Reasoning about Arrays

Aaron R. Bradley

CU Boulder

```
max := a[l];
for(i := l+1; i <= u; i++)
  if (a[i] > max) max := a[i];
```

Establish postcondition:

$$\forall j.\, \mathtt{l} \leq j \leq \mathtt{u} \rightarrow \mathtt{a}[j] \leq \mathtt{max}$$

Loop invariant:

$$\forall j.\, \mathtt{l} \leq j < \mathtt{i} \rightarrow \mathtt{a}[j] \leq \mathtt{max}$$

Also establish postcondition:

$$\forall j.\, \mathtt{a}[j] = \mathtt{a}_0[j]$$

```
for(i := 0; i < length(sets); i++)
  u := union(u, sets[i]);
```

Given postcondition of `union`$(u,\ v)$:

$$rv = u \cup v$$

Establish postcondition:

$$\forall j.\ 0 \leq j \leq |\texttt{sets}| \ \rightarrow\ \texttt{sets}[j] \subseteq \texttt{u}$$

Loop invariant:

$$\forall j.\ 0 \leq j < \texttt{i} \ \rightarrow\ \texttt{sets}[j] \subseteq \texttt{u}$$

Loop invariant:

$$\forall j.\, 0 \leq j < \mathtt{i} \;\rightarrow\; \mathtt{sets}[j] \subseteq \mathtt{u}$$

Translation:

$$\forall j.\, 0 \leq j < \mathtt{i} \;\rightarrow\; \forall e.\, \mathtt{sets}[j][e] \;\rightarrow\; \mathtt{u}[e]$$

Sets

- $e \in s$: $s[e]$
- $s \subseteq t$ $\forall e.\, s[e] \rightarrow t[e]$
- $s \subset t$ $(\forall e.\, s[e] \rightarrow t[e]) \wedge (\exists e_1.\, \neg s[e_1] \wedge t[e_1])$
- $s = t \cap u$ $\forall e.\, s[e] \leftrightarrow t[e] \wedge u[e]$
- $s = \overline{t}$ $\forall e.\, s[e] \leftrightarrow \neg t[e]$

Multisets (bags)

- $C(s, e)$ $s[e]$
- $s = t \uplus u$ $\forall e.\, s[e] = t[e] + u[e]$
- $\ldots$

```
assert(v >= 0);
ht := put(ht, k, v);
```

Precondition:

$$\forall j \in \mathsf{keys}(\mathtt{ht}).\, \mathsf{get}(\mathtt{ht}, j) \geq 0$$

Establish postcondition:

$$\forall j \in \mathsf{keys}(\mathtt{ht}).\, \mathsf{get}(\mathtt{ht}, j) \geq 0$$

Verification condition:

$$\begin{aligned} & (\forall j \in \mathsf{keys}(\mathtt{ht}).\, \mathsf{get}(\mathtt{ht}, j) \geq 0) \\ \wedge\ & \mathtt{v} \geq 0 \ \wedge\ h' = \mathsf{put}(\mathtt{ht}, \mathtt{k}, \mathtt{v}) \\ \rightarrow\ & (\forall j \in \mathsf{keys}(h').\, \mathsf{get}(h', j) \geq 0) \end{aligned}$$

Flat data structures

- Integer-indexed arrays
- Collections: sets, multisets (bags)
- Hashtables

Model and reason about them as arrays (uninterpreted functions).

First-Order Theory

$T : (\Sigma, \mathcal{A})$

- **Signature** Σ: non-logical symbols (a, b, $+$, $<$, ...)
- **Axioms** $\mathcal{A}$: formulae interpreting symbols

T-Interpretation $I : (D, \alpha)$

- **Domain** D: set of objects
- **Assignment** α: assigns Σ-symbols to domain elements, functions, predicates
- for each $F \in \mathcal{A}$, $I \models F$

Σ-formula F is T-valid iff for every T-interpretation I, $I \models F$.

F is T-valid

iff

$\neg F$ is T-unsatisfiable

Decision Problem for T

Decide if Σ-formula F is T-valid.

T is set of T-valid Σ-formulae.

