



Instituto Federal de Educação, Ciência e Tecnologia
Curso: Bacharelado em Ciência da Computação
Disciplina: CompiladoresI
Professor: Mário Luiz Rodrigues Oliveira
Atividade: Trabalho Prático1
Formiga, MG, 25 de setembro de 2019

INSTRUÇÕES:

1. O trabalho pode ser realizado individualmente ou em grupo com no máximo 2 integrantes.
2. Caso você ache que falta algum detalhe nas especificações, você deverá fazer as suposições que julgar necessárias e escrevê-las junto à resolução do trabalho. Pode acontecer também que a descrição dessa atividade contenha dados e/ou especificações supérfluas para sua solução. Utilize sua capacidade de julgamento para separar o supérfluo do necessário.
3. Cada arquivo-fonte deve ter um cabeçalho constando as seguintes informações: nome(s) do(s) aluno(s), matrícula(s) e data.
4. Devem ser entregues apenas os arquivos contendo os códigos fontes. Compacte todos os arquivos gerados **num único arquivo no formato RAR**.
5. Entregue apenas uma resolução por grupo. A atividade deve ser entregue via portal acadêmico acessado pela **URL: <https://meu.ifmg.edu.br/>**.
6. O prazo final para entrega desta atividade é até **23:59:00** do dia **28/10/2019**.
7. O envio é de total responsabilidade do aluno. **Não serão aceitos trabalhos enviados fora do prazo estabelecido.**
8. **Trabalhos plagiados serão desconsiderados, sendo atribuída nota 0 (zero) a todos os envolvidos.**
9. O valor desta atividade é 30 pontos.



1. Objetivos e Descrição do Trabalho

O objetivo deste trabalho é o projeto e a implementação dos analisadores léxico, sintático, tabela de símbolos e tratamento de erros para a linguagem de programação **P**, gerada pela gramática especificada na seção 2. O trabalho deve ser implementado, obrigatoriamente, em uma das seguintes linguagens de programação: C (padrão ANSI), C++, Java ou Python 3.

O programa implementado deverá, obrigatoriamente, receber o arquivo a ser analisado como parâmetro na linha de comando e disponibilizar uma opção que permita gravar a tabela de símbolos em um arquivo texto. Para a impressão da tabela de símbolos o usuário deve passar o parâmetro `-t nomearquivo.txt` na linha de comando. Ao ser executado com esse parâmetro o programa deverá gravar a tabela de símbolos no arquivo indicado usando essa opção.

A seção 2 descreve os aspectos léxicos, sintáticos e semânticos da linguagem de programação **P**. Também na seção 2 são apresentadas as especificações para a tabela de símbolos e tratamento de erros.

2. Especificações da linguagem de programação P

2.1 Aspectos léxicos

- identificadores da linguagem de programação **P** começam com letra seguidos de quaisquer sequências de letras e/ou dígitos, ou seja, identificadores são expressos pela seguinte expressão regular `letra(letra|dígito)*`. Identificadores são representados por *TOKENS* da classe ID.
- a linguagem de programação **P** não é case sensitive;
- identificadores válidos possuem, no máximo, 32 caracteres;
- constantes numéricas inteiras são formadas por sequências de dígitos, ou seja, são representadas pela seguinte expressão regular `dígito(dígito)*`;
- constantes numéricas reais usam o '.' como separador e são representadas pela seguinte expressão regular `dígito(dígito)*.(dígito)*`;
- constantes numéricas inteiras e reais são representadas por *TOKENS* da classe CTE;
- considere que letras pertencem ao intervalo [A-Za-z] e dígitos pertencem ao intervalo [0-9];
- a linguagem de programação **P** admite comentários de linha e bloco, conforme especificados na linguagem de programação C. Não é permitido o aninhamento de comentários;

- cadeias de caracteres são formadas por quaisquer sequências de caracteres delimitados por aspas duplas. Cadeias de caracteres são representadas por *TOKENS* da classe CADEIA.
- são palavras reservadas da linguagem de programação **P**: PROGRAMA, VARIÁVEIS, INTEIRO, REAL, LÓGICO, CARACTER, SE, SENÃO, ENQUANTO, LEIA, ESCREVA, FALSO, VERDADEIRO. A classe de uma palavra reserva é a própria palavra reservada.
- os tipos básicos da linguagem de programação **P** são: INTEIRO, REAL, LÓGICO, CARACTER;
- o comando de atribuição é definido pelos símbolos := e é representado por *TOKENS* da classe ATRIB;
- operadores relacionais =, <, >, <=, >=, <> são representados por *TOKENS* da classe OPREL;
- os operadores + e - são representados por *TOKENS* da classe OPAD;
- os operadores * e / são representados por *TOKENS* da classe OPMUL;
- o operador de negação ! é representado por *TOKENS* da classe OPNEG;
- o separador ; é representado por *TOKENS* da classe PVIRG;
- o separador : é representado por *TOKENS* da classe DPONTOS;
- o separador , é representado por *TOKENS* da classe VIRG;
- o separador (é representado por *TOKENS* da classe ABREPAR;
- o separador) é representado por *TOKENS* da classe FECHAPAR;
- o separador { é representado por *TOKENS* da classe ABRECH;
- o separador } é representado por *TOKENS* da classe FECHACH.

