

**INSTITUTO TECNOLÓGICO DE
IZTAPALAPA**

INGENIERIA EN SISTEMAS COMPUTACIONALES

TEMA:

General recursive functions

Materia:

INGENIERÍA DE SOFTWARE

Presenta:

**Martinez Muñoz Juan Daniel
171080100**

**Jiménez Pichardo Francisco Joel
161080157**

**Paredes Pegueros Juan Antonio
171080125**

**Alvarez Victoria Adrian
171080174**



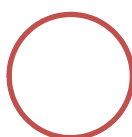
Resumen

La noción de recursión se emplea en diferentes sentidos, adquiriendo una variedad de significados. Así, en unos casos se usa para caracterizar una regla esencial que constituye un modo de definición en un sistema. Este sentido tiene su origen en la Lógica Matemática y la Teoría de la Computabilidad. En otros casos se aplica para indicar la organización interna de una estructura, tal como sucede en la Ciencia Cognitiva y la Ciencia de la Computación, al tiempo que también se emplea en el sentido anterior en estas mismas disciplinas.

En lógicas matemáticas e informática una función recursiva parcial o general, es una función parcial de números naturales a números naturales que son computables en un sentido intuitivo. En la teoría de la computabilidad son precisamente las funciones que pueden ser calculadas por la máquina de Turing (la máquina de Turing es un dispositivo que manipula símbolos sobre una tira de cinta de acuerdo con una tabla de reglas, a pesar de su simplicidad la máquina de Turing puede ser adaptada para simular la lógica de cualquier algoritmo de computador) las funciones recursivas están estrechamente relacionadas con las funciones de recursividad primitiva (permite definir una clase de funciones que forman un importante paso en la formalización de la noción de computabilidad) sin embargo, no todas las funciones recursivas totales son funciones recursivas primitivas. El objetivo de este trabajo es mostrar en cada caso las diferencias en el uso de ambos sentidos.

Palabras clave:

Recursión, Sistema, Origen, Logica, Turing, Mostrar, Diferencias, Estrechamente, Algoritmos.





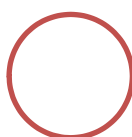
Abstract

The notion of recursion is used in different senses, acquiring a variety of meanings. Thus, in some cases it is used to characterize an essential rule that constitutes a mode of definition in a system. This sense has its origin in Mathematical Logic and Computability Theory. In other cases it is applied to indicate the internal organization of a structure, such as in Cognitive Science and Computer Science, while it is also used in the previous sense in these same disciplines.

In mathematical logic and computer science a partial or general recursive function is a partial function from natural numbers to natural numbers that are computable in an intuitive sense. In the theory of computability, these are precisely the functions that can be calculated by the Turing machine (the Turing machine is a device that manipulates symbols on a strip of tape according to a table of rules, despite its simplicity the machine Turing can be adapted to simulate the logic of any computer algorithm) recursive functions are closely related to primitive recursion functions (it allows defining a class of functions that form an important step in the formalization of the notion of computability) however , not all total recursive functions are primitive recursive functions. The objective of this work is to show in each case the differences in the use of both senses.

Keywords:

Recursion, System, Origin, Logic, Turing, Show, Differences, Tightly, Algorithms.





ÍNDICE GENERAL

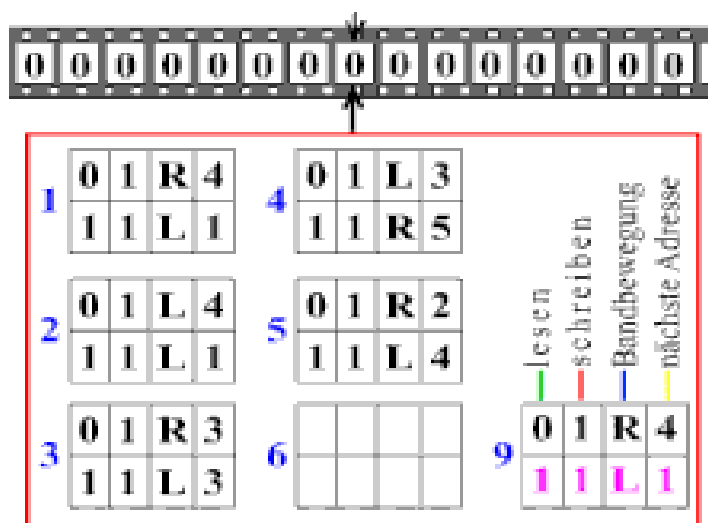
Portada	1
resumen	2
Adstract	3
Índice General	4
Índice	5
Introducción	7
Objetivos	8
Justificación	9
Marco teórico	10
Metodología de trabajo	14
Desarrollo e implementación	15
Conclusiones	16
Fuentes	17
Anexos	18



