Arrays, Maps, and Vectors With Abstract Domain for SMT[1]

Maria Paola Bonacina

Dipartimento di Informatica
Università degli Studi di Verona
Verona, Italy, EU

BIRS Workshop on Theory and Practice of SAT and Combinatorial Solving
Banff International Research Station
Banff, Alberta, Canada

15 January 2026

[1] Joint work with S. Graham-Lengrand and N. Shankar

A mainstay in SMT: the theory of arrays

- Basic operations: read/write or select/store
- Sorts: indices, values, arrays
- Select-over-store axioms [McCarthy 1993]:
  $\forall\ a,\ v,\ i.\ \text{select}(\text{store}(a,i,v),i) \simeq v$
  $\forall\ a,\ v,\ i,\ j.\ i \not\simeq j \rightarrow \text{select}(\text{store}(a,i,v),j) \simeq \text{select}(a,j)$
- Extensionality axiom:
  $\forall\ a,\ b.\ (\forall i.\ \text{select}(a,i) \simeq \text{select}(b,i)) \rightarrow a \simeq b$
- Not decidable, but the quantifier-free fragment (QFF) is [Stump, Barrett, Dill, Levitt 2001]
- Useful to reason about computer memory (e.g., heap)

Arrays: finite or infinite?

Programming languages:

- Integer-indexed arrays
- Finite: indices in the interval $[0, n-1]$, length n
  Ada: indices in the interval $[n, m]$, length $m - n + 1$
- A store within bounds works, error otherwise

Theory of arrays:

- All arrays have the same length given by the cardinality of the set used to interpret the sort of indices
- If integer-indexed: infinite arrays
- No distinction btw in-bounds and out-of-bounds store

Adding quantified formulas to the QFF

- Array property fragment (APF)
- Limited usage of $\forall$ over index variables
- Integer-index arrays
- Bounded array equality: $beq(a, b, l, u)$ iff
  $\forall\ i.\ l \le i \le u \ \rightarrow\ \mathrm{select}(a, i) \ \simeq\ \mathrm{select}(b, i)$
- APF is decidable: finitely many instances of $\forall$ +
  decision procedure for combination of arrays, integers, values

[Bradley, Manna, Sipma 2006] [Bradley, Manna 2007] [Ge, de Moura 2009]

The theory of arrays is unchanged: its limitations remain

Using finite sequences to model finite arrays

- Theory of sequences
- Empty sequence, binary associative concatenation: a monoid
- Unary constructor wrapping single element into sequence
- Extract or Slice: returns subsequence btw two positions
- Access: returns element at given position (similar to select)
- Length $|x|$: number of elements in sequence x

[Bjørner et al. 2012] [Jeż et al. 2023]

Theories of finite sequences to model finite arrays

- Theory Seq with integer indices $[0, |x|)$ and countably infinite element sort [Sheng et al. 2023]:
  - Add update function: access/update for select/store
  - Extensionality: same length n and same elements in $[0, n)$
  - Update axiom: update does not change length
    only an update in $[0, |x|)$ modifies the element
- Theory N-Seq [Ait-El-Hara, Bobot, Bury 2024] [Ait-El-Hara 2025] to model Ada arrays:
  - The first index is not necessarily 0
  - Add other functions: e.g., relocate
- Decidability of QFF: unknown
  Sound inference systems, neither termination, nor completeness

Quantifiers and sequences: APF with concatenation

- Arrays interpreted as finite integer-indexed sequences
- Add repeat: takes element e and length n and produces e^n
- Obtain update by concatenating a slice, e^1, and another slice
- More expressive than APF: allows index shifting (e.g., $a[i]$ and $a[i+n]$), concatenation can be defined
- Undecidable: halting problem of a two-register machine
- Decision procedure for certain formulas

[Wang, Appel 2023]

Summary: when addressing the limitations of the theory of arrays, it is easy to lose decidability; SMT with $\forall$ still a challenge

Adding a length function to the theory of arrays

- Maps every array to its length: $\mathsf{len}(a) \simeq n$
- Axiom of extensionality for integer-indexed arrays:
  $\forall\ a,\ b.\ [\ \mathsf{len}(a)\ \simeq\ \mathsf{len}(b) \wedge$
  $(\forall i.\ 0 \leq i < \mathsf{len}(a) \rightarrow \mathsf{select}(a, i)\ \simeq\ \mathsf{select}(b, i))\ ]\ \rightarrow\ a\ \simeq\ b$
- Arrays and integers share $<$ ... no longer disjoint theories
- Bridging functions [Sofronie-Stokkermans 2009] and bridging axioms [Ganzinger, Rueß, Shankar 2004]
- Most combination methods require disjoint theories (only shared symbol: $\simeq$)
- Seq, N-Seq, and APFC avoid non-disjoint combination by reasoning in terms of reduction to a base theory

