COURSE NAME

SOFTWARE QUALITY AND TESTING

CSC 4133

(UNDERGRADUATE)

CHAPTER 8

CONTROL FLOW TESTING

Dr. Mohammad Mahmudul Hasan
Associate Professor
Department of Computer Science
American international university – Bangladesh
Email: m.hasan@aiub.edu
Web: http://cs.aiub.edu/profile/m.hasan

Google Scholar

LinkedIn

ORCID

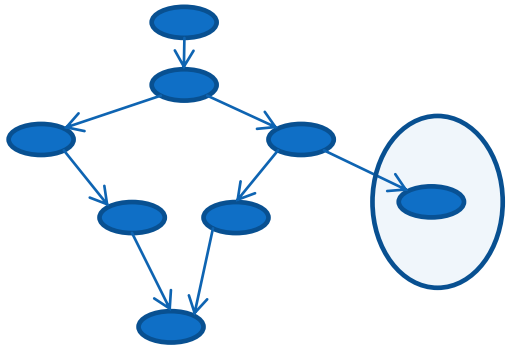
WEB OF SCIENCE™

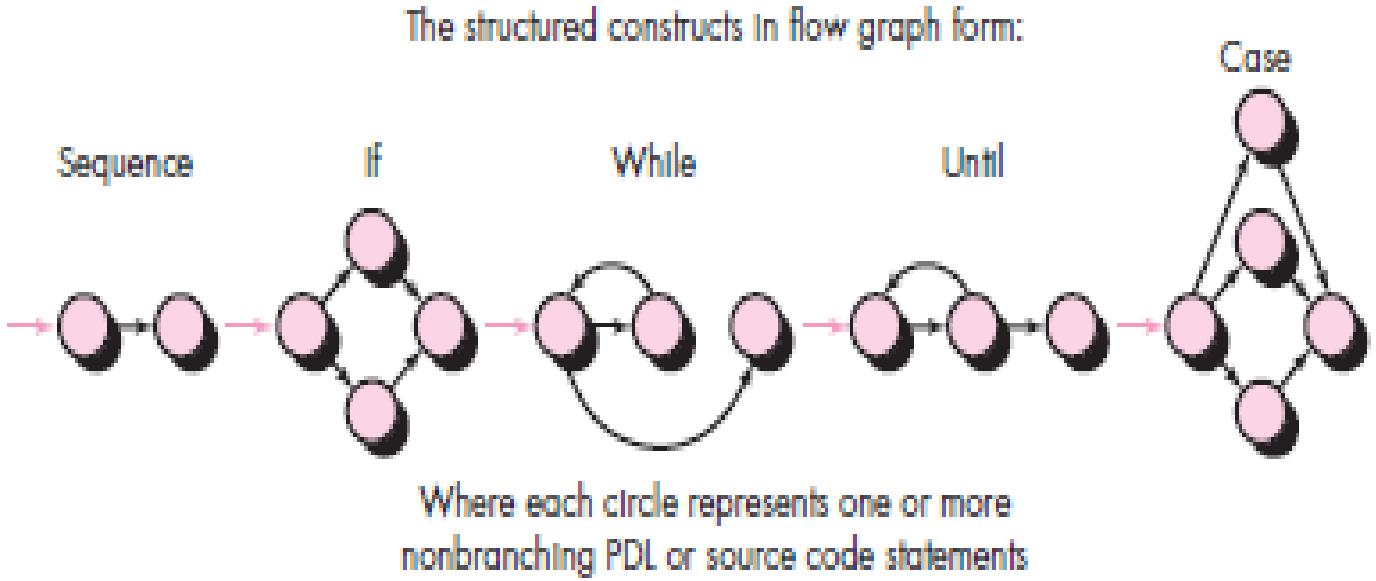
STATIC ANALYSIS

- Static Analysis is the task of analyzing a test object (e.g., source code, script, requirements) without executing the test object.
- Possible aspects to be checked with static analysis are:
  - Programming rules and standards (syntax, convention)
  - Programming design (control flow analysis)
  - Use of data (data flow analysis)
  - Complexity of the programming structure (metrics e.g. , Cyclomatic number)

CONTROL FLOW TESTING - CFT

- The Control flow diagram shows the program flow and allows for the detection of "dead branches" and "unreachable code".
- Aim is to find defects caused by wrong construction of program code (dead branches)
- Control flow graph, Directed graph
  - Nodes represent statement
  - Edge represent control flow as decision



