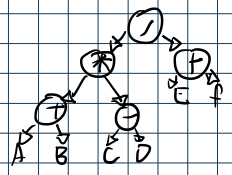
Clase 5 - Árboles de expresión

miércoles, 29 de marzo de 2023 17:03

Árbol de expresión → A. binario asociado a expresión aritmética.

Nodos internos: operadores

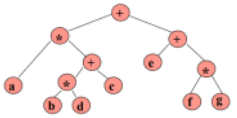
hojas externas: Operandos



- se usan en compil para análisis, optimización de programas
- Evaluar expresiones aritm-lógicas
- No requiere paréntesis
- notación: sufija, prefija, infija.

Inorden: (((a + b) * (c – d)) / (e + f))
Preorden: /*+ab-cd+ef
Postorden: ab+cd-*ef+/

a*(b*d +c) + (e+ f*g)



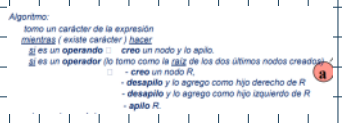
Expresión **prefija** → + * a + * b d c + e * f g
Expresión **postfija** → a b d * c + * e f g * + +
Expresión **infija** → ((a *((b * d) + c)) + (e + (f * g)))

Desde post-fija.

a B D*C + * e F g * + +

operando operador

crea nodo, se apila



crea nodo, desapila 1 nodo, apila como h. derecho

y el otro lo pone como izquierdo

Desde infija:

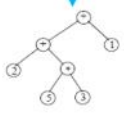
ArbolExpresión (A: ArbolBin, exp: string)

si exp nulo → nada.
si es un operador → - creo un nodo raíz R
- ArbolExpresión (subArbIzq de R, exp (sin 1° carácter))
- ArbolExpresión (subArbDer de R, exp (sin 1° carácter))

si es un operando → creo un nodo (hoja)

3) Construcción de un árbol de expresión a partir de una expresión infija

Expresión infija — 2+5*3+1
(i) Se usa una pila y se tiene en cuenta la precedencia de los operadores
Expresión postfija — 253*+1+
(ii) Se usa la estrategia 1)
Árbol de Expresión



para pasar de infija a post fija.

a) si es un operando → se coloca en la salida.
b) si es un operador → se maneja una pila según la prioridad del operador en relación al tope de la pila

operador con > prioridad que el tope –> se apila
operador con <= prioridad que el tope –> se desapila elemento colocándolo en la salida.
Se vuelve a comparar el operador con el tope de la pila

c) si es un "(" , ")" → "(" se apila
")" se desapila todo hasta el "(", incluído éste

d) cuando se llega al final de la expresión, se desapilan todos los elementos llevándolos a la salida, hasta que la pila quede vacía.

prioridad: ^ por.

*, /

+, -

=

(→ el de mayor prioridad se desapila sobre sí mismo)

???

Evaluar un árbol de expresión

Algoritmo:

EvaluarAE (A: ArbolBin)

*si dato es **operador*** □ *EvaluarAE (subArbIzq de A)*
EvaluarAE (subArbDer de A)

si es un operando □ *Retornar el dato del nodo (hoja)*

*si dato es **operador** //*
valorIzq = EvaluarAE (subArbIzq de A)
valorDer = EvaluarAE (subArbDer de A)
*según el valor **operador** {*
"+" : retornar valorIzq + valorDer
"-" : retornar valorIzq - valorDer
"" : retornar valorIzq * valorDer*
"/" : retornar valorIzq / valorDer }

*si es un **operando*** □ *Retornar el dato del nodo (hoja)*

Arboles Binarios
Recorridos

Preorden
Se procesa primero la raíz y luego sus hijos, izquierdo y derecho.
`40, 25, 10, 32, 78`

Inorden
Se procesa el hijo izquierdo, luego la raíz y último el hijo derecho
`10, 25, 32, 40, 78`

Postorden
Se procesan primero los hijos, izquierdo y derecho, y luego la raíz
`10, 32, 25, 78, 40`

Por niveles
Se procesan los nodos teniendo en cuenta sus niveles, primero la raíz, luego los hijos, los hijos de éstos, etc.
`40, 25, 78, 10, 32`