Cyclomatic Complexity

Cyclomatic Complexity Measures

Complexity is a software metric that given the quantitative measure of logical complexity of the program.

The Cyclomatic complexity defines the number of independent paths in the basis set of the program that provides the upper bound for the number of tests that must be conducted to ensure that all the statements have been executed atleast once.

There are three methods of computing Cyclomatic complexities.

Method 1: Total number of regions in the flow graph is a Cyclomatic complexity.

Method 2: The Cyclomatic complexity, V (G) for a flow graph G can be defined as

$$\mathbf{V(G) = E - N + 2}$$

Where: E is total number of edges in the flow graph.

N is the total number of nodes in the flow graph.

Method 3: The Cyclomatic complexity V (G) for a flow graph G can be defined as

$$\mathbf{V(G) = P + 1}$$

Where: P is the total number of predicate nodes contained in the flow G.

Let us understand computation of Cyclomatic complexity with the help of an example.

Consider following code fragment with line numbered

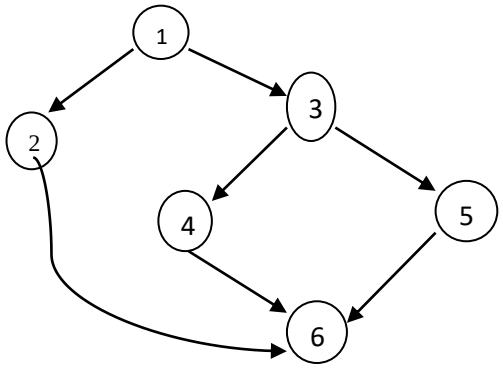
```
{
1. If (a<b)
2. F1 () ;
   else
```

{

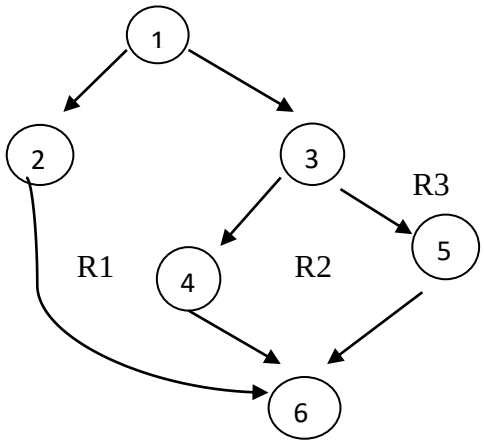
3. If (a<c)
4. F2 () ;
   else
5. F3 () ;
   }
6. }

To compute Cyclomatic complexity we will follow these steps –

Step 1. Design flow graph for given code fragment.



Step 2. Compute region, Predicate (i.e decision nodes) edges and total nodes in the flow graph



- There are 3 regions denoted by R1, R2 and R3.

- Nodes 1 and 3 are predicate nodes because which branch to be followed is decided at these points.
- Total edges = 7

  Total nodes = 6

Step 3. Apply formulas in order to compute Cyclomatic complexity.

1) Cyclomatic complexity **V(G)** = Total number of region = 3
2) Cyclomatic complexity **V(G)** = E - N + 2

   Cyclomatic complexity **V (G)** = 7- 6 +2 = 3

3) Cyclomatic complexity **V(G)** = P +1

   V (G) = 2 + 1 = 3

   Where P is predicate nodes (node 1 and node 2) are predicate nodes because from these nodes only the decision of which path is to be followed is taken.

   Thus Cyclomatic complexity is **3** for given code.