

Dada la siguiente expresión:

let

$$x = 4$$
$$y = \text{let } a = 4 \text{ in } +(a, 3)$$

$f = \text{proc}(x)$ if $>(x,4)$ then let $a = 8$ in $+(a,3)$ else 4

in

let

$$g = (f, 3)$$

11

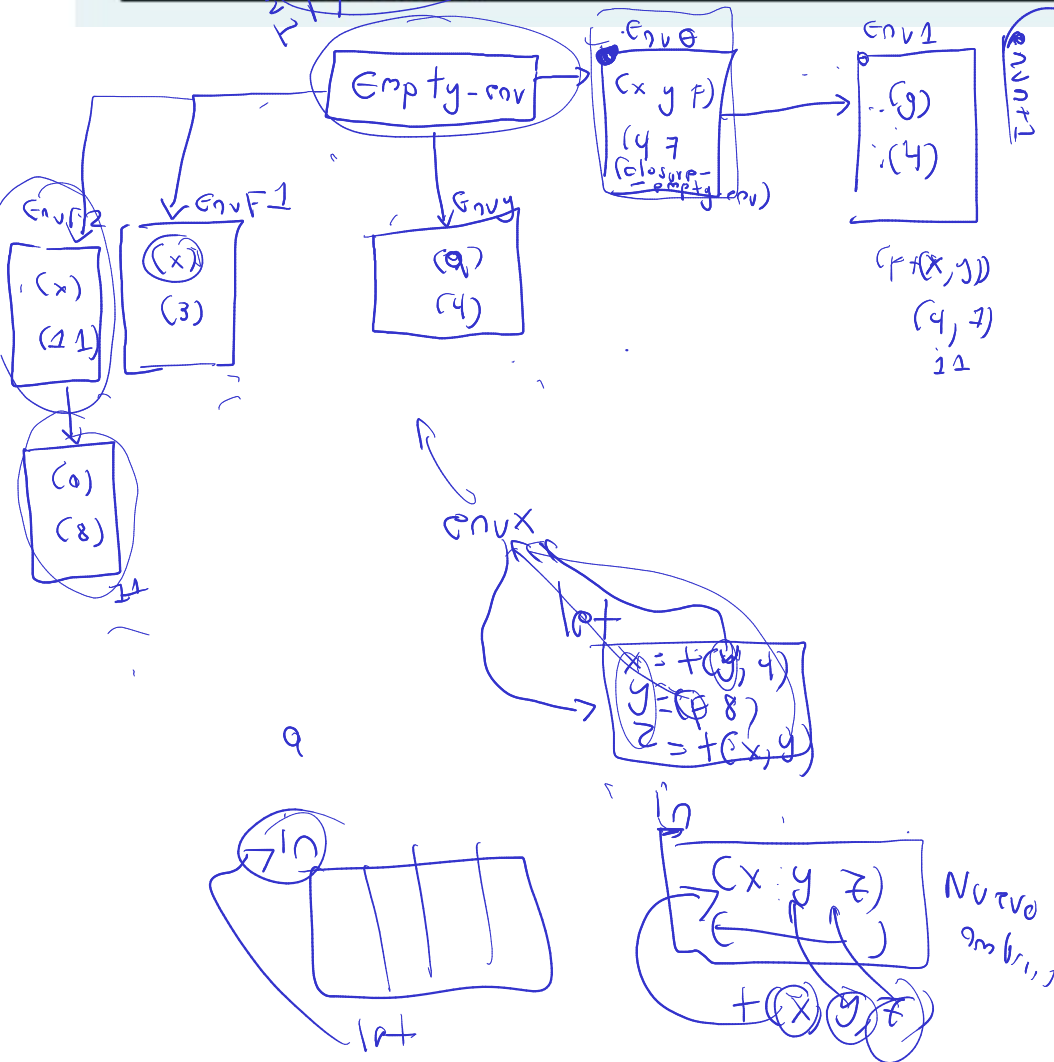
 $(E + (x, y))$

$h = \text{proc} \text{ --- } 11, (r \text{ --- } 11)$

Enn

$$\left(\begin{array}{c} | \sigma + \\ | \langle \sigma_{\text{rel}} \rangle \end{array} \right)$$

empty - env)



1) closure amb anterior

2) cada vez que se un proc
se genera un ambiente
que nace del ambiente
de la closure.

