mathrm{S}}_{\mathrm{INT}} \cup \Sigma^{\mathrm{S}}_{\mathrm{E}} \cup \{\mathrm{BAG}_s \mid s \in \Sigma^{\mathrm{S}}_{\mathrm{E}}\}$$
$$\Sigma^{\mathrm{F}}_{\mathrm{BAG}} = \Sigma^{\mathrm{F}}_{\mathrm{INT}} \cup \big\{|\cdot| : \mathrm{BAG}_s \to \mathrm{INT},\ \mathrm{count} : \mathrm{BAG}_s \times s \to \mathrm{INT},\ [\![]\!] : \mathrm{BAG}_s,\ [\![\cdot]\!]^{(\cdot)} : s \times \mathrm{INT} \to \mathrm{BAG}_s,\ \cap, \cup, \uplus : \mathrm{BAG}_s \times \mathrm{BAG}_s \to \mathrm{BAG}_s \;\big|\; s \in \Sigma^{\mathrm{S}}_{\mathrm{E}}\big\}$$
$$\mathcal{T}_{\mathrm{BAG}} = \mathcal{T}(\Delta_{\mathrm{BAG}}) \text{ where } \Delta_{\mathrm{BAG}} \text{ is the class of standard } \Sigma_{\mathrm{BAG}}\text{-algebras.}$$

Σ_{VEC}-theory $\mathcal{T}_{\mathrm{VEC}}$ of vectors

$$\Sigma^{\mathrm{S}}_{\mathrm{VEC}} = \Sigma^{\mathrm{S}}_{\mathrm{INT}} \cup \Sigma^{\mathrm{S}}_{\mathrm{E}} \cup \{\mathrm{VEC}_s \mid s \in \Sigma^{\mathrm{S}}_{\mathrm{E}}\}$$
$$\Sigma^{\mathrm{F}}_{\mathrm{VEC}} = \Sigma^{\mathrm{F}}_{\mathrm{INT}} \cup \big\{\mathrm{fst}, \mathrm{end} : \mathrm{VEC}_s \to \mathrm{INT},\ \cdot[\cdot] : \mathrm{VEC}_s \times \mathrm{INT} \to s,\ \mathrm{const} : s \times \mathrm{INT} \times \mathrm{INT} \to \mathrm{VEC}_s,\ \cdot[\cdot{:}\cdot] : \mathrm{VEC}_s \times \mathrm{INT} \times \mathrm{INT} \to \mathrm{VEC}_s,\ \cdot\{\cdot \leftarrow \cdot\} : \mathrm{VEC}_s \times \mathrm{INT} \times s \to \mathrm{VEC}_s \;\big|\; s \in \Sigma^{\mathrm{S}}_{\mathrm{E}}\big\}$$
$$\begin{aligned}\mathcal{T}_{\mathrm{VEC}} = \big\{&\forall u, v : \mathrm{fst}(u) = \mathrm{fst}(v) \wedge \mathrm{end}(u) = \mathrm{end}(v) \wedge \\ &\quad (\forall k : \mathrm{fst}(u) \le k < \mathrm{end}(u) \Rightarrow u[k] = v[k]) \Rightarrow u = v,\\ &\forall x, i, j : \mathrm{fst}(\mathrm{const}(x, i, j)) = i \wedge \mathrm{end}(\mathrm{const}(x, i, j)) = j,\\ &\forall x, i, j, k : i \le k < j \Rightarrow \mathrm{const}(x, i, j)[k] = x,\\ &\forall v, i, j : \mathrm{fst}(v[i{:}j]) = \max(i, \mathrm{fst}(v)) \wedge \mathrm{end}(v[i{:}j]) = \min(j, \mathrm{end}(v)),\\ &\forall v, i, j, k : \mathrm{fst}(v[i{:}j]) \le k < \mathrm{end}(v[i{:}j]) \Rightarrow v[i{:}j][k] = v[k],\\ &\forall v, i, x : \mathrm{fst}(v\{i \leftarrow x\}) = \mathrm{fst}(v) \wedge \mathrm{end}(v\{i \leftarrow x\}) = \mathrm{end}(v),\\ &\forall v, i, x : \mathrm{fst}(v) \le i < \mathrm{end}(v) \Rightarrow v\{i \leftarrow x\}[i] = x,\\ &\forall v, i, x, k : \mathrm{fst}(v) \le k < \mathrm{end}(v) \wedge i \ne k \Rightarrow v\{i \leftarrow x\}[k] = v[k]\big\}\end{aligned}$$

Σ_{BAGOF}-theory $\mathcal{T}_{\mathrm{BAGOF}}$ of bagof function on vectors

$$\Sigma^{\mathrm{S}}_{\mathrm{BAGOF}} = \Sigma^{\mathrm{S}}_{\mathrm{VEC}} \cup \Sigma^{\mathrm{S}}_{\mathrm{BAG}}$$
$$\Sigma^{\mathrm{F}}_{\mathrm{BAGOF}} = \Sigma^{\mathrm{F}}_{\mathrm{VEC}} \cup \Sigma^{\mathrm{F}}_{\mathrm{BAG}} \cup \{\mathrm{bagof} : \mathrm{VEC}_s \to \mathrm{BAG}_s \mid s \in \Sigma^{\mathrm{S}}_{\mathrm{E}}\}$$
$$\begin{aligned}\mathcal{T}_{\mathrm{BAGOF}} = \big\{&\forall v : |\mathrm{bagof}(v)| = \max(\mathrm{end}(v) - \mathrm{fst}(v), 0),\\ &\forall v : \mathrm{end}(v) - \mathrm{fst}(v) = 1 \Rightarrow \mathrm{bagof}(v) = [\![v[\mathrm{fst}(v)]]\!]^{(1)},\\ &\forall x, i, j : i \le j \Rightarrow \mathrm{bagof}(\mathrm{const}(x, i, j)) = [\![x]\!]^{(j-i)},\\ &\forall v, k : \mathrm{fst}(v) \le k \le \mathrm{end}(v) \Rightarrow\\ &\quad \mathrm{bagof}(v) = \mathrm{bagof}(v[\mathrm{fst}(v){:}k]) \uplus \mathrm{bagof}(v[k{:}\mathrm{end}(v)])\big\}\end{aligned}$$

