Array Fold Logic

Przemyslaw Daca et al.

October 7, 2020

Overview

- Contributions of this paper.
- Array fold logic: syntax, semantics and utilities.
- Theoretical results.
- Tool and experimental results.

Array Fold Logic: Syntax

Implicit Variables: $\{\mathbf{e}, \mathbf{c}_1, \ldots, \mathbf{c}_{m-1}, \mathbf{i}\} = FV^m$

- Array sort, $\mathtt{ASort}$
- Integer sort, $\mathtt{ISort}$
- Boolean sort, $\mathtt{BSort}$
- Integer vectors $\mathtt{VSort}^m$
- Functional constants $\mathtt{FSort}^m = \mathtt{VSort}^m \times \mathtt{ISort} \to \mathtt{VSort}^m$

Array Fold Logic: Syntax

aflsyn.png

Given a set of function branches Br, we can define a control flow graph $G = \langle S, E, \gamma \rangle$.

- $E = \bigcup_{grd \Rightarrow upd \in Br} \{(s_1, s_2) \mid s_1 \models grd \wedge s_2 = ite(\mathbf{s} \leftarrow n \in upd, n, s_1)\}$
- γ is the labeling of edges with the set of formulas $\Phi(grd)$ and $\Phi(upd)$.

Requirement: edges in the same SCC of G update the counters in a monotonic way.

Array Fold Logic: Semantics

$\sigma = \langle \lambda, \mu \rangle$ where $\mu : Var_I \to \mathbb{Z}, \lambda : Var_A \to \mathbb{Z}^*$.
$\kappa = FV^m \to \mathbb{Z}^{m+1}$

aflsema.png

Theoretical Results: Complexity

Definition (symbolic k-counter machine)

An SMC is a tuple $\mathcal{M} = (\vec{\eta}, X, Q, \delta, q^{init})$ where

- $\vec{\eta}$ is a vector of k counters $\eta_1, \ldots, \eta_n$.
- X is a finite set of integer variables.
- Q is a finite set of states.
- $\delta \subseteq Q \times \mathtt{CC}_k(X) \times \mathtt{IC}(X) \times Q \times \mathbb{Z}^k$ is the transition relation.
- $q^{init} \in Q$ is the initial state.

The effect of a transition $(q_1, \alpha, \beta, q_2, \kappa) \in \delta$.
Input constraints $\mathtt{IC}(X)$.
Counter constraints $\mathtt{CC}_k(X)$, here k means the counters are no greater than k.

Reversal and Reversal-Bounded

Definition (Reversal)

A counter machine makes a *reversal* if it makes an alternation between non-increasing and non-decreasing some counter.
A machine is *reversal-bounded* if there exists a constant $c \geq 0$ such that on all accepting runs every counter makes at most c reversal.

Example

Assume there is only one counter.
1,2,3,3,4$, 3, 2, 2,$ **3,5**$, 3, 1$

Translation from Function to SCM

Translation from a functional constant f of $\texttt{FSort}^m$ to an SCM.

Definition

We define the translation of functional constant f of sort $\texttt{FSort}^m$ ocurring in a formula ϕ, as an SCM $\mathcal{M}(f) = (\vec{\eta}, X, Q, \delta, q^{init})$.
Let $G = \langle G, E, \gamma \rangle$ be the CFG defined before, then $\vec{\eta} = \{\mathbf{i}, \mathbf{c_1}, \ldots, \mathbf{c_m}\}$, X are fresh free variables for each integer term T in f, $Q = S$, $q^{init} = 0$. For transitions the formula are translated from $\Phi(grd)$ and $\Phi(upd)$ in G.

The translated SVM is reversal bounded. Why?

$$SCC_1 \rightarrow SCC_2 \rightarrow \cdots \rightarrow SCC_m$$

Parallel composition of SCMs

para.png

Small model property

Lemma

There exists a constant $c \in \mathbb{N}$, such that an AFL formula Φ is satisfiable iff there exists a model σ it maps each variable in X to integer that $\leq 2^{|\Phi|^c}$ and array to sequence of $\leq 2^{|\phi|^c}$ where each integer of the array also lies in the bound.

Give fixed counter values.
Why we want reversal-bounded?

Theorem

The satisfiability problem of AFL is **PSPACE***-complete.*

Membership: NTM.
Hardness: DFA emptiness problem reduced to sat of AFL formula.

Undecidable Extension

Theorem

Array fold logic with $\exists^*\forall^*$ extension is undecidable.

Proof.

Reduction from Hilbert's Tenth Problem to the decidability of quantified AFL. $x = y \cdot z$.

undeafl.png

$\square$

Decision Procedure

Idea: translate the AFL formula ϕ into a quantifier-free PA formula $\psi = \psi_n \wedge \psi_e \wedge \psi_l$.

- ψ_n is part of ϕ that does not contain fold.
- ψ_e is the reduction from the reachability problem of SMC to QFPA.
- ψ_l is the link formula used for linking some constraints between initial and final configuration in ψ_e.

Lemma

The complexity of satifiability of m-AFL for a fixed m is **NP***-complete.*