Instituto Superior Técnico – LEIC, Alameda

Teoria da Computação

3 Maio 2022 MAP30–1 Repescagem Duração: 30m

Nome: _____ **Número:** _____

Considere o alfabeto $\Sigma = \{a, b\}$ e as linguagens $L_1, L_2 \subseteq \Sigma^*$ tais que:

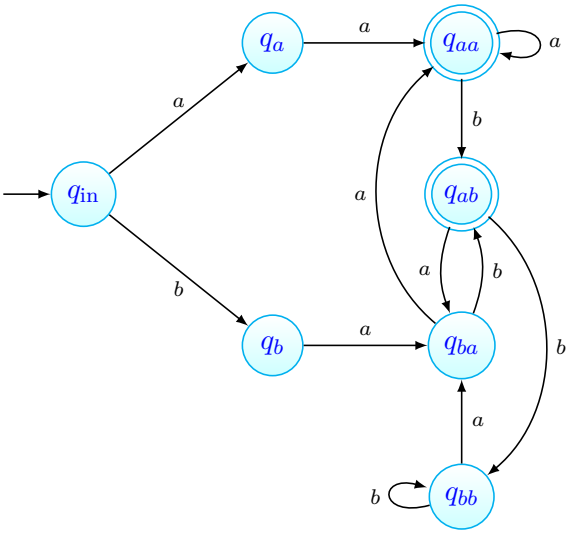
L_1: conjunto das palavras em que o segundo símbolo é a e o penúltimo também

L_2: conjunto das palavras da forma ww^R com $w \in \Sigma^*$

Por exemplo, a palavra baa pertence a L_1, pois o segundo e o penúltimo símbolo (que neste caso são o mesmo) é a, mas não pertence a L_2 (até porque tem comprimento ímpar). A palavra $aabbaa$ pertence a L_1, e também pertence a L_2 já que $aab^R = baa$.

a) (1.5 valores) Mostre (construindo um AFD) que L_1 é uma linguagem regular.

Resolução: Basta considerar o AFD que aceita precisamente as palavras de L_1.

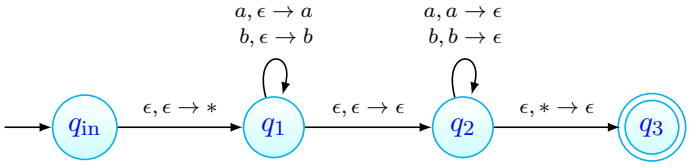


b) (1.5 valores) Mostre que L_2 não é uma linguagem regular.

Resolução: Se L_2 fosse regular, pelo lema da bombagem, existiria $k \in \mathbb{N}$ tal que se $w \in L_2$ e $|w| \geq k$ então $w = w_1w_2w_3$ com $|w_1w_2| \leq k$, $w_2 \neq \epsilon$ e $w_1w_2^iw_3 \in L_2$ para qualquer $i \in \mathbb{N}_0$. Dado k, considere-se $w = a^kbba^k \in L_2$. Então ter-se-ia $w_1 = a^j$, $w_2 = a^l$ e $w_3 = a^{k-(j+l)}bba^k$ com $l \neq 0$. Logo, por exemplo com $i = 0$, tem-se $w_1w_2^0w_3 = w_1w_2 = a^{k-l}bba^k \notin L_2$ pois $a^{k-l} \neq a^k$ já que $l \neq 0$.

c) (1.0 valores) Mostre (construindo um AP) que L_2 é independente do contexto.

Resolução: Basta considerar o AP que aceita precisamente as palavras de L_2.



Instituto Superior Técnico – LEIC, Alameda

Teoria da Computação

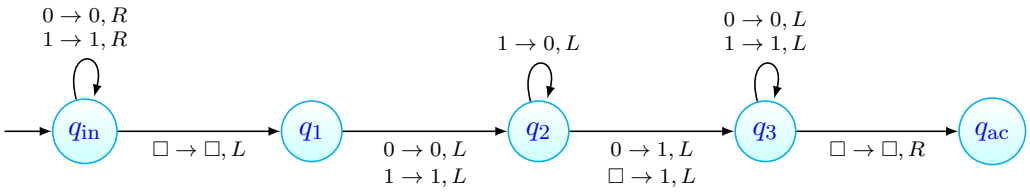
3 Maio 2022 MAP30–2 Repescagem Duração: 30m

Nome: _____ **Número:** _____

a) (2.0 valores) Mostre (construindo uma máquina de Turing determinista, possivelmente bidireccional, multifita e com movimentos-S) que é computável a função $f : \mathbb{N}_0 \to \mathbb{N}_0$ tal que $f(n) = n + 2$, em que *input* e *output* adoptam a representação binária.