Σ_{MAP}-theory $\mathcal{T}_{\mathrm{MAP}}$ of map function on vectors

$$\Sigma^{\mathrm{S}}_{\mathrm{MAP}} = \Sigma^{\mathrm{S}}_{\mathrm{VEC}}$$
$$\Sigma^{\mathrm{F}}_{\mathrm{MAP}} = \Sigma^{\mathrm{F}}_{\mathrm{VEC}} \cup \{f : s \to s' \mid (f{:}s \to s') \in \Sigma^{\mathrm{F}}_{\mathrm{MAP}} \wedge s, s' \in \Sigma^{\mathrm{S}}_{\mathrm{E}}\} \cup \{\mathrm{map}_f : \mathrm{VEC}_s \to \mathrm{VEC}_{s'} \mid (f{:}s \to s') \in \Sigma^{\mathrm{F}}_{\mathrm{MAP}} \wedge s, s' \in \Sigma^{\mathrm{S}}_{\mathrm{E}}\}$$
$$\mathcal{T}_{\mathrm{MAP}} = \big\{\forall v : \mathrm{fst}(\mathrm{map}_f(v)) = \mathrm{fst}(v) \wedge \mathrm{end}(\mathrm{map}_f(v)) = \mathrm{end}(v),\ \forall v, k : \mathrm{fst}(v) \le k < \mathrm{end}(v) \Rightarrow \mathrm{map}_f(v)[k] = f(v[k]) \;\big|\; \mathrm{map}_f \in \Sigma^{\mathrm{F}}_{\mathrm{MAP}}\big\}$$

Fig. 2. Theories of vectors and bags; see Section 3 for details.

Elements. $\mathcal{T}_{\mathrm{E}}$ is a given theory of elements (of vectors and bags). Its signature Σ_{E} is arbitrary but must be disjoint from all other signatures introduced in this section. The theory $\mathcal{T}_{\mathrm{E}}$ is arbitrary, too, but must be decidable and stably infinite so it can be coupled with the theory of multisets, see Section 4.1.

Presburger arithmetic. Σ_{INT} is the signature of Presburger arithmetic, with one sort, two constants and four binary function symbols (for addition, subtraction, minimum and maximum). We introduce the binary predicate symbols $\leq$ and $<$ as abbreviations; we may write $s \leq t$ instead of $\min(s,t) = s$ and $s < t$ instead of $s \leq t \wedge s \neq t$.

The theory $\mathcal{T}_{\mathrm{INT}}$ of Presburger arithmetic is defined as the set of all Σ_{INT}-sentences which are true in $\mathcal{A}_{\mathrm{INT}}$, the standard Σ_{INT}-algebra which interprets the sort INT as the integers and constants and function symbols by their usual meaning.

Multisets. The signature Σ_{BAG} of multisets (with cardinality) extends the signature of Presburger arithmetic with element sorts and multiset sorts BAG_s, one per element sort s. For each element sort, Σ_{BAG} extends Σ_{INT} with a constant $[\![]\!]$ for the empty multiset, a singleton constructor $[\![\cdot]\!]^{(\cdot)}$ (taking an element and its multiplicity), the usual binary operations $\cap$, $\cup$, $\uplus$ for intersection, union and sum, a destructor $\mathrm{count}(\cdot,\cdot)$ for counting the frequency of an element in a multiset, and a destructor $|\cdot|$ for measuring the cardinality (i. e., the number of elements, taking into account their multiplicities) of a multiset. We introduce the binary predicate symbol $\subseteq$ as an abbreviation; we may write $s \subseteq t$ instead of $s \cap t = s$.

Due to the cardinality function, the theory of multisets cannot be finitely axiomatised in our logic.[2] Therefore, the theory $\mathcal{T}_{\mathrm{BAG}}$ of multisets is defined as the set of all Σ_{BAG}-sentences that are true of Δ_{BAG}, the class of standard Σ_{BAG}-algebras. $\mathcal{A}$ is a standard Σ_{BAG}-algebra if it interprets the sort INT as the integers, the sorts BAG_s as the finite multisets over the interpretations of the sorts s, and the constants and function symbols by their usual meanings. Note that the theory $\mathcal{T}_{\mathrm{INT}}$ is contained in $\mathcal{T}_{\mathrm{BAG}}$; stable infiniteness will be relevant in Section 4.1.

Lemma 1. *$\mathcal{T}_{\mathrm{BAG}}$ is stably infinite.*

Vectors. We represent vectors by finite arrays of elements indexed by consecutive integers. The signature Σ_{VEC} of vectors extends the signature of Presburger arithmetic with element sorts and vector sorts VEC_s, one per element sort s. For each element sort, Σ_{VEC} extends Σ_{INT} with two destructors $\mathrm{fst}(\cdot)$ and $\mathrm{end}(\cdot)$ for accessing the first and last (more precisely, the first beyond the last) index of a vector, a destructor $\cdot[\cdot]$ for reading an element of a vector, a constructor $\mathrm{const}(\cdot,\cdot,\cdot)$ for creating a vector filled with a multiple occurrences of the same element, a constructor $\cdot[\cdot{:}\cdot]$ for slicing the subvector in between two indices out of a vector, and a constructor $\cdot\{\cdot \leftarrow \cdot\}$ for updating a vector at an index.

The theory $\mathcal{T}_{\mathrm{VEC}}$ axiomatises vectors. The first axiom is extensionality, equating all vectors that behave equally under the destructors. The remaining axioms define the constructors (uniquely due to extensionality) in terms of the destructors. Note Σ_{VEC} provides no $\mathrm{append}(\cdot,\cdot)$ because $\mathcal{T}_{\mathrm{VEC}}$ forces vector concatenation to be partial.

[2] See [7] for an axiomatisation in a first-order logic extended with an *infinite sum* quantifier.

Given a signature Σ extending Σ_{VEC}, a Σ-algebra $\mathcal{A}$ is called *vector complete* if for all element sorts $s \in \Sigma_{\mathrm{E}}^{\mathrm{S}}$, all integers i, and all finite sequences $x_0, \ldots, x_{k-1} \in s^{\mathcal{A}}$ there is a vector $v \in \mathrm{VEC}_s^{\mathcal{A}}$ such that $\mathrm{fst}(v)^{\mathcal{A}} = i$ and $\mathrm{end}(v)^{\mathcal{A}} = i + k$ and $v[i + l]^{\mathcal{A}} = x_l$ for all integers l with $0 \leq l < k$. A Σ-theory $\mathcal{T}$ is *vector complete* if every $\mathcal{T}$-satisfiable ground $\hat{\Sigma}$-formula has a vector complete model.

Bagof function. The signature Σ_{BAGOF} extends the union of the signature Σ_{VEC} and Σ_{BAG} with functions $\mathrm{bagof}(\cdot)$ mapping vectors to the multisets of their elements. The theory $\mathcal{T}_{\mathrm{BAGOF}}$ axiomatises these functions. The first axiom equates the length of the argument vector with the cardinality of the resulting multiset. The next two axioms define $\mathrm{bagof}(\cdot)$ for the special cases that the argument vector is of length one or constant. The last axiom admits recursive computation of $\mathrm{bagof}(\cdot)$ by splitting the argument into two subvectors and summing the results.

