Teoria da Computação
Linguagens e Expressões Regulares, Autómatos de Estados Finitos

Simão Melo de Sousa

24 de Março de 2015

Conteúdo

1 Linguagens e Expressões Regulares

Exercício 1 *Considere um alfabeto A e o monoíde livre A^* gerado por A.*

Sejam t, u, v e w palavras de A^*. Demonstre que se $tu = vw$ então existe uma única palavra z de A^* tal que

- *ou $t = v.z$ e $z.u = w$*
- *ou $v = t.z$ e $z.w = u$*

Este lema é conhecido por Lema de Levi.

Resposta □

Exercício 2 *Considere um alfabeto A e o monoíde livre A^* gerado por A.*

Recorde a definição da relação de ordem prefixo *sobre as palavras: para duas palavras a e b, $a \leq b$ se a é um prefixo de b (i.e. $\exists c \in A^*.b = a.c$).*

Sejam u_1, u_2, v três palavras de A^*. Demonstre que

$$(u_1 \leq v) \wedge (u_2 \leq v) \implies (u_1 \leq u_2) \vee (u_2 \leq u_1)$$

Resposta □

Exercício 3 *Considere um alfabeto A e o monoíde livre A^* gerado por A.*

Sejam a e b letras (elementos do alfabeto A) e u uma palavra de A^*. Demonstre que

$$u.a = b.u \implies a = b \wedge u \in a^*$$

Dica: Prossiga por indução sobre $|u|$.

□

Exercício 4 *Demonstre que para qualquer linguagem L, $L^* = (L^*)^*$*

Resposta □

Exercício 5 *Para cada uma das linguagens regulares seguintes, dar uma expressão regular que represente o complemento.*

- $(a + b)^*b$ Resposta

- $((a+b)(a+b))^*$

$\square$

Exercício 6 *Demonstre que cada uma das expressões regulares seguinte é equivalente a* $(a+b)^*$

- $(a^*b^*)^*$

Resposta

- $\emptyset + \epsilon + a(a+b)^* + b(a+b)^*$
- $a^* + (a^*b^*)^*b(a+b)^*$

$\square$

Exercício 7 *Dar uma expressão regular para a linguagem das palavras sobre o alfabeto* $\{a, b, c\}$ *tais que estas contém exactamente duas ocorrências da letra* a*, que cada ocorrência de* b *está imediatamente seguida de pelo menos duas ocorrências de* c *e que terminam por uma ocorrência de* a*.* Resposta $\square$

Exercício 8 *Considere o alfabeto* $A = \{a, b\}$ *e* A^**, o monoíde livremente gerado por* A*. Dê uma expressão regular para as linguagens seguintes:*

- *a linguagem das palavras onde cada ocorrência de a é imediatamente precedida de uma ocorrência de b;*
- *a linguagem das palavras que contém pelo menos uma ocorrência de abab;*
- *a linguagem das palavras que não contêm ocorrências de aa nem de bb;*
- *a linguagem das palavras que contêm um número ímpar de ocorrências de a e um número par de ocorrências de b;*
- *a linguagem das palavras que contém pelo menos uma ocorrência de ab e de ba.*

$\square$

Exercício 9 *A seguinte tabela lista algumas equivalências entre expressões regulares.*

1	$\alpha\epsilon = \epsilon\alpha = \alpha$	*elemento neutro da concatenação*
2	$\alpha + \emptyset = \emptyset + \alpha = \alpha$	*elemento neutro da união*
3	$(\alpha + \beta) + \gamma = \alpha + (\beta + \gamma)$	*associatividade da união*
4	$(\alpha\beta)\gamma = \alpha(\beta\gamma)$	*associatividade da concatenação*
5	$\alpha + \beta = \beta + \alpha$	*comutatividade da união*
6	$\alpha + \alpha = \alpha$	*idempotência*
7	$\alpha(\beta + \gamma) = \alpha\beta + \alpha\gamma$	*distributividade da concatenação a direita*
8	$(\beta + \gamma)\alpha = \beta\alpha + \gamma\alpha$	*distributividade da concatenação a esquerda*
9	$\alpha^+ = \alpha\alpha^* = \alpha^*\alpha$	*relação entre fechos*
10	$\alpha^* = \epsilon + \alpha^+$	*relação entre fechos e união*
11	$(\alpha + \epsilon)^+ = (\epsilon + \alpha)^+ = \alpha^*$	*relação entre fechos e união*

Utilize esta tabela para a resolução das alíneas seguintes.

1. *Mostre a equivalência de* $a^*b + a^*bb^+$ *com* a^*b^+

2. *Reduza a expressão regular* $ab^* + b(b^* + a^+)aa$

□

Exercício 10 *O seguinte algoritmo calcula uma expressão regular e a partir duma gramática regular* G *tal que* $\mathcal{L}(G) = \mathcal{L}(e)$

Algoritmo: *Conversão de uma gramática regular numa expressão regular*

Entrada: *Uma gramática regular*

Resultado: *Um sistema de equações de expressões regulares*

1. *Para cada símbolo não terminal* A*, que apareça no lado esquerdo de uma produção* $A \rightarrow \alpha_1\alpha_2\ldots\alpha_n$*, gerar a seguinte expressão:* $A = \phi_1\phi_2\ldots\phi_n$ *onde* ϕ_i *é a expressão regular correspondente a* α_i*.*

2. *Para cada símbolo não-terminal* A*, que apareça no lado esquerdo de uma produção* $A \rightarrow \alpha_1|\alpha_2|\ldots|\alpha_n$*, gerar a seguinte expressão regular* $A = \phi_1 + \phi_2 + \ldots + \phi_n$ *onde* ϕ_i *é a expressão regular correspondente a* α_i*.*

3. *Simplificar as expressões regulares usando as regras da tabela anterior ou as regras seguintes:*

   (a) $A = \beta + \alpha A$ *é equivalente a* $A = \alpha^*\beta$

   (b) $A = \beta + A\alpha$ *é equivalente a* $A = \beta\alpha^*$

4. *Resolver o sistema de equação em ordem ao símbolo inicial.*

Descreva por uma expressão regular a linguagem gerada pelas gramáticas descritas pelas produções seguintes:

Serão valorizadas as resoluções utilizando o algoritmo apresentado, embora outros tipos de resoluções (uma resolução ad-hoc *por exemplo) sejam aceites.*

gramáticas seguintes:

