Equivalência e minimização de autómatos[1]

No que se segue são descritos algoritmos que, dados dois autómatos finitos deterministas D e D', permitem determinar

- se as linguagens reconhecidas por D e D' são iguais
- se existe ou não um autómato cuja linguagem é a linguagem reconhecida por D mas tem menos estados que D
- um autómato mínimo (relativamente ao número de estados) que reconhece exactamente a mesma linguagem que D.

Estes algoritmos têm por base as noções de par de estados equivalentes e de par de estados distinguíveis e o algoritmo de procura exaustiva de pares distinguíveis. As noções de demonstração de uma sequência num autómato, de estado relevante e de estado produtivo são também utilizadas na sequência.

Definição: DEMONSTRAÇÃO DE SEQUÊNCIA
Seja $D = (Q, I, \delta, q_0, F)$ um autómato finito determinista. A demonstração de uma sequência $x = a_1 \ldots a_n \in I^*$, $n \geq 0$, em D é uma sequência

$$p_0 p_1 \cdots p_n$$

de estados de D tal que (i) $p_0 = q_0$, (ii) $p_n \in F$ e (iii) para cada $0 \leq j < n$, $p_{j+1} = \delta(p_j, a_{j+1})$.

Definição: ESTADO RELEVANTE, PRODUTIVO E INÚTIL
Seja $D = (Q, I, \delta, q_0, F)$ um autómato finito determinista e seja $q \in Q$

- q é um estado relevante de D sse existe $x \in I^*$ tal que $\delta^*(q_0, x) = q$;
- q é um estado produtivo de D sse existe $x \in I^*$ tal que $\delta^*(q, x) \in F$;
- q é um estado inútil de D sse não é um estado relevante de D ou não é um estado produtivo de D.

Exemplo Considere-se o autómato finito determinista $D = (Q, I, \delta, q_0, F)$ onde

- $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- $I = \{0, 1\}$
- $\delta : Q \times I \to Q$ tal que

δ	0	1
q_0	q_1	q_4
q_1	q_1	q_2
q_2	q_0	q_2
q_3	q_2	q_3
q_4	q_4	q_4

[1] Elaborado em 2001/2002

- $F = \{q_0, q_1\}$.

Tem-se que

– A demonstração de 000 em D é a sequência de estados $q_0q_1q_1q_1$ e a demonstração de 01100 é a sequência de estados $q_0q_1q_2q_2q_0q_1$.

– Estados relevantes de D: q_0, q_1, q_2 e q_4.

– Estados produtivos de D: q_0, q_1, q_2 e q_3.

– Estados inúteis de D: q_3 e q_4.

Estados inúteis são estados cuja presença não tem, em geral[2], qualquer importância do ponto de vista da linguagem que é reconhecida por um autómato. Com efeito, se um estado não é relevante, então nenhuma demonstração de uma sequência reconhecida pelo autómato inclui este estado. Do mesmo modo, se um estado não é produtivo, então nenhuma demonstração de uma sequência reconhecida pelo autómato inclui este estado.

Observação: Note-se que um estado inicial é sempre relevante e, portanto, se um estado inicial é inútil é porque não é produtivo. Por outro lado, um estado final é sempre produtivo e, portanto, se um estado final é inútil é porque não é relevante.

A partir de um autómato $D = (Q, I, \delta, q_0, F)$ com estados inúteis e tal que $L_D \neq \emptyset$ é possível construir um outro autómato D' sem estados inúteis e tal que $L_D = L_{D'}$. A proposição seguinte enuncia este resultado. A prova desta proposição encontra-se no Apêndice. Note-se que se $L_D \neq \emptyset$ então o estado inicial é produtivo. Assim, existe pelo menos um estado final que é relevante e, consequentemente, existe um estado final que não é inútil.

Proposição: Seja $D = (Q, I, \delta, q_0, F)$ um autómato finito determinista tal que $L_D \neq \emptyset$ e seja In_D o conjunto dos estados inúteis de D. O autómato

$$D' = (Q', I, \delta', q_0, F \backslash In_D)$$

onde

- $Q' = Q \backslash In_D$
- $\delta' : Q' \times I \to Q'$ tal que
  - $\delta'(q, i)$ não definido se $\delta(q, i) \in In_D$ ou $\delta(q, i)$ não definido
  - $\delta'(q, i) = \delta(q, i)$ caso contrário

[2]O único caso em que estados inúteis são necessários é a situação em que se pretende que um autómato D não aceite nenhuma sequência, isto é, $L_D = \emptyset$.

não tem estados inúteis e $L_D = L_{D'}$.

Note-se que é sempre possível calcular em tempo finito o conjunto dos estados inúteis de um autómato D.

Para calcular o conjunto dos estados não relevantes podem, por exemplo, construir-se os conjuntos Γ_0, $\Gamma_1, \ldots$ como explicado no âmbito do algoritmo para determinar se $L_D = \emptyset$, até se encontrar $k \geq 0$ tal que $\Gamma_k = \Gamma_{k+1}$. O conjunto $\Gamma = \Gamma_k$ é o conjunto de todos os estados relevantes. Os elementos de $Q \backslash \Gamma$ são os estados não relevantes.

Para determinar se um estado $q \in Q$ é ou não produtivo pode usar-se um algoritmo idêntico ao apresentado para determinar se $L_D = \emptyset$, mas considerando $\Gamma_0 = \{q\}$ (em vez de $\Gamma_0 = \{q_0\}$). Uma outra forma de calcular o conjunto dos estados não produtivos é construir-se os conjuntos Π_0, $\Pi_1, \ldots$ onde $B_0 = \Pi_0 = F$, $B_1 = \{q \in Q : \delta(q,i) \in B_0$ para algum $i \in I\}$ e $\Pi_1 = \Pi_0 \cup B_0$, etc. até se encontrar $k \geq 0$ tal que $\Pi_k = \Pi_{k+1}$. O conjunto $\Pi = \Pi_k$ é o conjunto de todos os estados produtivos. Os elementos de $Q \backslash \Pi$ são os estados não produtivos.

Definição: Estados equivalentes e estados distinguíveis
Seja $D = (Q, I, \delta, q_0, F)$ um autómato finito determinista e sejam $p, q \in Q$

- p e q são equivalentes em D sse, para cada $x \in I^*$

$$\delta^*(p,x) \in F \text{ sse}^3\ \delta^*(q,x) \in F;$$

- p e q são distinguíveis em D sse não são equivalentes, ou seja,

$$\text{existe } x \in I^* \text{ que distingue } p \text{ e } q \text{, i.e.,}$$
$$\text{existe } x \in I^* \text{ tal que } \delta^*(p,x) \in F \text{ e}^4\ \delta^*(q,x) \notin F \text{ ou vice-versa.}$$

Como é evidente, se $p = q$ os estados p e q são equivalentes. Informalmente, dois estados p e q são equivalentes sempre que, para cada sequência $x \in I^*$, se a partir de p se pode "chegar" a um estado final "lendo" todos os símbolos de x então o mesmo acontece a partir de q e vice-versa. Naturalmente, p e q são distinguíveis se existir uma sequência x que os distigue, ou seja, se a partir de p se pode "chegar" a um estado final "lendo" todos os símbolos de x, mas o mesmo não acontece a partir[5] de q, ou vice-versa.

Descreve-se seguidamente um algoritmo que determina todos os pares de estados distinguíveis de um autómato (e, consequentemente, todos os pares equivalentes).

Observação: No que se segue, sempre que se faça referência a um par de estados de um autómato D (cujo conjunto de estados é Q) tal deve ser entendido

[3]Naturalmente, $\delta^*(p,x) \in F$ sse $\delta^*(p,x)$ está definido e $\delta^*(p,x)$ é um estado final.

