Satisfying Error Conditions 4

Lecture 16
Overview of Verification Architecture

Program

Specification

Verification Condition

Error Condition

EC satisfiability checker

checkSatLits

checkSatLitsEquality

checkSatLitsArith

checkSatLitsArrays

SAT solver

Davis-Putnam

conjunction of literals

negate

Nelson-Oppen cooperating decision procedures
**EC Satisfiability Checker**

<table>
<thead>
<tr>
<th>{a=b}</th>
<th>{f(a)=f(b)}</th>
<th>{a=c}</th>
<th>{f(a)=f(c)}</th>
<th>Sat?</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

\[(a=b \implies f(a)=f(b)) \wedge b \neq c \wedge f(a) \neq f(c)\]

Explicated tautology removes many other truth assignments.

\{(a=b) \implies \{f(a)=f(b)\}\}
Overview of Verification Architecture

Program

Specification

Verification Condition

Error Condition

SAT solver
Davis-Putnam

checkSatLits

checkSatLitsArith

checkSatLitsEquality

checkSatLitsArrays

EC satisfiability checker

conjunction of literals

Nelson-Oppen cooperating decision procedures

negate
checkSatLitsArith: Difference Constraints

- A special case of linear arithmetic
- All constraints of the form:
  \[ x + c \leq y \]
- \( c \) is a constant
- Special variable \( z \) representing 0

Example
- \( x \leq y \)
- \( y+4 \leq w \)
- \( w-2 \leq x \)
- \( w+1 \leq z \)
• Consider: \( g(g(g(x))) = x \land g(g(g(g(g(x))))) = x \land g(x) \neq x \)
Overview of Verification Architecture

- Program
- Specification

Verification Condition

- SAT solver
- Davis-Putnam

Error Condition

- Davis-Putnam
- Nelson-Oppen
- cooperating decision procedures

EC satisfiability checker

checkSatLits

checkSatLitsArith

checkSatLitsEquality

checkSatLitsArrays
Cooperating Satisfiability Procedures

- Consider equality and arithmetic

\[ f(f(x) - f(y)) \neq f(z) \]
\[ x \leq y \]
\[ y + z \leq x \]
\[ 0 \leq z \]
\[ x = y \]
\[ f(x) = f(y) \]
\[ f(x) - f(y) = z \]
\[ 0 = z \]
\[ f(f(x) - f(y)) = f(z) \]

false
Nelson-Oppen Method (3)

3. Broadcast all discovered equalities and re-run sat. procedures
   • Until no more equalities are discovered or a contradiction arises

\[ f \geq y \geq f \geq x \geq f \geq x \text{ Contradiction} \]
Overview of Verification Architecture

Program

Specification

Verification Condition

Error Condition

negate

SAT solver

Davis-Putnam

EC

satisfiability

checker

checkSatLits

checkSatLitsArith

checkSatLitsEquality

checkSatLitsArrays

conjunction of literals

Nelson-Oppen
cooperating
decision
procedures
Theory of Arrays

• Syntax and informal semantics:
  - If $E$ denotes an address and $\mu$ a heap state then:
  - $\text{sel}(\mu, E)$ denotes the contents of memory cell
  - $\text{upd}(\mu, E, V)$ denotes a new heap state obtained from $\mu$ by writing $V$ at address $E$

• Decision procedure implements following rule

  \[ x = y \Rightarrow \text{sel}(\text{upd}(\mu, x, v), y) = v \]

  \[ x \neq y \Rightarrow \text{sel}(\text{upd}(\mu, x, v), y) = \text{sel}(\mu, y) \]

  - what if $x = y$ is unknown?
Theory of Arrays

• Syntax and informal semantics:
  - If $E$ denotes an address and $\mu$ a heap state then:
  - $\text{sel}(\mu, E)$ denotes the contents of memory cell
  - $\text{upd}(\mu, E, V)$ denotes a new heap state obtained from $\mu$ by writing $V$ at address $E$

• Decision procedure implements following rule

\[ x \neq y \lor \text{sel}(\text{upd}(\mu, x, v), y) = v \]
\[ x = y \lor \text{sel}(\text{upd}(\mu, x, v), y) = \text{sel}(\mu, y) \]

  - what if $x = y$ is unknown?
  - *non-convex theory*: input facts entail disjunction of equalities, but do not entail any individual equality
  - can add SAT literal \{x=y\}
Overview of Verification Architecture

Program

Specification

Verification Condition

Error Condition

EC satisfiability checker

checkSatLits

checkSatLitsArith

checkSatLitsEquality

checkSatLitsArrays

SAT solver Davis-Putnam

conjunction of literals

Nelson-Oppen cooperating decision procedures
Example

//@ requires x != y
//@ ensures *x < *y+1

void sort2(int *x, int *y) {
    if (*x > *y) {
        int t = *x;
        *x = *y;
        *y = t;
    }
}

• Class Challenge: Use theorem proving techniques to cooperatively verify the correctness of sort2
Overview of Verification Architecture

Program

Verification Condition

Katia

checkSatLits

conjunction of literals

checkSatLitsArith

Nathan

checkSatLitsEquality

Harry

checkSatLitsArrays

Min

SAT solver

Davis-Putnam

EC

satisfiability checker

Nelson-Oppen cooperating decision procedures

Dorrit

equalize
Verification Condition

\[ VC = VC1 \land VC2 \]

\[ VC1 = \neg (sel(u,x) > sel(y,y)) \]
\[ \Rightarrow (sel(u,x) < sel(u,y)+1) \]

\[ VC2 = (sel(u,x) > sel(y,y)) \]
\[ \Rightarrow (sel(u_2,x) < sel(u_2,y)+1) \]

\[ u_2 = store(store(u,x,sel(u,y)), y, sel(u,x)) \]
Error Condition

EC = EC1 || EC2

EC1 = !(sel(u,x) > sel(y,y))
    && !(sel(u,x) < sel(u,y)+1)

EC2 = (sel(u,x) > sel(y,y))
    && !(sel(u2,x) < sel(u2,y)+1)

u2 = store(store(u,x,sel(u,y)),
            y,sel(u,x))
Satisfying Assignment: Attempt 1

\(! (\text{sel}(u,x) > \text{sel}(y,y))\)
\&\& \(! (\text{sel}(u,x) < \text{sel}(u,y)+1)\)

becomes

\((\text{sel}(u,x) \leq \text{sel}(y,y))\)
\&\& \((\text{sel}(u,x) \geq \text{sel}(u,y)+1)\)
Satisfying Assignment: Attempt 2

\[(\text{sel}(u, x) > \text{sel}(y, y)) \land \neg (\text{sel}(u_2, x) < \text{sel}(u_2, y) + 1)\]

where

\[u_2 = \text{store}(\text{store}(u, x, \text{sel}(u, y)), y, \text{sel}(u, x))\]