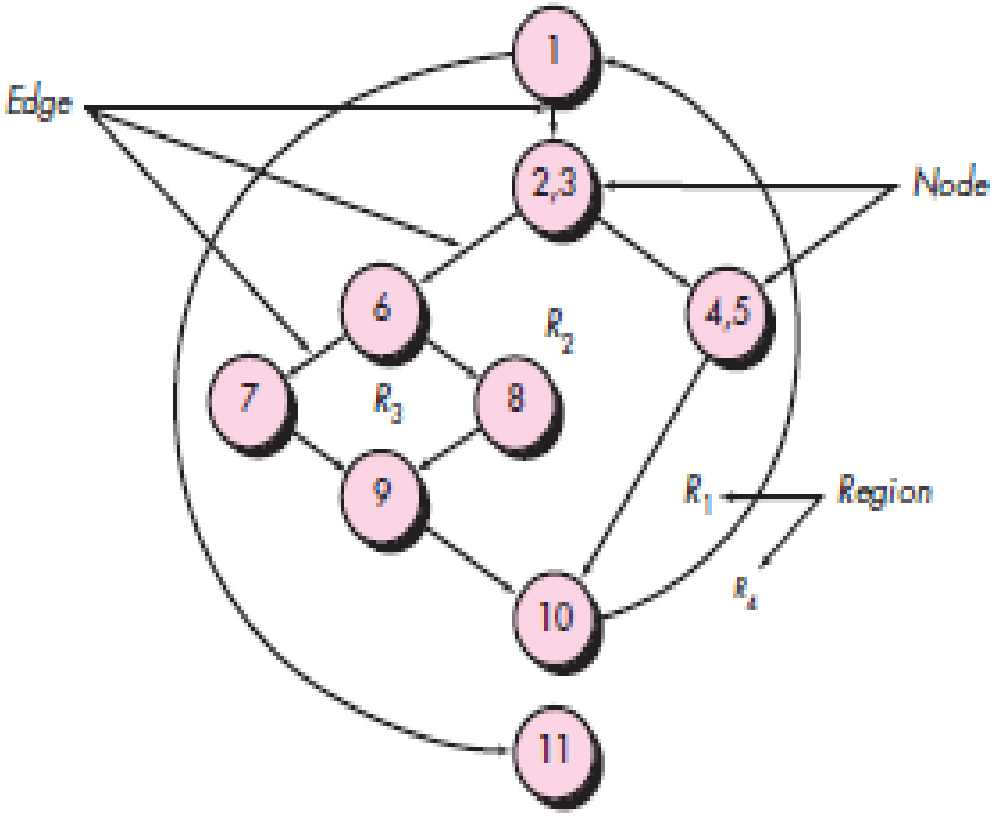


CONTROL FLOW TESTING - CFT

- **Two kinds of basic program statements:**
  - Assignment statements (Example: x = 2*y;)
  - Conditional statements (Example: if(), for(), while(), …)
- **Control flow**
  - In absence of conditional statements, program instructions are executed in the sequence they appear
  - Successive execution of program statements is viewed as flow of control
  - Conditional statements alter the default flow, sequential control flow in a program unit
- **Program path**
  - A program path is a sequence of statements from entry to exit
  - There can be a large number of paths in a program which have input, expected output for each path

CONTROL FLOW TESTING - CFT

- Structural testing performed by programmers to test code written by them
- Test cases are applied to small units of code, such as a function or method for unit testing
- The main idea in CFT is to appropriately select a few paths in a program unit and observe whether or not the selected paths produce the expected outcon

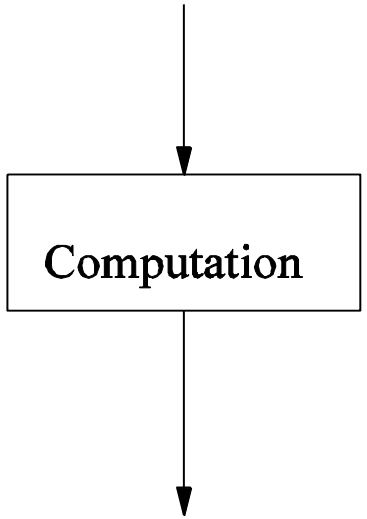


Path 1: 1-11
Path 2: 1-2-3-4-5-10-1-11
Path 3: 1-2-3-6-8-9-10-1-11
Path 4: 1-2-3-6-7-9-10-1-11

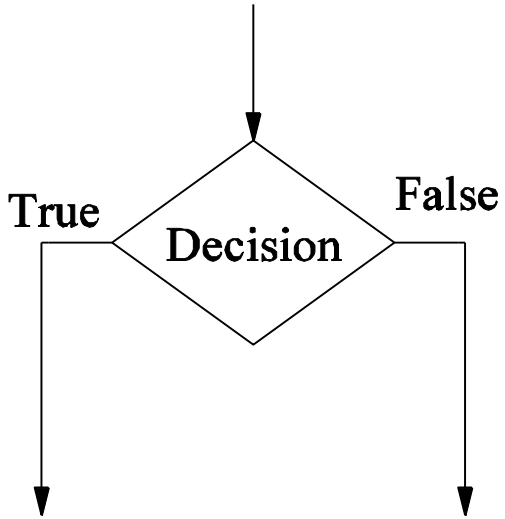
The path 1-2-3-4-5-10-1-2-3-6-8-9-10-1-11 is not considered to be an independent path because it is simply a combination of already specified paths and does not traverse any new edges.

