Program Slicing
M. Weiser, 1981

Elissa Newman
17-939a
10/8/01

Program Slices

- Program slices are sets of statements that could possibly affect all the given variables at a given line
- A slicing criterion is a tuple \( (i, v) \)
  - \( i \) is the line number of the program to slice at
  - \( v \) is the set of variable values to be observed
- An algorithm based on sets of affected statements starting from the program termination point and building levels of affected statements is given
- This algorithm is based on data-flow and control-flow analysis

Pro-forma abstract

The effectiveness of limiting the scope of code considered for debugging and enhancement tasks has been demonstrated. An enhanced method is described for the limitation of scope to relevant code based on taking program slices, which are sets of statements that affect the value of a given set of variables at a given line in the program. These slices can then be studied in isolation for debugging and maintenance tasks. A study on the usage of program slices implicitly by programmers is proffered, as well as several small examples of program slices.

Means to an End

- First, Weiser observed that programmers were already using program slicing algorithms in their heads to reduce the amount of code they needed to consider and understand when debugging
- Weiser formally defines a program via variables assignments/references and control flow. He uses this definition to provide an algorithm that computes the numbers of affected lines of code given a point in the program
- Program slices are effectively smaller executable versions of the source code reduced in scope to only those values that matter
Verification Method (1)

- Weiser tests his hypothesis that programmers use slicing for debugging activities with an empirical study.
- Experienced programmers debug 3 small programs.
- Programmers are then tested on recognition of "algorithms" from the program.
  - 2 adjacent - one with buggy statements
  - 3 non-adjacent - one is a slice on the buggy statement
- Subjects remembered the slice as well as the adjacent algorithms (as previously predicted).

Verification Method (2)

- An additional means of verification was to code a slicing tool and verify that slices provide a measurable advantage.
  - Number of lines in program reduced by more than half in "larger" programs (>60 lines).

More Applications

- Slicing-based metrics are suggested that measure things such as:
  - the overlap between slices
  - the amount of size reduction between a program and its slices
  - relation of slices to the original structure of the program
  - number of statements that are used in each slice.
- Also a notion of separation of concerns:
  - Sample slices from students' work on compilers
  - Grouped by type of computation of the code.