



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE INGENIERIA



ASIGNATURA: ESTRUCTURA DE DATOS Y ALGORITMOS
ACTIVIDAD #4 LUNES

NOMBRE DEL ALUMNO: RIVERA VARGAS DONOVAN JOSHEP

Realizar un trabajo de investigación escrito referente a la notación
polaca y la notación polaca inversa, además de
agregar su algoritmo de implementación para cada
una



Realizar un trabajo de investigación escrito referente a la notación polaca y la notación polaca inversa, además de agregar su algoritmo de implementación para cada una

Notación infijo, prefijo y sufijo

Cuando escribe una expresión aritmética como $B * C$, la forma de la expresión le proporciona información para que pueda interpretarla correctamente. En este caso, sabemos que la variable B se está multiplicando por la variable C ya que el operador de multiplicación $*$ aparece entre ellos en la expresión. Este tipo de notación se denomina infijo ya que el operador se encuentra *entre* los dos operandos en los que está trabajando.

Considere otro ejemplo infijo, $A + B * C$. Los operadores $+$ y $*$ todavía aparecen entre los operandos, pero hay un problema. ¿En qué operandos trabajan? ¿El $+$ funciona en A y B o el $*$ toma B y C ? La expresión parece ambigua.

De hecho, ha estado leyendo y escribiendo este tipo de expresiones durante mucho tiempo y no te causan ningún problema. La razón de esto es que tiene algún conocimiento sobre los operadores $+$ y $*$: cada operador tiene un nivel de precedencia. Los operadores de mayor precedencia se utilizan antes que los operadores de menor precedencia. Lo único que puede cambiar ese orden es la presencia de paréntesis. El orden de precedencia para los operadores aritméticos coloca la multiplicación y la división por encima de la suma y la resta. Si aparecen dos operadores de igual precedencia, se utiliza una ordenación o asociatividad de izquierda a derecha.

Interpretemos la expresión problemática $A + B * C$ usando la precedencia del operador. B y C se multiplican primero, y luego se agrega A a ese resultado. $(A + B) * C$ forzaría realizar la operación de adición de A y B antes de la multiplicación. En la expresión $A + B + C$, por precedencia, el $+$ más a la izquierda se haría primero.

Aunque todo esto puede ser obvio, es importante recordar que las computadoras necesitan saber exactamente que operadores realizar y en qué orden. Una forma de escribir una expresión que garantice que no habrá confusión con respecto al orden de las operaciones es crear lo que se llama una expresión entre paréntesis. Este tipo de expresión usa un par de paréntesis para cada operador. Los paréntesis dictan el orden de las operaciones; No hay ambigüedad. Tampoco es necesario recordar ninguna regla de precedencia.

La expresión $A + B * C + D$ puede reescribirse como $((A + (B * C)) + D)$ para mostrar que la multiplicación ocurre primero, seguida de la suma más a la izquierda. $A + B + C + D$ puede escribirse como $((((A + B) + C) + D)$ ya que las operaciones de suma se asocian de izquierda a derecha.

Existen otros dos formatos de expresión muy importantes que pueden no parecer obvios al principio. Considere la expresión infija $A + B$. ¿Qué sucedería si moviéramos el operador antes de los dos operandos? La expresión resultante sería $+ A B$. Asimismo, podríamos mover el operador al final. Obtendríamos $A B +$. Estos se ven un poco extraños.

Estos cambios en la posición del operador con respecto a los operandos crean dos nuevos formatos de expresión: prefijo y sufijo. La notación de expresión de prefijo requiere que todos los operadores precedan a los dos operandos en los que trabajan. El sufijo, por otro lado, requiere que sus operadores vengan después de los operandos correspondientes. Algunos ejemplos más deberían ayudar a que esto sea un poco más claro:

EXPRESIÓN INFIJO	EXPRESIÓN PREFIJO	EXPRESIÓN SUFIJO
$A + B$	$+ A B$	$A B +$
$A + B * C$	$+ A * B C$	$A B C * +$

$A + B * C$ se escribiría como $+ A * B C$ en la notación de prefijo. El operador de multiplicación viene inmediatamente antes de los operandos B y C, denotando que * tiene prioridad sobre +. El operador de suma aparece entonces antes de la A y el resultado de la multiplicación.

