



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL IPN

Proyecto 1. Evaluador de expresiones polinomiales

TECNOLOGÍAS DE LA PROGRAMACIÓN

Miembros del equipo:
Hernán Guillermo Dulcey Morán
Karla Jacquelin Guzmán Sánchez

29 de Octubre de 2019

1. Introducción

Se requiere implementar una calculadora que sea capaz de evaluar expresiones polinomiales. El proyecto basa su funcionamiento en expresiones regulares implementadas en el lenguaje de programación Python.

2. Requerimientos

El evaluador de expresiones polinomiales debe ser capaz de evaluar, reconocer y ejecutar expresiones matemáticas que incluyan o no alguna función implementada en el evaluador. A continuación se enlistan las funciones que estarán disponibles para el usuario.

- Suma
- Resta
- Multiplicación
- División
- Funciones trigonométricas
- Funciones hiperbólicas
- Funciones trigonométricas inversas
- Logaritmo natural
- Logaritmo base 2
- Logaritmo base 10
- Raíz cuadrada
- Módulo

De la misma forma, el evaluador deberá aceptar expresiones de polinomios de la forma:

$$a_mx^n + a_{m-1}x^{n-1} + a_{m-2}x^{n-2} + a_{m-3}x^{n-3} + \dots + a_1x + a_0$$

Donde a_i , x y m son valores en \mathbb{R} que corresponden a los coeficientes, variables independientes y exponentes respectivamente. La expresión polinomial también deberá aceptar más de una variable independiente.

La calculadora será capaz de procesar cualquier operación mencionada en los requerimientos siempre y cuando lleve la sintaxis adecuada.

3. Manual

La calculadora es capaz de resolver expresiones sin la necesidad de parentesis pero se recomienda hacer uso de estos para tener un mejor orden de la expresión.

La calculadora puede asignar valores a variables de la forma: $x=1$. En este caso se le ha asignado el valor de 1 a la variable x , por lo tanto en las siguientes expresiones que contengan la variable x , se reconocerá este valor y será reemplazado para resolver la ecuación. En caso de que el valor no haya sido asignado, la ecuación no puede resolverse.

El nombre de la variable siempre debe iniciar por una letra, pero apartir de ahí puede tomar cualquier valor excepto valores que hagan que el nombre de la variable coincida con una palabra reservada. La calculadora tiene las siguientes palabras reservadas, las cuales no pueden ser usadas para asignación de variables:

1. \ln : logaritmo natural
2. \log : logaritmo con base
3. $\sqrt{}$: raíz cuadrada
4. $\sin, \cos, \tan, \cot, \csc, \sec$: funciones trigonometricas
5. $\operatorname{asin}, \operatorname{acos}, \operatorname{atan}$: funciones trigonometricas inversas
6. $\sinh, \cosh, \tanh, \coth, \operatorname{csch}, \operatorname{sech}$: funciones hiperbólicas
7. $\operatorname{asinh}, \operatorname{acosh}, \operatorname{atanh}$: funciones hiperbólicas inversas.

Así como tambien existen caracteres especiales:

1. $*$: operador de multiplicación
2. $/$: operador de división
3. $\%$: operador de módulo
4. $+$: operador de suma
5. $-$: operador de resta
6. $^$: operador de potencia
7. $.$: separador de decimales
8. $,$: separador de base de logaritmo con base
9. $=$: operador de asignación

Cabe resaltar que la calculadora maneja el orden PEMDAS para las operaciones (Parentesis, exponente, multiplicación, división, adición y substracción). Las operaciones de logaritmo y trigonometricas son consideradas de orden aún mayor a las mencionadas anteriormente.

A continuación se describe la sintaxis que debe seguir el usuario para utilizar las funciones de manera correcta.

- Número entero: 3
- Número decimal: 2.3
- Suma: $2 + 3$
Respuesta esperada: 5
- Resta: $2 - 3$
Respuesta esperada: -1
- Multiplicación $2 * 3$
Respuesta esperada: 6
- División: $9 / 3$
Respuesta esperada: 3
- Módulo: $10 \% 3$
Respuesta esperada: 1
- Potencia: $2 ^ 2$
Respuesta esperada: 4
- Asignación: $x = 1$
Nota: Se hace distinción entre mayúsculas y minúsculas. **x** NO es equivalente a **X**.
- Agrupación
 $(2 + 2)*4$
Respuesta esperada: 16
- Raíz cuadrada: **sqrt** 4 ó **SQRT** 4
Respuesta esperada: 2
- Logaritmos:
 - Logaritmo natural: **ln** 1 ó **LN** 1
Respuesta esperada: 0
 - Logartimo con base: **log** número, base ó **LOG** 5,10
Ejemplo: Logaritmo base 10 de 5. **log** 5,10
Respuesta esperada: 0.6989700043360187
- Funciones trigonométricas
 - Seno: **sen** 90 ó **SEN** 90
 - Coseno: **cos** 90 ó **COS** 90
 - Tangente: **tan** 90 ó **TAN** 90

- Secante: **sec** 90 ó **SEC** 90
- Cosecante: **csc** 90 ó **CSC** 90
- Cotangente: **cot** 90 ó **COT** 90
- Funciones trigonométricas inversas
 - Arcoseno: **asen** 90 ó **ASEN** 90
 - Arcocoseno: **acos** 90 ó **ACOS** 90
 - Arco tangente: **atan** 90 ó **ATAN** 90
- Funciones hiperbólicas
 - Seno hiperbólico: **senh** 0.90 ó **SENH** 0.90
 - Coseno hiperbólico: **cosh** 0.90 ó **COSH** 0.90
 - Tangente hiperbólico: **tanh** 0.90 ó **TANH** 0.90
 - Secante hiperbólico: **sech** 0.90 ó **SECH** 0.90
 - Cosecante hiperbólico: **csch** 0.90 ó **CSCH** 0.90
 - Cotangente hiperbólico: **coth** 0.90 ó **COTH** 0.90
- Funciones hiperbólicas inversas
 - Arcoseno hiperbólico: **asenh** 0.90 ó **ASENH** 0.90
 - Arcocoseno hiperbólico: **acosh** 0.90 ó **ACOSH** 0.90
 - Arcotangente hiperbólico: **atanh** 0.90 ó **ATANH** 0.90

Nota: Todas las operaciones o funciones mencionadas pueden hacer uso de paréntesis para agrupar sus componentes si el usuario lo desea. Ejemplo: **ln** (5,10) es equivalente a **ln** 5,10 .

4. Conclusión

Las expresiones regulares nos ayudan a entender la sintaxis de una expresión y gracias a eso podemos añadir lógica al resultado de dicha expresión. Así como se analizaron sentencias simples, con nuevos procedimientos se podrían realizar el análisis y obtener el resultado de operaciones más complejas.