

apply-env: Retorna un valor directamente

apply-env-ref: Retorna una referencia que representa una variable. La referencia se construye en tiempo de ejecución. Cada vez que se consulta variable (operación, conocerla, pasarla como parametro) se retorna siempre una referencia.

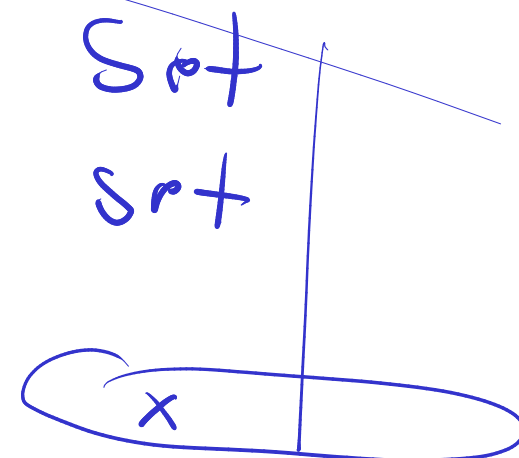
begin: Siempre se ejecuta sobre el mismo ambiente, el begin es una serie de expresiones.

foreach ... (eval-expression exp amb)

Begin tiene la característica de siempre retorna el ultimo valor

```
begin
  +(5,3);
  -(2,1);
  +(2,2)
end
```

Retorna 4



¿Que pasaria con el siguiente ejemplo?

```
let x = 100 in
  let
    p = proc(a)
      begin
        set x = add1(x);
        a
      end
  in
    +( (p x), (p x))
```

