Toward General Diagnosis of Static Errors
Danfeng Zhang and Andrew C. Myers (Cornell University)

A general, concise and accurate diagnostic method for static errors

Overview

Static Program Analyses
- Many flavors
  - Type systems
  - Dataflow analysis
  - Information-flow analysis
- Useful properties
  - Type safety
  - Memory safety
  - Information-flow security
- But, sometimes hard to localize the cause of static errors

Examples
- Better error report is needed

Constraint Graph

General representation of program analyses

Syntax of Constraints

\[ C := C_1 \land C_2 \mid A \land C \]

\[ F := E_1 \land E_2 \]

\[ E := a \mid \{ E_1, \ldots, E_m \} \mid F(E) \mid E_1 \circ E_2 \mid E_1 E_2 \mid E_1 \oplus E_2 \mid E_1 \land E_2 \mid T \land T \]

Constraints

OCaml

let rec foo(ts) =
  let acc = (0.0, 0.0) in
  List.rev ts

Evaluation

Correctness metric
- OCaml: user’s fix with larger time stamp
- Jif: errors marked by the programmer

Inferring Likely Missing Hypothesis

Idea: find minimal weakest hypothesis

\[ \arg \max_p P_W(H) \]

Inferring Likely Wrong Elements

Idea: heuristic search

\[ \arg \max_p p_{\text{true}}(H) \]

- “A” search
  - Optimal: return all most likely wrong elements
  - Efficient: \( \sim 10 \) seconds when the search space is over \( 2^{1000} \)

- The wrong elements are likely to be
  - Simple (\( E_i \))
  - Used on all unsatisfiable paths
  - Not used often on satisfiable paths (\( \| E_i \| \))

- But, sometimes hard to localize the cause of static errors

Better error report is needed

Information-Flow Analysis (Jif)

Bayes’ theorem

& simplifying assumptions

Simplifying Assumptions

All elements are equally likely to be wrong (with \( P_1 \))
Errors are unlikely (with \( P_2 < 0.5 \)) to appear on satisfiable paths

Error Diagnosis

Idea: maximum a posteriori (MAP) estimation

\[ \arg \max_p P(E_i, H) \]

Our tool identifies error locations more accurately

Correctness metric
- OCaml: user’s fix with larger time stamp
- Jif: errors marked by the programmer

Inferring Likely Missing Hypothesis

Idea: find minimal weakest hypothesis

\[ \arg \max_p P_W(H) \]

- Simplicity is not the only metric
  - \( T \leq L \) “explains” all errors

- Likely missing hypothesis is weak and simple

Errors

Our tool finds a correct error
Other tool misses the error

Our tool finds several errors
Other tool misses one error

Our tool finds correct error but the error is not the one

Our tool finds the error
Other tool misses the error

Our tool finds correct error
Other tool misses the error

Correctness metric
- OCaml: user’s fix with larger time stamp
- Jif: errors marked by the programmer

Analysis of 336 programs with type mismatch errors

OCaml

Same corpus used by the

Seminor tool [Lerner et al. ’07]

16 previous collected buggy programs

Contains both error types

The OCaml compiler

The Jif compiler (Wrong elements)

Missing hypothesis