[4]Naturalmente, $\delta^*(p,x) \notin F$ sse $\delta^*(p,x)$ não está definido ou $\delta^*(p,x) = p \in Q \backslash F$.

[5]ou porque se atinge um estado que não é final ou porque, a certa altura, se atinge um estado a partir do qual não existe transição associada ao símbolo de entrada que devia ser reconhecido

como um subconjunto de Q com cardinal[6] 2. Um par de estados $\{p, q\}$ será por vezes, por simplicidade, representado apenas por p, q (note-se que, de acordo com esta observação, p e q são distintos).

Algoritmo de procura exaustiva de pares distinguíveis:
Sendo $D = (Q, I, \delta, q_0, F)$ um autómato finito determinista tal que se δ não é função total então todos os estados são produtivos[7], o objectivo é encontrar todos os pares de estados distinguíveis de D (e, consequentemente, todos os pares equivalentes). Para tal procede-se do modo que seguidamente se descreve.

1. (Passo inicial) Inicialmente consideram-se todos os pares de estados q_1, q_2 que verifiquem alguma das condições seguintes onde $i, j \in \{1, 2\}$, $i \neq j$

   – $q_i \in F$ e $q_j \notin F$
   (i.e., procuram-se todos os pares que incluam um estado final e um estado não final)

   – existe $a \in I$ tal que
      - $\delta(q_i, a)$ está definido
      - $\delta(q_j, a)$ não está definido

   (i.e., procuram-se todos os pares de estados que incluam um estado a partir do qual exista uma transição associada a um dado símbolo de entrada a, e um estado a partir do qual não exista transição associada a a)

   Estes pares de estados são identificados como distinguíveis.

2. (Passo iterativo) Procura-se agora identificar novos pares de estados distinguíveis a partir de pares de estados que já tenham sido identificados como distinguíveis pelo algoritmo. Para tal, para cada $a \in I$ e para cada par de estados p, q já identificados como distinguíveis procuram-se estados p' e q' tais que

   – $\delta(p', a) = p$
   – $\delta(q', a) = q$.

   Estes estados p' e q' são identificados como distinguíveis.

3. (Condição de terminação) O algoritmo termina quando *alguma* das condições seguintes se verificar

   – já todos os pares de estados do autómato foram identificados como distinguíveis pelo algoritmo
   – o passo iterativo já foi aplicado a todos os pares identificados como distinguíveis pelo algoritmo.

[6] Não se trata portanto de pares *ordenados*, ou seja entidades do tipo $(p, q) \in Q \times Q$.
[7] Se δ não for uma função total, não há perda de generalidade pois, como foi referido, é sempre possível encontrar um autómato apenas com estados produtivos que reconheça exactamente a mesma linguagem.

Quando o algoritmo termina[8], todos os pares de estados p e q (de Q) que não foram considerados distinguíveis pelo algoritmo são pares de estados equivalentes em D (e, naturalmente, são todos os pares equivalentes em D que existem).

Exemplo 1: Considere-se o autómato finito determinista $D = (Q, I, \delta, q_0, F)$ onde

- $Q = \{q_0, q_1, q_2\}$
- $I = \{0, 1\}$
- $\delta : Q \times I \to Q$ tal que

δ	0	1
q_0	q_1	q_2
q_1	q_1	q_2
q_2	q_0	q_2

- $F = \{q_0, q_1\}$.

Aplicando o algoritmo de procura exaustiva de pares distinguíveis obtém-se a seguinte tabela (onde $\surd$ identifica pares de estados distinguíveis).

q_1		
q_2	$\surd$	$\surd$
	q_0	q_1

Num primeiro passo, identificam-se como distinguíveis os pares constituídos por um estado final e um estado não final, isto é, $\{q_0, q_2\}$ e $\{q_1, q_2\}$. A partir destes não se encontram mais pares distinguíveis pois não existem $q', q'' \in Q$ distintos e $i \in I$ tal que $\delta(q', i) = q_0$ e $\delta(q'', i) = q_2$ ou tal que $\delta(q', i) = q_1$ e $\delta(q'', i) = q_2$. Conclui-se assim que $\{q_0, q_1\}$ é o único par de estados equivalentes.

Exemplo 2: Considere-se o autómato finito determinista $D = (Q, I, \delta, q_0, F)$ onde

- $Q = \{q_0, q_1, q_2, q_3\}$
- $I = \{a, b\}$
- $\delta : Q \times I \to Q$ tal que

δ	a	b
q_0	q_1	q_2
q_1	q_3	q_3
q_2	nd	q_3
q_3	q_3	q_3

[8] e termina sempre, porque existe apenas um número finito de pares de estados e I é finito

- $F = \{q_3\}$.

Aplicando o algoritmo de procura exaustiva de pares distinguíveis constrói-se a seguinte tabela (onde $\surd$ identifica pares de estados distinguíveis). Note-se que todos os estados deste autómato são produtivos.

q_1	$\surd$		
q_2	$\surd$	$\surd$	
q_3	$\surd$	$\surd$	$\surd$
	q_0	q_1	q_2

Num primeiro passo, identificam-se como distinguíveis os pares constituídos por um estado final e um estado não final, isto é, $\{q_0, q_3\}$ e $\{q_1, q_3\}$ e $\{q_2, q_3\}$. Neste primeiro passo identificam-se também como distinguíveis os pares $\{q_0, q_2\}$ e $\{q_1, q_2\}$. Dado que $\delta(q_0, a)$ está definido e $\delta(q_2, a)$ não está, obtém-se o par $\{q_0, q_2\}$. Um raciocínio análogo pode ser feito para obter o par $\{q_1, q_2\}$.

No passo seguinte, a partir do facto de, por exemplo, o par $\{q_1, q_3\}$ já ter sido identificado como distinguível, e de $\delta(q_0, a) = q_1$ e $\delta(q_1, a) = q_3$ conclui-se que $\{q_0, q_1\}$ são distinguíveis. Conclui-se assim que não existem pares de estados equivalentes.

Exemplo 3: Considere-se o autómato finito determinista $D = (Q, I, \delta, q_0, F)$ onde

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$
- $I = \{0, 1\}$
- $\delta : Q \times I \to Q$ tal que

δ	0	1
q_0	q_1	q_5
q_1	q_6	q_2
q_2	q_0	q_2
q_3	q_2	q_6
q_4	q_7	q_5
q_5	q_2	q_6
q_6	q_6	q_4
q_7	q_6	q_2

- $F = \{q_2\}$.

Aplicando o algoritmo de procura exaustiva de pares distinguíveis constrói-se a seguinte tabela (onde $\surd$ identifica pares de estados distinguíveis).

q_1	$\surd$						
q_2	$\surd$	$\surd$					
q_3	$\surd$	$\surd$	$\surd$				
q_4		$\surd$	$\surd$	$\surd$			
q_5	$\surd$	$\surd$	$\surd$		$\surd$		
q_6	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	
q_7	$\surd$		$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
	q_0	q_1	q_2	q_3	q_4	q_5	q_6

Num primeiro passo, identificam-se como distinguíveis os pares constituídos por estado final e um estado não final, isto é, todos os pares que incluam q_2 ($\{q_0, q_2\}$, $\{q_1, q_2\}$, $\{q_2, q_3\}$, etc.) Outros pares distinguíveis obtêm-se partindo destes pares inicialmente identificados. Por exemplo, pelo facto de q_2 e q_7 serem distinguíveis e porque

- $\delta(q_3, 0) = \delta(q_5, 0) = q_2$
- $\delta(q_4, 0) = q_7$

conclui-se que os pares $\{q_3, q_4\}$ e $\{q_4, q_5\}$ são também distinguíveis. Todos os pares distinguíveis que não incluem q_2, excepto os pares $\{q_0, q_6\}$ e $\{q_4, q_6\}$, são obtidos a partir dos que incluem q_2. Os pares $\{q_0, q_6\}$ e $\{q_4, q_6\}$ são obtidos do par $\{q_4, q_5\}$ tendo em conta que

- $\delta(q_0, 1) = \delta(q_4, 1) = q_5$
- $\delta(q_6, 1) = q_4$

Conclui-se assim que existem apenas os seguintes pares de estados equivalentes: $\{q_0, q_4\}$, $\{q_1, q_7\}$ e $\{q_3, q_5\}$.

