UFO: Verification with Interpolants and Abstract Interpretation

Arie Gurfinkel and Sagar Chaki
Software Engineering Institute
Carnegie Mellon University

Aws Albarghouthi, Yi Li and Marsha Chechik
University of Toronto
UFO

- A framework and a tool for software verification
- Tightly integrates interpolation- and abstraction-based techniques

Check it out at:
http://bitbucket.org/arieg/ufo

References:
[SAS12] Craig Interpretation
[TACAS12] From Under-approximations to Over-approximations and Back
Verification with INTERP and AI

- Uses Cutpoint Graph (CPG)
- Maintains an unrolling of CPG
- Computes disjunctive invariants
- Uses novel powerset widening

- Uses SMT to check for CEX
- DAG Interpolation for Refinement
- Guided by AI-computed Invs
- Fills in “gaps” in AI
Implementation in UFO Framework

1. **C Program with assertions**
2. **C to LLVM**
3. **Optimizer**
4. **Cutpoint Graph**

- **Mathsat**
- **Z3**

**SMT interface**

- **ARG Constructor**
- **Refinement Strategy**
- **Abstract Post**
- **Expansion Strategy**
UFO in a Nutshell

Iteration 1

Unlabeled
Pred. abs. label
Interpolant label
UFO in a Nutshell

Iteration 1

Refinement

Iteration 2

false

Unlabeled
Pred. abs. label
Interpolant label
UFO in a Nutshell

Iteration 1:

Imprecise post $\rightarrow$ UD
Explore from root $\rightarrow$ OD

Iteration 2:

Refinement

Unlabeled
Pred. abs. label
Interpolant label

E
L

E
L
Secret Sauce

UFO Front-End

Boxes Abstract Domain

DAG Interpolation

Parallel
UFO Front End

In principle simple, but in practice very messy

- CIL passes to normalize the code (library functions, uninitialized vars, etc.)
- `llvm-gcc` (without optimization) to compile C to LLVM bitcode
- `llvm-opt` with many standard, custom, and modified optimizations
  - lower pointers, structures, unions, arrays, etc. to registers
  - constant propagation + many local optimizations
  - difficult to preserve *intended* semantics of the benchmarks
  - based on very old LLVM 2.6 (newer version of LLVM are “too smart”)

Many benchmarks discharged by front-end alone

- 1,321 SAFE (out of 1,592) and 19 UNSAFE (out of 380)
Boxes Abstract Domain: Semantic View

Boxes are “finite union of box values”
(alternatively)
Boxes are “Boolean formulas over interval constraints”
Linear Decision Diagrams in a Nutshell*

Linear Decision Diagram

- False edge: false
- Decision node: $x + 2y < 10$
- True edge: $z < 10$
- False terminal: 0
- True terminal: 1

Linear Arithmetic Formula

- $(x + 2y < 10) \text{ OR } (x + 2y \geq 10 \text{ AND } z < 10)$

Compact Representation

- Sharing sub-expressions
- Local numeric reductions
- Dynamic node reordering

Operations

- Propositional (AND, OR, NOT)
- Existential Quantification

*joint work w/ Ofer Strichman
DAG Interpolants: Solving the Refinement Prob.

Given a DAG $G = (V, E)$ and a labeling of edges $\pi: E \rightarrow \text{Expr}$. A **DAG Interpolant** (if it exists) is a labeling $I: V \rightarrow \text{Expr}$ such that

- for any path $v_0, \ldots, v_n$, and $0 < k < n$,
  \[ I(v_k) = \text{ITP} \left( \pi(v_0) \land \ldots \land \pi(v_{k-1}), \quad \pi(v_k) \land \ldots \land \pi(v_n) \right) \]
- $\forall (u, v) \in E : (I(u) \land \pi(u, v)) \Rightarrow I(v)$

\[
\begin{align*}
I_2 &= \text{ITP} \left( \pi_1, \quad \pi_8 \right) \\
I_2 &= \text{ITP} \left( \pi_1, \quad \pi_2 \land \pi_3 \land \pi_6 \land \pi_7 \right) \\
&\quad \ldots \\
(I_1 \land \pi_1) \Rightarrow I_2 \\
(I_2 \land \pi_8) \Rightarrow I_7 \\
(I_2 \land \pi_2) \Rightarrow I_3 \\
&\quad \ldots
\end{align*}
\]
Parallel Verification Strategy

Run 7 verification strategies in parallel until a solution is found

- **cpredO3**
  - all LLVM optimizations + Cartesian Predicate Abstraction
- **bpredO3**
  - all LLVM optimizations + Boolean PA + 20s TO
- **bigwO3**
  - all LLVM optimizations + BOXES + non-aggressive widening + 10s TO
- **boxesO3**
  - all LLVM optimizations + BOXES + aggressive widening
- **boxO3**
  - all LLVM optimizations + BOX + aggressive widening + 20s TO
- **boxesO0**
  - minimal LLVM optimizations + BOXES + aggressive widening
- **boxbpredO3**
  - all LLVM opts + BOX + Boolean PA + aggressive widening + 60s TO
UFO Family

Whale [VMCAI12]

- Interpolation-based interprocedural analysis
- Interpolants as procedure summaries
- State/transition interpolation
  - a.k.a. Tree Interpolants

UFO [TACAS12]

- Refinement with DAG interpolants
- Tight integration of interpolation-based verification with predicate abstraction

Vinta [SAS12]

- Refinement of Abstract Interpretation (AI)
- AI-guided DAG Interpolation
Thank You!

http://bitbucket.org/arieg/ufo
Contact Information

Presenter
Arie Gurfinkel
RTSS
Telephone: +1 412-268-7788
Email: arie@cmu.edu

Web:
www.sei.cmu.edu
http://www.sei.cmu.edu/contact.cfm

U.S. mail:
Software Engineering Institute
Customer Relations
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
USA

Customer Relations
Email: info@sei.cmu.edu
Telephone: +1 412-268-5800
SEI Phone: +1 412-268-5800
SEI Fax: +1 412-268-6257
NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

This Presentation may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.