

Avaliação parcial – 1 ... prática

Considere a linguagem X25a que possibilita descrever algoritmos básicos envolvendo operações aritméticas e comandos de entrada e saída em um console. Um programa na linguagem X25a tem uma estrutura simples composta apenas por uma sequência de declarações separadas por vírgulas (“,”) sem procedimentos ou funções.

Todas as variáveis são inteiros, e declaradas simplesmente pela atribuição de valores, empregando o operador “:=”, ou leitura do console. Variáveis, ou identificadores, apresentam-se como sequências de uma a três letras minúsculas.

Tem apenas duas declarações de controle: “SE” e “FAÇA ... ENQUANTO”. As duas declarações podem conter sequências de declarações. A declaração “SE ... ENTÃO” tem como opcional uma parte “SENÃO” e precisa terminar com a palavra-chave “FIM”.

Há também declarações de entrada/saída: “LEIA” e “ESCREVA”, respectivamente para leitura de valores e impressão do conteúdo de uma variável ou mensagem de texto entre apóstrofes (‘...’).

Comentários aparecem entre colchetes (“[...]”) e não podem ser aninhados.

Expressões na linguagem X25a são limitadas a expressões de aritmética de inteiros e de booleanos.

Uma expressão booleana é composta por uma comparação de duas expressões aritméticas através de um dos dois operadores “<” e “=” (menor ou igual).

Uma expressão aritmética pode ter constantes, variáveis, parênteses e qualquer dos quatro operadores de inteiros, “+”, “-”, “*” e “/” com as propriedades usuais.

Expressões booleanas podem ser usadas somente como teste em declarações de controle, não há: variáveis booleanas, atribuições booleanas, ou entrada/saída de valores booleanos.

Exemplos de programas empregando a linguagem X25a:

[1º exemplo: calculo do fatorial de um número > 1]

```
LEIA num,
fat := num, [inteiro de entrada]
SE 1 < num ENTÃO
    fat := 1,
    FAÇA
        fat := fat * num,
        num := num - 1,
    ENQUANTO 1 < num,
    ESCREVA fat, [apresenta o fatorial]
FIM
```

[2º exemplo: lê 2 valores e apresenta o maior valor]

```
LEIA a,
LEIA b,
SE a < b ENTÃO
    ESCREVA b, [apresenta o maior “b”]
SENÃO
    SE b < a ENTÃO
        ESCREVA a, [apresenta o maior “a”]
    SENÃO
        ESCREVA ‘iguais’, [mensagem entre ‘...’]
FIM
FIM
```

[3º exemplo: menu para direcionar a recepção de um banco]

```
FAÇA
    ESCREVA ‘digite opção:’,
    ESCREVA ‘1= caixa, 2= contratos’,
    LEIA op,
    SE op = 1 ENTÃO
        ESCREVA ‘1 : guichês 7 a 15’,
        op := 0,
    SENÃO
        SE op = 2 ENTÃO
            ESCREVA ‘2 : guichês 20 e 23’,
            op := 0,
        FIM
    FIM
    ENQUANTO 0 < op
```

[4º exemplo: calcula a média de uma série de números,
obs.: a série de entrada encerra-se com valor zero.]

```
tot := 0,
n := 0,
FAÇA
    ESCREVA ‘digite número: ’,
    LEIA x,
    n := n + 1,
    tot := tot + x,
    ENQUANTO 0 < x,
    med := tot/x,
    ESCREVA ‘média =’,
    ESCREVA med,
FIM
```

Tarefa: implementações em linguagem C dos analisadores léxico e sintático descendente preditivo ...

a) Análise léxica: implementação do Autômato Finito Determinístico, referenciada como implementação “manual”, identificando os tokens nas marcas em negrito, palavras reservadas, variáveis (identificadores) apresentadas do texto introdutório e exemplos. Obs.: ... teste o analisador empregando arquivos com programas escritos na linguagem X25a, gerando um arquivo de saída com os tokens detectados. Apresente seu AFD desenvolvido para análise léxica de X25a.

b) Modele a gramática da linguagem X25a para viabilizar a análise sintática.

c) Implemente a análise sintática descendente para a linguagem X25a desenvolvendo o analisador LL(1). Obs.: no teste do analisador utilize o arquivo de saída do item (a) para comprovar se o programa em X25a está correto (ou reconhecido).

Obs.: todos os analisadores devem prever o tratamento de erros.