Baltazar Coyotl Miguel Angel

Práctica número 2

Funciones recursivas

1° y 3° opción

28 de octubre del 2021

Grupo 3CM2

Materia: Paradigmas de Programación

Indice.-

Introducción.

Haskell es un lenguaje de programación declarativo donde el programador especifica lo que quiere hacer, en lugar de lidiar con el estado de los objetos. Es decir, las funciones estarían en un primer lugar y nos centraremos en expresiones que pueden ser asignadas a cualquier variable.

En la práctica número 2 se pide realizar el respectivo código en haskell por lo cual es importante que este lenguaje es puramente funcional y fue creado por Haskell Brooks Curry por ello su nombre. Los programas escritos en Haskell se representan siempre como funciones matemáticas, pero estas funciones nunca tienen efectos secundarios ni derivados. De este modo, cada función utilizada siempre devuelve el mismo resultado con la misma entrada, y el estado del programa nunca cambia. Por esto, el valor de una expresión o el resultado de una función dependen exclusivamente de los parámetros de entrada en el momento. En Haskell no pueden hacerse construcciones de lenguaje imperativo para programar una secuencia de declaraciones.

Desarrollo

En la practica 2 se nos pide realizar dos funciones las cuales en este caso van a ser la primera que es sobre calcular la serie geométrica y la tercera que es para insertar un átomo a la izquierda de la primera ocurrencia de un cierto átomo.

Para la **primera función** que en este caso se llamara **seriegeo** nos pide que calcule la siguiente serie para cualquier valor que se de

1+a+a^2+a^3+a^4+...+a^(n-1)+a^n

Por lo cual se define la función y se establece los parámetros que se le dará y el valor de retorno que tendrá

```
seriegeo :: Int -> Int -> Int
```

Consecuentemente se establece un caso base con el cual la función dejara su recursividad

```
seriegeo a 0 = 1
```

Finalmente se completa la función, sin olvidar su recursividad

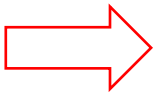
```
seriegeo a n = (a ^ n) + seriegeo a (n-1)
```

Se puede ver que está basado en esta serie:

1+a+a^2+a^3+a^4+...+a^(n-1)+a^n

Para que al compilarlo nos muetre lo siguiente:

```
putStrLn "enter value for a: "
input1 <- getLine
putStrLn "enter value for n: "
input2 <- getLine
let a = (read input1 :: Int)
let n = (read input2 :: Int)
putStrLn "The answer is: "
print (seriegeo a n)
```



```
enter value for a:
2
enter value for n:
10
The answer is:
2047
```

Para la siguiente funcion llamada **insertL** se define los parametros y el tipo de retorno

```
insertL :: Eq c => c -> c -> [c] -> [c]
```

Seguido a esto se establece un caso base

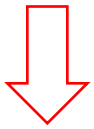
```
insertL _ _ [] = []
```

Finalmente se ocupan condicionales para que la funcion funcione como se pide, esto se puede ver a continuacion:

```
insertL new old (y:ys) =
    if y == old then new:y:ys
    else y: insertL new old ys
```

Al ejecutar el codigo se tiene lo siguiente:

```
putStrLn "Third exercise "
putStrLn "insertL 100 6 [ 1, 2, 3, 4, 5, 6, 7] "
print(insertL 100 6 [ 1, 2, 3, 4, 5, 6, 7])
```



```
Third exercise
insertL 100 6 [ 1, 2, 3, 4, 5, 6, 7]
[1,2,3,4,5,100,6,7]
```

Conclusión

Finalmente se logró identificar y aplicar lo aprendido sobre Haskell, utilizándolo para realizar ciertas funciones que son requeridas, así como se hizo con la primera función en la cual se resuelve la suma de unas serie que si se hiciera a mano el tiempo para hacerlo seria grande, sin embargo aplicando lo aprendido sobre haskell y utilizando la programación es que se puede tener ese resultado más rápido.

Bibliografía

(2019). Obtenido de EcuRed:

https://www.ecured.cu/LISP

Digital Guide. (9 de octubre de 2020). Obtenido de

https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/que-es-haskell/