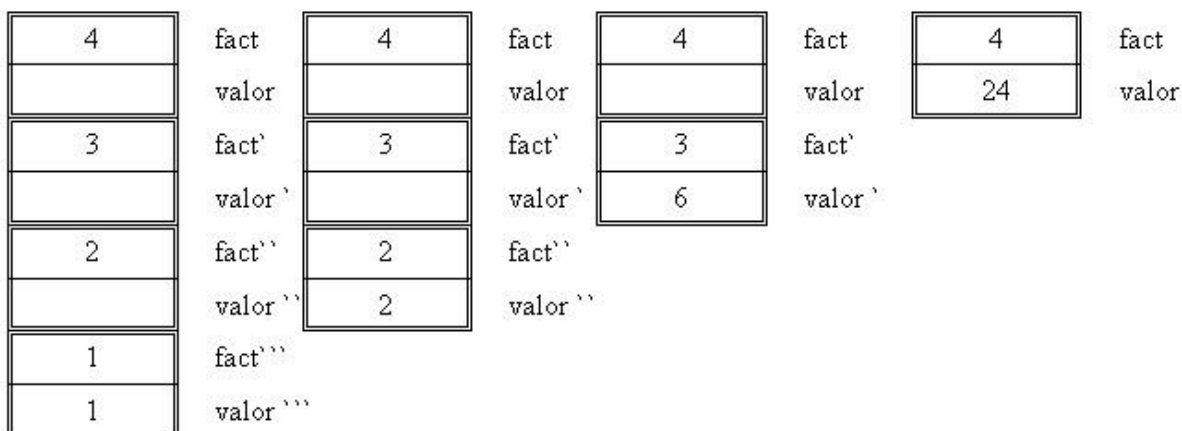


ÍNDICE

tabla de la máquina de turing

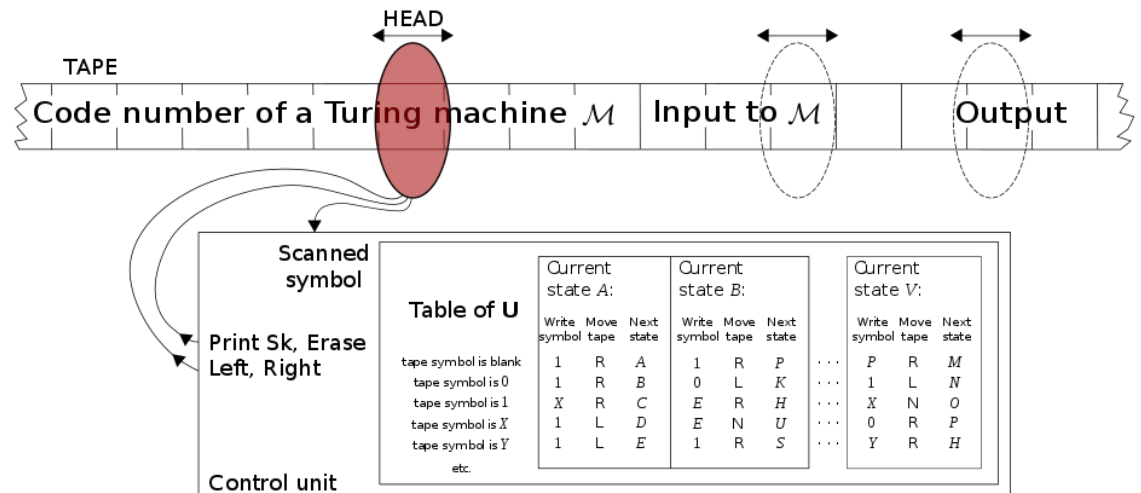
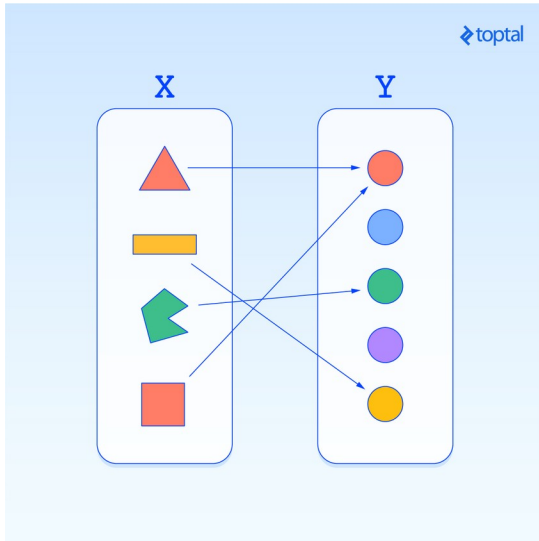


