

Symbolically execute the following program, and determine whether the exception at line 13 can or cannot be thrown.

Motivate your answer and show the path conditions you generated during the analysis also stating for each of them whether it is satisfiable or not.

If the exception can be thrown, provide concrete inputs that lead the system to reach it , by solving the execution condition computed with the symbolic execution.

```

1: public int foo(int x, int y) {
2:     int z = x - y + 1;
3:     if (z > 0) {
4:         z = x;
5:     } else if (z < 0){
6:         z = y;
7:     }
8:     else {
9:         z = 0;
10:    }
11:    z = 2 * z;
12:    if (z < 0) {
13:        throw new Exception("z is negative!");
14:    }
15:    return z;
16: }

```

STATEMENT	PATH CONDITIONS	SYMBOLIC VALUES
public int foo(int x, int y) {		$z = x - y + 1$
int z = x - y + 1;		$z = x - y + 1$
if (z > 0) {	$z > 0$	$z = x - y + 1$
z = x;	$z > 0$	$z = x - y + 1$
} else if (z < 0){	$z \leq 0 \wedge z < 0$	$z = x - y + 1$
z = y;	$z \leq 0 \wedge z < 0$	$z = x - y + 1$
} else {	$z = 0$	$z = x - y + 1$
z = 0; }	$z = 0$	$z = x - y + 1$
z = 2 * z;		$z = x - y + 1 \wedge (z > 0 \vee (z \leq 0 \wedge z < 0) \vee z = 0)$
if (z < 0) {	$z < 0$	$z = x - y + 1 \wedge (z > 0 \vee (z \leq 0 \wedge z < 0) \vee z = 0)$
throw new Exception("z is negative!");}	$z < 0$	$z = x - y + 1 \wedge (z > 0 \vee (z \leq 0 \wedge z < 0) \vee z = 0)$
return z;}		$z = x - y + 1 \wedge (z > 0 \vee (z \leq 0 \wedge z < 0) \vee z = 0)$

If the value of variable z is positive or equal to zero the exception at line 13 cannot be thrown, this is caused by the path condition that is generated by the lines of code that precede the if statement in line 12. On the other hand if the value of variable z is negative the path condition that enable the if at line 12 will be satisfied so the exception at line 13 can be thrown. It is also needed that the value of y must be negative because if the value of z is negative the program execution will overwrite the value of z with the value of y at line 6 and if the value of y isnt negative the program execution will not throw the exception at line 13.

If we consider as inputs  $x = -3$  and  $y = -1$  we will have the following execution:

```
1: public int foo(int x, int y) {
2:     int z = -3 + 1 + 1 = -1;
3:     if (-1 > 0) { -----NOT SATISFIED
4:         z = x;
5:     } else if (z < 0){ ----- SATISFIED
6:         z = -1;
7:     }
8:     else { -----NOT SATISFIED
9:         z = 0;
10:    }
11:    z = 2 * (-1) = -2;
12:    if (-2 < 0) { ----- SATISFIED
13:        throw new Exception("z is negative!");
14:    }
15:    return z;
16: }
```