Los llamados de las primitivas, se evaluan de izquierda a derecha

Paso por valor

+ ((p x), (p x))
 ((p 100) (p 101))
 100 101
 201

Paso por referencia

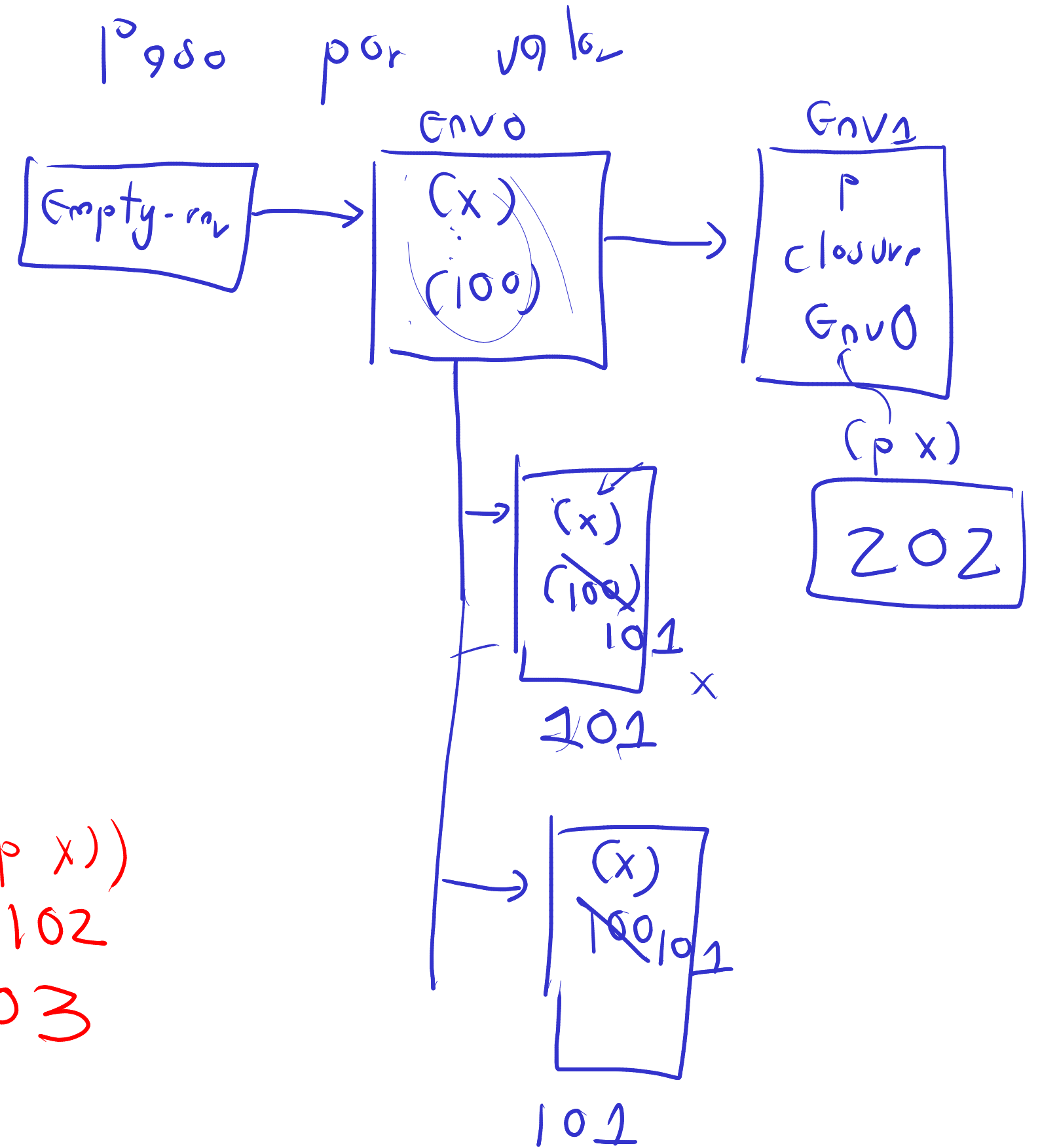
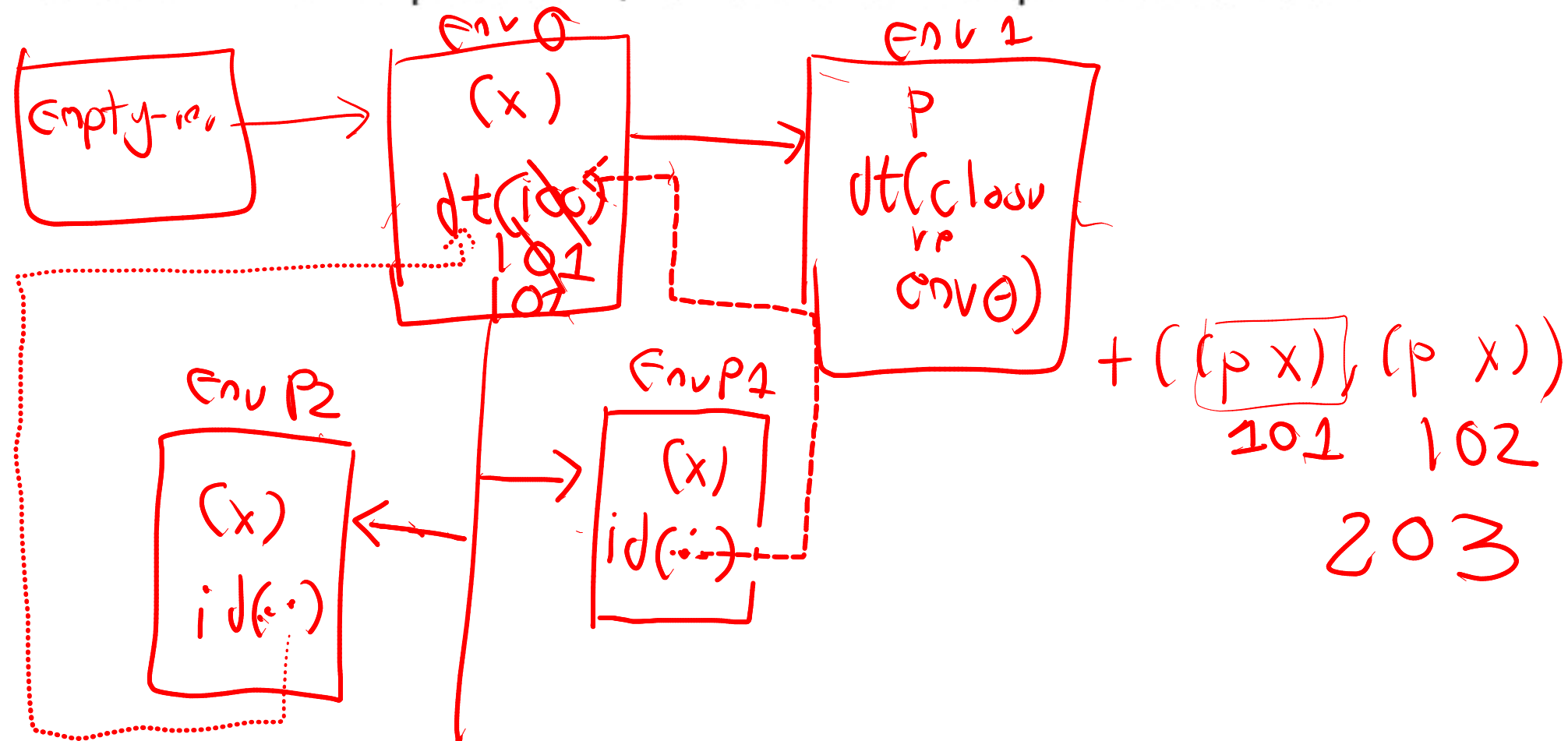
+ (p x), (p x))
- (p id(x)) (p x))
- 101 102
- 203

¿Que pasaria con el siguiente ejemplo?

```

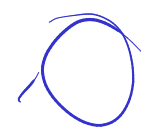
let x = 100 in
let
  p = proc(a)
  begin
    set x = add1(x);
  end
in
  +( (p x) , (p x))
  
