

Trabalho: Manipulação de Linguagens Regulares e Linguagens Livres de Contexto

Linguagens Formais e Compiladores
Prof^a. Jerusa Marchi

1. Objetivo do Trabalho:

O objetivo deste trabalho é a implementação dos algoritmos relacionados à manipulação de Linguagens Regulares e Livres de Contexto. Tais algoritmos são úteis na implementação de Geradores de Analisadores Léxicos e Sintáticos ou na implementação dos próprios analisadores, servindo de arcabouço para o desenvolvimento de Compiladores.

2. Definição do Trabalho:

Implementar algoritmos para manipular Autômatos Finitos, Gramáticas Regulares, Expressões Regulares, Gramáticas Livres de Contexto e Autômatos de Pilha.

Os seguintes algoritmos devem ser implementados:

- (a) Conversão de AFND (com e sem ε) para AFD (0,5pt)
- (b) Conversão de AFD para GR e de GR para AFND (0,5pt)
- (c) Minimização de AFD (1,0pt)
- (d) União e interseção de AFD (1,0pt)
- (e) Conversão de ER para AFD (usando o algoritmo baseado em árvore sintática - Livro Aho - seção 3.9) (1,5pt)
- (f) Reconhecimento de sentenças em AF (0,5pt)
- (g) Reconhecimento de sentenças em AP (via implementação de uma tabela Preditivo LL(1)) - Além dos algoritmos relacionados a análise da sentença de entrada, devem ser implementados os algoritmos para cálculo dos conjuntos First e Follow, Fatoração e Eliminação de Recursão à esquerda (2,5pt)
- (h) Reconhecimento de sentenças em AP (via implementação de um SLR(1) ou LR Canônico) - Além dos algoritmos relacionados aos conjuntos canônicos e Autômato LR, devem ser implementados os algoritmos para cálculo dos conjuntos First e Follow (2,5pt)

Observações:

- As entradas podem ser feitas via arquivo ou interface, a critério do grupo;
- Os resultados intermediários devem poder ser salvos. Exemplo: um AFND convertido para AFD deve poder ser unido com outro AFD, ou ainda poder ser minimizado;
- Para épsilon use a notação ε ;

- **Este trabalho é justamente para aprender a lidar com Expressões Regulares. Logo, não deve-se fazer uso de bibliotecas de Regex.**
- AFs podem ser apresentados na forma de tabelas de transição e ou diagramas de transição, a escolha do grupo;
- Todos os AFs (intermediários ou resultantes) devem ser reutilizáveis (passíveis de Edição)

3. Realização:

O trabalho deverá ser realizado em grupos de no máximo 4 integrantes, conforme estabelecido no link "Grupos para o trabalho de desenvolvimento" no moodle.

4. Formato de Entrega:

A entrega será no dia Dia 18/06 - agendar apresentação.

A entrega também deve ser feita pelo moodle, enviando um link para os fontes (ou os próprios num arquivo zipado). O grupo deve prover toda a documentação necessária permitindo executar o sistema e testar (o grupo deve prover exemplos editáveis de testes). Os exemplos devem ser abrangentes e variados, permitindo testar todas as funcionalidades. A falta de exemplos adequados ou a falta de informação sobre a forma e/ou necessidades para instalação e execução acarretará prejuízo à nota.

Os nomes dos arquivos de teste devem ser intuitivos e devem estar localizados em um diretório "testes".

Deve haver dentro do diretório principal um arquivo README (txt ou pdf) contendo cabeçalho (instituição, departamento, nomes dos integrantes do grupo, data) e informações sobre os fontes (linguagem utilizada, detalhes sobre a modelagem e Estruturas de dados utilizadas, etc.) e detalhes sobre o uso da aplicação. A não entrega deste relatório também acarretará prejuízo à nota.

Todos os arquivos fontes devem ter cabeçalho e devem ser bem comentados.

5. Avaliação:

O trabalho será avaliado pela corretude dos algoritmos, usabilidade e robustez. Além disso, quesitos como legibilidade do código e organização dos fontes também serão considerados em menor grau (Critérios Subjetivos).

6. Somente um integrante deve enviar o trabalho no moodle.