CS156: The Calculus of Computation

Zohar Manna

Winter 2008

Chapter 11: Arrays

Arrays I: Quantifier-free Fragment of T_A

Signature:

$$\Sigma_A : \{\cdot[\cdot], \ \cdot\langle \cdot \triangleleft \cdot \rangle, \ =\}$$

where

- $a[i]$ binary function –
  read array a at index i ("read(a,i)")
- $a\langle i \triangleleft v \rangle$ ternary function –
  write value v to index i of array a ("write(a,i,v)")

Axioms

1. the axioms of (reflexivity), (symmetry), and (transitivity) of T_E
2. $\forall a, i, j.\ i = j \ \rightarrow \ a[i] = a[j]$ (array congruence)
3. $\forall a, v, i, j.\ i = j \ \rightarrow \ a\langle i \triangleleft v \rangle[j] = v$ (read-over-write 1)
4. $\forall a, v, i, j.\ i \neq j \ \rightarrow \ a\langle i \triangleleft v \rangle[j] = a[j]$ (read-over-write 2)

Infinite Domain

We add an axiom schema to T_{A} that forbids interpretations with finite arrays.

For each positive natural number n, the following is an axiom:

$$\forall x_1, \ldots, x_n.\ \exists y.\ \bigwedge_{i=1}^{n} y \neq x_i$$

Equality in T_A

Note: $=$ is only defined for array elements:

$$a[i] = e \;\rightarrow\; a\langle i \triangleleft e\rangle = a$$

not T_A-valid, but

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

is T_A-valid.

Also

$$a = b \;\rightarrow\; a[i] = b[i]$$

is not T_A-valid: We only axiomatized a restricted congruence.

T_A is undecidable
Quantifier-free fragment of T_A is decidable

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?

Alternatively, 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?

Decision Procedure for T_A

Given quantifier-free conjunctive Σ_A-formula F.
To decide the T_A-satisfiability of F:

Step 1

If F does not contain any write terms $a\langle i \triangleleft v\rangle$, then

1. associate array variables a with fresh function symbol f_a, and replace read terms $a[i]$ with $f_a(i)$;
2. decide the T_E-satisfiability of the resulting formula.

Decision Procedure for T_A

Step 2

Select some read-over-write term $a\langle i \triangleleft v\rangle[j]$ (note that a may itself be a write term) and split on two cases:

1. According to (read-over-write 1), replace

$$F[a\langle i \triangleleft v\rangle[j]] \quad \text{with} \quad F_1 : \; F[v] \;\wedge\; i = j \;,$$

   and recurse on F_1. If F_1 is found to be T_A-satisfiable, return satisfiable.

2. According to (read-over-write 2), replace

$$F[a\langle i \triangleleft v\rangle[j]] \quad \text{with} \quad F_2 : \; F[a[j]] \;\wedge\; i \neq j \;,$$

   and recurse on F_2. If F_2 is found to be T_A-satisfiable, return satisfiable.

If both F_1 and F_2 are found to be T_A-unsatisfiable, return unsatisfiable.

Example

Consider Σ_A-formula

$$F : \; i_1 = j \;\wedge\; i_1 \neq i_2 \;\wedge\; a[j] = v_1 \;\wedge\; a\langle i_1 \triangleleft v_1\rangle\langle i_2 \triangleleft v_2\rangle[j] \neq a[j] \; .$$

F contains a write term,

$$a\langle i_1 \triangleleft v_1\rangle\langle i_2 \triangleleft v_2\rangle[j] \neq a[j] \; .$$

According to (read-over-write 1), assume $i_2 = j$ and recurse on

$$F_1 : \; i_2 = j \;\wedge\; i_1 = j \;\wedge\; i_1 \neq i_2 \;\wedge\; a[j] = v_1 \;\wedge\; v_2 \neq a[j] \; .$$

F_1 does not contain any write terms, so rewrite it to

$$F_1' : \; \underbrace{i_2 = j \;\wedge\; i_1 = j \;\wedge\; i_1 \neq i_2}_{} \wedge f_a(j) = v_1 \;\wedge\; v_2 \neq f_a(j) \; .$$

Contradiction — F_1' is T_E-unsatisfiable.

Returning, we try the second case:
according to (read-over-write 2), assume $i_2 \neq j$ and recurse on

$$F_2 : \; i_2 \neq j \;\wedge\; i_1 = j \;\wedge\; i_1 \neq i_2 \;\wedge\; a[j] = v_1 \;\wedge\; a\langle i_1 \triangleleft v_1\rangle[j] \neq a[j] \; .$$

F_2 contains a write term. According to (read-over-write 1), assume $i_1 = j$ and recurse on

$$F_3 : \; i_1 = j \;\wedge\; i_2 \neq j \;\wedge\; i_1 = j \;\wedge\; i_1 \neq i_2 \;\wedge\; \underbrace{a[j] = v_1 \;\wedge\; v_1 \neq a[j]} \; .$$

Contradiction. Thus, according to (read-over-write 2), assume $i_1 \neq j$ and recurse on

$$F_4 : \; i_1 \neq j \;\wedge\; i_2 \neq j \;\wedge\; i_1 = j \;\wedge\; i_1 \neq i_2 \;\wedge\; a[j] = v_1 \;\wedge\; \underbrace{a[j] \neq a[j]} \; .$$

Contradiction: all branches have been tried, and thus F is T_A-unsatisfiable.