1. $$\begin{array}{lcl} S & \rightarrow & cS \mid B \mid A \\ A & \rightarrow & a \mid aA \\ B & \rightarrow & b \mid bB \end{array}$$

2. $$\begin{array}{lcl} S & \rightarrow & Sa|B \\ B & \rightarrow & b|Bb \end{array}$$

3. $$\begin{array}{lcl} S & \rightarrow & SA \\ SA & \rightarrow & BS \\ BS & \rightarrow & ab \\ B & \rightarrow & a \\ A & \rightarrow & c \end{array}$$

□

Exercício 11 (Reconhecimento duma string por uma expressão regular) *Neste exercício, propomo-nos de escrever uma função que permita determinar se uma palavra pertence a uma linguagem $\mathcal{L}(r)$ gerada por uma expressão regular r. As expressões regulares são representadas elementos do tipo* OCaml *seguinte:*

```
type  expreg =
      | Vazio                                (* Linguagem vazia ∅ *)
      | Epsilon                              (* Palavra vazia ϵ *)
      | Caractere  of char                   (* Caracter c *)
      | Uniao      of expreg * expreg        (* r1|r2 *)
      | Produto    of expreg * expreg        (* r1r2 *)
      | Estrela    of expreg                 (* r* *)
```

1. *Defina uma função* `contem_epsilon` : `expreg` $\rightarrow$ `bool` *que determina se a palavra vazia* ϵ *pertence à linguagem gerada pela expressão regular dada em parâmetro.*

   Resposta

2. *Define-se por* resíduo *duma expressão regular* r *relativamente a um caracter* c *da forma seguinte :*

   $$Res(r, c) = \{\ w \mid cw \in L(r)\ \}$$

   (a) *Calcula* $Res(r, c)$ *no caso de* $r = (a|b)^*a$ *e* $c \in \{a, b\}$.

   (b) *A linguagem* $Res(r, c)$ *pode ser ela própria descrita por uma expressão regular. Escreva uma função* `residuo` : `expreg` $\rightarrow$ `char` $\rightarrow$ `expreg` *que calcula, a partir de* r *e de* c *uma expressão regular* r' *tal que* $L(r') = Res(r, c)$.

   (c) *Calcular* `residuo` r c *no caso onde* $r = (a|b)^*a$ *e* $c \in \{a, b\}$.

   Resposta

3. *Deduzir uma função* `reconheçe` : `expreg` $\rightarrow$ `char list` $\rightarrow$ `bool` *que determina se uma lista de caracteres pertence à linguagem definida pela expressão regular dada em parâmetro.* Resposta

4. *Aplicar esta função ao reconhecimento da palavra aba pela expressão regular* $r = (a|b) * a$. Resposta

□

2 Autómatos de Estados Finitos

2.1 Autómatos e Linguagens

Exercício 12 *Apresenta um autómato que reconhece a linguagem gerada pela expressão regular* $(aa)^*(c + b^+)^*$. □

Exercício 13 (construção de autómatos a partir de linguagens regulares) *Seja* $\Sigma = \{a, b\}$ *um alfabeto.*

- *Defina um autómato que reconheça palavras de comprimento ímpar*

□

Exercício 14 (Construção de autómatos a partir de expressões regulares) *Seja* $\Sigma = \{a, b, c, d\}$ *um alfabeto. Dê o autómato determinista minimal que reconhece as expressões regulares seguintes:*

- $((a + c)^3)^*$
- $ab(a + b)^*(bc + cd)$
- $(c + d)a(aa + bbb)^*$

□

2.1.1 Autómatos Determinísticos

Exercício 15 *Considere o autómato determinísticos* $(Q = \{q_0, q_1, q_2, q_3\}, \Sigma = \{a, b\}, \delta, q_0, F = \{q_2\})$ *cujas transições são:*

$\delta =$

	a	b
q_0	q_1	q_3
q_1	q_3	q_2
q_2	q_2	q_2
q_3	q_3	q_3

1. *desenhe o autómato considerado;*
2. *Qual é a linguagem aceite por este autómato?*

□

Exercício 16 (Retirado do livro de Papadimitriou) *Considere os autómatos das figuras 1 e 2.*

- *Apresenta, para cada um dos autómatos, o caminho seguido pelo reconhecimento da palavra* abaaabaab *(quer este seja bem sucedido ou não).*

