

Estrutura de Dados

Hamilton José Brumatto

Bacharelado em Ciências da Computação - UESC

23 de maio de 2016

Aplicações em Pilhas

Infixa, posfixa e prefixa

Modelo	Expressão
Infixa	$A + B$
Posfixa	$AB +$
Prefixa	$+AB$

Infixa, posfixa e prefixa

- A expressão infixa tem um formato visivelmente mais fácil de interpretar.
- A expressão infixa é ambígua, portanto é necessário estabelecer ordem de precedência, seja pela natureza da operação, seja pelo encapsulamento em parêntesis.
- As expressões posfixa e prefixa não são ambíguas, ao lado do operador estão dois operandos, se um deles é um operador, então este é realizado antes.
- As expressões posfixa e prefixa visualmente são mais complexas de se interpretar.

Ambiguidade

$$4/2/2$$

$$(4/2)/2 \neq 4/(2/2)$$

Precedência

- Considere as seguintes operações e ordem de precedência:
 - 1: \wedge (exponenciação)
 - 2: $*$ / (multiplicação e divisão)
 - 3: $+$ - (soma e subtração)
- Índice menor da operação indica precedência sobre operações de índices maiores, exponenciação tem a maior precedência.
- Dentro do mesmo índice a precedência ocorre na ordem em que aparecem. Assim, $4/2/2$ é interpretada como $(4/2)/2$.
- Exceto para a exponenciação em que a precedência é a ordem inversa: $A \wedge B \wedge C = A \wedge (B \wedge C)$
- Parêntesis sempre altera a ordem de precedência, as operações em parêntesis de nível mais interno devem ser realizadas primeiro.

Conversão: infixa \rightarrow posfixa

- A conversão ocorre na ordem da operação. $AopB \rightarrow ABop$
- Exemplo:

$$A + (B * C)$$

$$A + (BC*)$$

$$A(BC*)+$$

$$ABC * +$$

- Não é necessário os parêntesis na forma posfixa

Conversão: infixa \rightarrow posfixa

- A conversão ocorre na ordem da operação. $AopB \rightarrow ABop$
- Exemplo:

$$(A + B) * C$$

$$(AB+) * C$$

$$(AB+)C*$$

$$AB + C*$$

- Quando a precedência na infixa é diferente, na posfixa a forma é diferente.

Conversão: infixa \rightarrow prefixa

$$A + (B * C)$$

$$+A * BC$$

$$(A + B) * C$$

$$* + ABC$$

Exemplos de Conversão

Infixa	Posfixa
$A + B$	$AB +$
$A + B - C$	$AB + C -$
$(A + B) * (C - D)$	$AB + CD - *$
$A \wedge B * C - D + E / F / (G + H)$	$AB \wedge C * D - EF / GH + / +$
$((A + B) * C - (D - E)) \wedge (F + G)$	$AB + C * DE - - FG + \wedge$
$A - B / (C * D \wedge E)$	$ABCDE \wedge * / -$

Exemplos de Conversão

Infixa	Prefixa
$A + B$	$+AB$
$A + B - C$	$- + ABC$
$(A + B) * (C - D)$	$* + AB - CD$
$A \wedge B * C - D + E / F / (G + H)$	$+ - * \wedge ABCD // EF + GH$
$((A + B) * C - (D - E)) \wedge (F + G)$	$\wedge - + ABC - DE + FG$
$A - B / (C * D \wedge E)$	$- A / B * C \wedge DE$

Algoritmo de conversão usando pilha

- Lê-se uma expressão item a item
- Se for um operando ele é inserido direto na expressão posfixa
- Se for um abre parêntesis, ele é inserido na pilha
- Se for um fecha parêntesis, os operadores na pilha são removidos até o abre parêntesis
- Se for um operador:
 - Se no topo da pilha houver um operador de precedência menor, ou abre parêntesis, ele é inserido na pilha.
 - Senão (maior ou igual), o operador é removido até achar um de precedência menor, ou abre parêntesis ou pilha vazia. Então ele é inserido na pilha
- Se não houver mais itens na expressão, desempilha os operadores restantes.

Conversão - Exemplos

$$A + B * C$$

Lê	Posfixa	Pilha Operadores
A	A	
+	A	+
B	AB	+
*	AB	+*
C	ABC	+*
	ABC*	+
	ABC*+	

$$A * B + C$$

Lê	Posfixa	Pilha Operadores
A	A	
*	A	*
B	AB	*
+	AB*	+
C	AB * C	+
	AB * C+	

Conversão - Exemplos

$$A + B * C * D \wedge (E + F)$$

Lê	Posfixa	Pilha Operadores
A	A	
+	A	+
B	AB	+
*	AB	+*
C	ABC	+*
*	ABC*	+*
D	ABC * D	+*
\wedge	ABC * D	+* \wedge
(ABC * D	+* \wedge (
E	ABC * DE	+* \wedge (
+	ABC * DE	+* \wedge (+
F	ABC * DEF	+* \wedge (+
)	ABC * DEF+	+* \wedge
	ABC * DEF + \wedge	+*
	ABC * DEF + \wedge *	+
	ABC * DEF + \wedge * +	

Calculadora posfixa

- É muito comum encontrarmos calculadoras baseadas em expressões posfixa para sua operação.
- A Hewlet Packard (HP) tradicionalmente tem uma linha de calculadora que opera no modelo posfixo.
- A avaliação de uma expressão posfixa ocorre em um modelo de pilha.
- Se um item é um operando, ele é inserido na pilha.
- Se um item é um operador, retira dois operandos do topo da pilha, realiza a operação e empilha o resultado.

Exemplo

$$623 + -382 / + * 2 \wedge 3 +$$

Lê	op1	op2	res	Pilha
6				6
2				6,2
3				6,2,3
+	2	3	5	6,5
-	6	5	1	1
3				1,3
8				1,3,8
2				1,3,8,2
/	8	2	4	1,3,4
+	3	4	7	1,7
*	1	7	7	7
2				7,2
^	7	2	49	49
3				49,3
+	49	3	52	52

Algoritmo de Calculadora

```
Algoritmo CALCULADORA(expr)
  enquanto expr  $\neq$  vazia faça
    val  $\leftarrow$  Proximo(expr)
    se Éoperador(val) então
      op1  $\leftarrow$  Desempilha(S)
      op2  $\leftarrow$  Desempilha(S)
      res  $\leftarrow$  Calcula(op1,val,op2)
      Empilha(S,res)
    senão
      Empilha(S,val)
  retorne Desempilha(s)
```

Atividades

- Construa um algoritmo para a conversão de uma expressão infixa para uma expressão posfixa
- Construa um programa em C/C++ que realize esta função.
- Construa uma calculadora de expressão posfixa em C/C++