

Licenciatura em Engenharia Informática - DEI/ISEP
Linguagens de Programação 2019/2020

Resolução da Ficha PL 1

Expressões Regulares e Autómatos Finitos

EXPRESSÕES REGULARES

1. Descreva informalmente (por palavras) as linguagens representadas pelas seguintes expressões regulares:

a) $0(0|1)^*0$

Um número binário com sufixo e prefixo 0

b) $(01)^*$

Uma string vazia ou uma sequência de repetições de "01"

c) $(0|1)^*0(0|1)(0|1)$

Números binários em que o bit 2 (contando a partir do zero) é sempre zero

d) $0^*10^*10^*10^*$

Números binários em que existam apenas três uns, e todos os outros dígitos (se existirem) sejam zero

e) $(0|1)^*111(0|1)^*$

Números binários em que haja pelo menos uma sequência de três uns ("111")

2. Considerando o alfabeto $\Sigma = \{A, B, C, \dots, Z, a, b, c, \dots, z\}$, represente as seguintes linguagens utilizando expressões regulares:

a) $L(A) = \{u \in \Sigma^* : u \text{ tem pelo menos dois a's juntos}\}$

$[A-Za-z]^*aa[A-Za-z]^*$

b) $L(A) = \{u \in \Sigma^* : u \text{ tem comprimento inferior a 4}\}$

$[A-Za-z]\{0,3\}$

c) $L(A) = \{u \in \Sigma^* : u \text{ começa por uma maiúscula}\}$

$[A-Z][A-Za-z]^*$

d) $L(A) = \{u \in \Sigma^* : u \text{ tem um número par de maiúsculas e um número ímpar de minúsculas, e tem comprimento múltiplo de 4}\}$

Impossível de representar expressão regular dado que a linguagem é vazia.

3. Diga se são verdadeiras ou falsas as seguintes afirmações:

a) $(0|1)^* \Leftrightarrow (0^*|1^*)$

Falsa. Primeira, 0 ou 1 qualquer número de vezes.

Segunda, só zeros ou só uns, qualquer número de vezes.

Válido Primeira e inválido na Segunda: 010101

b) $(0|1)^* \Leftrightarrow (0^*1^*)^*$

Verdadeira

4. Considerando o alfabeto $\Sigma = \{0,1\}$, represente as seguintes linguagens utilizando expressões regulares:
- $L(A) = \{u \in \Sigma^* : u \text{ é um número binário múltiplo de 4}\}$
 $[01]^*00$
 - $L(A) = \{u \in \Sigma^* : 111 \text{ não é factor de } u\}$
 $0^*(11?0+)^*1?1?$
 - $L(A) = \{u \in \Sigma^* : u \text{ tem um número par de 1's}\}$
 $(0^*10^*10^*)^+$
 - $L(A) = \{u \in \Sigma^* : u \text{ é vazia ou tem dígitos todos iguais, sendo de comprimento par as sequências de 0's, e ímpar as sequências de 1's}\}$
 $(00)^* \mid 1(11)^*$
 - $L(A) = \{u \in \Sigma^* : u \text{ é múltiplo de 4 mas não de 8}\}$
 $[01]^*100$
 - $L(A) = \{u \in \Sigma^* : u \text{ tem comprimento 3}\}$
 $[01][01][01]$ Alternativa: $[01]\{3\}$
5. Para a classificação dos alunos de uma escola, cada nota encontra-se codificada através do par NT, em que N é um valor inteiro entre 0 e 20 e T pode ser a letra F, ou E ou o par de letras RF. Uma nota não pode conter um zero na casa das dezenas. Uma sequência de notas, terá de conter pelo menos uma nota. Escrever uma expressão regular que valide uma sequência ilimitada de notas.
 $((20 \mid 1?[0-9])(E \mid R?F))^*$
6. Para garantir a rastreabilidade dos produtos fabricados numa determinada unidade industrial, cada lote de fabrico deverá ser codificado de forma a representar a informação necessária. Os campos que devem ser definidos nesta codificação são os definidos na tabela 1.1.

Tabela 1.1: Representação da codificação dos lotes de fabrico

Campo	Tamanho	Descrição
Identificação do lote	5	2 dígitos para o ano seguidos de 3 para o ID antecidos dos caracteres LT
Referência da peça	4	Caracteres alfanuméricos
Identificação da máquina	2	2 dígitos
Quantidade	5	5 dígitos
Data (Mês/Dia)	4	2 dígitos para o mês (01-12) seguidos de 2 para o dia (01-31)

Escreva uma Expressão Regular que permita representar a codificação dos lotes de fabrico. A ordem a seguir deve ser a definida na tabela anterior e deve ser usado o hífen como separador. Ver exemplo seguinte:

LT16121-3z1a-23-13450-1205

$LT[0-9]\{5\}-[a-zA-Z0-9]\{4\}-[0-9]\{2\}-[0-9]\{5\}-(0[1-9]|1[1-2])(0[1-9]|1[1-2][0-9]|3[0-1])$

7. Considerando A empresa DEI-TB precisa de apoio técnico para operacionalizar a informação apresentada no seu canal televisivo, durante os jogos de futebol do EURO2016. O sistema é suportado por um ficheiro TXT com informação dos vários jogos a transmitir. Entre outras, a informação comporta a estrutura PT~MM:SS~EQP>G-G<EQP, onde:

- PT, é opcional, e corresponde a uma das palavras 1P, 2P, 1EP ou 2EP
- MM corresponde à componente a minutos, podendo ter com 2 ou 3 dígitos (valores entre 00 e 120)
- SS corresponde a à componente segundos, com 2 dígitos (entre 00 e 59)
- EQP corresponde à sigla duma equipa, composta por 3 letras maiúsculas
- G corresponde aos golos marcados duma por uma equipa com valor inferior a 10

NOTA: o tempo máximo de jogo é 120 minutos!