```

Los llamados de las primitivas, se evaluan de izquierda a derecha

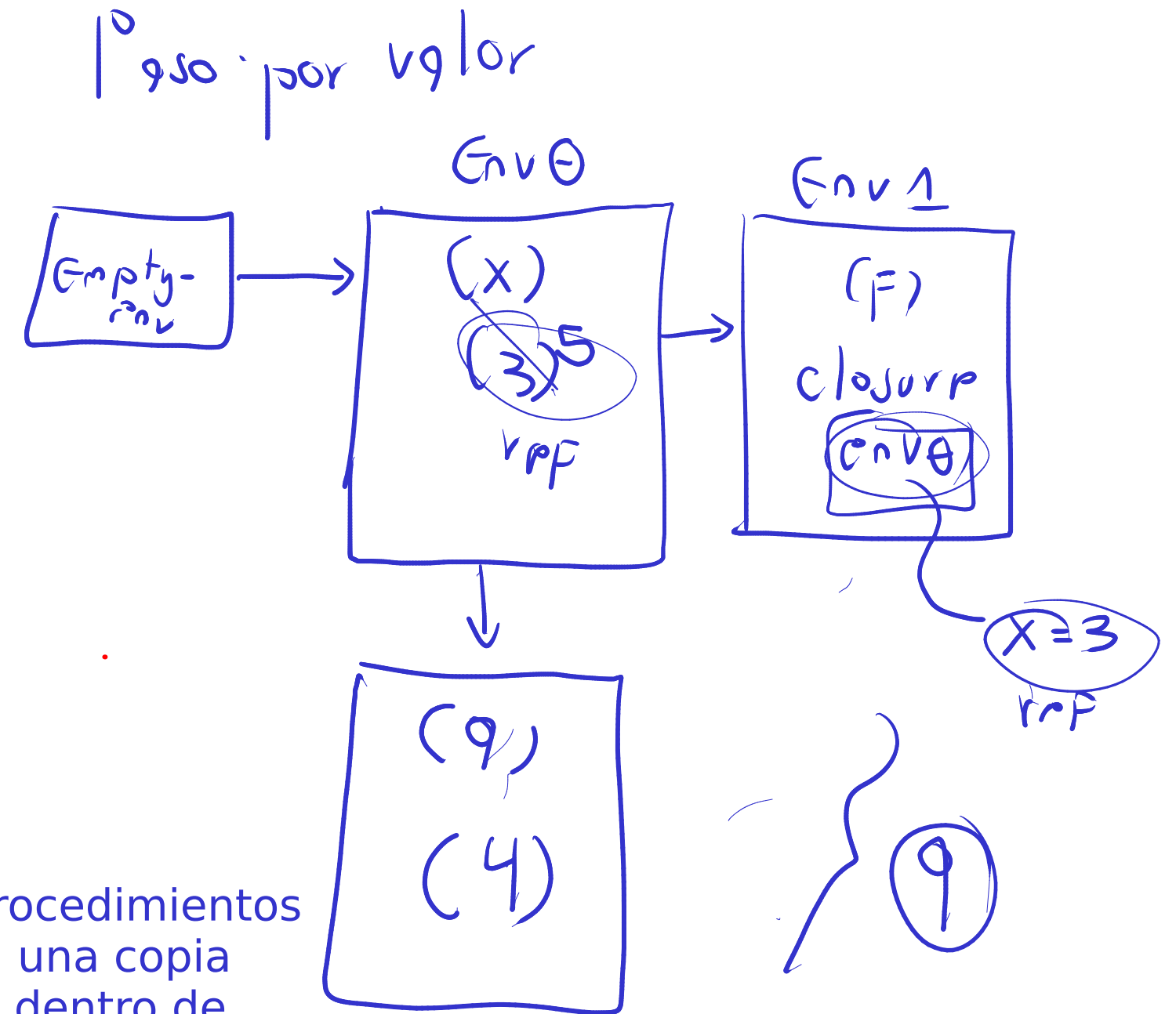


Dada la siguiente expresión

```
let x = 3
in
  let
    f = proc(a) +(a,x)
  in begin
    set x = 5;
    (f 4)
  end
```



Si no se manejan referencias los procedimientos
NO VEN los cambios porque tienen una copia
del ambiente en su estado original dentro de
su clausura.



WhatsApp ro-carlos/blog Inteligencia arti Videos Recurso clase 011121 tra 6. Asignaci Sapiencia: Ejerci Universidad del Meet - FLP

campusvirtual.univalle.edu.co/moodle/mod/hvp/view.php?id=1381134

(2) WhatsApp Tips Linux The Final Countdo... W3 Home | Web Access... Microsoft Word - T... clock | Font Aweso... Evaluación Heurística LinkedIn ¿Cómo usar bases...

Universidad del Valle Español - Internacional (es) PERDOMO LONDOÑO MANUEL ALEJANDRO

letrec f(x) =
if >(x,0) then *(x,(f sub1(x))) else 1
in
let j = (f 6)
in
begin set f = 6; +(j,f) end

Env1
Empty-env → (f) closure to env1

MÁS VIDEOS Activar Windows YouTube 1:30 / 5:24

○ x _____

D

letrec

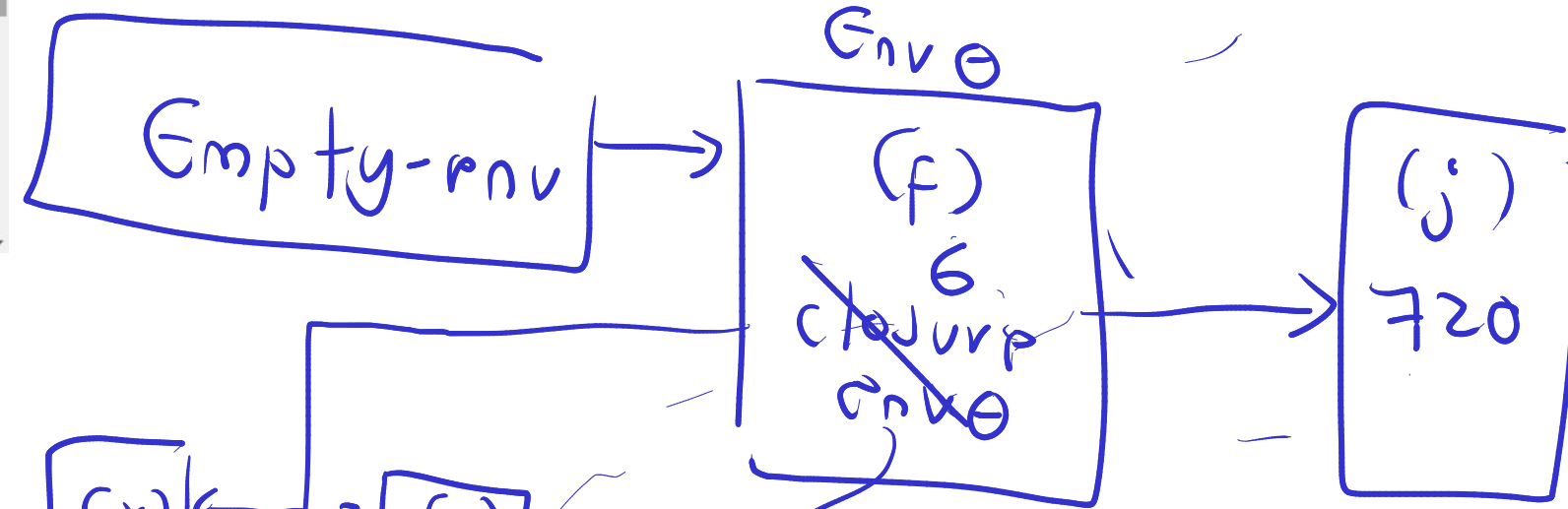
f(x) = if >(x,0) then *(x,(f sub1(x))) else 1

