



Instituto Federal de Educação, Ciência e Tecnologia
Curso: Bacharelado em Ciência da Computação
Disciplina: Compiladores I
Professor: Mário Luiz Rodrigues Oliveira
Atividade: Trabalho Prático I
Formiga, MG, 19 de outubro de 2023

INSTRUÇÕES:

1. O trabalho deve, obrigatoriamente, ser realizado **individualmente**.
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. O trabalho pode ser desenvolvido utilizando uma das seguintes linguagens de programação: C (padrão ANSI), C++, Java ou Python 3.
4. Cada arquivo-fonte deve ter um cabeçalho contendo as informações conforme modelo **disponível no arquivo regras-conduta-trabalho-compiladores1 e divulgado conjuntamente com o enunciado deste trabalho**.
5. Devem ser entregues arquivos contendo os códigos fontes e também o relatório gerado na resolução do trabalho. Compacte todos os arquivos gerados **num único arquivo no formato RAR ou ZIP**.
6. O trabalho deve ser entregue, obrigatoriamente, via página da disciplina no **Google Classroom**.
7. O prazo final para entrega desta atividade é até **23:59:00** do dia **20/11/2023**.
8. O envio é de total responsabilidade do aluno. **Não serão aceitos trabalhos enviados fora do prazo estabelecido**.
9. Indique claramente as fontes consultadas na resolução deste trabalho.
10. **Trabalhos plagiados serão desconsiderados, sendo atribuída nota 0 (zero) a todos os envolvidos**.
11. O valor desta atividade é 30 pontos.



Instituto Federal de Educação, Ciência e Tecnologia
Curso: Bacharelado em Ciência da Computação
Disciplina: Compiladores I
Professor: Mário Luiz Rodrigues Oliveira
Atividade: Trabalho Prático I
Formiga, MG, 19 de outubro de 2023

1. Objetivos e Descrição do Trabalho

O objetivo deste trabalho é o projeto e a implementação dos analisadores léxico e sintático para a linguagem de programação **Z**, 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.

Após realizar as implementações e testes, produza um relatório contendo, obrigatoriamente:

1. decisões de implementação que foram importantes no desenvolvimento do trabalho;
2. o ambiente computacional utilizado e a descrição dos procedimentos de testes realizados;
3. o resultado dos testes realizados. Deixe claro o que está implementado e funcionando corretamente e também explicita os erros, se houver, e
4. a contribuição deste trabalho para o aprendizado do conteúdo da disciplina.

A seção 2 descreve os aspectos léxicos e sintáticos da linguagem de programação **Z**. Também na seção 2 são apresentadas as especificações para o tratamento de erros.

2. Especificações da linguagem de programação **Z**

2.1 Aspectos léxicos

- identificadores da linguagem de programação **Z** 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 **Z** 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)*`;



Instituto Federal de Educação, Ciência e Tecnologia
Curso: Bacharelado em Ciência da Computação
Disciplina: Compiladores I
Professor: Mário Luiz Rodrigues Oliveira
Atividade: Trabalho Prático I
Formiga, MG, 19 de outubro de 2023

- constantes numéricas reais usam o '.' como separador e são representadas pela seguinte expressão regular $[+-]? \text{dígito}(\text{dígito})^* . (\text{dígito})^*$;
- 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 **Z** 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 **Z**: ***program, var, int, real, bool, char, if, else, while, read, write, false, true***. A classe de uma palavra reserva é a própria palavra reservada.
- os tipos básicos da linguagem de programação **Z** são: INT, REAL, BOOL, CHAR;
- o comando de atribuição é definido pelo símbolo = 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 de programação **Z** são definidos pela gramática livre de contexto G_1 a seguir:

$G_1 = \{\{\text{PROG, DECLS, C-COMP, LIST-DECLS, DECL-TIPO, D, LIST-ID, E, TIPO, LISTA-COMANDOS,}$



Instituto Federal de Educação, Ciência e Tecnologia
 Curso: Bacharelado em Ciência da Computação
 Disciplina: Compiladores I
 Professor: Mário Luiz Rodrigues Oliveira
 Atividade: Trabalho Prático I
 Formiga, MG, 19 de outubro de 2023