Question: Suppose instead that F does not contain the literal $i_1 \neq i_2$. Is this new formula T_A-satisfiable?

Decision Procedure for Arrays

The quantifier free fragment of T_{A} is *decidable*.
However *too weak* to express important properties:

- Containment: $\forall i.\ \ell \leq i \leq u \implies a[i] \neq e$
- Sortedness: $\forall i, j.\ \ell \leq i \leq j \leq u \implies a[i] \leq a[j]$
- Partitioning: $\forall i, j.\ \ell_1 \leq i \leq u_1 \ \wedge\ \ell_2 \leq j \leq u_2 \implies a[i] \leq a[j]$

The general theory of arrays T_{A} with quantifier is *not decidable*.

Is there a decidable fragment of T_{A} that contains the above formulae?

Example

We want to prove validity for a formula, such as:

$$(\forall i. a[i] \neq e) \ \wedge \ e \neq f \ \rightarrow \ (\forall i. a\langle j \triangleleft f \rangle[i] \neq e) \ .$$

Equivalently show unsatisfiability of

$$(\forall i. a[i] \neq e) \ \wedge \ e \neq f \ \wedge \ (\exists i. a\langle j \triangleleft f \rangle[i] = e) \ .$$

or the equisatisfiable formula

$$(\forall i. a[i] \neq e) \ \wedge \ e \neq f \ \wedge \ a\langle j \triangleleft f \rangle[i] = e \ .$$

We need to handle a universal quantifier.

Arrays II: Array Property Fragment of T_A

Decidable fragment of T_A that includes $\forall$ quantifiers

Array property

Σ_A-formula of form

$$\forall \bar{i}.\ \alpha[\bar{i}] \ \rightarrow\ \beta[\bar{i}]\ ,$$

where $\bar{i}$ is a list of variables.

- *index guard* $\alpha[\bar{i}]$:

$$\begin{array}{rcl} \text{iguard} & \rightarrow & \text{iguard} \wedge \text{iguard} \mid \text{iguard} \vee \text{iguard} \mid \text{atom} \\ \text{atom} & \rightarrow & \text{var} = \text{var} \mid \textit{evar} \neq \text{var} \mid \text{var} \neq \textit{evar} \mid \top \\ \text{var} & \rightarrow & \textit{evar} \mid \textit{uvar} \end{array}$$

  where *uvar* is any universally quantified index variable, and *evar* is any unquantified free variable.

Arrays II: Array Property Fragment of T_A (cont)

- *value constraint* $\beta[\bar{i}]$:
  Any qff, but a universally quantified index can occur only in a read $a[i]$, where a is an array term.

Array property Fragment:

Boolean combinations of quantifier-free Σ_A-formulae and array properties

Note: $a[b[k]]$ for unquantified variable k is okay, but $a[b[i]]$ for universally quantified variable i is forbidden. Cannot replace it by

$$\forall i, j. \ \ldots b[i] = j \ \wedge \ a[j] \ldots$$

In β, the universally quantified variable j may occur in $a[j]$ but not in $b[i] = j$.

Example: Array Property Fragment

Is this formula in the array property fragment?

$$F : \; \forall i.\; i \neq a[k] \;\rightarrow\; a[i] = a[k]$$

The antecedent is not a legal index guard since $a[k]$ is not a variable (neither a *uvar* nor an *evar*); however, by simple manipulation

$$F' : \; v = a[k] \;\wedge\; \forall i.\; i \neq v \;\rightarrow\; a[i] = a[k]$$

Here, $i \neq v$ is a legal index guard, and $a[i] = a[k]$ is a legal value constraint. F and F' are equisatisfiable.
However, no manipulation works for:

$$G : \; \forall i.\; i \neq a[i] \;\rightarrow\; a[i] = a[k] \; .$$

Thus, G is not in the array property fragment.

Array property fragment and extensionality

Array property fragment allows expressing equality between arrays (*extensionality*): two arrays are equal precisely when their corresponding elements are equal.

For given formula

$$F : \; \cdots \; \wedge \; a = b \; \wedge \; \cdots$$

with array terms a and b, rewrite F as

$$F' : \; \cdots \; \wedge \; (\forall i. \; \top \; \rightarrow \; a[i] = b[i]) \; \wedge \; \cdots \; .$$

F and F' are equisatisfiable.

Decision Procedure for Array Property Fragment

Basic Idea: Replace universal quantification $\forall i.F[i]$
by finite conjunction $F[t_1] \land \ldots \land F[t_n]$.