CONTROL FLOW GRAPH - CFG

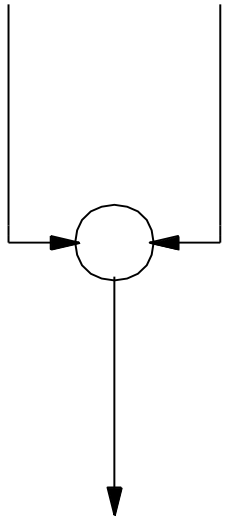
- CFG is a graphical representation of a program's control structure.
- Symbols are:



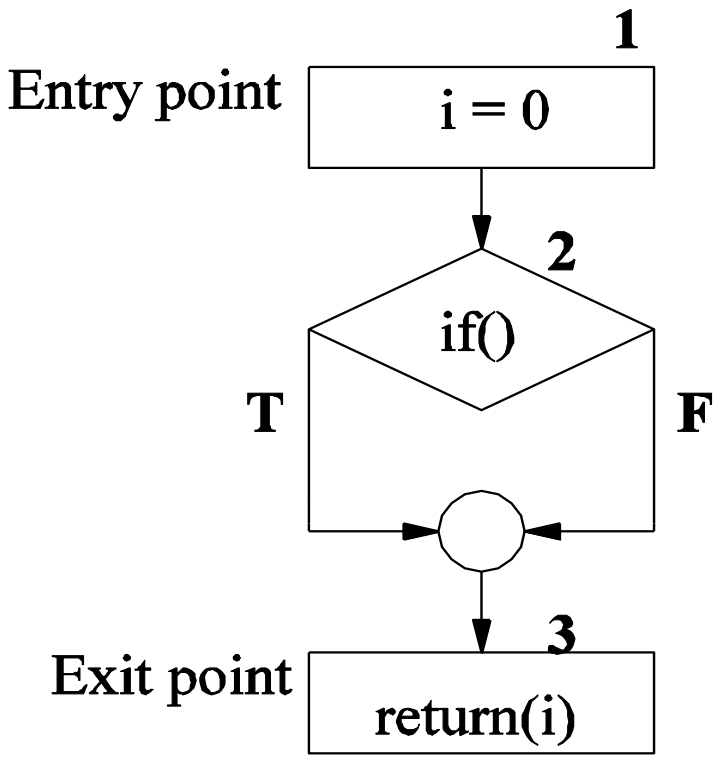
Sequential Computation



Decision Point



Merge Point



PATH SELECTION CRITERIA

- There are many paths between the entry and exit points of a typical routine
- Advantages of selecting paths based on defined criteria:
  - Ensure that all program constructs are executed at least once
  - Repeated selection of the same path is avoided
  - One can easily identify what features have been tested and what not

PATH SELECTION CRITERIA

1. Select all paths
   - 100% path coverage; execute all possible control flow paths through the program
   - If all the paths in a CFG are selected, then one can detect all faults, except those due to missing path errors, also we say that the all-path selection criteria has been satisfied
   - A program may contain a large number of paths, or even infinite # of paths
   - Selecting all the inputs will exercise all the program paths
   - All-path selection criterion is desirable, but difficult to achieve in practice

PATH SELECTION CRITERIA

2. Select paths to achieve complete statement coverage
   - Statement coverage means executing individual program statements and observing the output.
   - 100% statement coverage means all the statements have been executed at least once.
   - Cover all assignment statements.
   - Cover all conditional statements.
   - Less than 100% statement coverage is unacceptable.

PATH SELECTION CRITERIA

3. Select paths to achieve complete branch coverage

- A branch is an outgoing edge from a node in a CFG.
  - A condition node has two outgoing branches – corresponding to the True and False values of the condition.
  - All rectangular nodes have at most one outgoing branch (edge)
  - The exit node (end node) of a CFG does not have an outgoing branch
- Covering a branch means selecting a path that contains the branch.
- 100% branch coverage means selecting a set of paths i.e., every branch is included in at least one path

4. Select paths to achieve predicate coverage

If all possible combinations of truth values of the conditions affecting a path have been explored under some tests, then we say that predicate coverage has been achieved.

EFFECTIVENESS & LIMITATION OF CONTROL FLOW TESTING

EFFECTIVENESS

- Unit testing is dominated by control-flow testing methods
- Statement and branch testing dominates control-flow testing
- Studies show that control-flow testing catches 50% of all bugs caught during unit testing and about 33% of all bugs
- Control-flow testing is more effective for unstructured (goto, jump) code than for code that follows structured programming (block structure - for, while loops)

LIMITATION

- Interface mismatches and mistakes are not caught.
- Not all initialization mistakes are caught by control-flow testing
- Specification mistakes are not caught

REFERENCES

- Software Testing And Quality Assurance – Theory and Practice - Kshirasagar Naik & Priyadarshi Tripathy
- Software Quality Engineering: Testing, Quality Assurance and Quantifiable Improvement - Jeff Tian