

UNIVERSIDADE FEDERAL DA FRONTEIRA SUL
CIÊNCIA DA COMPUTAÇÃO – Campus Chapecó
CCR: Linguagens Formais e Autômatos

Descrição do PROJETO:

Construção de aplicação para geração, eliminação de epsilon transições, determinização e minimização de autômatos finitos.

Bráulio Mello
Última atualização: 08-05-18

Data limite de apresentação: até a penúltima semana letiva de aula.

Objetivo:

Entrada: arquivo com a relação de tokens e GRs de uma linguagem hipotética.

Saída:

Autômato Finito Determinístico (AFD), livre de épsilon transições, mínimo sem a aplicação de classes de equivalência entre estados.

Descrição:

A aplicação faz a carga de tokens (palavras reservadas, operadores, símbolos especiais, ...) e Gramáticas Regulares (GR) a partir de um arquivo fonte (texto). Exemplo de arquivo de entrada:

se
entao
senao

$\langle S \rangle ::= a\langle A \rangle \mid e\langle A \rangle \mid i\langle A \rangle \mid o\langle A \rangle \mid u\langle A \rangle$
 $\langle A \rangle ::= a\langle A \rangle \mid e\langle A \rangle \mid i\langle A \rangle \mid o\langle A \rangle \mid u\langle A \rangle \mid \epsilon$

Usar notação BNF para as GRs.

Para cada token e gramática, a aplicação gera o conjunto de transições rotuladas em um único AF durante o procedimento de carga. No AF, apenas o estado inicial é compartilhado entre diferentes tokens/gramáticas. Os demais estados são exclusivos para as transições dos demais símbolos dos tokens e/ou estados das GRs.

O AF será indeterminístico quando ocorrer uma ou mais situações em que dois tokens ou sentenças definidas por GR iniciam pelo mesmo símbolo.

Para os tokens e GR acima exemplificados, teremos o seguinte AFND:

δ	s	e	n	t	a	o	i	u
S	A, H	C, M			M	M	M	M
A		B						
*B								
C			D					
D				E				
E					F			
F						G		
*G								
H		I						
I			J					
J					K			
K						L		
*L								
*M		M			M	M	M	M

Neste AF exemplo, os estados finais e respectivos tokens que reconhecem são:

B: se

G: entao

L: senao

M: variavel

Eliminação de épsilon transições:

Após a construção do AF, eliminar as épsilon transições caso existam.

Determinização:

Aplicar o teorema de determinização para obter o AFD. A aplicação deve permitir o acompanhamento do processo de determinização e a visualização do AFD gerado.

Minimização:

O AFD resultante deve ser submetido ao processo de minimização, contudo, sem aplicar Classe de Equivalência. No AFD final os estados podem ser representados por números. Os símbolos podem ser representados pelo correspondente numérico de acordo com a tabela ASCII.

Estado de erro:

Ao final da minimização, acrescentar um último estado final. Este será o estado de erro. Todas as células da tabela de transição (AFD) não mapeadas devem ser ajustadas para levar (transição) ao estado de erro. Todas as transições a partir do estado de erro permanecem no estado de erro.

Entrega (até penúltima semana letiva de aula):

- Código fonte da aplicação
- Relatório, em formato de artigo, contendo: identificação autores, resumo, introdução, referencial teórico básico (conceitos essenciais para compreensão do trabalho e trabalhos correlatos), especificação e implementação da solução para gerar AFDs, conclusão e referencial bibliográfico.
- upload no moodle em arquivo único antes da apresentação
- a penúltima semana letiva de aula é a data limite, não a data de apresentação. O trabalho pode ser apresentado assim que estiver pronto no decorrer do semestre.

Apresentação e avaliação:

- Trabalho individual ou em duplas
- Aplicação em funcionamento: 50% da nota
- Apresentação (demonstração da aplicação e arguição): 50% da nota
- Resultados mínimos para que o trabalho possa ser apresentado: composição do AFND, determinização e relatório no formato de artigo.
- Qualidade da solução, requisitos contemplados, domínio do processo de especificação e implantação da aplicação, teor/clareza/conteúdo do artigo são os principais referenciais para composição da nota.