As proposições seguintes mostram que todos os pares identificados como distinguíveis pelo algoritmo anterior são de facto distinguíveis e que o algoritmo encontra *todos* os pares de estados distinguíveis (e, portanto, todos os pares de estados equivalentes de um autómato). As proposições são aqui apenas enunciadas. As provas encontram-se no Apêndice 1.

Proposição: Os pares identificados como distinguíveis no decorrer do algoritmo de procura exaustiva de pares distinguíveis acima descrito são de facto pares de estados distinguíveis do autómato.

Proposição: O algoritmo de procura exaustiva de pares distinguíveis acima descrito encontra todos pares de estados distinguíveis do autómato (e portanto, qualquer par de estados que não sejam identificados pelo algoritmo como distinguíveis é um par de estados equivalentes).

Mostra-se seguidamente como o algoritmo de procura exaustiva de pares distinguíveis pode ser usado para determinar se dois autómatos finitos deterministas são equivalentes, no sentido em que as linguagens reconhecidas por ambos são iguais.

Algoritmo de teste à igualdade das linguagens reconhecidas por dois autómatos finitos deterministas:

Sejam $D_1 = (Q_1, I, \delta_1, q_0^1, F_1)$ e $D_2 = (Q_2, I, \delta_2, q_0^2, F_2)$ dois autómatos finitos deterministas tais que se δ_1 ou δ_2 não forem funções totais então todos os estados de D_1 e todos os estados de D_2 são produtivos[9]. Assuma-se, sem perda de generalidade, que $Q_1 \cap Q_2 = \emptyset$. O objectivo é saber se $L_{D_1} = L_{D_2}$ ou $L_{D_1} \neq L_{D_2}$. Para tal procede-se do modo que seguidamente se descreve.

1. Considera-se o autómato $D = (Q, I, \delta, q_0, F)$ onde
   - $Q = Q_1 \cup Q_2$
   - $\delta : Q \times I \to Q$ é tal que
     - $\delta(q, a) = \delta_1(q, a)$ se $q \in Q_1$ e $\delta_1(q, a)$ está definido
     - $\delta(q, a) = \delta_2(q, a)$ se $q \in Q_2$ e $\delta_2(q, a)$ está definido
     - $\delta(q, a)$ não definido, nos outros casos
   - $q_0 = q_0^1$ (poder-se-ia também escolher $q_0 = q_0^2$)
   - $F = F_1 \cup F_2$.
2. Procuram-se todos os pares de estados equivalentes de D, usando o algoritmo de procura exaustiva de pares distinguíveis atrás descrito.
3. Se q_0^1 e q_0^2 não foram identificados como distinguíveis[10], $L_{D_1} = L_{D_2}$; caso contrário $L_{D_1} \neq L_{D_2}$.

Exemplo 4: Considerem-se os autómatos finitos deterministas $D_1 = (Q_1, \{a, b\}, \delta_1, q_0^1, F_1)$ e $D_2 = (Q_2, \{a, b\}, \delta_2, q_0^2, F_2)$ onde

- $Q_1 = \{q_0, q_1, q_2\}$
- $\delta_1 : Q_1 \times \{a, b\} \to Q_1$ tal que

δ	a	b
q_0	q_1	nd
q_1	q_2	q_1
q_2	q_2	q_1

- $F_1 = \{q_1, q_2\}$

e

- $Q_2 = \{p_0, p_1\}$

[9] No caso de alguma das funções de transição não ser aplicação, não existe perda de generalidade. Se algum dos autómatos tiver estados não produtivos, usando a construção atrás descrita, pode construir-se um autómato que reconheça a mesma linguagem e não tenha estados não produtivos.

[10] e portanto, como se viu anteriormente, são estados equivalentes

- $\delta_2 : Q_2 \times \{a, b\} \to Q_2$ tal que

δ_2	a	b
p_0	p_1	nd
p_1	p_1	p_1

- $F_2 = \{p_1\}$.

Pretende-se determinar se as linguagens reconhecidas pelos autómatos são ou não iguais, usando o algoritmo acima descrito. Para tal considera-se o autómato

$$D = (\{q_0, q_1, q_2, p_0, p_1\}, \{a, b\}, \delta, q_0^1, \{q_1, q_2, p_1\})$$

onde $\delta : \{q_0, q_1, q_2, p_0, p_1\} \times \{a, b\} \to \{q_0, q_1, q_2, p_0, p_1\}$ é

δ	a	b
q_0	q_1	nd
q_1	q_2	q_1
q_2	q_2	q_1
p_0	p_1	nd
p_1	p_1	p_1

Usando agora o algoritmo de procura exaustiva de pares distinguíveis constrói-se a seguinte tabela (onde $\surd$ identifica pares de estados distinguíveis). Note-se que todos os estados deste autómato são produtivos.

p_1	$\surd$			
q_0		$\surd$		
q_1	$\surd$		$\surd$	
q_2	$\surd$		$\surd$	
	p_0	p_1	q_0	q_1

Conclui-se assim que os estados q_0 e p_0 (os estados iniciais dos dois autómatos) são equivalentes. Deste modo $L_{D_1} = L_{D_2}$.

Na proposição seguinte enuncia-se o resultado que garante que o algoritmo anterior está correcto. A prova desta proposição encontra-se no Apêndice.

Proposição: Sejam $D_1 = (Q_1, I, \delta_1, q_0^1, F_1)$ e $D_2 = (Q_2, I, \delta_2, q_0^2, F_2)$ dois autómatos finitos deterministas nas condições enunciadas no algoritmo de teste à igualdade de linguagens reconhecidas por autómatos finitos deterministas. Se o algoritmo termina com a indicação de que $L_{D_1} = L_{D_2}$ então tem-se, de facto, a igualdade das linguagens. Se o algoritmo termina com a indicação de que $L_{D_1} \neq L_{D_2}$ então as linguagens são, de facto, diferentes.

Mostra-se seguidamente como o algoritmo de procura exaustiva de pares distinguíveis pode ser usado para determinar se, dado um autómato finito determinista D, existe um outro autómato finito determinista D' cuja linguagem seja também L_D mas que tenha menos estados que D.

Algoritmo de minimização de autómatos:

Sendo $D = (Q, I, \delta, q_0, F)$ um autómato finito determinista, o objectivo é encontrar um autómato finito determinista cuja linguagem seja também L_D mas que tenha o menor número possível de *estados*. Mais precisamente pretende-se encontrar um autómato finito determista $D' = (Q', I, \delta', q_0', F')$ tal que

- $L_{D'} = L_D$
- se $D'' = (Q'', I, \delta'', q_0'', F'')$ é um outro autómato finito determinista tal que $L_{D''} = L_D$ então $\#Q' \leq \#Q''$.

Para tal procede-se do modo que seguidamente se descreve.

1. Calcula-se o conjunto In_D dos estados inúteis de D.

   1.1 Se $In_D = \emptyset$ então passa-se para 2. fazendo $D_1 = D$.

   1.2 Se $In_D \neq \emptyset$

   1.2.1 se $q_0 \in In_D$ (o que significa que L_D é vazio[11]), o autómato

   $$D' = (\{q_0, q_1\}, I, \delta', q_0, \{q_1\})$$

   onde $\delta'(q, a)$ não está definido para nenhum $a \in I$ é o autómato pretendido.