Exemplos:

... 89:12~FRA>2-1<ROM ... 2P~28:06~ENG>1-0<RUS ...

... 1P~03:45~ESP>0-0<ROM ... EP~101:37~POR>4-2<ISL ...

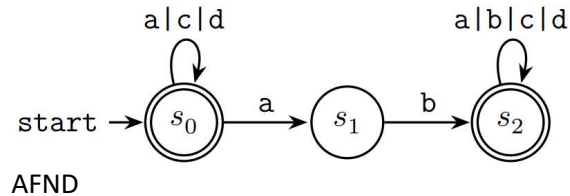
Construa uma expressão regular para captar a informação desejada.

$((1|2)E?P~)?((120:00)|(((0[0-9][0-9]|1[0-1][0-9]):([0-5][0-9]))~[A-Z]\{3\}>[0-9]-[0-9]<[A-Z]\{3\}$

AUTÓMATOS FINITOS

1. Para a linguagem constituída por todas as strings no alfabeto $\{a, b, c, d\}$ em que o primeiro "b" (se existir) é precedido de um "a";

a) Represente graficamente um autómato capaz de reconhecer a linguagem.



b) Defina formalmente o autómato A de acordo com a fórmula $A = (S, \Sigma, s_0, F, \sigma)$.

$A = (\{S_0, S_1, S_2\}, \{a,b,c,d\}, S_0, \{ S_2\}, \delta)$, onde

$\delta(S_0, a) = \{S_0, S_1\}$	$\delta(S_1, a) = \{S_1\}$	$\delta(S_2, a) = \{S_2\}$
$\delta(S_0, c) = \{S_0\}$	$\delta(S_1, b) = \{S_2\}$	$\delta(S_2, b) = \{S_2\}$
$\delta(S_0, d) = \{S_0\}$	$\delta(S_1, c) = \{S_0\}$	$\delta(S_2, c) = \{S_2\}$
	$\delta(S_1, d) = \{S_0\}$	$\delta(S_2, d) = \{S_2\}$

c) Elabore a tabela de transições respetiva.

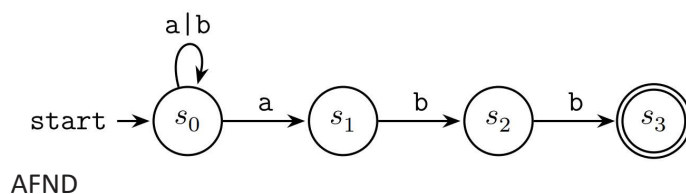
	a	B	c	d
$\rightarrow S_0$	$\{S_0, S_1\}$	\emptyset	$\{S_0\}$	$\{S_0\}$
S_1	$\{S_1\}$	$\{S_2\}$	$\{S_0\}$	$\{S_0\}$
$*S_2$	$\{S_2\}$	$\{S_2\}$	$\{S_2\}$	$\{S_2\}$

d) Represente essa linguagem usando uma expressão regular.

d) $[acd]^*(ab[abcd]^*)?$

2. Considere a seguinte expressão regular: $(a|b)^*abb$

a) Represente graficamente o autómato correspondente.



b) Defina formalmente o autómato A de acordo com a fórmula $A = (S, \Sigma, s_0, F, \sigma)$.

$A = (\{S_0, S_1, S_2, S_3\}, \{a, b\}, S_0, \{S_3\}, \delta),$ onde

$\delta(S_0, a) = \{S_0, S_1\}$

$\delta(S_0, b) = \{S_0\}$

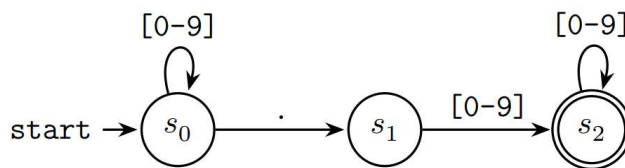
$\delta(S_1, b) = \{S_2\}$

$\delta(S_2, b) = \{S_3\}$

c) Elabore a tabela de transições respetiva.

	a	b
$\rightarrow S_0$	$\{S_0, S_1\}$	$\{S_0\}$
S_1	\emptyset	$\{S_2\}$
S_2	\emptyset	$\{S_3\}$
$*S_3$	\emptyset	\emptyset

3. Represente um autómato finito capaz de reconhecer números reais no seguinte formato: 'inteiro', 'inteiro'. O número real a reconhecer deverá ter sempre uma vírgula e uma casa decimal, não sendo obrigatória a existência de uma parte inteira.



