Licenciatura em Engenharia Informática – DEI/ISEP
Linguagens de Programação 2006/07

Ficha 4

Autómatos Finitos Determinísticos

Objectivos:

- Validação de palavras utilizando Autómatos Finitos;
- Conversão de autómatos finitos não determinísticos (AFN) em autómatos finitos determinísticos (AFD);
- Minimização de Autómatos Finitos Determinísticos;
- Aprendizagem dos conceitos através da realização de exercícios;

4.1 Validação de palavras utilizando Autómatos Finitos

Considere o autómato finito não determinístico que permite reconhecer números binários terminados em "10" (anteriormente apresentado na ficha 3).

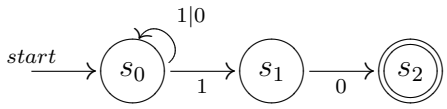


Figura 4.1: Representação gráfica do AF

Conforme descrito anteriormente, formalmente este autómato é definido da seguinte forma:

$$A = (S, \Sigma, s_0, F, \delta) = (\{s_0, s_1, s_2\}, \{0, 1\}, s_0, \{s_2\}, \delta)$$

onde,

$$\delta(s_0, 0) = \{s_0\}$$
$$\delta(s_0, 1) = \{s_0, s_1\}$$
$$\delta(s_1, 0) = \{s_2\}$$

Para verificar formalmente se uma determinada palavra é reconhecida pelo autómato, é necessário explorar todos os caminhos a partir do estado inicial. Para este fim é utilizada a transição estendida $\hat{\delta}$ que calcula o conjunto de estados passiveis de serem atingidos a partir de um determinado estado, após processar uma sequência de símbolos do alfabeto.

Note-se que a transição $\hat{\delta}(s_0, \varepsilon) = \{s_0\}$ indica que no caso de não ser consumido qualquer símbolo (representado por ε), não haverá lugar à transição de estado.

O reconhecimento da palavra "100110" é descrito através da seguinte sequência:

$$\hat{\delta}(s_0, \varepsilon) = \{s_0\}$$
$$\hat{\delta}(s_0, 1) = \delta(s_0, 1) = \{s_0, s_1\}$$
$$\hat{\delta}(s_0, 10) = \delta(s_0, 0) \cup \delta(s_1, 0) = \{s_0\} \cup \{s_2\} = \{s_0, s_2\}$$
$$\hat{\delta}(s_0, 100) = \delta(s_0, 0) \cup \delta(s_2, 0) = \{s_0\} \cup \emptyset = \{s_0\}$$
$$\hat{\delta}(s_0, 1001) = \delta(s_0, 1) = \{s_0, s_1\}$$
$$\hat{\delta}(s_0, 10011) = \delta(s_0, 1) \cup \delta(s_1, 1) = \{s_0, s_1\} \cup \emptyset = \{s_0, s_1\}$$
$$\hat{\delta}(s_0, 100110) = \delta(s_0, 0) \cup \delta(s_1, 0) = \{s_0\} \cup \{s_2\} = \{s_0, s_2\}$$

A representação gráfica da sequência de transições é apresentada na figura 4.2. Relembre-se que, o processamento bem sucedido da palavra implica o consumo integral de todos os símbolos que a compõe, e que após o consumo do último símbolo o autómato se encontre num estado final. Assim, a palavra em questão ("100110") poderia ser integralmente consumida no estado s_0, no entanto, como s_0 não se trata de um estado final esse caminho não seria válido, sendo apenas válido o caminho terminado em s_2.

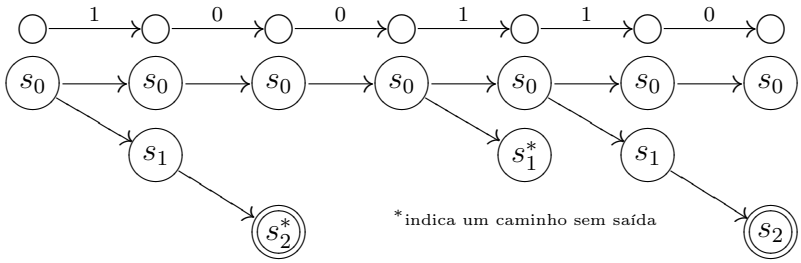


Figura 4.2: Representação gráfica da análise duma palavra pelo AF

4.2 Conversão dum AFN num AFD

A diferença mais relevante entre um AFN e um AFD consiste no facto de que, num AFD é sempre possível determinar qual é o estado para que o autómato transita após o consumo de um qualquer símbolo, pois o retorno da função de transição num AFD é um estado único, enquanto que num AFN é um conjunto de estados.