in

let j = (f 6)

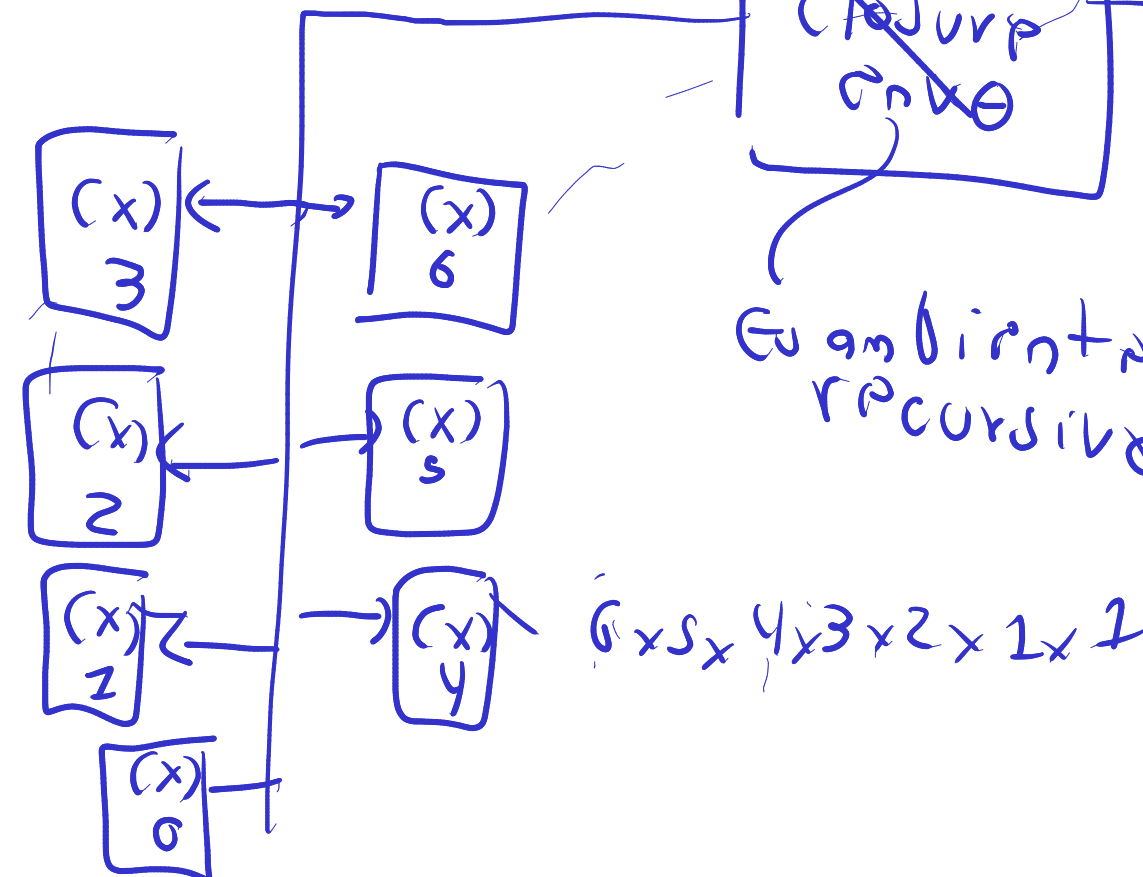
in

begin set f = 6; +(j,f) end



Environment
recursive

726

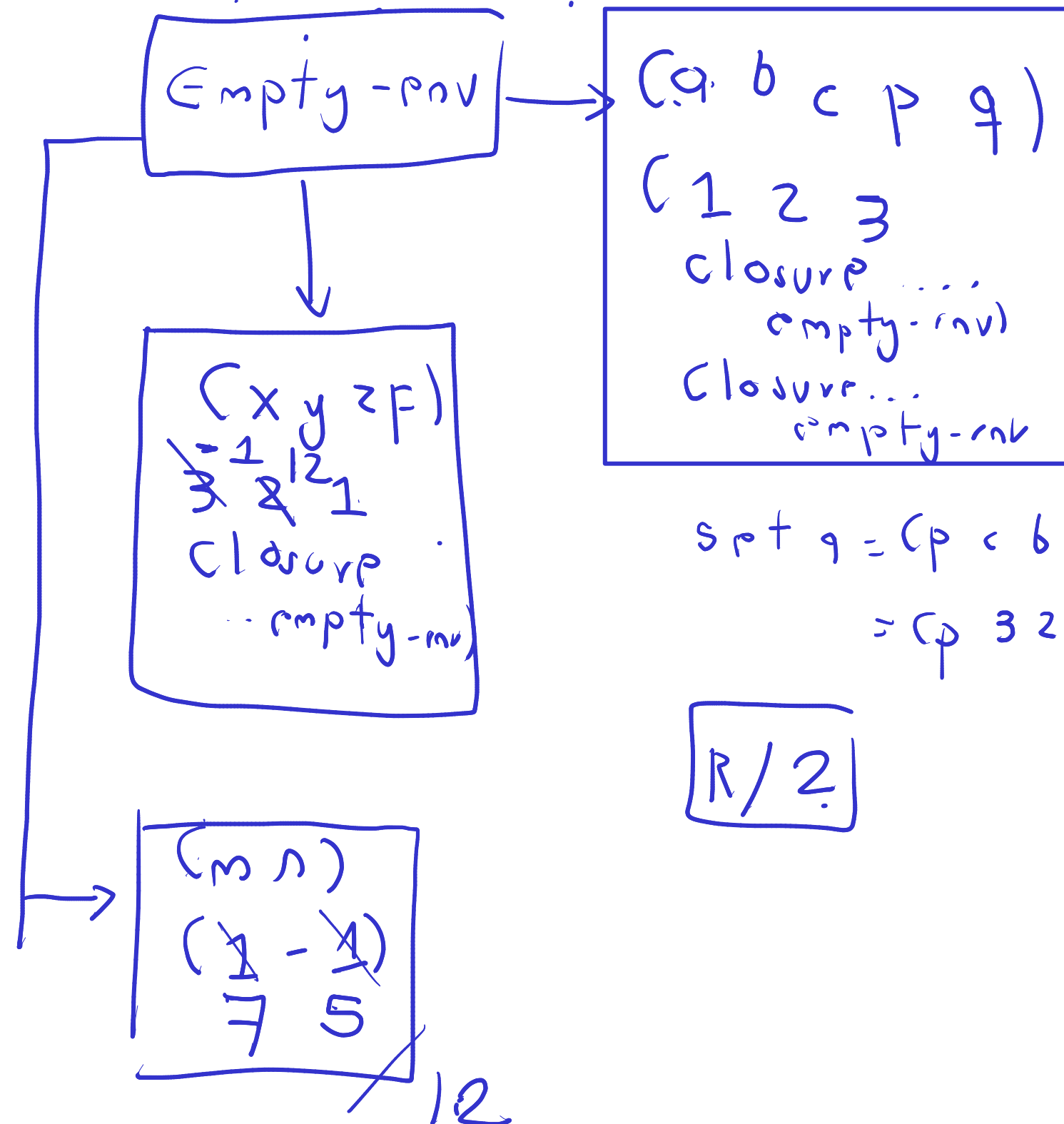


```

let a = 1
    b = 2
    c = 3
    p = proc (x, y, z, f)
        begin
            set x = -(y, x);
            set y = (f z x);
            x
        end
    q = proc (m, n)
        begin
            set n = *(m, 5);
            set m = +(m, 6);
            +(m, n)
        end
in
    begin
        set a = (p c b a q);
        +(a, c)
    end
end