AFD

4. Analise o seguinte código escrito em linguagem C e compare-o com o autómato desenvolvido no exercício 3, retire conclusões:

```

1 #include<stdio.h>
2
3 int main()
4 {
5     int a;
6     int real=1;
7     char estado='A' ;
8     int erro=0;
9
10    printf("Introduza dados:");
11    while ((a=getchar())!='\n') && (erro==0)
12        switch (estado)
13        {
14            case 'A' :
15                if (( a>='0') && (a<='9'))
16                    estado='A';
17                else
18                    if ( a=='.' )
19                        estado='B';

```

```

20                                     else
21                                     erro=1;
22     break;
23
24     case 'B' :
25         if((a>='0') && (a<='9'))
26             estado='C';
27         else
28             erro=1;
29     break ;
30
31     case 'C' :
32         if((a>='0') && (a<='9'))
33             estado='C' ;
34         else
35             erro =1;
36     break;
37 }
38 if ((!erro) && (estado=='C'))
39     printf ( "Foi lido um real.\n" ) ;
40 else
41     printf ("Não foi lido um real.\n" ) ;
42 return 0
43 }

```

Este código permite reconhecer números reais de acordo com o autómato finito representado no exercício 3. São representados os 3 estados através de um Switch-Case, permitindo fazer o reconhecimento do número real introduzido de acordo com a especificação do autómato finito; ou caso não seja reconhecido como válido, é impressa a informação de que não foi lido um número real.

5. As quantias monetárias podem ser expressas de diferentes formas.

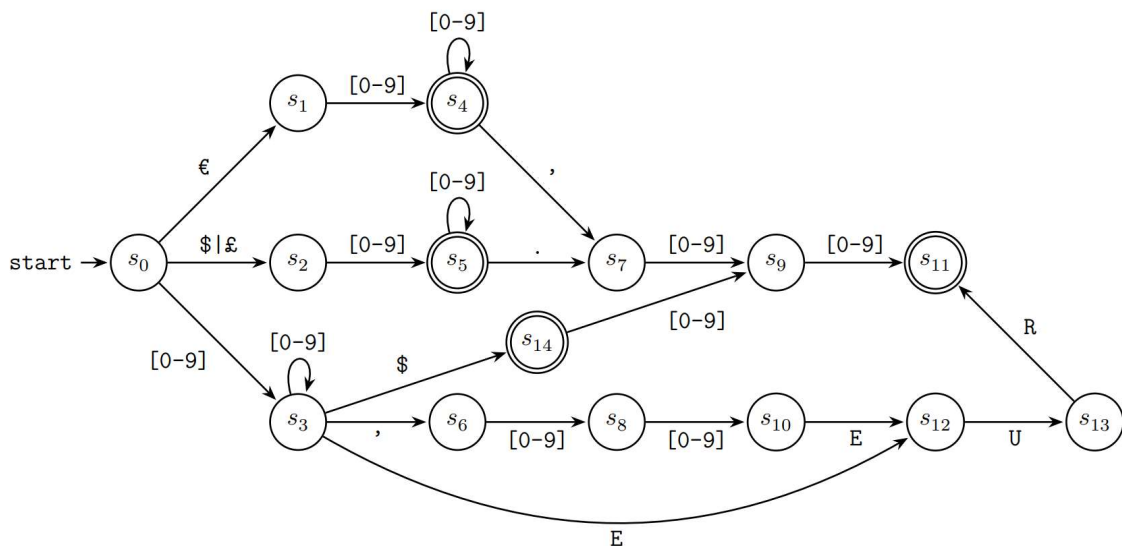
Tabela 2.1: Representação de moedas

Moeda	Exemplo
Euro	e12,23; e1,00; e2,35; 23,50EUR
Libra	£12.50; £22.12; £22.99
Dólar	\$25.13; \$5.00; \$0.30
Escudo	12\$50; 25\$00; 150\$00; 0\$50

- a) Escreva as expressões regulares que permitam realizar o reconhecimento das moedas apresentadas na tabela 2.1.

Euro (€[0-9]+([0-9]{2})?) | ([0-9]+([0-9]{2}))EUR
 Libra £[0-9]+([0-9]{2})?
 Dólar \$[0-9]+([0-9]{2})?
 Escudo [0-9]+\$([0-9]{2})?

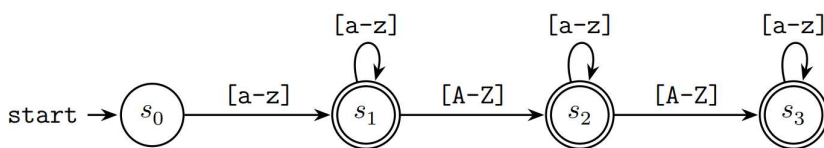
- b) Crie um AF capaz de identificar se uma quantia monetária está de acordo com a tabela 2.1.



