Universidad del Valle

Primer examen parcial
Fundamentos de lenguajes de programación
Duración: 2 horas

Carlos Andres Delgado S, Ing *

17 de Noviembre 2016

Nombre:_____
Código:_____

1. Teoría [25 puntos]

1. (10 puntos) ¿En un lenguaje de programación, cómo se garantiza la consistencia de los procedimientos o funciones?

2. (15 puntos) ¿Cuales son las partes de un TAD (Tipo abstracto de dato)? Explicar qué utilidad tiene cada una.

2. Abstracción de datos [55 puntos]

Responda los siguiente puntos, basados en la siguiente gramática.

```
<expresion> := <identificador>
               id-exp(id)
            := fun (<identificador>+) <expresion>
                   end
               fun-exp(id ids body)
            := (<identificador> <expresion>*)
               ev-exp(id lexp)

<identificador> := simbolo+
```

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

1. (5 puntos) Escriba una expresión válida para el caso **fun-exp**

2. (10 puntos) Utilizando **define-datatypes** defina la expresión.

3. (15 puntos) Utilizando una representación basada en procedimientos, defina las expresiones se cumple

   - Si es un **id-exp**, devuelve un procedimiento, que retorna el id (que es un símbolo)
   - Si es una **fun-exp**, devuelve un procedimiento, que retorna una lista que contiene, una lista de identificadores y el símbolo 'exp
   - Si es un **ev-exp**, devuelve un procedimiento, que retorna una lista con los identificadores y un símbolo 'exp.

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

   - Utilizando las definiciones provistas por **define-datatypes** (realizadas en el punto 2), diseñe una función para evaluar la expresión:
     - (5 puntos) Si es un **id-exp**, invoque la función **buscar-variable** para hallar su valor. (Suponga que la función ya existe, esta recibe un argumento y retorna su valor)
     - (15 puntos) Si es un **fun-exp**, se retorne una lista cuyo primer elemento es el resultado de aplicar **buscar-variable** en los identificadores y el segundo elemento el resultado de la evaluación de la expresión (body). Para esto, apoyarse en la función **map**. Ejemplo: (map (lambda (x) (* x x)) (list 1 2 3 4)) retorna (list 1 4 9 16).
     - (5 puntos) Si es un **ev-exp**, se retorna una lista, cuyo primer elemento es la evaluación de los identificadores con **buscar-variable** y el segundo es la evaluación de la expresión

3. Evaluación de expresiones [20 puntos]

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

```
let
   x = 7
   y = 5
   z = 6
   f = proc(x,y,z) +(x,-(y,z))
   g = (f x y z)
   in
       let
         t = (f x y z)
         s = +(x,y)
         g = (f g x y)
         in
             (f t s g)
```

1. (5 puntos) Indique el valor de la expresión.

2. (15 puntos) Dibuje los ambientes que se generan y muestre mediante flechas de que ambientes extienden.