```
add(x2, x1) {
    3:  x3 = x2 + x1 + 0 (int)
    2:  x4 = x3
    1:  return x4
}

main() {
    8:  x3 = 5
    7:  x2 = 3
    6:  x1 = "add"(x3, x2)
    5:  if (x1 <s x3) goto 3 else goto 4
    4:  x2 = x2 + 1 (int)
        goto 2
    3:  x2 = (- x2)
    2:  x4 = 0
    1:  return x4
}
```

L'équation produite sera :

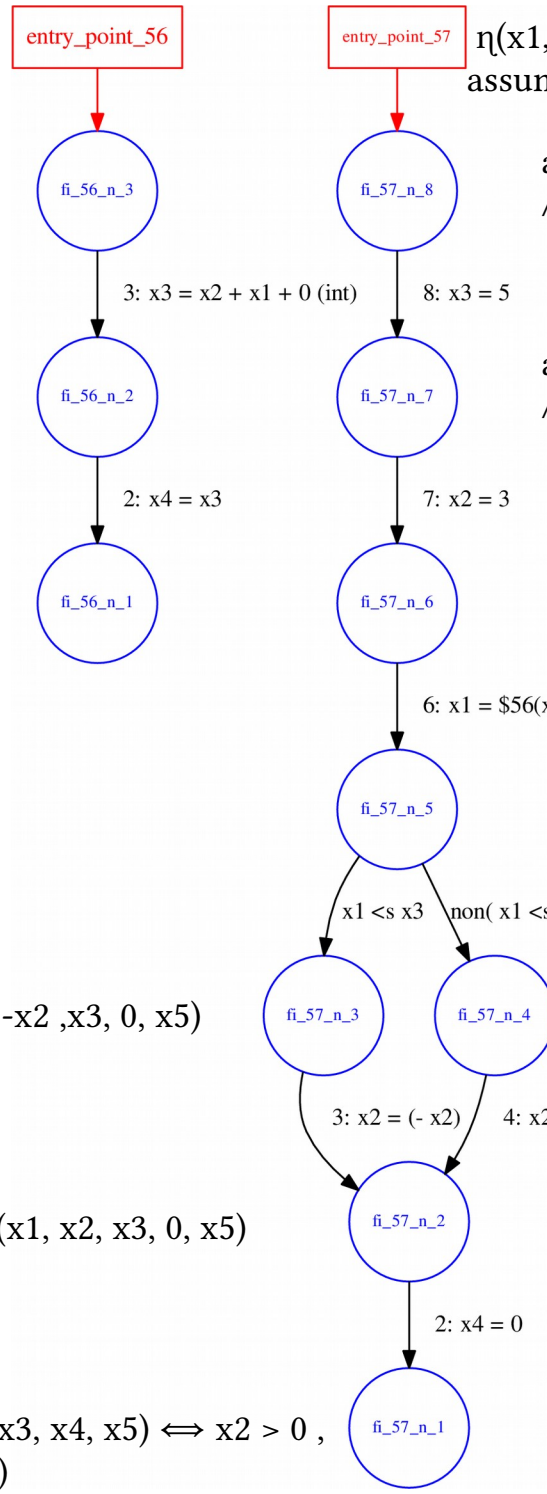
η

⟹

(add(5,3) < 5 ⟹υ(add(5,3) , -3 ,5, 0, x5)

∧

add(5,3) ≥ 5 ⟹ υ(add(5,3) ,3+1 ,5 ,0, x5))



η(x1, x2, x3, x4, x5) ⟺ x1, x2, x3, x4, x5 ∈ ℕ ,
assume(η)

add(5,3) < 5 ⟹υ(add(5,3) , -3 ,5, 0, x5)
∧
add(5,3) ≥ 5 ⟹ υ(add(5,3) ,3+1 ,5 ,0, x5)

add(x3,3) < x3 ⟹υ(add(x3,3) , -3 ,x3, 0, x5)
∧
add(x3,3) ≥ x3 ⟹ υ(add(x3,3) ,3+1 ,x3 ,0, x5)

add(x3,x2) < x3 ⟹υ(add(x3,x2) , -x2 ,x3, 0,x5)
∧
add(x3,x2) ≥ x3 ⟹ υ(add(x3,x2) ,x2+1 ,x3 ,0)

x1 < x3 ⟹υ(x1 , -x2 ,x3, 0, x5)
∧
x1 ≥ x3 ⟹ υ(x1 ,x2+1 ,x3 ,0, x5)

υ(x1 , -x2 ,x3, 0, x5)

υ(x1 ,x2+1 ,x3 ,0, x5)

υ(x1, x2, x3, 0, x5)

υ(x1, x2, x3, x4, x5) ⟺ x2 > 0 ,
Assert (υ)