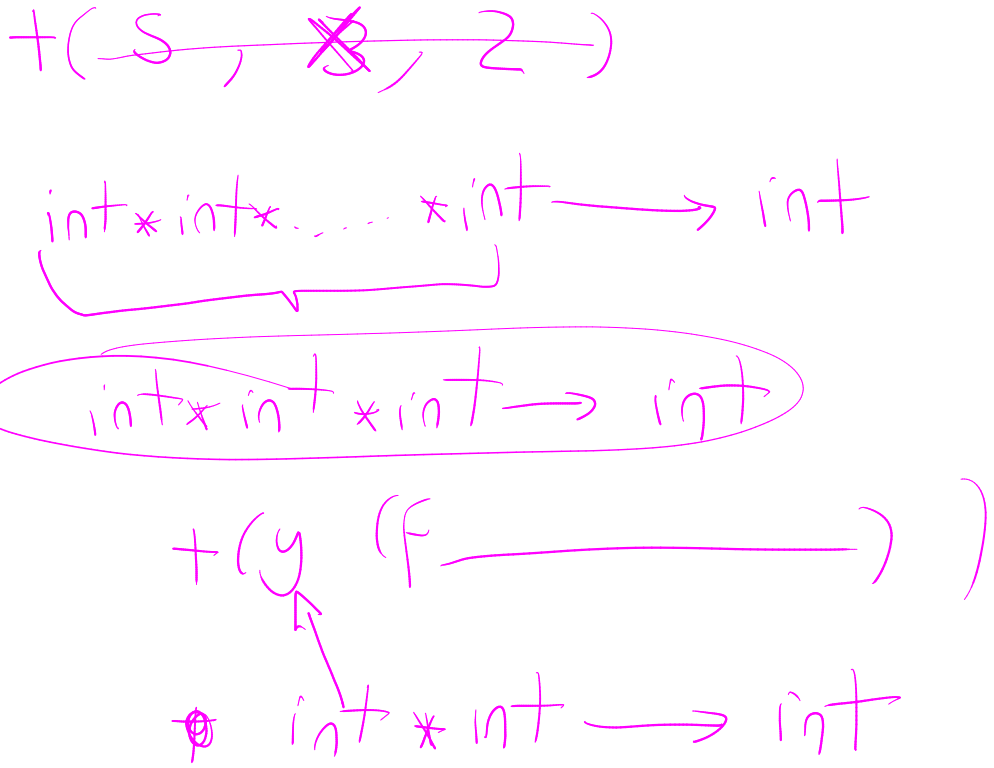
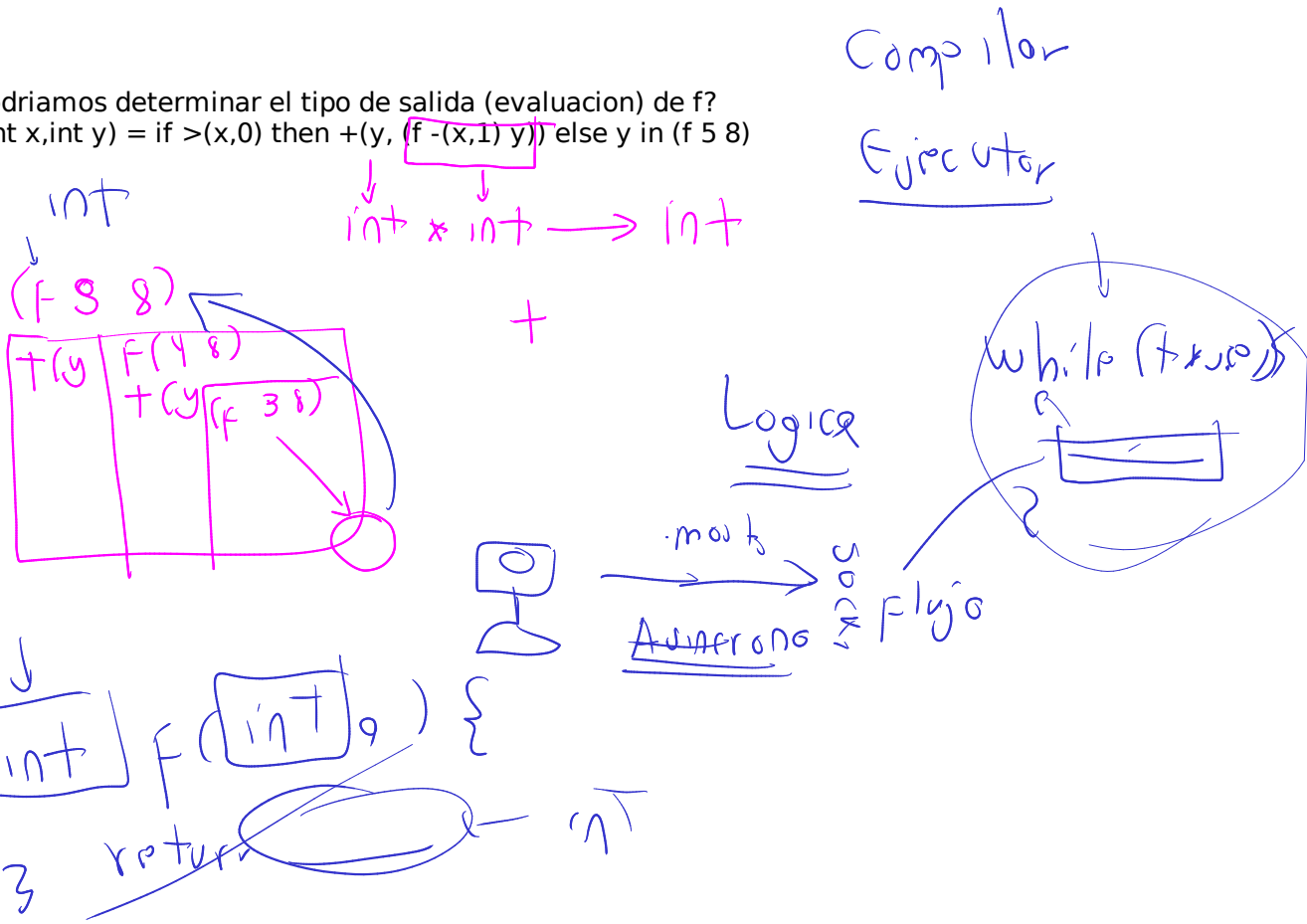
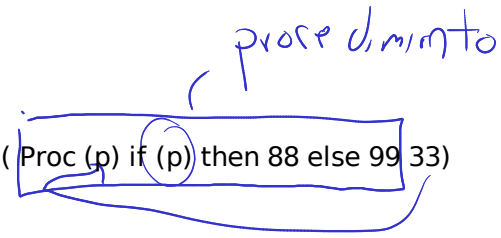
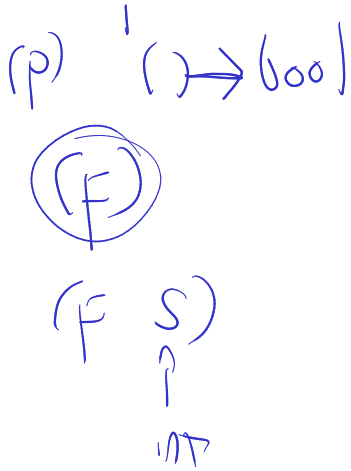