   1.2.2 Se $q_0 \notin In_D$ então considera-se o autómato

   $$D_1 = (Q_1, I, \delta_1, q_0, F_1)$$

   onde

   – $Q_1 = Q \backslash In_D$

   – $\delta_1 : Q_1 \times I \to Q_1$ tal que

   $\delta_1(q, a)$ não definido se $\delta(q, a) \in In_D$ ou $\delta(q, a)$ não definido

   $\delta_1(q, a) = \delta(q, a)$ caso contrário

   – $F_1 = F \backslash In_D$.

2. Procuram-se todos os pares de estados equivalentes de D_1 usando o algoritmo de procura exaustiva de pares distinguíveis atrás descrito.

   2.1 Se não existem pares de estados equivalentes então D_1 é o autómato pretendido.

   2.2 Se existem pares de estados equivalentes então passa-se para 3.

3. Considera-se a relação $R \subseteq Q_1 \times Q_1$ tal que $(p, q) \in R$ sse p e q são estados equivalentes em D_1 e constrói-se para cada $q \in Q_1$ o conjunto[12]

   $$\{q' \in Q_1 : (q, q') \in R\}$$

[11] Se $q_0 \in In_D$ é porque q_0 é um estado não produtivo (q_0 é sempre relevante) e, consequentemente, $L_D = \emptyset$.

[12] i.e., calcula-se o conjunto de todos os estados equivalentes a q (note-se que qualquer estado é equivalente a si próprio).

obtendo-se n conjuntos distintos

$$C_1, \ldots, C_n$$

com $n \leq \#Q_1$, que são subconjuntos de Q_1 não vazios.

4. O autómato $D' = (Q', I, \delta', q_0', F')$ onde
   - $Q' = \{C_1, \ldots, C_n\}$
   - $\delta' : Q' \times I \to Q'$ tal que
     - $\delta'(C, a) = C'$ se $\delta_1(q, a) \in C'$ para algum $q \in C$
     - $\delta'(C, a)$ não definido se $\delta_1(C, a)$ não definido para algum $q \in C$
   - $q_0' = C$ tal que $q_0 \in C$
   - $F' = \{C \in Q' : C \cap F_1 \neq \emptyset\}$

   é o autómato pretendido.

Observação: Seguem-se algumas observações relativas aos pontos 3. e 4. do algoritmo anterior.

- Em 3.
  - considerando os conjuntos $C_1, \ldots, C_n$ e sendo $1 \leq i, j \leq n$ e $i \neq j$
    - os conjuntos constituem uma partição de Q_1, isto é, para além de serem não vazios, são disjuntos dois a dois ($C_i \cap C_j = \emptyset$) e $Q_1 = \bigcup_{1 \leq k \leq n} C_k$;
    - se $p, q \in C_i$ então p e q são estados equivalentes;
    - se $p \in C_i$ e $q \in C_j$ então p e q são estados distinguíveis[13].
- Em 4., considerando os conjuntos $C_1, \ldots, C_n$, $a \in I$ e sendo $1 \leq i, j \leq n$
  - se $\delta_1(q, a) \in C_j$ para algum $q \in C_i$ então $\delta_1(q, a) \in C_j$ para qualquer $q \in C_i$ (ver Apêndice);
  - se $\delta_1(q, a)$ não está definido para algum $q \in C_i$ então $\delta_1(q, a)$ não está definido para nenhum $q \in C_i$ (ver Apêndice);
  - tem-se que se $C_i \cap F_1 \neq \emptyset$ então[14] $C_i \subseteq F_1$ (ver Apêndice).

Exemplo 5: Considere-se o autómato finito determinista $D = (Q, I, \delta, q_0, F)$ descrito no Exemplo 1. Pretende-se aplicar a D o algoritmo de minimização atrás descrito.

- Não existem estados inúteis pelo que se passa para o passo 2.

[13] ou seja, se dois estados estão no mesmo conjunto então são equivalentes e se estão em conjuntos diferentes então são distinguíveis

[14] ou seja, se um dos conjuntos contém um estado final então todos os outros elementos desse conjunto são também estados finais

- A tabela construída no referido exemplo identifica q_1 e q_0 como o único par de estados equivalentes pelo que
  - os estados q_0 e q_1 são equivalentes
  - o estado q_2 é equivalente a si próprio.

  Assim, tem-se uma partição de Q em dois conjuntos
  - $C_1 = \{q_0, q_1\}$
  - $C_2 = \{q_2\}$
- Constrói-se o autómato $D' = (Q', I, \delta', q'_0, F')$ onde
  - $Q' = \{C_1, C_2\}$
  - $\delta : Q' \times I \to Q'$ tal que

δ	0	1
C_1	C_1	C_2
C_2	C_1	C_2

  - $q'_0 = C_1$
  - $F' = \{C_1\}$

Exemplo 6: Considere-se o autómato finito determinista $D = (Q, I, \delta, q_0, F)$ descrito no Exemplo 3. Pretende-se aplicar a D o algoritmo de minimização atrás descrito.

- Não existem estados inúteis pelo que se passa para o passo 2.
- A tabela construída no referido exemplo identifica $\{q_0, q_4\}$, $\{q_1, q_7\}$ e $\{q_3, q_5\}$ como os únicos pares de estados equivalentes pelo que se tem uma partição de Q em cinco conjuntos
  - $C_1 = \{q_0, q_4\}$
  - $C_2 = \{q_1, q_7\}$
  - $C_3 = \{q_3, q_5\}$
  - $C_4 = \{q_2\}$
  - $C_5 = \{q_6\}$
- Constrói-se o autómato $D' = (Q', I, \delta', q'_0, F')$ onde
  - $Q' = \{C_1, C_2, C_3, C_4, C_5\}$
  - $\delta : Q' \times I \to Q'$ tal que

δ	0	1
C_1	C_2	C_3
C_2	C_5	C_4
C_3	C_4	C_5
C_4	C_1	C_4
C_5	C_5	C_1

- $q'_0 = C_1$
- $F' = \{C_4\}$

Na proposição seguinte enuncia-se o resultado que garante que o algoritmo de minimização anterior está correcto. A prova desta proposição encontra-se no Apêndice.

Proposição: Seja $D = (Q, I, \delta, q_0, F)$ um autómato finito determinista. O autómato D' construído no contexto do algoritmo de minimização de autómatos anterior é tal que

- $L_{D'} = L_D$
- se $D'' = (Q'', I, \delta'', q''_0, F'')$ é um outro autómato finito determinista tal que $L_{D''} = L_D$ então $\#Q' \leq \#Q''$.

Note-se que se não forem encontrados pares de estados equivalentes em D_1 então, considerando os subconjuntos $C_1, \ldots, C_n$ de Q_1 construídos no ponto 3 do algoritmo anterior, tem-se que cada um deles tem um único elemento. Consequentemente, $n = \#Q_1$ e o autómato D' tem o mesmo número de estados que o autómato de partida D (se este não tiver estados inúteis), o que significa que não existe nenhum autómato finito determinista com menos estados que D e que reconheça exactamente a mesma linguagem que D. Se forem encontrados pares de estados equivalentes, existe $1 \leq j \leq n$ tal que C_j tem mais de um elemento e $n < \#Q_1$. Assim, o autómato D' tem menos estados que o autómato de partida D, o que significa D não é o autómato com menor número de estados que é possível construir para reconhecer exactamente L_D.

O algoritmo seguinte determina se sim ou não um dado autómato finito determinista D é o autómato finito determinista com menor número de estados que é possível construir para reconhecer exactamente L_D.

Algoritmo para determinar se um autómato finito determinista tem o menor número possível de estados:

Sendo $D = (Q, I, \delta, q_0, F)$ um autómato finito determinista, o objectivo é saber se sim ou não existe um autómato finito determinista D' com menos estados que D e cuja linguagem é L_D. Para tal procede-se do modo que seguidamente se descreve.

