

## Avaliador de expressões aritméticas

A principal aplicação de linguagens livres de contexto é na construção de analisadores sintáticos, uma das principais ferramentas dos processos de compilação e interpretação. Uma grande parte das linguagens de programação atuais (C, C++, C#, Java, *Scheme*, PROLOG, dentre outras) possuem gramáticas livres de contexto como geradoras da linguagem. Para termos um exemplo disto, vamos considerar uma pequena gramática (escrita na notação BNF – *Backus-Naur Form*) para definição de expressões aritméticas e relacionais.

Uma especificação BNF é um conjunto de regras de derivação, escritas como:

$\langle \text{símbolo} \rangle ::= \langle \text{expressão com símbolos} \rangle$

Onde  $\langle \text{símbolo} \rangle$  (ou em maiúsculo) é um não terminal, e a expressão consiste em sequências de símbolos separadas pela barra vertical, '|', indicando uma escolha e  $\lambda$  representa a palavra vazia. Esta notação indica as possibilidades de substituição para o símbolo da esquerda. Os símbolos que nunca aparecem no lado esquerdo são ditos terminais. Considere a gramática abaixo:

$\langle \text{expr} \rangle ::= \langle \text{expr} \rangle + \langle \text{termo} \rangle \mid \langle \text{expr} \rangle - \langle \text{termo} \rangle \mid \langle \text{termo} \rangle$   
 $\langle \text{termo} \rangle ::= \langle \text{termo} \rangle * \langle \text{fator} \rangle \mid \langle \text{termo} \rangle / \langle \text{fator} \rangle \mid \langle \text{termo} \rangle \% \langle \text{fator} \rangle \mid \langle \text{fator} \rangle$   
 $\langle \text{fator} \rangle ::= \text{num} \mid (\langle \text{expr} \rangle)$

Considere que a produção inicial da gramática é o não terminal  $\langle \text{expr} \rangle$  e o terminal num pode ser dígitos de 0 | 1 | 2 | ... | 9

### Objetivo

O objetivo do trabalho é implementar um Analisador Sintático Descendente Recursivo (ASDR) para gramática acima, para tanto a gramática não pode ter recursividade à esquerda e deve estar fatorada, assim você deve reescrever as produções, eliminando a recursividade à esquerda, para que seja possível implementar o ASDR. Coloque no início da implementação do código fonte a gramática reescrita utilizada no ASDR.

As expressões serão testadas a partir da entrada pelo teclado e o seu programa deve decidir se a expressão está sintaticamente correta. Você pode assumir que cada palavra contém entre 1 e 50 caracteres do alfabeto = {0,1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, \*, /, %, (, )}.

Na saída, o seu programa deve informar na tela a expressão processada, a expressão na notação posfixa e o valor da expressão. Se a expressão não estiver sintaticamente correta, deve ser apresentada a expressão processada e a mensagem "Erro Sintático".

### Exemplos de entrada

5+2\*3  
5\*(2-4)/2)  
5-4\*((3)+2)  
4\*(9/3+2)-1  
)2\*5  
5+3%3

### Exemplos de saída

5+2\*3 => 523\*+=11  
5\*(2-4)/2) => Erro Sintático  
5-4\*((3)+2)=> 5432+\*--=15  
4\*(9/3+2)-1 => 493/2+\*1-=19 OK  
)2\*5 Erro Sintático  
5+3%3 => 533%+=5

### Observações importantes:

O programa deve estar bem documentado e pode ser feito em grupo de até **2 alunos**, não esqueçam de colocar o **nome dos integrantes** do grupo no arquivo fonte do trabalho. O trabalho será avaliado de acordo com os seguintes critérios:

- Funcionamento do programa, ou seja, programas com erros de compilação e não executando receberão nota 0 (zero);
- O programa deve estar na **linguagem C** e testados no compilador do **CodeBlocks**, caso programa apresentarem *warning* ao serem compilados serão penalizados;
- O quão fiel é o programa quanto à descrição do enunciado;
- Clareza e organização, programas com código confuso (linhas longas, variáveis com nomes não-significativos, etc.) e desorganizado (sem indentação, sem comentários, etc.) também serão penalizados;

Como este trabalho pode ser feito em **grupo**, evidentemente você pode “*discutir*” o problema dado com outros **grupos**, inclusive as “*dicas*” para chegar às soluções, mas você deve ser responsável pela solução final e pelo desenvolvimento do seu programa.