Centro Universitário Norte do Espírito Santo
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
DEPARTAMENTO DE COMPUTAÇÃO E ELETRÔNICA

Cursos: Engenharia de Computação; Ciência da Computação
Disciplina: **Linguagens Formas e Autômatos**
Professor: Henrique Monteiro Cristovão
Período letivo: 2018/1

Trabalho Computacional

Um Scanner (Analisador Léxico) e um Parser (Analisador Sintático) para o Subconjunto de uma Linguagem de Programação

Objetivos

- Criar uma gramática para um subconjunto de uma Linguagem de Programação.
- Criar uma máquina de Moore para processar o subconjunto da linguagem.
- Aplicar a técnica da análise preditiva na conversão de gramáticas em algoritmos e implementar um parser (analisador sintático) e um scanner (analisador léxico) para o subconjunto da linguagem.
- Melhorar a compreensão de elementos da Hierarquia de Chomsky e suas relações, bem como a sua aplicabilidade na construção de partes de um compilador.

Organização metodológica

- Trabalho em grupo de 5 ou 6 alunos (mesmos grupos dos Mapas Conceituais);
- Cada grupo deve escolher uma linguagem de programação diferente dos outros grupos. Não é aceita linguagem muito próxima a C, tais como C++, C#, Java. As escolhas devem ser anunciadas no grupo do Whatsapp.
- Primeira entrega parcial:
    1. desenho da máquina de Moore (rascunho);
    2. gramática livre de contexto (ainda com conflitos e recursividade à esquerda).
- Segunda entrega parcial:
    3. máquina de Moore desenhada por processamento gráfico (consulte exemplo no apêndice A das notas de aula);
    4. analisador léxico implementado em Java, advindo de aplicação da técnica da análise preditiva sobre a gramática regular escrita a partir da Máquina de Moore apresentada na primeira entrega parcial;
    5. conjunto de testes sobre o analisador léxico;
    6. ambiente cooperativo de implementação e de controle de versões no GitHub (https://github.com/) com todos os componentes do grupo e o professor (Id: hmcristovao) – sugere-se o uso da IDE Eclipse com o plugin Egit (http://www.eclipse.org/egit/);
    7. a gramática livre de contexto sem conflitos, ambiguidade e recursividade à esquerda.
- Entrega final: dia **21/junho**:
    8. analisador sintático implementado em Java, com aplicação da técnica da análise preditiva sobre a Gramática Livre de Contexto apresentada na segunda entrega parcial;
    9. conjunto de testes sobre o analisador léxico e analisador léxico;
    10. conjunto de HTML da documentação gerada pelo Javadoc.

Descrição do trabalho

Desenvolver um software com interface gráfica para executar um analisador léxico e sintático em Java para reconhecer um subconjunto da Linguagem de Programação escolhida pelo grupo. Esse subconjunto deve possuir, no mínimo, as seguintes estruturas equivalentes a Linguagem like C:

1. Comando **while.**
2. Comando **do-while.**
3. Comando **for.**
4. Comando **if.** Sem 'else'.
5. Comando **atribuição** simples (não aceita atribuição composta).
6. **Expressões** simples (formadas por números reais, variáveis, parênteses balanceados, operadores aritméticos binários, operadores relacionais e operadores lógicos binários e unários.
7. **Blocos de comandos** formados apenas pelos comandos apresentados nos itens de 1 a 5.

Detalhes necessários na implementação

- A implementação deve possuir uma documentação oriunda da ferramenta Javadoc. Ela deve ser completa relacionando os métodos implementados, autores, etc. Os métodos referentes aos estados da Máquina de Moore não precisam ser documentados, mas, os métodos oriundos dos símbolos não terminais (variáveis) da gramática devem possuir, em sua documentação, o trecho da gramática utilizado.
- O analisador léxico deve ser implementado através da técnica da análise preditiva feita a partir da Gramática Regular/Máquina de Moore.
- Campos de classes devem ser, na medida do possível, privados e com acesso por interface get e set.
- Identificação do número da linha, número da coluna e do lexema que provocou o erro sintático. Para isto, é indicada a criação de uma classe, por exemplo 'Token'. Uma das funcionalidades dessa classe seria a exibição do nome do token o seu respectivo lexema em caso de erros léxicos.

Critérios avaliativos

- Máquina de Moore (0,3 pontos);
- Gramática livre de contexto (0,3 pontos);
- Analisador léxico (0,3 pontos);
- Analisador sintático (0,3 pontos);
- Legibilidade do código e documentação através do Javadoc (0,2 pontos).
- Trabalho em equipe e de forma cooperativa, utilização do ambiente GitHub (0,3 pontos).
- Interface gráfica (0,3 pontos).

Obs.: valor máximo igual a 2 pontos, atribuído de forma individual.