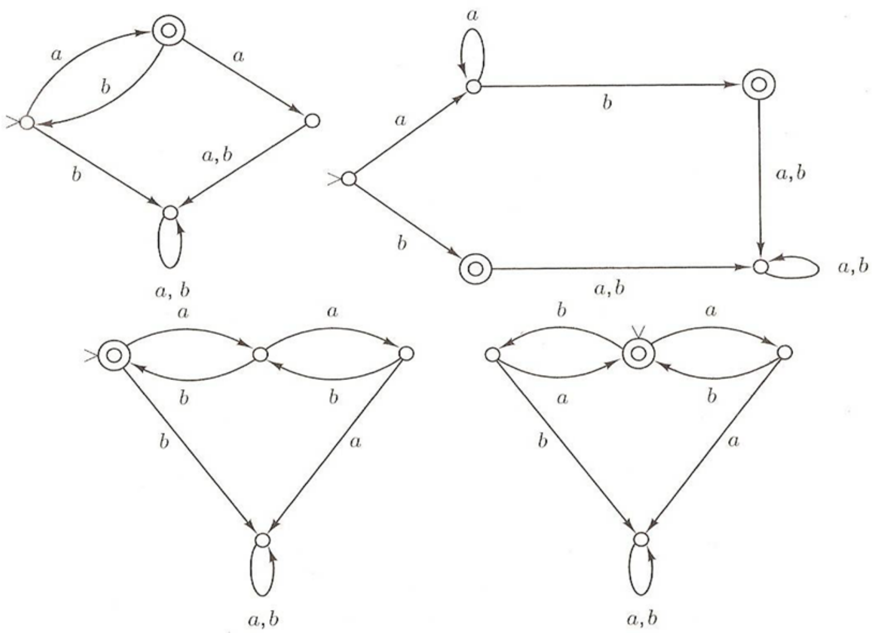


Figura 1: Autómatos determinísticos

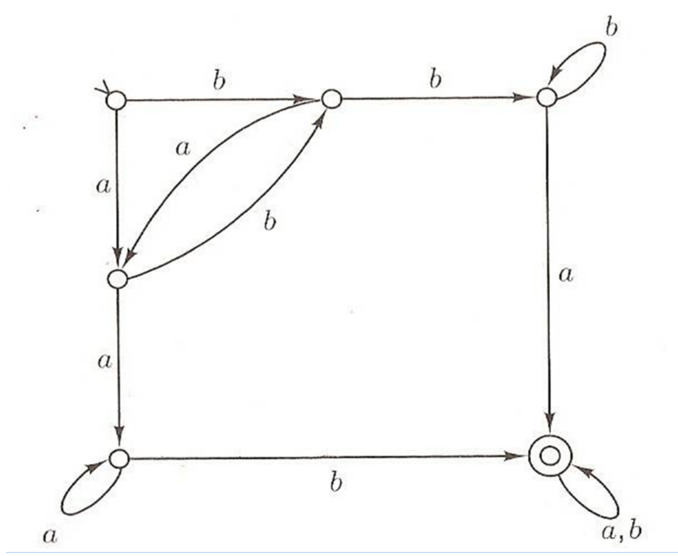


Figura 2: Autómatos determinísticos - bis

- *Descreva informalmente as linguagens que estes reconhecem.*

□

Exercício 17 *Relembra-se que as linguagens aceitas por autómatos finitos são regulares.*

Demonstre que as linguagens seguintes são regulares:

- $\{awa \mid w \in \{a,b\}^*\}$ Resposta

- $\{a^n \mid n \in \mathbb{N} \wedge n \geq 3\}$

- $\{a^n \mid n \in \mathbb{N} \wedge n \neq 3\}$

- $\{vwv \mid w \in \{a,b\}^* \wedge v \in \{a,b\}^* \wedge |v| = 2\}$

- *a linguagem dos números flutuantes da norma $IEE354$ (o conjunto dos `float` da linguagem C ou OCaml)*

□

Exercício 18 *Considere o alfabeto $A = \{a,b\}$ e A^*, o monoíde livremente gerado por A. Dê os autómatos finitos que reconheçam cada uma das linguagens seguintes:*

- *a linguagem das palavras onde cada ocorrência de a é imediatamente precedida de uma ocorrência de b;*

- *a linguagem das palavras que contém pelo menos uma ocorrência de abab;*

- *a linguagem das palavras que não contêm ocorrências de aa nem de bb;*

- *a linguagem das palavras que contêm um número ímpar de ocorrências de a e um número par de ocorrências de b;*

- *a linguagem das palavras que contém pelo menos uma ocorrência de ab e de ba.*

□

Exercício 19 *Dê um autómato determinista para as seguintes linguagens:*

- $\{ab^nb^m \mid n \geq 2, m \geq 3\}$

	0
q_0 (*inicial*)	q_1, q_4
q_1	q_2
q_2	q_3
q_3 (*final*)	
q_4	q_5
q_5 (*final*)	q_4

	0	1	ϵ
q_0 (*inicial e final*)		q_1	q_2
q_1	q_0, q_2		
q_2			

- $\{(ab)^n \mid n \geq 1\}$
- $\{ab^5wb^2 \mid w \in \{a,b\}^*\}$

□

Exercício 20 *Considere o alfabeto $\Sigma = \{a, b\}$, dê um autómato finito **determinista** que reconheça a linguagem $\{w_1abaw_2 \mid w_1 \in \Sigma^*, w_2 \in \Sigma^*, |w_1| \geq 3, |w_2| \leq 5\}$* □

Exercício 21 *Sejam $M = \{Q, \Sigma, \delta, s_0, F\}$ e $M' = \{Q, \Sigma, \delta, s_0, F'\}$ dois autómatos deterministas tais que F' é o conjunto dos estados $q' \in Q$ a partir dos quais um estado $q \in F$ é acessível em M (ou seja para os qais existe um palavra w tal que $(q', w) \vdash^* (q, \epsilon)$).*

Qual é a relação entre as linguagens aceites por M e por M'?

Resposta □

2.1.2 Autómatos Não-Determinísticos

Exercício 22 *Considere os autómatos não determinísticos cujas tabelas de transição são:*

1. *Desenhe os dois autómatos.*
2. *Define formalmente os dois autómatos.*
3. *Quais destas palavras são aceitas pelo segundo autómato?* 0101, 01, 0011, 110, 1010, 010101, 10100
4. *Construa um autómato determinista que aceita a linguagem reconhecida pelo primeiro autómato.*
5. *Defina também o autómato determinista para a linguagem complemento.*

□

Exercício 23 *Construa um autómato determinista e não determinista que reconheça cada uma das linguagens listadas a seguir:*

- *As representações binárias dos inteiros pares.*
- *As representações decimais dos inteiros divisíveis por* 3.
- *A linguagem sobre o alfabeto* $\{a, b\}$ *das palavras contendo uma ocorrência de aab ou uma ocorrência de aaab.*
- *Os números binários tais que o antepenúltimo digito é* 1.

Resposta

□

Exercício 24 *Defina um autómato não determinista com apenas três estados que aceite a linguagem* $\{ab, abc\}^*$. □

Exercício 25 *Defina um autómato não determinista que reconheça sequências de* 0 *cujo comprimento é múltiplo de* 2 *ou de* 3. Resposta

□

Exercício 26 *Defina um autómato não determinista com apenas cinco estados que aceite a linguagem* $\{abab^n \mid n \in \mathbb{N}\} \cup \{aba^n \mid n \in \mathbb{N}\}$. □

Exercício 27 *Considere as palavras* 00, 01001, 10010, 000, 0000 *e o autómato da figura 3. Diga que palavras são aceites pelo autómato.* □

