Parcial Número 2 de Leguajes II

Néstor Cataño
ncatano@puj.edu.co

Pontificia Universidad Javeriana

1. Representamos árboles binarios en OCaml con la ayuda del tipo `'a bin_tree` introducido a continuación. Adicionalmente, establecemos una relación entre árboles binarios y listas, a saber, la función `list_of_tree` convierte un árbol en una lista y la función `tree_of_list` una lista en un árbol. El operador `@` anexa dos listas en OCaml.

```
type 'a bin_tree =
  Empty
| Node of 'a bin_tree * 'a * 'a bin_tree;;

let rec list_of_tree bt =
 match bt with
  Empty -> []
| Node(left, y, right) ->
    (list_of_tree left) @ (y::(list_of_tree right));;

let rec insert x bt =
 match bt with
  Empty -> Node(Empty, x, Empty)
| Node(left, y, right) ->
    if x < y then Node(insert x left, y, right)
    else Node(left, y, insert x right);;

let rec tree_of_list l =
 match l with
  [] -> Empty
 | head::tail -> insert head (tree_of_list tail);;

let incognita x = list_of_tree (tree_of_list x);;
```

   (a) Cuál es el tipo más general asociado a la función `insert`?

   (b) Si `z = [4;2;3;1;5]`, a qué valor evalúa `(tree_of_list z)`?

   (c) Si `y = [1;3;5;2;4]`, a qué valor evalúa `(tree_of_list y)`?

   (d) A qué valor evalúa `(incognita z)`?

   (e) A qué valor evalúa `(incognita y)`?

2. El objetivo de este punto es definir una lista doblemente enlazada y circular en OCaml. Listas doblemente enlazadas son listas donde cada elemento conoce su predecesor y su antecesor. Las listas son circulares porque el último elemento de la lista apunta al primero y el primero apunta al último. Estamos interesados en definir una lista circular y doblemente enlazada que albergue datos de cualquier tipo. En la figura 1 se muestra un ejemplo de una lista circular doblemente enlazada que alberga datos de tipo entero. El nodo siguiente al nodo etiquetado con `8` es el nodo etiquetado con `5` y el nodo previo es el nodo etiquetado con `3`. La lista es circular, el nodo siguiente al nodo

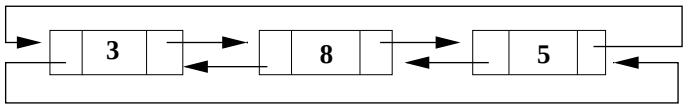

Figure 1: Listas Circulares Doblemente Enlazadas

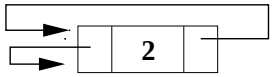

Figure 2: Operación Crear un Nodo

etiquetado con `5` es el nodo etiquetado con `3`, y el nodo previo al nodo etiquetado con `3` es el nodo etiquetado con `5`. En este punto usted deberá:

(a) Definir un tipo general que implemente listas doblemente enlazadas en OCaml usando al menos un registro con campos `mutable`.

(b) Escribir una función `create_node` que reciba un valor y devuelva una lista circular doblemente enlazada con un solo nodo, etiquetado con dicho valor. Por ejemplo, en el caso que la lista albergue valores de tipo entero, `create_node 2` creará la lista en la figura 2.

(c) Escribir una función `add` la cual anexe un nodo al final de una lista circular doblemente enlazada. Por ejemplo, si uno quiere adicionar un nodo etiquetado con el valor `2` al final de la lista mostrada en la figura 1 el resultado es la lista en la figura 3.

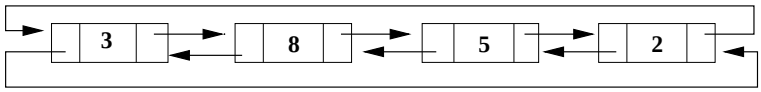

Figure 3: Adicionando un Nodo