Map function. The signature Σ_{MAP} extends Σ_{VEC} by adding a set F of unary functions on elements (i. e., $(f{:}s \to s') \in F$ implies $s, s' \in \Sigma_{\mathrm{E}}^{\mathrm{S}}$) and a set F_{map} of map functions on vectors such that $(\mathrm{map}_f : \mathrm{VEC}_s \to \mathrm{VEC}_{s'}) \in F_{\mathrm{map}}$ if and only if $(f{:}s \to s') \in F$. Note that Figure 2 specifies Σ_{MAP} by a fixpoint equation which has infinitely many solutions.[3]

The theory $\mathcal{T}_{\mathrm{MAP}}$ axiomatises the functions map_f, in terms of the vector destructors, thus uniquely defining these functions. Note that $\mathcal{T}_{\mathrm{MAP}}$ does not define the unary functions on elements; these functions are intended to be free.

Base theory. We define the Σ-theory $\mathcal{T}_{\mathrm{BASE}} = \mathcal{T}_{\mathrm{E}} \cup \mathcal{T}_{\mathrm{BAG}} \cup \mathcal{T}_{\mathrm{VEC}} \cup \mathcal{T}_{\mathrm{MAP}}$ as the union of the above theories excluding $\mathcal{T}_{\mathrm{BAGOF}}$, where $\Sigma = \Sigma_{\mathrm{E}} \cup \Sigma_{\mathrm{BAG}} \cup \Sigma_{\mathrm{VEC}} \cup \Sigma_{\mathrm{MAP}} \cup \Sigma_{\mathrm{BAGOF}}$ is the union of the above signatures (including Σ_{BAGOF}, i. e., $\mathcal{T}_{\mathrm{BASE}}$ leaves the bagof functions uninterpreted). The following model-theoretic properties will become relevant in Section 5.

Lemma 2. *$\mathcal{T}_{\mathrm{BASE}}$ is vector complete and stably infinite.*

4 Known Decision Procedures Applied to Bags and Vectors

This section employs known results to obtain a decision procedure for ground satisfiability in the combination of the theories of elements, multisets and vectors (including the theory of map functions). We will make repeated use of the following result on the combination of arbitrary theories with free functions.

Proposition 3 (Sofronie-Stokkermans 2005 [9]). *Let $\Sigma' \supseteq \Sigma$ be signatures and let $\mathcal{T}$ be a Σ-theory. If $\mathcal{T}$-satisfiability is decidable for ground $\hat{\Sigma}$-formulae then $\mathcal{T}$-satisfiability is decidable for ground $\hat{\Sigma}'$-formulae.*

[3] Extremal solutions are uninteresting. The least solution would yield $\Sigma_{\mathrm{MAP}} = \Sigma_{\mathrm{VEC}}$, and the greatest solution would likely violate the requirement that Σ_{E} and Σ_{MAP} be disjoint.

The decision procedure behind Proposition 3 reduces a ground $\hat{\Sigma'}$-formula in negation normal form[4] (NNF) to a $\mathcal{T}$-equisatisfiable ground $\hat{\Sigma}$-formula in NNF; the reduction may cause a quadratic blowup.

4.1 Combining the Theories of Elements and Multisets

A decision procedure for the theory $\mathcal{T}_{\mathrm{BAG}}$ of multisets with cardinality is known [7]. We combine this decision procedure with an arbitrary decision procedure for the theory $\mathcal{T}_{\mathrm{E}}$ of elements, using the Nelson-Oppen combination method [6, 11]. This is possible because $\mathcal{T}_{\mathrm{E}}$ and $\mathcal{T}_{\mathrm{BAG}}$ are stably infinite theories (cf. Figure 2 and Lemma 1) over disjoint signatures.

Proposition 4. *Ground $(\mathcal{T}_{\mathrm{E}} \cup \mathcal{T}_{\mathrm{BAG}})$-satisfiability is decidable.*

4.2 Deciding the Theory of Vectors (Including Map)

We use a decision procedure for the Array Property Fragment [3] to decide ground satisfiability in the union of the theories of vectors and map functions. The procedure reduces the satisfiability problem to ground satisfiability in the combination of the theories of Presburger arithmetic, uninterpreted functions and an unspecified theory of vector elements.

Proposition 5. *Let $\mathcal{T}_0$ be a Σ_0-theory where the signature Σ_0 shares no non-constant function symbols with $\Sigma_{\mathrm{VEC}} \cup \Sigma_{\mathrm{MAP}}$ except for the function symbols in Σ_{INT}, formally $\Sigma_0 \cap (\Sigma_{\mathrm{VEC}} \cup \Sigma_{\mathrm{MAP}}) \subseteq \hat{\Sigma_{\mathrm{INT}}}$. Let $\Sigma_1 = \Sigma_0 \cup \Sigma_{\mathrm{INT}} \cup \Sigma_{\mathrm{VEC}} \cup \Sigma_{\mathrm{MAP}}$ and $\mathcal{T}_1 = \mathcal{T}_0 \cup \mathcal{T}_{\mathrm{INT}} \cup \mathcal{T}_{\mathrm{VEC}} \cup \mathcal{T}_{\mathrm{MAP}}$. If $(\mathcal{T}_0 \cup \mathcal{T}_{\mathrm{INT}})$-satisfiability is decidable for ground $\hat{\Sigma}$-formulae, where Σ extends $\Sigma_0 \cup \Sigma_{\mathrm{INT}}$, then $\mathcal{T}_1$-satisfiability is decidable for ground $\hat{\Sigma_1}$-formulae.*

Proof. Let ϕ be a ground $\hat{\Sigma_1}$-formula (in NNF). Perform the following reductions.

1. Eliminate disequalities and updates: Normalise ϕ w. r. t. the rewrite rules NOTEQ and UPDATE from Figure 3. NOTEQ expresses disequalities $s \neq t$ using extensionality and Skolemisation. UPDATE is based on expressing equations $v = u\{i \leftarrow x\}$ by splitting u and v into three subvectors each (a prefix up to index i, a middle part of length 1 at index i and a suffix from index $i+1$) and equating these accordingly (in particular, equating the middle part of v to a constant vector). The resulting ground $\hat{\Sigma_1}$-formula ϕ' is $\mathcal{T}_1$-equisatisfiable to ϕ but contains no vector disequalities and updates.
2. Purify w. r. t. vector sorts: In a bottom up manner, rewrite $\phi'[t]$ to $\phi'[c] \wedge c = t$, where c is a fresh constant and t a non-constant vector term. The result of normalising ϕ' w. r. t. the above rule is a $\mathcal{T}_1$-equisatisfiable $\hat{\Sigma_1}$-formula ϕ'' such that
   - for all terms of the form $\mathrm{fst}(u)$ or $\mathrm{end}(u)$ or $u[i]$, u is a constant, and
   - all vector atoms are of the form $u = v$ or $v = u[i{:}j]$ or $v = \mathrm{const}(x, i, j)$ or $v = \mathrm{map}_f(u)$, where u and v are constants.

