Recursi?n_Backtracking

June 22, 2018

1 Resumen IIC-1103

Por: Alfonso Irarrázaval, Daniel Leal, Nicolás Quiroz.

---

2 Recursión

Se habla de recursion, o mas bien, de un elemento recursivo, cuando la definicion de este depende de su misma definicion.

Algunos de ustedes ya habran visto ejemplos de recursividad en otros ramos o materias, pero no necesariamente lo han visto por definicion como recursivos

Algunos de los ejemplos mas comunes son:

- Numeros Factoriales
- Series recursivas (eg. Serie de Fibonacci)

En el caso de los numeros factoriales:
n! = n * (n - 1)!
Pero a su vez (n - 1)! = (n - 1) * (n - 2)!, entonces reemplazando:
n! = n * (n - 1) * (n - 2)!
n! = n * (n - 1) * (n - 2) * (n - 3)!
y en general:
n! = n * (n - 1) * (n - 2) * (n - 3) * . . . * 2 * 1

Se ve de manera directa que para obtener n! debemos saber de antemano el valor de (n - 1)!, el cual a su vez depende de (n - 2)!, el cual a su vez depende de (n - 3)! y asi sucesivamente, hasta que llegamos al 0! (recordar que 0! = 1).

Aqui hemos llegado a una parte importante de la recursividad: el **caso base**

El **caso base** se refiere a ese caso, el cual depende completamente de lo que se busca (puede ser un numero, el largo de una lista o string, etc..), en el que la recursion se detiene y comienza a "devolverse" (mas adelante veremos a que se refiere ese término)

Volviendo al ejemplo de los numeros factoriales, por definicion estos residen en los numeros cardinales, por lo que el minimo valor al que se puede llegar corresponde al 0, después estamos en un rango donde no estan definidos por lo cual no podemos entrar, en otras palabras, nuestro caso base es el 0 (notese que tambien podria ser 1, ya que 1! = 0! = 1).

Todo esto se traduce en la siguiente funcion para los numeros factoriales:

rec

```
In [4]: #Partimos definiendo la funcion como cualquier otra, donde recibirá
        # el numero del cual queremos obtener el factorial
        def factorial(n):
            # Lo primero que previsamos es si el numero ingresado es el CASO BASE
            # que en este caso seria 0
            if n == 0:
                #Cuando el numero es 0, retornamos el caso base
                print("llegamos al caso base, nos empezamos a devolver!")
                return 1
            #en cualquier otro caso, queremos obtener n * (n - 1)!
            else:
                print("{}! = {} * {}!".format(n, n, n - 1))
                factorial_siguiente = factorial(n - 1)
                print("{}! = {} * {}".format(n, n, factorial_siguiente))
                return n * factorial_siguiente

        factorial(5)
```

```
5! = 5 * 4!
4! = 4 * 3!
3! = 3 * 2!
2! = 2 * 1!
1! = 1 * 0!
```