Este facto é verificável graficamente no autómato (**não determinístico**) da figura 4.1, pois no estado s_0 o consumo do símbolo "1" não determina objectivamente para que estado o autómato transitaria. Alternativamente este facto também é verificável na tabela de transições, pois o resultado dos pares (estado,símbolo) é um conjunto de estados, conforme se constata na tabela 4.1.

Tabela 4.1: Tabela de transições do AFN

	0	1
$\rightarrow s_0$	$\{s_0\}$	$\{s_0, s_1\}$
s_1	$\{s_2\}$	$\emptyset$
$*s_2$	$\emptyset$	$\emptyset$

De forma a converter um AFN num AFD, é necessário converter todos os conjuntos de estados em estados únicos. Desta forma, os n estados do AFN (tabela 4.1) darão lugar a 2^n estados no AFD (conforme tabela 4.2).

Tabela 4.2: Tabela de transições do AFD

	0	1
$\emptyset$	$\emptyset$	$\emptyset$
$\rightarrow \{s_0\}$	$\{s_0\}$	$\{s_0, s_1\}$
$\{s_1\}$	$\{s_2\}$	$\emptyset$
$*\{s_2\}$	$\emptyset$	$\emptyset$
$\{s_0, s_1\}$	$\{s_0, s_2\}$	$\{s_0, s_1\}$
$*\{s_0, s_2\}$	$\{s_0\}$	$\{s_0, s_1\}$
$*\{s_1, s_2\}$	$\{s_2\}$	$\emptyset$
$*\{s_0, s_1, s_2\}$	$\{s_0, s_2\}$	$\{s_0, s_1\}$

Nesta tabela a primeira coluna contém todas as combinações de estados possíveis para um autómato com n estados. Como os conjuntos que contenham um estado final, neste caso s_2, serão finais, devem por isso ser anotados em conformidade. As restantes colunas, cada uma das células contem a reunião dos estados para os quais o autómato transitaria estando em qualquer um dos estados apresentados na primeira coluna com o símbolo apresentado na primeira linha da matriz. Por exemplo, a transição $\hat{\delta}(\{s_0, s_1\}, 0)$ é calculada através de $\delta(s_0, 0) \cup \delta(s_1, 0) = \{s_0\} \cup \{s_2\} = \{s_0, s_2\}$

No passo seguinte, cada conjunto de estados será representado univocamente por um identificador, conforme se apresenta na tabela 4.3.

Tabela 4.3: Estados do AFD renomeados

	0	1
A	A	A
→ B	B	E
C	D	A
*D	A	A
E	F	E
*F	B	E
*G	D	A
*H	F	E

Nesta fase, a partir do estado inicial "B", são identificados os estados para os quais é possível transitar, neste caso B e E. Após o que, de forma iterativa, se procede à identificação dos estados para os quais se pode transitar a partir destes últimos, obtendo-se primeiro F e E, e posteriormente B e E a partir de F. Serão estas as únicas linhas a considerar da tabela, pois todas as outras correspondem a transições impossíveis. O resultado da aplicação deste algoritmo é apresentado na tabela 4.4.

Tabela 4.4: Tabela de transições final do AFD

	0	1
→ B	B	E
E	F	E
*F	B	E

Alternativamente, a tabela 4.2 poderia ter sido construída de forma iterativa a partir do estado inicial, expandindo apenas as transições válidas, evitando-se assim a representação de todos os estados inatingíveis. Isto é:

- Do estado $\{s_0\}$ são atingíveis os estados $\{s_0, s_1\}$ e o próprio ($\{s_0\}$);
- Expandindo o estado $\{s_0, s_1\}$ podemos chegar aos estados $\{s_0, s_2\}$ e ao próprio ($\{s_0, s_1\}$);
- Expandindo o estado $\{s_0, s_2\}$ podemos chegar aos estados $\{s_0\}$ e $\{s_0, s_1\}$.

O resultado destas iterações é apresentado na tabela 4.5.