[4] The procedure in [9] expects input in clause form; however, the reduction works just as well for formulae in NNF.

$$[\textsc{NotEq}]\ \frac{\phi\big[u \neq v\big]}{\phi\left[\begin{array}{l}\mathrm{fst}(u) \neq \mathrm{fst}(v) \vee \mathrm{end}(u) \neq \mathrm{end}(v) \vee \\ (\mathrm{fst}(u) \leq k < \mathrm{end}(u) \wedge u[k] \neq v[k])\end{array}\right]} \text{ if } u, v \text{ vectors} \wedge k \text{ fresh}$$

$$[\textsc{Read}]\ \frac{\phi\big[u[i]\big]}{\phi\big[x\big] \wedge u[i{:}i+1] = \mathrm{const}(x, i, i+1)} \text{ if } x \text{ fresh}$$

$$[\textsc{Update}]\ \frac{\phi\big[u\{i \leftarrow x\}\big]}{\phi\big[v\big] \wedge \psi(v, u, i, x)} \text{ if } v \text{ fresh}$$

$$\text{where } \psi(v,u,i,x) \equiv \left\{\begin{array}{l}\big(\mathrm{fst}(u) \leq i < \mathrm{end}(u) \vee u = v\big) \wedge \big(i < \mathrm{fst}(u) \vee \mathrm{end}(u) \leq i \vee \\ \quad \big(\mathrm{fst}(v) = \mathrm{fst}(u) \wedge \mathrm{end}(v) = \mathrm{end}(u) \wedge \\ \quad\ v[\mathrm{fst}(u){:}i] = u[\mathrm{fst}(u){:}i] \wedge v[i{:}i+1] = \mathrm{const}(x, i, i+1) \wedge \\ \quad\ v[i+1{:}\mathrm{end}(u)] = u[i+1{:}\mathrm{end}(u)]\big)\big)\end{array}\right.$$

$$[\textsc{Bagof}]\ \frac{\phi\big[\mathrm{bagof}(u)\big]}{\phi\big[b\big] \wedge b = \mathrm{bagof}(u)} \text{ if } b \text{ fresh}$$

$$[\textsc{SubConst}]\ \frac{\phi\big[v = u[k{:}l] \wedge u = \mathrm{const}(x, i, j)\big]}{\phi\big[v = \mathrm{const}(x, \max(k, i), \min(l, j)) \wedge u = \mathrm{const}(x, i, j)\big]}$$

$$[\textsc{MapConst}]\ \frac{\phi\big[v = \mathrm{map}_f(u) \wedge u = \mathrm{const}(x, i, j)\big]}{\phi\big[v = \mathrm{const}(f(x), i, j) \wedge u = \mathrm{const}(x, i, j)\big]}$$

Fig. 3. Vector transformations; see sections 4.2 and 5.1 for details.

3. Eliminate all subterms of the form $\mathrm{fst}(u)$ and $\mathrm{end}(u)$ in ϕ'' by replacing them with INT-constants fst_u and end_u, respectively, introducing two new INT-constants fst_u, end_u per vector constant u. Then normalise ϕ'' w. r. t. all rewrite rules in Figure 4. This results in a $\mathcal{T}_1$-equisatisfiable $\hat{\Sigma}_1$-formula ϕ''', which falls into the Array Property Fragment [3].
4. Use decision procedure for the Array Property Fragment outlined in [3]:
   - Instantiate universal quantifiers in ϕ'''.
   - Replace all constants u of sort VEC_s by unary functions $f_u : \mathrm{INT} \rightarrow s$, and replace all terms of the form $u[i]$ by $f_u(i)$.

   The resulting ground $\hat{\Sigma}$-formula ϕ'''' is $(\mathcal{T}_0 \cup \mathcal{T}_{\mathrm{INT}})$-satisfiable if and only if ϕ''' is $\mathcal{T}_1$-satisfiable, where Σ extends $\Sigma_0 \cup \Sigma_{\mathrm{INT}}$ with the above unary functions f_u and with the unary functions f on element sorts from signature Σ_{MAP}. □

4.3 Deciding the Base Theory

Finally, we pull the results of the previous subsections together to obtain a decision procedure for $\mathcal{T}_{\mathrm{BASE}}$, the union of all theories introduced in Section 3 excluding $\mathcal{T}_{\mathrm{BAGOF}}$. Recall that the signature Σ of $\mathcal{T}_{\mathrm{BASE}}$ includes Σ_{BAGOF}, i. e., $\mathcal{T}_{\mathrm{BASE}}$ treats the bagof functions as free.

$$[\text{EQ}]\ \frac{\phi\big[u = v\big]}{\phi\big[\mathit{fst}_u = \mathit{fst}_v \wedge \mathit{end}_u = \mathit{end}_v \wedge \forall k : \mathit{fst}_u \leq k < \mathit{end}_u \Rightarrow u[k] = v[k]\big]}$$

$$[\text{SUB}]\ \frac{\phi\big[v = u[i{:}j]\big]}{\phi\left[\begin{array}{l}\mathit{fst}_v = \max(i, \mathit{fst}_u) \wedge \mathit{end}_v = \min(j, \mathit{end}_u) \wedge \\ \forall k : \mathit{fst}_v \leq k < \mathit{end}_v \Rightarrow v[k] = u[k]\end{array}\right]}$$

$$[\text{CONST}]\ \frac{\phi\big[v = \text{const}(x, i, j)\big]}{\phi\big[\mathit{fst}_v = i \wedge \mathit{end}_v = j \wedge \forall k : \mathit{fst}_v \leq k < \mathit{end}_v \Rightarrow v[k] = x\big]}$$

$$[\text{MAP}]\ \frac{\phi\big[v = \text{map}_f(u)\big]}{\phi\big[\mathit{fst}_v = \mathit{fst}_u \wedge \mathit{end}_v = \mathit{end}_u \wedge \forall k : \mathit{fst}_v \leq k < \mathit{end}_v \Rightarrow v[k] = f(u[k])\big]}$$

Fig. 4. Translating to the Array Property Fragment; see Section 4.2 for details.