¿Como podriamos determinar el tipo de salida (evaluacion) de f?
letrec f(int x,int y) = if >(x,0) then +(y, (f -(x,1) y)) else y in (f 5 8)

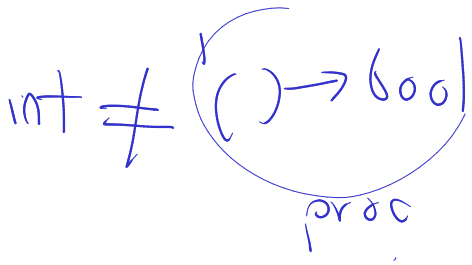




(Proc (p) if (p) then 88 else 99 33)

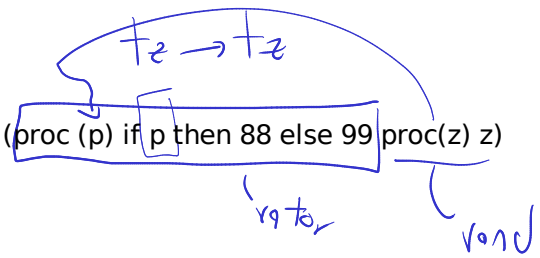


(proc (p) if p then 88 else 99 proc(z) z)

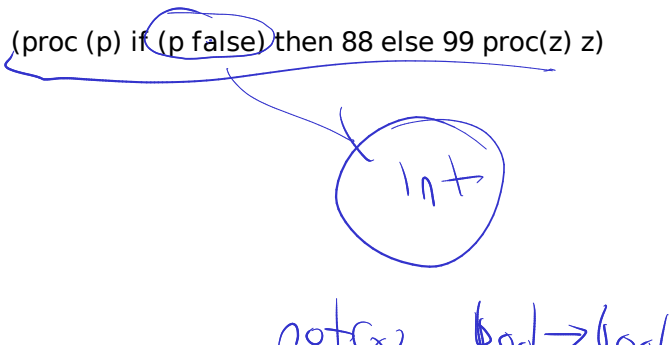


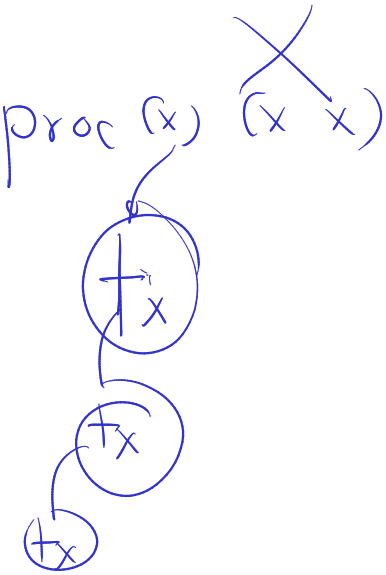
Proc (p) if (p) then 88 else 99 33)

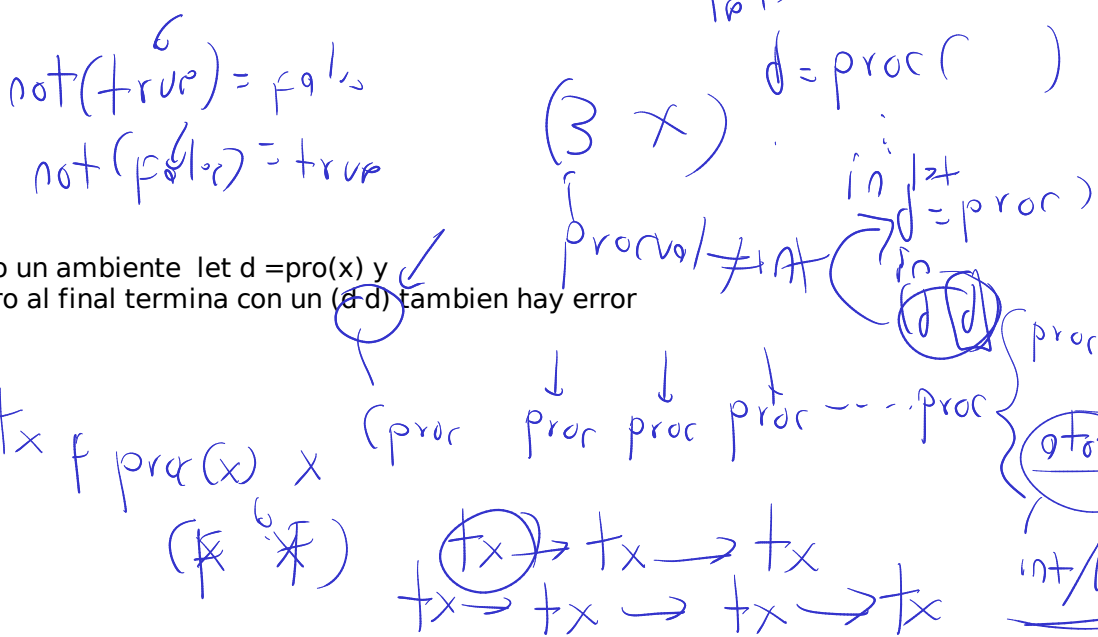
(Proc (p) if (p) then 88 else 99 proc() true)