We call $t_1, \ldots, t_n$ the index terms and they depend on the formula.

Example

Consider

$$F : \; a\langle i \triangleleft v \rangle = a \;\wedge\; a[i] \neq v \;,$$

which expands to

$$F' : \; \forall j.\; a\langle i \triangleleft v \rangle[j] = a[j] \;\wedge\; a[i] \neq v \;.$$

Intuitively, to determine that F' is T_{A}-unsatisfiable requires merely examining index i:

$$F'' : \; \left(\bigwedge_{j \in \{i\}} a\langle i \triangleleft v \rangle[j] = a[j] \right) \;\wedge\; a[i] \neq v \;,$$

or simply

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

Simplifying,

$$v = a[i] \;\wedge\; a[i] \neq v \;,$$

it is clear that this formula, and thus F, is T_{A}-unsatisfiable.

The Algorithm

Given array property formula F, decide its T_A-satisfiability by the following steps:

Step 1
Put F in NNF.

Step 2
Apply the following rule exhaustively to remove writes:

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

After an application of the rule, the resulting formula contains at least one fewer write terms than the given formula.

Step 3
Apply the following rule exhaustively to remove existential quantification:

$$\frac{F[\exists \bar{i}.\ G[\bar{i}]]}{F[G[\bar{j}]]} \text{ for fresh } \bar{j} \quad \text{(exists)}$$

Existential quantification can arise during Step 1 if the given formula has a negated array property.

Steps 4-6 accomplish the reduction of universal quantification to finite conjunction.
Main idea: select a set of symbolic index terms on which to instantiate all universal quantifiers. The set is sufficient for correctness.

Step 4

From the output F_3 of Step 3, construct the **index set** $\mathcal{I}$:

$$\begin{array}{rcll} \mathcal{I} & = & \cup & \{t \;:\; \cdot[t] \in F_3 \text{ such that } t \text{ is not a universally quantified variable}\} \\ & & \cup & \{t \;:\; t \text{ occurs as an } \textit{evar} \text{ in the parsing of index guards}\} \\ & & \cup & \{\lambda\} \end{array}$$

This index set is the finite set of "symbolic indices" that need to be examined. It includes

- all terms t that occur in some read $a[t]$ anywhere in F_3 (unless it is a universally quantified variable); e.g., k in $a[k]$.
- all terms t (unquantified variable) that are compared to a universally quantified variable in some index guard $F[i]$; e.g., k in $i = k$.
- λ is a fresh constant that represents all other index positions that are not explicitly in $\mathcal{I}$.

Step 5 (Key step)
Apply the following rule exhaustively to remove universal quantification:

$$\frac{H[\forall \bar{i}.\ \alpha[\bar{i}]\ \rightarrow\ \beta[\bar{i}]]}{H\left[\bigwedge_{\bar{i}\in\mathcal{I}^n} (\alpha[\bar{i}]\ \rightarrow\ \beta[\bar{i}])\right]} \quad \text{(forall)}$$

where n is the size of the list of quantified variables $\bar{i}$.

Step 6
From the output F_5 of Step 5, construct

$$F_6 :\ F_5\ \wedge \bigwedge_{t\ \in\ \mathcal{I}\setminus\{\lambda\}} \lambda \neq t\ .$$

The new conjuncts assert that the variable λ introduced in Step 4 is indeed unique.

Step 7
Decide the T_A-satisfiability of F_6 using the decision procedure for the quantifier-free fragment.

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

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

In array property fragment:

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

Eliminate write:

$$\begin{array}{ll} & (\forall j.\ a[j] = b'[j]) \\ \wedge & a[i] \neq v \\ \wedge & b'[i] = v \\ \wedge & (\forall j.\ j \neq i \rightarrow b'[j] = b[j]) \end{array}$$

Index set:

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

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

QF formula:

$$
\begin{array}{ll}
 & a[i] = b'[i] \;\wedge\; a[\lambda] = b'[\lambda] \\
\wedge & a[i] \neq v \;\wedge\; b'[i] = v \\
\wedge & (i \neq i \rightarrow b'[i] = b[i]) \;\wedge (\lambda \neq i \rightarrow b'[\lambda] = b[\lambda]) \\
\wedge & \lambda \neq i
\end{array}
$$

Simplified:

$$
\begin{array}{ll}
 & \boxed{a[i] = b'[i]} \;\wedge\; a[\lambda] = b'[\lambda] \\
\wedge & \boxed{a[i] \neq v} \;\wedge\; \boxed{b'[i] = v} \\
\wedge & b'[\lambda] = b[\lambda] \\
\wedge & \lambda \neq i
\end{array}
$$

Contradiction. So F is unsatisfiable.

