



Software Re-Engineering

Lecture: 13

Sequence [Today's Agenda]

Content of Lecture

- Control Flow Analysis

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.

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.

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 .

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.

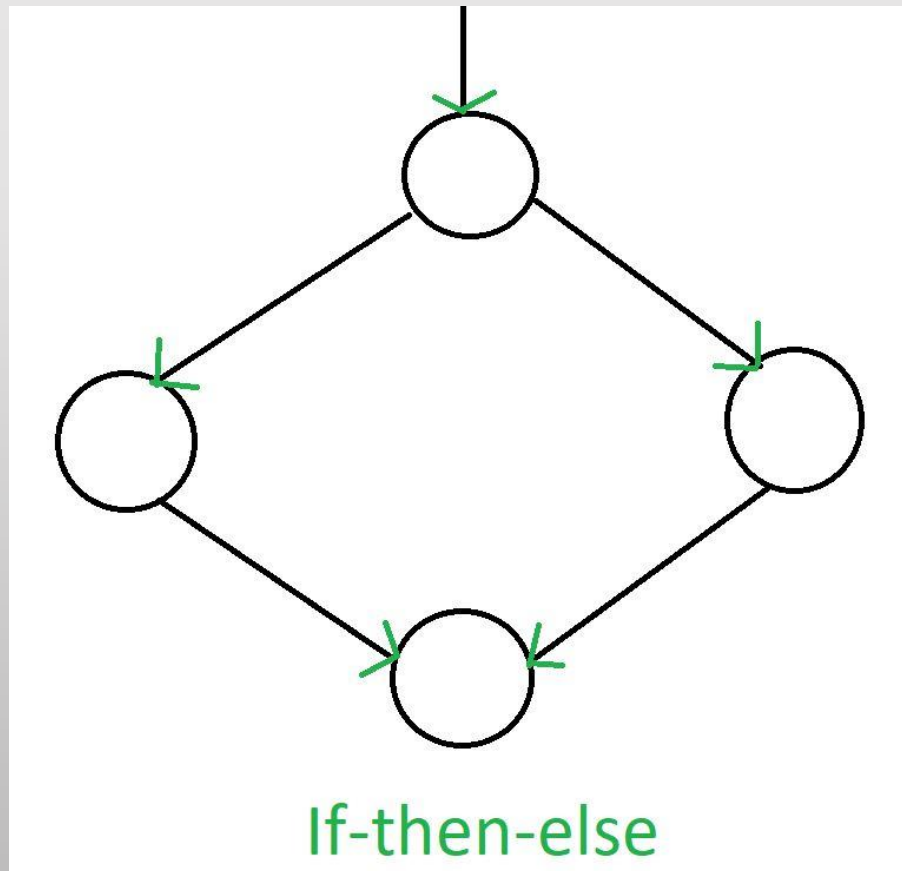
Control Flow Analysis- Control Flow Graph (CFG)

- 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.

