

# Estimation of Test Coverage Metrics and Structural Complexity



## Finding linearly independent paths from a given program

Consider the following simple C program.

```
01 // Initialize elements of a 2D array to 0
02
03 #include <stdio.h>
04
05 int
06 main(int argc, char **argv)
07 {
08     int a[5][10];
09     int i;
10     int j;
11     int nr;
12     int nc;
13
14     nr = 5;
15     nc = 10;
16
17     for (i = 0; i <= nr; i++) {
18         for (j = 0; j <= nc; j++) {
19             a[i][j] = 0;
20         }
21     }
22
23     return 0;
24 }
```

**Tasks:**

1. Compile the above program to generate it's CFG. Verify whetther the CFG corresponds to the basic blocks
2. Identify the linearly independent paths from the CFG. The paths would be indicated by the basic block numbers (instead of line numbers of the actual program)

**Learning Objectives:**

- Identify the linearly independent paths from a CFG

**Limitations:** The current workspace can generate CFGs only for the main function. In other words, this would not work with user-defined functions. However, in real life a program would contain several modules. All such modules have to be taken into account while determining the complexity.

Submit

### Write the C program below

search go to line fullscreen undo redo 10 pt

enable/disable some display features (smarter display but more CPU charge) toggle syntax highlight on/off

reset highlight (if desyncronized from text) toggle word wrapping mode about

```
37     }
38
39
40
41     return 0;
42
43 }
44
```

Position: Ln 44, Ch 1 Total: Ln 44, Ch 257 resize

Compile program (gcc -c -fdump-tree-vcg -fdump-tree-cfg test.c)

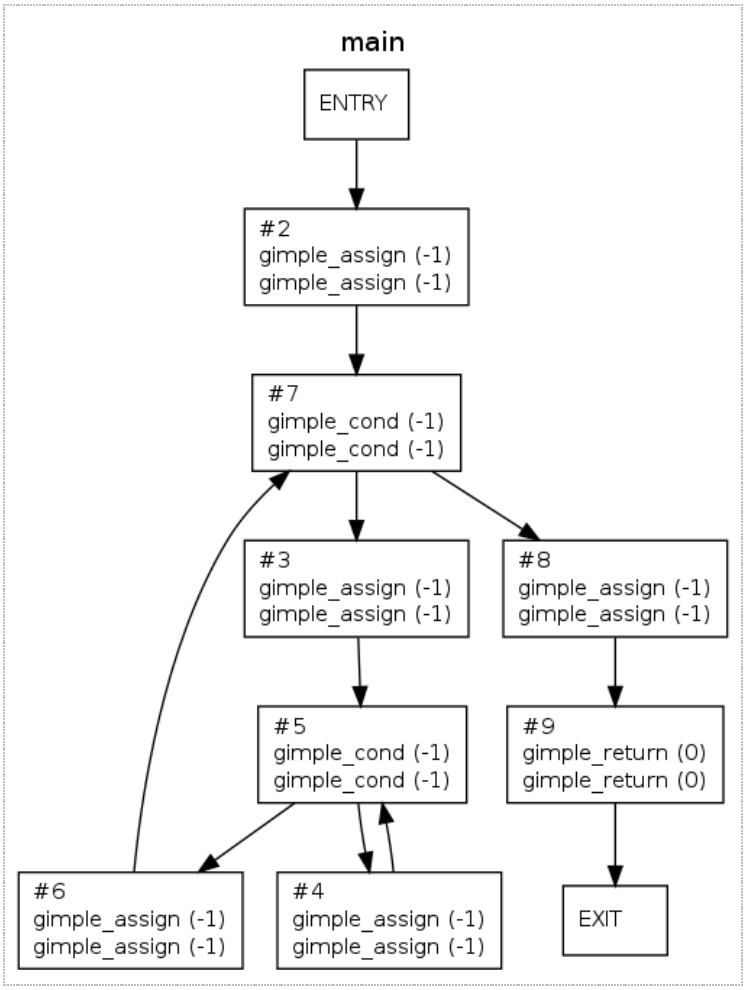
**Compilation error (if any)**

**Program representation in terms of Basic Blocks**

```
;; Function main (main)

Merging blocks 8 and 9
main (int argc, char * * argv)
{
  int nc;
  int nr;
  int j;
  int i;
  int a[5][10];
  int D.2061;
```

**Control Flow Graph of the program**




## Result