```

* p9so por valor



$$\text{set } q = (p \ c \ b \ q \ (f))$$

$$= (p \ 3 \ 2 \ 1 \ \text{closure}) = -1$$

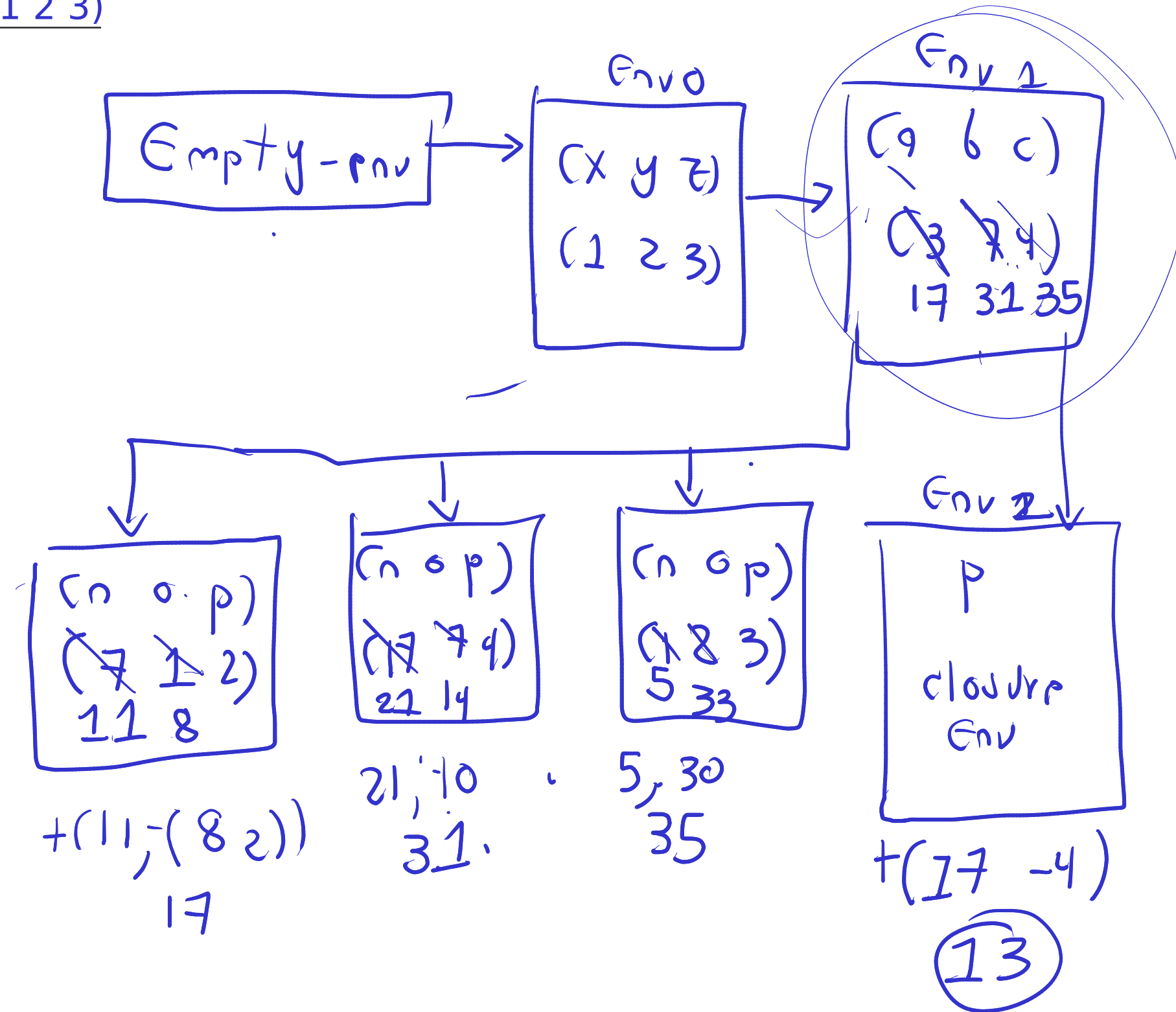
R/2

AMbiente inicial (x,y,z) (1 2 3)

```

let
  a = 3
  b = 7
  c = 4
  in
    let
      p = proc(n o p) begin
        set n = +(n, c);
        set o = +(o, b);
        +(n, -(o, p))
      end
    in
      begin
        set a = (p b x y);
        set b = (p a b c);
        set c = (p x y z);
        +(a, -(b, c))
      end