Of course

Νύττο
9m 6, 1, 2

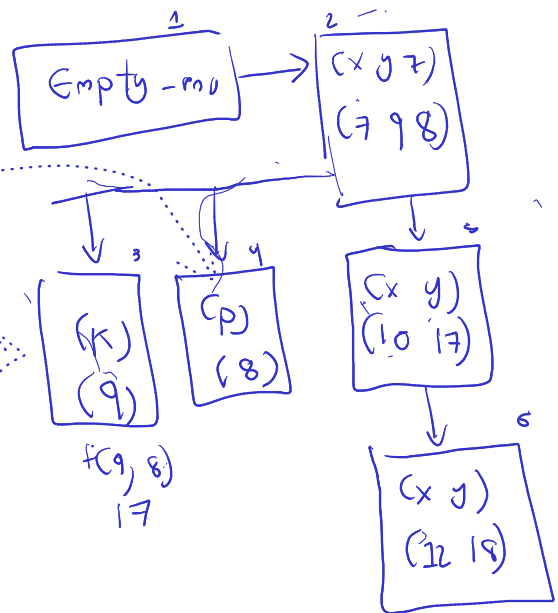
Al realizar la siguiente expresión, con ambiente inicial $(x,y,z) (7\ 8\ 9)$

let $k =$ let $p = +(x,1)$ in $+(p,2)$ y $=$ let $k = +(x,2)$ in $+(k,z)$
in let $x = +(x,2)$ y $= +(x,z)$
in $+(x,y)$

¿Cuántos ambientes se generan?

- 3 ☐
- 7 ☐
- 5** ☒
- 4 ☐
- 6 ☐

$\langle \text{exp } \text{vp} \text{ } \text{io} \rangle$





Se tiene el siguiente programa:

```
let (x = 5)
in let f = proc (y, z) +(y, -(z (x)))
in (f 2 28)
```

- Cuando se llama a `f`, su cuerpo debe ser evaluado en un ambiente que liga `y` a 2, `z` a 28 y `x` a 5.
- `x` es ligado a 5 ya que el alcance de la declaración interna no incluye la declaración del procedimiento.
- Las variables que ocurren libres en el procedimiento se evalúan

lambda x

(closure



let

a = (g a b c)

b = (g b c a)

c = (g c a b)

in

let

f = proc(a,b,c) if a then b else +(a,b)

in

let

x = (f (f a b c) (f a b c) 0)

y = (f (g a b c) (f a b c) 0)

z = (f (g a b c) (g a b c) 0)

in

+(x,* (y,z))

Dado el ambiente ambiente inicial env0 con identificadores (a b c g) y valores (2 3 4 (closure'(x y z) +(x,* (y z)) empty-env))

$g = \text{proc}(x,y,z) \rightarrow +(x, *(y,z))$

($\neq 90$)

