Universidad Nacional Autónoma de México

Facultad de Ingeniería

Asignatura: Estructura de Datos y Algoritmos 1

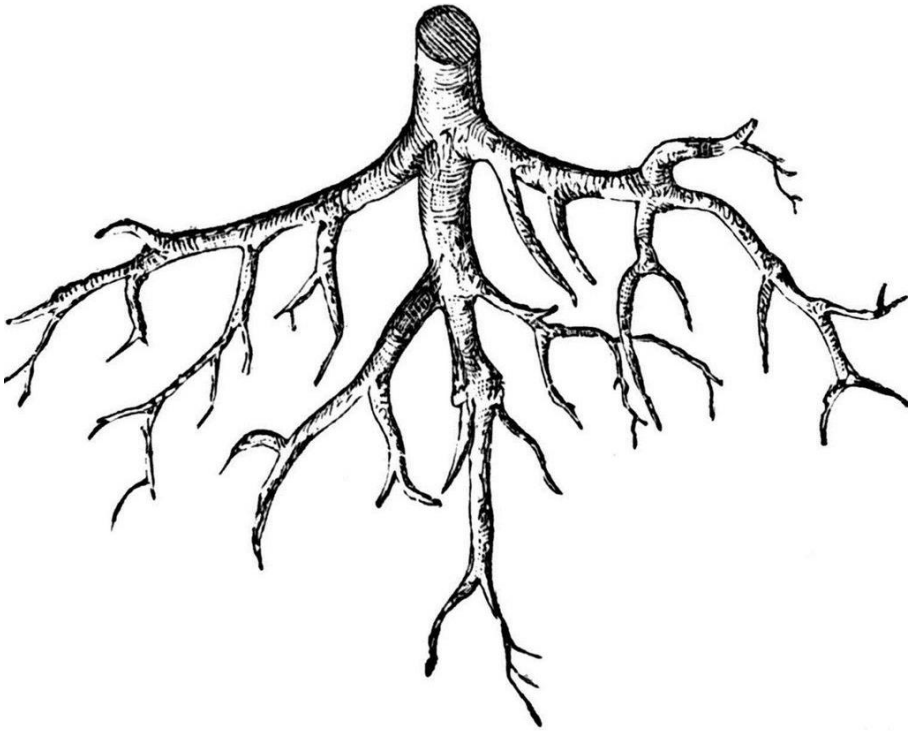
Actividad 4: Notación Polaca y Notación Polaca Inversa

Alumna: Hernández Vázquez Daniela

Profesor: M.I. Marco Antonio Martínez Quintana

Fecha: 28/06/2021

2021-2



Actividades:

- Realizar un trabajo de investigación escrito referente a la notación polaca y la notación polaca inversa
- Agregar su algoritmo de implementación para cada una

¿Qué es la notación polca?

Cuando se tiene una expresión de tipo B * C se denomina notación infija ya que el operador de multiplicación * se encuentra entre los dos operandos en los que está trabajando. Pero si ocurre que, por ejemplo, A + B * C. Los operadores + y * también aparecen entre los operandos, pero en este caso no se sabe con exactitud que operador trabaja primero si + o *, entonces, la expresión se vuelve ambigua.

Claro, nosotros sabemos que primero B es multiplicado por C y posteriormente sumado a A, podemos resolver la operación sin problema y esto es gracias a que nosotros sabemos que cada operador tiene un nivel o lugar en la jerarquía y entre mayor sea su grado dentro de esta se utilizan antes que los operadores de menor jerarquía. Lo único que puede cambiar ese orden es la presencia de paréntesis.

Ahora bien, existen otros formatos para ordenar la expresión. La notación polaca, también conocida como notación prefija, es una forma de representar una expresión aritmética, algebraica o lógica diferenciada por colocar los operadores a la izquierda de sus operandos + A B y se lee de derecha a izquierda. Mientras que la notación sufija o polaca inversa se caracteriza por tener los operadores a la derecha de los operandos A B + y se lee de izquierda a derecha.

Estas notaciones pueden leerse sin ambigüedad y sin la necesidad de recurrir a paréntesis y demás signos de agrupación, haciendo este tipo de escritura más compacta.

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

Para estos casos, menos la notación infija, ya no se requiere la intervención de símbolos adicionales pues los operadores ya no son ambiguos con respecto a los operandos en los que trabajan. El orden de las operaciones dentro de las expresiones de prefijo y de sufijo está completamente determinado por la posición del operador.

La notación de prefijo fue propuesta en 1924 por el matemático, lógico y filósofo polaco Jan Łukasiewicz (1878-1956), de allí el nombre alternativo por la que se conoce.

Łukasiewicz introdujo esta notación con la intención de simplificar la lógica proposicional. El matemático y lógico Alonzo Church la mencionaba en su libro clásico Introduction to Mathematical Logic (1956) como una notación digna de observación. Aunque dejó pronto de utilizarse en lógica, encontró su lugar en las ciencias de la computación. Por ejemplo, el lenguaje de programación LISP basa precisamente su sintaxis en la notación polaca.

Las calculadoras Hewlett-Packard usan la notación polaca inversa, económica en número de entradas, pero que requiere un esfuerzo adicional para la interpretación del resultado. Esta empresa utilizó este sistema por primera vez en 1968, en la calculadora de sobremesa HP-9100A. Y fue también ésta la notación de la primera calculadora científica de bolsillo, la HP-35, usada entre 1972 y 1975.

En ciencias de la computación, la notación de postfijo se usa en lenguajes de programación orientados a pila.

En algunos lugares nos mencionan que podemos definir la expresión polaca de una operación mediante la ramificación por impbolos y por jerarquías y leyendo estas ramificaciones en un sentido u otro para poder escribirla notación polaca o la notación polaca inversa. Pero que ocurre cuando no se puede descomponer una expresión gráfimente, es decir, la computadora no puede leer los simbolor y aomodarlos en jaraquías si es que no le damos antes un método e instrucciones a seguir.

