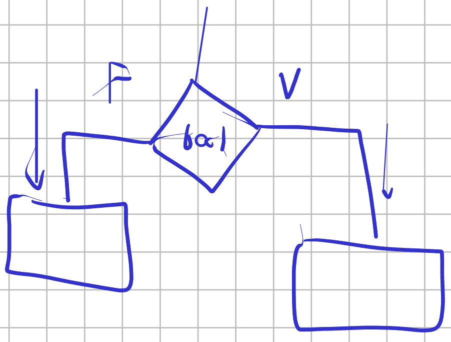
Condicionales

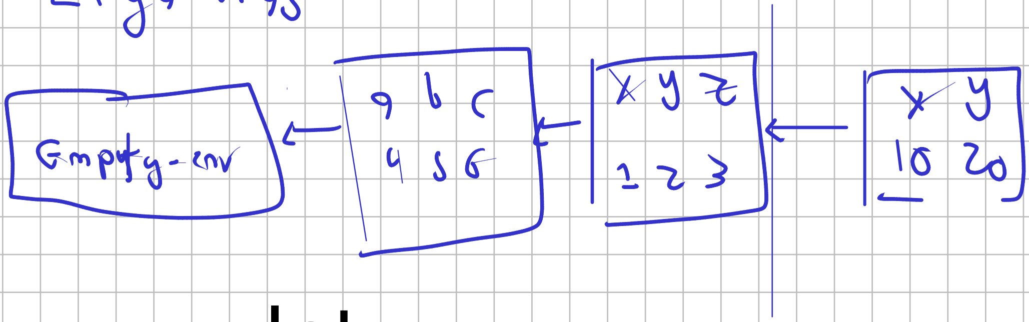


<expression> ::= "true"
(true-exp) bool

<expression> ::= "false"
(false-exp)

<expression> ::= "if" <expression> "then"
<expression>
"else"
<expression>

Ligaduras



let

x = 10
y = 20

in

+(x,y)

Procedimientos

if <(x,18) then "menor de edad"
else "mayor de edad"

Creación

```
public int suma(...)
    return ...
```

Uso / llamado

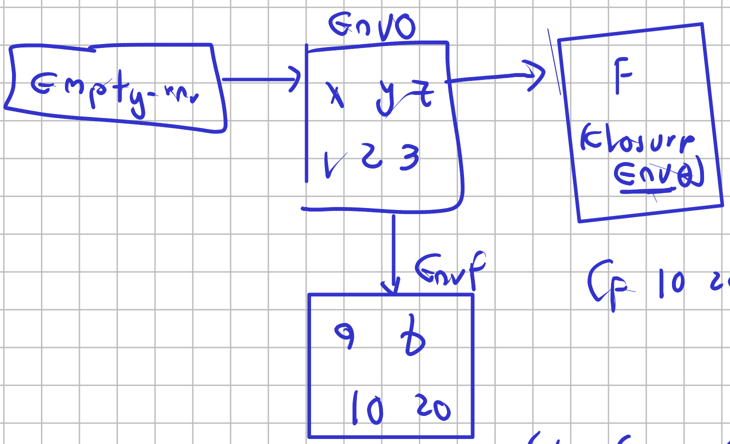
```
suma(...)
```

Clausura

- argumento
- Cuerpo
- Ambiente creado

```
let
    f = proc(a,b) if zero?(a)
                        10
                    else
                        +(b (f -(a,1) b))
    in   (f 10 20)
```

x y z   1 2 3



(f 10 20)

