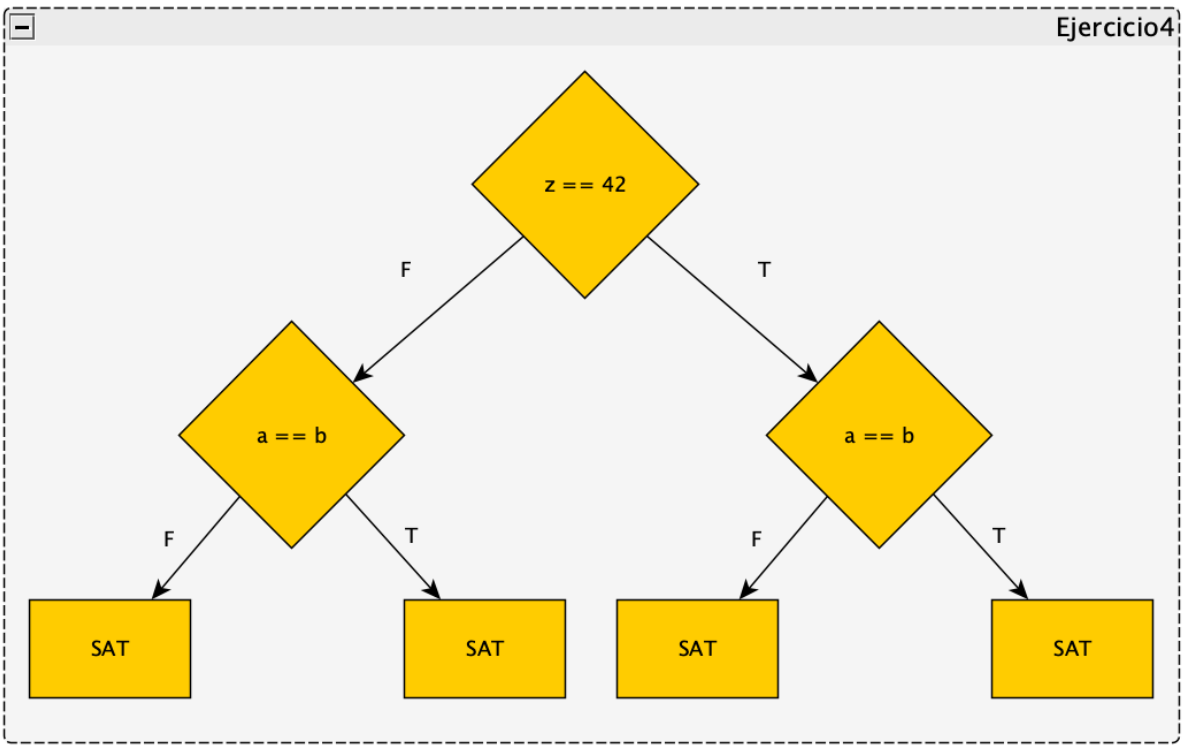
## Ejercicio 4



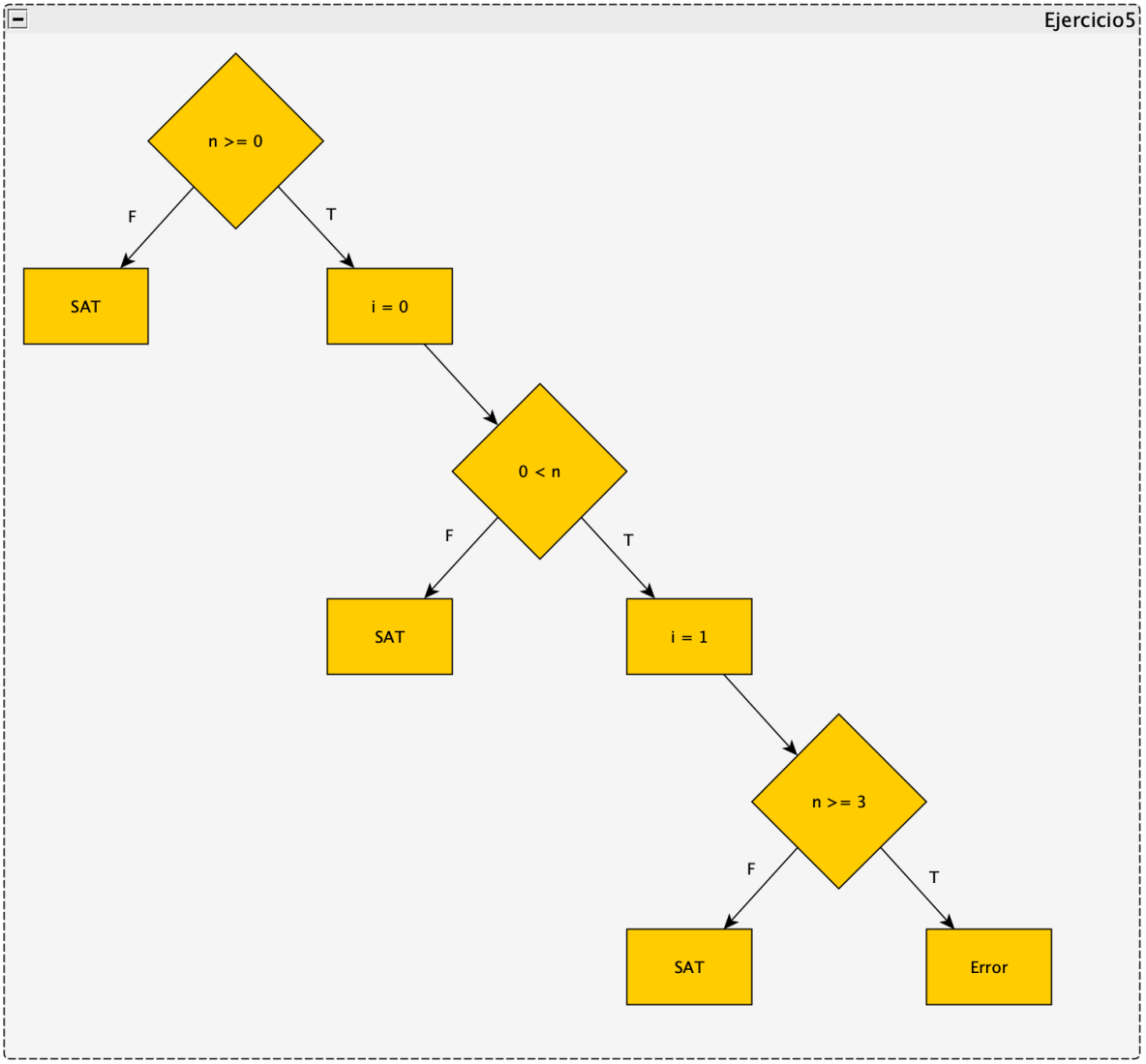

## Ejercicio 5

```
int foo(int n) {
      int acc = 1;
      if( n>= 0) { //C1
        int i = 0;
        if( i < n) { //C2
            acc = acc * (i+1)
            i = i +1;
            if( i < n) { //C3
                  acc = acc * (i+1)
                  i = i +1;
            }
         }
      }
      return acc;
}
```

n=3

```
int foo(int n) {
      int acc = 1;
      if( n>= 0) { //C1
        int i = 0;
        if( i < n) { //C2
            acc = acc * (i+1)
            i = i +1;
            if( i < n) { //C3
                  acc = acc * (i+1)
                  i = i +1;
                  if (n >= 3)
                    return Error
            }
         }
      }
      return acc;
}
```

