Program Slicing in the Presence of Variability

Sandro Schulze, Frederik Kanning, FOSD meeting, 4.-7.05.2014
Problem

Testing: incremental pairwise CIT

...
Motivation

Change

e.g., bug fix, feature request

V 1.0

? 

V 2.0
Determining the Impact of Change

Features affected by a certain change?

Variants affected by a certain change?

Which tests have to be re-run?

Regression Testing
Program Slicing (Mark Weiser, 1984)

What program statements can influence specified variables at a certain statement?

Slicing criterion:

<16, \{\text{lines}\}>

```plaintext
1 read(text);
2 read(n);
3 lines = 1;
4 chars = 1;
5 subtext = "";
6 c = getChar(text);
7 while (c != EOF)
   8   if (c == \n)
      9     then lines = lines + 1;
     10     chars = chars + 1;
     11   else chars = chars + 1;
     12   if (n != 0)
     13     then subtext = subtext ++ c;
     14     n = n - 1;
     15     c = getChar(text);
   16 write(lines);
   17 write(chars);
   18 write(subtext);
```
Static Forward Slicing

Determining how a modification affects (part of) a program.

```c
void main()
{
    int a, b, c;
    a = 32;
    b = 23;
    c = 2 * b;
    print(a);
    print(c);
    a += 1;
    c = b;
    print(a);
    print(c);
}
```
Variability-Aware Forward Slicing

```c
void main()
{
    int a, b, c;

    a = 32;
    b = 23;
    c = 2 * b;

    print(a);
    print(c);

    a += 1;
    c = b;

    print(a);
    print(c);
}
```

```c
// lines 1-3 omitted

a = 32;
b = 23;
c = 2 * b;

print(a);
print(c);

#ifdef X
    a = b;
#endif

#ifdef Z
    c = b;
#else
    a = 2*b;
#endif

print(a);
print(c);
```
General Approach

- Data flow dependence
  - Compute Def-Use-Chains via reachable uses analysis
- Control dependence analysis
  - How does dependent predicates affect definitions in subordinate branches?

- Taking interdependence into account

- Slice $\rightarrow$ transitive closure of all dependencies

- Based on TypeChef and Monotone framework
Which features are affected if there is a change in the respective statement?

Under which configurations particular statements belong to the slice?
Control Flow Dependence

void control()  
{
    int x, y, a = 0;
    x = 0;
    if(y < 5) {
        y += 1;
        #ifdef X
        a = 2*x;  //\<X>
        #endif
    #endif
    }
    print(x);
    print(y);
}

Slicing criterion: <5, \{x\}>
Slicing criterion: <5, \{x\}>

Slicing criterion: <5, \{x\}>
Slicing criterion: <5, \{x\}>

#ifdef Y
    y += a;  //\{Y\}
#endif
#ifdef X
    a += 1;  //\{X\}
#endif
print(x);  //\{Y\}
print(y);  //\{Y\}
Collecting all dependencies

Slicing criterion

Iterating through all indirect dependencies.
Infrastructure

TypeChef

- Source Code
  - Annotations
  - Macros
- Feature Model

Preprocessor → Lexer → Parser → Type Checking → Monotone Framework

AST

Includes

SAT Solver

Slicing

- Variability Analysis
- Evolutionary Analysis
- Control Dependence Analysis
- Reachable Uses Analysis

CodRunner

- Slicing

Reachable Uses Analysis

Control Dependence Analysis

Evolutionary Analysis

Variability Analysis
A Long Road To Go…

We are probably here
Current State & Ongoing Work

Currently implemented/Ongoing:
- Data flow dependency analysis (on statement level)
- Control dependence (work in progress)
- Changes to function parameters (work in progress)

Future work
- Support more tricky constructs → pointers, go-to
- Interprocedural forward slicing
- Optimizations (performance, resolving variability)
- Evaluation wrt regression testing
Ultimate Goal

Interprocedural, variability-aware slicing