1. Calcula-se o conjunto In_D dos estados inúteis de D.

   1.1 Se $In_D \neq \emptyset$ e $D = (Q, I, \delta, q_0, F)$ onde Q tem apenas dois elementos, $q_0 \notin F$ e $\delta(q, i)$ não está definido para quaisquer $q \in F$ e $i \in I$, então não existe um tal autómato D'. Caso contrário, existe D' nas condições descritas.

   1.2 Se $In_D = \emptyset$ passa-se para 2.

2. Procuram-se todos os pares de estados equivalentes de D usando o algoritmo de procura exaustiva de pares distinguíveis atrás descrito. Se existirem pares de estados equivalentes então existe um tal autómato D'. Caso contrário não existe D' nas condições descritas.

Exemplo 7:

- A partir da tabela construída no Exemplo 1 conclui-se que o autómato D aí apresentado não é o autómato finito determinista com o menor número de estados que é possível construir para reconhecer exactamente L_D.
- A partir da tabela construída no Exemplo 2 e tendo em conta que D não tem estados inúteis, conclui-se que não existe nenhum autómato finito determinista D' que reconheça exactamente L_D e tenha menos estados que D.
- A partir da tabela construída no Exemplo 3 conclui-se que o autómato D não é o autómato finito determinista com o menor número de estados que é possível construir para reconhecer exactamente L_D.

A correcção do algoritmo anterior é uma consequência simples da correcção do algoritmo de minimização.

Apêndice: Provas das proposições enunciadas

Proposição: Seja $D = (Q, I, \delta, q_0, F)$ um autómato finito determinista tal que $L_D \neq \emptyset$ e seja In_D o conjunto dos estados inúteis de D. O autómato

$$D' = (Q', I, \delta', q_0, F \backslash In_D)$$

onde

- $Q' = Q \backslash In_D$
- $\delta' : Q' \times I \to Q'$ tal que
  - $\delta'(q, a)$ não definido se $\delta(q, a) \in In_D$ ou $\delta(q, a)$ não definido
  - $\delta'(q, a) = \delta(q, a)$ caso contrário

não tem estados inúteis e $L_D = L_{D'}$.

Prova (esboço): Esta prova não foi apresentada nas aulas teóricas Se $L_D \neq \emptyset$ então $q_0 \notin In_D$ e existe um estado final relevante. Assim, $q_0 \in Q'$ e o conjunto dos estados finais não é vazio pelo D' está bem definido.

Dado que Q' é constituído por todos os estados de Q que são relevantes e produtivos e tendo em conta a definição de δ' facilmente se conclui que D' não tem estados inúteis.

Para provar que $L_D = L_{D'}$ basta considerar que, dado $wx \in L_D$, uma demonstração $p_0 \ldots p_n$ de w em D só contém estados em $Q \backslash In_D$. Tendo em conta a definição de δ', conclui-se facilmente que $p_0 \ldots p_n$ é também uma demonstração de w em D' e $\delta'^*(q_0', w) \in F \backslash In_D$, pelo que $w \in L_{D'}$. Um raciocínio análogo pode ser feito para concluir que se $w \in L_{D'}$ então $w \in L_D$.

Proposição: Os pares identificados como distinguíveis no decorrer do algoritmo de procura exaustiva de pares distinguíveis acima descrito são de facto pares de estados distinguíveis do autómato.

Prova (esboço): (A) Mostra-se, em primeiro lugar, que se $\{p, q\}$ são identificados como distinguíveis no passo 1 do algoritmo então são de facto distinguíveis.

(i) Se um dos estados (suponha-se p) é um estado final e o outro não é, então ϵ distingue os dois estados pois $\delta^*(p, \epsilon) = p \in F$ e $\delta^*(q, \epsilon) = q \notin F$.

(ii) Seja $a \in I$ tal que $\delta(p, a) = p'$ e $\delta(q, a)$ não está definido (o caso contrário é análogo). Como, por hipótese, todos os estados do autómato são produtivos, existe $x \in I^*$ tal que

$$\delta^*(p', x) \in F.$$

Facilmente se prova então que

$$\delta^*(p, ax) \in F.$$

Mas, pelo facto de $\delta(q, a)$ não estar definido, facilmente se prova também que

$$\delta^*(q, ax) \notin F$$

e portanto a sequência ax distingue p e q.

(B) Prova-se agora que se $\{p, q\}$ são distinguíveis e existem $a \in I$ e $p', q' \in Q$ tais que $\delta(p', a) = p$ e $\delta(q', a) = q$ então os estados p' e q' são distinguíveis.

Se p e q são distinguíveis então existe $x \in I^*$ que os distingue, i.e,

$$\delta^*(p, x) \in F \text{ e } \delta^*(q, x) \notin F \text{ (ou vice-versa).}$$

Dado que

$$\delta(p', a) = p \text{ e } \delta(q', a) = q$$

facilmente se conclui que

$$\delta^*(p', ax) \in F \text{ e } \delta^*(q', ax) \notin F$$

e portanto ax distingue p' e q'.

(C) A partir da análise do algoritmo é possível concluir que para cada par de estados $\{p, q\}$ que são identificados pelo algoritmo como distinguíveis é possível encontrar uma sequência

$$\{p_1, q_1\}\{p_2, q_2\} \ldots \{p_n, q_n\}$$

$n \geq 1$, de pares de estados do autómato tal que $\{p, q\} = \{p_n, q_n\}$, $\{p_1, q_1\}$ são identificados como distinguíveis no passo 1 do algoritmo e, para cada $1 \leq j < n$, $\{p_{j+1}, q_{j+1}\}$ é obtido como se descreve no passo 2 do algoritmo a partir do par $\{p_j, q_j\}$ (também previamente identificados pelo algoritmo).

Tendo em conta os resultados provados em (A) e (B), é possível então provar (por indução) que cada par de estados $\{p, q\}$ que são identificados pelo algoritmo com distinguíveis são de facto distinguíveis. Deixa-se como exercício esta prova.

Proposição: O algoritmo de procura exaustiva de pares distinguíveis acima descrito encontra todos os pares de estados distinguíveis do autómato (e portanto, qualquer par de estados que não sejam identificados pelo algoritmo como distinguíveis é um par de estados equivalentes).

Prova: Esta prova não foi apresentada nas aulas teóricas
Vai fazer-se uma prova por absurdo, ou seja, assume-se que a afirmação é falsa (i.e., existem pares de estados distinguíveis que não são identificados pelo algoritmo) e vai chegar-se a uma contradição. Seja $D = (Q, I, \delta, q_0, F)$ um autómato nas condições enunciadas na descrição do algoritmo.

Se a afirmação é falsa, o conjunto

$$A = \{ \{p, q\} : p, q \in Q \text{ são distinguíveis mas não são identificados como tal pelo algoritmo}\}$$

não é vazio.

1. Sejam $\{p, q\} \in A$ e $w = a_1 \ldots a_n \in I^*$ tais que[15]

[15] i.e., (i) os estados p e q são distinguíveis mas o algoritmo não os identifica como tal, (ii) w distingue p e q e (iii) não existem pares de estados em A (ou seja, pares de estados distinguíveis mas não identificados como tal pelo algoritmo) que sejam distinguidos por sequências de comprimento menor que o de w.

- w distingue p e q
- não existem $\{p', q'\} \in A$ e $w' \in I^*$ tais que w' distingue p' e q' e $|w'|<|w|$.

2. Por 1., em particular, p e q são distinguíveis e $\delta^*(p, w) \in F$ e $\delta^*(q, w) \notin F$ (ou vice-versa, mas o raciocínio seria idêntico).

3. De 2., se $w = \epsilon$, então $p \in F$ e $q \notin F$. Assim, p e q são distinguíveis e teriam sido identificados como tal no passo 1 do algoritmo pelo que $\{p, q\} \notin A$. Tem-se então uma contradição com 1. a qual permite concluir que, necessariamente, $w \neq \epsilon$ e portanto $n \geq 1$.