T_{A}: First-Order Theory of Arrays

Signature:

$$\Sigma_{\mathsf{A}} : \{a[i],\ a\langle i \triangleleft v\rangle,\ =\}$$

Axioms:

- Equality axioms
- Infinite domain axiom schema: for all $n > 0$

$$\forall i_1, \ldots, i_n.\ \exists j.\ \bigwedge_{k=1}^{n} j \neq i_k$$

- Read-over-write

$$\forall a, i, j, v.\ i = j \ \rightarrow\ a\langle i \triangleleft v\rangle[j] = v$$

$$\forall a, i, j, v.\ i \neq j \ \rightarrow\ a\langle i \triangleleft v\rangle[j] = a[j]$$

$T_{\mathsf{A}}^{\mathbb{Z}}$: First-Order Theory of Integer-Indexed Arrays

Signature:

$$\Sigma_{\mathsf{A}}^{\mathbb{Z}} : \ \Sigma_{\mathsf{A}} \cup \Sigma_{\mathbb{Z}} = \{a[i],\ a\langle i \triangleleft v\rangle,\ =,\ 0,\ 1,\ +,\ \geq\}$$

Axioms:

- Axioms of integer arithmetic
- Equality axioms
- Read-over-write

$$\forall a, i, j, v.\ i = j \ \rightarrow \ a\langle i \triangleleft v\rangle[j] = v$$
$$\forall a, i, j, v.\ i \neq j \ \rightarrow \ a\langle i \triangleleft v\rangle[j] = a[j]$$

Fragment of T

Subset of T given by syntactic restriction.

Example: “quantifier-free” fragment (QFF) of T_{A}

Is

$$a[i] = e_1 \ \wedge \ e_1 \neq e_2 \ \rightarrow \ a\langle i \triangleleft e_2 \rangle[i] \neq a[i]$$

T_{A}-valid?

Alternately, is

$$a[i] = e_1 \ \wedge \ e_1 \neq e_2 \ \wedge \ a\langle i \triangleleft e_2 \rangle[i] = a[i]$$

T_{A}-unsatisfiable?

Nelson-Oppen Combination Method

Given:

- Theories $T_1, \ldots, T_k$ that share only $=$ (*and are stably infinite*)
- Decision procedures $P_1, \ldots, P_k$
- Quantifier-free $(\Sigma_1 \cup \cdots \cup \Sigma_k)$-formula F

Decide if F is $(T_1 \cup \cdots \cup T_k)$-satisfiable **using** $P_1, \ldots, P_k$.

Think about arrays in context of Nelson-Oppen.

History

- 1962: John McCarthy formalizes arrays as first-order theory T_{A}.
- 1969: James King describes and implements DP for QFF of T_{A}.
- 1979: Nelson & Oppen describe combination method for QF theories sharing $=$.
- 1980s: Suzuki, Jefferson; Jaffar; Mateti describe DPs for QFF of theories of arrays with predicates for sorted, partitioned, *etc*.
- 1997: Levitt describes DP for QFF of extensional theory of arrays in thesis.
- 2001: Stump, Barrett, Dill, Levitt describe DP for QFF of extensional theory of arrays.

- 2006: Bradley, Manna, Sipma describe DP for **array property fragment** of T_{A}, $T_{\mathsf{A}}^{\mathbb{Z}}$.
- Other recent references:
  - Sofronie-Stokkermans *et al.*: local theory extensions
  - Ghilardi, Nicolini, Ranise, Zucchelli
  - Iosef, Habermehl, Vojnar: use flat counter automata
  - Fontaine: Combinations with Bernays-Schonfinkel-Ramsey class

Array Property Fragment of T_{A}

Array property:

$$\forall i.\, F[i] \;\rightarrow\; G[a[i]]$$

- F: **index guard**

$$\begin{aligned} \mathsf{iguard} &:= \mathsf{iguard} \wedge \mathsf{iguard} \mid \mathsf{iguard} \vee \mathsf{iguard} \mid \mathsf{atom} \\ \mathsf{atom} &:= \mathsf{var} = \mathsf{var} \mid evar \neq \mathsf{var} \mid \mathsf{var} \neq evar \mid \top \\ \mathsf{var} &:= evar \mid uvar \end{aligned}$$

- G: **value constraint**

  i only appears in $a[i]$ (possibly within nested array properties)

Array property fragment: Boolean combination of array properties and QF formulae.

