Primer examen opcional
Fundamentos de lenguajes de programación
Duración 2 horas

Carlos Andres Delgado S, Msc
carlos.andres.delgado@correounivalle.edu.co

02 de Marzo de 2021

Importante: Debe explicar el procedimiento realizado en cada uno de los puntos, no se considera válido únicamente mostrar la respuesta.

1. Reglas

- Entregue el interpretador modificado para el punto 1 y un archivo PDF o imágen con los ambientes para el punto 2.
- No envíe como solución enlaces externos para las capturas o archivos del exámen, no se valdrán.
- Las capturas de los puntos deben estar en buena calidad, si alguna no se entiende no se le valdrá el punto en cuestión.
- El examen opcional puede ser realizado en parejas, hacer **un sólo envío por pareja**
- El examen debe ser entregado en el formulario de google especificado por el docente. El opcional va hasta las 4:00pm, de allí se dan 20 minutos de gracia para entregarlo, es decir se recibe sin penalización hasta las 4:20:00pm
- Usted puede entregar después de las 4:20:00pm pero cada 5 minutos de retraso o fracción le descontaré 0.5 en la nota. Por ejemplo si entrega a las 4:31pm tendrá 11 minutos de retraso y 1.5 menos en la nota del examen.

2. Examen

1. (60 puntos) En este ejercicio se le propone extender el conjunto de valores expresados del lenguaje para manejar arboles binarios de búsqueda.

   Los arboles binarios de búsqueda tienen las siguientes propiedades

   a) Todo nodo tiene asociado un valor entero x

   b) Dado un nodo cualquiera con un valor x, el subarbol hijo izquierdo contiene valores **menores o iguales a x**

   c) Dado un nodo cualquiera con un valor x, el subarbol hijo derecho contiene valores **mayores o iguales a x**

   a) (35 puntos) Implemente la creación de arboles binarios de búsqueda **create-bintree(expresion)**.

   ```
   create-bintree([4, 3, 1, 7, 6])
   ```

   Los elementos se van insertando al árbol siguiente las reglas en el orden de la lista por lo que debe crear el siguiente árbol:

   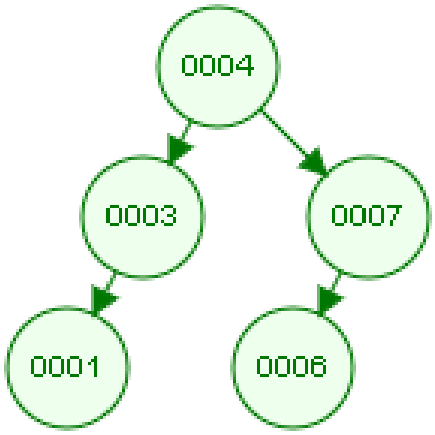

   Figura 1: Arbol binario de búsqueda

   En este punto debe:

   - Puede usar la implementación de listas que se hizo en el parcial ;), ya que estas se requieren para la creación del árbol.
   - Explicar con comentarios los cambios que hizo en la gramática
   - El árbol puede ser representado con árboles de sintaxis abstracta (struct)
   - Explicar con comentarios los cambios que hizo en la función evaluar-expresión y en las auxiliares que haya creado.
   - Deje comentado 3 ejemplos de árboles binarios de búsqueda creados con create-bintree.

   b) (25 puntos) Implemente **viewtree(expresion)** la cual recibe un árbol binario de búsqueda y retorna una representación en listas **(valor hijoizquierdo hijoderecho)**, por ejemplo:

```
let
  miarbol = create-bintree([4, 3, 1, 7, 6])
  in
  viewtree(miarbol)
```

Debe retornar

```
( 4
        (3 (1 'emptybintree 'emptybintree) 'emptybintree)
        (7 (6 'emprybintree 'emptybintree) 'emptybintree)
)
```

Entregue el interpretador de procedimientos recursivos con esta regla introducida, haga comentarios donde hizo los cambios para validar su trabajo.

2. (40 puntos) Dibuje los ambientes para la expresión: Considere la siguientes expresiones en el lenguaje visto en el curso (procedimientos), con ambiente inicial*env0* con identificadores **(x y z f)** y valores **(4 3 2(closure'(x y z) +(y,*(x,z)) empty-env))**

```
let
        x = 8
        y = 2
        z = (f x y z)
        in
        letrec
                f(a, b, c) =
                        if >(c,-(x,10)) then *(2, (f a b -(c,2)))
                           else 4
                m(c, d) = if >(c,0) then +(*(3,d), (m -(c,1) d))
                   else +(x,y)
        in
                let
                        x = (m 7 4)
                        in
                                (f (m 4 5) x z)
```

El valor de la expresión es 512