4. Por 2. e 3., tem-se que $\delta(p, a_1)$ está definido (suponha-se $\delta(p, a_1) = r$) mas $\delta(q, a_1)$ pode ou não estar definido.

   (a) Se $\delta(q, a_1)$ não está definido, então p e q são distinguíveis (ver proposição anterior) e no passo 1 do algoritmo teriam sido identificados como tal, pelo que $\{p, q\} \notin A$. Tem-se então de novo uma contradição com 1. a qual permite concluir que, necessariamente, $\delta(q, a_1)$ tem de estar definido.

   (b) Se $\delta(q, a_1)$ está definido (suponha-se $\delta(q, a_1) = s$), então, por 2., facilmente se conclui que $\delta^*(r, a_2 \dots a_n) \in F$ e $\delta^*(s, a_2 \dots a_n) \notin F$ pelo que r e s são distingidos por $w' = a_2 \dots a_n$ com $|w'|<|w|$. Por 1., $\{r, s\} \notin A$.

5. Como, por 4., r e s são distinguíveis e $\{r, s\} \notin A$, tal significa que r e s foram identificados como distinguíveis pelo algoritmo.

6. Tendo em conta 5., tem-se que durante o algoritmo o par $\{r, s\}$ e o símbolo a_1 tiveram de ser analisados na tentativa de encontrar novos estados distinguíveis e como, por 4., $\delta(p, a_1) = r$ e $\delta(q, a_1) = s$, o algoritmo identifica p e q como distinguíveis. Como, por 2., p e q são, de facto, distinguíveis, $\{p, q\} \notin A$.

7. Chega-se a uma contradição entre 1. e 6. a qual permite então concluir que o enunciado desta proposição é necessariamente uma asserção verdadeira.

Proposição: Sejam $D_1 = (Q_1, I, \delta_1, q_0^1, F_1)$ e $D_2 = (Q_2, I, \delta_2, q_0^2, F_2)$ dois autómatos finitos deterministas nas condições enunciadas no algoritmo de teste à igualdade de linguagens reconhecidas por autómatos finitos deterministas. Se o algoritmo termina com a indicação de que $L_{D_1} = L_{D_2}$ então tem-se, de facto, a igualdade das linguagens. Se o algoritmo termina com a indicação de que $L_{D_1} \neq L_{D_2}$ então as linguagens são, de facto, diferentes.

Prova (esboço): Esta prova não foi apresentada nas aulas teóricas
Seja $D = (Q, I, \delta, q_0, F)$ o autómato construído no ponto 1. do algoritmo.

(A) Da definição de δ facilmente se conclui que, para cada $j \in \{1, 2\}$, $q \in Q_j$ e $w \in I^*$, $\delta^*(q, w) \in F$ sse $\delta_j^*(q, w) \in F_j$.

(B) Se o algoritmo termina com a indicação de que $L_{D_1} = L_{D_2}$, então é porque o algoritmo de procura exaustiva de pares distinguíveis não identifica como distinguíveis os estados q_0^1 e q_0^2. Pelas duas proposições anteriores, estes estados são equivalentes. Assim, para cada $w \in I^*$, $\delta^*(q_0^1, w) \in F$ sse $\delta^*(q_0^2, w) \in F$. Por (A), para cada $w \in I^*$, $\delta_1^*(q_0^1, w) \in F_1$ sse $\delta_2^*(q_0^2, w) \in F_2$, ou seja, $L_{D_1} = L_{D_2}$.

(C) Se o algoritmo termina com a indicação de que $L_{D_1} \neq L_{D_2}$, então é porque o algoritmo de procura exaustiva de pares distinguíveis identifica como distinguíveis os estados q_0^1 e q_0^2. Por uma das proposições anteriores, q_0^1 e q_0^2 são estados distinguíveis pelo que existe $w \in I^*$ tal que $\delta^*(q_0^1, w) \in F$ e $\delta^*(q_0^2, w) \notin F$ (ou vice-versa, mas o raciocínio é o mesmo). Por (A), $\delta_1^*(q_0^1, w) \in F_1$ e $\delta_2^*(q_0^2, w) \notin F_2$, ou seja, $L_{D_1} \neq L_{D_2}$.

Antes de provar a correcção do algoritmo de minimização de autómatos, provam-se algumas propriedades relacionadas com os estados do autómato D' construído no ponto 4. do algoritmo de minimização de autómatos.

Proposição: Seja $D = (Q, I, \delta, q_0, F)$ um autómato finito determinista, seja $D_1 = (Q_1, I, \delta_1, q_0^1, F_1)$ o autómato considerado no ponto 2 do algoritmo de minimização de autómatos anteriormente apresentado e seja $D' = (Q', I, \delta', q_0', F')$ o autómato construído no ponto 4 do mesmo algoritmo. Sendo $C_1, \ldots, C_n$ os conjuntos construídos no ponto 3. do referido algoritmo, $a \in I$ e $1 \leq i, j \leq n$

1. $C_1, \ldots, C_n$ constituem uma partição de Q_1 (i.e, são conjuntos não vazios, disjuntos dois a dois e $Q_1 = \bigcup_{1 \leq k \leq n} C_k$;
2. se $p, q \in C_i$ então p e q são estados equivalentes;
3. se $i \neq j$, $p \in C_i$ e $q \in C_j$ então p e q são estados distinguíveis;
4. se $\delta_1(q, a) \in C_j$ para algum $q \in C_i$ então $\delta_1(q, a) \in C_j$ para qualquer $q \in C_i$;
5. se $\delta_1(q, a)$ não está definido para algum $q \in C_i$ então $\delta_1(q, a)$ não está definido para qualquer $q \in C_i$;
6. se $C_i \cap F_1 \neq \emptyset$ então $C_i \subseteq F_1$.

Prova: Esta prova não foi apresentada nas aulas teóricas
1. Para cada $1 \leq i \leq n$, C_i não é vazio porque cada estado é sempre equivalente a si próprio.

O conjunto C_i é o conjunto de *todos* os estados que são equivalentes a um dado estado r, o mesmo acontecendo a C_j face a um dado estado s. Se $p \in C_i \cap C_j$, então p e r são estados equivalentes e p e s são estados equivalentes pelo que r e s são estados equivalentes. Facilmente se conclui então que todos os estados equivalentes a r também são equivalentes a s e portanto $C_i \subseteq C_j$. De igual modo, todos os estados equivalentes a s são também equivalentes a r e portanto $C_j \subseteq C_i$. Conclui-se assim que $C_i = C_j$. Deste modo, se $i \neq j$, então $C_i \cap C_j = \emptyset$.

É trivial concluir que $Q_1 = \bigcup_{1 \leq k \leq n} C_k$ pois cada um dos conjuntos C_i é um subconjunto de Q_1 e cada $q \in Q_1$ pertence a algum C_i.

2. O conjunto C_i é constituído por estados que são equivalentes a um dado estado r pelo que se $p, q \in C_i$ então são ambos equivalentes a r e portanto equivalentes entre si.

3. O conjunto C_i é o conjunto de *todos* os estados que são equivalentes a um dado estado r. Se $p \in C_i$ então p é equivalente a r. Se $q \in C_j$, então, como se viu em 1., $q \notin C_i$ pelo que q não é equivalente a r e, consequentemente, não é equivalente a p. Deste modo, p e q são estados distinguíveis.

