Análise de uma expressão aritmética

Segundo Tenembaum, “Estruturas de Dados Usando C, examine uma expressão matemática que inclui vários conjuntos de parênteses agrupados. Por exemplo: 7 - ((X * ((X + Y) / (J - 3)) + Y) / (4 - 2 . 5)).
Queremos garantir que os parênteses estejam corretamente agrupados, ou seja, desejamos verificar se:
1. Existe um número igual de parênteses esquerdos e direitos.
2. Todo parêntese da direita está precedido por um parêntese da esquerda correspondente.
Expressões como ((A + B) ou A + B{, violam o critério 1, e,
expressões como:)A + B(-C ou (A + B)) - (C + D violam o critério 2.
Para solucionar esse problema, imagine cada parêntese da esquerda como uma abertura de um escopo, e cada parêntese da direita como um fechamento de escopo.
Agora, alteremos ligeiramente o problema e suponhamos a existência de três tipos diferentes de delimitadores de escopo. Esses tipos são indicados por parênteses ((e)), colchetes ([e]) e chaves ({e}). Um finalizador de escopo deve ser do mesmo tipo de seu iniciador. Sendo assim, strings como: (A + B], [(A + B]), {A - (B]} são inválidas.
É necessário rastrear não somente quantos escopos foram abertos como também seus tipos. Estas informações são importantes porque, quando um finalizador de escopo é encontrado, precisamos conhecer o símbolo com o qual o escopo foi aberto para assegurar que ele seja corretamente fechado.
Uma pilha pode ser usada para rastrear os tipos de escopos encontrados. Sempre que um iniciador de escopo for encontrado, ele será empilhado. Sempre que um finalizador de escopo for encontrado, a pilha será examinada. Se a pilha estiver vazia, o finalizador de escopo não terá um iniciador correspondente e a string será, consequentemente, inválida. Entretanto, se a pilha não estiver vazia, desempilharemos e verificaremos se o item desempilhar corresponde ao finalizador de escopo. Se ocorrer uma coincidência, continuaremos. Caso contrário, a string será inválida. Quando o final da string for alcançado, a pilha deverá estar vazia; caso contrário, existe um ou mais escopos abertos que ainda não foram fechados, e a string será inválida. Veja a seguir o algoritmo para esse procedimento.

```
algoritmo analiseaExpressao;
variáveis:
   expressão : string;
   i : inteiro;
   símbolo : char;
   valido : boolean;
   p : Pilha
Início
  leia(expressão);
  i := 0;
  valido := true;
  enquanto  i < comprimento(Expressão) faca
  inicio_enquanto
       símbolo := expressão[i];
       se ( (símbolo='{') ou (símbolo='[') ou (símbolo='(') ) entao
         p.empilha(símbolo);
       senão
         se ((símbolo='}') ou (símbolo=']') ou (símbolo=')'))  entao
             se p.pilhaVazia() então
               valido:=false
             senão
               se (símbolo = '}') e (p.elementoDoTopo() = '{' ) então
                       p.desempilha()
               senão
                       se (símbolo = ']') e (p.elementoDoTopo() ='[' ) então
                     p.desempilha()
                       senão
                     se (símbolo = ')') e (p.elementoDoTopo() = '(' ) então
                       p.desempilha();
                         senão
                           break;
                         fimse
                   fimse
                 fimse
               fimse
         fimse
       fimse
       i := i + 1;
  fim_enquanto;
   Se p.pilhaVazia() e valido então
       Escrever('Expressão Correta')
   Senão
       Escrever('Expressão Incorreta')
   fimse
fim algoritimo.
```