



Fundamentos de lenguajes de programación

Duración 2 horas

Carlos Andres Delgado S, Msc

carlos.andres.delgado@correounivalle.edu.co

07 de Mayo de 2022

1. (35 puntos) Considerando el ambiente inicial (x,y,z,f) , $(4,2,1, \text{closure } '(a,b) +(2,+(a,b)) \text{ empty-env})$ Dibuje los ambientes para la expresión:

```
let
  a = (f x y)
  b = (f x z)
  c = (f y z)
in
  letrec
    g(x,y) = if >(y,0) then +(*(2, x), (g +(x,1) (y,1))) else (f a b)
  in
    (proc (n,m) (g n m) b c)
```

El resultado es 107

2. (35 puntos) Se desea agregar las listas al lenguaje, mediante la siguiente gramática.

$\langle \text{expresion} \rangle ::= "[(\langle \text{expresion} \rangle (,)) * "]"$

a) (15 puntos) Indique la expresión que se agrega a la gramática en el formato SLLGEN.

b) (20 puntos) Indique la producción (cases) que se agregaría a la función evaluar expresión.

3. (30 puntos) Dada la siguiente gramática:

$\langle \text{expresión} \rangle ::= \langle \text{identificador} \rangle$

var-exp (id)

$::= (\text{lambda } (\langle \text{identificador} \rangle) \langle \text{expresión} \rangle)$

$\text{lambda-exp (id body)}$

$::= (\langle \text{expresión} \rangle \langle \text{expresión} \rangle)$

$\text{app-exp (rator rand)}$

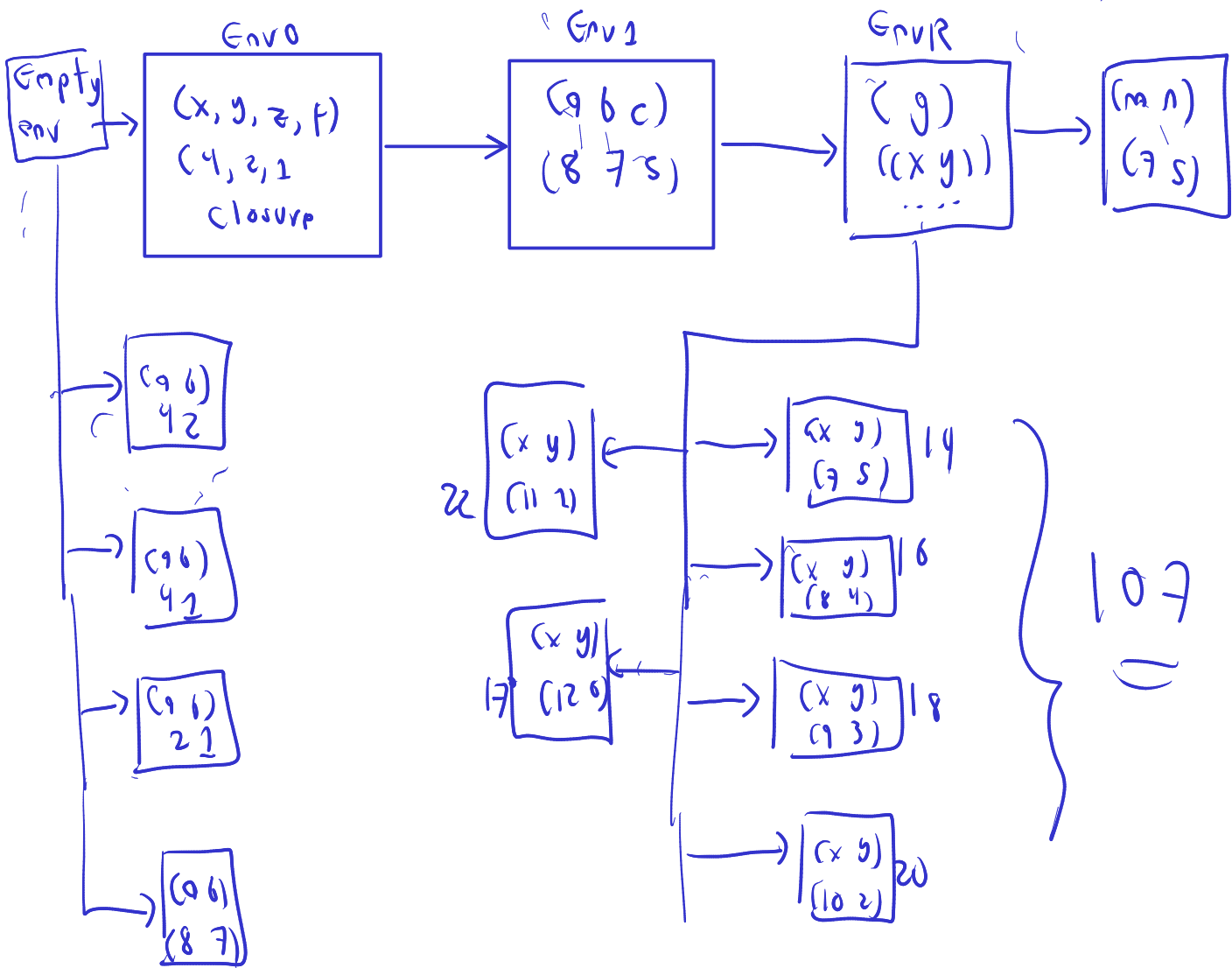
Dibuje el árbol de sintaxis abstracta de la siguiente expresión:

$((\text{lambda } (x) (x y)) (a (\text{lambda } (x) (a (\text{lambda } (y) (a (b c)))))))$

```

let
  a = (f x y)
  b = (f x z)
  c = (f y z)
in
  letrec
    g(x,y) = if >(y,0) then +(*(2, x), (g +(x,1) -(y,1))) else (f a b)
  in
    (proc (n,m) (g n m) b c)

```



(35 puntos) Se desea agregar las listas al lenguaje, mediante la siguiente gramática.

`<expresion> ::= "[" (<expresion> (,))* "]"`

a) (15 puntos) Indique la expresión que se agrega a la gramática en el formato SLLGEN.

b) (20 puntos) Indique la producción (cases) que se agregaría a la función evaluar expresión.

1) (expresion ("[" (separated-list expresion ",") "]")
list-exp)

2) (list-exp (lexp)
(map (lambda (x) (evaluar-expresion x amb)
lexp))

3. (30 puntos) Dada la siguiente gramática:

`<expresión> ::= <identificador>`

`var-exp (id)`

`::= (lambda (<identificador>) <expresión>)`

`lambda-exp (id body)`

`::= (<expresión> <expresión>)`

`app-exp (rator rand)`

Dibuje el árbol de sintaxis abstracta de la siguiente expresión:

`((lambda (x) (x y)) (a (lambda (x) (a (lambda (y) (a (b c)))))))`