2.2 Algoritmos e Transformações de Autómatos

2.2.1 Remoção de transições epsilon

Exercício 28 *Remova as transições* ϵ *do autómato da figura 4.* □

Exercício 29 *Remova as transições* ϵ *do autómato da figura 5.* □

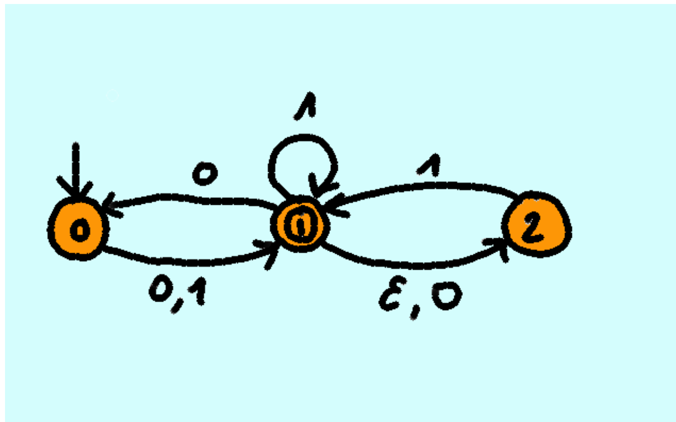


Figura 3:

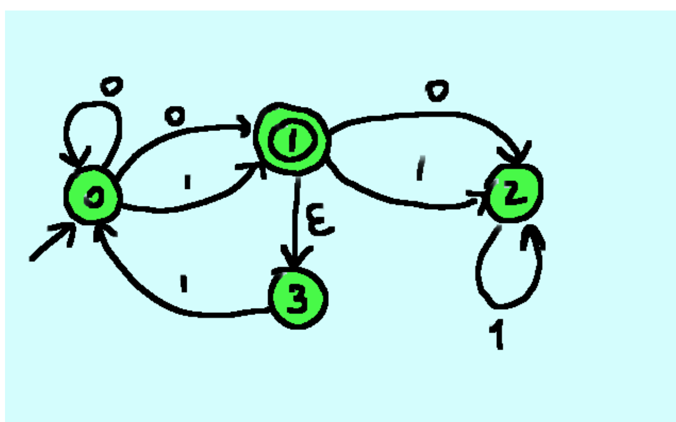


Figura 4:

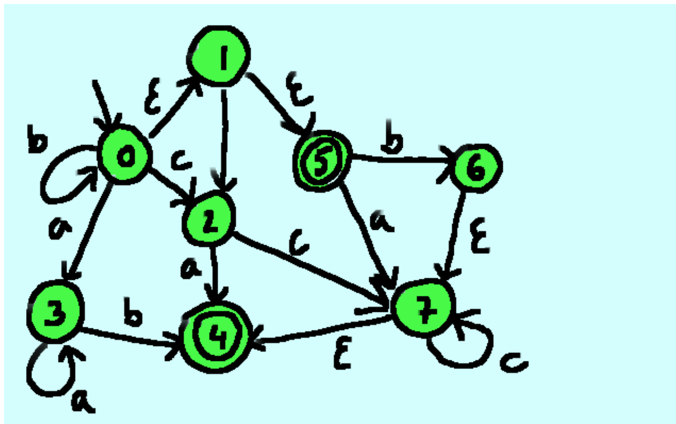


Figura 5:

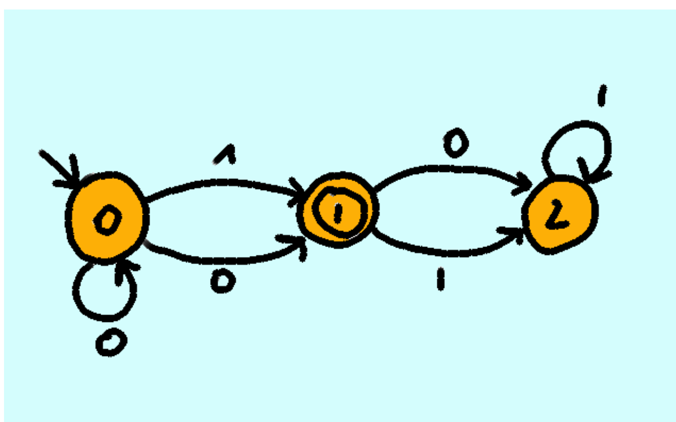


Figura 6:

2.2.2 Determinização

Exercício 30 *Determinise o autómato da figura 6.* □

Exercício 31 *Determinise o autómato da figura 3.* □

2.2.3 Minimização

Exercício 32 *Minimize o autómato da figura 7.* □

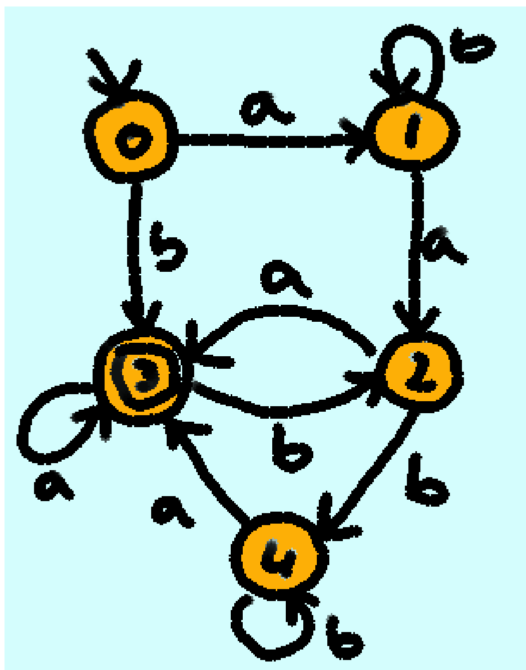


Figura 7:

Exercício 33 *Minimize o autómato da figura 8.* □

Exercício 34 *Minimize o autómato da figura 9.* □

Exercício 35 *Minimize o autómato da figura 10.* □

Exercício 36 *Minimize o autómato da figura 11.* □

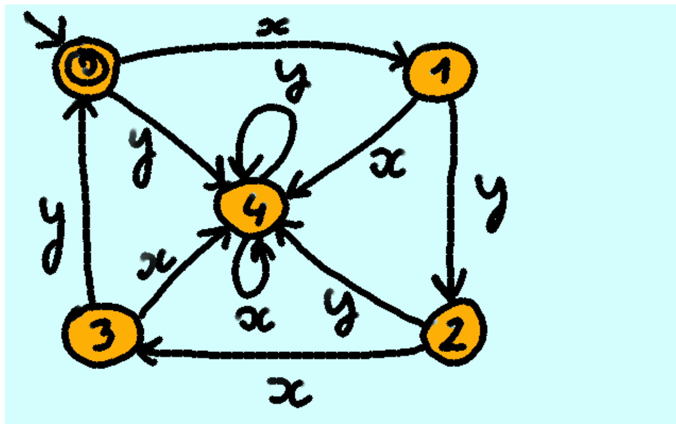


Figura 8:

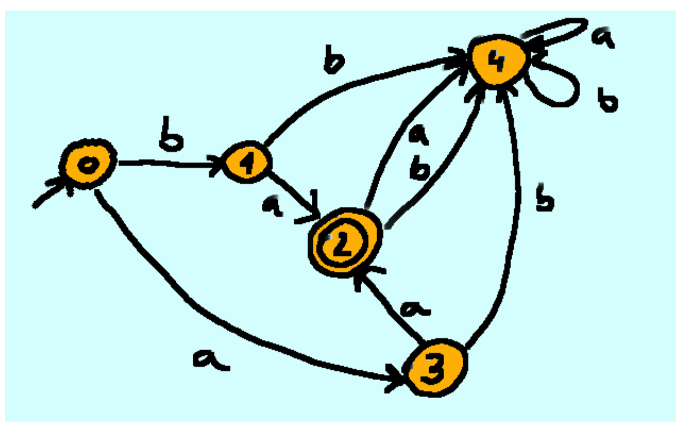


Figura 9:

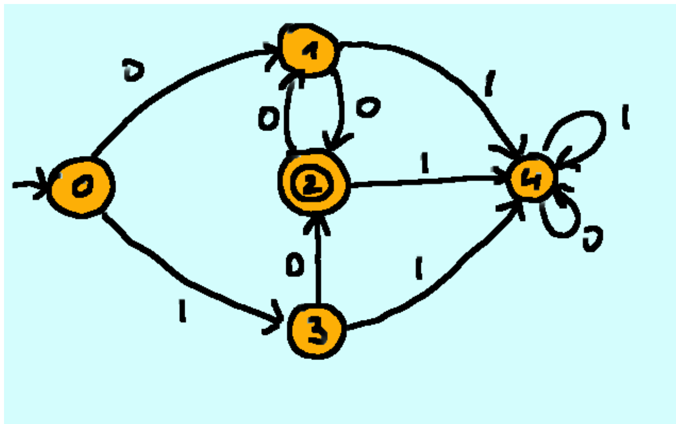


Figura 10:

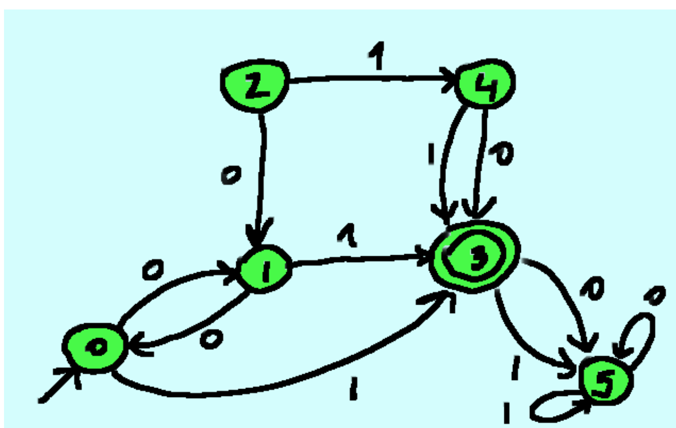


Figura 11:

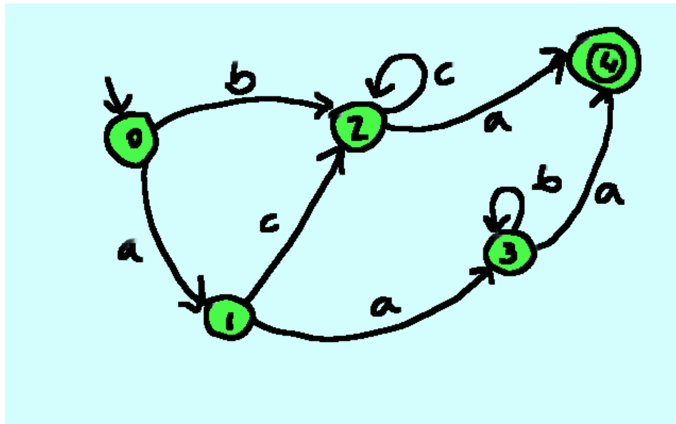


Figura 12:

2.2.4 Completação e estado poço

Exercício 37 *Complete o autómato da figura 12.* □

2.2.5 Autómatos canónicos

Exercício 38 *Utilizando os algoritmos apresentados, remova os estados inacessíveis e não produtivos do autómato da figura 13.* □

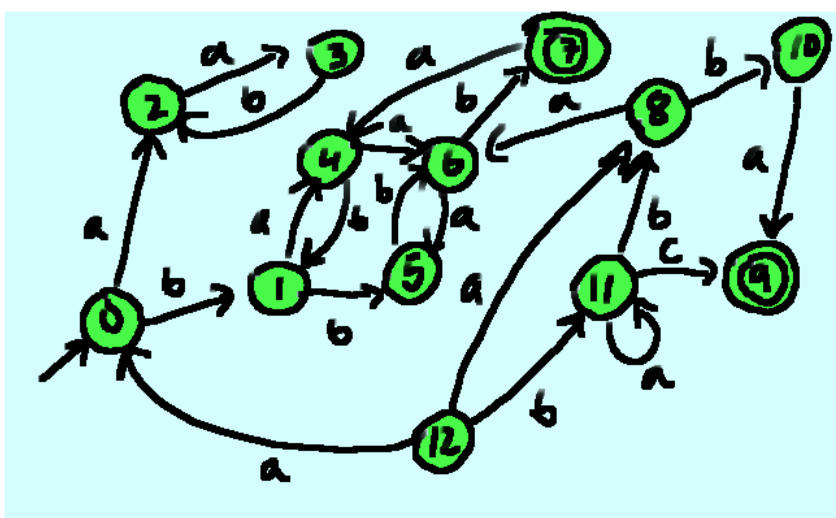


Figura 13:

Exercício 39 *Utilizando os algoritmos apresentados, remova os estados inacessíveis e não produtivos do autómato da figura 14.* □