función recursiva parcial

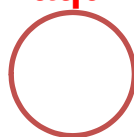




teoría de la computabilidad



universal máquina de turing





INTRODUCCIÓN

El sentido original de recursión está claramente establecido dentro de la teoría de la computabilidad y de la lógica matemática si bien dentro de estos dominios parece haberse empleado en dos sentidos diferentes. Uno de ellos se refiere a la definición por recursión (o definición recursiva) y el otro a la noción de computabilidad (y lo mismo vale para los adjetivos 'recursivo' y 'computable'. El primero de los sentidos mencionados, el de la definición por recursión, hace referencia a un método usado para definir funciones y predicados. Su origen se remonta al siglo XIX.

Las funciones recursivas son funciones parciales que toman tuplas finitas de números naturales y devuelven un solo número natural. son clases más pequeñas de funciones parciales que incluye las funciones iniciales y está cerrada bajo composición, recursividad primitiva y el operador.

La clase más pequeña de funciones, incluidas las funciones iniciales y cerrada en composición y recursividad primitiva (es decir, sin minimización) es la clase de funciones recursivas primitivas. Si bien todas las funciones recursivas primitivas son totales, esto no es cierto para las funciones recursivas parciales; por ejemplo, la minimización de la función sucesora no está definida.

Las funciones recursivas primitivas son un subconjunto de las funciones recursivas totales, que son un subconjunto de las funciones recursivas parciales.





OBJETIVO GENERAL

El objetivo principal del presente documento es dar a conocer los pasos a seguir para la solución de problemas relacionados con los números naturales sin la utilización de la tan conocida máquina de Turín y en su lugar dar una explicación lo más detallada posible sobre el método de funciones recursivas generales así como su funcionamiento.

OBJECTIVOS ESPECÍFICOS

- 1.-Dar a conocer las ideas principales sobre el tema de Función recursiva general y posterior a ello proponer un problema y dar una solución al mismo y buscar un programa donde se pueda expresar de una manera más clara del funcionamiento del presente tema de investigación
- 2.-Dar a conocer en qué consiste la teoría de la computabilidad, teoría de la computación, modelos de computación y funciones recursivas así como su importancia dentro del entorno de la computación.
- 3.- Dar a conocer que es la máquina de Turín y saber cuál es su importancia para la solución de problemas relacionados con la computación lógica y matemáticas lógicas.
- 4.- Dar a conocer un poco de la historia del método de funciones recursivas generales y quienes son los principales autores de la creación del mismo.





JUSTIFICACIÓN

En la historia de las ciencias de la computación es importante determinar, estudiar, conocer todo sobre el principal personaje Alan turing por lo que sus de los principales y lo que dio origen a lo que hoy conocemos como computación, trataremos de explicar de una manera coherente la relación que existe entre las General recursive function (Funciones recursivas generales) y el modelo computacional de la máquina de turing.

¿Pero es tan necesario conocer esta relación?

