Env 0

let
    x = 3 ← ID
          ↳ Args
  → y = 2
    in

$+(x, y)$ ← Evalúa en un ambiente extendido

```
'(x y)
(3 2)
env 0
```

---

empty-env 0 → (x y z) (4 z 5)
- env

let
  /x = -(y, z)   X = 1
  in

let
    x = +(x, z) = 3
    in
      add1(x)

env 1
```
'(x)
(1)    env 0
       (x y z)
       (4 z 5)
       empty-env
```

env 2
```
'(x)
(3)
env 1
```

bind

empty-env

env0 (x y z) (4 2 5)

let
    z = 5
    t = sub1(x) = 3
in

env1
(z,t)
(5, 3)
env2

env2
(x)
(2)
env1

env3
(y)
(4)
env2

let
    x = -(t, 1)   x = 2
in
    let
        y = 4
    in
        *(t, -(z, -(x, y)))
        *(3, -(5, -(2, 4))) → 21

-(5, -2)
*(3 7) -

let
    extend
in

empty-env

Ejercicio diap. 28 (38)

env0

'(x y z)
'(4 2 5)

env_t

'(p q)
(5 2)

env1

'(z t)
'(5 7)

env2

'(x)
(6)

*(+, *(x, y))

*(7, *(6, 2))
                12
        84

Emptyenv

env0
'(x y z)
'(3 7 1)

env1
'(y m)
'(5 18)

env_m
'(t)
(6)

×(t, x)
*(6, 3)
18

4→True
if sub1(y) then +(add1(y), m)

6, 18
24

env2
(y)
(24)

-(y, m)
(24, 18) →6

env3
(t)
(6)

+(t, +(y, -(z, 3)))
+(6, +(24, -(1, 3)))
-2
22
28