Nomeadamente, para o *input* 110 deverá obter-se o *output* 1000, pois $6 + 2 = 8$.

Resolução: Basta considerar a MT seguinte, que calcula a função.



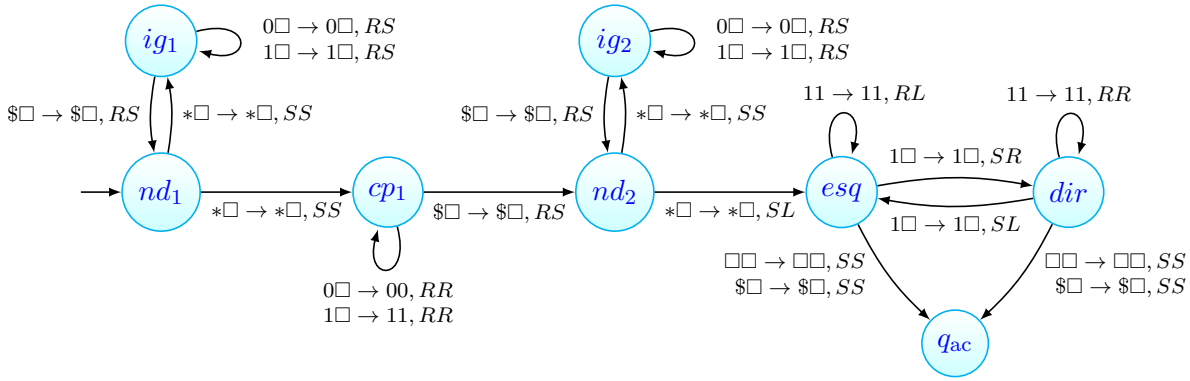
A máquina avança pelo *input* (q_{in}) e posiciona a cabeça de leitura/escrita no bit mais à direita (q_1). Como 2 é 10 em binário, esse bit é mantido (q_2) e a soma é executada (q_3), posicionando-se no final a cabeça de leitura/escrita no bit mais à esquerda (q_{ac}).

b) (2.0 valores) Mostre (construindo uma máquina de Turing possivelmente bidireccional, multifita e com movimentos-S, e tirando partindo do não-determinismo) que é reconhecível a linguagem $L \subseteq \{1, \$\}^*$ constituída por todas as palavras da forma

$$1^{n_1}\$1^{n_2}\$\ldots\$1^{n_k}$$

em que $k \in \mathbb{N}_0$ e existem $i < j$ tais que n_i é divisor de n_j. Note, nomeadamente, que $11\$1111\$111 \in L$ pois 2 é divisor de 4, mas $1111\$11\$111 \notin L$ pois 2 ocorre depois de 4, bem como $11111\$11\$111 \notin L$ já que 2, 3 e 5 são números primos distintos.

Resolução: Basta considerar a MT não determinista de duas fitas seguinte, que reconhece a linguagem L.



A máquina usa não-determinismo (nd_1) para ignorar (ig_1) elementos do *input* até escolher copiar o potencial divisor n_i para a fita 2 (cp_1). Depois, volta a usar não-determinismo (nd_2) para ignorar (ig_2) mais elementos do *input* até escolher o potencial dividendo n_j. Então, é testada a divisibilidade (esq, dir) e aceita-se o *input* caso n_i seja divisor de n_j.