Example

Is this $T_A^=$-formula (arrays with extensionality) valid?

$$F : (\forall i.\ i \neq k \ \rightarrow \ a[i] = b[i]) \ \wedge \ b[k] = v \ \rightarrow \ a\langle k \triangleleft v \rangle = b$$

Check unsatisfiability of T_A-formula:

$$\neg((\forall i.\ i \neq k \ \rightarrow \ a[i] = b[i]) \wedge b[k] = v \ \rightarrow \ (\forall i.\ a\langle k \triangleleft v \rangle[i] = b[i]))$$

Step 1: NNF

$$F_1 : (\forall i.\ i \neq k \ \rightarrow \ a[i] = b[i]) \wedge b[k] = v \wedge (\exists i.\ a\langle k \triangleleft v \rangle[i] \neq b[i])$$

Step 2: Remove array writes

$$F_2 : (\forall i.\ i \neq k \ \rightarrow \ a[i] = b[i]) \ \wedge \ b[k] = v \ \wedge \ (\exists i.\ a'[i] \neq b[i]) \\ \wedge \ a'[k] = v \ \wedge \ (\forall i.\ i \neq k \ \rightarrow \ a'[i] = a[i])$$

Example (cont)

Step 3: Remove existential quantifier

$$F_3 : (\forall i.\ i \neq k \ \rightarrow\ a[i] = b[i]) \ \wedge\ b[k] = v \ \wedge\ a'[j] \neq b[j] \ \wedge\ a'[k] = v \ \wedge\ (\forall i.\ i \neq k \ \rightarrow\ a'[i] = a[i])$$

Example (cont)

Step 4: Compute index set $\mathcal{I} = \{\lambda, k, j\}$
Step 5+6: Replace universal quantifier:

$$
\begin{aligned}
F_6 : (\lambda \neq k \;\rightarrow\; & a[\lambda] = b[\lambda]) \\
\wedge\; & (k \neq k \;\rightarrow\; a[k] = b[k]) \\
\wedge\; & (j \neq k \;\rightarrow\; a[j] = b[j]) \\
\wedge\; & b[k] = v \;\wedge\; a'[j] \neq b[j] \;\wedge\; a'[k] = v \\
\wedge\; & (\lambda \neq k \;\rightarrow\; a'[\lambda] = a[\lambda]) \\
\wedge\; & (k \neq k \;\rightarrow\; a'[k] = a[k]) \\
\wedge\; & (j \neq k \;\rightarrow\; a'[j] = a[j]) \\
\wedge\; & \lambda \neq k \;\wedge\; \lambda \neq j
\end{aligned}
$$

Case distinction on $j = k$ (4th line) and $j \neq k$ (3rd line, 4th line, and 7th line) proves unsatisfiability of F_6.
Therefore F is valid.

The importance of λ

Is this formula satisfiable?

$$F : (\forall i. i \neq j \;\rightarrow\; a[i] = b[i]) \;\wedge\; (\forall i. i \neq k \;\rightarrow\; a[i] \neq b[i])$$

The algorithm produces (for $\{\lambda, j, k\}$):

$$\begin{aligned} F_6 : \lambda \neq j \;&\rightarrow\; a[\lambda] = b[\lambda] \\ \wedge\; j \neq j \;&\rightarrow\; a[j] = b[j] \\ \wedge\; k \neq j \;&\rightarrow\; a[k] = b[k] \\ \wedge\; \lambda \neq k \;&\rightarrow\; a[\lambda] \neq b[\lambda] \\ \wedge\; j \neq k \;&\rightarrow\; a[j] \neq b[j] \\ \wedge\; k \neq k \;&\rightarrow\; a[k] \neq b[k] \\ \wedge\; \lambda \neq j \;&\wedge\; \lambda \neq k \end{aligned}$$

The 1st, 4th and last lines give a contradiction! F is unsatisfiable.

The importance of λ (cont)

Without λ we had the formula:

$$\begin{aligned} F'_6 : j \neq j \ &\rightarrow \ a[j] = b[j] \\ \wedge \ k \neq j \ &\rightarrow \ a[k] = b[k] \\ \wedge \ j \neq k \ &\rightarrow \ a[j] \neq b[j] \\ \wedge \ k \neq k \ &\rightarrow \ a[k] \neq b[k] \end{aligned}$$

which simplifies to:

$$j \neq k \ \rightarrow \ a[k] = b[k] \ \wedge \ a[j] \neq b[j] \,.$$

This formula F is satisfiable!

Example

Consider array property formula

$$F \quad : \quad a\langle \ell \triangleleft v \rangle[k] = b[k] \ \wedge \ b[k] \neq v \ \wedge \ a[k] = v \\ \wedge \ \underbrace{(\forall i.\ i \neq \ell \ \rightarrow \ a[i] = b[i])}_{\text{array property}}$$

By Step 2, rewrite F as

$$F_2 : \begin{array}{l} a'[k] = b[k] \ \wedge \ b[k] \neq v \ \wedge \ a[k] = v \ \wedge \ (\forall i.\ i \neq \ell \ \rightarrow \ a[i] = b[i]) \\ \quad \wedge \ a'[\ell] = v \ \wedge \ (\forall j.\ j \neq \ell \ \rightarrow \ a[j] = a'[j]) \end{array}$$