Array Property Fragment of $T_{\mathsf{A}} \cup T$

Same definition when T is a Nelson-Oppen theory.

Decision Procedure

Given: Array property formula F

1. F_1: push negations to atoms
2. F_2: Eliminate writes

$$\frac{G[a\langle i \triangleleft v\rangle]}{G[a'] \ \wedge \ a'[i] = v \ \wedge \ (\forall j.\, j \neq i \ \rightarrow \ a[j] = a'[j])}$$

3. Construct **index set**

$$\mathcal{I} : \{t \ : \ t \text{ is symbolic index}\} \cup \{\kappa\}$$

4. F_4: κ is unique

$$F_2 \wedge \bigwedge_{t \in \mathcal{I} \setminus \kappa} \kappa \neq t$$

5. F_5: Instantiate quantifiers

$$H[\forall i.\, G[i]] \implies H\left[\bigwedge_{t \in \mathcal{I}} G[t]\right]$$

6. F_5 is QF. Decide satisfiability using Nelson-Oppen DP.

Example: Extensional theory (Stump *et al.*, 2001)

$$a = b\langle i \triangleleft v \rangle \ \wedge \ a[i] \neq v$$

In array property fragment:

$$(\forall j.\, a[j] = b\langle i \triangleleft v \rangle[j]) \ \wedge \ a[i] \neq v$$

Eliminate write:

$$(\forall j.\, a[j] = b'[j]) \ \wedge \ a[i] \neq v$$
$$\wedge \ b'[i] = v \ \wedge \ (\forall j.\, j \neq i \ \rightarrow \ b'[j] = b[j])$$

Index set:

$$\mathcal{I} : \{i, \kappa\}$$

QF formula:

$$a[i] = b'[i] \ \wedge \ a[\kappa] = b'[\kappa]$$
$$\wedge \ a[i] \neq v \ \wedge \ b'[i] = v$$
$$\wedge \ (i \neq i \ \rightarrow \ b'[i] = b[i]) \ \wedge \ (\kappa \neq i \ \rightarrow \ b'[\kappa] = b[\kappa])$$
$$\wedge \ \kappa \neq i$$

Simplified:

$$\boxed{a[i] = b'[i]} \ \wedge \ a[\kappa] = b'[\kappa]$$
$$\wedge \ \boxed{a[i] \neq v} \ \wedge \ \boxed{b'[i] = v}$$
$$\wedge \ b'[\kappa] = b[\kappa]$$
$$\wedge \ \kappa \neq i$$

Why κ?

$$(\forall i.\, a[i] > 0) \;\wedge\; (\forall i.\, a[i] < 0)$$

But requires infinite domain for indices. Recall axiom schema:

For all $n > 0$

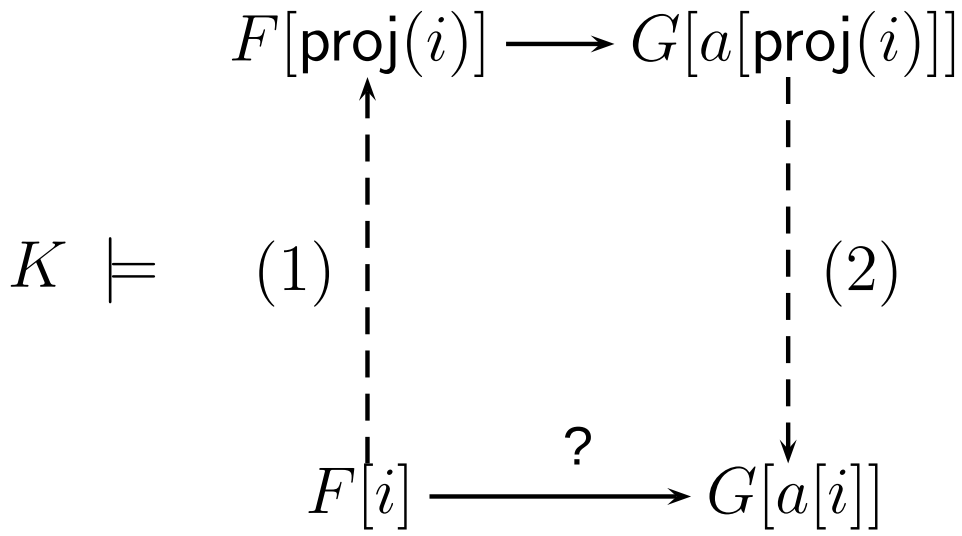
$$\forall i_1, \ldots, i_n.\, \exists j.\, \bigwedge_{k=1}^{n} j \neq i_k$$