(proc (p) if (p false) then 88 else 99 proc(z) z)







Profe digamos que tengo un ambiente let d =pro(x) y va otras operaciones pero al final termina con un (d d) tambien hay error

Pre: Si ① es bool y
Si ② es igual al tipo ③

Coste: $\mathcal{O}(\ldots)$ tipo ③

proc(t_1 a, t_2 b, t_3 c) t_p

Si a = t_1 y
Si b = t_2 y
Si c = t_3 y
Si body t_p

$t_1 * t_2 * t_3 \longrightarrow t_p$

(f x y z)

f = (int x bool * int) → bool

Si f es proc val y
Si |F| es tamaño de llamada y
Si x = int y y = bool y z = int
$\mathcal{O}(\ldots)$ bool

$int \to (int \to bool)$

$X^{zero(x)}$ $int \to bool$

proc(int x) proc(int z) >(x, z)

$(int \to (bool \to int)) \to (int \to bool)$

f = int → (bool → int)

proc(n) proc(h) if h then n else 4

proc(f) [proc(x) >(x, ((f 4) true))]

int int

$t_f = int \to (bool \to int)$

$(int \to (bool \to int)) \to (int \to bool)$

proc

$(int \times (int \to bool) \times int) \to (int \to (int \times int \to bool))$

proc(x, y, z) proc(a) proc(b, c)

if (y + (x, z))

then >(a, b)

else >(a, c)

X = int

y = int → bool

z = int

a = int

b = int

c = int

```
let
    j = proc(int x, (int->bool) y)
        if (y 2) then +(x,2) else -(x,3)
    t = proc((int*int->bool) k, int a, int b, int c)
        (k +(a,b) c)
    s = proc(int w) >(w,0)
    in
      let
        p = proc((int->(bool->int)*int*int->int) m , int->(bool->int) n)
            (m n 1 3 s)
         in
            (p t j)
```

$t_t = t_m \qquad t_j = t_n$

$(int*int \to bool) * int * int * int \to bool = (int \to (bool \to int)) * int * int \to int$

1) Nro parametros $4 \neq 3$

2) $t_t \neq t_m$

$t_j = t_n$

$int * (int \to bool) \to int = int \to (int \to bool)$

2) 2 parametros = 1 parametro

2) $t_j \neq t_m$

(m n 1 3 s) 1) num de argumentos

int int int → bool

int → (bool → int)

```
(int * (int ->bool) * (int ->(bool ->int)) ->(int ->bool)
(int * bool *(int->int) * int ->(bool->int)) ->(int ->(bool->int))
(int * (bool->bool) * (int ->(bool->int)))->(bool->int)
```

1)

proc(a, b, c) proc(d)

if (b a) then >(a, ((c 3) false))

else >(d, 4)

and

2)

```
(int * (int ->bool) * (int ->(bool ->int)) ->(int ->bool)
(int * bool *(int->int) * int ->(bool->int)) ->(int ->(bool->int))
(int * (bool->bool) * (int ->(bool->int)))->(bool->int)
```

proc(a, b, c, d) proc(e) proc(f)

if and(b, f)

then +(a, (c 3), (d 4) false))

else -(e, 3)

```
(int * (int ->bool) * (int ->(bool ->int)) ->(int ->bool)
(int * bool *(int->int) * int ->(bool->int)) ->(int ->(bool->int))
(int * (bool->bool) * (int ->(bool->int)))->(bool->int)
```

Usando la Regla

proc(a, b, c) proc(e)

if and(e, (b false))

then +((c 8) false), a)

else -(a, 3)