F_2 does not contain any existential quantifiers. Its index set is

$$\mathcal{I} \ = \ \{\lambda, k, \ell\} \ .$$

Example (cont)

Thus, by Step 5, replace universal quantification (and step 6):

$$F_6 : \quad \begin{array}{l} a'[k] = b[k] \ \wedge \ b[k] \neq v \ \wedge \ a[k] = v \ \wedge \ \bigwedge_{i \, \in \, \mathcal{I}} (i \neq \ell \rightarrow a[i] = b[i]) \\ \wedge \ a'[\ell] = v \ \wedge \ \bigwedge_{j \, \in \, \mathcal{I}} \left(j \neq \ell \ \rightarrow \ a[j] = a'[j]\right) \\ \wedge \ \lambda \neq k \ \wedge \ \lambda \neq \ell \end{array}$$

Expanding produces

$$F_6' : \quad \begin{array}{l} a'[k] = b[k] \ \wedge \ b[k] \neq v \ \wedge \ a[k] = v \\ \wedge \ (\lambda \neq \ell \ \rightarrow \ a[\lambda] = b[\lambda]) \\ \wedge \ (k \neq \ell \ \rightarrow \ a[k] = b[k]) \\ \wedge \ (\ell \neq \ell \ \rightarrow \ a[\ell] = b[\ell]) \\ \wedge \ a'[\ell] = v \\ \wedge \ (\lambda \neq \ell \ \rightarrow \ a[\lambda] = a'[\lambda]) \\ \wedge \ (k \neq \ell \ \rightarrow \ a[k] = a'[k]) \ \wedge \ (\ell \neq \ell \ \rightarrow \ a[\ell] = a'[\ell]) \\ \wedge \ \lambda \neq k \ \wedge \ \lambda \neq \ell \end{array}$$

Example (cont)

Simplifying,

$$
F_6'': \quad \begin{array}{l}
a'[k] = b[k] \ \wedge \ b[k] \neq v \ \wedge \ a[k] = v \\
\wedge \ a[\lambda] = b[\lambda] \ \wedge \ (k \neq \ell \ \rightarrow \ a[k] = b[k]) \\
\wedge \ a'[\ell] = v \\
\wedge \ a[\lambda] = a'[\lambda] \ \wedge \ (k \neq \ell \ \rightarrow \ a[k] = a'[k]) \\
\wedge \ \lambda \neq k \ \wedge \ \lambda \neq \ell
\end{array}
$$

There are two cases to consider.

- If $k = \ell$, then $a'[\ell] = v$ (3rd line) and $a'[k] = b[k]$ (1st line) imply $b[k] = v$,
  yet $b[k] \neq v$.
- If $k \neq \ell$, then $a[k] = v$ (1st line) and $a[k] = b[k]$ (2nd line) imply $b[k] = v$,
  but again $b[k] \neq v$.

Hence, F_6'' is T_A-unsatisfiable, indicating that F is T_A-unsatisfiable.

Correctness of Decision Procedure

Theorem

Consider a Σ_A-formula F from the array property fragment of T_A. The output F_6 of Step 6 of the algorithm is T_A-equisatisfiable to F.

This also works when extending the Logic with an arbitrary theory T with signature Σ for the elements:

Theorem

Consider a $\Sigma_A \cup \Sigma$-formula F from the array property fragment of $T_A \cup T$. The output F_6 of Step 6 of the algorithm is $T_A \cup T$-equisatisfiable to F.

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}, $T_{\mathsf{A}}^{\mathbb{Z}}$.

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

Signature:

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

$\leq$ enables reasoning about subarrays and properties such as whether the subarray is sorted or partitioned.

Axioms of $T_{\mathsf{A}}^{\mathbb{Z}}$: both axioms of T_{A} and $T_{\mathbb{Z}}$

Array Property Fragment of $T_A^{\mathbb{Z}}$

Array property: $\Sigma_A^{\mathbb{Z}}$-formula of the form

$$\forall \bar{i}.\ \alpha[\bar{i}] \ \rightarrow \ \beta[\bar{i}]\ ,$$

where $\bar{i}$ is a list of integer variables.

- $\alpha[\bar{i}]$ *index guard*:

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

  where *uvar* is any universally quantified integer variable, and *evar* is any unquantified free integer variable.

  Note: Why both pexpr and pexpr$'$? E.g., in $i \leq 3k + j$, the expression $3k + j$ is pexpr, but not k or j.

Array Property Fragment of $T_A^{\mathbb{Z}}$ (cont)

- *value constraint* $\beta[\bar{i}]$:
  Any qff, but a universally quantified index can occur only in a read $a[i]$, where a is an array term.

Array property Fragment (APF):

Boolean combinations of quantifier-free $\Sigma_A^{\mathbb{Z}}$-formulae and array properties

Note: $a[b[k]]$ for unquantified variable k is okay, but $a[b[i]]$ for universally quantified variable i is forbidden.