AFD

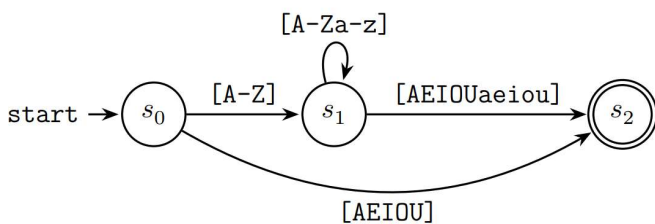
6. Considerando o alfabeto = {A,B,C,...,Z, a, b, c, ..., z}, represente as seguintes linguagens utilizando um AF:

- a) $L(A) = \{u \in \Sigma^* : u \text{ começa por um carácter minúsculo e tem no máximo duas maiúsculas}\}$



AFD

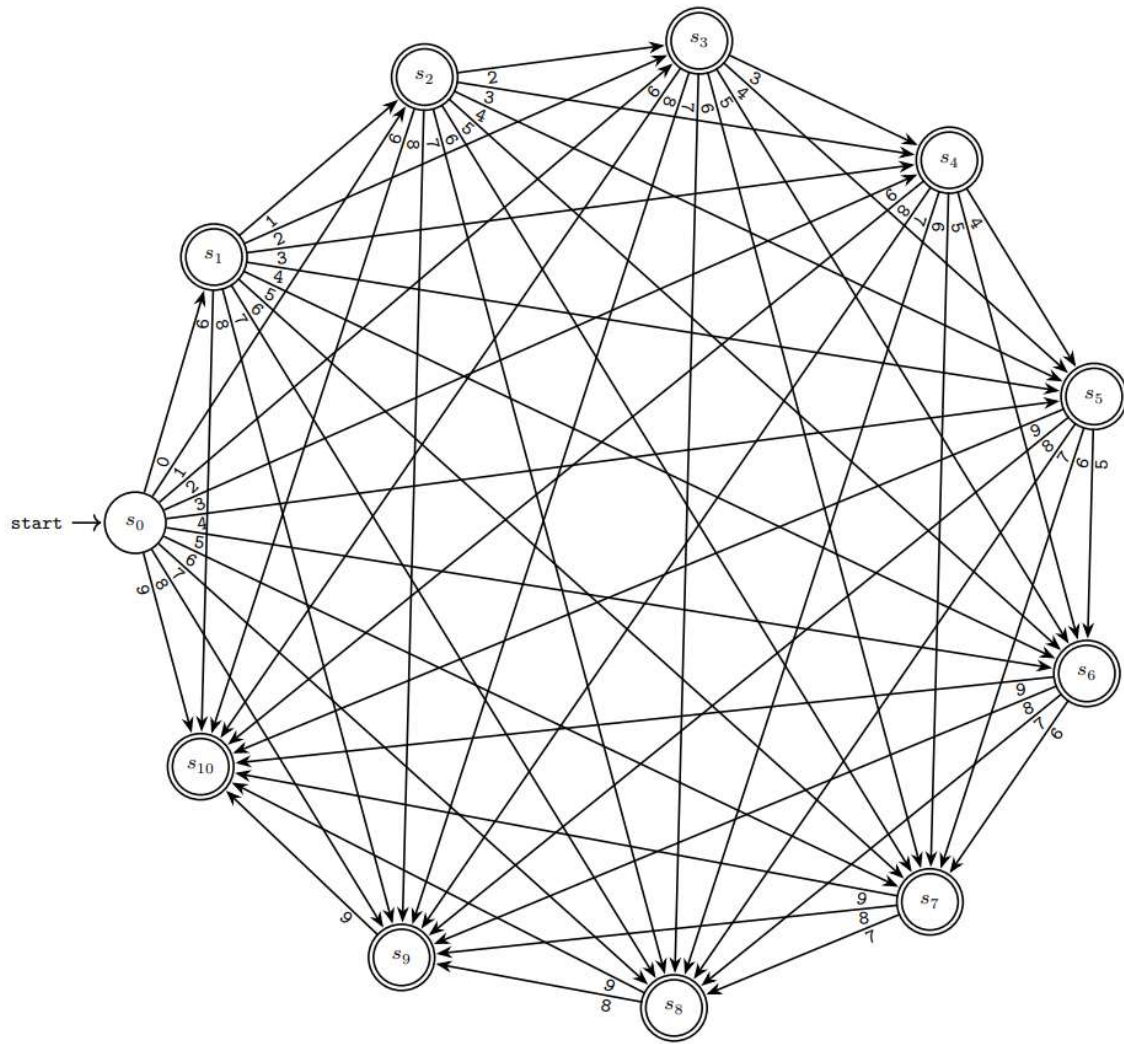
- b) $L(A) = \{u \in \Sigma^* : u \text{ começa por uma maiúscula e termina sempre numa vogal}\}$



AFND

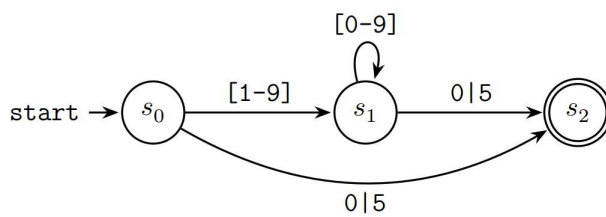
7. Considerando o alfabeto $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, represente as seguintes linguagens utilizando um AF:

- a) $L(A) = \{u \in \Sigma^* : u \text{ é uma sequência numérica crescente}\}$



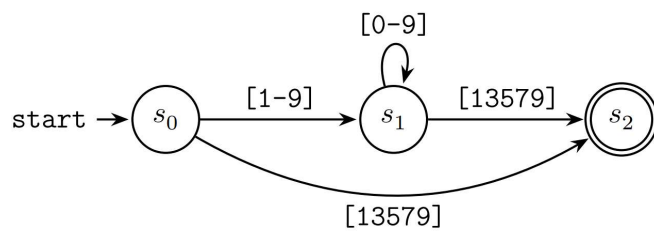
AFND

b) $L(A) = \{u \in \Sigma^* : u \text{ é múltiplo de } 5\}$



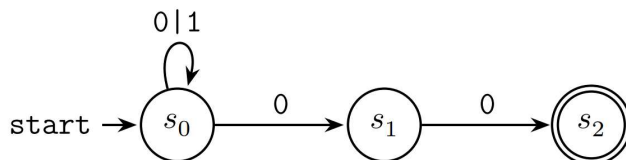
AFND

c) $L(A) = \{u \in \Sigma^* : u \text{ é um número ímpar}\}$



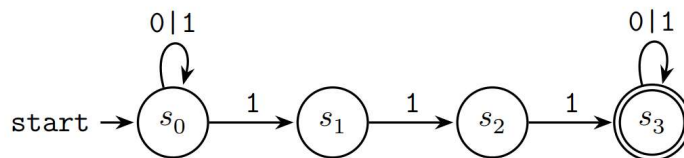
8. Considerando o alfabeto $\Sigma = \{0, 1\}$, represente as seguintes linguagens utilizando um AF. Classifique, justificando, os AF obtidos:

- a) $L(A) = \{u \in \Sigma^* : u \text{ é um número binário múltiplo de } 4\}$



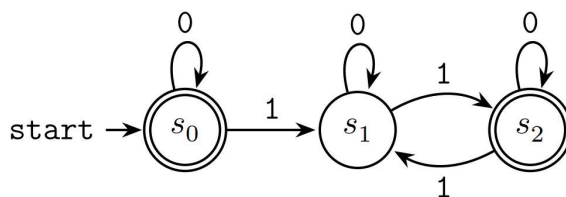
Não determinístico. No estado s_0 temos 2 transições possíveis perante o símbolo 0: para s_1 ou permanece em s_0 .

- b) $L(A) = \{u \in \Sigma^* : 111 \text{ é factor de } u\}$



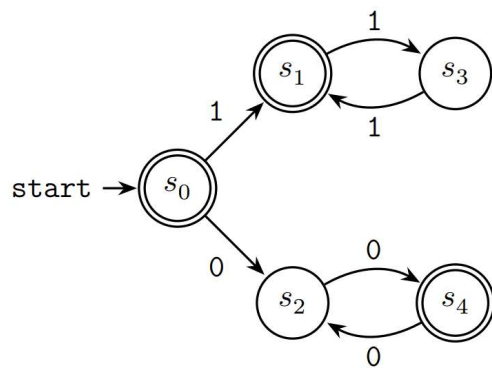
Não determinístico. No estado s_0 temos 2 transições possíveis perante o símbolo 1: para s_1 ou permanece em s_0 .

- c) $L(A) = \{u \in \Sigma^* : u \text{ tem um número par de 1's}\}$



Determinístico. Em cada estado existe apenas uma transição possível para cada valor recebido (0 ou 1).

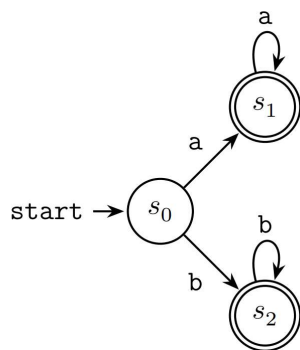
- d) $L(A) = \{u \in \Sigma^* : u \text{ é vazia ou tem dígitos todos iguais, sendo de comprimento par as sequências de 0's, e ímpar as sequências de 1's}\}$



Determinístico. Em cada estado existe apenas uma transição possível para cada valor recebido (0 ou 1).

9. Represente graficamente e através da tabela de transições, os AF capazes de reconhecer as seguintes expressões regulares e indique, justificando, se são determinísticos ou Não Determinísticos:

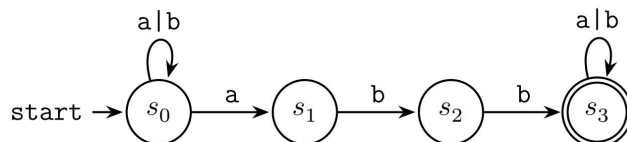
a) $aa^*|bb^*$



	a	b
$\rightarrow s_0$	$\{s_1\}$	$\{s_2\}$
$\star s_1$	$\{s_1\}$	\emptyset
$\star s_2$	\emptyset	$\{s_2\}$

Determinístico. Em cada estado existe apenas uma transição possível para cada valor recebido (a ou b).