Proposition 6. *Ground $\mathcal{T}_{\text{BASE}}$-satisfiability is decidable.*

Proof. Let ϕ be $\hat{\Sigma}$-formula (in NNF).

1. Reduce ϕ to a $\mathcal{T}$-equisatisfiable ground $\hat{\Sigma'}$-formula ϕ' where $\Sigma' = \Sigma_{\text{E}} \cup \Sigma_{\text{BAG}} \cup \Sigma_{\text{VEC}} \cup \Sigma_{\text{MAP}}$, using the decision procedure for free functions (Proposition 3).
2. Reduce ϕ' to a ground $\hat{\Sigma''}$-formula ϕ'' using the decision procedure for vectors (Proposition 5; the Σ_0-theory $\mathcal{T}_0$ there is $\mathcal{T}_{\text{E}} \cup \mathcal{T}_{\text{BAG}}$ here). The resulting signature Σ'' extends $\Sigma_{\text{E}} \cup \Sigma_{\text{BAG}}$ by free unary functions on element sorts (stemming from signature Σ_{MAP}) and free unary functions from INT to element sorts (arising from encoding arrays as unary functions). The formula ϕ'' is $(\mathcal{T}_{\text{E}} \cup \mathcal{T}_{\text{BAG}})$-satisfiable iff ϕ' is $\mathcal{T}$-satisfiable.
3. Reduce ϕ'' to a $(\mathcal{T}_{\text{E}} \cup \mathcal{T}_{\text{BAG}})$-equisatisfiable ground $\hat{\Sigma'''}$-formula ϕ''' where $\Sigma''' = \Sigma_{\text{E}} \cup \Sigma_{\text{BAG}}$, using the decision procedure for free functions (Proposition 3).
4. Check $(\mathcal{T}_{\text{E}} \cup \mathcal{T}_{\text{BAG}})$-satisfiability of ϕ''' using the combined decision procedure for elements and multisets (Proposition 4). □

5 A Decision Procedure for Bags, Vectors and Bagof Functions

Recall the Σ-theory $\mathcal{T}_{\text{BASE}}$, defined in Section 3 as the union of all theories excluding $\mathcal{T}_{\text{BAGOF}}$, where Σ is the union of all signatures (including Σ_{BAGOF}). For this section, let $\mathcal{T} = \mathcal{T}_{\text{BASE}} \cup \mathcal{T}_{\text{BAGOF}}$ be the Σ-theory extending $\mathcal{T}_{\text{BASE}}$ with the axioms for the bagof functions.

5.1 Decision Procedure

The decision procedure relies on reducing ground $\mathcal{T}$-satisfiability to ground $\mathcal{T}_{\text{BASE}}$-satisfiability by instantiating axioms of $\mathcal{T}_{\text{BAGOF}}$. The reduction is shown in Figure 5. Termination is obvious. Soundness is established by the lemma below.

Input: Ground $\hat{\Sigma}$-formula ϕ_0 (in NNF).
Output: Ground $\hat{\Sigma}$-formula ϕ_6.
Algorithm:

1. Eliminate definable vector operators, purify and simplify:
   (a) Construct ϕ_1 by normalising ϕ_0 w. r. t. the rule NOTEQ (Figure 3).
   (b) Construct ϕ_2 by normalising ϕ_1 w. r. t. the rules READ, UPDATE and BAGOF (Figure 3).
   (c) Construct ϕ_3 by purifying ϕ_2 w. r. t. vector sorts: In a bottom up manner, rewrite $\phi_2[t]$ to $\phi_2[c] \wedge c = t$, where c is a fresh constant and t a non-constant vector term.
   (d) Construct ϕ_4 by converting ϕ_3 into disjunctive normal form (DNF).
   (e) Construct ϕ_5 by normalising ϕ_4 w. r. t. the rules SUBCONST and MAPCONST (Figure 3).
2. Determine the sets of *vector constants* C, *element terms* E and *index terms* I:

$$\begin{aligned} C &= \{v \mid v \text{ vector constant occurring in } \phi_5\} \\ E &= \{x \mid \exists i, j : \mathrm{const}(x, i, j) \text{ occurs in } \phi_5\} \text{ and} \\ I &= \{\mathrm{fst}(u), \mathrm{end}(u) \mid u \in C\} \cup \\ &\quad \{i, j \mid \exists x : \mathrm{const}(x, i, j) \text{ occurs in } \phi_5 \vee \exists u : u[i{:}j] \text{ occurs in } \phi_5\} \ . \end{aligned}$$

3. Instantiate (variants of) the $\mathcal{T}_{\mathrm{BAGOF}}$ axioms with terms generated from C, E and I:

$$\phi_6 \equiv \phi_5 \wedge \bigwedge_{u \in C; i,j \in I} \mathrm{Ax}_1^{u,i,j} \wedge \bigwedge_{x \in E; i,j \in I} \mathrm{Ax}_3^{x,i,j} \wedge \bigwedge_{u \in C; i,j,k \in I} \mathrm{Ax}_4^{u,i,j,k}$$

$$\begin{aligned} \text{where} \quad \mathrm{Ax}_1^{u,i,j} &\equiv \mathrm{fst}(u) \le i \le j \le \mathrm{end}(u) \Rightarrow |\mathrm{bagof}(u[i{:}j])| = j - i \\ \mathrm{Ax}_3^{x,i,j} &\equiv i \le j \Rightarrow \mathrm{bagof}(\mathrm{const}(x, i, j)) = [\![x]\!]^{(j-i)} \\ \mathrm{Ax}_4^{u,i,j,k} &\equiv \mathrm{fst}(u) \le i \le k \le j \le \mathrm{end}(u) \Rightarrow \\ &\quad \mathrm{bagof}(u[i{:}j]) = \mathrm{bagof}(u[i{:}k]) \uplus \mathrm{bagof}(u[k{:}j]) \end{aligned}$$

Fig. 5. Reduction to base theory by instantiating $\mathcal{T}_{\mathrm{BAGOF}}$ axioms.

Lemma 7 (Soundness). *If ϕ_0 is $\mathcal{T}$-satisfiable then ϕ_6 is $\mathcal{T}_{\mathrm{BASE}}$-satisfiable.*

Proof. As ϕ_0 and ϕ_5 are $\mathcal{T}$-equisatisfiable, it suffices to show that every $\mathcal{T}$-model is a model of the instances $\mathrm{Ax}_1^{u,i,j}$, $\mathrm{Ax}_3^{x,i,j}$ and $\mathrm{Ax}_4^{u,i,j,k}$, for all $u \in C$, $x \in E$ and $i, j, k \in I$.

