EXA869 - MI Processadores de Linguagem de Programação

Problema 2: Gramática

Cronograma

Sessão	Data	Assunto
1	02/09/21	Apresentação problema 2
2	09/09/21	Problema 2
3	14/09/21	Problema 2
4	16/09/21	Problema 2
5	21/09/21	Problema 2
6	23/09/21	Problema 2
7	28/09/21	Problema 2
8	30/09/21	Problema 2

Problema

Os alunos do MI de Processadores de Linguagens de Programação concluíram a primeira etapa do compilador, conforme requisitado pelo grupo de cientistas.

Na primeira etapa os alunos conseguiram separar os lexemas e classificá-los de acordo com as expressões regulares que definiam cada tipo de lexema.

A próxima etapa consiste na definição de uma linguagem que defina claramente como os comandos são construídos. Para tal, um documento contendo as especificações sintáticas desejadas para a linguagem foi encaminhado aos alunos. Bom trabalho!

Produto

O grupo tutorial deverá construir uma gramática **fatorada à esquerda, sem recursão à esquerda, respeitando a precedência e associatividade dos operadores aritméticos, lógicos e relacionais**, seguindo as especificações disponíveis no Anexo A. Somente as estruturas léxicas definidas no Problema 1 serão utilizadas na definição da gramática. A gramática deverá ser entregue por e-mail ao respectivo tutor até as **23h59m do dia 01/10/2021**. Haverá penalidade de 2 pontos por UM dia de atraso na entrega. Após este prazo, o trabalho não será mais aceito.

Recursos para Aprendizagem

- AHO, A. V., LAM, M. S., SETHI, R., ULLMAN, J. D. **Compiladores: Princípios, Técnicas e Ferramentas**, 2ª ed., Addison-Wesley, 2008.
- AHO, A. V.; SETHI, R., ULMMAN, J. D. **Compiladores: Princípios, Técnicas e Ferramentas.** Rio de Janeiro, LTC, 1995.
- LOUDEN, K. C. **Compiladores – Princípios e Práticas.** São Paulo, Thomson, 2004.
- HOPCROFT, J. E. *et al*. **Introdução à Teoria dos Autômatos, Linguagens e Computação**. 1ª edição, Editora Campus, 2002.
- MENEZES, P. F. B. **Linguagens Formais e Autômatos**. 5ª edição, Editora Sagra-Luzzatto, 2005.

Anexo A: Características gerais da linguagem

1 Todo o código-fonte deve ser feito em único arquivo.

2 Deve ser possível declarar múltiplas variáveis ou constantes do mesmo tipo na mesma linha.

3 As declarações de uma ou mais constantes devem ser feitas dentro de um bloco que se inicia com a palavra **constantes**.

4 As declarações de uma ou mais variáveis devem ser feitas dentro de um bloco que se inicia com a palavra **variaveis**.

5 O código-fonte pode ser divido em subprogramas chamados funções. Cada função deve conter inicialmente a palavra **funcao**, seguido do tipo de retorno, seguido do nome da função e seguido da lista de parâmetros formais. O valor de retorno da função se dará através da palavra **retorno**.

6 A função **algoritmo** é responsável por iniciar o programa e cada programa deve conter uma única função **algoritmo.**

7 A linguagem permite fazer a chamada de uma função A a partir de uma função B. Para isso, deve-se utilizar o nome de A seguido da lista de parâmetros reais separados por vírgula.

8 A linguagem permite o uso de vetores e matrizes. Deve ser possível acessar (ler ou inserir) os elementos de um vetor ou matriz.

9 A linguagem permite a sobrecarga de funções ou procedimentos, mas não a sobrescrita.

10 A linguagem permite a criação de tipos compostos com a palavra **registro**.

11 Delimitadores **{** e **}** marcam início e fim de blocos (de comandos, funções, etc.), respectivamente.

12 A linguagem permite o uso de expressões relacionais, lógicas e aritméticas. O operador lógico **!** é aplicável somente à variáveis do tipo booleano.

13 Uma expressão aritmética resulta em um único valor aritmético (**inteiro** ou **real**). Uma expressão relacional ou lógica resulta em um único valor booleano (**verdadeiro** ou **falso**).

14 Comandos:

- **se .. senao**: Na condição do comando só são permitidas as expressões relacionais, as expressões lógicas ou valores booleano. O comando **senao** não é obrigatório.
- **enquanto**: Na condição do comando só são permitidas as expressões relacionais, as expressões lógicas ou valores booleano.
- **para**: O comando exige três cláusulas: inicialização, parada e modificação. As cláusulas devem estar entre parênteses e separadas por ponto-e-vírgula. As { } delimitarão o conteúdo do comando. Nas condições de cada cláusula são requisitados comandos específicos:
    - Inicialização: São permitidas as expressões relacionais.
    - Parada: São permitidas as expressões relacionais, as expressões lógicas ou valores booleano.
    - Modificação: São permitidas as expressões relacionais ou expressões aritméticas.
- **escreva**: O comando iniciará com a palavra **escreva** e o que deverá ser impresso entre parênteses, finalizando com ponto-e-vírgula. O comando pode imprimir cadeias de caracteres, constantes, variáveis, elementos de vetores ou matrizes e elementos de tipos compostos. Múltiplas impressões no mesmo comando devem ser separadas por vírgula.
- **leia**: O comando iniciará com a palavra **leia** e o nome da variável (inclusive elementos de vetores ou matrizes e elementos de tipos compostos) entre parênteses, finalizando com ponto-e-vírgula. Múltiplas leituras no mesmo comando deverão ser separadas por vírgula.