

Aplicación de pilas en expresiones

Tratamiento de expresiones aritméticas.

Un problema interesante en computación es poder convertir expresiones en notación infija a su equivalente en notación postfija (o prefija). Revisemos los conceptos:

- Dada la expresión $A + B$ se dice que está en notación infija, y su nombre se debe a que el operador (+) está entre los operandos (A y B).
- Dada la expresión $AB+$ se dice que está en notación postfija, y su nombre se debe a que el operador (+) está después de los operandos (A y B).
- Dada la expresión $+AB$ se dice que está en notación prefija, y su nombre se debe a que el operador (+) está antes que los operandos (A y B).

La ventaja de usar expresiones en notación polaca postfija o prefija radica en que no son necesarios los paréntesis para indicar orden de operación, ya que éste queda establecido por la ubicación de los operadores con respecto a los operandos.

Para convertir una expresión dada en notación infija a una en notación postfija (o prefija), deberán establecerse previamente ciertas condiciones:

- Solamente se manejarán los siguientes operadores (están dados ordenadamente de mayor a menor según su prioridad de ejecución) :
 - \wedge (potencia)
 - $* /$ (multiplicación y división)
 - $+ -$ (suma y resta)
- Los operadores de más alta prioridad se ejecutan primero.
- Si hubiera en una expresión dos o más operadores de igual prioridad, entonces se procesarán de izquierda a derecha.
- Las subexpresiones entre paréntesis tendrán más prioridad que cualquier operador.

A continuación, se presentan ejemplos de traducción de notación infija a postfija. El primero de ellos es una expresión simple, mientras que el segundo presenta un mayor grado de complejidad. En la siguiente tabla se presentan los pasos necesarios para lograr la traducción de la primera expresión:

a) Ejemplo 1:

Expresión infija: $X + Z * W$

Paso	Exp. Infija	Símbolo Analizado	Pila	EPOS
0	$X + Z * W$			
1	$+ Z * W$	X		X
2	$Z * W$	+	+	X
3	$* W$	Z	+	XZ
4	W	*	+ *	XZ
5		W	+ *	XZW
6			+	XZW*
7				XZW*+

EPOS : Expresión en postfijo

Expresión postfija : $XZW*+$

En los pasos 1, 3 y 5 el símbolo analizado (un operando) se agrega directamente a EPOS. Al analizar el operador +, paso 2, se verifica si en la PILA hay operadores con mayor o igual prioridad. En este caso PILA está vacía, por lo tanto, se pone el símbolo en el tope de ella. Con el operador *, paso 4, sucede algo similar. En la PILA no existen operadores de mayor o igual prioridad (la suma tiene menor prioridad que la multiplicación), por lo que se agrega el operador * a PILA. En los dos últimos pasos, 6 y 7, se extraen de PILA sus elementos, agregándolos a EPOS.

Algoritmo para convertir una expresión infija a postfija:

Conv_postfija(EI,EPOS)

{Este algoritmo traduce una expresión infija EI a postfija EPOS, haciendo uso de una pila PILA}

{TOPE es una variable de tipo entero}

1. Hacer $TOPE \leftarrow 0$; $int\ x = 0$;
Mientras existan símbolos en la hilera
Leer el simb
Si es operando entonces
Traslade a EPOS
Sino
Si $simb == '('$ entonces
Push(S)
Sino
Si simb es ')' entonces
Sacar los elementos de S y ponerlos en EPOS
Sino
Si simb es un operador y tiene mayor precedencia que el de (S) entonces

Poner en S
 Sino
 Sacar de S y poner en EPOS
 Poner simb en la pila
 Fin_si
 Fin_si
 Fin_si
 Fin_si
 Vaciar la pila
 Fin_del_algoritmo

Nota: Algunas condiciones requieren analizar todos los operadores que estén en la pila.

b) Ejemplo 2:

Expresión infija: $(X + Z) * W / T ^ Y - V$

En la siguiente tabla se presentan los pasos necesarios para lograr la traducción deseada, siguiendo el algoritmo:

PASO	EI	SIMBOLO ANALIZADO	PILA	EPOS
0	$(X+Z)*W/T^Y-V$			
1	$X+Z)*W/T^Y-V$	((
2	$+Z)*W/T^Y-V$	X	(X
3	$Z)*W/T^Y-V$	+	(+	X
4	$) *W/T^Y-V$	Z	(+	XZ
5	$*W/T^Y-V$)	(XZ+ XZ+
6	W/T^Y-V	*	*	XZ+
7	$/T^Y-V$	W	*	XZ+W
8	T^Y-V	/	/	XZ+W* XZ+W*
9	^Y-V	T	/	XZ+W*T
10	$Y-V$	^	/ ^	XZ+W*T
11	$- V$	Y	/ ^	XZ+W*TY
12	V	-	/	XZ+W*TY^ XZ+W*TY^/ XZ+W*TY^/
13		V	-	XZ+W*TY^/V
14				XZ+W*TY^/V-

Expresión postfija: $XZ+W*TY^/V-$

Ejercicios:

Utilizando el algoritmo correspondiente, convertir las siguientes expresiones infijas a expresiones en postfijo:

- a) $A + B * C$
- b) $A + B - C$
- c) $X \wedge Y * W$
- d) $A - B * C / D$
- e) $A + B * (A + B * C)$
- f) $(A + B) * E + F$
- g) $(A + B) * (C - D) \wedge E * F$
- h) $(A + B) * (C \wedge (D - E) + F) - G$
- i) $M * P + (A / B + C)$
- j) $A * (B + C) * (C + B - A)$

EVALUACION DE EXPRESIONES EN POSTFIJO

Otras de las aplicaciones de las pilas son en la evaluación de expresiones en postfijo. Para lo cual cada operador en una hilera en postfijo se refiere a los dos operandos anteriores en la hilera. Por supuesto que cualquiera de los dos operandos puede ser el resultado de haber aplicado un operador previo.

Procedimiento:

- Cada vez que leemos un operando lo empujamos en una pila.
- Cuando alcanzamos un operador, sus operandos serán entonces los dos elementos superiores de la pila.
- Entonces, podemos sacar dos elementos, realizar la operación indicada en ellos, y empujar el resultado en la pila de tal manera que estará disponible para su uso como un operando del siguiente operador.

Algoritmo para evaluar una expresión en postfijo.
 { Este algoritmo evalúa una expresión en postfijo }

Mientras existan más caracteres en postfijo hacer

Inicio

symb = carácter leído

if symb es un operando entonces

 push(S,symb)

else

 operando2 = pop()

 operando1 = pop()

 valor = resultado de operar al operando1 y operando2 el operador leído

 push (S, valor)

fin_si

resultado = pop()

fin_del_algoritmo

Ejemplo: Evaluar la siguiente expresión:

$$6 + 4 * (9 + 5 * 2 - 3)$$

1- Convertir la expresión a postfijo.

Después de aplicado el procedimiento obtenemos la sig. Expresión en postfijo:

$$6\ 4\ 9\ 5\ 2\ *\ +\ 3\ -\ *\ +$$

Symb	Operando1	Operando2	Valor	Pila(S)
6				6
4				6,4
9				6,4,9
5				6,4,9,5
2				6,4,9,5,2
*	5	2	10	6,4,9,10
+	9	10	19	6,4,19
3	10	9	19	6,4,19,3
-	19	3	16	6,4,16
*	4	16	64	6,64
+	6	64	70	70

Resultado = 70

Observaciones:

- La pila contendrá los operandos. Cada operando es empujado hacia la pila cuando es encontrado. El tamaño máximo de la pila es el número de operandos que aparecen en la expresión de entrada. Aunque en la mayoría de los casos el tamaño real de la pila requerida puede ser menor que el máximo, debido a que un operador retira operandos de la pila.

- Cada línea de entrada está en la forma de una hilera de dígitos y de operadores.
- Los operandos son dígitos simples no negativos (0,1,...,9).
- La hilera es de caracteres.
- Los símbolos se leen como caracteres, hay que buscar la forma para convertir los operandos en números y los caracteres que representan operadores a operaciones.

Ejercicios:

Evaluar usando el algoritmo correspondiente, las siguientes expresiones aritméticas:

- a) $5 * 4 + (9 / 3 + 8 * 2)$
- b) $7 + 3 * (9 + 5 * 2 ^ 3 - 8)$
- c) $4 * (2 + 3 - 2) * (4 + 8 - 5)$
- d) $8 + 4 + ((5 ^ 2 + 6) * 4)$
- e) $6 * 2 + 8 - 3 * 2 / 2$

- ***Implementar programa en Python para convertir y evaluar expresiones en postfijo utilizando estructura de datos tipo pila.***