

CS434 Software Engineering and Testing

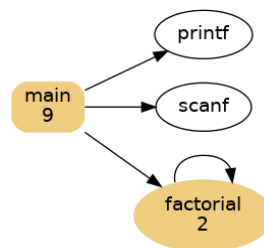
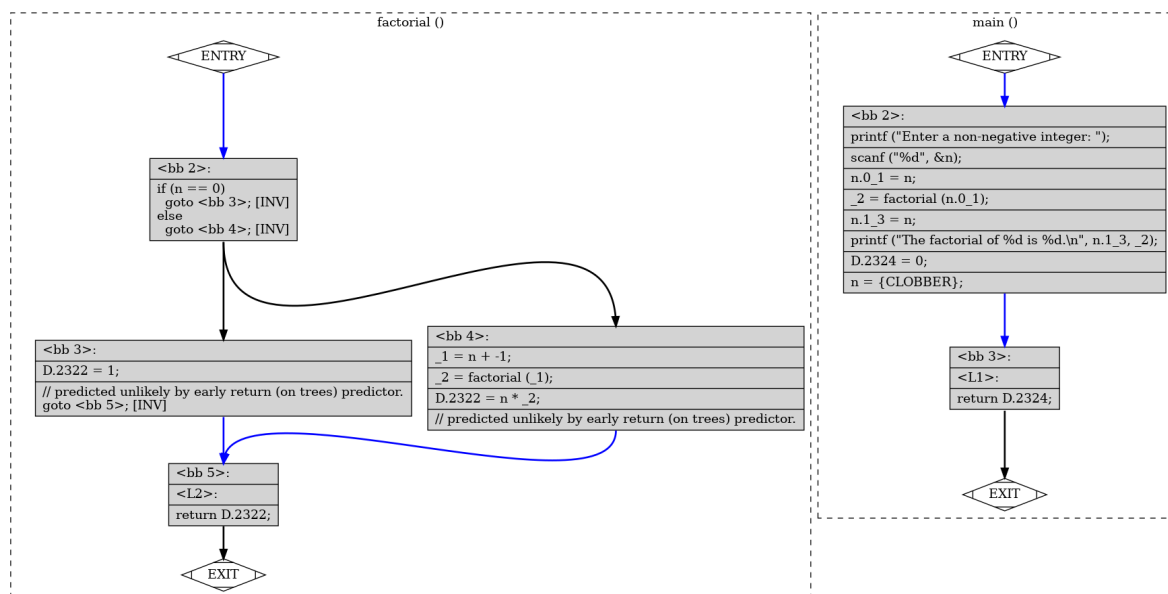
Tutorial #2 (15/01/2024)

Q1. Draw the control flow graph (CFG) of the following code fragments. Is the CFG interprocedural or intraprocedural?

C	Python
<pre>#include<stdio.h> int main() { int y=1; if(y<0) if (y>0) y=3; else y=0; printf("%d\n",y); }</pre>	<pre>if __name__ == "__main__": y=1; if(y<0): if (y>0): y=3; else: y=0; print(y);</pre>
<pre>graph TD ENTRY([ENTRY]) --> BB2["<bb 2>: y = 1; if (y < 0) goto <bb 3>; [INV] else goto <bb 6>; [INV]"] BB2 --> BB3["<bb 3>: if (y > 0) goto <bb 4>; [INV] else goto <bb 5>; [INV]"] BB3 --> BB4["<bb 4>: y = 3; goto <bb 6>; [INV]"] BB3 --> BB5["<bb 5>: y = 0;"] BB4 --> BB6["<bb 6>: printf(\"%d\\n\", y); D.2322 = 0;"] BB5 --> BB6 BB6 --> BB7["<bb 7>: <L5>: return D.2322;"] BB7 --> EXIT([EXIT])</pre>	<pre>graph TD Entry{if __name__ == '__main__':} --> Y1[y = 1] Y1 --> IfY0{if y < 0:} IfY0 --> IfYgt0{if y > 0:} IfYgt0 -- "y > 0" --> Y3[y = 3] IfYgt0 -- "y <= 0" --> Y0[y = 0] Y3 --> PrintY[print(y)] Y0 --> PrintY PrintY -- calls --> Print[print]</pre>

