

1.1 Árboles de expresiones

Los árboles de expresiones representan el código de nivel del lenguaje en forma de datos. Los datos se almacenan en una estructura con forma de árbol. Cada nodo del árbol de expresión representa una expresión, por ejemplo, una llamada al método o una operación binaria, como $x < y$.

Un árbol de expresión sirve para evaluar expresiones del tipo: **$(a + b) * c / d$**

Para que un árbol represente una expresión se deben tomar en cuenta 2 características muy importantes:

- Cualquier hoja está etiquetada sólo con un operando.
- Cualquier nodo interior n está etiquetada por un operador.

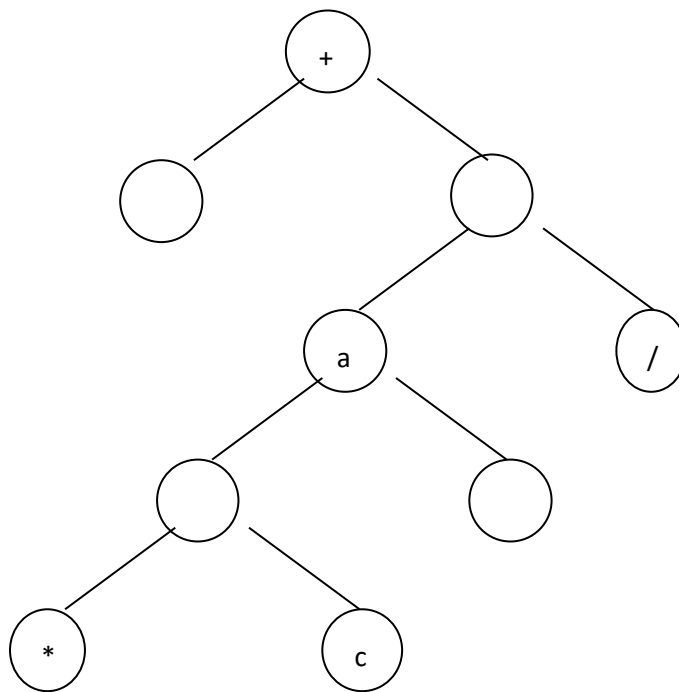


Imagen 1. Árbol de expresiones

Al introducir la expresión debemos de tomar en cuenta las siguientes características:

- La raíz siempre debe ser un operador
- Las hojas siempre deben ser operandos
- Los nodos deben estar etiquetados por operadores
- Si un operador tiene mayor prioridad que la raíz se coloca como hijo.
- Si un operador tiene igual o menor prioridad que un nodo se coloca como padre.
 - Un nodo puede contener como hijo otro subárbol que contiene una pequeña expresión.

En los árboles de expresión, la sucesión del preorden de etiquetas nos da lo que se conoce como la forma prefija de una expresión

Análogamente, la sucesión postorden de las etiquetas de un árbol expresión nos da lo que se conoce como la representación postfija de una expresión

Finalmente, el inorden de una expresión en un árbol de expresión nos da la expresión infijo en sí misma, pero sin paréntesis

Construcción de un árbol de expresión

Algoritmo

- Mientras carácter diferente de nulo
- Leer carácter de la lista
- Si es paréntesis pasar al siguiente carácter
 - Crear un nodo nuevo que contenga ese carácter

Operando

- Si el árbol está vacío hacer raíz a nuevo, si no recorrer el árbol por la derecha hasta llegar a un nodo con hojas, si la hoja izquierda,

no está etiquetada colocar operando, si no colocarlo en la hoja derecha.

Operador

- Si la raíz es un operando, insertar nuevo en ese nodo, y convertir el operando en el hijo izquierdo, si no si hay un paréntesis abierto insertar nuevo en la última hoja derecha y colocar operando como hijo izquierdo.
- Si el carácter anterior es paréntesis izquierdo si el siguiente carácter es paréntesis derecho si solo hay un operador en el árbol nuevo se convierte en raíz, si no se inserta en el último nodo derecho, y el nodo se convierte en hijo izquierdo.
- Si no se cumple ninguna de las condiciones anteriores si la raíz es de igual prioridad o menor prioridad convertir la raíz en el hijo izq. de nuevo si no la prioridad del nodo raíz es mayor al de nuevo insertar nuevo como hijo derecho y colocar el nodo reemplazado como hijo izquierdo.

