Software Re-Engineering

Lecture: 13

Dr. Syed Muazzam Ali Shah
Department of Software Engineering
National University of Computer & Emerging Sciences
muazzam.ali@nu.edu.pk

Sequence [**Todays Agenda**]

Content of Lecture

Reverse Engineering – Techniques

- Lexical Analysis
- Syntactic Analysis
- **Control Flow Analysis**
- Data Flow Analysis
- Program Slicing
- Visualization
- Program metrics

Reverse Engineering – Techniques

Control Flow Analysis

- After determining the structure of a program, control flow analysis (CFA) can be performed on it.
- The two kinds of control flow analysis are:
  - **Intra-procedural Analysis:** It shows the order in which statements are executed within a subprogram.
  - **Inter-procedural Analysis:** It shows the calling relationship among program units.

Reverse Engineering – Techniques

Control Flow Analysis - Control Flow Graph (CFG)

Intra-procedural Analysis:

- The idea of basic blocks is central to constructing a CFG.
- A basic block is a maximal sequence of program statements such that execution enters at the top of the block and leaves only at the bottom via a conditional or an unconditional branch statement.
- A basic block is represented with one node in the CFG, and an arc indicates possible flow of control from one node to another.
- A CFG can directly be constructed from an AST by walking the tree to determine basic blocks and then connecting the blocks with control flow arcs.

Reverse Engineering – Techniques

Control Flow Analysis - Control Flow Graph (CFG)

❖ Inter-procedural Analysis:

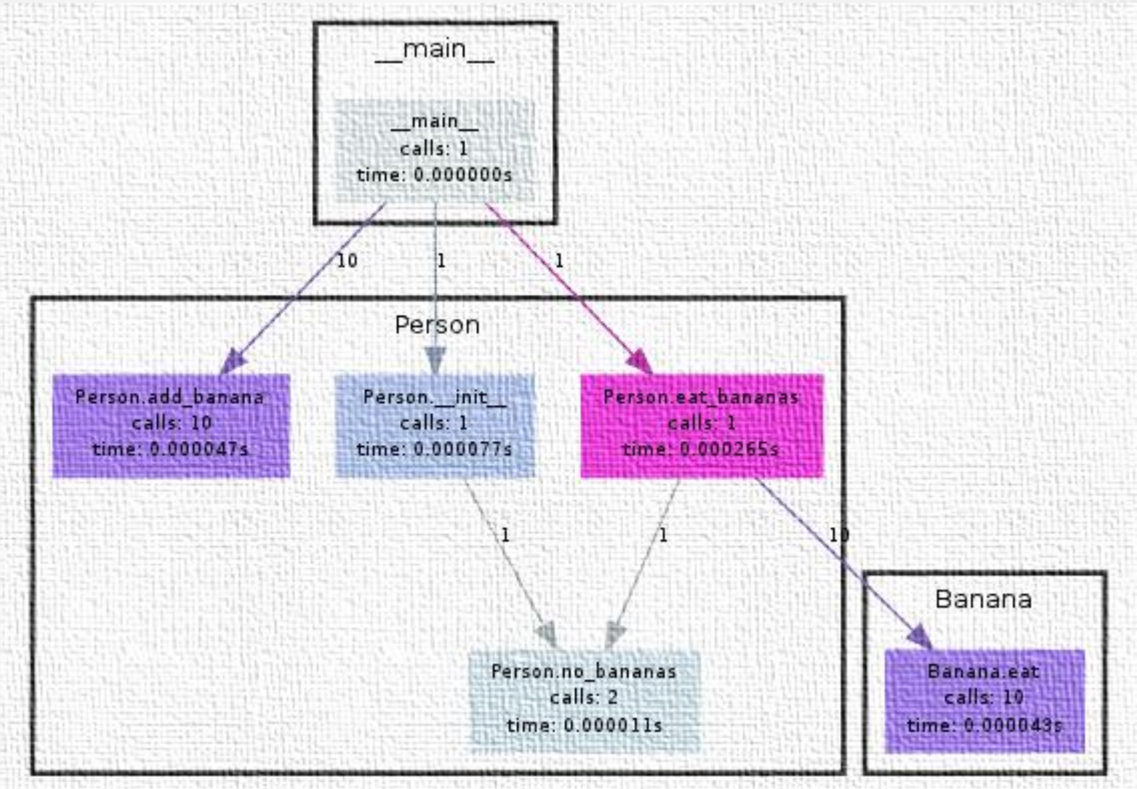
- Inter-procedural analysis is performed by constructing a call graph.
- Calling relationships between subroutines in a program are represented as a call graph which is basically a directed graph.
- Specifically, a procedure in the source code is represented by a node in the graph, and the edge from node ***f*** to **g** indicates that procedure ***f*** calls procedure **g**.

