

Instituto Tecnológico de Nuevo León



Lenguaje y autómatas II

Unidad I

Proyecto I: Árbol de Expresiones

Maestro: Rosas Baldazo Juan Pablo

Alumno: Erick Mauricio Medina Hernandez

Matricula: #13480548

## Introducción

Un árbol binario es una estructura de datos en la cual cada nodo puede tener un hijo izquierdo y un hijo derecho. No pueden tener más de dos hijos (de ahí el nombre "binario"). Si algún hijo tiene como referencia a null, es decir que no almacena ningún dato, entonces este es llamado un nodo externo. En el caso contrario el hijo es llamado un nodo interno. Usos comunes de los árboles binarios son los árboles binarios de búsqueda, los montículos binarios y Codificación de Huffman.

Tipos de árboles binarios: Es un árbol en el que ningún nodo puede tener más de dos subárboles. En un árbol binario cada nodo puede tener cero, uno o dos hijos (subárboles). Se conoce el nodo de la izquierda como hijo izquierdo y el nodo de la derecha como hijo derecho.

## Desarrollo

Se realizó un árbol binario de búsqueda de expresiones el cual automáticamente se encarga de reconocer los operadores y los números de manera que genera el árbol binario, el objetivo es que el usuario introduzca una expresión y el programa arroje la expresión en infija y postfija.

Esta actividad se hizo en uno de los programas de java llamados NetBeans.

**Paso 1:** Lo primero que se hizo fue crear un “Pseudocódigo” el cual ayuda a comprender un poco más la logística con la que debería contar el código para que pudiera realizar el objetivo final.

**Paso 2:** Hacer el Pseudocódigo.

pedir expresión por teclado

almacenar la expresión en una variable “exp”

declarar las pilas que se utilizaran En, Pt, Sa

se agregó la expresión a la pila entrada

se declaró la jerarquía de cada uno de los operadores

```
if (op.equals("^")) {
```

```
    prf = 5;
```

```
}
```

```
if (op.equals("*") || op.equals("/")) {
```

```
    prf = 4;
```

```
}
```

```
if (op.equals("+") || op.equals("-")) {
```

```
    prf = 3;
```

```
}
```

Se creó un ciclo para para infijo a postfijo en base a la jerarquía de cada operador iremos acomodando los datos de la pila entrada en pt para al final reacomodarlo en la ps.

### **Experimento**

Equipo: Laptop HP Pavilion

Memoria RAM: 8 GB DDR4

Procesador: AMD A8 ELITE QUAD-CORE

Sistema: Windows 10 64 bits

### **Pruebas**

Estas son las siguientes expresiones que se realizó:

Expresión	Tiempo de ejecución(s)
$3^3+(5-2)*6/3$	11 seconds
$(4+4)/(2-3)+2$	2 seconds
$6-9*8/7+3$	1 second
$1+2/3*4-5$	3 seconds
$(12^2)^2 / 5 + 6 / 2 - 4 * 1(14^2)^2$	14 seconds
$1+4(6*6)$	1 second
$8-2+4/5^2$	5 seconds
$3+9-9/6-5+7*4$	4 second
$(9+(4+5))^2$	22 seconds
$(12^2)^2 / 5 + 6 / 2 - 4 * 1$	56 seconds

### Resultado

En esta grafica se muestra el tiempo de ejecución que se tardó cada una de las expresiones.



### Conclusión

En este proyecto se realizó un programa que fuera capaz de realizar una función completa y de mismo modo que se tomara el tiempo que se tardaba en ejecutar.

La función tenía que ser con números y operadores aritméticos, ya que después tenía que mostrar el árbol de expresiones, que es llamado la notación.

### Bibliografía

Obtenido de

<http://interactivepython.org/runestone/static/pythoned/Trees/RepresentacionDeListaDeListas.html>