Instituto Superior Técnico – LEIC, Alameda

Teoria da Computação

3 Maio 2022 MAP30–3 Repescagem Duração: 30m

Nome: _____ **Número:** _____

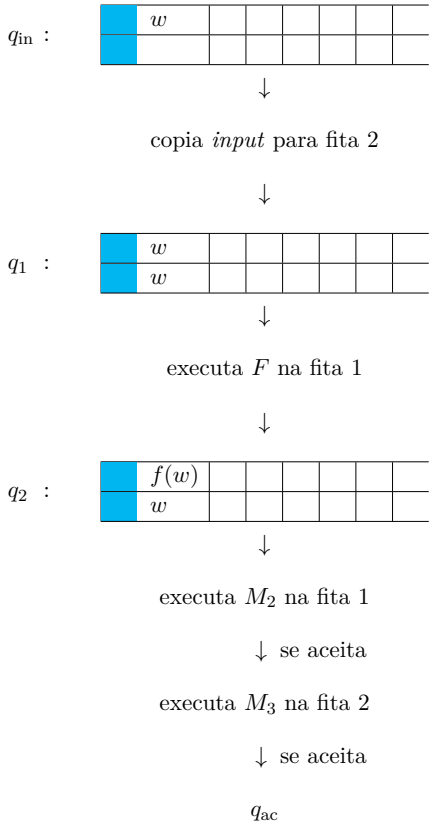
Considere um alfabeto Σ.

a) (2.0 valores) Sejam $L_1, L_2, L_3 \subseteq \Sigma^*$ linguagens tais que $L_1 \leq L_2$ e $L_2, \overline{L_3}$ são reconhecíveis. Mostre, justificando, que também é reconhecível a linguagem $L_1 \setminus L_3$.

Resolução: Como $L_1 \leq L_2$, seja F a máquina que calcula a função total $f : \Sigma^* \to \Sigma^*$ tal que $x \in L_1$ se e só se $f(x) \in L_2$.

Sejam ainda M_2 a máquina que reconhece L_2 e M_3 a máquina que reconhece $\overline{L_3}$.

Considere-se a máquina M, com duas fitas, descrita por:



Tem-se que:

- Se $w \in L_1 \setminus L_3 = L_1 \cap \overline{L_3}$ então $w \in L_1$ pelo que $f(w) \in L_2$ e M_2 aceita $f(w)$, e também que $w \in \overline{L_3}$ e M_3 aceita w, pelo que M aceita w, já que a cópia do *input* e o cálculo de f terminam sempre.
- Se M aceita w então M_2 aceita $f(w)$ pelo que $f(w) \in L_2$ e portanto $w \in L_1$, e também M_3 aceita w pelo que $w \in \overline{L_3}$, tendo-se portanto que $w \in L_1 \cap \overline{L_3} = L_1 \setminus L_3$.

Conclui-se que $L_{\text{ac}}(M) = L_1 \setminus L_3$ e portanto a linguagem é reconhecível.

b) (2.0 valores) Seja $A \subseteq \Sigma^*$ uma linguagem e $w \in \Sigma^* \setminus A$ uma palavra. Mostre, usando o teorema de Rice, que é indecidível a linguagem

$$L = \{M \in \mathcal{M}^\Sigma : A \subseteq L_{\text{ac}}(M) \text{ e } w \notin L_{\text{ac}}(M)\}.$$

Resolução: Pelo teorema de Rice, precisamos de verificar quatro condições.

1. $L \subseteq \mathcal{M}^\Sigma$, por definição.
2. $L \neq \mathcal{M}^\Sigma$ pois a seguinte máquina $M_w \notin L$:

$$M_w : \begin{array}{l} \text{INPUT } x \\ \text{SE } x = w \text{ ENTÃO ACEITA SENÃO REJEITA} \end{array}$$

   já que $L_{\text{ac}}(M_w) = \{w\}$ e portanto $w \in L_{\text{ac}}(M_w)$.
3. $L \neq \emptyset$ pois a seguinte máquina $M_w^- \in L$:

$$M_w^- : \begin{array}{l} \text{INPUT } x \\ \text{SE } x = w \text{ ENTÃO REJEITA SENÃO ACEITA} \end{array}$$

   já que $L_{\text{ac}}(M_w^-) = \Sigma^* \setminus \{w\}$ e como sabemos que $w \notin A$, tem-se $A \subseteq L_{\text{ac}}(M_w^-)$ e também $w \notin L_{\text{ac}}(M_w^-)$.
4. Se $M_1 \equiv M_2$ e $M_1 \in L$ então tem-se que $A \subseteq L_{\text{ac}}(M_1)$ e também $w \notin L_{\text{ac}}(M_1)$, mas nesse caso $L_{\text{ac}}(M_1) = L_{\text{ac}}(M_2)$, e portanto tem-se que $A \subseteq L_{\text{ac}}(M_2)$ e também $w \notin L_{\text{ac}}(M_2)$, ou seja, $M_2 \in L$.