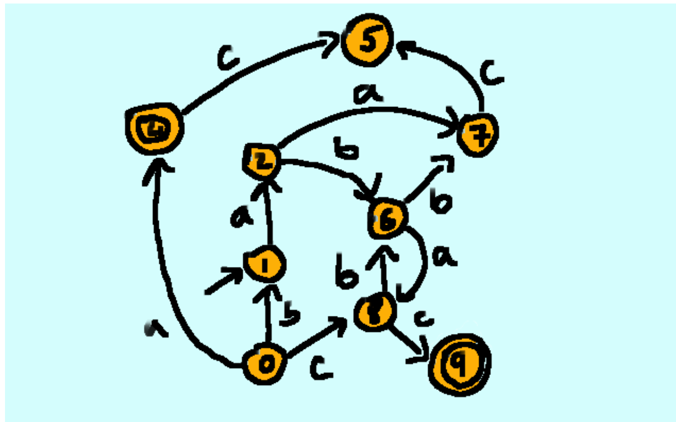


Figura 14:

2.2.6 Algoritmos Complementares e Exercícios Completos

Exercício 40 *Dê um autómato determinista minimal equivalente ao autómato da figura 15.* □

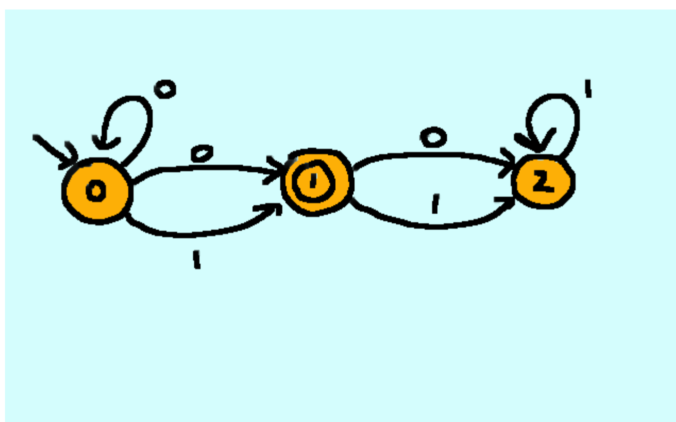


Figura 15:

Exercício 41 *Dê um autómato determinista minimal equivalente ao autómato da figura 4.* □

Exercício 42 *Sejam $\mathcal{A} \triangleq \{a, b, c\}$, um alfabeto e $r \triangleq (a|bb)^*cc(a)^+b$, uma expressão regular sobre o alfabeto $\mathcal{A}$.*

Dê o autómato finito determinista minimal que reconhece a linguagem $\mathcal{L}(r)$.

Sugerimos que comece por determinar um autómato não determinista com transições ϵ, que remova as transições ϵ, que torne determinista o autómato resultante e que, finalmente, aplique o algoritmo de minimização. □

Exercício 43 *Considere o alfabeto $\mathcal{A} = \{a, b\}$ e o autómato A_1 com transições ϵ representado na figura 16. Nesta figura as transições ϵ são representadas por $\xrightarrow{@}$*

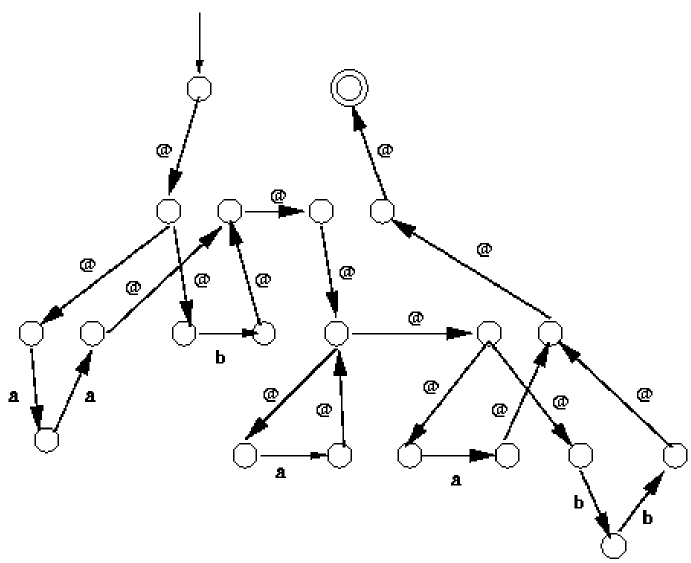


Figura 16: Autómato A_1 (com transições ϵ)

- *Que linguagem reconhece este autómato?*
- *Dê um autómato A_2 sem transições ϵ equivalente a A_1.*
- *Determinise, caso necessário, o autómato A_2. O autómato resultante será designado por A_3.*
- *Minimise A_3.*

□

Exercício 44 *Considere o autómato da figura 17, definido sobre o alfabeto $\mathcal{A} = \{a, b, c\}$:*

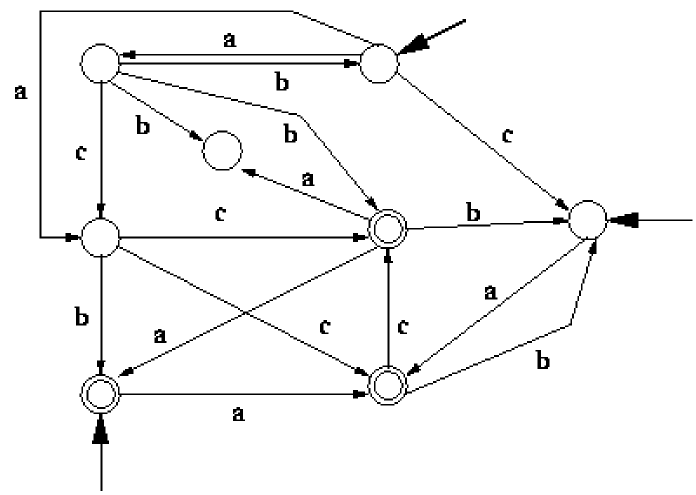


Figura 17: O Autómato A_1

- *Determinise o autómato A_1. O autómato resultante será designado por A_2.*
- *Minimise A_2. O autómato resultante será designado por A_3.*
- *Que linguagem reconhecem os autómatos A_1, A_2 e A_3?*

□

Exercício 45 *Utilizando os algoritmos apresentados na disciplina,*

- *apresenta um autómato que reconhece a linguagem gerada pela expressão regular $a^* + (b\, b^+)$;*
- *remova as transições ϵ do automato resultante;*
- *transforme o autómato A_1 da figura 18 num autómato determinístico;*
- *minimise o autómato resultante.*