Reverse Engineering – Techniques

Control Flow Analysis - Control Flow Graph (CFG)

❖ Inter-procedural Analysis:

⌗ Example of a Call Graph



Reverse Engineering – Techniques

Control Flow Analysis - Control Flow Graph (CFG)

- A Control Flow Graph (CFG) is the graphical representation of control flow or computation during the execution of programs or applications.
- Control flow graphs are mostly used in static analysis as well as compiler applications, as they can accurately represent the flow inside a program unit.

Reverse Engineering – Techniques

Control Flow Analysis - Control Flow Graph (CFG)

Characteristics of Control Flow Graph

- The control flow graph is process-oriented.
- The control flow graph shows all the paths that can be traversed during a program execution.
- A control flow graph is a directed graph.
- Edges in CFG portray control flow paths and the nodes in CFG portray basic blocks.

Reverse Engineering – Techniques

Control Flow Analysis - Control Flow Graph (CFG)

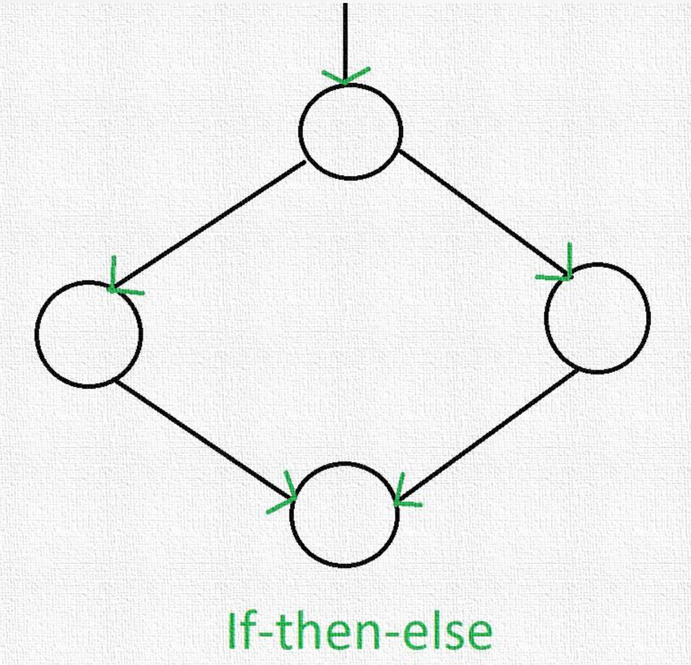
- There exist 2 designated blocks in the Control Flow Graph:
  - **Entry Block:** The entry block allows the control to enter into the control flow graph.
  - **Exit Block:** Control flow leaves through the exit block.
- Hence, the control flow graph comprises all the building blocks
  - Such as the start node, end node and flows between the nodes

Reverse Engineering – Techniques

Control Flow Analysis - Control Flow Graph (CFG)

- General Control Flow Graphs

If-else

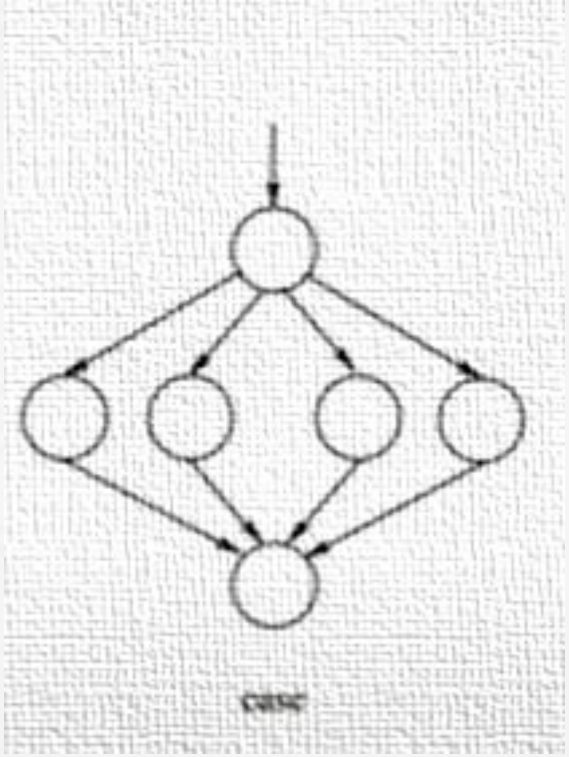


Reverse Engineering – Techniques

Control Flow Analysis - Control Flow Graph (CFG)