Sendo satisfeitas todas as condições conclui-se que a linguagem L é indecidível.

Instituto Superior Técnico – LEIC, Alameda

Teoria da Computação

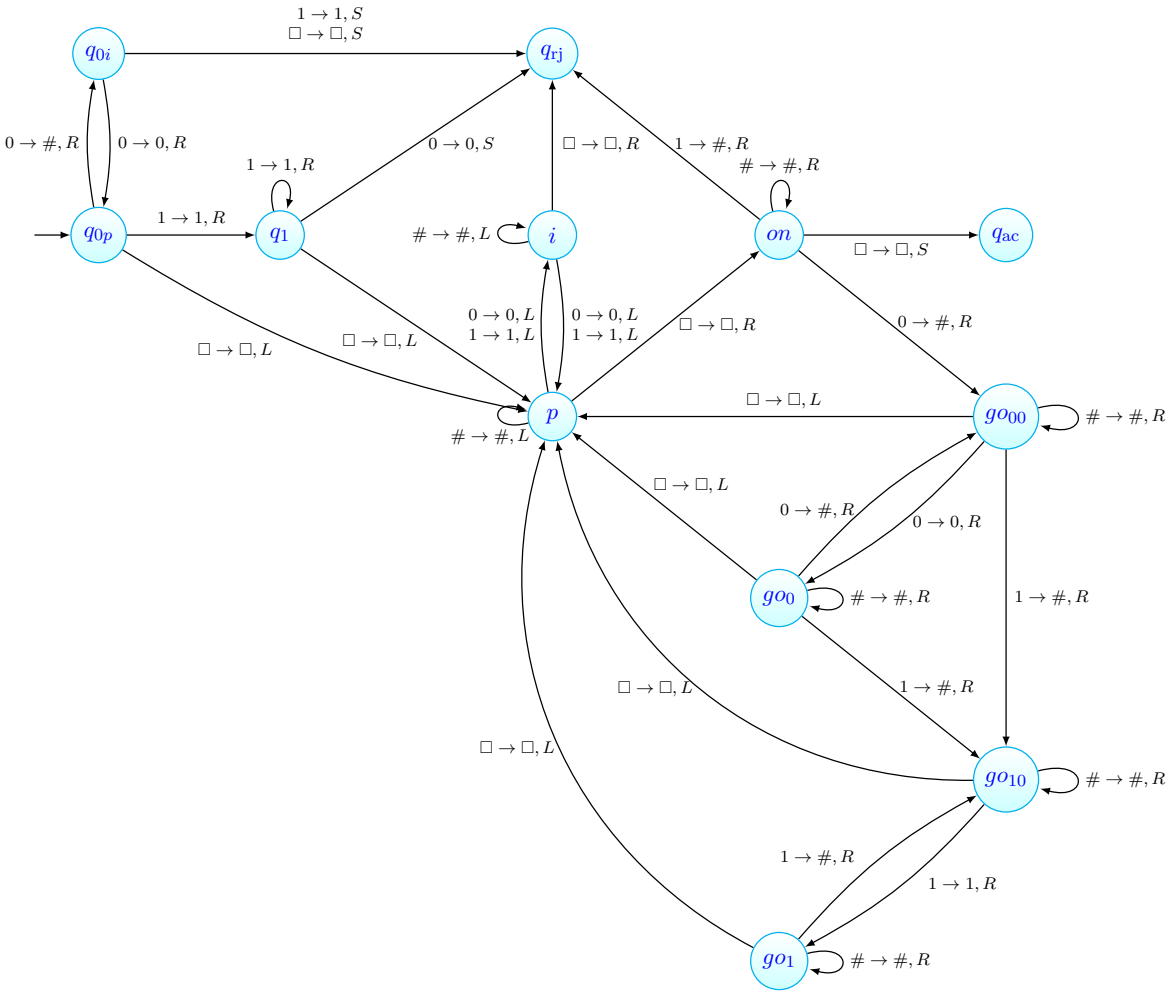
3 Maio 2022 MAP30–4 Repescagem Duração: 30m

Nome: _____ **Número:** _____

Considere a linguagem $L = \{0^{2n}1^n : n \in \mathbb{N}_0\}$.

a) (3.0 valores) Mostre que $L \in \mathbf{SPACE}(n) \cap \mathbf{TIME}(n.\log(n))$.