    end
  end

```

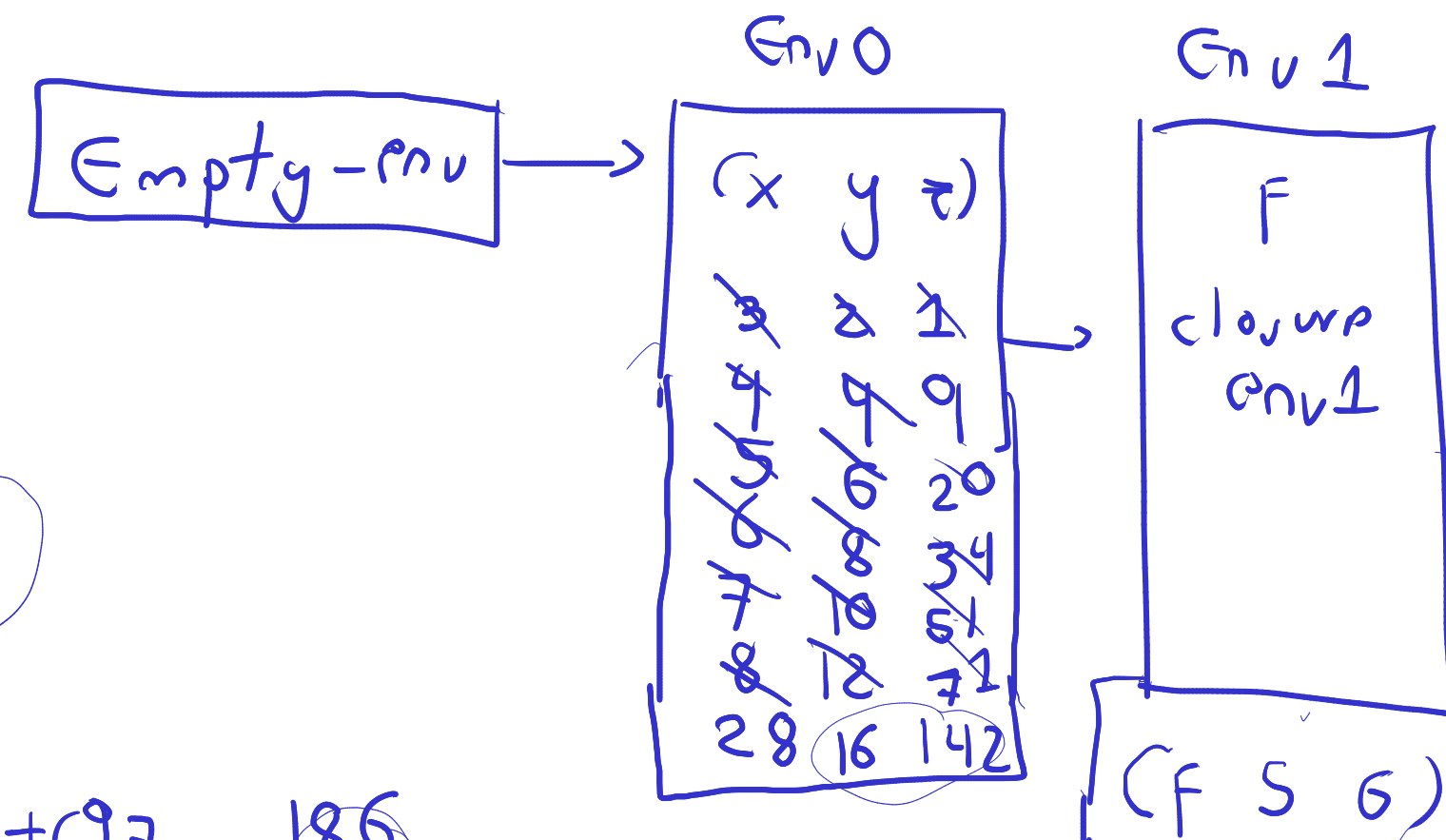


```

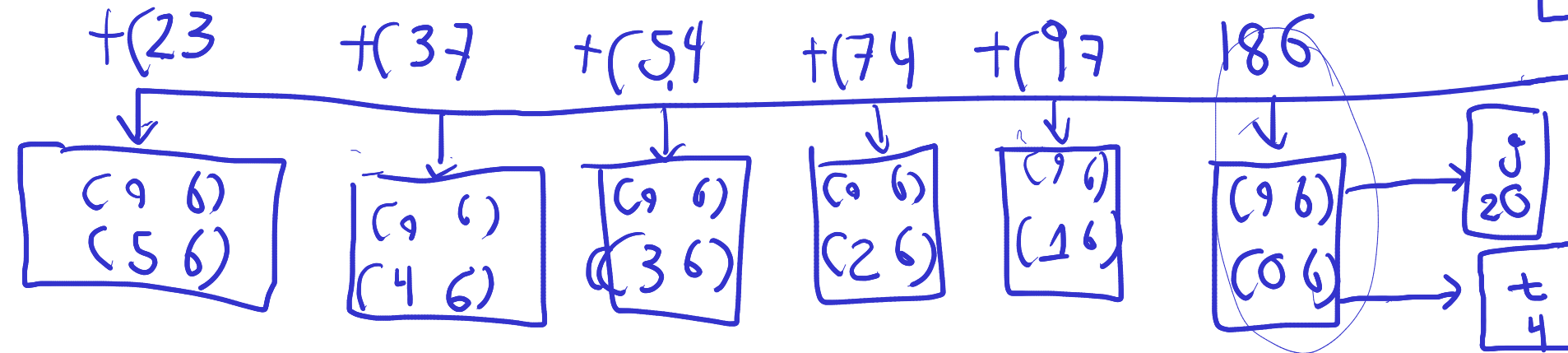
let
  x = 3
  y = 2
  z = 1
in
  letrec
    f(a,b) = if >(a,0) then begin
      set x = +(x,1);
      set y = +(y,2);
      set z = +(z,x,y);
      +(b,x,y,z, (f -(a,1) b))
    end
  else
    begin
      set x = let j = +(x,y) in +(x,j);
      set y = let t = 4 in if >(x,4) then +(y,t) else -(y,t);
      set z = *(2,z);
      +(x,y,z)
    end
  in
    (f +(x,y) +(x,y,z))