□

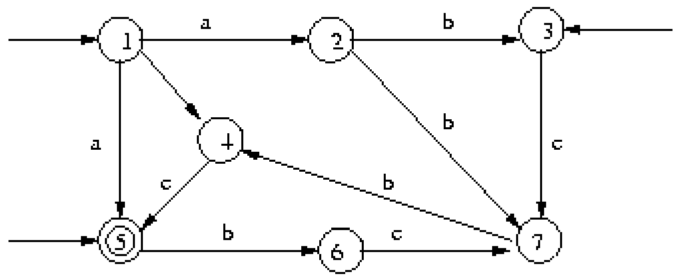


Figura 18: Autómato A_1

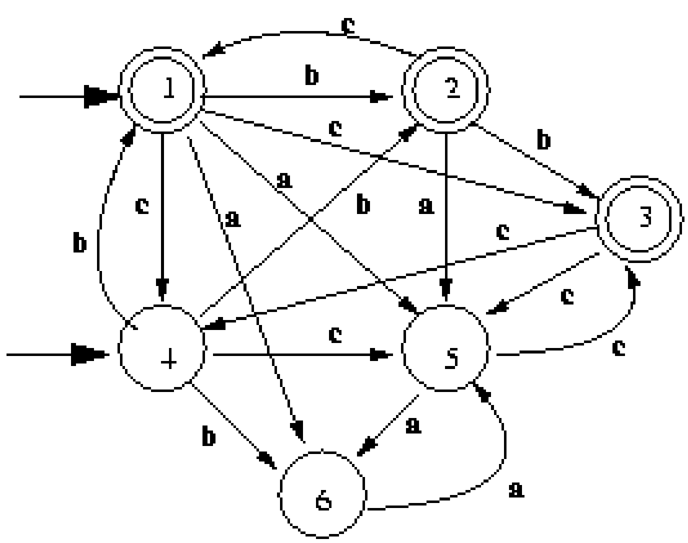


Figura 19: Autómato A_1

Exercício 46

- *Transforme o autómato A_1 da figura 19 num autómato determinístico;*
- *minimise o autómato resultante.*

□

Exercício 47 *Considere um autómato $M = \{Q, \Sigma, I, F, R_\delta\}$ não determinista com transições ϵ com $|I| > 1$. É sempre possível transformar um autómato como M num autómato não determinístico com transições ϵ equivalente M' possuindo um só estado inicial.*

1. *Proponha um algoritmo que realize tal transformação.*
2. *Demonstre (ou pelo menos dê um esboço de demonstração) que o autómato resultante M' é equivalente ao autómato M (ou seja que $L(M) = L(M')$). Esta propriedade é a propriedade de correcção do algoritmo proposto.*

□

2.3 Teorema de Kleene: Autómatos e Expressões Regulares

2.3.1 Das Expressões regulares para os autómatos

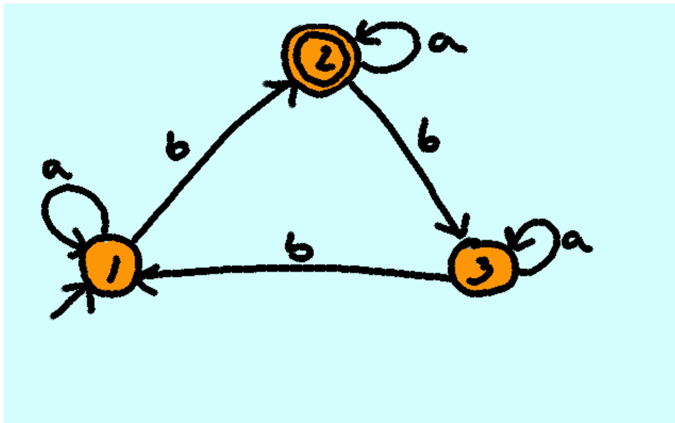
Exercício 48

1. *Construa um autómato determinista que aceite as palavras sobre o alfabeto $\Sigma = \{a, b\}$ que tem um numero par de ocorrências de a e um número ímpar de ocorrências de b.*
2. *Usando os algoritmos apresentados nas aulas, determine que expressão regular que este autómato reconhece.*

□

Exercício 49 *Com base numa construção directa de autómatos adequada, mostre que a classe das lingagens aceites por autómatos finitos é fechada pelas operações:*

- *União*



- *Concatenação*
- *Fecho de Kleene*
- *Complemento*
- *Intersecção*

□

Exercício 50

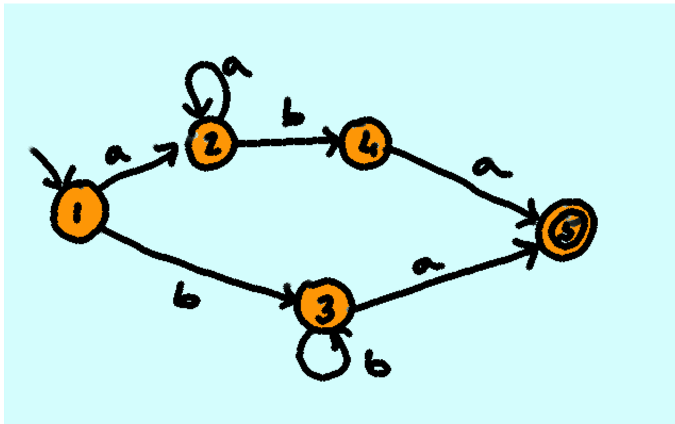
- *Usando o algoritmo apresentado nas aulas, dê o NDFA associado a expressão regular* $(ab + ba + c)(ab)^*$.
- *Usando as propriedades de fecho das linguagens regulares (ver exercício 49), dê o NDFA associado a mesma expressão regular.*
- *Determinise e minimize o autómato resultante.*
- *verifique estes resultados utilizando o JFLAP.*

