

Recursividad en ciencias de la computación

Recursion in computer science

Valentina Varela Alzate

Risaralda, Universidad tecnológica de Pereira, Pereira, Colombia

Valentina.varela@utp.edu.co

Resumen—La recursividad no es una estructura de datos, sino que es una técnica de programación que nos permite que un bloque de instrucciones se ejecute n veces. Reemplaza en ocasiones a estructuras repetitivas. Este concepto será de gran utilidad para el capítulo de la estructura de datos tipo árbol.

La recursividad es un concepto difícil de entender en principio, pero luego de analizar diferentes problemas aparecen puntos comunes.

Cuando un método se llama a sí mismo, se asigna espacio en la pila para las nuevas variables locales y parámetros.

Al volver de una llamada recursiva, se recuperan de la pila las variables locales y los parámetros antiguos y la ejecución se reanuda en el punto de la llamada al método.

Palabras clave— Recursividad, bucles, programación funcional, ciclos.

Abstract— The recursion is not a data structure, but it is a programming technique that allows us to a block of statements to run n times. Replaces sometimes repetitive structures. This concept will be very useful for the chapter of the tree data structure. Recursion is a difficult concept in principle, but after analyzing different problems appear common points.

Key Word — Recursion, loops, functional programming, cycles.

I. INTRODUCCIÓN

Recursión es una de las ideas centrales de la ciencia de la computación. Resolver un problema mediante recursión significa que la solución depende de las soluciones de pequeñas instancias del mismo problema.

Un programa recursivo está basado en un algoritmo recursivo. Los programas recurrentes se invocan o llaman a sí mismos ya que están definidos en términos de sí mismos, dicha llamada se conoce como llamada recursiva. Crear una subrutina recursiva requiere principalmente la definición de un "caso base" y que con cada llamada recursiva el problema se reduzca de forma que al final llegue al caso base.

En los programas recursivos cada vez que un subprograma llama otro debe guardarse una indicación del punto donde se debe retornar al finalizar la ejecución del programa. Las

llamadas a procedimientos pueden encadenarse arbitrariamente.

Hay una estructura de datos donde se almacenan sucesivos puntos de retorno; La estructura irá creciendo con cada nueva llamada y decreciendo al terminar la ejecución de cada subprograma.

Esta estructura se comporta como una pila, el tope de la pila es el punto donde debe retornarse el control tras la terminación del subprograma corriente. Por lo tanto, si el subprograma corriente llama a otro, el correspondiente punto de retorno debe colocarse como nuevo tope de la pila y al finalizar un subprograma, se usa el tope como dirección de retorno y se lo remueve de la pila. La pila es una estructura lineal dinámica donde se agregan y quitan elementos sólo en uno de sus extremos.

Cuando se hace un programa recursivo se debe tener cuidado de no caer en la circularidad, en caso de hacerlo el programa terminara con un error de ejecución: no hay más memoria; lo que hace el programa es que la pila crece infinitamente, pero la memoria de la máquina es finita, por lo tanto, en algún momento no hay más memoria.

II. CONTENIDO

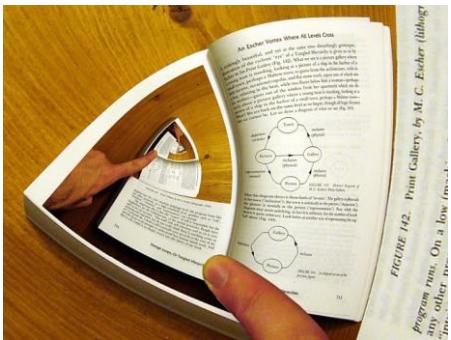
Todos los algoritmos recursivos deben obedecer una serie de leyes:

- Un algoritmo recursivo debe tener un caso base: un caso base es la condición que permite que el algoritmo detenga la recursividad
- Un algoritmo recursivo debe cambiar su estado y moverse hacia el caso base: se debe organizar un cambio de estado que obligue el algoritmo hacia el caso base. Para esto se modifican algunos datos que el algoritmo está usando. Por lo general, los datos que representan un problema se hacen cada vez más pequeños.
- Un algoritmo recursivo debe llamarse a sí mismo, recursivamente: la definición misma de recursividad,

un problema que se resuelva llamando a la función, que a su vez se llame a si misma para completar el proceso.

```
def factorial(n)
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)
```

Mas allá de hablar de la recursividad, también conocida como recursión, en el campo de la computación se puede tratar desde el campo artístico, la recursividad aplicada a imágenes.



La recursividad es la forma en la cual se especifica un proceso basado en su propia definición. Aunque es más difícil de adaptar al mundo de la imagen o el diseño gráfico, se pueden encontrar diversidades de obras que han sido hechas con esta técnica. Un ejemplo de esto podría ser una cuadrícula bidimensional, es una imagen recursiva que no tiene necesidad de recurrir a las matemáticas. Cada cuadrado está formado por cuadrados de menor tamaño que a su vez están formados por otros cuadrados más pequeños y así una y otra vez. No importa realmente que sean cuadrados, triángulos equiláteros, rectángulos áureos, curvas de Koch, ... No importa tanto la forma, sino que la imagen se utiliza a sí misma hasta el infinito.