La respuesta es sí ya que en muchas aplicaciones de la computación, y a lo largo de la historia las funciones recursivas han sido una de las principales bases para la implementación tanto de este modelo descrito en el siguiente trabajo como de otros tipos de modelos computacionales.

El principal motivo es que en el campo de las matemáticas como en la ciencias de la computación se relacionan mediante lo siguiente:

*“Las **funciones recursivas** son todas las funciones que pueden ser calculadas por la máquina de turing” según la teoría de la computabilidad de Alan Turing.*





MARCO TEÓRICO

Teoría de la computación:

La teoría de la computación se ocupa de determinar qué problemas pueden ser resueltos computacionalmente y con qué eficiencia. La teoría considera distintos modelos de cómputo, como los autómatas finitos (que son los más sencillos), las máquinas de Turing (que son las computadoras usuales de hoy en día) y las computadoras cuánticas (cuyo funcionamiento no es digital). Las lógicas y los lenguajes formales juegan un rol central en la teoría de la computación porque permiten expresar propiedades de los programas y razonar sobre su comportamiento. La teoría de la computación también se encarga de entender el límite entre los problemas computables y los no-computables y, dentro del mundo de lo computable, clasificarlos de acuerdo a su grado de simpleza o dificultad.

Teoría de la Computación. (s. f.). ICC - Instituto de Ciencias de la Computación.

Recuperado 21 de junio de 2021, de <https://icc.fcen.uba.ar/teoria-de-la-computacion/>

Teoría de la computabilidad:

La teoría de la computabilidad, también conocida como teoría de la recursividad, es una rama de la lógica matemática, la informática y la teoría de la computación que se originó en la década de 1930 con el estudio de funciones computables y grados de Turing. Desde entonces, el campo se ha expandido para incluir el estudio de la computabilidad generalizada y la definibilidad. En estas áreas, la teoría de la computabilidad se superpone con la teoría de la prueba y la teoría descriptiva de





conjuntos efectiva.

Aunque existe una superposición considerable en términos de conocimiento y métodos, los teóricos de la computabilidad matemática estudian la teoría de la computabilidad relativa, las nociones de deducibilidad y las estructuras

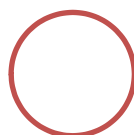
de grado; los del campo de la informática se centran en la teoría de jerarquías subrecursivas, métodos formales y lenguajes formales.

Wikipedia contributors. (2021, 7 mayo). Computability theory. Wikipedia.
https://en.wikipedia.org/wiki/Computability_theory

Modelos de computación:

MÁQUINA DE TURING

Alan Mathison Turing, matemático y computador científico inglés, desarrolló en torno a 1935 y 1945 un modelo computacional hipotético que permite en teoría resolver cualquier problema matemático siempre y cuando se reduzca a un algoritmo (conjunto ordenado y finito de operaciones que permiten la resolución de un problema, como por ejemplo la extracción de raíces cuadradas o el cálculo de una suma, producto o división de números, etc). De esta forma será posible calcular funciones dadas a partir de las operaciones más simples posibles y, aunque sea un algoritmo muy complejo, será posible descomponerlo en una mayor cantidad de pasos hasta resolverlo. Este modelo, que en principio vino a llamarse Máquina de computación lógica ("Logical Computing Machine"), pero que en honor a su idea se le acabó llamando máquina de Turing, es considerada la precursora de la computación digital moderna a pesar de su sencillez estructural. Según la Tesis de Church-Turing (llevada a cabo por Alan Turing y Alonzo Church a mediados del siglo XX de forma independiente), se demuestra que un problema es computable si se puede construir una máquina de Turing capaz de llevar a cabo el procedimiento para





resolverlo y que para cualquier otra máquina alternativa a ésta, se puede crear una Máquina de Turing equivalente capaz de resolver cualquier computación que se realice en la otra máquina.

MODELO DE JOHN VON NEUMANN

Desde tiempos muy antiguos el hombre ha necesitado hacer cálculos. Para ello se han buscado máquinas que lo ayudaran, como el ábaco, las hileras de John Napier... La primera máquina interesante es la “La Máquina Analítica” de Charles Babbage.