Correctness

- Sound? It's just quantifier elimination (except for κ).
- Complete?

  Assume $I \models F_5$. Construct J such that $J \models F$.

$$\text{proj}(t) = \begin{cases} i & \text{if } \alpha_I[t] = \mathsf{v}_i \text{ for some } i \in \mathcal{I} \\ \kappa & \text{otherwise} \end{cases}$$


$F[\mathsf{proj}(i)]$
$G[a[\mathsf{proj}(i)]]$
$K \models$
(1)
(2)
?
$F[i]$
$G[a[i]]$

Array Property Fragment of $T_{\mathsf{A}}^{\mathbb{Z}}$

Array property:

$$\forall i.\, F[i] \;\rightarrow\; G[a[i]]$$

- F: **index guard**

$$\begin{array}{rcl} \mathsf{iguard} & := & \mathsf{iguard} \wedge \mathsf{iguard} \mid \mathsf{iguard} \vee \mathsf{iguard} \mid \mathsf{atom} \\ \mathsf{atom} & := & \mathsf{expr} \leq \mathsf{expr} \mid \mathsf{expr} = \mathsf{expr} \\ \mathsf{expr} & := & uvar \mid \mathsf{pexpr} \\ \mathsf{pexpr} & := & \mathsf{pexpr}' \\ \mathsf{pexpr}' & := & \mathbb{Z} \mid \mathbb{Z} \cdot evar \mid \mathsf{pexpr}' + \mathsf{pexpr}' \end{array}$$

- G: **value constraint**

  i only appears in $a[i]$ (possibly within nested array properties)

Array property fragment: Boolean combination of array properties and QF formulae.

Decision Procedure

Given: Array property formula F

1. F_1: push negations to atoms
2. F_2: Eliminate writes

$$\frac{G[a\langle i \triangleleft v\rangle]}{\begin{array}{l} G[a'] \ \wedge \ a'[i] = v \\ \wedge \ (\forall j.\, j \leq i-1 \ \vee \ i+1 \leq j \ \rightarrow \ a[j] = a'[j]) \end{array}}$$

3. Construct **index set**

$$\mathcal{I} : \{t \ : \ t \text{ is symbolic index}\} \quad (\{0\} \text{ if empty})$$

4. F_4: Instantiate quantifiers

$$H[\forall i.\, G[i]] \quad \Longrightarrow \quad H\left[\bigwedge_{t \in \mathcal{I}} G[t]\right]$$

5. F_4 is QF. Decide satisfiability using Nelson-Oppen DP.

Example

$$\mathsf{sorted}(a, \ell, u) : \forall i, j.\, \ell \leq i \leq j \leq u \rightarrow a[i] \leq a[j]$$

Is

$$\mathsf{sorted}(a\langle 0 \triangleleft 0\rangle\langle 5 \triangleleft 1\rangle, 0, 5) \wedge \mathsf{sorted}(a\langle 0 \triangleleft 10\rangle\langle 5 \triangleleft 11\rangle, 0, 5)$$

$(T^{\mathbb{Z}}_{\mathsf{A}} \cup T_{\mathbb{Z}})$-satisfiable?

