

# Primer examen parcial - Fundamentos de lenguajes de programación

Duración: 2 horas

Carlos Andres Delgado S, Ing \*

24 de Abril 2017

Nombre: \_\_\_\_\_

Código: \_\_\_\_\_

## 1. Especificación recursiva de datos [20 puntos]

- (10 puntos) Escriba la especificación inductiva de las listas de los números múltiplos de 7.
- (10 puntos) Escriba la especificación gramatical de las listas de parejas de números. Tenga en cuenta que **empty** es la lista vacía.

## 2. Abstracción de datos [50 puntos]

Dada la siguiente gramática:

```

expresion := identificador
           | var-exp (id)
           := numero
           | number-exp (dato)
           := "execute" "(" expresion " (" ")" "do"
              primitiva "end"
           | prim-exp (lid prim)
           := "if" "(" identificador "=" expresion "
              )" "then" expresión "else" expresion
           | if-exp (id exp1 trueExp falseExp)

identificador := simbolo+
numero := digito+
primitiva := "+"
           | sum-prim
           := "-"
           | minus-prim

```

En este caso los valores denotados son **símbolos**

- (10 puntos) Construya el árbol de sintaxis abstracta para la siguiente expresión:

```

if ( x = execute (1,2) do + end ) then x else
4

```

- (15 puntos) Utilizando **define-datatypes** defina la gramática de la expresión.

```

(define-datatype expresion expresion?
  ...
)
(define-datatype primitiva primitiva?
  ...
)

```

Para el caso de identificador y número utilice **symbol?** y **number?** respectivamente.

- (5 puntos) Utilizando la definición del **datatype** de expresión construya una expresión abstracta de **prim-exp**
- (20 puntos) Genere una función **evaluar-expresion** la cual recibe una expresión y un ambiente. Tenga en cuenta:

- Los valores expresados son **números**.
- En el caso del if **identificador = expresion**, ya existe **evaluar-condicional** que los recibe y retorna falso o verdadero.
- Ya existe un tipo de dato para ambiente
- Ya existe una función **apply-env**

## 3. Evaluación de expresiones [30 puntos]

Considere la siguiente expresión en el lenguaje visto en el curso (procedimientos), con ambiente inicial *env0* con identificadores (**a b c g**) y valores (**2 3 4**) (**closure'(x y z) +(x,\*(y z)) empty-env**)

```

let
  a = (g a b c)
  b = (g b c a)
  c = (g c a b)
in
  let
    f = proc(a,b) if a then b else *(a,b)
  in
    let
      a = (f a b)
      b = (f a c)
    in
      +(a,b,c)

```

- (5 puntos) Indique el valor de la expresión.
- (25 puntos) Dibuje los ambientes que se generan y muestre mediante líneas de que ambientes extienden.

\*carlos.andres.delgado@correounivalle.edu.co