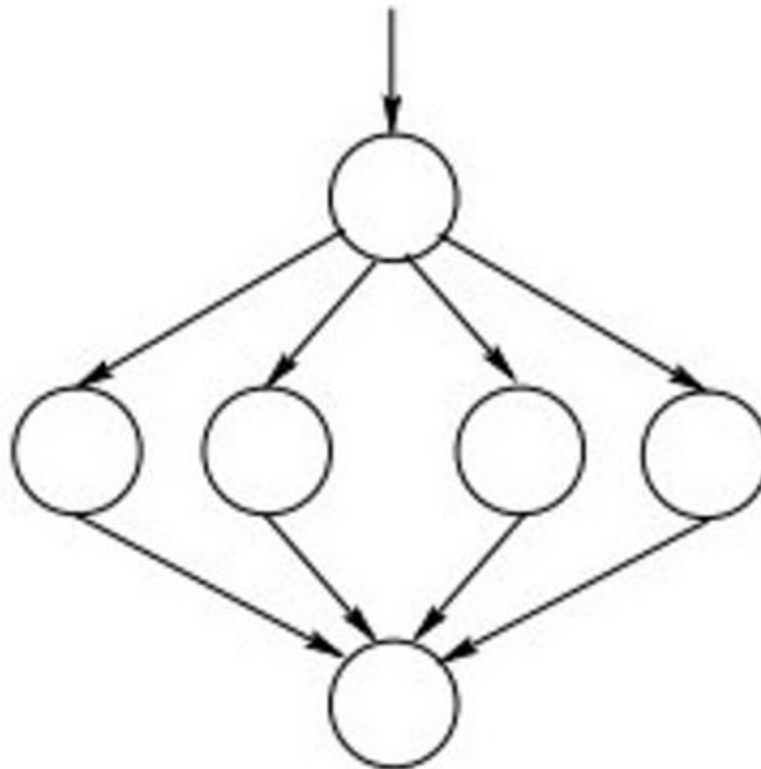
-
- There exist 2 designated blocks in the Control Flow Graph:
 - Entry Block:
 - The entry block allows the control to enter into the control flow graph.
 - Exit Block:
 - Control flow leaves through the exit block.
 - Hence, the control flow graph comprises all the building blocks such as the start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

General Control Flow Graph

- Control Flow Graph is represented differently for all statements and loops.
- If-Else

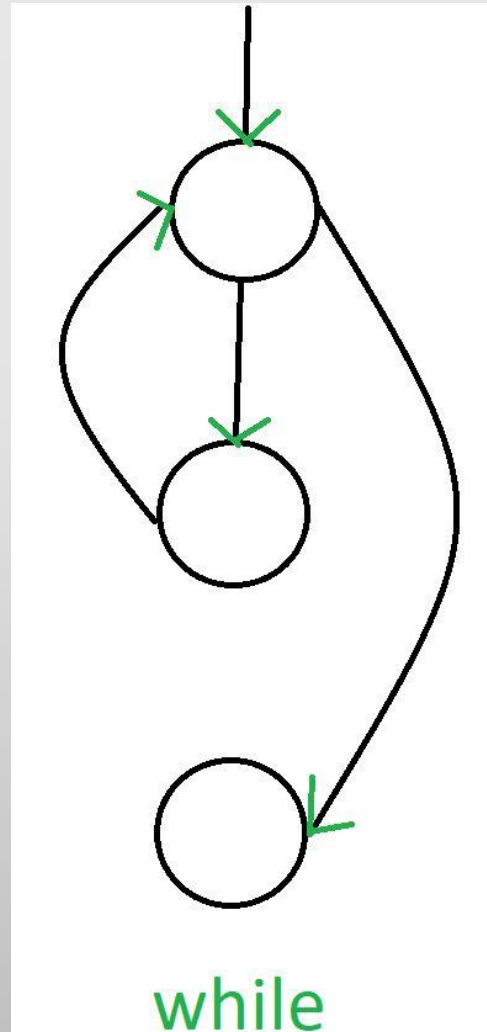


-
- Case (Switch)