Tabela 4.5: Tabela simplificada da conversão de um AFN num AFD

	0	1
$\rightarrow \{s_0\}$	$\{s_0\}$	$\{s_0, s_1\}$
$\{s_0, s_1\}$	$\{s_0, s_2\}$	$\{s_0, s_1\}$
$*\{s_0, s_2\}$	$\{s_0\}$	$\{s_0, s_1\}$

A representação gráfica do AFD obtido é apresentada na figura 4.3.

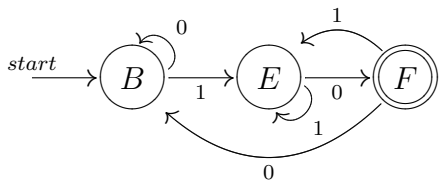


Figura 4.3: Representação gráfica do AFD

4.3 Minimização de Autómatos Finitos Determinísticos

O processo de minimização para um determinado AFD, pretende calcular de forma expedita o menor autómato possível **equivalente**. Considere-se, por exemplo, o autómato representado na figura 4.4, na exemplificação do algoritmo de simplificação.

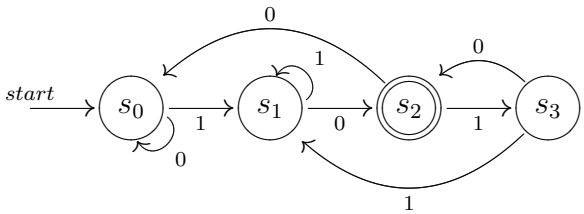


Figura 4.4: Representação gráfica do AFD a simplificar

Inicialmente é necessário construir a tabela de transições do autómato dividindo-a em dois grupos distintos, os estados finais e os não finais (ver tabela 4.6).

Tabela 4.6: Tabela de transições do AFD, 1º passo

	0	1
$\rightarrow s_0$	s_0	s_1
s_1	s_2	s_1
s_3	s_2	s_3
$*s_2$	s_0	s_3

No passo seguinte, cada grupo é por sua vez dividido em grupos cuja característica é cada símbolo implicar transições para o mesmo grupo. No exemplo da tabela 4.6, o grupo dos estados não finais pode ser dividido nos grupos (s_0) e (s_1, s_3), pois os estados s_1 e s_3 quando recebem o símbolo "0" transitam para s_2 que pertence ao grupo dos estados finais, enquanto que, quando recebem "1" transitam para o seu próprio grupo. O mesmo não se passa com o estado s_0 pois este com o símbolo "0" transita para próprio grupo. O resultado deste passo é apresentado na tabela 4.7.

Tabela 4.7: Tabela de transições do AFD, 2º passo

	0	1
$\rightarrow s_0$	s_0	s_1
s_1	s_2	s_1
s_3	s_2	s_3
$*s_2$	s_0	s_3

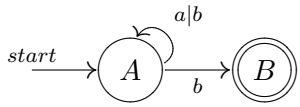
Este processo terá de ser repetido enquanto for possível proceder a subdivisões dos grupos existentes. No final é escolhido, de forma arbitrária, apenas um estado por cada grupo. Neste caso, no grupo (s_1, s_3), seria eliminado um dos estados, obtendo-se assim o autómato apresentado na tabela 4.8.

Tabela 4.8: Tabela de transições do AFD mínimo

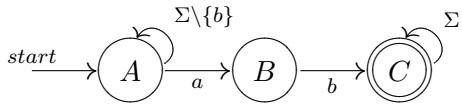
	0	1
$\rightarrow s_0$	s_0	s_1
s_1	s_2	s_1
$*s_2$	s_0	s_1

4.4 Propostas de exercícios

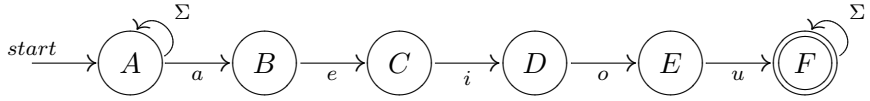
a) Converta os seguintes AFNs em AFDs.

