

Linguagens Formais e Autómatos

Guião de exercícios teórico-práticos (1^a parte)

Miguel Oliveira e Silva

Departamento de Electrónica, Telecomunicações e Informática
Universidade de Aveiro

2017–2018, 2^o semestre

1 Expressões regulares

Exercício 1.1

Sobre o alfabeto $B = \{0, 1\}$, determine as seguintes expressões regulares.

- a) Uma ER em que dois dígitos **1** não podem aparecer juntos.
- b) Uma ER que represente o conjunto de números binários maiores ou iguais do que a quantidade decimal 7^1 .

Exercício 1.2

Sobre os alfabetos $A = \{x, y, z\}$ e $B = \{0, 1\}$, determine uma expressão regular para cada uma das seguintes linguagens:

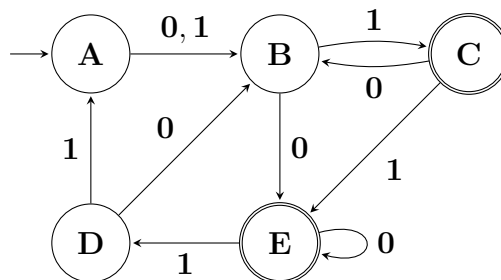
$$\begin{aligned}L_1 &= \{u \in B^* \mid \exists_i u_i = u_{i-1}\} \\L_2 &= \{u \in B^* \mid \#(01, u) > 1\} \\L_3 &= \{u \in B^* \mid \#(0, u) = 0 \vee \#(1, u) \bmod 2 = 0\} \\L_4 &= \{u \in B^* \mid \#(0, u) = \#(1, u) \wedge \#(1, u) \leq 2\} \\L_5 &= \{u \in B^* \mid \#(0, u) \geq 2 \vee \#(1, u) \geq 2\} \\L_6 &= \{u \in B^* \mid \#(10, u) < 3\} \\L_7 &= \{u \in A^* \mid \exists_i u_i = u_{i-1} = u_{i-2}\}\end{aligned}$$

¹O número 7 na base decimal corresponde ao número 111 na base 2.

2 Autômatos finitos

Exercício 2.1

Considere o seguinte diagrama de transição de um autômato finito determinista:



- Construa a tabela de transição.
- Se possível, reduza o autômato.

Exercício 2.2

Considere a seguinte tabela de transição:

STATE	a	b	c
$\rightarrow A$	C	B	B
B_f	C	D	B
C	A	B	B
D	A	B	C

- Construa o diagrama de transição.
- Se possível, reduza o autômato.

Exercício 2.3

Considere a seguinte tabela de transição:

STATE	0	1	ε
$\rightarrow A$	{A}	{A, B, D}	\emptyset
B	{E}	\emptyset	{A}
C	{A}	{D, A}	\emptyset
D_f	\emptyset	{B, C}	{E}
E_f	\emptyset	{A}	{B}

- Construa o diagrama de transição.
- Converta para um autômato finito determinista.
- Se possível, reduza o autômato.

3 Expressões regulares e autómatos finitos

Exercício 3.1

Sobre o alfabeto $A = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$, considere as seguintes expressões regulares:

$$\begin{aligned}e_1 &= (\mathbf{a|b|abc})^* \\e_2 &= (\mathbf{a|b})^* \mathbf{c?} (\mathbf{a|b})^+ \\e_3 &= (\mathbf{a|b|c(a|b)^*})^* \mathbf{a(b|c)}\end{aligned}$$

- a) Para cada uma destas expressões regulares, obtenha um autômato finito (não generalizado) que represente a mesma linguagem.
- b) Se necessário, converta para um autômato finito determinista.
- c) Se possível, reduza o autômato.

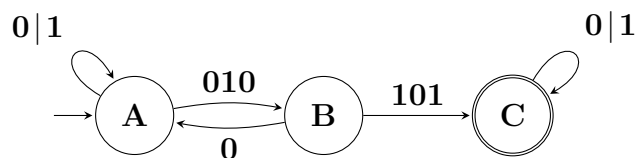
Exercício 3.2

Considerando as expressões regulares obtidas no exercício 1.2.

- a) Crie um autômato finito (não generalizado) que represente a mesma linguagem.
- b) Se necessário, converta para um autômato finito determinista.
- c) Se possível, reduza o autômato.

Exercício 3.3

Considere o seguinte diagrama de transição de um autômato finito generalizado:



- a) Converta num autômato finito generalizado reduzido.

4 ANTLR

Exercício 4.1

Considere uma linguagem definida pela seguinte gramática:

```
main: (header '\n' (content '\n')*)? EOF;
header: WORD+;
content: NUMBER+;
WORD: [a-z]+;
NUMBER: [0-1]+;
WS: [ \t\r]+ -> skip;
```

- Dê um exemplo de uma entrada que seja aceite por esta gramática e que utilize todas as regras expressas.
- Altere esta gramática por forma a, para além dos actuais, aceitar números reais representados com a notação de vírgula fixa.
- Altere esta gramática por forma a aceitar conteúdos (linhas que não a primeira) com *strings*. Uma *string* é definida como sendo uma sequência de letras, dígitos ou espaços, delimitada por aspas (ex: "123", "abc", "1 2 3", "Today is friday").

Exercício 4.2

Considere uma linguagem definida pela seguinte gramática:

```
main: i * EOF;
i: e;
e: e '*' e
  | e '+' e
  | v
  ;
v: N;
N: [0-1]+;
WS: [ \t\n\r]+ -> skip;
```

- Dê um exemplo de uma entrada que seja aceite por esta gramática e que utilize todas as regras expressas.
- Altere esta gramática por forma a aceitar também as operações de divisão (símbolo /) e de subtração. Faça com que essas operações tenham a mesma precedência que, respectivamente, a multiplicação e a soma.
- Altere esta gramática por forma a incluir a definição e utilização de *variáveis*. Para esse fim considere que uma variável é uma sequência não vazia de letras não acentuadas (minúsculas ou maiúsculas). Considere também que a instrução de atribuição de valor a uma variável tem a seguinte sintaxe: `variável = expressão`