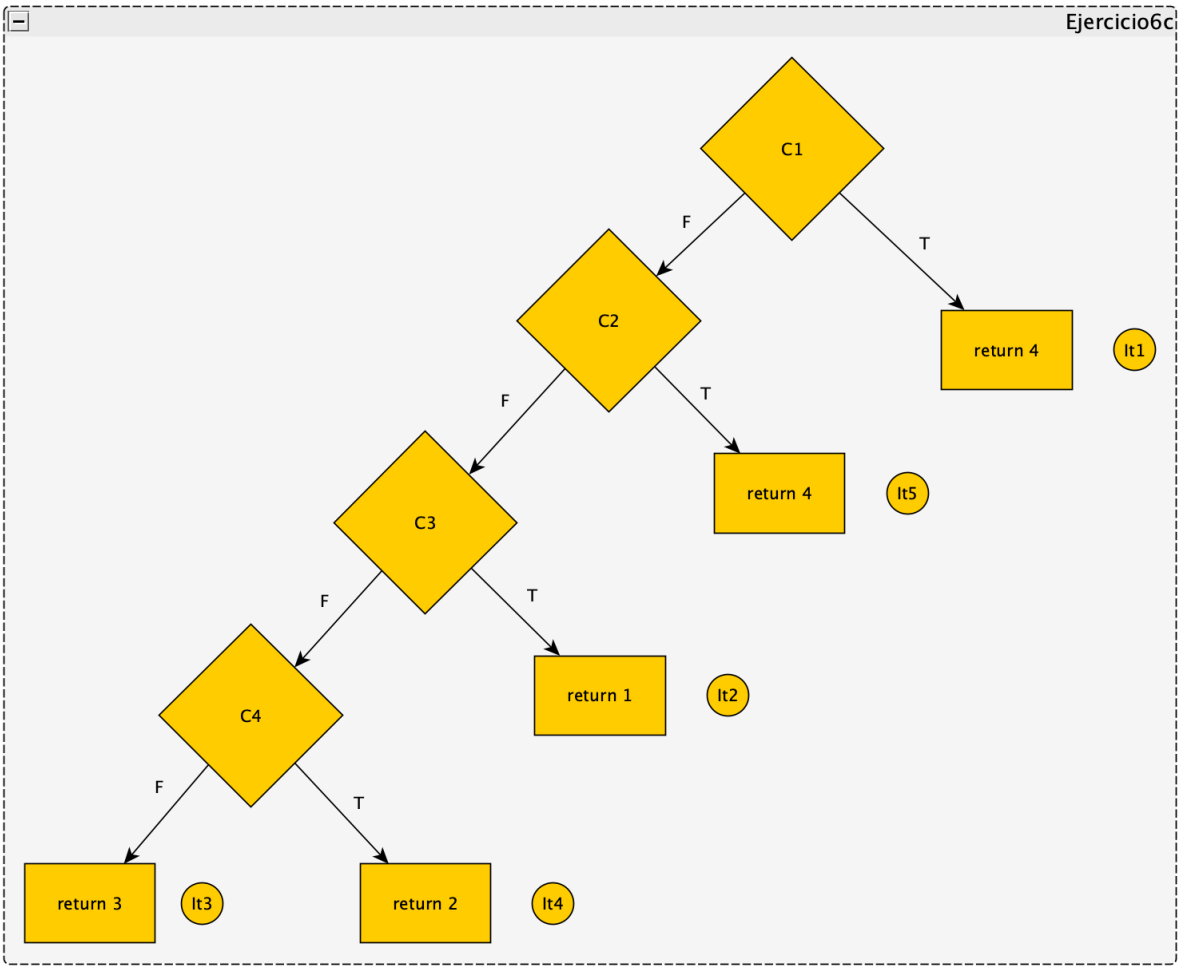
B




# Ejercicio 6

| Iteracion | Input Concreto | Condicion de Ruta | Especificacion para Z3 | Resultado Z3 |
|---|---|---|---|---|
| 1 | a=0, b=0, c=0 | C1 | not C1 | a0 = 1, b0 = 1, c0 = 1 |
| 2 | a = 1, b = 1, c = 1 | not C1 and not C2 and C3 | not C1 and not C2 and not C3 | a0 = 2, b0 = 3, c0 = 4 |
| 3 | a = 2, b = 3, c = 4 | not C1 and not C2 and not C3 and not C4 | not C1 and not C2 and not C3 and C4 | a0 = 2, b0 = 1, c0 = 2 |
| 4 | a =2, b = 1, c = 2 | not C1 and not C2 and not C3 and C4 | not C1 and C2 | a0 = 1, b0 = 1, c0 = 2 |
| 5 | a = 1, b = 1, c = 2 | not C1 and C2 | END | END |
| | | | | |
| Branch coverage: 100% | | | | |