Solution: a theory of arrays with abstract domain

- Neither quantifiers nor sequences
- Enrich the theory of arrays itself
- Abstract domain: indices do not have to be integers, nor even linearly ordered
- Also maps, and vectors meaning dynamic arrays
- View the problem as non-disjoint theory combination
- Extend the theory combination method CDSAT to predicate-sharing theories: soundness, termination, completeness
- The QFF is decidable: follows from fitting the three theories in CDSAT + termination and completeness of CDSAT

The theory of arrays with abstract domain: signature

- ArrAD: theory of arrays with abstract domain
- Sorts: indices I, values V, arrays A, lengths L, and $Prop$
- $\mathsf{select}\colon A \times I \to V$ $\quad$ $\mathsf{store}\colon A \times I \times V \to A$ $\quad$ $\mathsf{len}\colon A \to L$
- Free admissibility predicate: $\mathsf{Adm}\colon I \times L \to Prop$
  $\mathsf{Adm}(i, l)$: index i is admissible wrt length l
- Abstract domain: definition of Adm
- Concrete domain: set of admissible indices given Adm's definition and the interpretation of I
- Adm is shared with another theory $\mathcal{T}$ that defines it

The theory of arrays with abstract domain: axioms

- Select-over-store axioms:
  - $\forall\ a,\ v,\ i.\ \mathrm{select}(\mathrm{store}(a,i,v),i) \simeq v$ is replaced by
    $\forall\ a,\ v,\ i.\ \mathrm{Adm}(i,\mathrm{len}(a)) \rightarrow \mathrm{select}(\mathrm{store}(a,i,v),i) \simeq v$
    a store at an inadmissible index has no effect
  - $\forall\ a,\ v,\ i,\ j.\ i \not\simeq j \rightarrow \mathrm{select}(\mathrm{store}(a,i,v),j) \simeq \mathrm{select}(a,j)$
- Store does not change length:
  $\forall\ a,\ i,\ v.\ \mathrm{len}(\mathrm{store}(a,i,v)) \simeq \mathrm{len}(a)$
- Extensionality:
  $\forall\ a,\ b.\ [\ \mathrm{len}(a) \simeq \mathrm{len}(b)\ \wedge$
  $(\forall i.\ \mathrm{Adm}(i,\mathrm{len}(a)) \rightarrow \mathrm{select}(a,i) \simeq \mathrm{select}(b,i))\ ]$
  $\rightarrow\ a \simeq b$

The most common interpretation of admissibility

- Let LIA be the theory defining Adm
- Say LIA interprets indices as integers
  lengths as integers
  and defines Adm by

$$\forall\ i,\ \ n.\ \ \mathrm{Adm}(i, n)\ \leftrightarrow\ 0 \le i < n$$

- The set of admissible indices is the interval $[0, n)$
- Under this interpretation extensionality in ArrAD covers
  - Extensionality for integer-index arrays with length
  - Extensionality in the theory Seq of sequences and in APFC

Admissibility captures bounded equality as in APF

- Let LIA be the theory defining Adm
- Say LIA interprets indices as integers
  lengths as pairs of integers
  and defines Adm by

$$\forall\ i,\ l,\ u.\ \mathrm{Adm}(i,(l,u)) \ \leftrightarrow\ l \leq i \leq u$$

- The set of admissible indices is the interval $[l, u]$
- Under this interpretation extensionality in ArrAD covers
  - Bounded equality in APF
  - Extensionality in the theory N-Seq of sequences

Admissibility captures array equality in programming

- Let $\mathcal{T}$ be the theory defining Adm
- Say $\mathcal{T}$ interprets indices as integers, lengths as pairs $(addr, n)$:
  $addr$ is a binary number: the starting address
  $n \geq 0$: the number of admissible indices
  and defines Adm by
  $$\forall\ i,\ addr,\ n.\ \mathrm{Adm}(i, (addr, n))\ \leftrightarrow\ 0 \leq i < n$$
  where the starting address plays no role
- Two arrays a and b with
  same interval of admissible indices, say $[0, 5)$
  but $\mathrm{len}(a) = (000100, 5)$ and $\mathrm{len}(b) = (010100, 5)$
  are different

Admissibility as generic set membership

- Let $\mathcal{T}$ be the theory defining Adm
- Say $\mathcal{T}$ interprets the sort of indices as a set S
  the sort of lengths as the powerset of S
  and defines Adm by
  $$\forall i,\ N.\ \mathrm{Adm}(i, N)\ \leftrightarrow\ i \in N$$
- The set of admissible indices is the subset $N \subseteq S$

The set S does not have to be a set of numbers
does not have to be linearly ordered
does not have to be ordered