- General Control Flow Graphs

- **Case (Switch)**

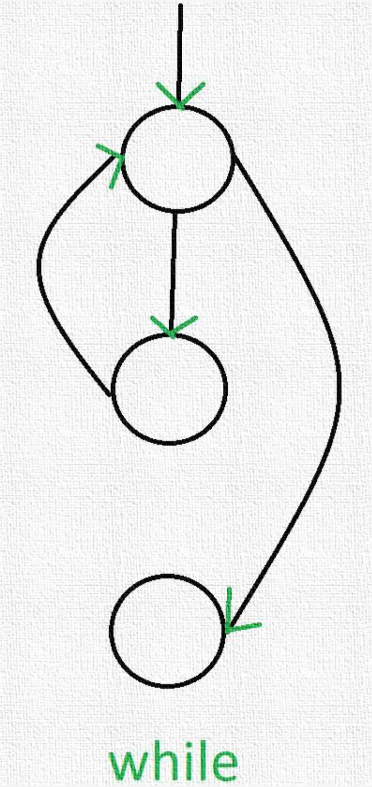


Reverse Engineering – Techniques

Control Flow Analysis - Control Flow Graph (CFG)

- General Control Flow Graphs

- **While**



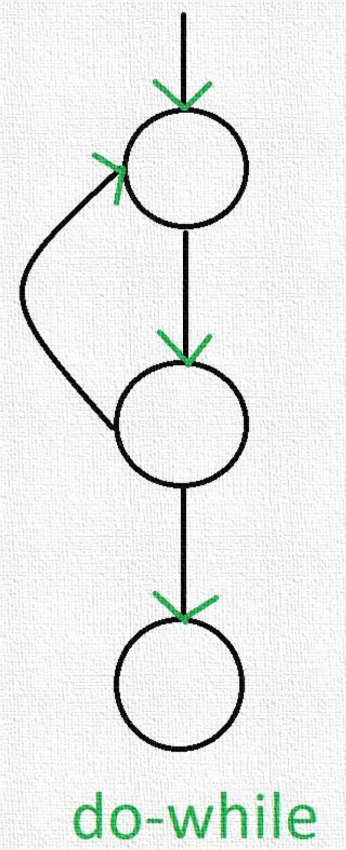
while

Reverse Engineering – Techniques

Control Flow Analysis - Control Flow Graph (CFG)

- General Control Flow Graphs

- **Do-while**

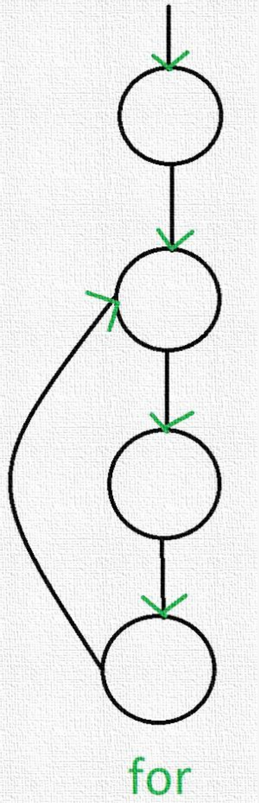


Reverse Engineering – Techniques

Control Flow Analysis - Control Flow Graph (CFG)

- General Control Flow Graphs

For



Reverse Engineering – Techniques

Control Flow Analysis - Control Flow Graph (CFG)

- Example

```
if A = 10 then
if B > C
A = B
else A = C
endif
Endif
print A, B, C
```