b) $(a|b)^*abb(a|b)^*$

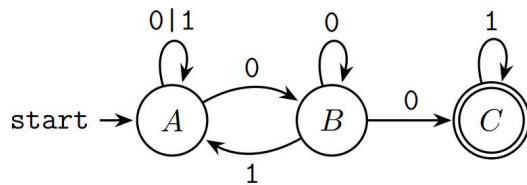


	a	b
$\rightarrow S_0$	$\{S_0, S_1\}$	$\{S_0\}$
S_1	\emptyset	$\{S_2\}$
S_2	\emptyset	$\{S_3\}$
$*S_3$	$\{S_3\}$	$\{S_3\}$

Não determinístico. No estado S_0 existem duas transições possíveis perante o símbolo **a** (permanecer em S_0 ou transitar para S_1).

10. Converta os seguintes AFN num AFD e minimize-os.

a)



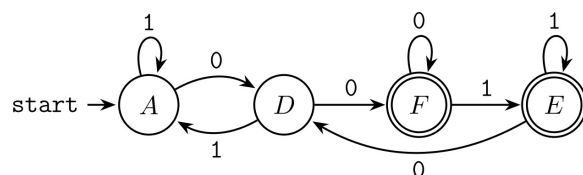
Processo completo:

	0	1
$\rightarrow \{A\}$	$\{A, B\}$	$\{A\}$
$\{B\}$	$\{B, C\}$	$\{A\}$
$*\{C\}$	\emptyset	$\{C\}$
$\{A, B\}$	$\{A, B, C\}$	$\{A\}$
$*\{A, C\}$	$\{A, B\}$	$\{A, C\}$
$*\{B, C\}$	$\{B, C\}$	$\{A, C\}$
$*\{A, B, C\}$	$\{A, B, C\}$	$\{A, C\}$

	0	1
$\rightarrow A$	D	A
B	F	A
$*C$	\emptyset	C
D	G	A
$*E$	D	E
$*F$	F	E
$*G$	G	E

	0	1
$\rightarrow A$	D	A
B	F	A
D	G	A
$*C$	\emptyset	C
$*E$	D	E
$*F$	F	E

	0	1
$\rightarrow A$	D	A
D	F	A
$*E$	D	E
$*F$	F	E

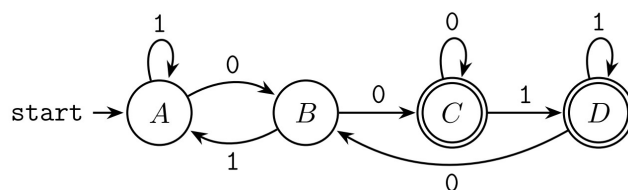


AFD

Processo simplificado e minimização:

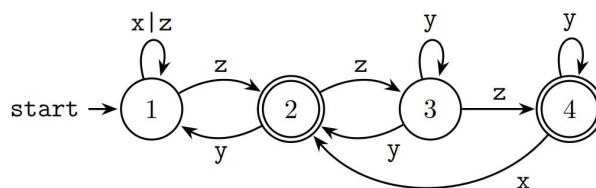
		0	1
A	$\rightarrow A$	$\{A, B\}^A$	A^A
	$\{A, B\}$	$\{A, B, C\}^B$	A^A
B	$*\{A, B, C\}$	$\{A, B, C\}^B$	$\{A, C\}^B$
	$*\{A, C\}$	$\{A, B\}^A$	$\{A, C\}^B$

		0	1
	$\rightarrow A$	B	A
	B	C	A
	$*C$	C	D
	$*D$	B	D



Não é possível minimizar o AFD porque após a 1ª Iteração não existem grupos com estados cujas transições sejam equivalentes.

b)



	x	y	z
$\rightarrow 1$	$\{1\}$	\emptyset	$\{1,2\}$
$*2$	\emptyset	$\{1\}$	$\{3\}$
3	\emptyset	$\{2,3\}$	$\{4\}$
$*4$	$\{2\}$	$\{4\}$	\emptyset

Erro! Não existe nenhum texto com o estilo especificado no documento.-1 - Tabela de transições do AFN

	x	y	z
\emptyset	\emptyset	\emptyset	\emptyset
$\rightarrow 1$	{1}	\emptyset	{1,2}
*2	\emptyset	{1}	{3}
3	\emptyset	{2,3}	{4}
*4	{2}	{4}	\emptyset
*{1,2}	{1}	{1}	{1,2,3}
{1,3}	{1}	{2,3}	{1,2,4}
*{1,4}	{1,2}	{4}	{1,2}
*{2,3}	\emptyset	{1,2,3}	{3,4}
*{2,4}	{2}	{1,4}	{3}
*{3,4}	{2}	{2,3,4}	{4}
*{1,2,3}	{1}	{1,2,3}	{1,2,3,4}
*{1,2,4}	{1,2}	{1,4}	{1,2,3}
*{1,3,4}	{1,2}	{2,3,4}	{1,2,4}
*{2,3,4}	{2}	{1,2,3,4}	{3,4}
*{1,2,3,4}	{1,2}	{1,2,3,4}	{1,2,3,4}

