

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Facultad de Ingeniería

Estructura de datos y
Algoritmos I

Actividad #4 Notación

Fausto Ángel Reséndiz Álvarez

Lunes 28 de junio del 2021

Semestre 2021 - 2



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 polaca o prefija

Es una forma de notación para la lógica, la aritmética, y el álgebra. Su característica distintiva es que coloca los operadores a la izquierda de sus operandos. Si la aridad de los operadores es fija, el resultado es una sintaxis que carece de paréntesis u otros signos de agrupación, y todavía puede ser analizada sin ambigüedad.

- Notación polaca inversa o postfija

Es un método algebraico alternativo de introducción de datos. Su característica es que coloca a cada operador antes de sus operandos, es decir, primero están los operandos y después viene el operador que va a realizar los cálculos sobre ellos.

- Nota:

“Tanto la notación polaca como la notación polaca inversa no necesitan usar paréntesis para indicar el orden de las operaciones mientras la aridad del operador sea fija.”

Algoritmo N. polaca

DATOS DE ENTRADA: $(1*(2-3)) + (4+5)$

DATOS DE SALIDA: $+*1-23+45$

Algoritmo:

1. Iniciar la pila
2. Leer cada dato de derecha a izquierda.
 - 2.1 Si el dato es un operador se guardará en la pila
 - 2.2 Si el dato es un operando se pondrá en el resultado
3. Se lee el primer dato que es un paréntesis
 - 3.1 Será guardado en la pila
4. Se lee el segundo dato que es un operando (5)
 - 4.1 se mostrará anotado de izquierda a derecha
5. Se lee el tercer dato que es un operador (+)
 - 5.1 Se guardará en la pila
6. Se lee el cuarto dato que es un operando (4)
 - 6.1 Se mostrará anotado de izquierda a derecha
7. Se lee el quinto dato que es un paréntesis
 - 7.1 Se guarda en la pila
 - 7.2 El signo más que esta en medio de los paréntesis es anotado de izquierda a derecha con los operandos
 - 7.3 Los paréntesis son descartados de la pila
8. Se lee el sexto dato que es un operador (+)
 - 8.1 Se guardará en la pila
9. Se lee el séptimo y octavo dato que son paréntesis
 - 9.1 Serán guardados en la pila
10. Se lee el noveno dato que es un operando (3)

-
- 10.1 Se mostrará anotado de izquierda a derecha
 11. Se lee el onceavo dato que es un operador (-)
 - 11.1 Se guarda en la pila
 12. Se lee el doceavo dato que es un operando (2)
 - 12.1 Se mostrará anotado de izquierda a derecha
 13. Se lee el treceavo dato que es un paréntesis
 - 13.1 Se guarda en la pila
 - 13.2 El signo menos que esta en medio de los paréntesis es anotado de izquierda a derecha con los operandos
 - 13.3 Los paréntesis son descartados de la pila
 14. Se lee el catorceavo dato que es un operador (*)
 - 14.1 Se guardará en la pila
 15. Se lee el quinceavo dato que es un operando (1)
 - 15.1 Se mostrará anotado de izquierda a derecha
 16. Se lee el dieciseisavo dato que es un paréntesis
 - 16.1 Se guardará en la pila
 - 16.2 El signo "por" que está en medio de los paréntesis es anotado de izquierda a derecha con los operandos
 - 16.3 Los paréntesis son descartados de la pila
 17. Vaciamos la pila al ya no haber paréntesis en la pila
 - 17.1 Colocamos los datos vaciados en el resultado
 18. Voltar el orden del resultado
 19. Finalizar proceso

Algoritmo N. polaca inversa

DATOS DE ENTRADA: $(1*(2-3)) + (4+5)$

DATOS DE SALIDA: 123-*45++

Algoritmo:

1. Iniciar la pila
2. Leer cada dato de izquierda a derecha
 - 2.1 Si el dato es un operador se guardará en la pila
 - 2.2 Si el dato es un operando se pondrá en el resultado
3. Se lee el primer dato que es un paréntesis
 - 3.1 Será guardado en la pila
4. Se lee el segundo dato que es un operando (1)
 - 4.1 Se mostrará anotado de izquierda a derecha
5. Se lee el tercer dato que es un operador (*)
 - 5.1 Se guardará en la pila
6. Se lee el cuarto dato que es un operando (2)
 - 6.1 Se mostrará anotado de izquierda a derecha
7. Se lee el quinto dato que es un operador (-)
 - 7.1 Se guarda en la pila
8. Se lee el sexto dato que es un operando (3)
 - 8.1 Se mostrará anotado de izquierda a derecha
9. Se lee el séptimo y octavo dato que son paréntesis
 - 9.1 Se guardarán en la pila

-
- 9.2 El signo menos que está en dentro de los paréntesis es anotado de izquierda a derecha con los operandos
 - 9.3 El signo “por” que está en dentro de los paréntesis es anotado de izquierda a derecha con los operandos
 - 9.4 Los paréntesis son descartados de la pila
 - 10. Se lee el noveno dato que es un operador (+)
 - 10.1 Se guardará en la pila
 - 11. Se lee el décimo dato que es un paréntesis
 - a. Se guarda en la pila
 - 12. Se lee el onceavo dato que es un operando (4)
 - a. Se mostrará anotado de izquierda a derecha
 - 13. Se lee el doceavo dato que es un operador (+)
 - 13.1 Se guardará en la pila
 - 14. Se lee el treceavo dato que es un operando (5)
 - 14.1 Se mostrará anotado de izquierda a derecha
 - 15. Se lee el catorceavo dato que es un paréntesis
 - 15.1 Se guardará en la pila
 - 15.2 El signo más que está en medio de los paréntesis es anotado de izquierda a derecha con los operandos
 - 15.3 Los paréntesis son descartados de la pila
 - 16. Vaciamos la pila al ya no haber paréntesis en la pila
 - 16.1 Colocamos los datos vaciados en el resultado
 - 17. Finalizar proceso

Referencias

- https://www.researchgate.net/profile/Reiner-Creutzburg/publication/277814853_Algoritmos_y_Estructura_de_Datos_Parte_7_Arboles_y_Graficas/links/558fdefd08ae1e1f9badf8e6/Algoritmos-y-Estructura-de-Datos-Parte-7-Arboles-y-Graficas.pdf