- $\mathrm{Ax}_1^{u,i,j}$ follows from the first $\mathcal{T}_{\mathrm{BAGOF}}$ axiom (after instantiating v with $u[i{:}j]$) as in $\mathcal{T}_{\mathrm{VEC}}$, $\mathrm{fst}(u) \le i \le j \le \mathrm{end}(u)$ implies $\max(\mathrm{end}(u[i{:}j]) - \mathrm{fst}(u[i{:}j]), 0) = j - i$.
- $\mathrm{Ax}_3^{x,i,j}$ is an instance of the third $\mathcal{T}_{\mathrm{BAGOF}}$ axiom.
- $\mathrm{Ax}_4^{u,i,j,k}$ follows from the fourth $\mathcal{T}_{\mathrm{BAGOF}}$ axiom (after instantiating v with $u[i{:}j]$ and k with k) because in $\mathcal{T}_{\mathrm{VEC}}$, the antecedent $\mathrm{fst}(u) \le i \le k \le j \le \mathrm{end}(u)$ implies $u[i{:}j][\mathrm{fst}(u[i{:}j]){:}k] = u[i{:}k]$ and $u[i{:}j][k{:}\mathrm{end}(u[i{:}j])] = u[k{:}j]$. □

Before we show completeness of the reduction, we point out that step 1 converts the input formula ϕ_0 to a ground DNF formula ϕ_5 such that

- bagof occurs only in atoms of the form $b = \mathrm{bagof}(u)$, where b and u are constants,

– all vector atoms are of the form $u = v$ or $v = u[i{:}j]$ or $v = \mathrm{const}(x, i, j)$ or $v = \mathrm{map}_f(u)$, where u and v are constants,
– all other vector terms are of the form $\mathrm{fst}(u)$ or $\mathrm{end}(u)$, where u is a constant, and
– the arguments of map_f are non-constant, i. e., whenever $\mathrm{map}_f(u)$ occurs in a disjunct ψ then there are no terms x, i and j such that the atom $u = \mathrm{const}(x, i, j)$ would logically follow from ψ in theory $\mathcal{T}_{\mathrm{BASE}}$. Note that this last property is achieved by conversion to DNF and propagation of constant vectors within each disjunct (steps 1d and 1e in Figure 5).

5.2 Completeness in the Absence of Map Functions

We call the signature Σ_{MAP} *trivial* if $\Sigma_{\mathrm{MAP}} = \Sigma_{\mathrm{VEC}}$, i. e., there are no unary functions on elements and no map functions. By model-theoretic arguments, we prove completeness of the reduction shown in Figure 5, given that Σ_{MAP} is trivial.

Lemma 8 (Completeness without map**).** *Assume Σ_{MAP} trivial. If ϕ_6 is $\mathcal{T}_{\mathrm{BASE}}$-satisfiable then ϕ_0 is $\mathcal{T}$-satisfiable.*

Proof. Assume a $\hat{\Sigma}$-algebra $\mathcal{A}$ which is a $\mathcal{T}_{\mathrm{BASE}}$-model of ϕ_6; w. l. o. g. we assume that $\mathcal{A}$ is vector complete (cf. Lemma 2). It suffices to construct a $\hat{\Sigma}$-algebra $\mathcal{A}'$ which is a $\mathcal{T}$-model of one disjunct ψ of ϕ_5; we assume that $\mathcal{A} \models \psi$.

Recall that C is the set of vector constants occurring in ϕ_5. We choose $\mathcal{A}'$ so that

1. $\mathcal{A}$ and $\mathcal{A}'$ agree on the interpretations of all sorts, all constants except vector constants occurring in ϕ_5, and all function symbols except the bagof functions,
2. $\mathcal{A}'$ interprets $\mathrm{bagof} : \mathrm{VEC}_s \to \mathrm{BAG}_s$ as functions mapping vectors in $\mathrm{VEC}_s^{\mathcal{A}'}$ to the multisets of their elements in $\mathrm{BAG}_s^{\mathcal{A}'}$,
3. $\mathcal{A}'$ interprets vector constants u occurring in ϕ_5 such that $\mathcal{A}$ and $\mathcal{A}'$ agree
   (a) on the interpretations of the ground terms $\mathrm{fst}(u)$ and $\mathrm{end}(u)$, and
   (b) on the interpretations of the ground term $\mathrm{bagof}(u)$.

We have to explain how the interpretations of vector constants can be chosen in such a way that item (3b) holds, i. e., how to keep the interpretations of ground terms $\mathrm{bagof}(u)$ invariant even though the interpretations of the bagof functions change.

Recall the set of index terms I defined in step 2 of the reduction (Figure 5). Let $\langle i_1, \ldots, i_n \rangle$ be an enumeration of I such that $\mathcal{A}$ orders their interpretations in ascending sequence $i_1^{\mathcal{A}} \leq \cdots \leq i_n^{\mathcal{A}}$. Items (1) to (3a) ensure that $\mathcal{A}$ and $\mathcal{A}'$ agree on the interpretations of index terms $i_j \in I$, hence $\mathcal{A}'$ orders their interpretation $i_j^{\mathcal{A}'}$ in the same sequence.

Item (3b) is achieved by an inductive process. Let $j < n$ be minimal such that there is $u \in C$ with $\mathrm{fst}(u)^{\mathcal{A}} \leq i_j^{\mathcal{A}} \leq i_{j+1}^{\mathcal{A}} \leq \mathrm{end}(u)^{\mathcal{A}}$ and $\mathrm{bagof}(u[i_j{:}i_{j+1}])^{\mathcal{A}}$ differing from the multiset of elements in $u[i_j{:}i_{j+1}]^{\mathcal{A}}$. Note that there can be no $x \in E$ — recall the set E of element terms occurring in ϕ_5 — such that $\mathrm{const}(x, i_j, i_{j+1})^{\mathcal{A}} = u[i_j{:}i_{j+1}]^{\mathcal{A}}$. For if there were such $x \in E$ then the $\mathcal{T}_{\mathrm{BAGOF}}$ instance $\mathrm{Ax}_3^{x, i_j, i_{j+1}}$ (appearing as a conjunct in ϕ_6) would ensure that $\mathrm{bagof}(u[i_j{:}i_{j+1}])^{\mathcal{A}}$ equals the multiset of elements

in $u[i_j{:}i_{j+1}]^{\mathcal{A}}$. Now let C_u be the set of vector constants whose slice between i_j and i_{j+1} happens to equal $u[i_j{:}i_{j+1}]$ in $\mathcal{A}$, formally

$$C_u = \{v \in C \mid \mathcal{A} \models \mathrm{fst}(v) \leq i_j \leq i_{j+1} \leq \mathrm{end}(v) \wedge u[i_j{:}i_{j+1}] = v[i_j{:}i_{j+1}]\} \ .$$