En la notación de sufijo, la expresión sería $A B C * +$. Nuevamente, el orden de las operaciones se conserva ya que * aparece inmediatamente después de B y C, lo que denota que * tiene prioridad, con + después. Aunque los operadores se movieron y ahora aparecen antes o después de sus respectivos operandos, el orden de los operandos se mantuvo exactamente igual entre sí.

Ahora considere la expresión infija $(A + B) * C$. Recuerde que en este caso, la notación infija requiere los paréntesis para forzar la operación de la suma antes de la multiplicación. Sin embargo, cuando $A + B$ se escribió en la notación de prefijo, el operador de suma simplemente se movió antes que los operandos, $+ A B$. El resultado de esta operación se convierte en el primer operando para la multiplicación. El operador de multiplicación se mueve delante de toda la expresión, dándonos $* + A B C$.

Considere estas expresiones nuevamente.

EXPRESIÓN INFIJO	EXPRESIÓN PREFIJO	EXPRESIÓN SUFIJO
$(A + B) * C$	$* + A B C$	$A B + C *$

Algo muy importante ha sucedido. ¿A dónde fueron los paréntesis? ¿Por qué no los necesitamos en la notación de prefijo y de sufijo? La respuesta es que los operadores ya no son ambiguos con respecto a los operandos en los que trabajan. Solo la notación infija requiere los símbolos adicionales. El orden de las operaciones dentro de las expresiones de prefijo y de sufijo está completamente determinado por la posición del operador y nada más.

La siguiente tabla muestra algunos ejemplos adicionales de expresiones en notación de infijo y su equivalente en notación de prefijo y sufijo. Es importante poner atención en como son expresiones equivalentes por el orden en que se deben realizar las operaciones.

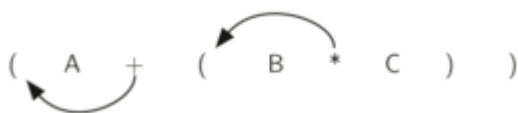
EXPRESIÓN INFJO	EXPRESIÓN PREFIJO	EXPRESIÓN SUFIJO
$A + B * C + D$	$++A * B C D$	$A B C * + D +$
$(A + B) * (C + D)$	$* + A B + C D$	$A B + C D + *$
$A * B + C * D$	$+ * A B * C D$	$A B * C D * +$
$A + B + C + D$	$+++A B C D$	$A B + C + D +$

Convertir expresiones de infijo a expresiones de pefijo

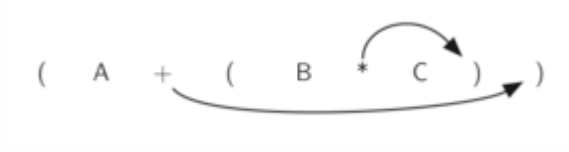
Hasta ahora, hemos utilizado métodos ad hoc para convertir entre expresiones en notación de infijo a expresiones equivalentes en notación de prefijo y de sufijo. Como es de esperar, hay formas algorítmicas para realizar la conversión que permiten que cualquier expresión de cualquier complejidad se transforme correctamente.

La primera técnica utiliza la noción de una expresión completamente entre paréntesis que se discutió anteriormente. Recuerde que $A + B * C$ puede escribirse como $(A + (B * C))$ para mostrar explícitamente que la multiplicación tiene prioridad sobre la suma. En una observación más cercana, sin embargo, puede ver que cada par de paréntesis también denota el comienzo y el final de un par de operandos con el operador correspondiente en el medio.