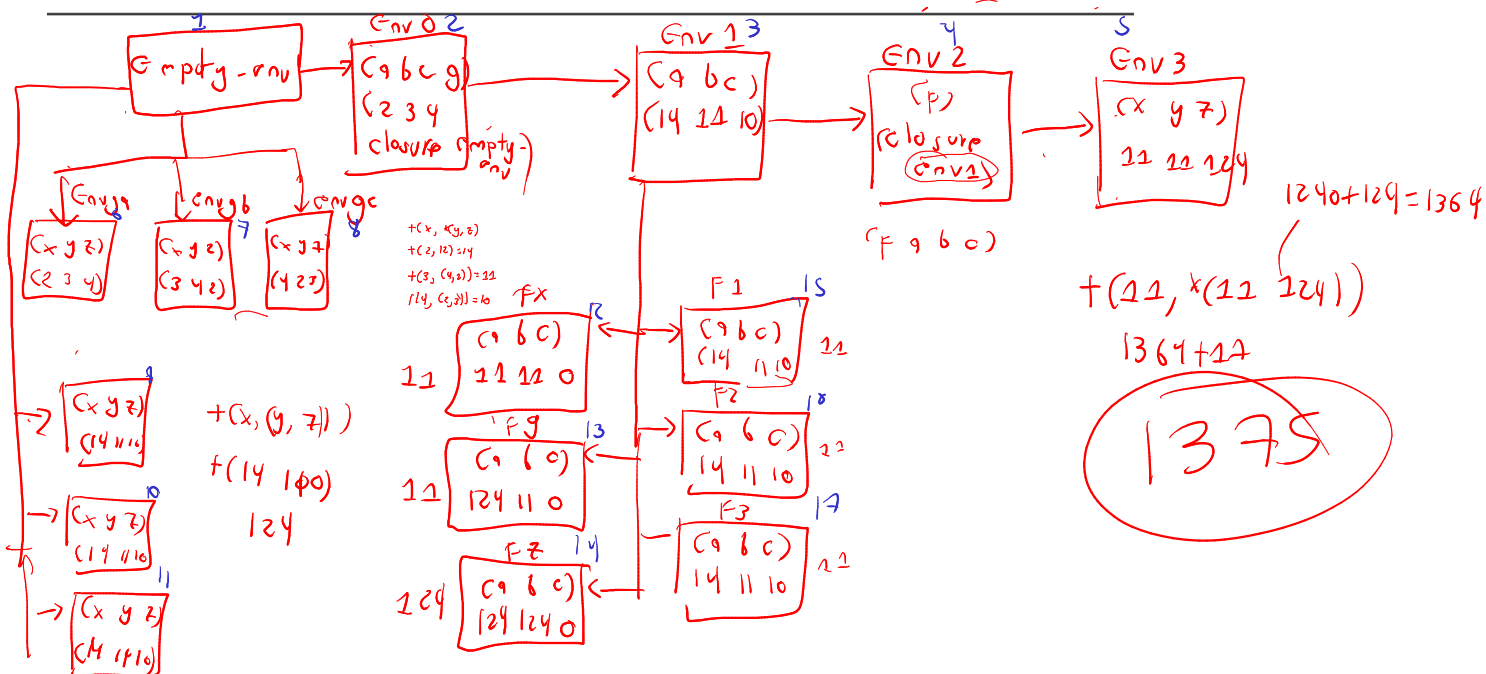
$9 \neq 0$
 $9 = 0$

1375

$(f \ 2 \ 2 \ 0) = 22$

$(f \ 124 \ 2 \ 0) = 12$

$(f \ 124 \ 124 \ 0) = 124$

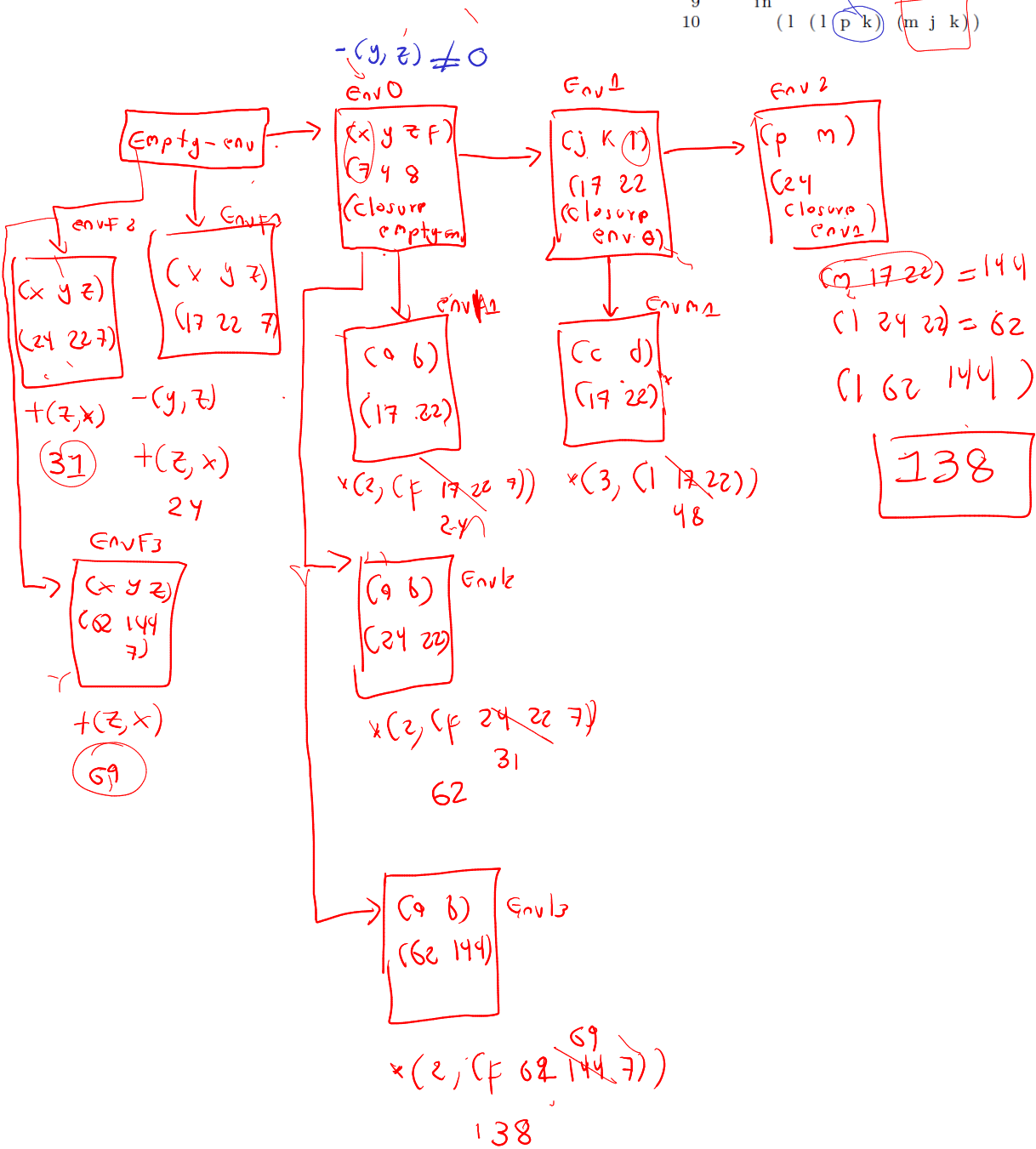


env0 con identificadores (x y z f) y valores (7 4 8 (closure'(x y z) if -(y,z) then +(z,x) else -(y,z) empty-env))

```

1 let
2   j = 17
3   k = 22
4   l = proc(a, b) *(2, (f a b x))
5 in
6   let
7     p = 24
8     m = proc(c, d) *(3, (l c d))
9     in
10      (l (l p k) (m j k))