4. Suponha-se, por absurdo, que $p, q \in C_i$, $\delta_1(q, a) = q' \in C_j$, $\delta_1(p, a) = p' \in C_{j'}$ e $C_j \neq C_{j'}$. Por 3., p' e q' são distinguíveis e portanto existe $x \in I^*$ tal que $\delta_1^*(q', x) \in F$ e $\delta_1^*(p', x) \notin F$ (ou vice-versa). Então, $\delta_1^*(q, ax) \in F$. Por outro lado, $\delta_1^*(p', x) = p'' \notin F$ ou $\delta_1^*(p', x)$ não está definido. Em ambos os casos, $\delta_1^*(p, ax) \notin F$ pelo que p e q são distinguíveis. Chega-se assim a uma contradição, pois, por 2., p e q são equivalentes.

5. Suponha-se, por absurdo, que $p, q \in C_i$, $\delta_1(q, a)$ não está definido e $\delta_1(p, a) = p'$. Como todos os estados de D_1 são produtivos (ver prova da proposição seguinte), existe $x \in I^*$ tal que $\delta_1^*(p', x) \in F$ e portanto $\delta_1^*(p, ax) \in F$. Tem-se necessariamente que $\delta_1^*(q, ax) \notin F$ pelo que p e q são distinguíveis. Chega-se assim a uma contradição, pois, por 2., p e q são equivalentes.

6. Suponha-se que $p, q \in C_i$, $p \in F_1$ e $q \notin F_1$. Então a sequência ϵ distingue p e q, pelo que estes estados são distinguíveis. Chega-se assim a uma contradição, pois, por 2., p e q são equivalentes.

Proposição: Seja $D = (Q, I, \delta, q_0, F)$ um autómato finito determinista e seja $D' = (Q', I, \delta', q_0', F')$ o autómato considerado no ponto 4 do algoritmo de minimização de autómatos anteriormente apresentado. Tem-se que

1. D' está bem definido;
2. D' só tem estados produtivos e relevantes;
3. são distinguíveis quaisquer dois estados distintos de D'.

Prova: Esta prova não foi apresentada nas aulas teóricas

O autómato D' construído no ponto 4 é obtido a partir de um autómato $D_1 = (Q_1, I, \delta_1, q_0^1, F_1)$ que é o próprio D se $In_D = \emptyset$ ou é obtido no ponto 1.2.2 do algoritmo. Se $In_D = \emptyset$, trivialmente, D_1 está bem definido. Se $In_D \neq \emptyset$ e $q_0 \notin In_D$: (i) $q_0 \in Q_1$, pelo que $Q_1 \neq \emptyset$ e (ii) q_0 é produtivo pelo que existe um estado final relevante (e consequentemente não inútil) e, assim, $F_1 \neq \emptyset$. Deste modo, assegura-se que D_1 está bem definido. Tendo em conta a definição de D_1, pela primeira proposição deste Apêndice, D_1 não tem estados inúteis.

1. Por 1. da proposição anterior, $C_1, \ldots, C_n$ é uma partição de Q_1 e portanto são verdadeiras as asserções seguintes.

Sendo $a \in I$ e $q \in C_i$, para algum $1 \leq i \leq n$, existe $1 \leq j \leq n$ tal que $\delta_1(q, a) \in C_j$ se $\delta_1(q, a)$ está definido. Usando também 4 da proposição anterior, se $\delta_1(q, a) \in C_j$ então $\delta_1(q', a) \in C_j$ para cada $q' \in C_i$ e, usando 5, se $\delta_1(q,a)$ não está definido então $\delta_1(q',a)$ também não está definido para cada $q' \in C_i$. Deste modo a função δ' fica definida sem ambiguidade.

Tem-se que $q_0 \in C_i$ para um e um só $1 \leq i \leq n$ e portanto q'_0 está definido e sem ambiguidade.

Existe $1 \leq i \leq n$ tal que $C_i \cap F_1 \neq \emptyset$ e portanto $F' \neq \emptyset$.

2. Dado $1 \leq j \leq n$, como, por 1. da proposição anterior, $C_j \neq \emptyset$, existe $p \in C_j \subseteq Q_1$. Como foi referido acima, todos os estados de D_1 são produtivos e, portanto, existe uma sequência $w = a_1 \ldots a_m \in I^*$ $(m \geq 0)$ tal que $\delta_1^*(p,w) \in F_1$. Considerando a derivação $p_0 \ldots p_m$ de w a partir de p em D_1, tem-se que $p_0 = p$, $p_m \in F_1$ e $\delta_1(p_j, a_{j+1}) = p_{j+1}$ para cada $0 \leq j < m$. Considere-se a sequência $P_0 \ldots P_m$ de estados de D' onde, para cada $0 \leq j \leq m$, P_j é o estado de Q' que contém p_j (por 1. da proposição anterior, existe sempre um e um só estado de Q' nestas condições). Tendo em conta a definição de D', tem-se que $P_0 = C_j$, $P_m \in F'$ e $\delta'(P_j, a_{j+1}) = P_{j+1}$ para cada $0 \leq j < m$. Facilmente se conclui então que ${\delta'}^*(C_j, w) \in F'$ pelo que C_j é um estado produtivo.

Por um raciocínio semelhante prova-se que D' só tem estados relevantes.

3. Suponha-se, por absurdo, que existem $1 \leq i, j \leq n$ tal que $i \neq j$ e C_i e C_j são estados equivalentes. Por 1. da proposição anterior, existem $p \in C_i \subseteq Q_1$ e $q \in C_j \subseteq Q_1$. Por 3. da proposição anterior, p e q são distinguíveis. Logo, existe uma sequência $w = a_1 \ldots a_m \in I^*$ $(m \geq 0)$ tal que $\delta_1^*(p,w) \in F_1$ e $\delta_1^*(q,w) \notin F_1$ (ou vice-versa).

Considerando a derivação $p_0 \ldots p_m$ de w a partir de p em D_1 e raciocinando como em 2., conclui-se que ${\delta'}^*(C_i, w) \in F'$.

Como se assume que C_i e C_j são estados equivalentes, ${\delta'}^*(C_j, w) \in F'$. Considere-se a derivação $P_0 \ldots P_m$ de w a partir de C_j em D'. Tem-se que $P_0 = C_j$ (e portanto $q \in P_0$), $P_m \in F'$ (e portanto $P_m \cap F_1 \neq \emptyset$) e $\delta'(P_j, a_{j+1}) = P_{j+1}$ para cada $0 \leq j < m$. Tendo em conta a definição de δ', se $\delta'(P_j, a_{j+1}) = P_{j+1}$ é porque existe $p' \in P_j$ tal que $\delta_1(p', a_{j+1}) \in P_{j+1}$ (e, por 4. da proposição anterior, $\delta_1(p'', a_{j+1}) \in P_{j+1}$ para cada $p'' \in P_j$). Considere-se então sequência $p_0 \ldots p_m$ de estados de D_1 onde $p_0 = q (\in P_0)$ e para cada $0 \leq j < m$, p_{j+1} é o elemento de P_{j+1} que é igual a $\delta_1(p_j, a_{j+1})$. Por 6. da proposição anterior, dado que $P_m \cap F_1 \neq \emptyset$, então $P_m \subseteq F_1$ e portanto $p_m \in F_1$. Facilmente se conclui então que $\delta_1^*(q,w) \in F_1$. Chega-se assim a uma contradição e, consequentemente, a asserção 3. é verdadeira.

Proposição: Seja $D = (Q, I, \delta, q_0, F)$ um autómato finito determinista. O autómato D' construído no contexto do algoritmo de minimização de autómatos anterior é tal que

- $L_{D'} = L_D$
- se $D'' = (Q'', I, \delta'', q''_0, F'')$ é um outro autómato finito determinista tal que $L_{D''} = L_D$ então $\#Q' \leq \#Q''$.

Prova (esboço): Esta prova não foi apresentada nas aulas teóricas

(A) Suponha-se que $In_D \neq \emptyset$ e $q_0 \in In_D$. Assim, q_0 não é produtivo e $L_D = \emptyset$. Tendo em conta a definição de autómato finito determinista, é imediato concluir que o autómato D' apresentado no ponto 1.2.1 do algoritmo verifica as asserções 1. e 2. acima.