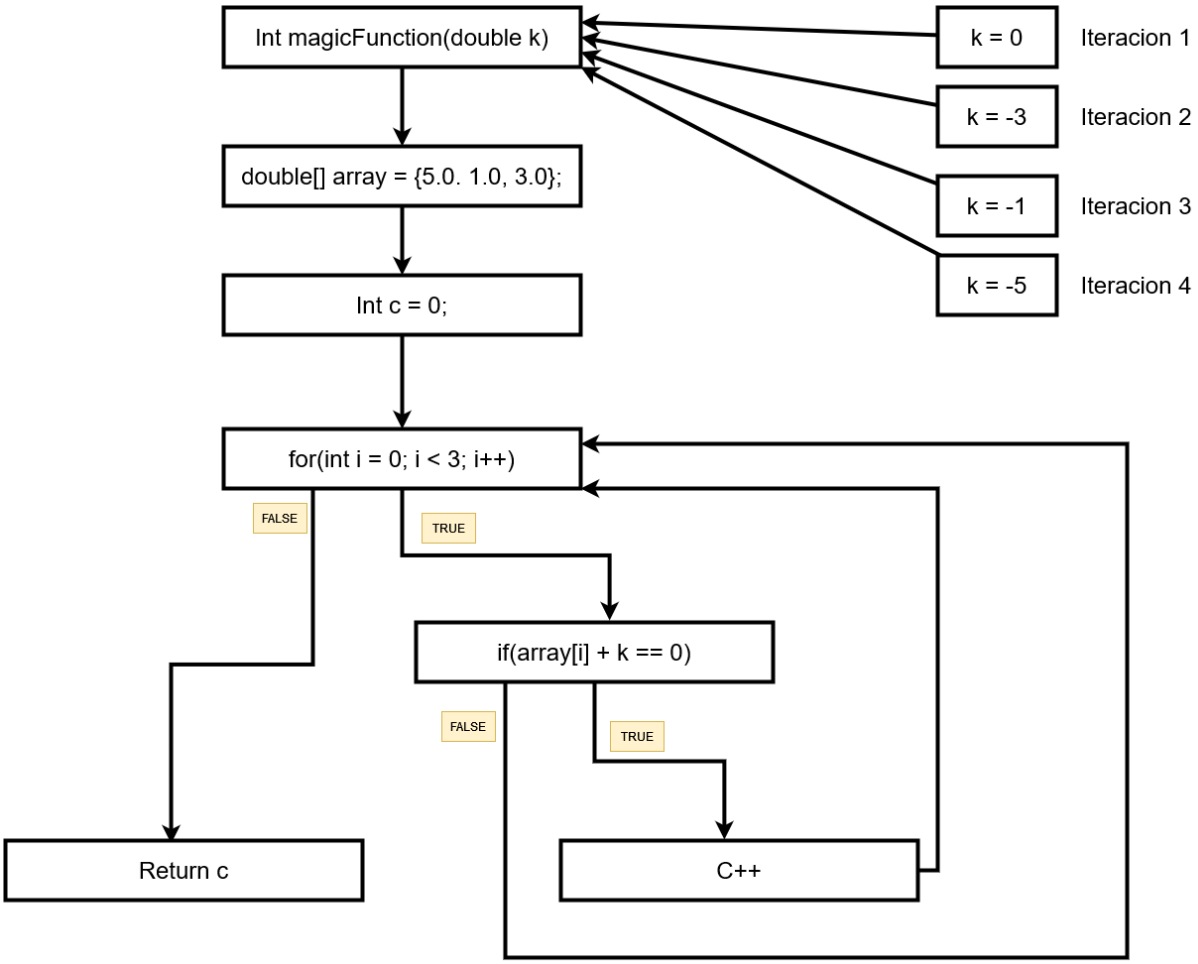
C




# Ejercicio 7

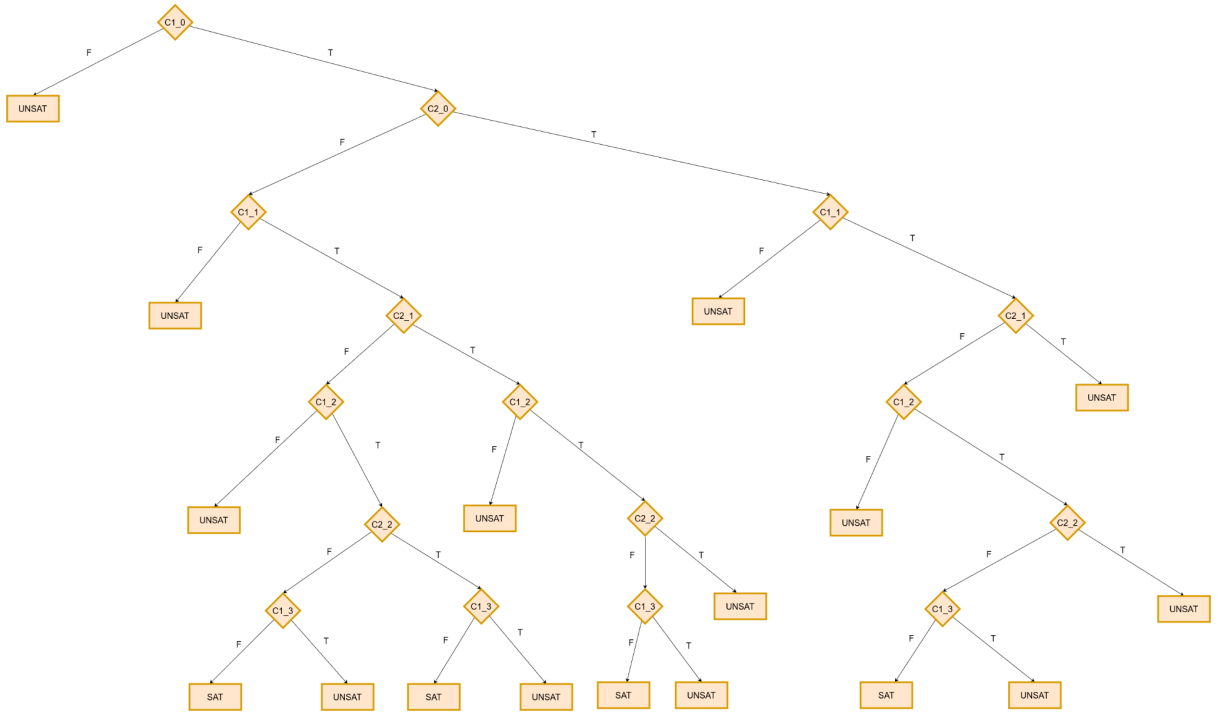
| Iteración | Input concreto | Condición de ruta | Especificación para Z3 | Resultado z3 |
|---|---|---|---|---|
| 1 | k=0 | C1_0 and not C2_0 and C1_1 and not C2_1 and C1_2 and not C2_2 and not C1_3 | C1_0 and not C2_0 and C1_1 and not C2_1 and C1_2 and not C2_2 and C1_ | UNSAT |
| | | | C1_0 and not C2_0 and C1_1 and not C2_1 and C1_2 and C2_2 | k=-3 |
| 2 | k=-3 | C1_0 and not C2_0 and C1_1 and not C2_1 and C1_2 and C2_2 and not C1_3 | C1_0 and not C2_0 and C1_1 and not C2_1 and C1_2 and C2_2 and C1_3 | UNSAT |
| | | | C1_0 and not C2_0 and C1_1 and not C2_1 and not C1_2 | UNSAT |
| | | | C1_0 and not C2_0 and C1_1 and C2_1 | k=-1 |
| 3 | k=-1 | C1_0 and not C2_0 and C1_1 and C2_1 and C1_2 and not C2_2 and not C1_3 | C1_0 and not C2_0 and C1_1 and C2_1 and C1_2 and not C2_2 and C1_3 | UNSAT |
| | | | C1_0 and not C2_0 and C1_1 and C2_1 and C1_2 and C2_2 | UNSAT |