Q2. Draw the intra-procedural control flow graph (CFG), and call graph for the following C program. What will the intra-procedural graph look like?

```
#include <stdio.h>
int factorial(int n)
{
    if (n == 0)
        return 1;
    else
        return n * factorial(n - 1);
}
int main()
{
    int n;
    printf("Enter a non-negative integer: ");
    scanf("%d", &n);
    printf("The factorial of %d is %d.\n", n, factorial(n));
    return 0;
}
```

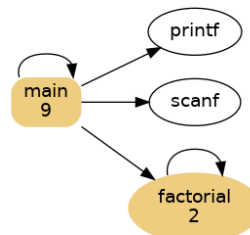
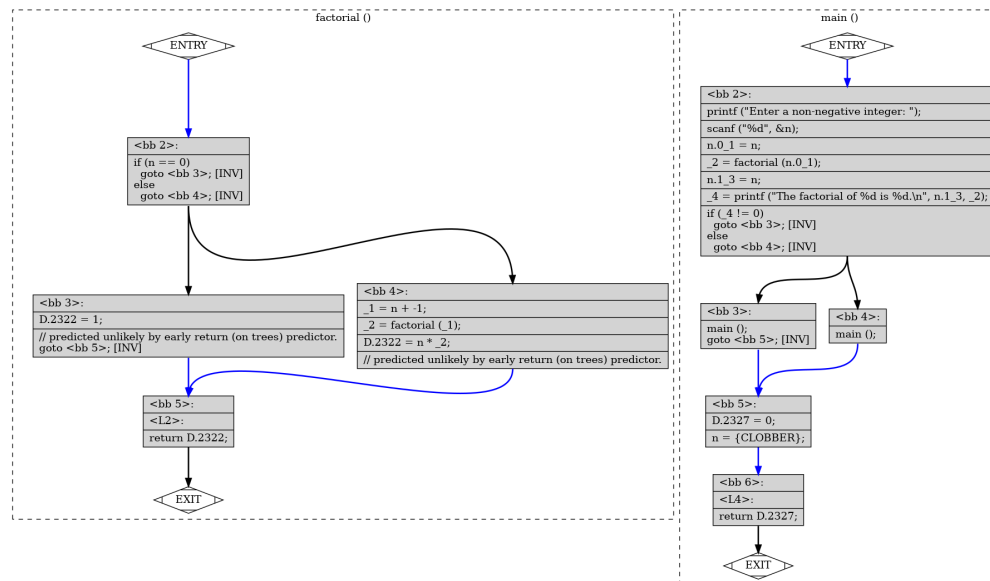


Call graph

q2.c

Q3. Draw the intra-procedural control flow graph (CFG), and call graph for the following C program. What will the inter-procedural graph look like?

```
#include <stdio.h>
int factorial(int n)
{
    if (n == 0)
        return 1;
    else
        return n * factorial(n - 1);
}
int main()
{
    int n;
    printf("Enter a non-negative integer: ");
    scanf("%d", &n);
    if(printf("The factorial of %d is %d.\n", n, factorial(n)))
    {
        main();
    }
    else
        main();
    return 0;
}
```



Call graph

q3.c

Q4. Algebraically derive the cyclomatic complexity of the following C program?

```
#include <stdio.h>
int main()
{
    int n;
    scanf("%d", &n);
    if(n-1)
    {
        printf("%d\n",n-1);
        if(n-2)
        {
            printf("%d\n",n-2);
            if(n-3)
            {
                printf("%d\n",n-3);
                if(n-4)
                {
                    printf("%d\n",n-4);
                    if(n-5)
                    {
                        printf("%d\n",n-5);
                    }
                }
            }
        }
    }
    return(0);
}
```

Algebraic

Domain specific recurrence

$$\begin{aligned} cc(n) &= 1 + cc(n-1) \\ &= 1 + 1 + cc(n-2) \\ &\dots \\ &= 5 + cc(n-5) \\ &= 5 + 1 = 6 \end{aligned}$$

Therefore,
 $cc(\text{main}) = 6$

Sanity check #1

$$\begin{aligned} cc(\text{main}) &= \# \text{regions in CFG} \\ &= 5 + 1 = 6 \end{aligned}$$

Sanity check #2

$$\begin{aligned} cc(\text{main}) &= \pi + 1 \\ &= 5 + 1 = 6 \end{aligned}$$

π is the #decision nodes
(here ifs)