El arte nos ha demostrado que la perspectiva cambia la realidad. Forma, tamaño y dimensión se alteran al tratar de expresar de otra manera la existencia. La literatura, la pintura y la fotografía hacen uso de diferentes tropos para explicar los misterios que impone el mundo real y así jugar con las reglas de la naturaleza para crear su propia visión del espacio, tiempo y sentido.

Por otro lado, las matemáticas y las ciencias exactas han demostrado que siempre existe una explicación lógica, que todos aquellos juegos mentales que suceden en la capacidad de cada persona tienen una explicación completamente justificable. Las herramientas de la geometría fractal, por ejemplo, se han convertido en elementos insustituibles en el trabajo de muchos físicos, químicos, biólogos, fisiólogos, economistas, etc. pues les han permitido reformular viejos problemas en términos novedosos, y de esta forma tratan problemas complejos de forma muy simple.

Las formas fractales, que durante mucho tiempo se consideraron meras "monstruosidades" geométricas e inaplicables divertimentos matemáticos, se encuentran en fenómenos y estructuras tan variadas como la distribución de las estrellas del Universo, la ramificación alveolar en los pulmones, la frontera difusa de una nube, las fluctuaciones de precios en un mercado y en la frecuencia de repetición de las palabras de los textos

Entre los más conocidos exponentes de la recursividad que se han destacado a lo largo del tiempo en distintos campos se pueden encontrar: Escher, Gödel y Bach.

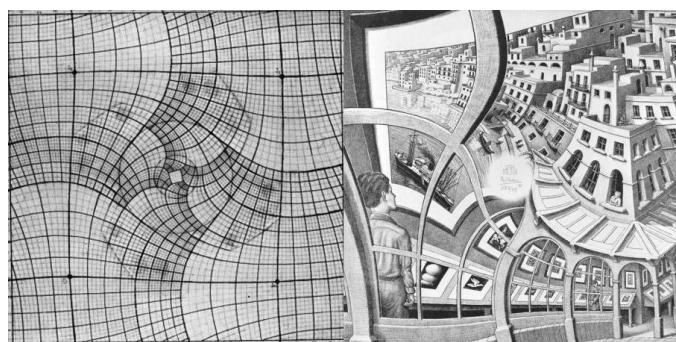
1) Maurits Cornelis Escher (1898-1972). Artista que nació en Holanda en 1898 y en su juventud estudió en la Escuela de Arquitectura y Diseño Ornamental de la ciudad de Haarlem. A lo largo de su vida viajó por toda Europa y residió en Italia, Suiza y Bélgica. Murió en su país natal en 1972, momento en el que el reconocimiento de su obra comenzaba a adquirir carácter global. Hoy en día su obra es mundialmente conocida y aparece en multitud de lugares, desde camisetas a portadas de libros científicos.

El arte de Escher entra de lleno en el concepto de arte matemático; un artista figurativo que sepa algo de matemáticas puede hacer una composición sobre un tema matemático de la misma manera que los artistas del Renacimiento lo hicieron con los temas religiosos o los artistas rusos con los temas políticos. Sus obras cuelgan en los despachos de matemáticos y científicos de todo el mundo.

“Con frecuencia me siento más próximo a los matemáticos que a mis colegas los artistas”; “Mis ideas están basadas en mi asombro y admiración por las leyes contenidas en el mundo que nos rodea. Quien se maravilla de algo, toma conciencia de algo maravilloso.”

M.C. Escher

Sus litografías exploraron diferentes técnicas especialmente enfocadas a jugar con el espacio. De esta forma en sus obras se destacan las teselas, retículas, imágenes recursivas, estudios de las superficies y la partición regular del plano.



Actualmente es posible reproducir con facilidad el efecto Escher con ayuda de algoritmos matemáticos, cálculos fracturales y la aplicación Matemática.

2). Kurt Gödel (1906-1978). Fue un lógico, matemático y filósofo austriaco-estadounidense. Reconocido como uno de los más importantes lógicos de todos los tiempos. El trabajo de Gödel ha tenido un impacto inmenso en el pensamiento científico y filosófico del siglo XX.

Intentó emplear la lógica y la teoría de conjuntos para comprender los fundamentos de la matemática. A Gödel se le conoce mejor por sus dos teoremas de la incompletitud, publicados en 1931.

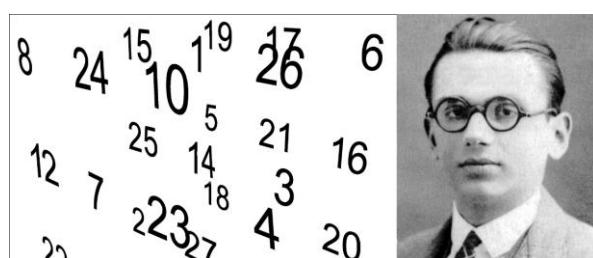
- **Primer teorema de incompletitud de Gödel:** Cualquier teoría aritmética recursiva que sea consistente es incompleta. Gödel desarrolló un método para codificar signos y fórmulas mediante números, llamado numeración de Gödel usando esa numeración, es posible traducir las propiedades de una teoría formal T, tales como estos signos constituyen una fórmula o estas fórmulas no son una demostración en T», a propiedades aritméticas de dichos números.