A theory of maps with abstract domain

- MapAD: theory of maps with abstract domain
- Store at inadmissible index i makes i admissible:
  $\forall\ a,\ j,\ i,\ v.\ \mathsf{Adm}(j, \mathsf{len}(\mathsf{store}(a, i, v))) \leftrightarrow (\mathsf{Adm}(j, \mathsf{len}(a)) \vee j \simeq i)$
- Store does not change length if the index is admissible:
  $\forall\ a,\ i,\ v.\ \mathsf{Adm}(i, \mathsf{len}(a)) \rightarrow \mathsf{len}(\mathsf{store}(a, i, v)) \simeq \mathsf{len}(a)$
- Select-over-store axioms:
  - Restored: $\forall\ a,\ v,\ i.\ \mathsf{select}(\mathsf{store}(a, i, v), i) \simeq v$
  - $\forall\ a,\ v,\ i,\ j.\ i \not\simeq j \rightarrow \mathsf{select}(\mathsf{store}(a, i, v), j) \simeq \mathsf{select}(a, j)$
- Extensionality unchanged: $\forall a,\ b.\ [\ \mathsf{len}(a) \simeq \mathsf{len}(b) \wedge$
  $(\forall i.\ \mathsf{Adm}(i, \mathsf{len}(a)) \rightarrow \mathsf{select}(a, i) \simeq \mathsf{select}(b, i))\ ] \rightarrow a \simeq b$

A theory of vectors (dynamic arrays) with abstract domain

- VecAD: theory of vectors with abstract domain
- Store at an inadmissible index i makes i and the indices smaller than i admissible:
  $\forall\ a,\ j,\ i,\ v.\ \mathsf{Adm}(j, \mathsf{len}(\mathsf{store}(a, i, v))) \ \leftrightarrow\ (\mathsf{Adm}(j, \mathsf{len}(a)) \vee j \leq i)$
- Everything else as in MapAD
  except for adding to the signature an ordering $<$ on indices (does not have to be linear)

MapAD and VecAD: dynamic data structures modeled for the first time

Reasoning about ArrAD, MapAD and VecAD?

CDSAT

CDSAT: Conflict-Driven SATisfiability in n theories

- Orchestrates theory modules in a conflict-driven model search
- The theory modules work on a shared trail: not a stack
- Generalizes MCSAT to theory combination:
  - Assignments of values to terms: both Boolean and first-order
  - Theory conflict explanation by theory inferences that can generate new terms
- Propositional logic is one of the theories: no hierarchy btw Boolean reasoning and theory reasoning
- Input first-order assignments: Satisfiability Modulo Assignment
- Sound, terminating, and complete for predicate-sharing theories without requiring stable infiniteness

How to fit a component theory in CDSAT?

- A theory module $\mathcal{I}_k$ for theory $\mathcal{T}_k$: an inference system (abstraction of a decision procedure)
- Requirements on a theory module:
  - Soundness (for the soundness of CDSAT)
  - Finite local basis: $\text{basis}_k(X)$ – all the terms that $\mathcal{I}_k$ can generate from set X of input terms
    Used to construct the finite global basis for the theory union (for the termination of CDSAT)
  - Completeness(for the completeness of CDSAT):
    - Leading theory $\mathcal{T}_1$: has all sorts and all shared predicates
    - Leading theory $\mathcal{T}_1$: $\mathcal{I}_1$ is complete
    - All other theories $\mathcal{T}_k$: $\mathcal{I}_k$ is leading-theory complete

Theory modules for ArrAD, MapAD, VecAD

- From axioms to inference rules, e.g.:
  - $n \simeq m,\ i \simeq j,\ \mathrm{Adm}(i,n),\ \neg\,\mathrm{Adm}(j,m)\ \vdash\ \bot$
  - $a \simeq b\ \vdash\ \mathrm{len}(a) \simeq \mathrm{len}(b)$
  - $b \simeq \mathrm{store}(a,i,v),\ \mathrm{len}(b) \not\simeq \mathrm{len}(a)\ \vdash\ \bot$
    for ArrAD
  - $\mathrm{len}(a) \simeq n,\ \mathrm{Adm}(i,n),\ b \simeq \mathrm{store}(a,i,v),\ \mathrm{len}(b) \not\simeq \mathrm{len}(a)\ \vdash\ \bot$
    for MapAD and VecAD
- Some rules generate $\bot$ (conflict detection) others do not: balancing finite local basis design and completeness
- A finite local basis for ArrAD, MapAD, VecAD

Interpretation of arrays with abstract domain

Interpretation of arrays:

- An array: a function from indices to values
- Sort of arrays: an updatable function set X:
  g differs from $f \in X$ at finitely many indices: $g \in X$

Interpretation of arrays with abstract domain:

