Primer examen opcional
Fundamentos de lenguajes de programación - 2 horas

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

03 de Marzo 2020

1. **(35 puntos)** Un árbol binario de búsqueda es aquel que tiene la siguiente gramática:

```
<arbol-binario> ::= (arbol-vacio)
                ::= (nodo) <numero> <arbol-binario> <arbol-binario>
```

Sin embargo, estos tienen una característica, dado un nodo cualquiera, todos los descendientes izquierdos son menores o iguales que el valor contenido en el nodo, y sus descendientes derechos son mayores, un ejemplo puede verse a continuación:

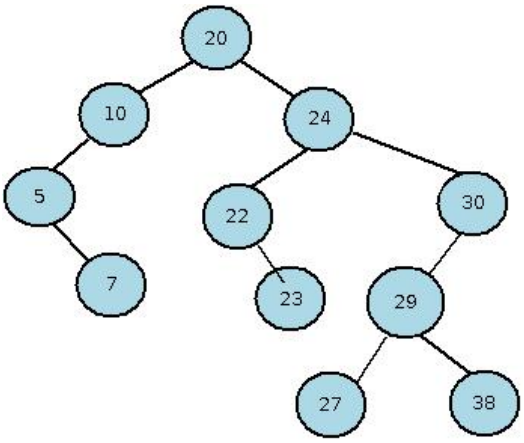

Figura 1: Ejemplo árbol binario de búsqueda

Debe diseñar un TAD (tipo de dato abstracto) usando procedimientos, de los que debe diseñar:

a) Procedimientos contructores **(5 puntos)**

b) Procedimientos observadores: predicados y extractores **(5 puntos)**

c) Diseñar la función insertar, la cual recibe un árbol binario de búsqueda y un número. Como restricción de la función el número es distinto a cualquiera del árbol. Este retorna el árbol con el valor insertado **(25 puntos)**.

A continuación en la figura 2 se muestra el árbol con el valor 8 insertado, como puede observar, dado que es mayor que el 7 se inserta a su derecha.

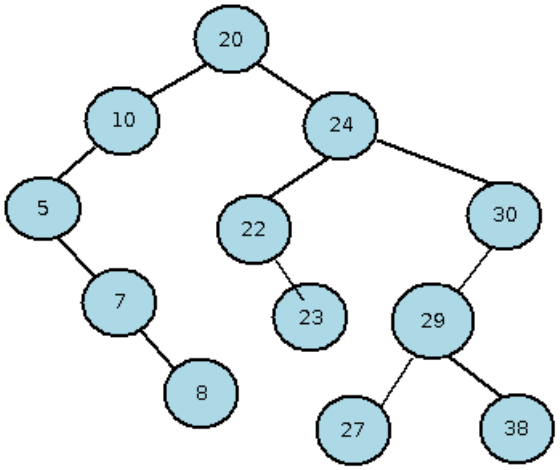

Figura 2: Ejemplo árbol binario de búsqueda con valor insertado 8

2. **(35 puntos)** Dada la siguiente expresión en lenguaje visto en el curso, suponiendo como ambiente inicial (x,y,s) (2,3,(closure '(x y) *(+(x,y),2) empty-env))

```
let
    f = proc(a b) proc(c d) -(-(c, d),-(b,a))
    g = proc(c d) proc(i j) proc(k l) if (i j) then +(c,d) else +(k,l)
    in
       let
         h = ((f x y) +(x,10) +(y,16))
         i = (g x y)
         in
           letrec
              j(m,n) = if m then (j add1(m) n)
                                 else (n  proc(a) -(a,3)  +(x,y))
           in
              let
                k = (j h i)
              in
                (k (s x y) x)
```

Dibuje los ambientes con los respectivos valores de variables (30 pts) e indique el resultado de la expresión (5 pts).

3. (30 puntos) Muestre como se agregan los arreglos al interpretador, bajo las siguientes condiciones

   - Los arreglo se declaran de esta forma [X1,X2,...,XN], donde Xi es un valor expresado
   - Los arreglos se acceden mediante la sentencia access, la cual recibe un identificador y una expresión que indica la posición del arreglo que se desea acceder.

     ```
     <expression> ::= "access" <identifier> "." <expression>
     ```

   a) (10 puntos) Escriba las sentencias que se agrega en la especificación gramatical

   b) (20 puntos) Escriba las clausulas que se agregan a las funciones eval-expression y apply-primitive. Desarrolle el código necesario para que los arreglos trabajen.