Let $\langle x_0, x_1, \ldots, x_{k-1}\rangle$ be an enumeration of the multiset $\mathrm{bagof}(u[i_j{:}i_{j+1}])^{\mathcal{A}}$. Note that the $\mathcal{T}_{\mathrm{BAGOF}}$ instance $\mathrm{Ax}_1^{u,i_j,i_{j+1}}$ constrains the size of the multiset so that $k = i_{j+1}^{\mathcal{A}} - i_j^{\mathcal{A}}$. As $\mathcal{A}$ is vector complete, we can choose the interpretations of all $v \in C_u$ such that for all $l < k$, $v^{\mathcal{A}'}$ stores x_l at index $i_j^{\mathcal{A}'} + l$. This ensures that $\mathcal{A}' \models u[i_j{:}i_{j+1}] = v[i_j{:}i_{j+1}]$. The construction proceeds from there by induction on j.

After the construction is completed, one can show that $\mathcal{A}$ and $\mathcal{A}'$ do in fact agree on the interpretation of $\mathrm{bagof}(u)$, for all $u \in C$. The proof is by induction on the length $\mathrm{end}(u) - \mathrm{fst}(u)$ of u and uses the $\mathcal{T}_{\mathrm{BAGOF}}$ instances $\mathrm{Ax}_4^{u,i,j,k}$, for all $i, j, k \in I$ such that $\mathcal{A} \models \mathrm{fst}(u) \leq i \leq k \leq j \leq \mathrm{end}(u)$.

Obviously, $\mathcal{A}'$ is a model of $\mathcal{T}_{\mathrm{BAGOF}}$ (and thus of $\mathcal{T}$) as that is how the interpretation of the bagof functions was chosen. To show that $\mathcal{A}' \models \psi$, it suffices to show that $\mathcal{A}'$ satisfies every vector atom that $\mathcal{A}$ satisfies (because $\mathcal{A}$ and $\mathcal{A}'$ agree on the interpretation of non-vector literals and all vector literals occurring in ψ are positive). In the case of atoms of the form $v = \mathrm{const}(x, i, j)$ this is so because the construction does not change the interpretation of v. In the case of atoms of the form $u = v$ or $v = u[i{:}j]$, the construction alters the interpretations of corresponding slices of u and v uniformly. □

The decidability of ground satisfiability in the theories of elements, multisets, vectors (excluding map functions) and the bagof function follows from soundness and completeness of the reduction (lemmas 7 and 8) and from decidability of the base theory (Proposition 6).

Theorem 9. *Assume Σ_{MAP} trivial. Then ground $\mathcal{T}$-satisfiability is decidable.*

We remark that the conversion to DNF (step 1d in Figure 5) during the reduction is not necessary if Σ_{MAP} is trivial; NNF is all that's required in that case.

5.3 Completeness in the Presence of Map Functions

To prove completeness of the reduction from Figure 5 when Σ_{MAP} is not trivial, we need syntactic restrictions on the occurrences of map functions in the input formula.

Given a set of element sorts $S \subseteq \Sigma_{\mathrm{E}}^{\mathrm{S}}$, we say a term t is a *S-term* (resp. *VEC_S-term*) if t is a s-term (resp. VEC_s-term) for some $s \in S$. A ground $\hat{\Sigma}$-formula ϕ is *stratified* if there is a partition $\{S_1, \ldots, S_m\}$ of the set of element sorts $\Sigma_{\mathrm{E}}^{\mathrm{S}}$ such that

- for every subterm $\mathrm{map}_f(u)$ of ϕ there are strata S_i and S_{i+1} such that u is a VEC_{S_i}-term and $\mathrm{map}_f(u)$ is a $\mathrm{VEC}_{S_{i+1}}$-term, and
- all arguments of $\mathrm{bagof}(\cdot)$ in ϕ are uniformly VEC_{S_m}-terms.

The verification condition VC from Figure 1 is an example of a stratified formula. Given the strata $S_1 = \{\texttt{String}\}$ and $S_2 = \{\texttt{Resource}\}$, it is easy to check that $\mathrm{map}_{\texttt{MessageResource}}$ maps vectors of strings to vectors of resources, and that all arguments of $\mathrm{bagof}(\cdot)$ are vectors of resources. On the other hand, a formula containing a function symbol $\mathrm{map}_f : \mathrm{VEC}_s \to \mathrm{VEC}_{s'}$ fails to be stratified if $s = s'$, for instance.

Lemma 10 (Completeness for stratified input). *Assume ϕ_0 stratified. If ϕ_6 is $\mathcal{T}_{\mathrm{BASE}}$-satisfiable then ϕ_0 is $\mathcal{T}$-satisfiable.*

Proof (Sketch). Let $S_1, \ldots, S_m$ be the strata for ϕ_0. As stratification is preserved by step 1 of the reduction, ϕ_5 is stratified w. r. t. the same strata. Recall the set C of vector constants defined in step 2 of the reduction. Stratification induces a partition $\{C_1, \ldots, C_m\}$ of C such that each C_i contains the VEC_{S_i}-constants occurring in ϕ_5. We modify step 3 of the reduction slightly by generating instances of $\mathrm{Ax}_1^{u,i,j}$ and $\mathrm{Ax}_4^{u,i,j,k}$ only for $u \in C_m$.

Now, assume a $\hat{\Sigma}$-algebra $\mathcal{A}$ (which due to Lemma 2 can be assumed vector complete and stably infinite[5]) which is a $\mathcal{T}_{\mathrm{BASE}}$-model of ϕ_6. The construction of a $\mathcal{T}$-model $\mathcal{A}'$ of a disjunct ψ of ϕ_5 is similar to the one in Lemma 8 except for the fact that now $\mathcal{A}'$ may not only change the interpretations of $\mathrm{bagof}(\cdot)$ and of vector constants but also the interpretations of function symbols from signature Σ_{MAP}. The construction proceeds in m phases, yielding a sequence $\langle \mathcal{A}_m, \mathcal{A}_{m-1}, \ldots, \mathcal{A}_1 \rangle$ of $\hat{\Sigma}$-algebras.

The first phase constructs a $\hat{\Sigma}$-algebra $\mathcal{A}_m$ fixing the interpretations of the bagof functions and the vector constants in C_m; this construction is analogous to the proof of Lemma 8. Changing the interpretation some constant $v \in C_m$ may falsify some atom of the form $v = \mathrm{map}_f(u)$. To rectify this, the second phase constructs a $\hat{\Sigma}$-algebra $\mathcal{A}_{m-1}$ fixing the interpretations of vector constants in C_{m-1} (and possibly changing the interpretations of functions in Σ_{MAP}) in order to restore the truth of $v = \mathrm{map}_f(u)$. This in turn may falsify some other map atom, whose truth is restored by constructing $\mathcal{A}_{m-2}$, and so on.