0	w	x	y	z	1

10	w	x	y	z	11

Example

$\mathsf{sorted}(a\langle 0 \lhd 0\rangle\langle 5 \lhd 1\rangle, 0, 5) \ \wedge \ \mathsf{sorted}(a\langle 0 \lhd 10\rangle\langle 5 \lhd 11\rangle, 0, 5)$

Index set: $\{-1, 0, 1, 4, 5, 6\}$

- $\{0, 5\}$ from $0 \leq i \leq j \leq 5$
- $\{-1, 1\}$ from $\cdot\langle 0 \lhd \cdot\rangle$
- $\{4, 6\}$ from $\cdot\langle 5 \lhd \cdot\rangle$

Contradiction: $0 \leq a[1] \leq 1 \ \wedge \ 10 \leq a[1] \leq 11$

Need 1 or 4 in index set.

Complexity

Quantifier elimination is in NEXP for Nelson-Oppen theories:

1. $|\mathcal{I}|$ is linear in size of F, so linear-time quantifier instantiation.
2. NP DPs applied to QF formula at most exponentially larger than F.
3. Exponential in largest stack of universal quantifiers.

   Fixing stack height ("extensional", "sorting" fragment) gives NP procedure.

Complexity

NEXP-hard even for uninterpreted domain and range.

- Bernays-Schonfinkel-Ramsey (BSR) class: $\exists^*\forall^*$, only predicates
- Deciding satisfiability is NEXP-complete
- Reduction:

$$\exists x.\, F[x] \implies \exists x.\, d(x) \wedge F[x]$$
$$\forall x.\, F[x] \implies \forall x.\, d(x) \rightarrow F[x]$$

  Why d? Only infinite T_{A}-interpretations, but possible finite BSR-interpretations.

Thanks to De Moura, Bjorner, Kuncak for mentioning BSR.

Undecidable Extensions

- Extra quantifier alternation
- Nested reads under $\forall i$: $a[a[i]]$
- No separation: $\forall i.\ F[a[i], i]$
- Arithmetic: $a[i+1]$ when i is universal
- Strict comparison: $i < j$ when i, j are universal
- Permutation predicate

Reduce from undecidability of Diophantine equations:

$$p(x_1, \ldots, x_n) = 0$$

(over nonnegative $\overline{x}$)

"Walk":

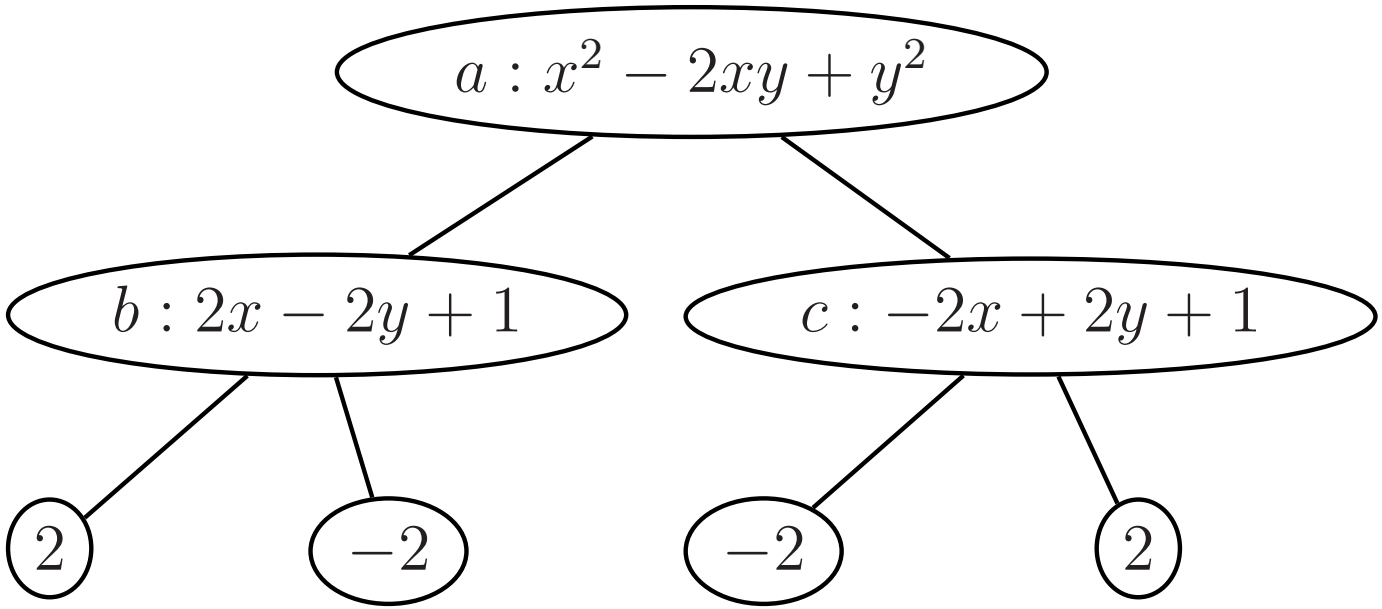
- Begin at origin.
- At each step, increment one x_i.
- End at solution.