Una vez que esté resuelto, el operador y los dos operandos son reemplazados por un nuevo operando. Puesto que un operador y dos operandos son eliminados y un operando es añadido, hay una pérdida neta de un operador y un operando, lo cual todavía deja una expresión con N operadores y N+1 operandos, permitiendo así que el proceso iterativo continúe.

Ésta es la teoría general tras el uso de stacks en lenguajes de programación para evaluar una sentencia en la notación de prefijo, aunque hay varios algoritmos que manipulan el proceso. Una vez que es analizada una sentencia en la notación de prefijo, llega a ser menos intimidante mientras que permite una cierta separación desde la convención con una añadida conveniencia. Un ejemplo muestra la facilidad con la

A continuación, se aplicarán estos conceptos a el algoritmo anteriormente desarrollado.

~Algoritmo Notación polaca~

*/ El siguiente algoritmo recibe una operación Infija (numérica, con los operadores entre los números), la convierte a pretfija (para que pueda ser interpretada por la computadora) y posteriormente la resuelve y devuelve el resultado.

Ejemplo Operación Infija: 3 + 5 * 10 Ejemplo Operación prefija: * + 3 5 10

1. Inicio
2. Crear 2 pilas
3. Introducir la operación o cadena de caracteres
4. Leer la operación termino a término de derecha a izquierda
   i. Si el elemento es un número se guarda en la primera Pila. Si no pase al punto 5
   ii. Si el elemento es un operador se guarda en la segunda Pila.
5. ° Caso1: Si el operador es un paréntesis cerrado ‘)’ se desapilan los elementos de la segunda pila y se apilan en la primera pila hasta que encuentre un paréntesis abierto ‘(’ pero evitando introducir este.
   ° Caso 2: Si el operador actual tiene mayor jerarquía que el ultimo operador de la pila 2 se desapilan los elementos de la segunda pila y se apilan en la primera pila hasta que el ultimo operador sea de menor o igual jerarquía del elemento actual o la pila quede vacía. Posteriormente se apila en la segunda Pila el elemento pendiente.
6. Los elementos que queden en la segunda pila se desapilan y se apilan en la primera.
7. Se lee termino a término la operación prefija de derecha a izquierda hasta llegar a un operador.
8. Cuando se encuentra con un operador se realiza la operación con los dos últimos números anteriores.
9. Imprime el resultado
10. Fin

~Algoritmo Notación polaca inversa~

*/ El siguiente algoritmo recibe una operación Infija (numérica, con los operadores entre los números), la convierte a postfija o sufija (para que pueda ser interpretada) y posteriormente la resuelve y devuelve el resultado.

Ejemplo Operación Infija: 3 + 5 * 10 Ejemplo Operación sufija: 3 5 10 * +

1. Inicio
2. Crear 2 pilas
3. Introducir la operación
4. Leer la operación termino a término de izquierda a derecha
   i. Si el elemento es un número se guarda en la primera Pila. Si no pase al punto 5
   ii. Si el elemento es un operador se guarda en la segunda Pila.
5. ° Caso1: Si el operador es un paréntesis cerrado ')' se desapilan los elementos de la segunda pila y se apilan en la primera pila hasta que encuentre un paréntesis abierto '(' pero evitando introducir este.
   ° Caso 2: Si el operador actual tiene mayor jerarquía que el ultimo operador de la pila 2 se desapilan los elementos de la segunda pila y se apilan en la primera pila hasta que el ultimo operador sea de menor o igual jerarquía del elemento actual o la pila quede vacía. Posteriormente se apila en la segunda Pila el elemento pendiente.
6. Los elementos que queden en la segunda pila se desapilan y se apilan en la primera.
7. Se lee termino a término la operación posfija (La operación creada a partir de los pasos anteriores) de izquierda a derecha hasta llegar a un símbolo
8. Cuando se encuentra con un operador se realiza la operación con los dos últimos números anteriores.
9. Imprime el resultado
10. Fin

Como podemos observar en la implementación de ambos algoritmos, vemos que son muy similares entre sí y lo que hace que su notación cambie es el orden en el cual se lee la expresión infija y el orden que se introducen los operandos y los operadores al interior de las pilas. Las jerarquías siguen teniendo su valor y de igual modo en ambos casos el operador se realiza con los dos últimos operandos que le anteceden.

Bibliografía

- El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.
- Anónimo. (2018). CALCULADORAS ¿Como Funcionan?. (20/06/21), de Usbac Sitio web: https://www.youtube.com/watch?v=ZBetobLYIBo
- Anónimo. (2018). CALCULADORAS ¿Como Funcionan?. (20/06/21), de Usbac Sitio web: GitHub - Usbac/Calculator: Calculator with GUI, hierarchy of operators and trigonometric functions.
- Universitat Politècnica de València - UPV (2018). S4.11 Notación polaca. | | UPV. (29/06/21), de Universitat Politècnica de València - UPV Sitio web: S4.11 Notación polaca. | | UPV - YouTube
- Anónimo. (2019). La notación polaca, la de Jan Łukasiewicz (29/06/21), de MATEMOCIÓN Sitio web: La notación polaca, la de Jan Łukasiewicz — Cuaderno de Cultura Científica (culturacientifica.com)
- Anónimo. (s.f). Notación infijo, prefijo y sufijo (29/06/21),Notación polaca | Portal OpenPyme
- Anónimo. (2019). Notación polaca (20/06/21), de Blogspot Sitio web: Lenguajes Y Autómatas II: Notación Polaca (lucyleonmoreno.blogspot.com)