En 1945 John Von Neumann (1903 – 1957) creó un modelo computacional que se caracteriza por disponer de una única memoria principal en la que se almacenan los datos y las instrucciones. A esta memoria se accede a través de un sistema de buses único: bus de datos, de direcciones y de control. Este modelo se usa hoy día para describir los lenguajes de programación convencionales y es la base de prácticamente todos los modelos de ordenadores.

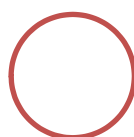
MODELO DE HARVARD

La arquitectura Harvard se caracteriza por tener la memoria de datos separada de la memoria del programa y estas a su vez están unidas a la CPU a través de buses independientes, luego, pueden tener distintos contenidos en la misma dirección. La arquitectura de Harvard permite a la CPU acceder simultáneamente a las dos memorias. Además, propicia numerosas ventajas al funcionamiento del sistema.

**Alejandro Aguilar García Natalia Domínguez Corbacho Jerónimo Valverde
Martínez Manuel Chacón Martín María Cuevas Rodríguez Carlos Porta Gómez.**

(2015, enero). MODELOS COMPUTACIONALES (N.o 1).

<http://informatica.utem.cl/~mcast/CCOMPUTACION/Introduccion/ModelosComputacionales.pdf>



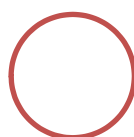


General recursive functions

Las funciones recursivas μ (o funciones recursivas generales) son funciones parciales que toman tuplas finitas de números naturales y devuelven un solo número natural. Son la clase más pequeña de funciones parciales que incluye las funciones iniciales y está cerrada bajo composición, recursividad primitiva y el operador μ .

La clase más pequeña de funciones, incluidas las funciones iniciales y cerrada en composición y recursividad primitiva (es decir, sin minimización) es la clase de funciones recursivas primitivas. Si bien todas las funciones recursivas primitivas son totales, esto no es cierto para las funciones recursivas parciales; por ejemplo, la minimización de la función sucesora no está definida. Las funciones recursivas primitivas son un subconjunto de las funciones recursivas totales, que son un subconjunto de las funciones recursivas parciales. Por ejemplo, se puede probar que la función de Ackermann es totalmente recursiva y no primitiva.

Wikipedia contributors. (2021a, mayo 6). General recursive function. Wikipedia.
https://en.wikipedia.org/wiki/General_recursive_function





METODOLOGÍA DE TRABAJO

DESIGN THINKING(EL PENSAMIENTO DE DISEÑO)

Escogimos esta Metodología del pensamiento del diseño que Aplica la forma de pensamiento y de trabajo de las personas especialistas en diseño, integrando enfoques de distintos campos y metodologías. Favorece y promueve la empatía, la intuición, la creatividad y la generación de ideas innovadoras. Aplicada a la educación favorece el desarrollo de competencias para la resolución de problemas mediante el trabajo en grupo, de forma creativa. El alumnado “aprende haciendo” y siente que aporta su granito de arena. Se convierte en protagonista de su propio aprendizaje, experimentando con herramientas y procesos que combinan momentos de divergencia, de convergencia y de síntesis.

DEFINIR E INTERPRETA

Comparte lo que se ha investigado en código de historias.

Encuentra los grandes temas, agrupando la información en temas o titulares.

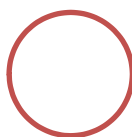
Convierte los problemas en oportunidades mediante preguntas generadoras de ideas.

IDEA

Da rienda suelta a tu imaginación y piensa en propuestas de soluciones viables y valiosas.

Agrupar, seleccionar y discutir las ideas prometedoras

PROTOTIPA Y EXPERIMENTA





Experimenta y prototipa

Construye prototipos

Obtén retroalimentación: comparte los prototipos e identifica los aspectos que deben mejorarse.