```

4 7 1



in
(f +(x,y) +(x,y,z))



4 7 1

6 7 8 6

letrec f(x,y) = if >(x,0) then
begin
set y = let p = *(2,y) in +(p,y);
+(y, (f -(x,1) y))
end

else
let p = proc(x) *(x,2) in (p y)

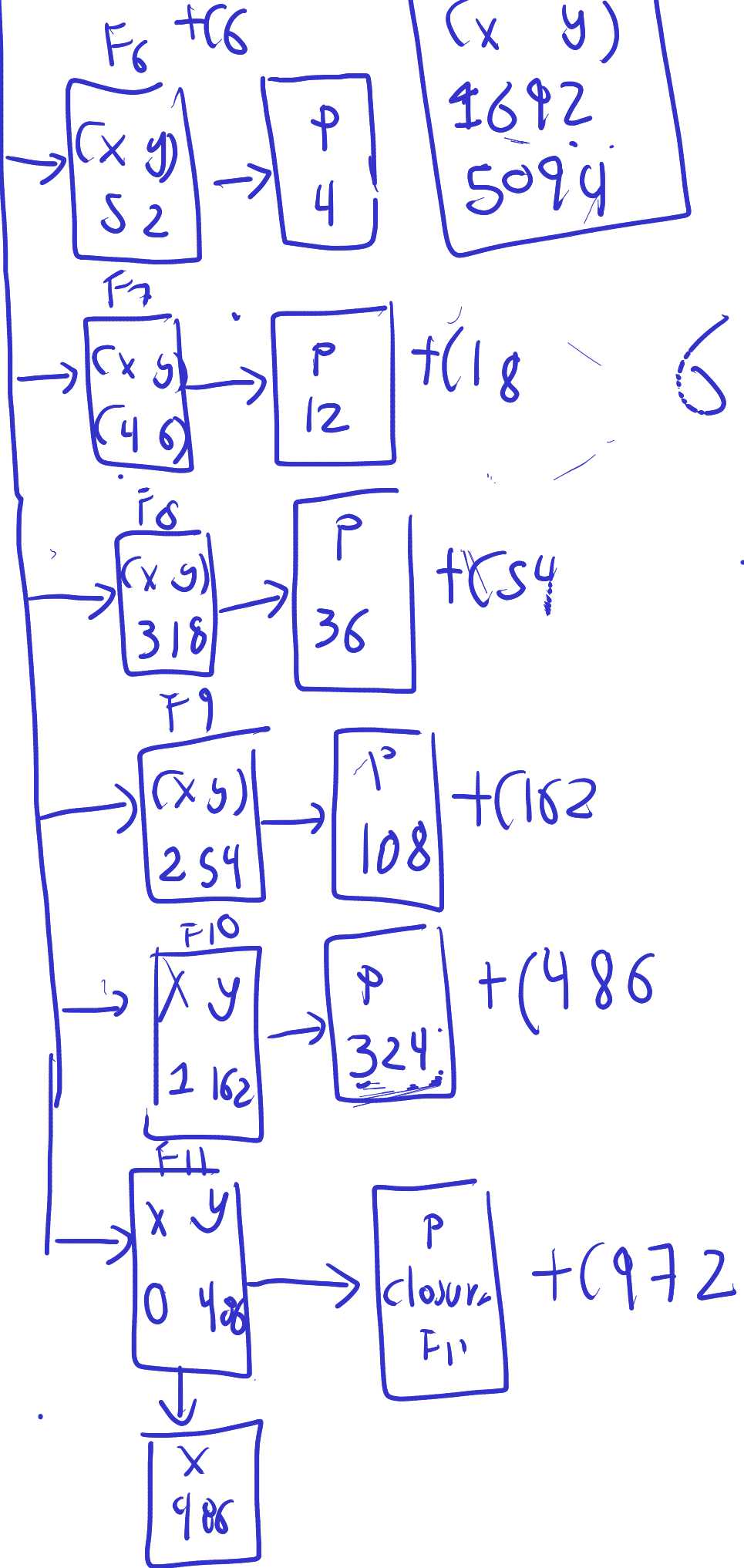
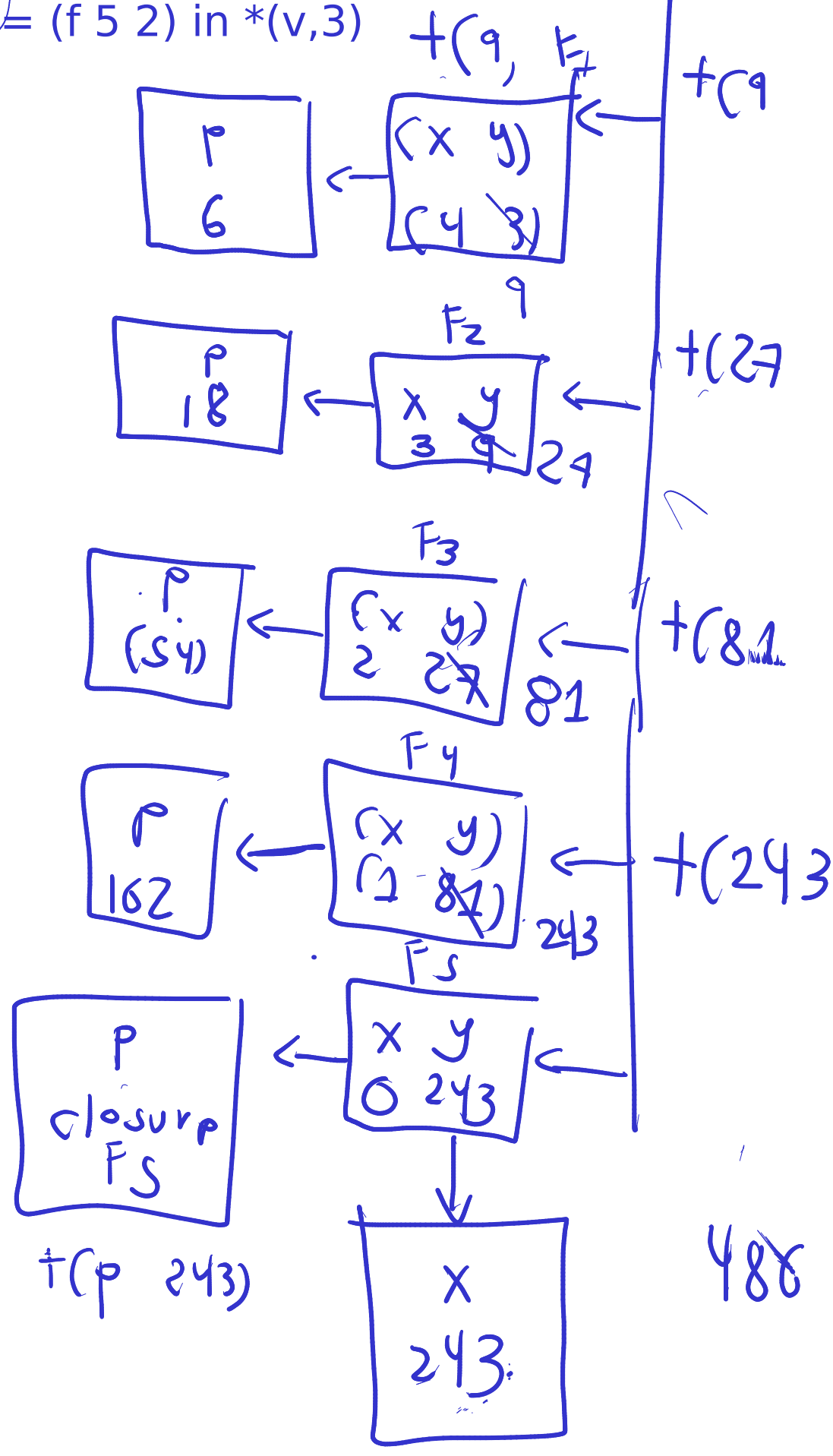
in
let
x = let u = (f 4 3) in *(u,2)
y = let v = (f 5 2) in *(v,3)
in
+(x,y)

Empty-env

Env Θ
(f)
closure
env Θ

u
846

v
1698



6 7 8 6 ✓