



# Weakest Pre-condition

1) Starting with the post condition and statements, fill in the intermediate assertions and weakest precondition:

1) if(  $x > y$  )

$x = x - y;$

else

$y = y - x;$

$\{gcd(x, y) = z\}$

2)  $w = 2 * w;$

$z = v - 2 ;$

$y = -w;$

$x = \min(y, z);$

$\{x < 0\}$

3) if(  $x > y$  )

$z = x;$

else

$z = y;$

$\{z = \max(x, y)\}$

4)  $x = 2 * x;$

$y = x + 2;$

$z = y/2;$

$\{z > x\}$

# Weakest Pre-condition

5) Compute  $wp(A, x < y)$  in as simple form as you can, where  $A$  is the following code fragment

```
if (x > y)
    tmp = x ;
    x = y;
    y = tmp;
```

6) Compute *weakest precondition* for the following loop code

```
while(x > 0)
    x = x-1;
{x == 0}
```

# Program Abstraction

Check if the following code computes the maximum of two no.s for any given set of input

```
{ true }  
if ( x > y ) : r = x;  
else : r = y;
```

Analyze the program using the domain Parity and then Sign.

```
y = 5;  
x = -2 * y;  
if (x > 0){  
    x = x - (y%2 - 1);  
    y = x * (y - 1);  
else  
    y = -1;
```

# Program Verification

```
L1 :   a = b = i = 0;  
L2 :   while (a <= 1000) {  
L3 :       a = b + i;  
L4 :       b = a + 1;  
L5 :       i = i + 1;  
L6 :   }  
L7 :   if (b > 2000) {  
L8 :       error: exit(-1);  
L9 :   }
```

1. Construct a Boolean program corresponding to this program P, using only the predicates ( $a \leq 1000$ ), ( $b \geq 0$ ), ( $i = 0$ ), and ( $b \leq 2000$ )
2. Show that the error location is reachable in the Boolean program you constructed
3. Explain whether the error identified in this Boolean program is a spurious counter-example.
4. If the counterexample is spurious can you suggest refined predicates such that the counterexample disappears?

# Program Verification

Use abstract interpretation on the following program to determine whether the assertion is true. Use the sign domain as your abstract domain, which has only three elements, [+ , - , 0], representing +ve, -ve, and zero respectively. Show the values collected in each location. Also indicate the outcome:

```
L1 :    x = 1;
L2 :    if (y <= 10){
L3 :                y = 10;
                }
L3 :    else{
L5 :                while (x < y){
L6 :                    x = 2 * x;
L7 :                    y = y - 1;
                }
    }
L8 :    x = y + 1;
L9 :    assert (x > 0);
```