1. (30 puntos) Dada la siguiente gramática:

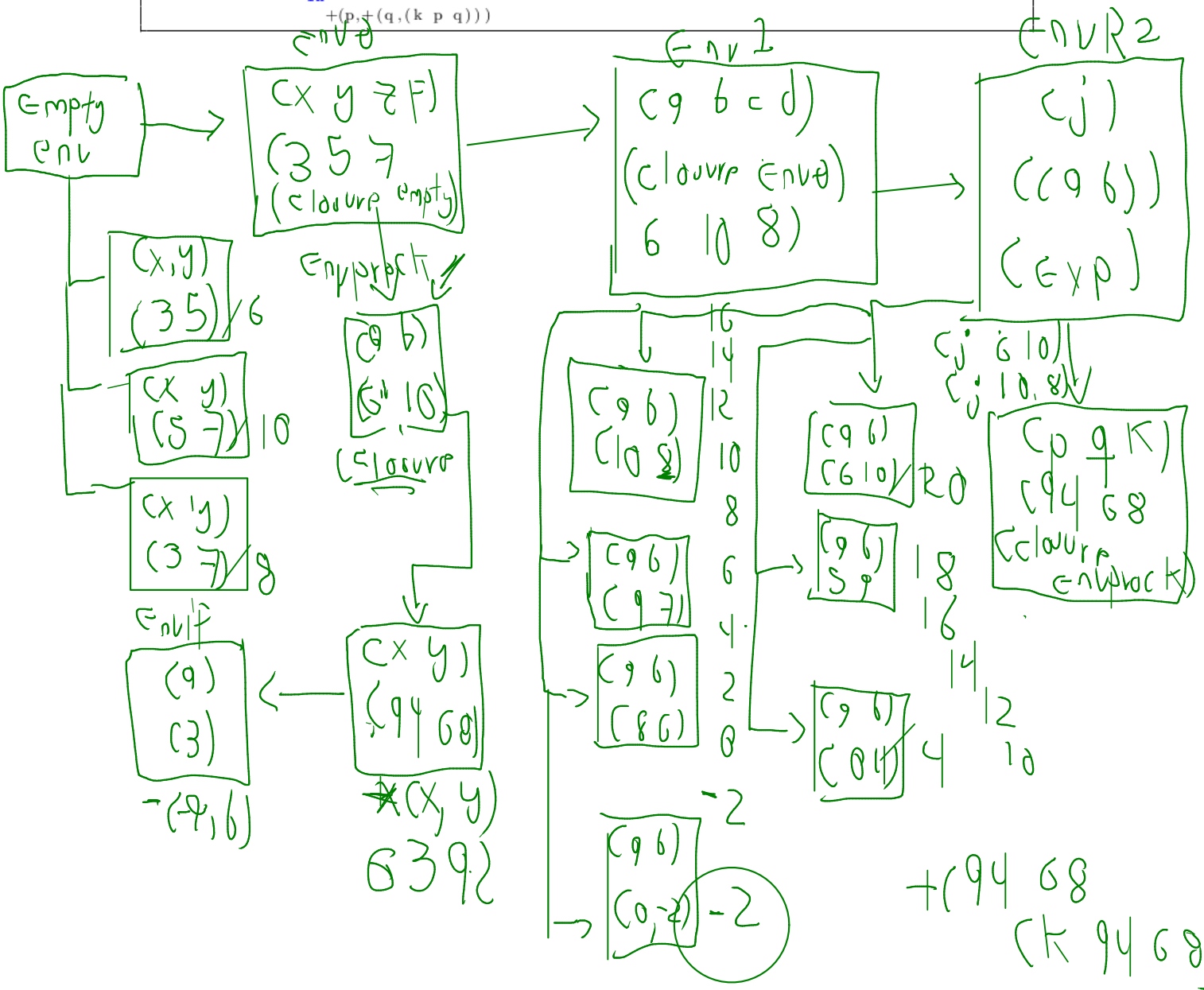
```
<lista-S> ::= (lista-Svacia)    '()
          ::= (lista-Snovacia) <lst> <lista-S>

<lst> ::= (sim) <simbolo>
      ::= (lsim) <simbolo> <lst>
```

Diseñe las funciones constructoras, predicados y extractores que considere necesarios usando una representación basada en **procedimientos**. Además construya la función buscar-s que recibe una ¡lista-S¿y un símbolo, esta función retorna verdadero si dentro de la estructura está el símbolo ingresado.

Revisar archivo adjunto

(30 puntos) Dado la siguiente expresión con ambiente inicial '((x y z f) (3 5 7 (closure '(x y) +(x,-(y,2))) empty-env))

```
let
  a = proc(a b) proc(x y z) if let a = 3 in -(a,b) then *(x,y) else *(y,z)
  b = (f x y)
  c = (f y z)
  d = (f x z)
  in
     letrec
          j(a,b) = if a then +(*(2,b), (j -(a,1) -(b,1))) else b
          in
             let
                p = (j b c)
                q = (j c d)
                k = (a b c)
                in
                  +(p,+(q,(k p q)))
```



let a = 3
in
let
p = p r o c (x) −(x , a)
a = 5
Nota:
in
−(a , (p 2))