En particular, la sentencia de Gödel G es una fórmula aritmética cuyo significado es que no existe una demostración de G en la teoría T, o, en otras palabras, que no es demostrable en la teoría T.

- **Segundo teorema de incompletitud de Gödel:** En toda teoría aritmética recursiva consistente T, la fórmula Consistente T no es un teorema.

La demostración del segundo teorema requiere traducir el primero a una fórmula. La fórmula que afirma la consistencia de T es Consis T, mientras que la fórmula que afirma la indemostrabilidad de G es la propia G. La fórmula que traduce el primer teorema es Consis T \Rightarrow G, donde \Rightarrow significa implicación.

Gödel demostró que esta fórmula es un teorema,⁶ y que por lo tanto Consis T no es un teorema: si lo fuera, de las reglas básicas de T como teoría formal se deduciría que G es demostrable, en contradicción con el enunciado del primer teorema de incompletitud.



3) Johann Sebastian Bach (1685-1750). Sus obras son consideradas piezas artísticas debido a su profundidad intelectual, su perfección técnica y su belleza melódica. Además, es reconocido como el último gran maestro del arte de contrapunto, técnica de composición musical que relaciona dos voces independientes para crear un equilibrio armónico.

El alemán compuso sus piezas musicales pensando en Fuga y Canon, que son elementos musicales que usan la repetición de melodías en diferente escala o velocidad con o sin contrapunto, pero que no precisamente son la misma melodía. El uso del canon y la fuga es similar a la naturaleza fractal, pero este término no se conocía en 1685 sino hasta 1975 cuando fue acuñado por el matemático Benoît Mandelbrot.

En piezas autosemejantes de Bach los mismos motivos son repetidos una y otra vez con distintas variaciones dentro de una región mayor de la pieza. Así, por ejemplo, distintas voces se repiten al doble de velocidad la melodía de la voz principal.

A partir de la música de Bach se descubrió que repetir un proceso con el objetivo de alcanzar una meta deseada, en conjunto con las matemáticas, resultan elementos que se aplican como una extensión de la composición musical.

Así, los fractales proveen inesperadas conexiones entre las artes musicales y muchos procesos naturales, ya que mezclan cualidades determinantes y probables para producir naturalmente un agradable balance entre lo que se espera y la novedad, pues al escuchar la repetición de cada una de las notas por separado, la pieza resultaría un caos, al igual que una imagen que no entra en armonía con todos los elementos.



III. CONCLUSIONES

En el libro Gödel, Escher, Bach: An Eternal Golden Braid (más conocido como GEB), su teoría explica que cuando un sistema entra en un bucle extraño, especialmente de tipo recursivo y autorreferente (a lo Gödeliano), es cuando toma conciencia de sí mismo y surge el verdadero reconocimiento del yo, del uno mismo o lo que podría definirse como conciencia.

Esto, que de algún modo ya existe en el cerebro de los seres humanos y otros animales, podría del mismo modo surgir en máquinas inteligentes algún día, si se construyera un sistema suficientemente complejo como para que entrara en uno de esos bucles recursivos autorreferentes e infinitos.

” ¿Qué es la recursividad? Incrustaciones y variaciones de incrustaciones. El concepto es muy amplio (relatos dentro de relatos, películas dentro de películas, muñecas rusas dentro de muñecas rusas, o comentarios entre paréntesis dentro de comentarios entre paréntesis, son solamente algunos de los encantos de la recursividad”

Douglas R. Hofstadter; Gödel, Escher, Bach: un Eterno y Grácil Bucle

WEBGRAFIA

<http://www.tutorialesprogramacionya.com/javaya/detalleconcepto.php?codigo=123&punto=&inicio=>

<https://www.fing.edu.uy/inco/cursos/fpr/wiki/index.php/Recursividad>

<http://interactivepython.org/runestone/static/pythoned/Recursion/LasTresLeyesDeLaRecursividad.html>

<https://www.educ.ar/recursos/126158/escher-el-cruce-maravilloso-del-arte-con-el-hechizo-de-la-matematica>

<https://mimoriarty.wordpress.com/2011/04/11/imagenes-recursivas-efectos-droste-y-escher/>

https://www.ecured.cu/Kurt_G%C3%B6del

https://es.wikipedia.org/wiki/Teoremas_de_incompletitud_de_G%C3%B6del

<https://culturacolectiva.com/musica/los-fractales-en-la-musica-de-sebastian-bach>

<https://fjotap.wordpress.com/2008/10/10/recursividad/>

<https://www.microsiervos.com/archivo/ordenadores/recursividad-infinita.html>