

Notas de aula

CCR: GEX101 - Linguagens formais e autômatos			Criado em: 09/08/21	Alterado em:09/08/21
Turma: 31508	Turno: Vespertino	Ano/Sem: 2021/1		
Encontro síncrono: 09/08/2021		Período Assíncrono: de 10/08//21 a 14/08/21		
Carga horária da semana: 5ha			Professor: Braulio Mello	
Conteúdo: Autômato Finito (AFD e AFND), Teorema de determinização e minimização (Projeto: implementar gerador de AFD. Entrada: Gramática regular e/ou tokens. Saída: Autômato finito determinístico. Tema pode ser substituído ao longo do semestre). Apresentação Projeto até a penúltima semana letiva de aula. Sinc(5h)				

Material de apoio

Aula síncrona na quarta dia 11/08: projeto prático

Discussão AvPEA-2

1) $L(G) = \{x \mid x \in a^i b^j d^k \text{ onde } i, j, k > 0, i > k\}$

$S ::= aSd \mid aAd$

$A ::= aA \mid aB$

$B ::= bB \mid b$

$S \rightarrow aSd$

$\rightarrow aaSdd$

$\rightarrow aaaAddd$

$\rightarrow aaaaBddd$

2) $L(G) = \{x \mid x \in (se, \text{então}, \text{senão}, op, opl, \text{faça}, \text{até}, \{, \})^* \text{ onde a linguagem permita estruturas aninhadas de condição e repetição}\}$

$EST ::= se\ opl\ \text{então}\ \{ EST \} \mid se\ opl\ \text{então}\ \{ EST \}\ \text{senão}\ \{ EST \} \mid \text{faça}\ \{ EST \}\ \text{até}\ opl \mid op$

$se\ opl\ \text{então}\ \{$

$se\ opl\ \text{então}\ \{ op \}\ \text{senão}\ \{$

$faça\ \{$

$se\ opl\ \text{então}\ \{ op \}$

$\}\ \text{até}\ opl$

}
}

EST \rightarrow se opl então { EST }

\rightarrow se opl então { se opl então { EST } senão { EST } }

\rightarrow se opl então { se opl então { op } senão { EST } }

\rightarrow se opl então { se opl então { op } senão { faça { EST } até opl } }

\rightarrow se opl então { se opl então { op } senão { faça { se opl então { EST } } até opl } }

\rightarrow se opl então { se opl então { op } senão { faça { se opl então { op } } até opl } }

Determinização e minimização (eliminação de inalcançáveis e mortos, classe de equivalência) de AFs

$S ::= aB \mid aC \mid bD \mid cC \mid d$

$A ::= bC \mid aD \mid aB$

$B ::= bD \mid cC \mid \varepsilon$

$C ::= dD$

$D ::= bB \mid bC$

Determinização do AFND

AFND

AFD

δ	a	b	c	d		δ	a	b	c	d
S	B, C	D	C	X		S	[BC]	D	C	X
*X	-	-	-	-		*[BC]	-	D	C	D
A	D, B	C	-	-		D	-	[BC]	-	-
*B	-	D	C	-		C	-	-	-	D
C	-	-	-	D		*X	-	-	-	-
D	-	B, C	-	-						

Inalcançáveis: não são alcançados, pelo fecho transitivo, a partir de S

mortos: A partir do estado A, não alcançam estados finais e A não é final. (pag 24 apostila)

$S = \{B, C, D, X\}$

$X = \{$

~~$A = \{D, B, C\}$~~

$B = \{D, C, B\}$

$C = \{D, B, C\}$

$D = \{B, C\}$

$S ::= 0B \mid 1A \mid 1 \mid \varepsilon$

$A ::= 0B \mid \varepsilon$

$B ::= 0C \mid 0 \mid 1D$

$C ::= 0B \mid 1A \mid 1$

$D ::= 1C \mid 1$

AFND

AFD

δ	0	1				δ	0	1			
$* \rightarrow S$	B	A,X				$* \rightarrow S$	B	[AX]			
$*X$	-	-				B	[CY]	D			
$*A$	B	-				$*[AX]$	B	-			
B	C, Y	D				$*[CY]$	B	[AZ]			
$*Y$	-	-				D	-	[CW]			
C	B	A,Z				$*[AZ]$	B	-			
$*Z$	-	-				$*[CW]$	B	[AZ]			
D	-	C, W									
$*W$	-	-									

Construir AFND, eliminar inalcançáveis e mortos, e determinar alcançáveis:

S {B, A, X, C, Y, D, W, Z

A {B

B {C, Y, D

C {B, A, Z

D {C, W

Vivos:

B {C, Y, D, **B, A, Z**, ... (caso não houvesse estado final nos alcançáveis de B, os alcançáveis de C também são de B)

Atividade orientada

Objetivo: Compreensão do processo de determinização e minimização de AF's

Construir o AF para cada uma das gramáticas a seguir, eliminar estados mortos e inalcançáveis e determinar os AFNDs:

(1)

$S ::= aS \mid aB$

$A ::= aC \mid bA$

$B ::= aS$

$C ::= bA \mid a$

(2)

$S ::= aB \mid cC \mid dC$

$A ::= bC \mid aE$

$B ::= bD \mid cC \mid dC$

$C ::= dD \mid \varepsilon$

$D ::= bB \mid bC$

$E ::= bB \mid cA$

Data/horário limite para entrega (upload no Moodle):

Entrega até 15/08/21 as 23h em arquivo pdf (upload moodle).

Atividade Avaliativa

PROJETO PRÁTICO:

Construção de uma aplicação para construção, determinização e minimização (eliminação de mortos e inalcançáveis) de autômatos finitos.

Objetivo:

Compreender o processo de determinização/minimização de AFND's. Implementar funcionalidade a ser utilizada na disciplina de compiladores para implementação de analisador léxico.

Descrição:

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

Saída: Autômato Finito Determinístico (AFD) e livre de estados mortos e inalcançáveis.

A aplicação executa 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
então
senao

$\langle S \rangle ::= e \langle A \rangle$
 $\langle A \rangle ::= n \langle B \rangle$
 $\langle B \rangle \dots$

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

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 são:

B: se
G: então
L: senao
M: variavel

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. 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 para apresentação. O trabalho pode ser apresentado assim que estiver pronto no decorrer do semestre.
- a última semana letiva de aula pode ser utilizada para apresentar o trabalho como oportunidade de recuperação de rendimento.

Apresentação e avaliação:

- Trabalho individual ou em duplas
- Aplicação em funcionamento e artigo: 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.

Avaliação:

Obrigatório: Construção e determinização do AF: 9,0

Minimização (funcionalidade para eliminar estados mortos e inalcançáveis): 1,0

Data/horário limite para entrega (upload no Moodle) e apresentação em sessão síncrona:

Entrega e apresentação até **13/10/21** às 23h.

Entrega e apresentação até **20/10/21** às 23h na modalidade de recuperação de rendimento.