

Derivadas con árboles binarios

Sara Palacios, Jessenia Piza, Laura Salazar

Repositorio: <https://github.com/jessepiza/ProyectoAlgoritmos>

1 Resumen ejecutivo

La derivada de una función es la razón de cambio instantánea con la que varía el valor de dicha función matemática, según se modifique el valor de su variable independiente. Geométricamente hablando, la derivada de una función en un punto, es la pendiente de la recta tangente a la gráfica de la función en dicho punto.

Es una herramienta esencial en varios campos como la Física, Química, Biología, Economía e incluso Sociología. Es usada usualmente para determinar si una función es creciente o decreciente, encontrar sus puntos máximos o mínimos, concavidad, puntos de inflexión, entre otros.

Ahora bien, no todas las personas saben cómo sacar la derivada de una función. Para aquellos que son amantes de las matemáticas tener una herramienta que los ayude a verificar sus resultados es indispensable, para los demás es una gran ayuda puesto que calcularlas a mano es un proceso, en algunos casos, muy tedioso. Por lo tanto, hemos propuesto como proyecto final para esta asignatura una Calculadora de derivadas para ayudar a las personas con estos cálculos.

2 Funcionalidad

Presentamos al usuario una interfaz gráfica en donde se debe ingresar la función que desea derivar.

Funciona de la siguiente manera:

- Cuando el usuario oprime el botón *derivar*, se crea un archivo de texto en la misma carpeta llamado *Funcion.txt*. Luego, este archivo, se lee en *main.cpp*, y este crea otro archivo de texto llamado *Derivada.txt*. El archivo de python donde está la interfaz gráfica lee este archivo y crea un cuadro de texto donde se observa la respuesta del archivo. Esto se logra gracias a que en el archivo de python importamos la librería *subprocess*, y al llamar *subprocess.call('./a.exe')*, empieza a correr el ejecutable creado por los archivos de c++.

3 Descripción

Para el desarrollo de esta herramienta utilizamos dos lenguajes de programación: Python con la librería *tkinter* para la interfaz gráfica y C++ en donde se toma la función pasada por el usuario y se realiza el debido procedimiento para dar una solución.

Implementamos en C++ las siguientes reglas para resolver la derivada de la función ingresada por el usuario:

- Derivada de función de grado n: Teniendo $f(x) = x^n$, entonces $f'(x) = nx^{n-1}$
- Derivada de la suma: $(f + g)'(x) = f'(x) + g'(x)$

- Derivada de un producto: $(f \cdot g)'(x) = f'(x) \cdot g(x) + f(x) \cdot g'(x)$
- Derivada de un cociente: $(\frac{f(x)}{g(x)})' = \frac{f'(x) \cdot g(x) - f(x) \cdot g'(x)}{g(x)^2}$
- Regla de la cadena: $(f \circ g)'(x) = f'(g(x)) \cdot g'(x)$
- Derivada de funciones logarítmicas y exponenciales:
 - Con $f(x) = c^{ax}$, tenemos que $f'(x) = \ln(c) \cdot a, c > 0$
 - Con $f(x) = e^x$, tenemos que $f'(x) = e^x$
 - Con $f(x) = \ln x$, tenemos que $f'(x) = \frac{1}{x}$
- Derivadas de funciones trigonométricas:

$(\sin x)' = \cos x$	$(\arccos x)' = -\frac{1}{\sqrt{1-x^2}}$
$(\cos x)' = -\sin x$	$(\arctan x)' = \frac{1}{1+x^2}$
$(\tan x)' = \sec^2 x$	$(\operatorname{arcsec} x)' = \frac{1}{ x \sqrt{x^2-1}}$
$(\sec x)' = \sec x \cdot \tan x$	$(\operatorname{arccsc} x)' = -\frac{1}{ x \sqrt{x^2-1}}$
$(\csc x)' = -\csc x \cdot \cot x$	$(\operatorname{arccot} x)' = -\frac{1}{1+x^2}$
$(\cot x)' = -\csc^2(x)$	
$(\arcsin x)' = \frac{1}{\sqrt{1-x^2}}$	

Además, hicimos uso de algunos conocimientos adquiridos en la asignatura Lógica para Ciencias de la Computación como lo es la Notación Polaca, la cual toma los operadores primero, seguidos de los operandos.

- Ejemplo: $2x+4$ se denota como: $+ [2x] [4]$

Esta notación es usada en la creación de la solución de la derivada.

4 Algoritmos y Estructuras de datos

Hemos utilizado un árbol binario implementado mediante una clase llamada Tree, la cual tiene como atributos:

- String Signo: operador con mayor relevancia en la función
- Tree *left: Operando antes del signo.
- Tree *right: operando después del signo.

Sus métodos más importantes son:

- Derivacion(): Argumento: objeto de la clase Tree el cual contiene la función a derivar. Retorna un objeto de la clase Tree que contiene la función ya derivada.
- display_tree(): Argumento: objeto de la clase Tree. Imprime el árbol.
- polaca(): Argumento: string. Retorna el string escrito en notación polaca.
- polaca_inv(): Argumento: string escrito en Notación polaca. Retorna el string invertido.
- stringtotree(): Argumento: string escrito en notación polaca inversa. Realiza el árbol con el string pasado.
- copyT(): Argumento: objeto de la clase Tree. Retorna un árbol exactamente igual al pasado como argumento.

Por último, el constructor copia se crea utilizando el método anteriormente mencionado, `stringtotree()`. Analiza primero si el string a derivar esta escrito en notación polaca inversa, si es así realiza el árbol, si no, hace la transformación y luego lo realiza.

Un ejemplo de la organización del árbol para la función $f(x) = 4x^2 + 2x$ se puede observar en la figura 1.

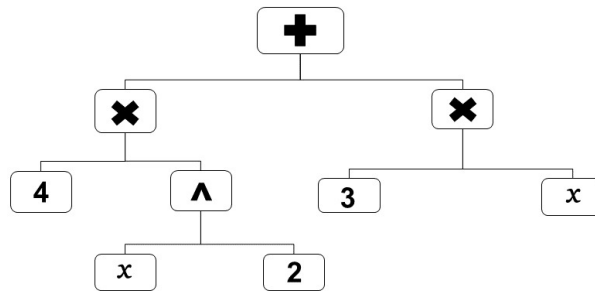


Figure 1: Árbol de $f(x) = 4x^2 + 2x$

Un algoritmo recursivo permite dar la solución a un problema mediante las llamadas a si mismo, le idea de esto es reducir la complejidad del problema. Se cuenta con un caso base en donde se indica el momento en el que algoritmo se debe detener y el caso recursivo en donde se realiza la reducción.

En este proyecto utilizamos algoritmos recursivos en varios casos, por ejemplo, en el método `polaca()` la recursión se realizó sobre las distintas partes del argumento dado, es decir, el paso recursivo se hace en el substring que se encuentra antes del operador con mayor significancia y el substring que se encuentra después. Por otro lado, en el método `derivacion()` se toma como caso base un nodo que es una constante, en dicho caso retorna 0, si no, identifica si es una variable, en dicho caso retorna 1; si el nodo no es ninguna de estas dos funciones pasa a hacer las reglas anteriormente presentadas dependiendo del operador. La recursión se realiza cuando es necesario calcular la derivada de alguno de los hijos del nodo.