A walk exists iff $p(\overline{x}) = 0$ has a solution.

1. $q(\overline{x}) = p(\overline{x})^2$

   $q(\overline{x}) > 0$ on walk, except at solution

2. Difference tree:



3. Walk arithmetic is **linear arithmetic**:

Initially	$(a, b, c) = (0, 1, 1)$
$x := x + 1$	$(a, b, c) := (a + b, b + 2, c - 2)$
$y := y + 1$	$(a, b, c) := (a + c, b - 2, c + 2)$

Initial condition:

$$\theta(i) : \bigwedge_p a_p[i] = p(\overline{0})$$

Transition relation for variable x:

$$\rho_x(i,j) : \bigwedge_p a_p[j] = a_p[i] + a_{\Delta_x p}[i]$$

Idle transition:

$$\rho_0(i,j) : \overline{a}[i] = \overline{a}[j]$$

Easy: extra quantifier alternation

$$\exists \overline{a}, s.\ \forall i.\ \exists j.\ \left[\begin{array}{l} \theta(0) \\ \quad \wedge\ \left(\rho_0(i,j)\ \vee\ \bigvee_x \rho_x(i,j)\right) \\ \quad \wedge\ s[j] = s[i] - a_q[i] \\ \quad \wedge\ s[i] > 0 \end{array}\right]$$

Tricky: permutation predicate $\mathsf{perm}(a, b)$

Define identifiers:

$$\mathsf{perm}(a, b) \ \wedge \ \forall i.\, b[i] = a[i] + 1$$

Bounded case for integer-indexed arrays: n identifiers

$$a[0] = n \ \wedge \ b[0] = 0 \ \wedge \ \mathsf{perm}(a, b, 0, n)$$
$$\wedge \ \forall i.\, 0 < i \leq n \ \rightarrow \ 0 \leq a[i], b[i] \leq n \ \wedge \ b[i] = a[i] + 1$$

Unbounded case:

$$\exists \overline{a}, d, e, z, n.\ \forall i, j.$$

$$\left[\begin{array}{l} \theta(z) \\ \quad \wedge \left[\begin{array}{l} \left[\begin{array}{l} (d[j] > 0 \ \wedge \ d[j] = d[i] + 1) \\ \quad \vee \ (d[j] < 0 \ \wedge \ d[j] = d[i] - 1) \end{array}\right] \\ \qquad\qquad \rightarrow \bigvee_x \rho_x(i, j) \end{array}\right] \\ \quad \wedge \ a_q[n] = 0 \\ \quad \wedge \ d[z] = 0 \ \wedge \ \mathsf{perm}(d, e) \ \wedge \ \forall i.\ e[i] = d[i] + 1 \end{array}\right]$$

Open: $\forall i, j.\ i \neq j \ \rightarrow \ a[i] \neq a[j]$

- For integer-indexed arrays: undecidable
- Otherwise: ?

Incremental Instantiation

1. Instantiate F to F'
2. Find $I \models F'$ (otherwise, F is unsatisfiable)
3. Construct $G : \neg F[I]$
4. Find $J \models G$ (otherwise, F is satisfiable)
5. Enlarge instantiation set (according to I and J) and repeat

May avoid full instantiation (whether satisfiable or unsatisfiable)

Summary

- Array property fragments allow encoding
  - properties of arrays and array segments
  - operations on sets, multisets, and hashtables
- Simple decision procedure: quantifier instantiation
- Larger natural fragments are undecidable