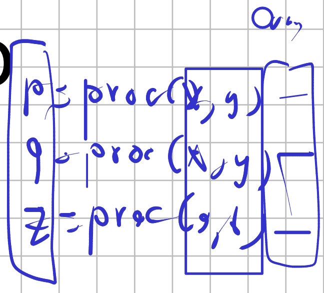
$+(b, (f\ -(a,1)\ b))$

$+(20, (\boxed{F}\ 9\ 20))$

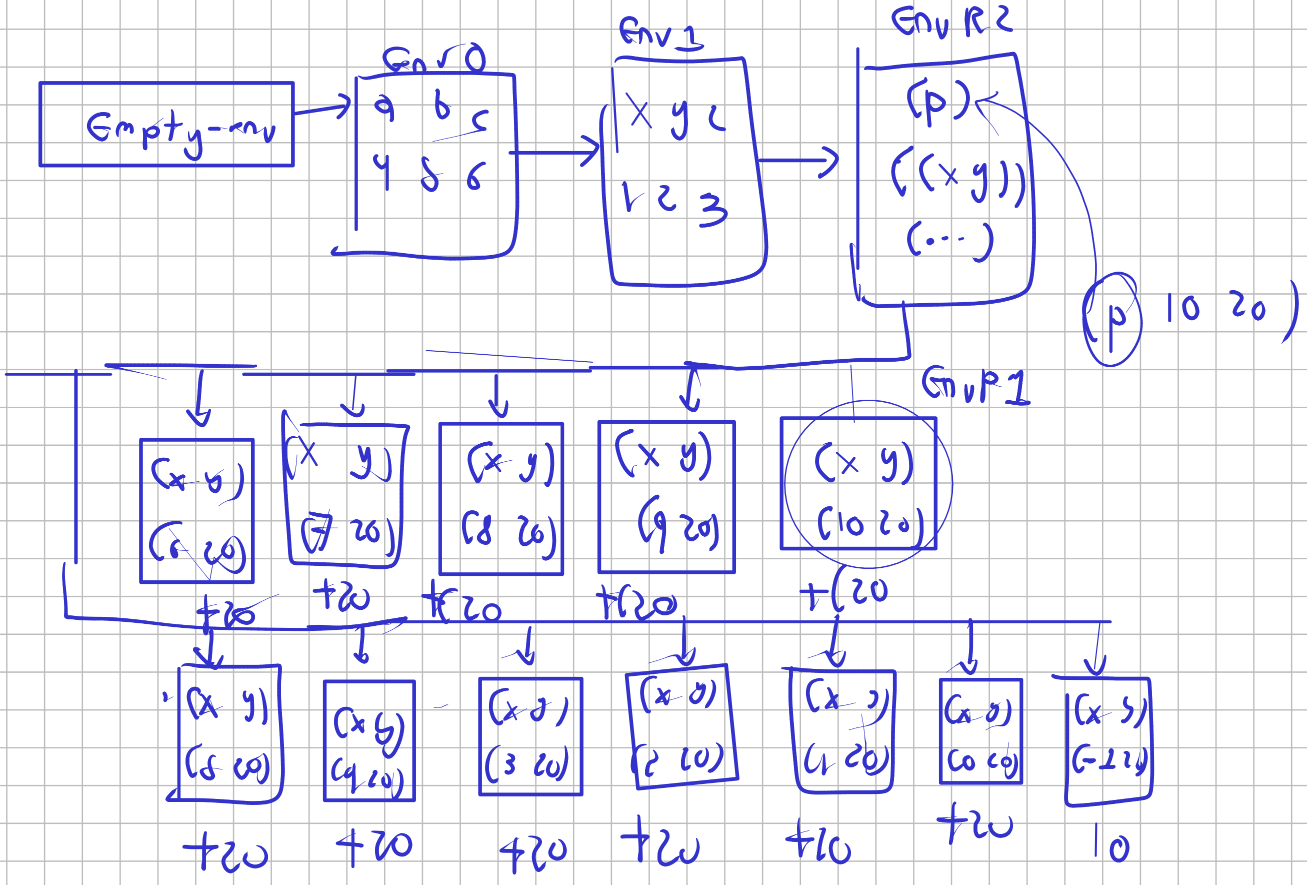
Ambiente extendido recursivo

Sólo tiene definiciones de FUNCIONES.

```
(ambiente-extendido-recursivo
          (proc-names)
           (lista de los id de proc)
           (lista de los cuerpos)
           ambiente-anterior
```