TESTEA Y EVOLUCIONA TU IDEA

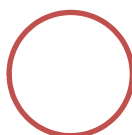
Haz seguimiento de los aprendizajes

Desarrollo e implementación

Como primer paso para la el desarrollo de este proyecto primero tenemos tener en claro que son las funciones recursivas podemos decir que son funciones que al momento de esta siendo ejecutadas por una modelo de computación se llaman a sí mismas para poder realizar acciones que se ejecutan n número de veces hasta poder llegar a un resultado definido un ejemplo claro es el siguiente:

En el siguiente pseudocódigo se muestra una simple cuenta regresiva en la cual se muestra que la cuenta comienza en un número dado previamente el cual se hace un pregunta si el número es mayor o se llama otra vez a las función y se resta un número así lo hasta que se cumpla no se cumpla dicha condición lo que cambiará el resultado a el caso contrario lo que hará que la cuenta termine y aparece el otro mensaje con lo que la función termina.

```
def cuenta_regresiva(número):  
    ...    número -= 1  
    ...    si numero > 0:  
    ...        escribe numero  
    ...        cuenta_regresiva(número)  
    ...    sí no:  
    ...        escribe "Boooooooooom!"
```





```
...     escribe "Fin de la función", número  
...  
>>> cuenta_regresiva(5)
```

Conclusión

Tras un largo estudio de la teoría de computación podemos concluir que a pesar de que existen distintos modelos de computación, cada uno de ellos contribuye a la teoría de la computación de manera significativa, ayudando así a resolver distintos problemas computables e ir clasificándolos por grado de complejidad.

Así mismo nos damos cuenta que ha contribuido a diversas ramas de la ciencia como la física, las matemáticas ayudando a resolver problemas que serían imposibles de calcular por mano propia debido a la complejidad de estos problemas, dando a entender al mundo que la computación es más que una máquina, si no también debe ser considerada como una ciencia y tratada como tal.

como podemos recordar es gracias a la máquina de turing que la computación como hoy la conocemos nace, y es gracias a esta máquina que funciona con algoritmos lógicos y simples que se logran resolver problemas increíblemente difíciles de realizar, contribuyen a que la segunda guerra mundial terminara dos años antes de lo esperado salvando millones de vidas humanas, sin duda de no haber mantenido el secreto por tantos años hoy la humanidad estaría un par de generaciones de computación más avanzada.

De esta forma tan peculiar es como la computación ha tenido un gran avance en el área de la ciencia, pero tampoco podemos ignorar el hecho de que hay mas grandes





contribuidores a la teoría de la computación como los fue John Von Neumann que crea el modelo computacional de una única memoria, o la arquitectura de harvard que permitió que la CPU accediera simultáneamente a dos memorias además de traer beneficios al funcionamiento de los sistemas.

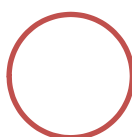
fuentes de información

Agualongo, C. O. O. (s. f.). *Funciones recursivas*. C.o.o. Recuperado 23 de junio de 2021, de <https://www.slideshare.net/CesarOsorio2/funciones-recursivas-63012069>

Agualongo, C. O. O. (s. f.). *Funciones recursivas*. C.o.o. Recuperado 23 de junio de 2021, de <https://www.slideshare.net/CesarOsorio2/funciones-recursivas-63012069>

colaboradores de Wikipedia. (2021b, abril 5). *Máquina de Turing*. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/M%C3%A1quina_de_Turing

Wikipedia contributors. (2021, 22 junio). *General recursive function*. Wikipedia. [https://en.wikipedia.org/wiki/General_recursive_function#Total_recursive function](https://en.wikipedia.org/wiki/General_recursive_function#Total_recursive_function)





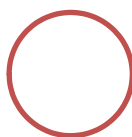
Anexos

Conjuntos computables y no computables

se genera en 1930, se establece la teoría de computabilidad de turing como la formación correcta de la idea informal del calculo efectivo, los resultados llevaron a [Stephen Kleene](#) a acuñar los dos nombres "Tesis de Church" (Kleene 1952: 300) y "Tesis de Turing" (Kleene 1952: 376).

Llegando a las primeras pruebas de que hay problemas en matemáticas que no pueden ser resueltos. el resultado mostró que no existe un procedimiento algorítmico que pueda decidir correctamente si las proposiciones matemáticas arbitrarias son verdaderas o falsas.

https://en.wikipedia.org/wiki/Computability_theory





: define el éxito y
EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA
Avanza.



TECNOLÓGICO
NACIONAL DE MÉXICO®

