HiFrog: Interpolation-based Software Verification using Theory Refinement

Sepideh Asadi

joint work with Karine Even Mendoza, Grigory Fedyukovich, Antti Hyvärinen, Hana Chockler, Natasha Sharygina

Formal Verification and Security Lab
University of Lugano (USI), Switzerland

FMCAD 2017
What is HiFrog?

- An SMT-based bounded model checker for C
- Computes and reuses **Function Summaries**
  - Based on Craig interpolation
What is HiFrog?

- An SMT-based bounded model checker for C

- Computes and reuses Function Summaries
  - Based on Craig interpolation

- Controllable interpolation system for SMT
  - Flexible in Size & Strength
  - Compact and readable summaries
What is HiFrog?

- An SMT-based bounded model checker for C
- Computes and reuses **Function Summaries**
  - Based on Craig interpolation
- Controllable interpolation system for SMT
  - Flexible in **Size & Strength**
  - **Compact and readable** summaries
- Automatic adjustment of abstraction using different theories
  - **Theory Refinement**
HiFrog and Function Summarization

sources + assertions

parser

assertions optimizer

SSA slicing

symbolic execution

SMT encoder

EUF

LRA

BV

BOOL

Interpolating SMT solver

theory solvers

proof

proof compressor

itp for EUF

itp for BOOL

itp for LRA

Interpolating SMT solver

Function Summaries

EUF

LRA

BOOL

assertion holds!

assertion violated

& error trace

assertion holds!

UNSAT
HiFrog and Function Summarization

sources + assertions

parser
assertions optimizer

symbolic execution
SSA slicing

SMT encoder

EUF
LRA
BV
BOOL

assertion violated
& error trace

×

theory refiner

assertion holds!

✓

UNSAT

Interpolating SMT solver

theory solvers

proof

proof compressor

itp for EUF

itp for BOOL

itp for LRA

Function Summaries

EUF
LRA
BOOL

interpolation-based summaries
HiFrog and Theory Refinement
HiFrog and Theory Refinement

Program + Assertions

Symbolic Encoding
HiFrog and Theory Refinement

Program + Assertions

Symbolic Encoding

initial entire encoding

less precise theory

OpenSMT
HiFrog and Theory Refinement

Program + Assertions

Symbolic Encoding

0

1

initial entire encoding

OpenSMT

less precise theory

2

Safe

UNSAT
HiFrog and Theory Refinement

Program + Assertions

Symbolic Encoding

Initial entire encoding

OpenSMT

Safe

UNSAT

less precise theory

SAT + model

less precise theory
HiFrog and Theory Refinement

Program + Assertions

0

Symbolic Encoding

1

initial entire encoding

3

sequence of all terms

Counter-example validator

2

SAT + model

less precise theory

OpenSMT

Safe
HiFrog and Theory Refinement

Program + Assertions

0. Symbolic Encoding

1. Initial entire encoding

2. SAT + model

3. Sequence of all terms

4. Term vs CEX

Safe

OpenSMT

OpenSMT

Precise theory
HiFrog and Theory Refinement

Program + Assertions

Symbolic Encoding

Counter-example validator

Counter-example validator

OpenSMT

OpenSMT

Safe

Program + Assertions

Symbolic Encoding

Counter-example validator

OpenSMT

OpenSMT

Safe
HiFrog and Theory Refinement

Program + Assertions

Symbolic Encoding -> Counter-example validator

3 sequence of all terms

Counter-example validator -> Refiner

1 initial entire encoding

Refiner -> OpenSMT

2 less precise theory

OpenSMT -> OpenSMT

Safe

OpenSMT -> OpenSMT

precise theory

4 term vs CEX

OpenSMT -> OpenSMT

UNSAT

Sat

OpenSMT -> Do Nothing!

Sat
HiFrog and Theory Refinement

Program + Assertions

Symbolic Encoding

Counter-example validator

Refiner

OpenSMT

Precise theory

Safe

OpenSMT

Less precise theory

Binding

Precise theory

SAT vs CEX

UNSAT

Do Nothing!

Term vs CEX

Term(s)

Local refinements

Initial entire encoding

Sequence of all terms

0

1

2

3

4

5

6
HiFrog and Theory Refinement

1. **Symbolic Encoding**: Initial entire encoding.
   - **Term(s)** to refine
   - **Safe/Unsafe**: Symbolic Encoding

2. **Binding**: Safe/Unsafe
   - **OpenSMT**: Theory Refinement
   - **Safe/Unsafe**: OpenSMT

3. **Counter-example validator**: Sequence of all terms
   - **Term vs CEX**: UNSAT
   - **Safe/Unsafe**: Counter-example

4. **Refiner**: SAT + model
   - **Safe/Unsafe**: OpenSMT

5. **OpenSMT**: Precise theory
   - **Safe/Unsafe**: OpenSMT

6. **OpenSMT**: Initial entire encoding
   - **Safe/Unsafe**: OpenSMT

7. **Unsafe + Counter-example**: Unsatisfiable
   - **Safe/Unsafe**: Unsafe

Program + Assertions

- **Do Nothing!**
http://verify.inf.usi.ch/hifrog/

Looking forward to seeing you at poster sessions!