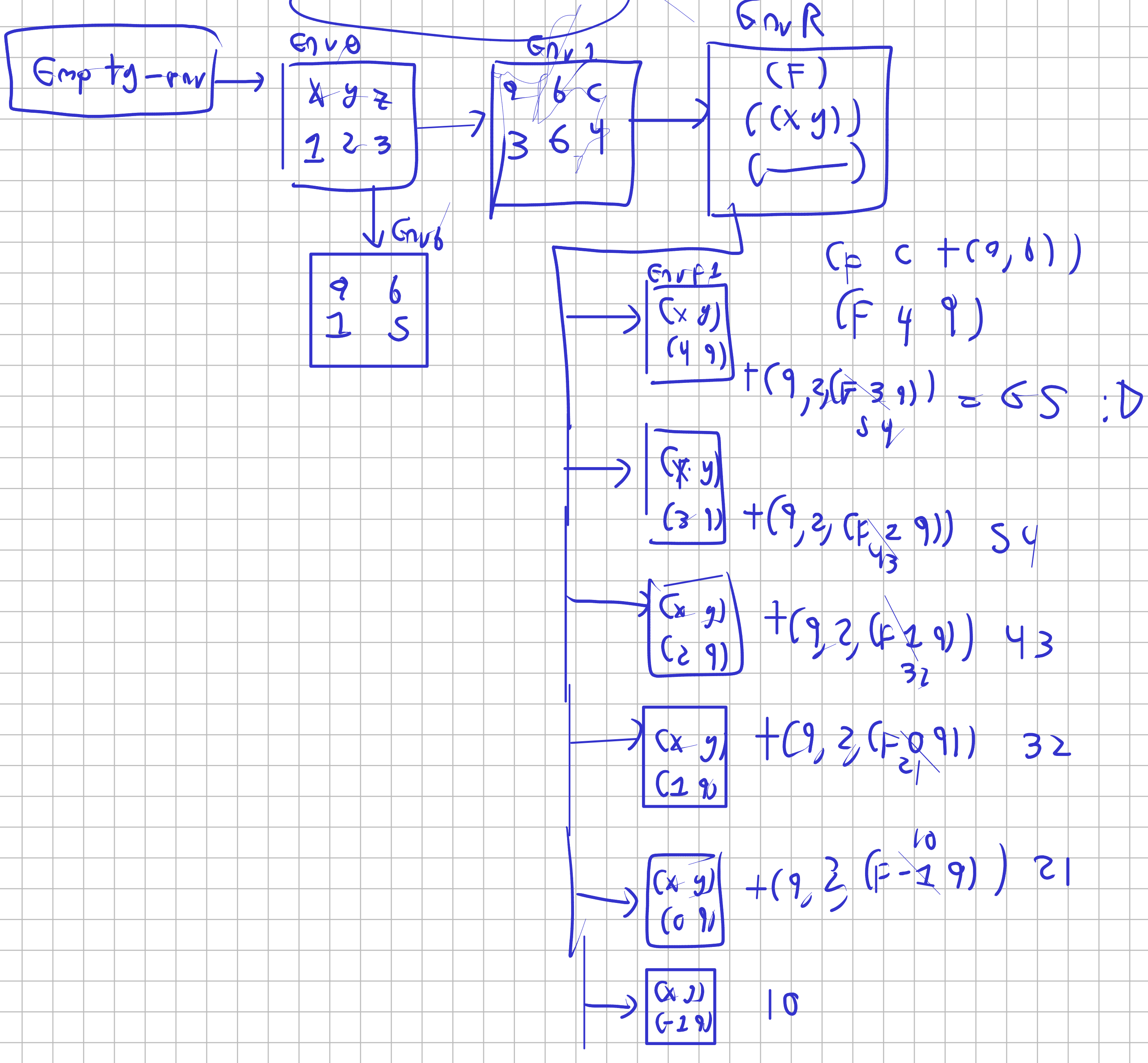


```
localrec p(x,y) =
   si opera <[x,0] entonces 10
   sino opera +[y, (p opera -[x,1] y )] en (p 10 20)
```



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

```
let
      a = +(x,y)
      b  = let a = x b= +(y,3) in +(a,b)
      c = 4
 in
           letrec
                   f(x,y) = if >=(x,0) then
                              +(y, 2, (f -(x,1) y)) else 10
            in
               (f c +(a,b))
```

