

**Estruturas de Dados e Introdução à Projeto e Análise de
Algoritmos - EDPAA
Laboratório 02 - Pilha
Prof. Luis Souza**

I. Contexto

A notação tradicional para expressões aritméticas, que representa uma operação binária na forma $x+y$, ou seja, com o operador entre seus dois operandos, é conhecida como notação in-fixa. Uma notação alternativa para esse tipo de expressão é a notação pós-fixa, na qual o operador é expresso após seus operandos. O atrativo da notação pós-fixa é que ela dispensa o uso de parênteses. Por exemplo, as expressões:

$a*b+c;$
 $a*(b+c);$
 $(a+b)*c;$
 $(a+b)*(c+d);$

seriam representadas nesse tipo de notação respectivamente como

$a\ b\ *\ c\ +$
 $a\ b\ c\ +\ *$
 $a\ b\ +\ c\ *$
 $a\ b\ +\ c\ d\ +\ *$

Expressões em formato intermediário usando a notação pós-fixa podem ser eficientemente resolvidas em máquinas baseadas em pilhas, também conhecidas como máquinas de zero endereços. Nesse tipo de máquina, operandos são explicitamente introduzidos e retirados do topo da pilha por instruções push e pop, respectivamente. Além disso, a aplicação de um operador retira do topo da pilha seus operandos e retorna ao topo da pilha o resultado de sua aplicação.

II. Implementação

Para a conversão de uma expressão em notação in-fixa para pós-fixa, segue-se o seguinte algoritmo (em meta-linguagem):

while (expressão não chegou ao fim):
 p = próximo elemento da expressão (operador ou operando)
 if (p é operando): coloque na pós-fixa
 if (p é operador):
 tire da pilha e coloque na pós-fixa todos os operadores
 com prioridade maior ou igual a p, na mesma ordem de
 retirada da pilha
 empilhe p;
 if (p é abre parêntesis): empilha p;
 if (p é fecha parêntesis):
 desempilhe os operadores até o primeiro abre e coloque
 na pós-fixa na mesma ordem de retirada da pilha;

desempilhe todos os operadores que ainda estão na pilha e coloque na
pós-fixa na mesma ordem de retirada da pilha

O comportamento da pilha para a conversão é representado nos exemplos abaixo.

a+b

	Pilha	Pós-fixa
a		a
+	+	a
b	+	ab
		ab+

a+b+c

	Pilha	Pós-fixa
a		a
+	+	a
b	+	ab
+	+	ab+
c	+	ab+c
		ab+c+

a*b+c

	Pilha	Pós-fixa
a		a
*	*	a
b	*	ab
+	+	ab*
c	+	ab*c
		ab*c+

a+b*c

	Pilha	Pós-fixa
a		a

+	+	a
b	+	ab
*	+*	ab
c	+*	abc
		abc*+

$a*(b+c)$

	Pilha	Pós-fixa
a		a
*	*	a
(*(a
b	*(ab
+	*(+	ab
c	*(+	abc
)	*	abc+
		abc+*

$(a+b)*c$

	Pilha	Pós-fixa
((
a		a
+	(+	a
b	(+	ab
)		ab+
*	*	ab+
c	*	ab+c
		ab+c*

Para a resolução de uma expressão em notação pós-fixa, segue-se o seguinte algoritmo:

```

while (expressão pós-fixa não chegou ao fim):
    seja p o próximo elemento da expressão
    if (p é operando): empilha p;
    if (p é operador unário):
        faz a operação com o elemento do topo da pilha
    if (p é operador binário):
        faz a operação com os 2 elementos do topo da pilha
        neste caso a pilha diminui de 1 elemento

```

Um exemplo de comportamento da pilha na resolução de uma expressão em notação pós-fixa pode ser observado abaixo.

Considerando a expressão aritmética (in-fixa) abaixo:

$$a * (b + c * (d + e))$$

Que na notação pós-fixa ficaria:

$$a b c d e + * + *$$

Supondo $a=1, b=2, c=3, d=4, e=5$:

a	b	c	d	e	+	*	+	*
				5				
			4	4	9			
		3	3	3	3	27		
	2	2	2	2	2	2	29	
1	1	1	1	1	1	1	1	29

Tarefa: Implementar a conversão de expressões (in-fixa -> pós-fixa) e a resolução de expressões em notação pós-fixa utilizando o template disponibilizado.