Application: array property fragments

- Array equality $a = b$ in T_A:

$$\forall i.\ a[i] = b[i]$$

- Bounded array equality $\text{beq}(a, b, \ell, u)$ in $T_A^{\mathbb{Z}}$:

$$\forall i.\ \ell \leq i \leq u \ \rightarrow\ a[i] = b[i]$$

- Universal properties $F[x]$ in T_A:

$$\forall i.\ F[a[i]]$$

- Bounded universal properties $F[x]$ in $T_A^{\mathbb{Z}}$:

$$\forall i.\ \ell \leq i \leq u \ \rightarrow\ F[a[i]]$$

- Bounded sorted arrays $\text{sorted}(a, \ell, u)$ in $T_A^{\mathbb{Z}}$ or $T_A^{\mathbb{Z}} \cup T_{\mathbb{Q}}$:

$$\forall i, j.\ \ell \leq i \leq j \leq u \ \rightarrow\ a[i] \leq a[j]$$

- Partitioned arrays $\text{partitioned}(a, \ell_1, u_1, \ell_2, u_2)$ in $T_A^{\mathbb{Z}}$ or $T_A^{\mathbb{Z}} \cup T_{\mathbb{Q}}$:

$$\forall i, j.\ \ell_1 \leq i \leq u_1 < \ell_2 \leq j \leq u_2 \ \rightarrow\ a[i] \leq a[j]$$

The Decision Procedure (Step 1–2)

The idea again is to reduce universal quantification to finite conjunction.

Given F from the array property fragment of $T_{\mathsf{A}}^{\mathbb{Z}}$, decide its $T_{\mathsf{A}}^{\mathbb{Z}}$-satisfiability as follows:

Step 1

Put F in NNF.

Step 2

Apply the following rule exhaustively to remove writes:

$$\frac{G[a\langle i \triangleleft e\rangle]}{G[a'] \ \wedge \ a'[i] = e \ \wedge \ (\forall j.\ j \neq i \ \rightarrow \ a[j] = a'[j])} \text{ for fresh } a' \quad \text{(write)}$$

To meet the syntactic requirements on an index guard, rewrite the third conjunct as

$$\forall j.\ j \leq i - 1 \ \vee \ i + 1 \leq j \ \rightarrow \ a[j] = a'[j]\ .$$

The Decision Procedure (Step 3–4)

Step 3
Apply the following rule exhaustively to remove existential quantification:

$$\frac{F[\exists \bar{i}.\ G[\bar{i}]]}{F[G[\bar{j}]]} \text{ for fresh } \bar{j} \qquad \text{(exists)}$$

Existential quantification can arise during Step 1 if the given formula has a negated array property.

Step 4
From the output of Step 3, F_3, construct the index set $\mathcal{I}$:

$$\mathcal{I} = \begin{array}{l} \{t \ : \ \cdot[t] \in F_3 \text{ such that } t \text{ is not a universally quantified variable}\} \\ \cup \ \{t \ : \ t \text{ occurs as a } \underline{\text{pexpr}} \text{ in the parsing of index guards}\} \end{array}$$

If $\mathcal{I} = \emptyset$, then let $\mathcal{I} = \{0\}$. The index set contains all relevant symbolic indices that occur in F_3. Note: no λ!

The Decision Procedure (Step 5–6)

Step 5

Apply the following rule exhaustively to remove universal quantification:

$$\frac{H[\forall \bar{i}.\ F[\bar{i}] \ \rightarrow \ G[\bar{i}]]}{H\left[\bigwedge_{\bar{i}\in\mathcal{I}^n} \left(F[\bar{i}] \ \rightarrow \ G[\bar{i}]\right)\right]} \quad \text{(forall)}$$

n is the size of the block of universal quantifiers over $\bar{i}$.

Step 6

F_5 is quantifier-free in the combination theory $T_{\mathsf{A}} \cup T_{\mathbb{Z}}$. Decide the $(T_{\mathsf{A}} \cup T_{\mathbb{Z}})$-satisfiability of the resulting formula.

Example

$\Sigma_A^{\mathbb{Z}}$-formula:

$$F: \begin{array}{l} (\forall i.\ \ell \le i \le u \ \rightarrow \ a[i] = b[i]) \\ \ \wedge\ \neg(\forall i.\ \ell \le i \le u+1 \ \rightarrow \ a\langle u+1 \triangleleft b[u+1]\rangle[i] = b[i]) \end{array}$$

In NNF, we have

$$F_1: \begin{array}{l} (\forall i.\ \ell \le i \le u \ \rightarrow \ a[i] = b[i]) \\ \ \wedge\ (\exists i.\ \ell \le i \le u+1 \ \wedge \ a\langle u+1 \triangleleft b[u+1]\rangle[i] \ne b[i]) \end{array}$$