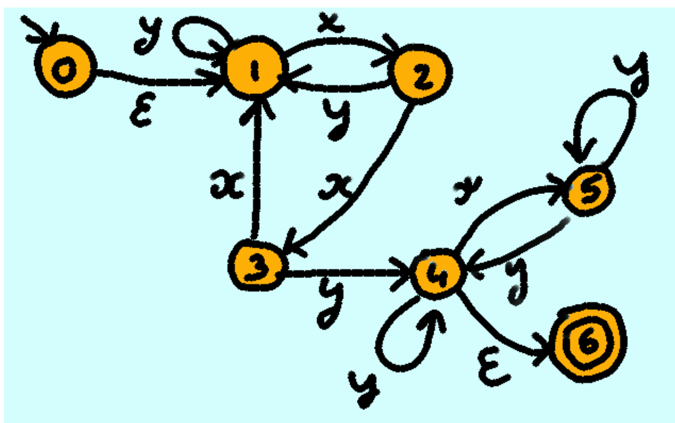
□

2.3.2 Dos Autómatos às Expressões Regulares

Exercício 51 *Aplicar o algoritmo de Mac-Naughton-Yamada sobre o autómato da figura 51 para obter a expressão regular associada. Ou seja, calcule* $R(1, 2, 4)$

□





Exercício 52 *Aplique o algoritmo de eliminação de estado para calcular a expressão regular associado ao autómato da figura 52*

□

Exercício 53 *Aplique o algoritmo de eliminação de estado para calcular a expressão regular associado ao autómato da figura 53*

□

2.4 Exercícios completos

Exercício 54 *1. Considere o alfabeto $\Sigma = \{a, b\}$, defina um autómato* determinista *que reconheça a linguagem $\{w_1abw_2 \mid w_1, w_2 \in \{a, b\}^*\}$.*

Resposta

2. *Considere o alfabeto* $\Sigma = \{a, b\}$ *e o autómato* A_1 *da figura 20:*

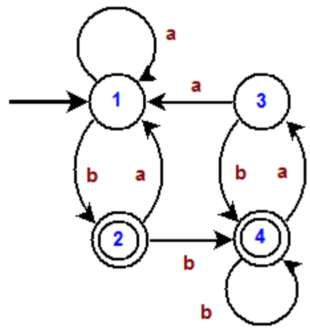


Figura 20: Autómato A_1

   (a) *Minimize o autómato* A_1.

   Resposta

   (b) *Olhando para o autómato minimal resultante, descreva de forma sucinta e informal (i.e. em português) a linguagem aceite por* A_1.

   Resposta

3. *Considere o autómato* A_2 *da figura 21.*

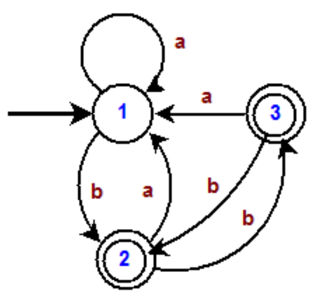


Figura 21: Autómato A_2

Utilize o algoritmo de Mac Naughton-Yamada *para inferir que expressão/linguagem regular aceita este autómato. Pretende-se aqui que apre-*

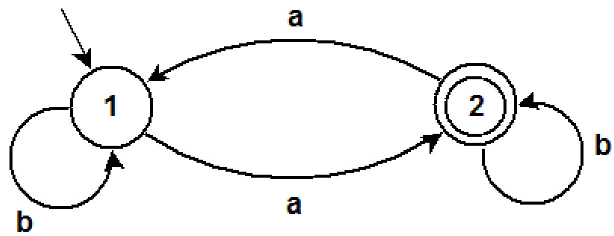


Figura 22: Autómato A_1

sente somente *o resultado final (a expressão regular calculada) e o valor de* $R(1,3,2)$.

Resposta

□

Exercício 55 *Considere o alfabeto* $\Sigma = \{a, b\}$ *e o autómato* A_1 *da figura 22.*

1. *Considere as* derivações completas *(derivações que começam no estado inicial e que terminam no estado final, também designados por* caminhos bem sucedido*s).*

   (a) *Define por indução o conjunto das derivações completas do autómato* A_1*.*
   Dica: Antes de mais, todas as derivações completas devem ser consideradas pelo conjunto.

      i. *Determine qual é a menor derivação completa possível em* A_1*? Está será o caso de base.*

      ii. *Como se constrói uma derivação completa a partir de outra? Deduza assim o caso indutivo.*

   Repare que o conjunto das etiquetas das derivações completas formam o conjunto das palavras aceites pelo autómato A_1*.*

   (b) *Qual é o princípio de indução associado à definição indutiva anterior?*

   (c) *Para uma palavra* p, $|p|_a$ *representa o número de a's em* p*. Demonstre por indução sobre as derivações completas que* $\forall p \in L(A_1), |p|_a$ *é impar.*

2. *Utilize o algoritmo de MacNaughton e Yamada para calcular a expressão regular correspondente ao autómato A_1. Para este efeito deverá preencher a tabela seguinte*

	$K = 1$	$K = 2$
$R(1, 1, K)$		
$R(1, 2, K)$		
$R(2, 1, K)$		
$R(2, 2, K)$		

□

3 Limites das Linguagens Regulares

Exercício 56 *Demonstre que as seguintes linguagens não são regulares:*

- $\{a^n \mid n \in \mathbb{N} \land \sqrt{n} \in \mathbb{N}\}$.
- $\{a^{n!} \mid n \in \mathbb{N}\}$.
- $\{a^{n^2} \mid n \in \mathbb{N}\}$.
- $\{a^n b^n \mid n \in \mathbb{N}\}$.
- $\{a^n \mid n \in \mathbb{N} \land n\ primo\}$
- $\{w \in \{a, b\}^* \mid |w|_a = |w|_b\}$ *(onde* $|w|_a$ *= número das ocorrências de* a *em* w*)*
- $\{ww \mid w \in A^*\}$ *para um alfabeto* A.
- $\{a^{3^n} \mid n \in \mathbb{N}\}$. Resposta

□

Exercício 57 *Determine se as asserções seguinte são verdadeiras ou falsas.*

1. *Qualquer linguagem regular tem um subconjunto próprio regular.*
2. *Se* L *é regular então* $\{w.w^r | w \in L\}$ *é regular. (se* $w = a_1 a'_{,,} a_3 \ldots a_n$ *então* $w^r = a_n \ldots a_3 a_2 a_1$*)*

3. *Se L_1 e L_2 são linguagens regulares, então $\{x \mid x \in L_1 \land x \notin L_2\}$ é regular*

4. *Qualquer união finita de linguagens regulares é regular.*

5. *Qualquer união infinita de linguagens regulares é regular.*

$\square$