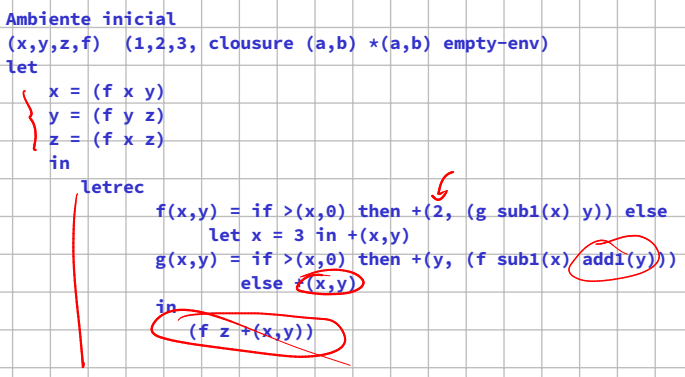
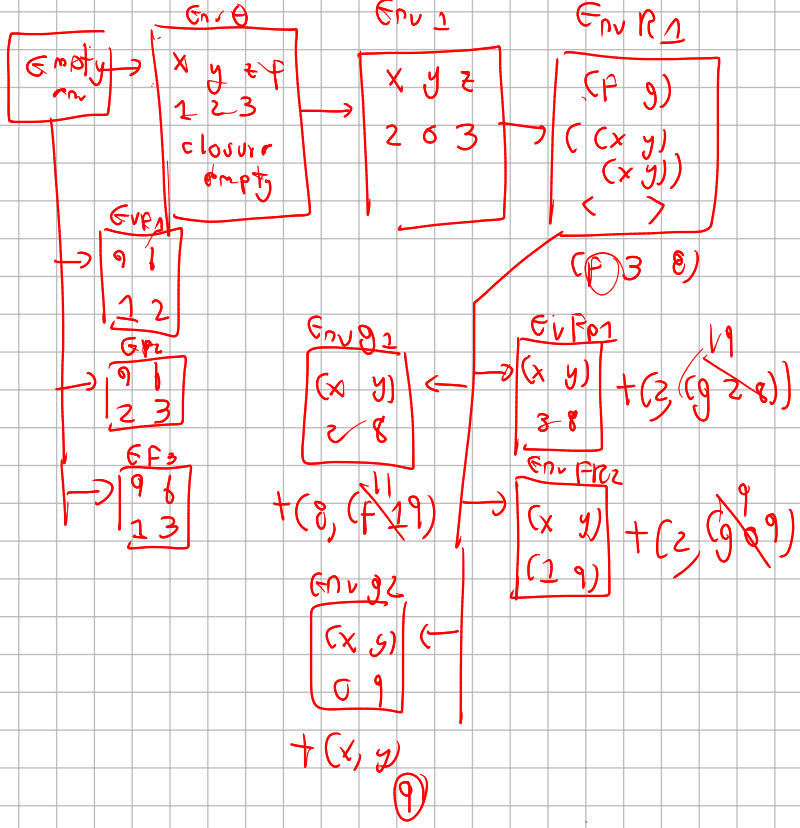
```
Ambiente inicial
(x,y,z,f)  (1,2,3, clousure (a,b) *(a,b) empty-env)
let
    x = (f x y)
    y = (f y z)
    z = (f x z)
    in
       letrec
              f(x,y) = if >(x,0) then +(2, (g sub1(x) y)) else
                    let x = 3 in +(x,y)
              g(x,y) = if >(x,0) then +(y, (f sub1(x) add1(y)))
                       else +(x,y)
              in
                 (f z +(x,y))
```






```
let
   x = ( proc(x,y) +(x,y)
         letrec a(x,y) = if >(x,0) then +(y, (a sub1(x) add1(y)))
         else y in (a 5 3)
         let f = proc(x,y) *(x,y) in
            (f letrec q(x,y) = if >(x,0) then
                     +( let f = +(x,y) in +(f,2),(q sub1(x) y))
                       else y in (q 4 5)
              let f = proc(x,y) *(x,y) in (f 6 5)))
   in x
```

Ambiente inicial vacio

```
let
    x = letrec f(x,y) = if >(x,0) then (f -(x,1) +(y,x)) else y
        in let x = (f 5 3) in +(x,2)
    y = letrec g(x,y) = if >(x,0) then (g -(x,2) +(y,2)) else y
        in let y = (g 6 7) in +(y, (g 3 8))
    in
        let
            k = proc(x,y) +(x,y)
            l = proc(a,b)
                    letrec h(x,y) = if >(x,0) then (h sub1(x) +(y,2))
                                    else y in (h a b)
        in +( (k x y), let p = (l x y) in
                        (proc (x,y) +(x,y,2) x y))
```

92