2.2 Aspectos sintáticos

Os aspectos sintáticos da linguagem são definidos pela gramática livre de contexto G1 a seguir:

G1 = {{PROG, DECLS, C-COMP, LIST-DECLS, DECL-TIPO, D, LIST-ID, E, TIPO, LISTA-COMANDOS, G, COMANDOS, IF, WHILE, READ, ATRIB, WRITE, EXPR, H, LIST-W, L, ELEM-W, SIMPLES, P, R, TERMO, S, FAT}} {programa, id, variaveis, inteiro, real, logico, caracter, abrepar, fechapar, se, abrech, fechach, senao, enquanto, leia, atrib, escreva, cadeia, cte, verdadeiro, falso, oprel, opad, opmul, opneg, pvirg, virg, dpontos}, P, PROG}

P={
PROG → programa id pvirg **DECLS C-COMP**



DECLS $\rightarrow \varepsilon \mid \text{variaveis } \mathbf{LIST-DECLS}$
LIST-DECLS $\rightarrow \mathbf{DECL-TIPO D}$
D $\rightarrow \varepsilon \mid \mathbf{LIST-DECLS}$
DECL-TIPO $\rightarrow \mathbf{LIST-ID}$ dpontos **TIPO** pvirg
LIST-ID $\rightarrow \text{id } \mathbf{E}$
E $\rightarrow \varepsilon \mid \text{virg } \mathbf{LIST-ID}$
TIPO $\rightarrow \text{inteiro} \mid \text{real} \mid \text{logico} \mid \text{caracter}$
C-COMP $\rightarrow \text{abrech } \mathbf{LISTA-COMANDOS}$ fechach
LISTA-COMANDOS $\rightarrow \mathbf{COMANDOS G}$
G $\rightarrow \varepsilon \mid \mathbf{LISTA-COMANDOS}$
COMANDOS $\rightarrow \mathbf{IF} \mid \mathbf{WHILE} \mid \mathbf{READ} \mid \mathbf{WRITE} \mid \mathbf{ATRIB}$
IF $\rightarrow \text{se abrepar } \mathbf{EXPR}$ fechapar **C-COMP H**
H $\rightarrow \varepsilon \mid \text{senao } \mathbf{C-COMP}$
WHILE $\rightarrow \text{enquanto abrepar } \mathbf{EXPR}$ fechapar **C-COMP**
READ $\rightarrow \text{leia abrepar } \mathbf{LIST-ID}$ fechapar pvirg
ATRIB $\rightarrow \text{id atrib } \mathbf{EXPR}$ pvirg
WRITE $\rightarrow \text{escreva } \text{abrepar } \mathbf{LIST-W}$ fechapar pvirg
LIST-W $\rightarrow \mathbf{ELEM-W L}$
L $\rightarrow \varepsilon \mid \text{virg } \mathbf{LIST-W}$
ELEM-W $\rightarrow \mathbf{EXPR} \mid \text{cadeia}$
EXPR $\rightarrow \mathbf{SIMPLES P}$
P $\rightarrow \varepsilon \mid \text{oprel } \mathbf{SIMPLES}$
SIMPLES $\rightarrow \mathbf{TERMO R}$
R $\rightarrow \varepsilon \mid \text{opad } \mathbf{SIMPLES}$
TERMO $\rightarrow \mathbf{FAT S}$
S $\rightarrow \varepsilon \mid \text{opmul } \mathbf{TERMO}$
FAT $\rightarrow \text{id} \mid \text{cte} \mid \text{abrepar } \mathbf{EXPR}$ fechapar $\mid \text{verdadeiro} \mid \text{falso} \mid \text{opneg } \mathbf{FAT}$

2.3 Tabela de Símbolos

Exclusivamente para os objetivos desse trabalho armazene na tabela de símbolos apenas os identificadores e palavras reservadas da linguagem de programação **P**. Ademais considere as seguintes especificações:

- é de livre escolha do grupo decidir a estrutura de dados a ser utilizada para implementar a tabela de símbolos;
- também é de livre escolha do grupo a escolha do módulo que implementará a tabela de símbolos.

2.4 Tratamento de erros

Para tratamento de erros use o método pânico.



Instituto Federal de Educação, Ciência e Tecnologia
Curso: Bacharelado em Ciência da Computação
Disciplina: CompiladoresI
Professor: Mário Luiz Rodrigues Oliveira
Atividade: Trabalho Prático1
Formiga, MG, 25 de setembro de 2019

A seção 3 apresenta os critérios de correção desta atividade.

3. Critérios de Correção

Conforme descrito no plano de ensino, os critérios de avaliação do trabalho são:

1. somente serão corrigidos os trabalhos com códigos fontes portáteis e sem de erros de compilação;
2. análise código fonte: corretude, modularização, uso adequado de comentários, legibilidade e indentação do código; (10%)
3. execução correta numa bateria de testes práticos; (70%)
4. uso adequado de TAD e estruturas de dados; (10%), e
5. documentação (10%)

Cada discente (grupo) deverá entregar um relatório de forma a documentar os resultados obtidos nos trabalhos práticos. A critério exclusivo do docente poderá ser agendada uma apresentação oral do trabalho. Na ausência de plágio, as notas dos trabalhos serão computadas da seguinte forma: nota-apresentação * nota-trabalho, ou seja, a nota final é ponderada pela nota da apresentação.

Na seção 4 indica-se a bibliografia consultada para a confecção deste documento.

4. Bibliografia

AHO, A. V. et al. Compiladores. 2 ed. São Paulo: Pearson Addison-Wesley, 2008.