```



```

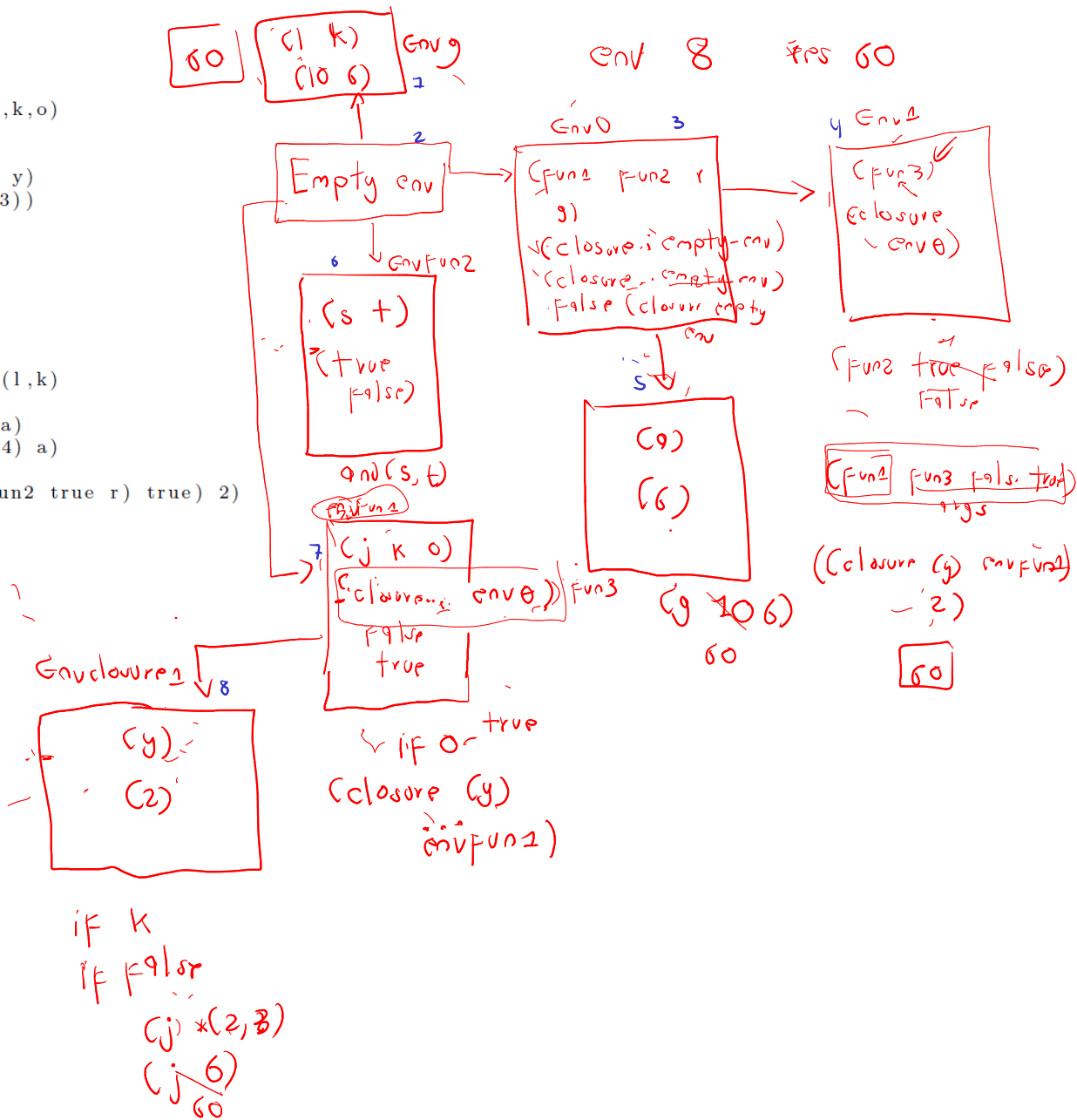
let fun1 = proc (j,k,o)
  if o then
    proc (y)
      if k then (j y)
      else (j *(y,3))
    else proc(w)
      *(w,3)

