Incremental CSA

Philipp Dominik Schubert,  Balázs Benics
Agenda

Motivation
The engine as of today
The prototype
Preliminary results
Questions
Agenda

Motivation

The engine as of today
The prototype
Preliminary results
Questions
Motivation
Motivation

symbolic execution

parsing

(Seconds)

Translation units
Agenda

Motivation

The engine as of today

The prototype

Preliminary results

Questions
Call graph

```
Call graph

f1 -> f2
f3
f4
f9

f6 -> f7
f8
f5
```

©2023, SonarSource S.A, Switzerland.
Call graph

```
  e1
   ├── f1
   │    │
   │    └── f4
   │        │
   │        └── f9
   └── f2
        │
        └── f5
  e2
   ├── f6
   │    │
   │    └── f8
   │         │
   │         └── f9
   └── f7
```

©2023, SonarSource S.A, Switzerland.
Inlining heuristic side effects

Has body? [yes](#) [no](#)

Wants to inline? [yes](#) [no](#)

Map arguments and jump

Invalidate arguments and globals
Inlining heuristic side effects

- Can decide not to inline, even if we could.
- Once inlined → always inlined.
- Inlined functions considered “covered”
- Might remember to never inline something again.
Call graph
Call graph
Call graph
Analysis times per entry point
More frequent

Runs longer

occurrences

Entry point running times

0 1 sec 2 sec 3 sec 4 sec 5

©2023, SonarSource S.A, Switzerland.
Let’s fix long analysis times…

- Serious engineering
- Unchanged since the dawn
- Multiple stakeholders
- High risk
Agenda

Motivation

The engine as of today

The prototype

Preliminary results

Questions
Guiding principles

● Minimal
● Self-contained
● Significant speedup for the “usual” cases
Prototype architecture

- Analysis cache
- Oracle
- Report replayer
- Report recorder
Prototype architecture

- Analysis cache
- Oracle
- Report replayer
- Report recorder
Prototype architecture

- Analysis cache
- Oracle
- Report replayer
- Report recorder
Prototype architecture

- Analysis cache
- Oracle
- Report replayer
- Report recorder
Prototype architecture

- Analysis cache
- Oracle
- Report replayer
- Report recorder
Anchor decls, relative references

```c
namespace num {
    void add(int x, int y) {
        return x + y;
    }
}
```
namespace num {
    void add(int x, int y) {
        return x + y;
    }
}

FunctionDecl add 'int (int, int)'
|-ParmVarDecl used x 'int'
|-ParmVarDecl used y 'int'
 ` CompoundStmt
  ` ReturnStmt
   ` BinaryOperator 'int' '+'
    |-ImplicitCastExpr 'int' <LValueToRValue>
     | ` DeclRefExpr 'int' lvalue ParmVar 'x' 'int'
     `-ImplicitCastExpr 'int' <LValueToRValue>
       `-DeclRefExpr 'int' lvalue ParmVar 'y' 'int'

seq{2,0,0,1,0}
Relocatable diagnostics

- Anchor decl
- AST index sequence
- Getter function
- Message
Oracle

- Preprocessor token watcher
- Hash:
  - source text
  - call dependencies
  - type dependencies
Diagnostic relocation

- Relocate diagnostics eagerly
- Might have absolute line refs
  - “Control jumps to line 80”
  - “[...] call to alloca() on line 55 returned to caller”
  - “Loop condition is false. Execution continues on line 44”
Agenda

Motivation

The engine as of today

The prototype

Preliminary results

Questions
Preliminary results
Whitespace, comment changes

Default Analysis

parsing: 48 ms
symbolicExecution: 1984 ms

gzip:inflate.c

Incremental Analysis

parsing: 62 ms
symbolicExecution: 42 ms
Vim

Default analysis time

Time saved with incremental mode

(minutes)

Commits
Vim

Default analysis time

Time saved with incremental mode

(seconds)

Translation units
Bitcoin

Time saved with incremental mode

(seconds) Translation units
Weaknesses

Changes in commonly inlined function ("sharing")

- e1
- e2
- e3
- e4
- e5

©2023, SonarSource S.A. Switzerland.
Entry point running times and sharing

(seconds)  Translation units  (default run)

Shared nodes
Entry point running times and sharing

(seconds)  Translation units  (default run)  Shared nodes

©2023, SonarSource S.A, Switzerland.
Conclusion

- Moderate improvements overall
- Only a few cache-hits for C++
- A lot of potential
- Works well for trivial, narrow changes
Agenda

Motivation

The engine as of today

The prototype

Preliminary results

Questions