Resolução: Considere-se a MT M seguinte.



A máquina avança pelo *input*, da esquerda para a direita, garantindo que tem um bloco inicial de $0s$ de tamanho par e cancelando com # metade dos 0s, nos estados q_{0p}, q_{0i}, e depois um bloco final de 1s no estado q_1; de seguida repetidamente, a máquina volta a percorrer a fita, agora da direita para a esquerda, garantindo que o número de 0s e 1s remanescente é par, nos estados p e i, e depois novamente da esquerda para a direita, cancelando com # metade dos 0s, nos estados on, go_{00}, go_0, e também metade dos 1s, nos estados go_{10}, go_1. O *input* é aceite caso todos os seus símbolos sejam cancelados com #, e rejeitado caso alguma das condições intermédias não se verifique.

Naturalmente, a segunda fase repete-se um número logarítmico de vezes e portanto tem-se

$$time_M(n) = \mathcal{O}(n) + (\log_2(n) + 1).\mathcal{O}(n) = \mathcal{O}(n.\log(n))$$
$$space_M(n) = n + 2 = \mathcal{O}(n)$$

e podemos concluir que $L \in \mathbf{SPACE}(n) \cap \mathbf{TIME}(n.\log(n))$.

b) (1.0 valor) Seja $S \subseteq \{0,1\}^*$ uma linguagem tal que $\overline{S} \leq_P L$. Use o resultado da alínea anterior para mostrar que $S \in \mathbf{P}$.

Resolução: Da alínea anterior sabemos que $L \in \mathbf{TIME}(n.\log(n)) \subseteq \mathbf{TIME}(n^2) \subseteq \mathbf{P}$. Seja M a respectiva máquina classificadora.

Como $\overline{S} \leq_P L$, seja F a máquina de tempo polinomial $time_F(n) = \mathcal{O}(n^a)$ que calcula a função total $f : \{0,1\}^* \to \{0,1\}^*$ tal que $x \in \overline{S}$ se e só se $f(x) \in L$.

Considere-se a seguinte máquina de Turing T descrita abaixo.

$$T: \begin{array}{l} \text{INPUT } w \\ \text{EXECUTA } F \text{ PARA CALCULAR } f(w) \\ \text{EXECUTA } M \text{ SOBRE } f(w) \\ \text{SE } M \text{ ACEITA ENTÃO REJEITA SENÃO ACEITA} \end{array}$$

Comecemos por verificar que T decide a linguagem S.

- T é classificadora pois F termina sempre e M é classificadora.
- T aceita w sse M rejeita $f(w)$ sse $f(w) \notin L$ sse $w \notin \overline{S}$ sse $w \in S$.

Para mostrar que $S \in \mathbf{P}$ basta agora verificar que T é de tempo polinomial. Na verdade, tem-se:

$$time_T(n) \leq time_F(n) + time_M(n + time_F(n)) = \mathcal{O}(n^a) + \mathcal{O}((n + n^a)^2) = \mathcal{O}(n^{2a}).$$

Instituto Superior Técnico – LEIC, Alameda

Teoria da Computação

3 Maio 2022 MAP30–5 Repescagem Duração: 30m

Nome: _____ **Número:** _____

a) (2.5 valores) Seja Σ um alfabeto, $\$ \notin \Sigma$, e considere linguagens $A, B, C, L \subseteq \Sigma^*$ tais que:

- $A \leq_P L$,
- $B \leq_P A$,
- $C \leq_P (A \setminus B) \cup (B \setminus A)$.

Mostre, justificando, que $C \leq_P \{w_1\$w_2 : w_1, w_2 \in \Sigma^* \text{ com } w_1 \in L \text{ se e só se } w_2 \notin L\}$.

Resolução: Seja $R = \{w_1\$w_2 : w_1, w_2 \in \Sigma^* \text{ com } w_1 \in L \text{ se e só se } w_2 \notin L\}$.