(B) Suponha-se que $In_D \neq \emptyset$ e $q_0 \notin In_D$ e seja D_1 o autómato apresentado no ponto 1.2.2 do algoritmo. É simples mostrar que $L_D = L_{D_1}$. Dado $w = a_1 \ldots a_m \in L_D$ ($m \geq 0$), considere-se a demonstração $p_0 \ldots p_m$ de w em D onde, como se sabe, $\delta(p_j, a_{j+1}) = p_{j+1}$ para cada $0 \leq j < m$. É fácil concluir que nesta demonstração só ocorrem estados de D não pertencentes a In_D (e portanto estados de D_1). Tendo em conta a definição de δ_1, tem-se também que $\delta_1(p_j, a_{j+1}) = p_{j+1}$ para cada $0 \leq j < m$, o que permite facilmente provar que $w \in L_{D_1}$. Considerando agora $w \in L_{D_1}$ e raciocinando de modo semelhante, concluir-se-ia que $w \in L_D$.

Se $In_D = \emptyset$, tendo em conta o ponto 1.1 do algoritmo, é trivial a igualdade $L_D = L_{D_1}$.

Deste parágrafo conclui-se então que o autómato D_1 ao qual vai ser aplicado o algoritmo de procura exaustiva de pares distinguíveis verifica $L_D = L_{D_1}$.

(C) Prova-se agora que $L_{D'} = L_D$ assumindo que D' é o autómato construído no ponto 4 do algoritmo. Por (B), basta mostrar que $L_{D'} = L_{D_1}$.

Seja então $w = a_1 \ldots a_m \in L_{D_1}$ ($m \geq 0$) e considere-se a demonstração $p_0 \ldots p_m$ de w em D_1. Tem-se que $p_0 = q_0$, $p_m \in F_1$ e $\delta_1(p_j, a_{j+1}) = p_{j+1}$ para cada $0 \leq j < m$. Raciocinando como na prova de 2. da proposição anterior, conclui-se que $\delta'^*(q'_0, w) \in F'$ pelo que $w \in L_{D'}$.

Seja agora $w = a_1 \ldots a_m \in L_{D'}$ ($m \geq 0$) e considere-se a demonstração $P_0 \ldots P_m$ de w em D'. Tem-se que $P_0 = q'_0$ (e portanto $q_0 \in P_0$), $P_m \in F'$ (e portanto $P_m \cap F_1 \neq \emptyset$) e $\delta'(P_j, a_{j+1}) = P_{j+1}$ para cada $0 \leq j < m$. Raciocinando como na prova de 3. da proposição anterior, conclui-se então que $\delta_1^*(q_0, w) \in F_1$ pelo que $w \in L_{D_1}$.

(D) O objectivo é mostrar agora que o autómato finito determista D'' nas condições do enunciado verifica $\#Q' \leq \#Q''$ assumindo que D' é o autómato construído no ponto 4 do algoritmo.

Pode assumir-se, sem perda de generalidade, que $Q' \cap Q'' = \emptyset$. Pode também assumir-se que D'' só tem estados produtivos (caso tenha, existe um autómato finito determista com menos estados que reconhece exactamente $L_{D''}$ e pode passar-se a considerar esse novo autómato). Por 2. da proposição anterior, D' só tem estados produtivos e relevantes. Considere-se o autómato $D_U = (Q_U, I, \delta_U, q_0^U, F_U)$ que resulta da "união" de D' e D'' e é obtido como descrito no algoritmo de teste à igualdade das linguagens reconhecidas por dois autómatos.

(i) Os estados q'_0 e q''_0 são estados equivalentes de D_U (pois $L_{D'} = L_{D''}$).

(ii) Sejam $p \in Q' \subseteq Q_U$ e $q \in Q'' \subseteq Q_U$ estados equivalentes.

Suponha-se que existe $a \in I$ tal que $\delta_U(p, a) = p'$ e $\delta_U(q, a)$ não definido. Tem-se que $\delta'(p, a) = p \in Q'$. Como, por 2. da proposição anterior, p' é estado produtivo de D', conclui-se facilmente que p' é estado produtivo de D_U e portanto existe $w \in I^*$ tal que $\delta_U^*(p', w) \in F_U$, pelo que $\delta_U^*(p, aw) \in F_U$. Mas $\delta_U^*(q, aw) \notin F_U$ e portanto aw distingue p e q. Chega-se assim a um absurdo.

Se $\delta_U(p, a)$ não definido e $\delta_U(q, a) = q'$, como D'' também só tem estados produtivos, chegar-se-ia de novo a um absurdo.

Conclui-se assim que sendo $p \in Q' \subseteq Q_U$ e $q \in Q'' \subseteq Q_U$ estados equivalentes, dado $a \in I$, $\delta_U(p, a)$ está definido sse $\delta_U(q, a)$ está definido.

(iii) Mostra-se agora que cada estado $p \in Q' \subseteq Q_U$ é equivalente a pelo menos um estado $q \in Q'' \subseteq Q_U$.

Seja $p \in Q' \subseteq Q_U$. Por 2. da proposição anterior, p é estado relevante de D' e portanto existe $w = a_1 \ldots a_m \in I^*$ $(m \geq 0)$ tal que $\delta'^*(q'_0, w) = \delta^*_U(q'_0, w) = p$. Considere-se a derivação $p_0 \ldots p_m$ de w a partir de q'_0 em D_1. Tem-se que $p_0 = q'_0$, $p_m = p$ e $\delta'(p_j, a_{j+1}) = \delta_U(p_j, a_{j+1}) = p_{j+1}$ para cada $0 \leq j < m$. Por (i), q'_0 e q''_0 são equivalentes e portanto, como $\delta_U(q'_0, a_1) = p_1$, por (ii), $\delta_U(q''_0, a_1) = q_1 \in Q''$. Se p_1 e q_1 fossem distinguíveis então, q'_0 e q''_0 também o seriam. Dado que são equivalentes então também p_1 e q_1 são equivalentes. Raciocinando de modo análogo para cada $1 \leq j < m$ (ou, mais rigorosamente, fazendo uma prova por indução), concluiu-se que $p_m (= p)$ é equivalente a um certo estado $q \in Q''$.

(iv) Por (iii) é possível definir uma função $f : Q' \to Q''$ tal que $f(p)$ é um estado de Q'' equivalente a p. Se $\#Q'' < \#Q'$, têm de existir dois estados distintos p e q de D' tais que $f(p) = f(q) = p' \in Q''$. Isto siginifica que p é equivalente a p' e q é equivalente a p' em D_U. Consequentemente, p e q são equivalentes em D_U. Então, para cada $w \in I^*$, $\delta^*_U(p, w) \in F_U$ sse $\delta^*_U(q, w) \in F_U$, ou seja, tendo em conta a definição de D_U, $\delta'^*(p, w) \in F'$ sse $\delta'^*(q, w) \in F'$. Deste modo p e q são estados distintos de D' e são equivalentes o que contradiz 3. da proposição anterior. Conclui-se então que $\#Q' \leq \#Q''$ como se pretendia.

(E) As alíneas (C) e (D) acima provam as asserções enunciadas nesta proposição assumindo que D' é o autómato construído no ponto 4 do algoritmo. Resta agora considerar o caso relativo ao ponto 2.1 do algoritmo em que D' é D_1. Neste caso, por (B), a asserção $L_D = L_{D'}$ é trivial. Quanto à desigualdade $\#Q' \leq \#Q''$, basta ter em conta que se em D_1 não existem estados equivalentes então, se forem aplicados os pontos 3 e 4 a este autómato, cada um dos conjuntos $C_1, \ldots, C_n$ é um conjunto singular e $n = \#Q_1 = \#Q'$. Raciocinando como em (D) concluir-se-ia a desigualdade pretendida.