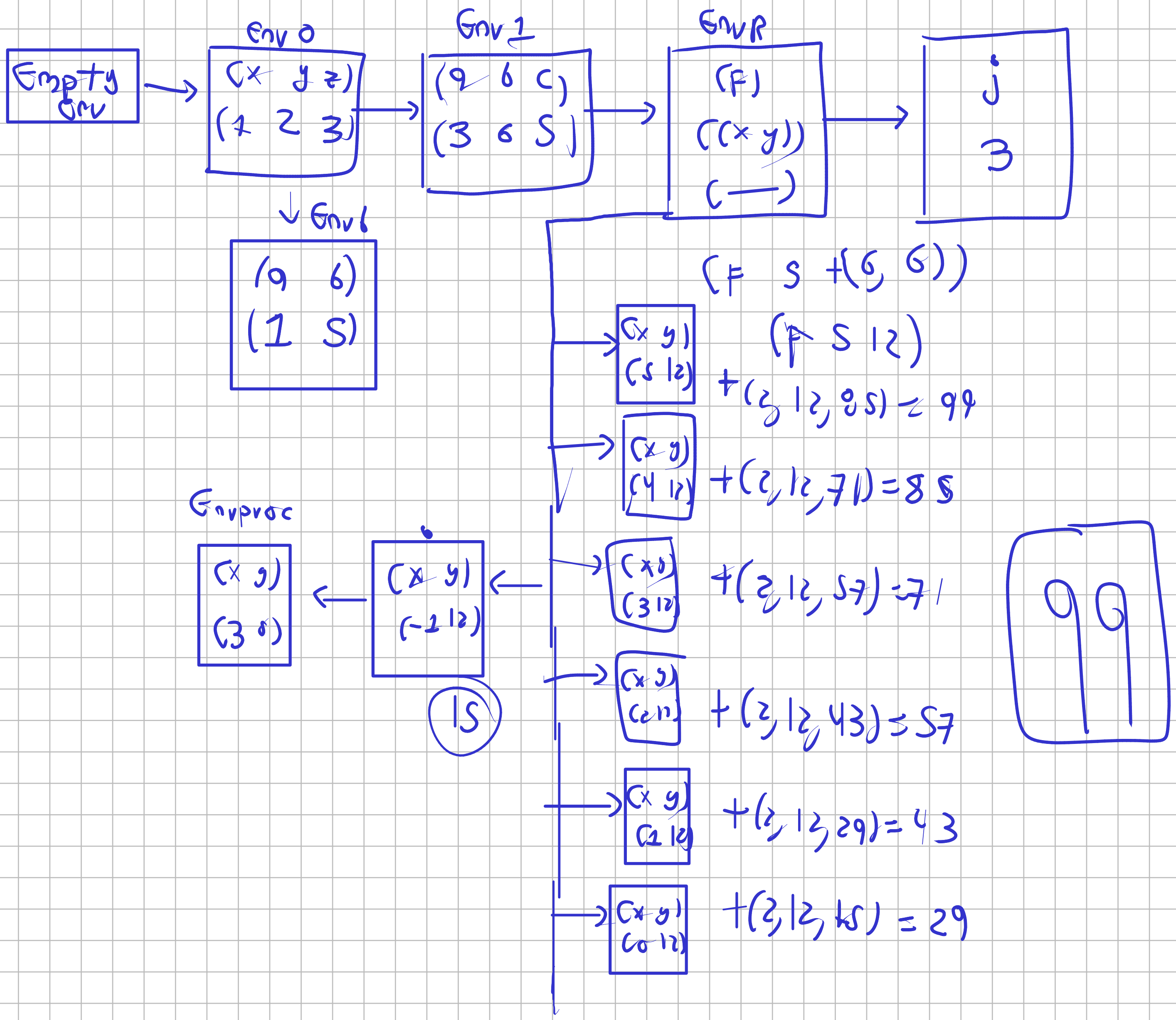


Ambiente inicial (x, y,z) (1,2,3)

```
let
      a = +(x,y)
      b  = let a = x b= +(y,3) in +(a,b)
      c = 5
     in
         letrec
               f(x,y) = if >=(x,0) then
                              +(y, 2, (f -(x,1) y))
                  else ( proc(x,y) *(x,y) a c)
          in
              (f c +(let j = +(x,y) in *(2,j),b))
```

99

Env R



Ambiente inicial (x,y,z,f)
(4,2,3, closure (a) *(2,a) empty-env))

```
let
   a = let k = (f (f x)) in k
   b = let t = let j = (f y) in j in +(t,(f z))
   c = 5
   in
       letrec
                p(x,y) = if >(x,0) then +(y, (q -(x,1) y))
                      else 12
                q(a,b) = if >(a,0) then *(b, (p -(a,1) y))
                     else let t = 9 in +(t,a)
        in
            let
                  s = (p a c)
                  t = (q a b)
                  in
                       (proc (x) +(x,y) +(s,t))
```

25577