```

```

fun2 = proc(s,t)
  and(s,t)
  r = false
  g = proc (l k) *(l,k)
in
  let fun3 = proc(a)
    (g +(a, 4) a)
  in
    ((fun1 fun3 (fun2 true r) true) 2)

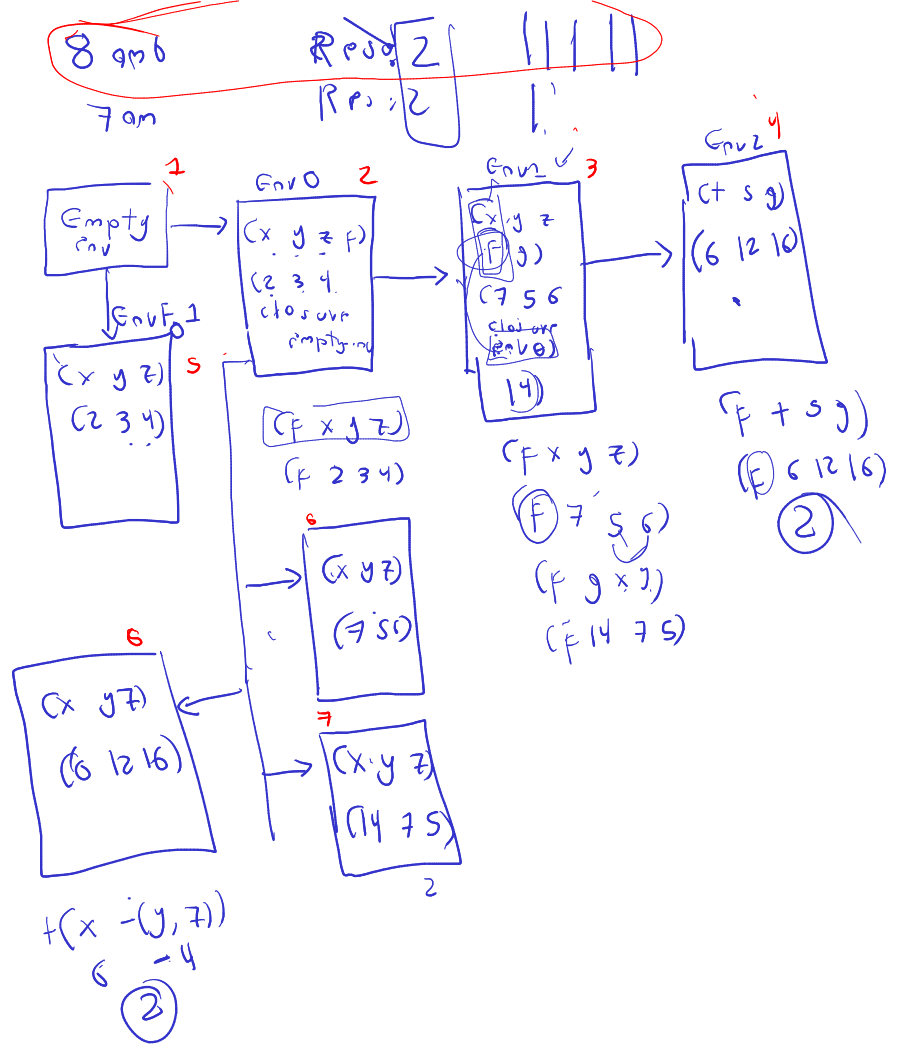
```



3. Evaluación de expresiones [20 puntos]

Considere la siguiente expresión en el lenguaje visto en el curso (procedimientos), con ambiente inicial *env0* con identificadores (*x y z*) y valores (2 3 4) (*closure* (*x y z*) *+(x, *(y z))* *empty-env*)

```
let
  x = 7
  y = 3
  z = 6
  f = proc(x, y, z) +(x, -(y, z))
  g = (f x y z)
in
  let
    t = (f x y z)
    s = +(x, y)
    g = (f g x y)
  in
    (f t s g)
```



25

```

let
  a = 2
  b = 3
  c = 4
  f = proc(x,y) proc(a,b) proc(c,d) +(x,y,a,b,c,d)
  → g = proc(a) >(a,3)
in
  let
    x = if (g a) then (if a b c) else 4
    y = ((if 2 3) 3 4) 5 4
  in
    +(x,y)

```

closure envF)

closure envF2)