Step 2 produces

$$F_2: \begin{array}{l} (\forall i.\ \ell \le i \le u \ \rightarrow \ a[i] = b[i]) \\ \ \wedge\ (\exists i.\ \ell \le i \le u+1 \ \wedge \ a'[i] \ne b[i]) \\ \ \wedge\ a'[u+1] = b[u+1] \\ \ \wedge\ (\forall j.\ j \le u \ \vee \ u+2 \le j \ \rightarrow \ a[j] = a'[j]) \end{array}$$

Step 3 removes the existential quantifier by introducing a fresh constant k:

$$F_3 : \quad \begin{array}{l} (\forall i.\ \ell \le i \le u \ \rightarrow \ a[i] = b[i]) \\ \ \wedge \ \ell \le k \le u + 1 \ \wedge \ a'[k] \ne b[k] \\ \ \wedge \ a'[u+1] = b[u+1] \\ \ \wedge \ (\forall j.\ j \le u \ \vee \ u + 2 \le j \ \rightarrow \ a[j] = a'[j]) \end{array}$$

The index set is

$$\mathcal{I} = \{k, u+1\} \ \cup \ \{\ell, u, u+2\} \ ,$$

which includes the read indices k and $u + 1$ and the terms ℓ, u, and $u + 2$ that occur as pexprs in the index guards.

Step 5 rewrites universal quantification to finite conjunction over this set:

$$F_5 : \quad \begin{array}{l} \displaystyle\bigwedge_{i \,\in\, \mathcal{I}} (\ell \leq i \leq u \;\rightarrow\; a[i] = b[i]) \\ \wedge\; \ell \leq k \leq u + 1 \;\wedge\; a'[k] \neq b[k] \\ \wedge\; a'[u+1] = b[u+1] \\ \wedge\; \displaystyle\bigwedge_{j \,\in\, \mathcal{I}} (j \leq u \;\vee\; u + 2 \leq j \;\rightarrow\; a[j] = a'[j]) \end{array}$$

Expanding the conjunctions according to the index set $\mathcal{I}$ and simplifying according to trivially true or false antecedents (*e.g.*, $\ell \leq u + 1 \leq u$ simplifies to $\perp$, while $u \leq u \;\vee\; u + 2 \leq u$ simplifies to $\top$) produces:

$$
F'_5 : \quad
\begin{array}{ll}
(\ell \le k \le u \ \rightarrow \ a[k] = b[k]) & (1) \\
\ \wedge \ (\ell \le u \ \rightarrow \ a[\ell] = b[\ell] \ \wedge \ a[u] = b[u]) & (2) \\
\ \wedge \ \ell \le k \le u + 1 & (3) \\
\ \wedge \ a'[k] \ne b[k] & (4) \\
\ \wedge \ a'[u+1] = b[u+1] & (5) \\
\ \wedge \ (k \le u \ \vee \ u + 2 \le k \ \rightarrow \ a[k] = a'[k]) & (6) \\
\ \wedge \ (\ell \le u \ \vee \ u + 2 \le \ell \ \rightarrow \ a[\ell] = a'[\ell]) & (7) \\
\ \wedge \ a[u] = a'[u] \ \wedge \ a[u+2] = a'[u+2] & (8)
\end{array}
$$

$(T_{\mathsf{A}} \cup T_{\mathbb{Z}})$-unsatisfiability of this quantifier-free $(\Sigma_{\mathsf{A}} \cup \Sigma_{\mathbb{Z}})$-formula can be decided using the techniques of Combination of Theories.

Informally, $\ell \le k \le u + 1$ (3)

- If $k \in [\ell, u]$ then $a[k] = b[k]$ (1). Since $k \le u$ then $a[k] = a'[k]$ (6), contradicting $a'[k] \ne b[k]$ (4).
- if $k = u + 1$, $a'[k] \ne b[k] = b[u+1] = a'[u+1] = a'[k]$ by (4) and (5), a contradiction.

Hence, F is $T_{\mathsf{A}}^{\mathbb{Z}}$-unsatisfiable.

Correctness of Decision Procedure

Theorem

Consider a $\Sigma_{\mathsf{A}}^{\mathbb{Z}} \cup \Sigma$-formula F from the array property fragment of $T_{\mathsf{A}}^{\mathbb{Z}} \cup T$.
The output F_5 of Step 5 of the algorithm is $T_{\mathsf{A}}^{\mathbb{Z}} \cup T$-equisatisfiable to F.

Example

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

Is

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

$T_{\mathsf{A}}^{\mathbb{Z}}$-satisfiable?

0	w	x	y	z	1

10	w	x	y	z	11

Example

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

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

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

Contradiction:

$$\begin{array}{rcl} a[0] \leq a[1] \leq a[5] & \wedge & a[0] \leq a[1] \leq a[5] \\ 0 \leq a[1] \leq 1 & \wedge & 10 \leq a[1] \leq 11 \end{array}$$

Need 1 or 4 in index set.