Como $A \leq_P L$, seja F a máquina de tempo polinomial $time_F(n) = \mathcal{O}(n^a)$ que calcula a função total $f : \Sigma^* \to \Sigma^*$ tal que $w \in A$ se e só se $f(w) \in L$.

Como $B \leq_P A$, seja G a máquina de tempo polinomial $time_G(n) = \mathcal{O}(n^b)$ que calcula a função total $g : \Sigma^* \to \Sigma^*$ tal que $w \in B$ se e só se $g(w) \in A$.

Como $C \leq_P (A \setminus B) \cup (B \setminus A)$, seja H a máquina de tempo polinomial $time_H(n) = \mathcal{O}(n^c)$ que calcula a função total $h : \Sigma^* \to \Sigma^*$ tal que $w \in C$ se e só se $h(w) \in (A \setminus B) \cup (B \setminus A)$.

Para mostrar que $C \leq_P R$ considere-se a função $k : \Sigma^* \to (\Sigma \cup \{\$\})^*$ definida por

$$k(w) = f(h(w))\$f(g(h(w))).$$

A função k é total pois f, g, h são totais, e é facilmente computável pois, tirando partindo das máquinas F, G, H podemos considerar a máquina K, de duas fitas, descrita a seguir.

K :
INPUT w
EXECUTA H NA FITA 1 PARA CALCULAR $h(w)$
COPIA $h(w)$ PARA A FITA 2
EXECUTA F NA FITA 1 PARA CALCULAR $f(h(w))$
EXECUTA G NA FITA 2 PARA CALCULAR $g(h(w))$
EXECUTA F NA FITA 2 PARA CALCULAR $f(g(h(w)))$
OUTPUT $f(h(w))\$f(g(h(w)))$

A máquina K é de tempo polinomial pois

$$\begin{aligned}
&time_K(n)\\
&\leq\\
&time_H(n)+\\
&\mathcal{O}(n + time_H(n))+\\
&time_F(n + time_H(n))+\\
&time_G(n + time_H(n))+\\
&time_F(n + time_H(n) + time_G(n + time_H(n)))\\
&\leq\\
&\mathcal{O}(n^c + (n + n^c) + (n + n^c)^a + (n + n^c)^b + (n + n^c + (n + n^c)^b)^a)\\
&=\\
&\mathcal{O}(n^{abc})
\end{aligned}$$

que é um polinómio, e continuará a ser mesmo com uma desaceleração quadrática, por K ter duas fitas.

Finalmente, note-se a seguinte sequência de equivalências:

$$w \in C$$
$$h(w) \in (A \setminus B) \cup (B \setminus A)$$
$$h(w) \in A \text{ sse } h(w) \notin B$$
$$h(w) \in A \text{ sse } g(h(w)) \notin A$$
$$f(h(w)) \in L \text{ sse } f(g(h(w))) \notin L$$
$$f(h(w))\$f(g(h(w))) \in R$$
$$k(w) \in R$$

b) (1.5 valores) Demonstre que $\mathbf{P} \neq \mathbf{NP}$ se e só se não existe $L \in \mathbf{P}$ tal que L é **NP**-difícil.

Resolução: Equivalentemente, mostramos que $\mathbf{P} = \mathbf{NP}$ se e só se existe $L \in \mathbf{P}$ tal que L é **NP**-difícil.

- Se $\mathbf{P} = \mathbf{NP}$ então tome-se uma qualquer linguagem L que seja **NP**-completa (como **SUBSETSUM** ou **SAT**). Obviamente L é **NP**-difícil e $L \in \mathbf{NP} = \mathbf{P}$.
- Se existe $L \in \mathbf{P}$ tal que L é **NP**-difícil, então tem-se que $A \leq_P L$ qualquer que seja $A \in \mathbf{NP}$. Mas como $L \in \mathbf{P}$, se $A \leq_P L$, temos que também $A \in \mathbf{P}$, qualquer que seja $A \in \mathbf{NP}$. Daqui concluímos que $\mathbf{NP} \subseteq \mathbf{P}$. Como sabemos *a priori* que $\mathbf{P} \subseteq \mathbf{NP}$ (pois as máquinas de Turing deterministas são casos particulares da definição de máquina de Turing não-determinista), obtemos $\mathbf{P} = \mathbf{NP}$.