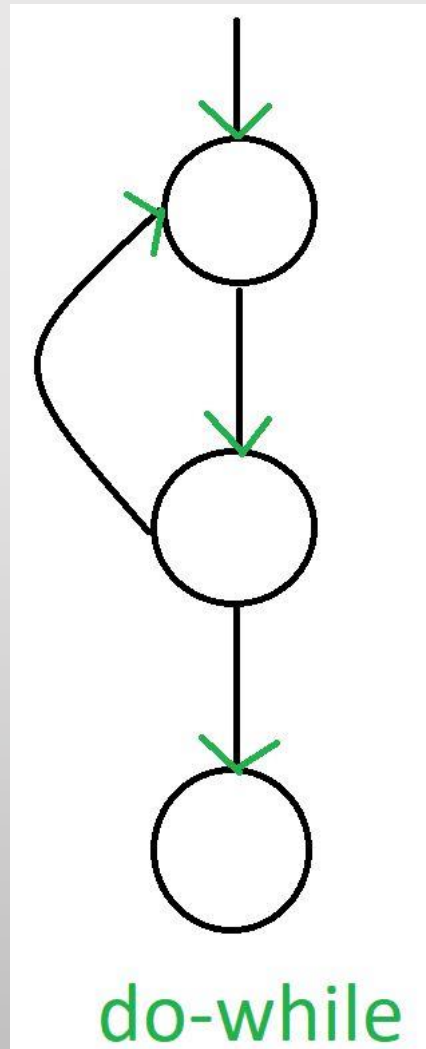


case

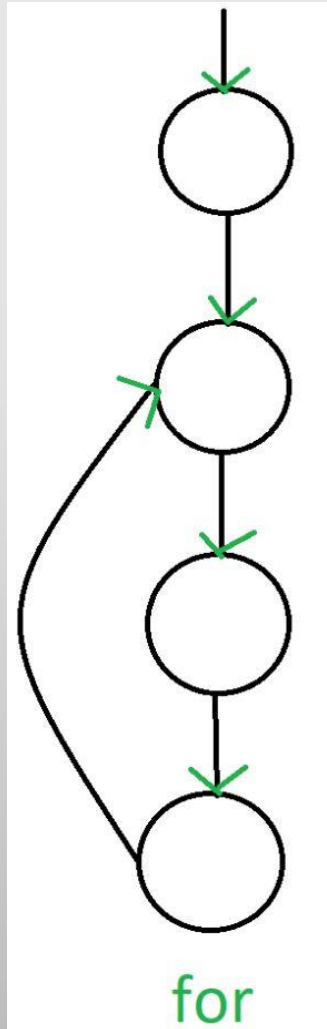
-
- While



-
- Do-While



-
- For



Example

if A = 10 then

if B > C

A = B

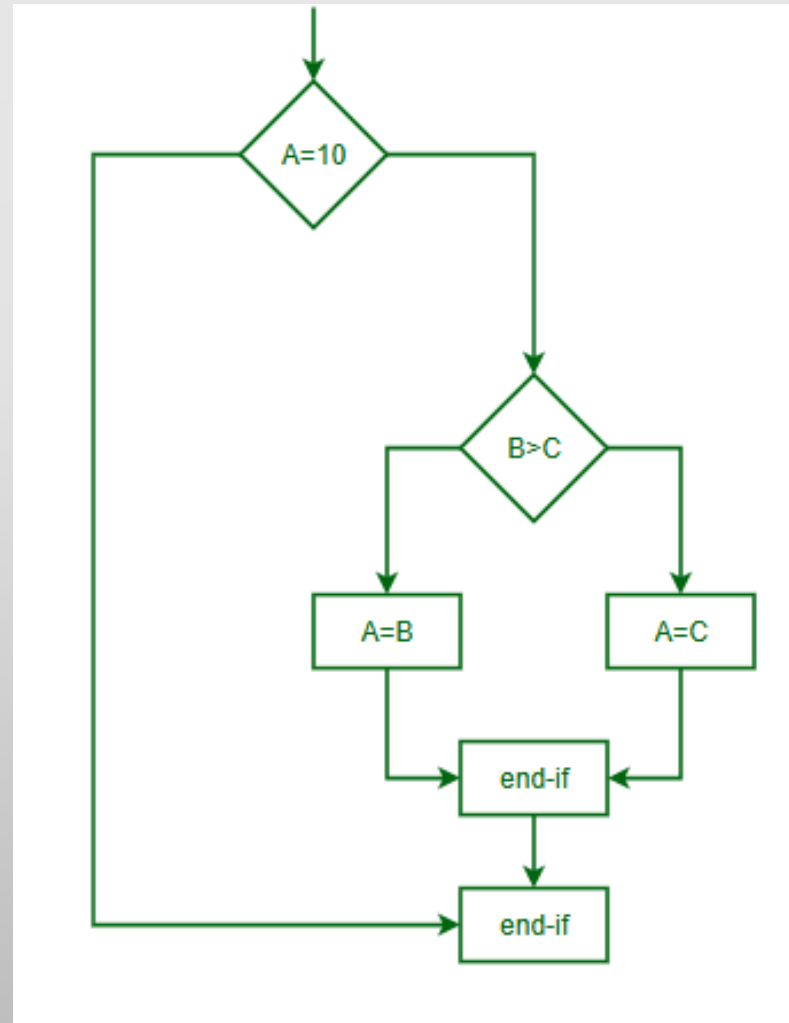
else A = C

endif

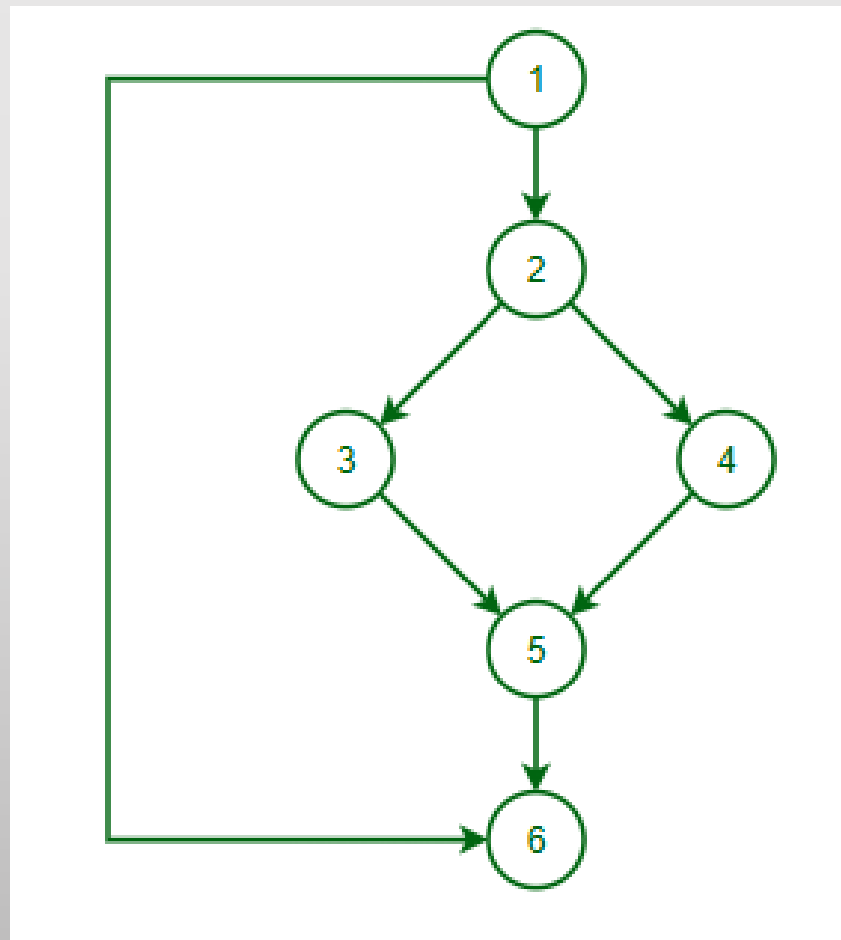
endif

print A, B, C

Flow Chart:



Control Flow Graph:



-
- Advantage of CFG:
 - **Visualizes program flow:** Easy to see how a program runs.
 - **Helps find errors:** Detects unreachable code or infinite loops.
 - **Useful for optimization:** Improves program performance.
 - **Aids testing:** Ensures all parts of the code are tested.

-
- Disadvantages of CFG:
 - Complex for big programs: Hard to understand.
 - No unpredictable behavior: Can't show unclear paths.
 - No data info: Only shows program flow.
 - Not scalable: Becomes messy for large projects.

Thank You!