REGLAS PARA LA CONSTRUCCION DE ARBOLES DE EXPRESION

Para contruir el árbol de expresiones que represente nuestra expresión matemática es necesario construir primero la misma expresión pero en la notación polaca correspondiente y a partir de esta es que se construye el árbol. El algoritmo usado para transformar una expresión infija a prefija es explicado a continuación.

Sea A una expresión infija cualquiera, formada por operadores, paréntesis (izquierdos y derechos) y operandos, también se usará una pila para los operadores. El procedimiento seguido es el siguiente:

Se lee un elemento de A, si este es un operador o un paréntesis izquierdo, entonces se actúa según la regla I y si es un operando entonces se envía directamente a la expresión de notación polaca. Si el elemento leído de A es un paréntesis derecho, entonces se desapilarán elementos de la pila de operadores hasta encontrar el correspondiente paréntesis izquierdo. Cada elemento desapilado pasa a formar parte de la notación polaca, excepto los paréntesis. Cuando no queden elementos en A, entonces se desapilan operadores de la pila, hasta que esta quede vacía.

Regla I:

Existe un orden de prioridad para los operadores, que de menor a mayor es el siguiente: suma (+) y resta (-), multiplicación (*) y división (/), exponenciación (^), operadores unarios. El paréntesis izquierdo lo trataremos como un operador (aunque no lo es) cuyo orden de prioridad es el mayor de todos cuando se quiera apilar y el menor de todos cuando esté en la cima de la pila.

Cuando se intente apilar algún operador se hará lo siguiente: si es un operador unario entonces se apila, si es un operador binario, se comparará su prioridad con el último insertado en la pila (el de la cima), si su prioridad es mayor, entonces se apilará. Si ocurre lo contrario (su prioridad es menor o igual) entonces el operador de la cima de la pila se desapilará y pasará a formar parte de la notación polaca. Se volverá a intentar apilar el operador siguiendo la misma regla, hasta que se pueda apilar, si la pila queda vacía también se apila. El paréntesis izquierdo siempre se apilará y no podrá ser desapilado por ningún operador y por tanto no formará parte de la notación polaca inversa.

El siguiente ejemplo, ayudará a entender mejor lo dicho anteriormente. Sea la siguiente expresión infija: $2^{\sin(y+x)} - \ln(x)$.

En la siguiente tabla se muestra paso a paso la conversión a notación postfija. Se usa el color rojo para señalar los casos en que es necesario des apilar operadores de la pila.

Construcción del árbol binario de expresiones

Una vez obtenida la expresión en notación postfija, se puede evaluar mediante el uso nuevamente de una pila. Sin embargo, en nuestro caso se trabaja con un árbol binario de expresiones, así que lo que se hace es construir el árbol. El algoritmo usado para construir el árbol no usa como tal la expresión postfija ya conformada, sino que el árbol se va construyendo usando las mismas reglas con las que se construye la notación postfija, una pila para los operadores y otra para los nodos del árbol, ambas no son necesarias al terminar el árbol. El algoritmo es el siguiente:

Se siguen las mismas reglas expuestas anteriormente usando la pila de operadores, pero cuando se encuentra un operando o un operador es desapilado, entonces se crea el nodo correspondiente y se actúa según la regla II. Al finalizar el algoritmo solo debe quedar un nodo apilado en la pila de nodos, el que constituye el nodo raíz de nuestro árbol de expresiones.

Regla II.

Si el nodo corresponde a un operando, entonces se apila. Si el nodo corresponde a un operador unario entonces se desapila un nodo de la pila de nodos y es enlazado a la rama izquierda del nodo correspondiente al operador unario y este último es apilado. Si el nodo corresponde a un operador binario entonces dos nodos son desapilados de la pila de nodos, el primero es enlazado a la rama derecha del nodo binario y el segundo a la rama izquierda, nuevamente este nodo es apilado.