

Trabalho: Geradores de Analisadores Léxicos e Sintáticos

Linguagens Formais e Compiladores
Prof^a. Jerusa Marchi

1. Objetivo do Trabalho:

O objetivo deste trabalho é o desenvolvimento de um framework para gerar analisadores léxicos e sintáticos. Faremos a execução em 2 passos, análise léxica e sintática separadamente. Para a construção de um gerador de analisador léxico são necessários os seguintes algoritmos (Parte I do trabalho):

- (a) Conversão de Expressão Regular para Autômato Finito Determinístico (livro do Aho).
- (b) União de Autômatos via ϵ -transições
- (c) Determinização de Autômatos
- (d) Construção da TS (incluir palavras reservadas e outras informações pertinentes)
- (e) Reconhecimento de um texto fonte, gerando uma lista de tokens

O fluxo de execução da primeira parte é o seguinte:

- A interface **de projeto** deve permitir a inclusão de expressões regulares para todos os padrões de tokens;
- Para cada ER deve ser gerado o AFD corresponde
- Os AFD devem ser unidos
- O AFND resultante deve ser determinizado gerando a tabela de análise léxica
- A interface **de execução** deve permitir a entrada de um texto fonte
- O texto fonte será analisado e deve gerar um arquivo de saída com todos os tokens encontrados.

Para a construção de um gerador de analisador sintático são necessários os seguintes algoritmos (Parte II do trabalho), a depender do algoritmo de análise implementado:

- Se preditivo LL(1): Eliminação de recursão à esquerda, Fatoração, Cálculo de First e Follow, Geração da tabela de análise; Autômato de pilha para análise de sentenças
- Se SLR: Cálculo de First e Follow, Algoritmos correspondentes ao analisador SLR (conforme livro do Aho).

A interface **de projeto** deve receber e validar a Gramática Livre de Contexto que descreve a linguagem, identificando terminais (tokens) e não terminais. O fluxo de execução da segunda parte é o seguinte:

- Leitura token a token do arquivo resultante da parte 1
- Uso da tabela de análise para validação da sentença de entrada
- Saída: Mensagem validando ou invalidando o código

Observações:

- As notações devem ser as utilizadas em sala (para notacionar ϵ usar o ε);
- As tabelas de análise, tanto léxica quanto sintática, devem poder ser "visualizadas";
- **Este trabalho é justamente para aprender a lidar com Expressões Regulares. Logo, não deve-se fazer uso de bibliotecas de Regex.**

2. Realização:

O trabalho deverá ser realizado em grupos de no mínimo 2 e no máximo 4 integrantes, conforme estabelecido no link "Grupos para o trabalho de desenvolvimento" no moodle. Também deve ser disponibilizado um vídeo explicativo (como baixar, executar e usar framework) com exemplos para que o trabalho possa ser melhor avaliado.

3. Formato de Entrega:

As entregas serão parciais:

- Dia 06/06 - entrega da parte 1 - Parte referente às linguagens regulares - agendar apresentação
- Dia 11/07 - entrega da parte 2 - Parte referente às linguagens livres de Contexto - agendar apresentação

4. Avaliação:

O trabalho será avaliado pela robustez, corretude dos algoritmos e usabilidade. Além disso, quesitos como legibilidade do código e organização dos fontes também serão considerados.