Undecidable Extensions

- Extra quantifier alternation (e.g., $\forall i \exists j. \cdots$)
- Nested reads: $a[a[i]]$
- No separation: $\forall i.\ F[a[i], i]$ (e.g., $a[i] = i$)
- Arithmetic: $a[i + 1]$ when i is universal
- Strict comparison: $i < j$ when i, j are universal
- Permutation predicate (even weak permutation)

Theory of Sets

Consider a theory T_{set} of sets with signature

$$\Sigma_{\text{set}} : \{\in,\ \subseteq,\ =,\ \subset,\ \cap,\ \cup,\ \setminus\}\ ,$$

where symbols are intended as follows:

- $e \in s$: e is a member of s;
- $s \subseteq t$: s is a subset of t;
- $s = t$: s and t are equal;
- $s \subset t$: s is a *strict* subset of t;
- $s \cap t$ is the intersection of s and t;
- $s \cup t$ is the union of s and t;
- $s \setminus t$, the set difference of s and t, is the set that includes all elements of s that are not members of t.

Theory of Sets (cont)

Let us encode an arbitrary Σ_{set}-formula as a Σ_{E}-formula (or a Σ_{A}-formula). To do so, simply consider the atoms:

- $e \in s$: let $s(\cdot)$ be a unary predicate; then replace

$$\boxed{e \in s \text{ by } s(e)}$$

- $s \subseteq t$: $\forall e.\ e \in s \ \rightarrow\ e \in t$, or in other words, $\forall e.\ s(e) \ \rightarrow\ t(e)$;
- $s = t$: $\forall e.\ s(e) \ \leftrightarrow\ t(e)$;
- $s \subset t$: $s \subseteq t \ \wedge\ s \neq t$;
- $u = s \cap t$: $\forall e.\ u(e) \ \leftrightarrow\ s(e) \wedge t(e)$;
- $u = s \cup t$: $\forall e.\ u(e) \ \leftrightarrow\ s(e) \vee t(e)$;
- $u = s \setminus t$: $\forall e.\ u(e) \ \leftrightarrow\ s(e) \wedge \neg t(e)$.

Theory of Sets (cont)

Atoms with complex terms can be written more simply via "flattening" (as in the Nelson-Oppen procedure); for example, write

$$s \cap (t \cap u) \quad \text{as} \quad s \cap w \ \wedge \ w = t \cap u \ .$$

Then the encodability of an arbitrary Σ_{set}-formula into a Σ_{E}-formula (or a Σ_{A}-formula) follows by structural induction.

Claim

Satisfiability of the quantifier-free fragment of T_{set} is decidable:

- simply apply the decision procedure for T_E (or T_{A}) to the new formula.

Theory of Multisets

Consider a theory T_{mset} of multisets with signature

$$\Sigma_{\text{mset}} : \{C,\ \leq,\ =,\ <,\ \uplus,\ \cap,\ -\}\ .$$

Multisets can have multiple occurrences of elements.
For example: $\{1, 3, 5\}$ is a set and $\{1, 1, 3, 5, 5, 5\}$ is a multiset.
The symbols are intended as follows:

- $C(s, e)$: the number of occurrences (the "count") of e in s;
- $s \leq t$: the count of each element of s is bounded by its count in t;
- $s = t$: element counts are the same in s and t;
- $s < t$: the count of each element of s is bounded by its count in t, and some element has a lower count;
- $s \uplus t$ is the multiset union, whose counts are the element-wise sums of counts in s and t;

Theory of Multisets (cont)

- $s \cap t$ is the multiset intersection, whose counts are the element-wise minima of counts in s and t;
- $s - t$ is the multiset difference, whose counts are the element-wise maxima of 0 and the difference of counts in s and t.

Let us encode an arbitrary Σ_{mset}-formula as a $(\Sigma_{\mathsf{E}} \cup \Sigma_{\mathbb{Z}})$-formula (or a $(\Sigma_{\mathsf{A}} \cup \Sigma_{\mathbb{Z}})$-formula). A multiset is modeled by an uninterpreted function whose range is the nonnegative integers.

Theory of Multisets (cont)

Now consider the atoms:

- $C(s,e)$: let s be a unary function whose range is $\mathbb{N}$; then replace

$$\boxed{C(s,e) \text{ by } s(e)}$$

  and conjoin $\forall e.\ s(e) \geq 0$ to the formula;
- $s \leq t$: $\forall e.\ s(e) \leq t(e)$;
- $s = t$: $\forall e.\ s(e) = t(e)$;
- $s < t$: $s \leq t \ \wedge \ s \neq t$;
- $u = s \uplus t$: $\forall e.\ u(e) = s(e) + t(e)$;
- $u = s \cap t$:

$$\forall e.\ (s(e) < t(e) \wedge u(e) = s(e)) \vee \\ (s(e) \geq t(e) \wedge u(e) = t(e))\ ;$$

Theory of Multisets (cont)

- $u = s - t$:

$$\begin{aligned} \forall e.\ & (s(e) < t(e) \wedge u(e) = 0) \vee \\ & (s(e) \geq t(e) \wedge u(e) = s(e) - t(e)) \ . \end{aligned}$$

As before, encodability follows by structural induction.