Universidad Nacional Autónoma de México

Facultad de Ciencias

Lenguajes de Programación | 7098

Semanal 7 : | Combinador de punto fijo, continuaciones y recurrsión de cola

Sosa Romo Juan Mario | 320051926

Legorreta Esparragoza Juan Luis | 319317532

Erik Eduardo Gómez López | 320258211

15/10/24

1. Dada la siguiente expresión en MiniLisp:

```
(let (sum (lambda (n) (if0 n 0 (+ n (sum (- n 1))))))
    (sum 5))
```

   (a) Ejecutarla y explicar el resultado.

   (b) Modificarla usando el combinador de punto fijo Y, volver a ejecutarla y explicar el resultado.

2. Evaluar la siguiente expresión en Racket.

```
> (define c #f)
> (+ 1 (+ 2 (+ 3 (+ (let/cc k (set! c k) 4) 5))))
> (c 10)
```

   (a) Explicar su resultado:

   (b) Dar la continuación asociada a evaluar usando la notación $\lambda \uparrow$.

3. Realizar los siguientes ejercicios en Haskell:

   (a) Definir la función recurisva ocurrenciasElementos que toma como argumentos dos listas y devuelve una lista de parejas, en donde cada pareja contiene en su parte izquierda un elemento de la segunda lista y en su parte derechael número de veces que aparece dicho elemento en la primera lista. Por ejemplo:

```
> ocurrenciasElementos [1,3,6,2,4,7,3,9,7] [5,2,3]
[(5,0),(2,1),(3,2)]
```

   Yo llegue a la siguiente solución, que usa una función auxiliar para contar la cantidad de veces que aparece un elemento en una lista:

```
-- Ocurrencias.hs
cuentaElemento :: Int -> [Int] -> Int
cuentaElemento _ [] = 0
cuentaElemento x [y]
    | x == y    = 1
    | otherwise = 0
cuentaElemento x (y:ys)
    | x == y    = 1 + cuentaElemento x ys
    | otherwise = cuentaElemento x ys
```

```
ocurrenciasElementos :: [Int] -> [Int] -> [(Int, Int)]
ocurrenciasElementos _ [] = []
ocurrenciasElementos xs (y:ys) = (y, cuentaElemento y xs) :
                                                    ocurrenciasElementos xs ys
```

La idea es que la función ocurrenciasElementos recorre la lista de elementos a contar y por cada elemento llama a la función cuentaElemento que cuenta la cantidad de veces que aparece el elemento en la lista. La función cuentaElemento recorre la lista de elementos y por cada elemento compara si es igual al elemento a contar, si es así suma 1 al contador y sigue con el resto de la lista, si no es igual sigue con el resto de la lista. Cuando la lista esta vacía devuelve el contador.

(b) Mostrar los registros de activación generados por la función definida en el ejercicio anterior con la llamada ocurrenciasElementos [1,2,3] [1,2].

(c) Optimizar la función definida usando recursión de cola. Deben transformar todas las funciones auxiliares que utilicen.

(d) Mostrar los registros de activación generados por la versión de cola con la misma llamada.