- An array of length n: a function from the set I_n of admissible indices for length n to values
- Sort of arrays: a collection of updatable function sets $(X_n)_n$ one for each n in the interpretation of the sort L of lengths

Interpretation of maps with abstract domain

- A map of length n: a function from the set I_n of admissible indices for length n to values
- Sort of maps: an incrementally updatable collection of function sets $(X_n)_n$:
  one for each n in the interpretation of the sort L of lengths
  g differs from $f \in X_n$ at finitely many indices: $\exists m,\ g \in X_m$
- Either $m = n$: store at an admissible index
- Or $I_m = I_n \cup \{i\}$: store at an inadmissible index i that is admissible in the resulting map

Interpretation of vectors with abstract domain

- A vector of length n: a function from the set I_n of admissible indices for length n to values
- Sort of vectors: an extensibly updatable collection of function sets $(X_n)_n$:
  one for each n in the interpretation of the sort L of lengths
  g differs from $f \in X_n$ at finitely many indices: $\exists m,\ g \in X_m$
- Either $m = n$: store at an admissible index
- Or $I_m = I_n \cup \{j \mid j \leq i\}$: store at an inadmissible index i that is admissible in the resulting vector together with the smaller indices

Leading-theory-completeness for ArrAD

- **Theorem:** the module for ArrAD is leading-theory-complete for all ArrAD-suitable leading theories $\mathcal{T}_1$
- A leading theory $\mathcal{T}_1$ is ArrAD-suitable if:
  - $\mathcal{T}_1$ has all the sorts of ArrAD
  - $\mathcal{T}_1$ shares with ArrAD equality and Adm
  - For all $\mathcal{T}_1$-models $\mathcal{M}_1$ there exists a collection of updatable function sets $(X_n)_n$ such that
    - n ranges over all possible values for lengths according to $\mathcal{M}_1$
    - $f \in X_n$ is a function from admissible indices to values in the $\mathcal{M}_1$-interpretation of indices, admissibility, and values
    - The sum of the cardinalities of the X_n determines the cardinality of the sort A of arrays in $\mathcal{M}_1$
- Suitability does not restrict combinability

Leading-theory-completeness for MapAD

- **Theorem:** the module for MapAD is leading-theory-complete for all MapAD-suitable leading theories $\mathcal{T}_1$
- A leading theory $\mathcal{T}_1$ is MapAD-suitable if:
  - $\mathcal{T}_1$ has all the sorts of MapAD
  - $\mathcal{T}_1$ shares with MapAD equality and Adm
  - For all $\mathcal{T}_1$-models $\mathcal{M}_1$ there exists an incrementally updatable collection of function sets $(X_n)_n$ such that
    - n ranges over all possible values for lengths according to $\mathcal{M}_1$
    - $f \in X_n$ is a function from admissible indices to values in the $\mathcal{M}_1$-interpretation of indices, admissibility, and values
    - The sum of the cardinalities of the X_n determines the cardinality of the sort A of maps in $\mathcal{M}_1$

Leading-theory-completeness for VecAD

- **Theorem:** the module for VecAD is leading-theory-complete for all VecAD-suitable leading theories $\mathcal{T}_1$
- A leading theory $\mathcal{T}_1$ is VecAD-suitable if:
  - $\mathcal{T}_1$ has all the sorts of MapAD
  - $\mathcal{T}_1$ shares with MapAD equality, Adm, and $<$
  - For all $\mathcal{T}_1$-models $\mathcal{M}_1$ there exists an extensibly updatable collection of function sets $(X_n)_n$ such that
    - n ranges over all possible values for lengths according to $\mathcal{M}_1$
    - $f \in X_n$ is a function from admissible indices to values in the $\mathcal{M}_1$-interpretation of indices, admissibility, and values
    - The sum of the cardinalities of the X_n determines the cardinality of the sort A of maps in $\mathcal{M}_1$

Future work

- Add concatenation (may subsume sequences): QF decidability to be determined
- Other theories and bridging functions: appropriate shared predicates and CDSAT modules
- QSMA(CDSAT) (for quantified satisfiability)
- Implementation ... AR SW crisis!

References

- Conflict-driven satisfiability for theory combination: transition system and completeness.
  JAR 64(3):579–609, 2020 (Conference version at CADE 2017)
- Conflict-driven satisfiability for theory combination: modules, lemmas, and proofs.
  JAR 66(1):43–91, 2022 (Conference version at CPP 2018)
- The CDSAT method for satisfiability modulo theories and assignments: an exposition.
  Proc. CiE-21, LNAI 15764, 1–16, Springer, July 2025.
- CDSAT for predicate-sharing theories: arrays, maps, and vectors with abstract domain.
  Submitted to a journal, 45 pages (Short version at SMT 2022)

Authors: MPB, S. Graham-Lengrand, and N. Shankar

Thank you!