

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:

Elaborar uma aplicação, com interface gráfica para facilitar a interação, para manipular Autômatos Finitos, Gramáticas Regulares, Expressões Regulares e Gramáticas Livres de Contexto e Autômatos de Pilha. A aplicação deve suportar:

- (a) Leitura, gravação e edição de AF, GR e ER (0,25pt)
- (b) Conversão de AFND (com e sem ε) para AFD (0,5pt)
- (c) Conversão de AFD para GR e de GR para AFND (0,5pt)
- (d) Reconhecimento de sentenças em AF (0,5pt)
- (e) Minimização de AFD (1,0pt)
- (f) União e interseção de AFD (1,0pt)
- (g) Conversão de ER para AFD (usando o algoritmo baseado em árvore sintática - Livro Aho - seção 3.9) (1,5pt)
- (h) Leitura, gravação e edição de GLC. (0,5pt)
- (i) Transformação de GLC para uma GLC na forma normal de Chomsky (0,5pt)
- (j) Eliminação de recursão à esquerda (1,0pt)
- (k) Fatoração (1,0pt)
- (l) Reconhecimento de sentenças em AP (via implementação de alguma das tabelas de análise) (1,75pt)

Observações:

- AFs podem ser apresentados na forma de tabelas de transição;
- Todos os AFs (intermediários ou resultantes) devem ser reutilizáveis (passíveis de Edição)
- Em Anexo são dados os padrões dos arquivos.

3. Realização:

O trabalho deverá ser realizado em duplas, conforme estabelecido no link "Grupos para o trabalho de Implementação" no moodle.

4. Formato de Entrega:

As entregas serão parciais:

- Dia 05/11 - entrega da parte 1 - itens (a), (b), (c), (d), (e), (f) e (g) da Definição do Trabalho
- Dia 02/12 - entrega da parte 2 - itens (h), (i), (j), (k) e (l) da Definição do Trabalho

A entrega deve ser feita pelo moodle em um arquivo <Nome1Nome2>.zip que quando descompactado gere um diretório <Nome1Nome2> e dois subdiretórios: Aplicação e Testes.

Os arquivos de teste devem ser na forma <teste_ifuncionalidade.ext>. Exemplo: teste1AfdAfd.jff. E devem seguir o formato especificado em anexo.

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.

Bibliotecas auxiliares, que não aquelas específicas de Autômatos, RegEx ou Gramáticas, podem ser utilizadas visando limpar o código (para fazer entrada e saída de dados, por exemplo). Todos os arquivos fontes devem ter cabeçalho e devem ser bem comentados.

IMPORTANTE! Trabalhos fora do formato ou que não sigam o padrão especificado para os arquivos de entrada e saída não serão avaliados e será atribuída nota 0.

5. Avaliação:

O trabalho será avaliado pela corretude dos algoritmos, usabilidade e robustez (70%). Além disso, quesitos como legibilidade do código e organização dos fontes (20%) também serão considerados (Critérios Subjetivos). Parte da nota final decorrerá da apresentação do trabalho completo que acontecerá no período de 03/12 a 10/12 durante os horários de aula. O agendamento dos horários será feito pelo moodle.

6. Nota:

A nota final do trabalho será atribuída individualmente. A nota final será dada por

$$\left(\sum_{f=a}^l \text{funcionalidade}(f) + \text{Critérios Subjetivos} + \text{Apresentação Individual} \right) / 3$$

1 Anexo I - Formato de entrada e saída dos AF

Os arquivos com Af devem seguir o padrão:

número de estados

estado inicial

estados finais

alfabeto

transições (uma por linha)

- Exemplo afd:

5
0
1,2
a,b
0,a,1
0,b,2
1,a,1
1,b,3
2,a,4
2,b,2
3,a,1
3,b,3
4,a,4
4,b,2

- Exemplo afnd sem ε -transições:

4
0
3
a,b
0,a,0-1
0,b,0
1,b,2
2,b,3

- Exemplo afnd com ε -transições (representada por &):

4
0
3
a,b,&
0,&,1-2

1,a,1
1,b,2
2,a,1
2,&,3
3,b,3

2 Anexo II - Formato de entrada de ER

Os arquivos com ER devem seguir o padrão:

def-reg1: ER1
def-reg2: ER2
...
def-regn: ERn

As ER devem aceitar grupos como [a-zA-Z] e [0-9] e os operadores usuais de * (fecho), + (fecho positivo), ? (0 ou 1).

- Exemplo 1:
digit: [0-9]
letter: [a-zA-Z]
id: letter(letter | digit)*
- Exemplo 2:
er: a?(a | b)+

3 Anexo III - Formato de entrada de Gramáticas

Os arquivos com GR e GLC devem seguir o padrão usual utilizado em sala com \rightarrow representando \rightarrow e $\&$ representando ε :

- Exemplo GR:
S \rightarrow aA | a
A \rightarrow bA | a
- Exemplo GLC:
S \rightarrow aSb | $\&$