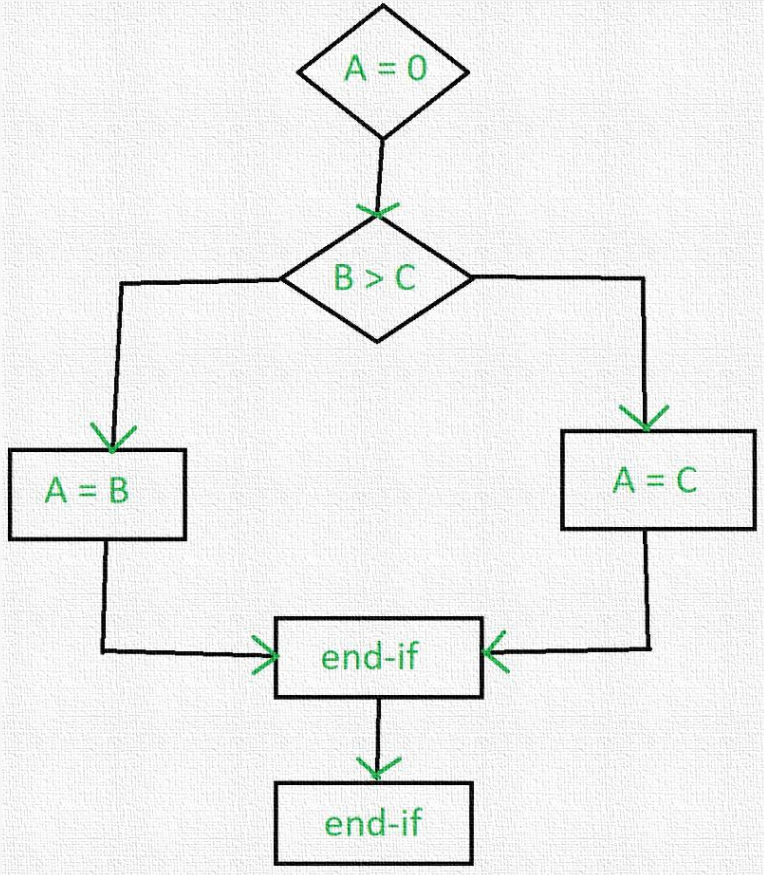
Reverse Engineering – Techniques

Control Flow Analysis - Control Flow Graph (CFG)

- Example

```
if A = 10 then
if B > C
A = B
else A = C
endif
Endif
print A, B, C
```

Flow Chart

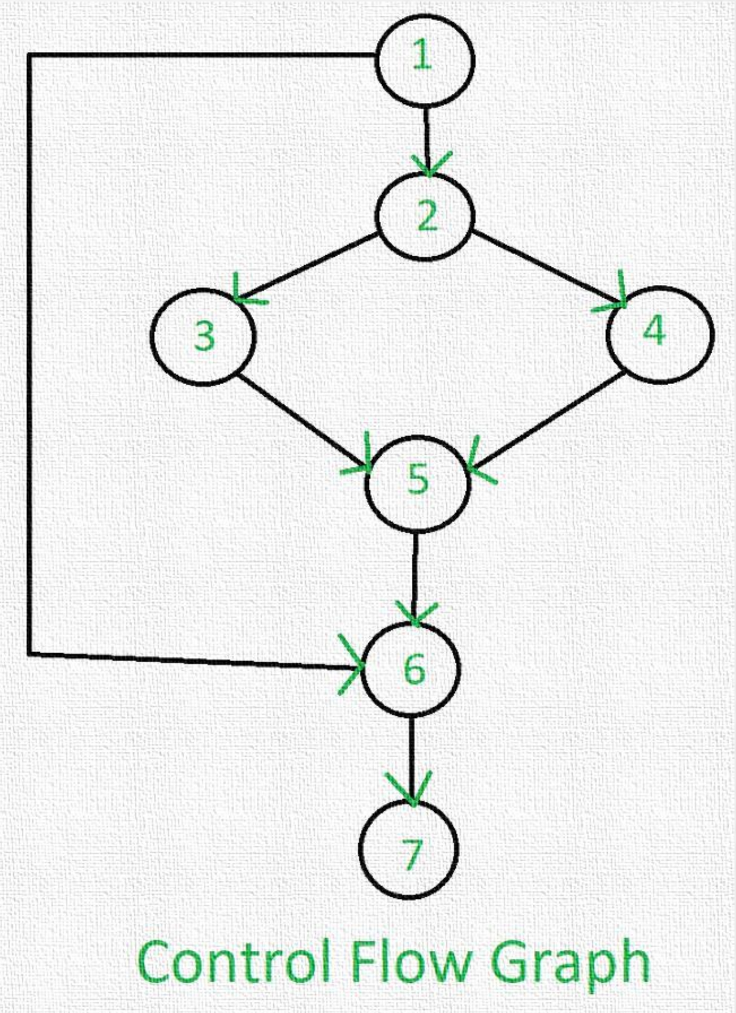


Reverse Engineering – Techniques

Control Flow Analysis - Control Flow Graph (CFG)

- Example

```
if A = 10 then
if B > C
A = B
else A = C
endif
Endif
print A, B, C
```

