

Recursividad

Cuando se habla de recursividad se hace referencia a problemas en los cuales, dicho problema es parte de la solución. Se trata de un modelo matemático cuya definición incluye a sí mismo “se autodefine”; es decir que se llama a sí misma repetidamente hasta que satisface una determinada condición para evitar caer en un bucle infinito, y donde cada resultado depende de la resolución de la siguiente o anterior repetición, hasta alcanzar dicha condición de fin.

La recursividad la vamos a definir como una función, va a estar dentro de una función.

Entonces, este modelo recursivo tiene dos características fundamentales para ser denominado como tal (función):

1. Debe tener al menos una condición clara de fin o caso base del problema (que en algún momento se termine), puede tener más de una de ser necesario.
2. Debe llamarse a sí mismo, es decir que el modelo o solución sea parte de su propia definición. Esta llamada puede ser única, si se llama una sola vez dentro de la función, o múltiple, si se realizan más de una llamada.

#La recursividad es bueno para problemas recursivos, pero no todo se resuelve con recursividad, hay que tomarlo como una herramienta. Hay que saber **cuándo** y para que **usarlo**.



Figura 1. Definición gráfica de recursividad

#La recursividad vienen como para suplantar a los ciclos, en vez de tener un ciclo debemos de tener una llamada recursiva (una invocación de la función)

A partir de esta definición, se puede decir que la recursividad es una técnica muy útil para desarrollar algoritmos recursivos que den solución a distintos problemas. Este modelo recursivo se lo debe pensar como una función recursiva que servirá para dar solución a ciertos problemas –estos son casos particulares donde la naturaleza del problema es recursiva–. No cualquier problema puede resolverse de forma recursiva, aunque muchas veces puede implementarse una solución recursiva para problemas que no son de esta naturaleza. Es importante aclarar que dentro de una función o algoritmo recursivo no deben existir ciclo o bucles, salvo algunos casos muy particulares para resolver alguna operación transversal a la función (no para la esencia de resolver el problema).

Se puede clasificar la recursividad de acuerdo a su tipo, pueden ser algoritmos recursivos directos o indirectos, los primeros se llaman a sí mismos mientras que los segundos llaman a un algoritmo

(intermedio) y este llama al algoritmo inicial para generar la recursividad. Otra manera de clasificarla es respecto a la finalización de los algoritmos si son finales o no finales, es decir, si al finalizar

las llamadas recursivas no queda más nada por hacer o aún quedan instrucciones por ejecutar.

Para comenzar, podemos observar un ejemplo sencillo: el problema del factorial, si se quiere calcular el factorial de un número n . La solución iterativa sería la siguiente (si $n=5$):

$5! = 5 * 4 * 3 * 2 * 1$ Pero si se lo piensa de forma recursiva, es decir, si se descompone el problema inicial en pequeños subproblemas y se resuelve una parte del problema en cada llamada (que quedará pendiente hasta que se alcance el caso base del problema), entonces se podría resolver de la siguiente manera: $5! = 5 * 4! \rightarrow$ Llamada recursiva De esta forma, la solución al problema anterior quedaría así:

Problema	Solución parcial	Solución
$5!$	$5 * 4!$ (pendiente en espera de solución de $4!$)	Llamada recursiva $4!$
$4!$	$4 * 3!$ (pendiente en espera de solución de $3!$)	Llamada recursiva $3!$
$3!$	$3 * 2!$ (pendiente en espera de solución de $2!$)	Llamada recursiva $2!$
$2!$	$2 * 1!$ (pendiente en espera de solución de $1!$)	Llamada recursiva $1!$
$1!$	$1 * 0!$ (pendiente en espera de solución de $0!$)	Llamada recursiva $0!$
$0!$	$0! = 1$ (condición de fin o caso base)	1

Una vez que se alcanza la condición de fin, se empiezan a resolver las llamadas recursivas previas que quedaron pendientes.

Problema	Solución parcial	Solución
$0!$	$0!$	1 (condición de fin o caso base)
$1!$	$1 * 0!$	$1 * 0! \rightarrow 1 * 1 = 1$ (utilizando resultado de $0!$)
$2!$	$2 * 1!$	$2 * 1! \rightarrow 2 * 1 = 2$ (utilizando resultado de $1!$)
$3!$	$3 * 2!$	$3 * 2! \rightarrow 3 * 2 = 6$ (utilizando resultado de $2!$)
$4!$	$4 * 3!$	$4 * 3! \rightarrow 4 * 6 = 24$ (utilizando resultado de $3!$)
$5!$	$5 * 4!$	$5 * 4! \rightarrow 5 * 24 = 120$ (utilizando resultado de $4!$)