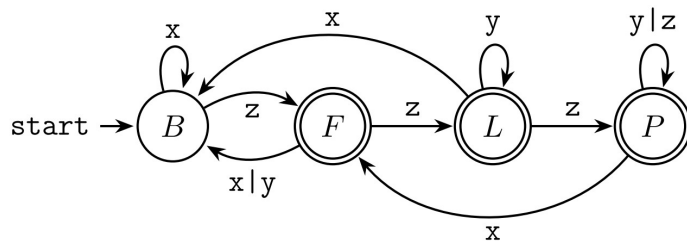
Erro! Não existe nenhum texto com o estilo especificado no documento.-2 - Tabela de transições do AFD

	x	y	z
A	A	A	A
$\rightarrow B$	B	A	F
*C	A	B	D
D	A	I	E
*E	C	E	A
*F	B	B	L
G	B	I	M
*H	F	E	F
*I	A	L	K
*J	C	H	D
*K	C	O	E
*L	B	L	P
*M	F	H	L
*N	F	O	M
*O	C	P	K
*P	F	P	P

Erro! Não existe nenhum texto com o estilo especificado no documento.-3 - Estados do AFD renomeados

	x	y	z
$\rightarrow B$	B	\emptyset	F
*F	B	B	L
*L	B	L	P
*P	F	P	P

Erro! Não existe nenhum texto com o estilo especificado no documento.-4 - Tabela simplificada do AFD



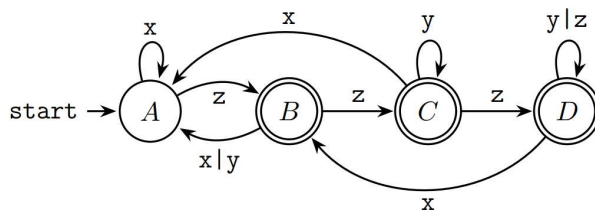
AFD

Minimização (1ª Iteração):

		x	y	z
S ₀	->B	B (S ₀)	∅ (S ₀)	F (S ₁)
	*F	B (S ₀)	B (S ₀)	L (S ₁)
S ₁	*L	B (S ₀)	L (S ₁)	P (S ₁)
	*P	F (S ₁)	P (S ₁)	P (S ₁)

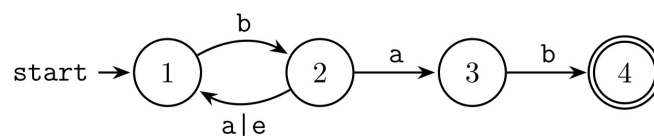
Erro! Não existe nenhum texto com o estilo especificado no documento.-5 - Tabela do AFD, dividida e classificada

Não é possível minimizar o AFD porque após a 1ª Iteração não existem grupos com estados cujas transições sejam equivalentes.



AFD

11. Indique uma expressão regular e um autómato finito determinístico que reconheça a mesma linguagem que o seguinte autómato finito não determinístico.



Expressão regular:

$b((a|e)b)^*ab$

Conversão AFN-AFD (versão completa):

	a	b	e
→1	∅	{2}	∅
2	{1,3}	∅	{1}
3	∅	{4}	∅
*4	∅	∅	∅

Tabela de transições do AFN

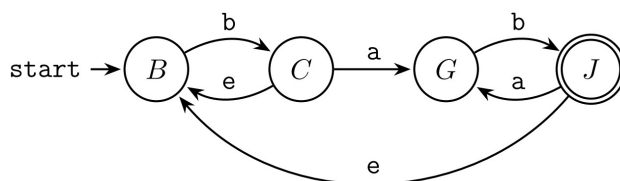
	a	b	e
\emptyset	\emptyset	\emptyset	\emptyset
$\rightarrow 1$	\emptyset	$\{2\}$	\emptyset
2	$\{1,3\}$	\emptyset	$\{1\}$
3	\emptyset	$\{4\}$	\emptyset
*4	\emptyset	\emptyset	\emptyset
$\{1,2\}$	$\{1,3\}$	$\{2\}$	$\{1\}$
$\{1,3\}$	\emptyset	$\{2,4\}$	\emptyset
* $\{1,4\}$	\emptyset	$\{2\}$	\emptyset
$\{2,3\}$	$\{1,3\}$	$\{4\}$	$\{1\}$
* $\{2,4\}$	$\{1,3\}$	\emptyset	$\{1\}$
* $\{3,4\}$	\emptyset	$\{4\}$	\emptyset
$\{1,2,3\}$	$\{1,3\}$	$\{2,4\}$	$\{1\}$
* $\{1,2,4\}$	$\{1,3\}$	$\{2\}$	$\{1\}$
* $\{1,3,4\}$	\emptyset	$\{2,4\}$	\emptyset
* $\{2,3,4\}$	$\{1,3\}$	$\{4\}$	$\{1\}$
* $\{1,2,3,4\}$	$\{1,3\}$	$\{2,4\}$	$\{1\}$