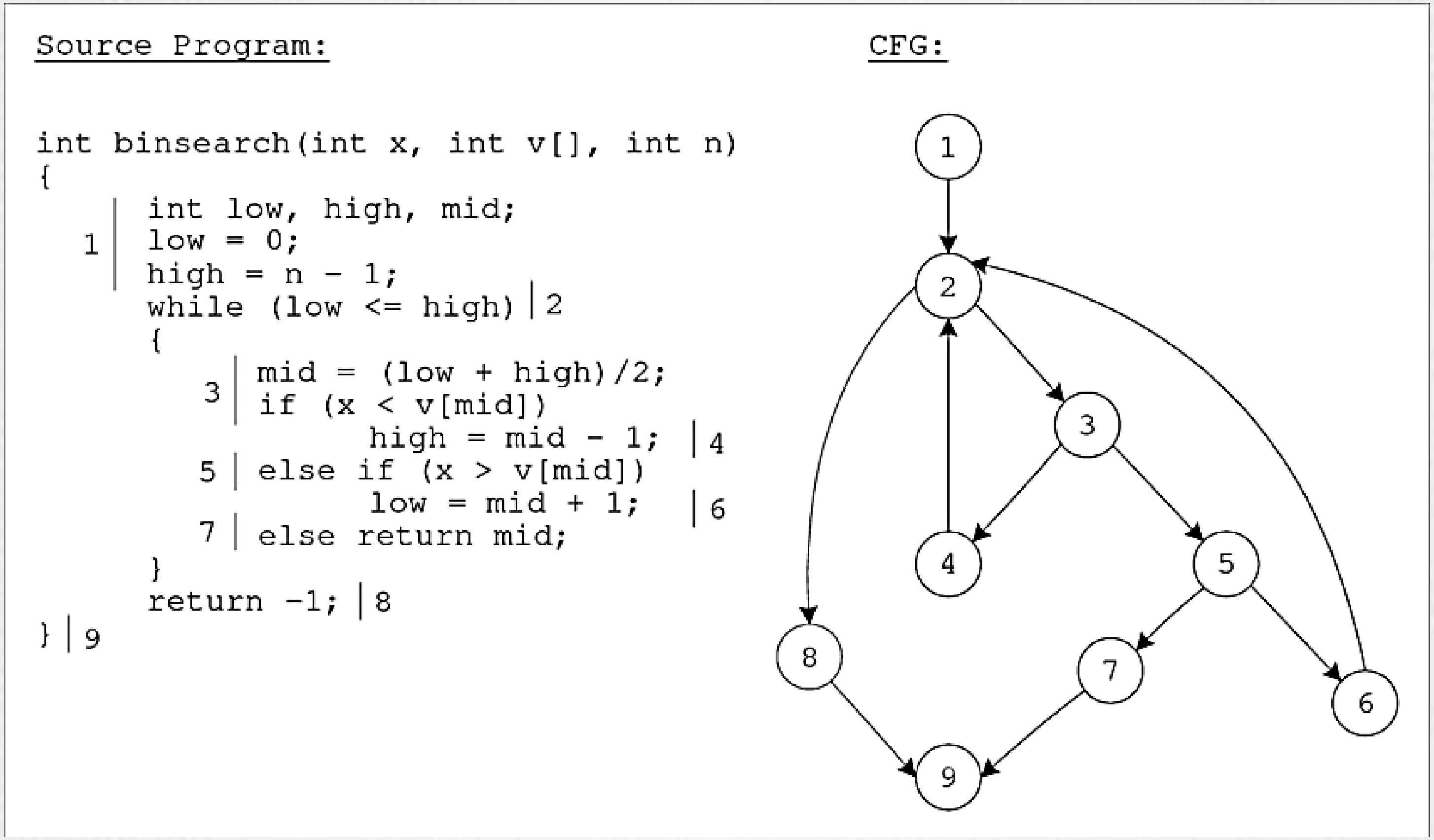


Control Flow Graph

Reverse Engineering – Techniques

Control Flow Analysis - Control Flow Graph (CFG)

- Example



Source Program:

```
int binsearch(int x, int v[], int n)
{
      | int low, high, mid;
    1 | low = 0;
      | high = n - 1;
        while (low <= high) | 2
        {
              | mid = (low + high)/2;
            3 | if (x < v[mid])
                       high = mid - 1;  | 4
            5 | else if (x > v[mid])
                       low = mid + 1;   | 6
            7 | else return mid;
        }
        return -1; | 8
} | 9
```

CFG:

Thank You!