We present the construction of $\mathcal{A}_{m-1}$ in more detail; recall that we assume that $\mathcal{A} \models \psi$, and that ψ is a conjunction of literals. Let $\langle i_1, i_2, \ldots, i_n \rangle$ be the ascending enumeration of index terms as defined in the proof of Lemma 8. Let $j < n$ be minimal such that ψ contains some atom $v = \mathrm{map}_f(u)$ with $\mathcal{A}_m \not\models v[i_j{:}i_{j+1}] = \mathrm{map}_f(u[i_j{:}i_{j+1}])$. Because $\mathcal{A}$ and $\mathcal{A}_m$ essentially differ in the interpretations of vector constants in C_m, we conclude that $v \in C_m$, hence $u \in C_{m-1}$ due to stratification. In $\mathcal{A}_{m-1}$, we change the interpretation of u (and of all u' with $\mathcal{A}_m \models u'[i_j{:}i_{j+1}] = u[i_j{:}i_{j+1}]$) such that the elements of $u[i_j{:}i_{j+1}]^{\mathcal{A}_{m-1}}$ are fresh and pairwise distinct. Freshness means that the elements of $u[i_j{:}i_{j+1}]^{\mathcal{A}_{m-1}}$ occur neither in the $\mathcal{A}$- nor in the $\mathcal{A}_m$-interpretation of any element or vector constant. Because $\mathcal{A}$ and $\mathcal{A}_m$ (which features the same carriers) are stably infinite and vector complete, we can always find enough fresh elements and create arbitrary vectors from them. Next, we change the interpretation of the free function f. Define $f^{\mathcal{A}_{m-1}}$ such that $f^{\mathcal{A}_{m-1}}(u^{\mathcal{A}_{m-1}}[l]) = v^{\mathcal{A}_{m-1}}[l]$, for all integers l with $i_j^{\mathcal{A}_{m-1}} \leq l < i_{j+1}^{\mathcal{A}_{m-1}}$. Due to freshness of the elements in $u[i_j{:}i_{j+1}]^{\mathcal{A}_{m-1}}$, the function $f^{\mathcal{A}_{m-1}}$ is well-defined. The construction proceeds by induction on j.

It is obvious that $\mathcal{A}_{m-1} \models v[i_j{:}i_{j+1}] = \mathrm{map}_f(u[i_j{:}i_{j+1}])$. What remains to be shown is that the construction preserves the truth of other vector atoms occurring in ψ. In the case of atoms of the form $u' = u$ or $u' = u[i{:}j]$, the argument is the same as in the proof of Lemma 8: Both sides are altered uniformly. Finally, the case of atoms of the form $u = \mathrm{const}(x, i, j)$ cannot arise because if it did then step 1e of the reduction would

[5] By abuse of notation, we call a Σ-algebra $\mathcal{A}$ *stably infinite* if all its carriers are infinite.

have propagated the constant vector through map_f, replacing the atom $v = \mathrm{map}_f(u)$ with $v = \mathrm{const}(f(x), i, j)$. □

The decidability of satisfiability of stratified ground formulae in the theories of elements, multisets, vectors, map functions and the bagof function follows; the proof is similar to Theorem 9.

Theorem 11. *Ground $\mathcal{T}$-satisfiability is decidable for stratified ground $\hat{\Sigma}$-formulae.*

Relation to local theory extensions. The way the reduction in Figure 5 instantiates universal quantifiers with selected ground terms is reminiscent of local theory extensions [5], and one may wonder whether the theory $\mathcal{T}$ can be viewed as a local extension of the theory $\mathcal{T}_{\mathrm{BASE}}$. However, our model construction does not fit entirely into the framework of local theory extensions because not only does it extend partial extension functions (like the bagof functions) to total ones but also changes the interpretations of base constants and free base functions. It remains to be seen whether the framework of local theory extensions can be suitably generalised to encompass our construction.

Acknowledgements. This work was funded in part by the Sixth Framework programme of the European Community under the MOBIUS project FP6-015905. This paper reflects only the author's views and the European Community is not liable for any use that may be made of the information contained therein.

References

[1] D. Aspinall, P. Maier, and I. Stark. Monitoring external resources in Java MIDP. *Electr. Notes Theor. Comput. Sci.*, 197(1):17–30, 2008.

[2] D. Aspinall, P. Maier, and I. Stark. Safety guarantees from explicit resource management. In *Proc. FMCO 2007*, LNCS 5382, pages 52–71. Springer, 2008.

[3] A. R. Bradley, Z. Manna, and H. B. Sipma. What's decidable about arrays? In *Proc. VMCAI 2006*, LNCS 3855, pages 427–442. Springer, 2006.

[4] S. Ghilardi, E. Nicolini, S. Ranise, and D. Zucchelli. Decision procedures for extensions of the theory of arrays. *Ann. Math. Artif. Intell.*, 50(3-4):231–254, 2007.

[5] C. Ihlemann, S. Jacobs, and V. Sofronie-Stokkermans. On local reasoning in verification. In *Proc. TACAS 2008*, LNCS 4963, pages 265–281. Springer, 2008.

[6] G. Nelson and D. C. Oppen. Simplification by cooperating decision procedures. *ACM Trans. Program. Lang. Syst.*, 1(2):245–257, 1979.

[7] R. Piskac and V. Kuncak. Decision procedures for multisets with cardinality constraints. In *Proc. VMCAI 2008*, LNCS 4905, pages 218–232. Springer, 2008.

[8] R. Piskac and V. Kuncak. Linear arithmetic with stars. In *Proc. CAV 2008*, LNCS 5123, pages 268–280. Springer, 2008.

[9] V. Sofronie-Stokkermans. Hierarchic reasoning in local theory extensions. In R. Nieuwenhuis, editor, *Proc. CADE-20*, LNCS 3632, pages 219–234. Springer, 2005.

[10] N. Suzuki and D. Jefferson. Verification decidability of presburger array programs. *J. ACM*, 27(1):191–205, 1980.

[11] C. Tinelli and C. G. Zarba. Combining decision procedures for sorted theories. In *Proc. JELIA 2004*, LNCS 3229, pages 641–653. Springer, 2004.

[12] C. G. Zarba. Combining multisets with integers. In *Proc. CADE-18*, LNCS 2392, pages 363–376. Springer, 2002.