Mire el paréntesis izquierdo en la subexpresión $(B * C)$ anterior. Si movemos el símbolo de multiplicación a esa posición y eliminamos el paréntesis derecho correspondiente, quedando la expresión de la siguiente manera $* B C$, heos convertido la subexpresión en notación de prefijo. Si el operador de suma también se moviera a su correspondiente posición del paréntesis izquierdo y se eliminara el paréntesis derecho correspondiente, se obtendría la expresión completa en notación de prefijo.

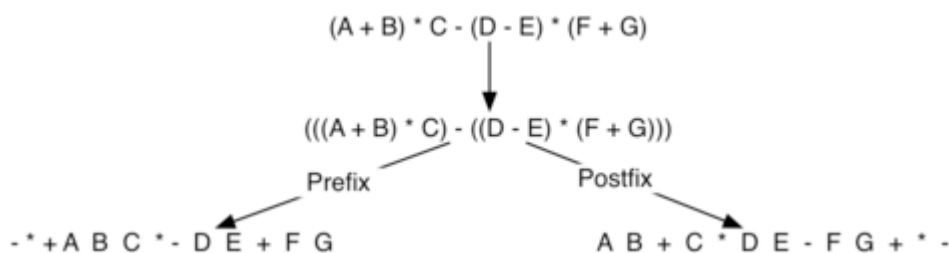


Si realizamos la misma operación pero moviendo los símbolos de operación a la derecha del paréntesis en lugar de a la izquierda, obtendremos la notación de sufijo.



Entonces, para convertir una expresión, no importa cuán compleja sea, ya sea a la notación de prefijo o sufijo, primero se debe generar una expresión completamente entre paréntesis usando el orden de las operaciones. Luego hay que mover el operador adjunto a la posición del paréntesis izquierdo o derecho dependiendo de si desea la notación de prefijo o sufijo.

Aquí hay una expresión más compleja: $(A + B) * C - (D - E) * (F + G)$. La siguiente figura muestra la conversión a notaciones de prefijo y sufijo.



Notación polaca inversa

La notación polaca inversa (en inglés, *Reverse Polish Notation*, o *RPN*), es una forma alternativa de escribir expresiones matemáticas. Por ejemplo, la expresión $20 - (4 + 3) * 2$ en RPN es $20\ 4\ 3\ +\ 2\ *\ -$

Para evaluar una expresión en RPN, usamos una lista auxiliar (inicialmente vacía) y recorremos la expresión de izquierda a derecha. Cada vez que encontramos un número, lo añadimos a la lista auxiliar. Cuando encontramos un operador, retiramos los dos números que hay al principio de la pila, utilizamos el operador con ellos y los quitamos de la lista y le añadimos el resultado. Cuando alcancemos el final de la expresión, debemos tener un solo número en la lista auxiliar si la expresión estaba bien formada, y éste representa el resultado de la expresión. Por ejemplo, la evaluación de RPN $20\ 4\ 3\ +\ 2\ *\ -$ es la siguiente

""	[]
"20"	[20]
"20 4"	[4, 20]
"20 4 3"	[3, 4, 20]
"20 4 3 +"	[7, 20]
"20 4 3 + 2"	[2, 7, 20]
"20 4 3 + 2 *"	[14, 20]
"20 4 3 + 2 * -"	[6]

Definir la función

```
valor :: String -> Float
```

tal que (valor cs) es el valor de la expresión RPN cs. Por ejemplo,

```
valor "4" == 4.0
valor "4 3 +" == 7.0
valor "4 3 + 2 *" == 14.0
valor "20 4 3 + 2 * -" == 6.0
valor "3 7 + 2 /" == 5.0
```

Soluciones

```
valor :: String -> Float
valor = head . foldl aux [] . words
  where aux (x:y:ys) "+" = (x + y) : ys
        aux (x:y:ys) "-" = (y - x) : ys
        aux (x:y:ys) "*" = (x * y) : ys
        aux (x:y:ys) "/" = (y / x) : ys
        aux xs cs      = read cs : xs
```