G, COMANDOS, SE, ENQUANTO, LEIA, ESCREVA, ATRIBUICAO, EXPR, H, LIST-W, L, ELEM-W, SIMPLES, P, R, TERMO, S, FAT} {program, id, var, int, real, bool, char, abrepar, fechar, if, abrech, fechach, else, while, read, atrib, write, cadeia, cte, true, false, oprel, opad, opmul, opneg, pvirg, virg, dpontos}, P, PROG}

P = {
PROG → program id pvirg **DECLS C-COMP**
DECLS → ε | var **LIST-DECLS**
LIST-DECLS → **DECL-TIPO D**
D → ε | **LIST-DECLS**
DECL-TIPO → **LIST-ID dpontos TIPO pvirg**
LIST-ID → id **E**
E → ε | virg **LIST-ID**
TIPO → int | real | bool | char
C-COMP → abrech **LISTA-COMANDOS** fechach
LISTA-COMANDOS → **COMANDOS G**
G → ε | **LISTA-COMANDOS**
COMANDOS → **SE | ENQUANTO | LEIA | ESCREVA | ATRIBUICAO**
SE → if abrepar **EXPR** fechar **C-COMP H**
H → ε | else **C-COMP**
ENQUANTO → while abrepar **EXPR** fechar **C-COMP**
LEIA → read abrepar **LIST-ID** fechar pvirg
ATRIBUICAO → id atrib **EXPR** pvirg
ESCREVA → write abrepar **LIST-W** fechar pvirg
LIST-W → **ELEM-W L**
L → ε | virg **LIST-W**
ELEM-W → **EXPR** | cadeia
EXPR → **SIMPLES P**
P → ε | oprel **SIMPLES**
SIMPLES → **TERMO R**
R → ε | opad **SIMPLES**
TERMO → **FAT S**
S → ε | opmul **TERMO**
FAT → id | cte | abrepar **EXPR** fechar | true | false | opneg **FAT**}

2.3 Tratamento de erros

O tratamento de erros será muito simples. Ao encontrar o primeiro erro o programa deverá encerrar o processo de análise e encerrar a execução.

A seção 3 apresenta os critérios de correção desta atividade.



Instituto Federal de Educação, Ciência e Tecnologia
Curso: Bacharelado em Ciência da Computação
Disciplina: Compiladores I
Professor: Mário Luiz Rodrigues Oliveira
Atividade: Trabalho Prático I
Formiga, MG, 19 de outubro de 2023

3. Critérios de Correção

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

1. somente serão corrigidos trabalhos de acordo com as especificações e com códigos portáteis e sem de erros de compilação;
2. análise código fonte: modularização, uso adequado de comentários, legibilidade e indentação do código (5%);
3. eficiência, clareza e concisão (5%);
4. uso correto dos conceitos de analisadores léxicos e sintáticos. Uso adequado TAD e estruturas de dados (20%);
5. execução correta numa bateria de testes práticos (60%), e
6. documentação (10%)

A critério exclusivo do docente poderá ser agendada uma apresentação oral dos trabalhos. Neste caso, é obrigatória a apresentação oral do trabalho por parte do discente e a nota final em cada trabalho será computada da seguinte forma: $\text{nota-apresentação} * \text{nota-pratica}$, ou seja, a nota final é ponderada pela nota da apresentação. A nota-pratica será computada conforme os critérios para correção dos trabalhos. Discentes ausentes na apresentação oral receberão nota **zero** nesta etapa da avaliação. Se não houver apresentação, a nota final em cada trabalho será igual a nota-pratica. É importante salientar que a nota final somente será válida na ausência de plágio. Sendo constatado plágio todos os envolvidos receberão nota **zero** nesta etapa da avaliação.

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.