1) 

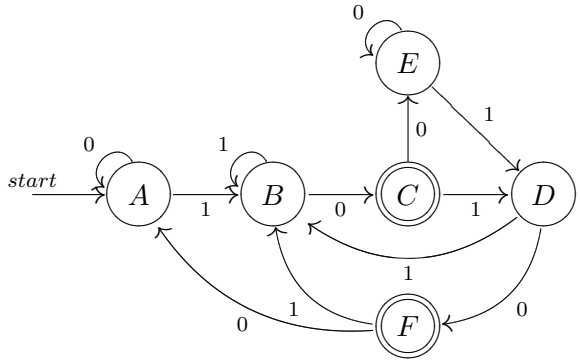
2) $\Sigma = \{a, b, c, d\}$; No autómato, $\Sigma \backslash \{b\}$ representa todo o alfabeto Σ excepto o conjunto $\{b\}$

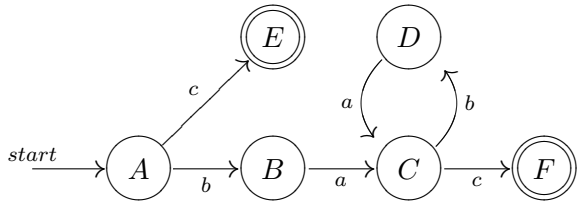


3) $\Sigma = \{a, b, c, d, \ldots, y, z\}$



b) Minimize os seguintes autómatos.

1) 

2) 

c) Implemente um autómato finito determinístico para cada uma das seguintes gramáticas:

1) Para o alfabeto $\Sigma = \{a, b, c, d\}$, qualquer palavra com um número par de símbolos "b".

2) Para o alfabeto $\Sigma = \{a, b, c, d\}$, qualquer palavra que contém a sequência "aab"

3) Para o alfabeto $\Sigma = \{0, 1\}$, qualquer palavra com um número par de zeros e um número par de uns.

d) Considere a seguinte tabela de transições:

	0	1
→ S	A	F
A	C	B
B	F	A
C	A	F
*F	B	C

1) Represente graficamente o autómato finito determinístico correspondente.

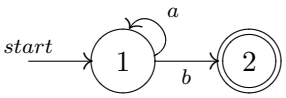
2) Verifique se as seguintes palavras pertencem à linguagem reconhecida pelo autómato:

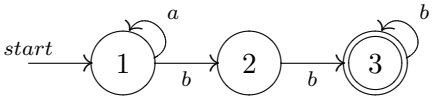
   i. 1
   ii. 0110110
   iii. 001
   iv. 1001
   v. 10101
   vi. 001101

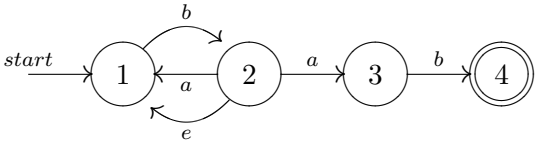
3) Minimize o autómato.

4) Descreva informalmente (por palavras) as linguagem representada pelo autómato.

e) Para cada um dos seguintes autómatos finitos determinísticos indique uma expressão regular que reconheça a mesma linguagem.

1) 

2) 

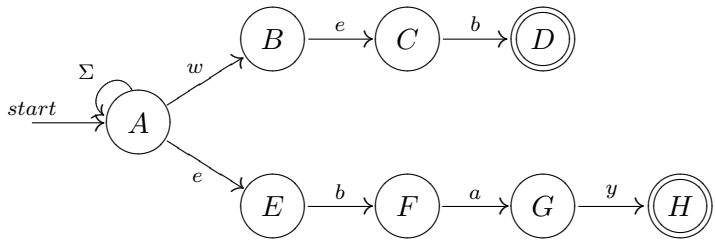
3) 

f) Implemente um autómato finito determinístico para cada uma das seguintes expressões regulares:

1) [0-9]+
2) [0-9]+"."[0-9]+
3) [f-o][k-z3-7]*
4) ("---"[a-z]*"\n")|(" "|"\n"|"\t")

g) Outros exercícios

1) Converta o AFN em AFD.
$\Sigma = \{a, b, c, d, \ldots, y, z\}$



2) Verifique se o autómato para reconhecimento de moedas criado na aula anterior está minimizado.

3) Represente as seguintes linguagens por uma expressão regular e por um AFD:

   i. Para o alfabeto $\Sigma = \{0, 1\}$
   $L(A) = \{u \in \Sigma^* : u$ começa por 1 e tem um número par de 1's$\}$

   ii. Para o alfabeto $\Sigma = \{+, -, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, e, i, .\}$
   $L(A) = \{u \in \Sigma^* : u$ é um número imaginário$\}$

   iii. Para o alfabeto $\Sigma = \{I, V, X, L, C\}$
   $L(A) = \{u \in \Sigma^* : u$ é um número romano menor que 100$\}$