Tabela de transições do AFD

	a	b	e
A	A	A	A
$\rightarrow B$	A	C	A
C	G	A	B
D	A	E	A
*E	A	A	A
F	G	C	B
G	A	J	A
*H	A	C	A
I	G	E	B
*J	G	A	B
*K	A	E	A
L	G	J	B
*M	G	C	B
*N	A	J	A
*O	G	E	B
*P	G	J	B

Estados do AFD renomeados

	a	b	e
$\rightarrow B$	\emptyset	C	\emptyset
C	G	\emptyset	B
G	\emptyset	J	\emptyset
*J	G	\emptyset	B

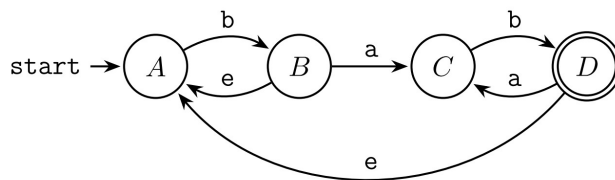


AFD

Conversão (versão simplificada):

	a	b	e
→1	∅	{2}	∅
2	{1,3}	∅	{1}
{1,3}	∅	{2,4}	∅
*{2,4}	{1,3}	∅	{1}

	a	b	e
→A	∅	B	∅
B	C	∅	A
C	∅	D	∅
*D	C	∅	A

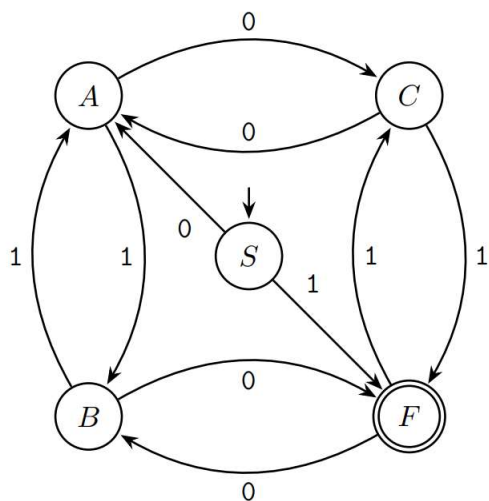


AFD

12. Considere a seguinte tabela de transições:

	0	1
→S	A	F
A	C	B
B	F	A
C	A	F
*F	B	C

- a) Represente graficamente o autômato finito correspondente. Classifique o autômato resultante.



AFD

Nota: nesta resolução considera-se que 0 ocorrências de 0 não é par.

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

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

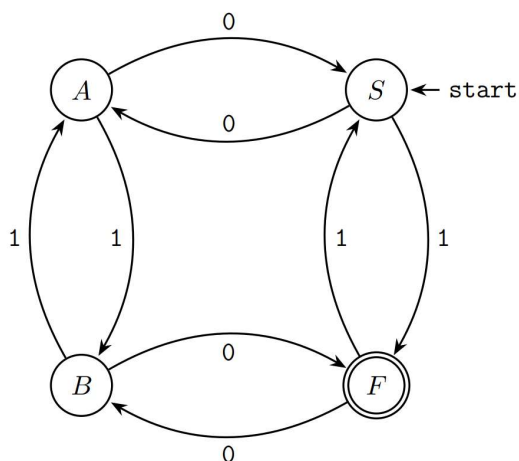
- i. V
- ii. F
- iii. V
- iv. F
- v. V
- vi. F

c) Minimize o autómato finito da alínea a).

		0	1
A	→S	A ^A	F ^B
	A	C ^A	B ^A
	B	F ^B	A ^A
	C	A ^A	F ^B
B	*F	B ^A	C ^A

		0	1
A	→S	A ^B	F ^C
	∈	A ^B	F ^C
B	A	C ^A	B ^B
	B	F ^C	A ^B
C	*F	B ^B	C ^A

	0	1
→S	A	F
A	S	B
B	F	A
*F	B	S



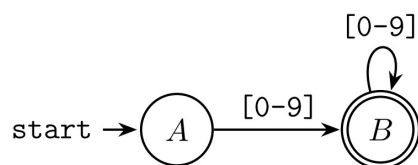
AFD

d) Descreva informalmente (por palavras) as linguagens representadas pelo autómato.

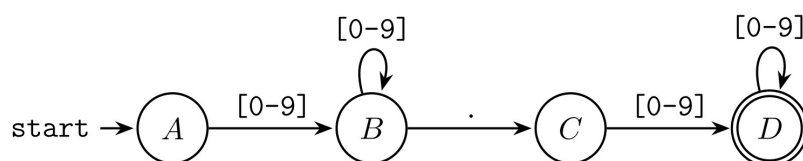
O autómato representa todos os números binários com número ímpar de 1 e, se existirem, par de 0.

13. Implemente um autómato finito determinístico para cada uma das expressões regulares:

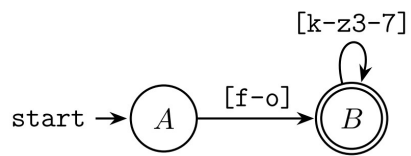
(a) $[0-9]^+$



(b) $[0-9]^+ \cdot "[0-9]^+$



(c) $[f-o][k-z3-7]^*$



(d) $(---[a-z]^*\backslash n)|(" "\backslash n|\backslash t)$