| | | | C1_0 and not C2_0 and C1_1 and C2_1 and not C1_2 | UNSAT |
| | | | C1_0 and not C2_0 and not C1_1 | UNSAT |
| | | | C1_0 and C2_0 | k=-5 |
| 4 | k=-5 | C1_0 and C2_0 and C1_1 and not C2_1 and C1_2 and not C2_2 and not C1_3 | C1_0 and C2_0 and C1_1 and not C2_1 and C1_2 and not C2_2 and C1_3 | UNSAT |
| | | | C1_0 and C2_0 and C1_1 and not C2_1 and C1_2 and C2_2 | UNSAT |
| | | | C1_0 and C2_0 and C1_1 and not C2_1 and not C1_2 | UNSAT |
| | | | C1_0 and C2_0 and C1_1 and C2_1 | UNSAT |
| | | | C1_0 and C2_0 and not C1_1 | UNSAT |
| | | | not C1 | UNSAT |
| | | | | |

B




El test suite compuesto por los inputs generados unicamente con ejecución simbólica dinámica, vamos a tener 100% de coverage branch

C


C1_0
F
T
UNSAT
C2_0
F
T
C1_1
C1_1
F
T
F
T
UNSAT
C2_1
UNSAT
C2_1
F
T
F
T
C1_2
C1_2
C1_2
UNSAT
F
T
F
T
F
T
UNSAT
C2_2
UNSAT
C2_2
UNSAT
C2_2
F
T
F
T
F
T
C1_3
C1_3
C1_3
UNSAT
F
T
F
T
F
T
SAT
UNSAT
SAT
UNSAT
SAT
UNSAT
SAT
UNSAT