Universidade de Brasília - Instituto de Ciências Exatas
Departamento de Ciência da Computação

Estruturas de dados - Trabalho 1

Avaliação de Expressões Aritméticas (forma posfixa)

Prof. Eduardo A. P. Alchieri
24 de agosto de 2017

1 Objetivo

O presente projeto tem como objetivo a concretização dos conceitos de pilhas, através da avaliação de expressões aritméticas na forma posfixa.

2 Descrição do Projeto

O projeto será desenvolvido em duplas, e é constituído de três partes:

1. Relatório (com no mínimo 5 páginas), contendo:

   - Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa.
   - Implementação: descrição sobre a implementação do programa. Deve ser detalhada a estrutura de dados utilizada (de preferência com diagramas ilustrativos), o funcionamento das principais funções e procedimentos utilizados, o formato de entrada e saída de dados, bem como decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado.
   - Estudo de Complexidade: estudo da complexidade do tempo de execução dos procedimentos implementados e do programa como um todo (notação O).
   - Listagem de testes executados: os testes executados devem ser simplesmente apresentados.
   - Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.

- Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso.

2. Código fonte;

3. Apresentação oral do projeto.

O relatório deverá ser entregue em uma via impressa e outra digital (arquivo em formato pdf).

2.1 Prazo para entrega da versão impressa

A versão impressa do relatório deverá ser entregue no início da aula do dia **15/09/2017**.

2.2 Prazo para entrega da versão digital

A versão digital do projeto (relatório + código fonte) deverá ser enviada via email até às 10h00 do dia **15/09/2017**. Email para envio: `alchieri@unb.br`

A apresentação oral será feita pelo grupo na data especificada em sala de aula. Cada grupo terá até 10 minutos para apresentar o trabalho, e todos os componentes do grupo devem participar da apresentação.

3 Avaliação de Expressões Aritméticas

Para representar as expressões aritméticas, usualmente usa-se a notação infixa, i.e., os operandos estão localizados entre os operadores como na expressão A + B. No entando, existem duas notações alternativas:

- Notação prefixa (notação polonesa): + A B
- Notação posfixa (notação polonesa reversa): A B +

Os prefixos "pre", "pos" e "in" referem-se à posição relativa do operador em relação aos dois operandos da expressão. O que dificulta a avaliação de expressões na forma infixa, por um sistema computacional, é a existência de prioridades entre os operadores, como por exemplo A + B * C, e a existência de parênteses que modificam esta prioridade, como por exemplo (A + B) * C. Estes problemas não são encontrados nas outras representações (prefixa e posfixa), uma vez que a ordem de execução das operações é estabelecida apenas pela ordem dos operandos e operadores na expressão.

Neste trabalho, desenvolveremos um programa para calcular expressões aritméticas da seguinte forma:

- O programa receberá como entrada uma expressão na forma (usual) infixa
- Primeiramente, o programa deverá analisar a validade da expressão infixa

- Caso seja válida, o programa transformará a expressão da forma infixa para a forma posfixa
- Finalmente, o programa calculará o resultado da expressão

A seguir, analizaremos cada uma destas etapas e, conforme descrito, em todas cada etapa do processamento usaremos uma pilha como suporte no desenvolvimento dos algoritmos.

3.1 Entrada e saída de dados

A expressão infixa de entrada pode ser recebida através da entrada padrão (console) através do comando *scanf* da linguagem de programação C. Por fim, o resultado final também pode ser impresso na saída padrão através do comando *printf*.

3.2 Validação da expressão

O segundo passo é analisar os delimitadores da equação (rastrear escopo).
Exemplo de expressão válida: ((A + B) * C + (D + E)/(F + G) + H) / I
Exemplo de expressão inválida: (A + B) * C + (D + E)/(F + G) + H) / I
Para isso, utilizaremos o seguinte algoritmo.

Algoritmo. Percorrer a expressão da esquerda para direita da seguinte forma:

- Inicie uma pilha vazia
- Se um inicializador de escopo for encontrado, o mesmo é empilhado
- Se um finalizar de escopo for encontrado, a pilha é verificada
  - Se estiver vazia, então a equação não é válida
  - Se não, desempilhar e comparar com o finalizador
- Ao final, a pilha deve estar vazia

3.3 Transformação da forma infixa para posfixa

O passo seguinte é transformar a expressão infixa em posfixa.
Exemplo de expressão infixa: A * B + C - (D/E + F)
Exemplo de expressão posfixa: A B * C + D E / F + -
Para esta transformação, utilizaremos o seguinte algoritmo.

Algoritmo. O Processo de conversão infixa para posfixa pode ser efetuado da seguinte forma:

- Inicie uma pilha vazia
- Realize uma varredura na expressão infixa, copiando todos os operandos encontrados diretamente para a expressão de saída
  - Ao encontrar um operador:
    - Enquanto a pilha não estiver vazia e houver no seu topo um operador com prioridade maior ou igual ao encontrado (trabalharemos apenas com os operadores +, -, /, *), desempilhe o operador e copie-o na saída
    - Empilhe o operador encontrado
  - Ao encontrar um parêntese de abertura, empilhe-o
  - Ao encontrar um parêntese de fechamento, remova um símbolo da pilha e copie-o na saída, até que seja desempilhado o parêntese de abertura correspondente
- Ao final da varredura, esvazie a pilha, movendo os símbolos desempilhados para a saída

Veja o funcionamento da versão final do algoritmo na conversão da expressão A*(B+C)/D.

3.4 Avaliação da expressão

Finalmente, a última etapa consiste na avaliação da expressão posfixa obtida anteriormente. Para isso, usaremos o seguinte algoritmo:

Algoritmo. Os componentes da expressão são processados da esquerda para a direita como a seguir:

- Inicie uma pilha vazia
- Se o próximo componente da expressão é um operando, o valor do componente é colocado na pilha
- Se o próximo componente da expressão é um operador, então os seus operandos estão na pilha. O número requerido de operandos é retirado da pilha, a operação específica é realizada, e o resultado é armazenado de volta na pilha.
- Ao final, a pilha conterá um único dado que é o valor final da expressão

Simule o funcionamento deste algoritmo com a expressão posfixa obtida da expressão infixa A*(B+C)/D.

DICA: Teste a implementação desenvolvida com a seguinte expressão: 3-1*2+3-1. O resultado correto é 3.