RECURSIVIDAD

Recursividad paper

Juan Estaban Corrales Gallego

*Facultad de Ingenierías. Universidad Tecnológica de Pereira*

Correo: j.corrales@utp.edu.co

***Resumen*— La recursión o recursividad es la forma de solucionar y optimizar soluciones de una forma fácil y simple.**

**La recursión es una palabra muy mencionada en el mundo de la tecnología ya que es usada para todo tipo de algoritmos en el mundo de la computación, por lo que es importante y elemental con una amplia ayuda.**

***Abstract—* The recursion is the way to solve and optimize solutions in an easy and simple way. The recursion is a word very mentioned in the world of the technology since it is used for all types of algorithms in the world of the computation, reason why it is important and elementary with an ample aid**

***Palabras clave— “algoritmo” ,“problema”, “recursión”***

1. INTRODUCCIÓN

La resolución de problemas es la estructura fundamental de la humanidad y sus componentes es parte de nuestra naturaleza, y a la vez de que encontramos nuevas soluciones encontramos nuevos problemas, desde lo más mínimo hasta lo complejo que pone a duda nuestro verdadero propósito, y de todo lo recopilado, creamos mecanismos para facilitar nuestro camino de las soluciones, como lo es la automatización, también la recursividad, elementos que nos han expandido nuestro rango de conocimiento y que nos han optimizado gran cantidad de tiempo y evitado grandes errores. Aquí abarcaremos la recursividad desde un punto de vista como una ayuda fundamental.

1. CONTENIDO

La recursividad es la forma en que un problema se reparte en sub-parte totalitarias llamándose bajo su propia definición. Ésta toma el concepto de la sinergia y lo aplica para cada subelemento que conforme el elemento total, donde cada pedazo es totalitario y, aunque funcione de manera autónoma con sus respectivas variables, complementa para el problema en total, y éste a la vez funciona llamándose a sí mismo dentro de su definición. ***1*** Su conjunto de nombre ***n*** contiene parte menores que ***n (<n)***, n (***<= n), n (=n)*** y se le impone un límite condicional para finalizar su ejecución, de lo contrario será de fin indeterminado o infinito.

1. Informática

La aplicabilidad de la recursividad en la informática ha sido fundamental, gracias a su metodología simple de dividir el problema en pequeñas partes y realizar un bucle que llame a la función, e imponer un límite o una condicional; la composición de los problemas en pequeñas partes y hacer un bucle que permita llamar la función y dar el valor de su variable hasta el infinito o el límite determinado.

Entre los ejemplos más destacables, está el buscador de artículos relaciones del motor de búsqueda de Google, que utiliza la recursión para mostrar artículos que tengan relación o similitud a la sugerencia escrita por el usuario, la cual no para de buscar hasta llegar a un límite impuesto.

1. Características

“Un problema que pueda ser definido en función de su tamaño, sea este n, pueda ser dividido en instancias más pequeñas (< n), n (***<= n), n (=n)*** del mismo problema y se conozca la solución explícita a las instancias más simples, lo que se conoce como casos base, se puede aplicar inducción sobre las llamadas más pequeñas y suponer que estas quedan resueltas.

Para dar solución a un problema a partir del empleo de la recursividad, es necesario asegurar que cada llamada recursiva debe estar definida sobre un problema menos complejo que el que dio lugar a la llamada, y que debe existir al menos un caso base al que se garantice la llegada del problema en un momento determinado. Si no se cumpliera esta condición se generaría una secuencia infinita de llamadas y el algoritmo que intenta dar solución al problema no terminaría nunca.”

1. Manejo y ejemplos

La recursión se puede manifestar de varias formas, inclusive en la vida cotidiana o en problemas matemáticos de alta complejidad.

Un ejemplo de la recursividad en un campo útil como las matemáticas es la ecuación de la factorial de un número n ***(n!)*** y que se define de la siguiente manera:

;ejercicio 1

;diga determinadas veces una palabra

(define (ejercicio1 n x )

(if (> n x)

(begin

(displayln "palabras")

(ejercicio1 n (+ x 1))

)

)

(void)

)

;llamado

(displayln "por favor digite cuantas veces desea la palabra")

(define n (read))

(displayln "por favor digite cuantas veces desea la palabra")

(define x (read))

(ejercicio1 n x)

;--------------------------------------------------------

;ejercicio2

(define (ejercicio2 n i)

(if (>= n i)

(begin

(displayln "la sumatoria es ")

(displayln (+ i i))

(ejercicio2 n (+ i 1))

(

)

(void)

)

)

)

;llamado

(displayln "por favor defina el numero que desea ")

(ejercicio2 (read) 0)

1. REFERENCIAS

***[1:*** <https://es.wikipedia.org/wiki/Recursi%C3%B3n>

1. ***:*** <https://www.ecured.cu/Recursividad>
2. ***:*** DrRacket ejemplo propio

: