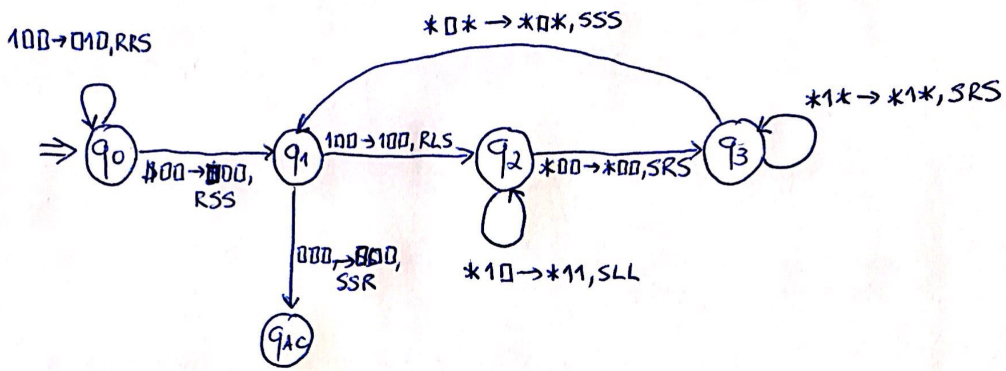
(3 valores)

Construa uma máquina de Turing que dados números n e m calcule o seu produto $n \times m$, assumindo que todos os números estão representados em unário.

Nomeadamente, para o *input* 111\$11 deve obter-se o *output* 111111, pois $3 \times 2 = 6$.

Considere-se a máquina com 3 fitas bidireccionais definida por



onde * representa qualquer símbolo do alfabeto {1,$,□}.

A máquina começa por copiar a primeira string de 1s para a fita 2 (estado q_0).

Depois, a partir do estado q_1, para cada 1 da segunda string de 1s na fita 1 copia (estado q_2) todos os 1s da fita 2 para a fita 3, voltando à posição inicial na fita 2 (estado q_3) e repetindo até percorrer todos os 1s da segunda string.

O output devido é obtido na fita 3.

(3 valores)

Considere a linguagem $L \subseteq \{a, b, \#\}^*$ constituída pelas palavras da forma

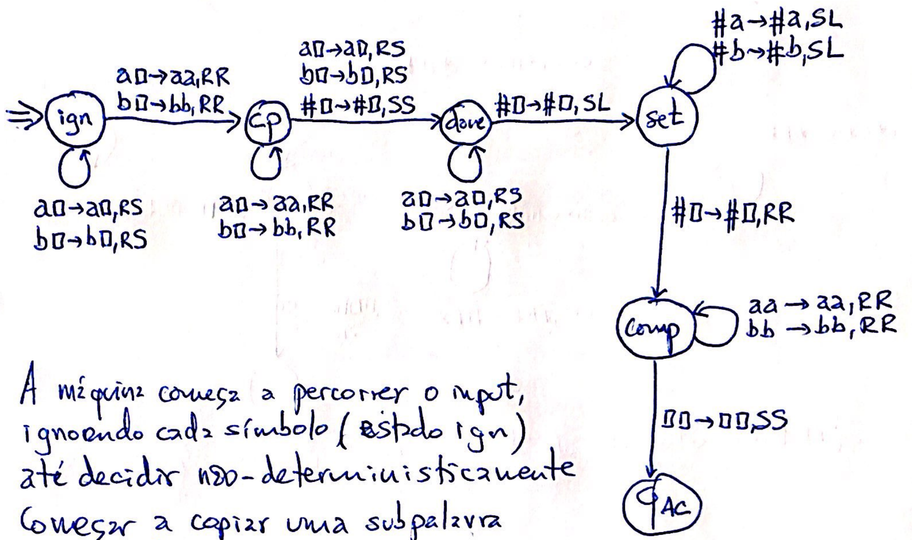
$$w_1 \# w_2$$

em que $w_1, w_2 \in \{a, b\}^*$ e w_2 é uma *subpalavra* não vazia de w_1.

Por exemplo, $abbababa\#aba \in L$ pois aba é subpalavra de $abb\underline{aba}ba$ (como sublinhado), mas $abba\#aa \notin L$ pois aa não é subpalavra de $abba$.

Mostre que $L \in \mathbf{NTIME}(n^2)$.

Considere-se a seguinte máquina de Turing não-determinista com 2 fitas.



A máquina começa a percorrer o input, ignorando cada símbolo (estado ign) até decidir não-deterministicamente começar a copiar uma subpalavra para a fita 2 (estado cp).

Daí, a máquina decide não-deterministicamente parar a cópia, transitando para o estado done onde percorre e ignora todos os símbolos até ao separador #.
Então, no estado set, a palavra escrita na fita 2 é rebobinada e finalmente comparada (estado comp) com a subpalavra pretendida, aceitando-se caso sejam iguais.

Agrupando ign, cp, done na fase 1, set na fase 2, e comp na fase 3, a performance temporal da máquina é a soma das três fases, dada por

$$\mathrm{ntime}_M(n) = O(n) + O(n) + O(n) = O(n).$$

Sendo a máquina equivalente a outra com apenas 1 fita com desaceleração quadrática, conclui-se que

$$L \in \mathrm{NTIME}(n^2).$$

(3 valores)

Considere a linguagem

$$L_1 = \{M \in \mathcal{M}^{\{0,1\}} : \text{ a computação de } M \text{ sobre o } \textit{input } M \text{ é infinita}\}.$$

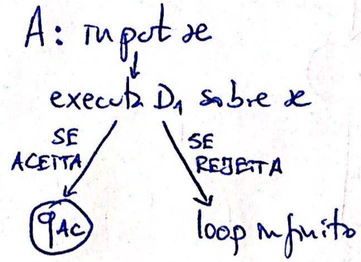
Mostre directamente, por redução ao absurdo, que L_1 é indecidível.

Supõe-se, por absurdo, que

$$L_1 = \{ M \in \mathcal{M}^{\{0,1\}} : \text{comp. de } M \text{ sobre } M \text{ é infinita} \}$$

fosse decidível.

Usando uma máquina D_1 que decidisse L_1 podíamos então construir a seguinte máquina.



Tomando como input o código da própria máquina A tem-se então que

a computação de A sobre A é infinita
se e só se
D_1 rejeita A
se e só se
$A \notin L_1$
se e só se
a computação de A sobre A <u>não</u> é infinita

<u>CONTRADIÇÃO</u>

(3 valores)

Seja Σ um alfabeto, $L_1, L_2 \subseteq \Sigma^*$.

Assuma que $L_1 \in \mathbf{NP}$ e $\overline{L_2} \leq_P L_1$.
Pode garantir que a linguagem $L_1 \setminus L_2$ está em:

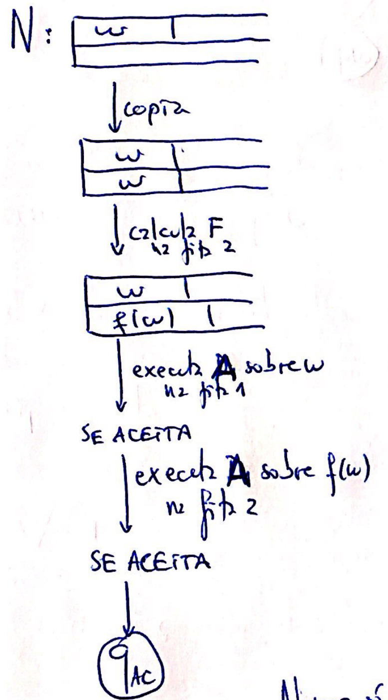
(a) $\mathbf{NP}$?
(b) $\mathbf{PSPACE}$?

Justifique cuidadosamente ambas as respostas.

(a) Sim, $L_1 \setminus L_2 \in NP$ pois $L_1 \setminus L_2 = L_1 \cap \overline{L_2}$,
$L_1 \in NP$, por $\overline{L_2} \leq_p L_1$ também $\overline{L_2} \in NP$,
e NP é fechada para intersecções.

Concretamente, seja A a máquina não-determinista que decide L_1 com $ntime_A(n) = O(n^a)$, e seja F a máquina determinista que calcula f tal que $w \in \overline{L_2}$ sse $f(w) \in L_1$, com $time_F(n) = O(n^b)$.

Considere-se a seguinte máquina não-determinista com 2 fitas.



N é classificadora pois f é total e A é classificadora.

N aceita w sse
A aceita w e A aceita f(w) sse
$w \in L_1$ e $f(w) \in L_1$ sse
$w \in L_1$ e $w \in \overline{L_2}$ sse
$w \in L_1 \setminus L_2$.

Tem-se ainda

$$ntime_N(n) = \overbrace{O(n)}^{\text{cópia}} + \overbrace{O(n^b)}^{F} + \overbrace{O(n^a)}^{A \text{ sobre } w} + \overbrace{O((n+n^b)^a)}^{A \text{ sobre } f(w)}$$
$$= O(n^{ab})$$

Numa só fita, mesmo com desaceleração quadrática tem-se ainda que

$$L_1 \setminus L_2 \in NTIME(n^{2ab}) \subseteq NP.$$

(b) Sim, pois $NP \subseteq PSPACE$.

De facto $NP \subseteq NPSPACE$ pois $nspace_M(n) \leq ntime_M(n)$, já que cada transição numa máquina com 1 fita visita no máximo 1 nova célula de memória, e portanto sendo N' a máquina não determinista com 1 fita equivalente a N tem-se $ntime_{N'}(n) = O(n^{2ab})$ e portanto $nspace_{N'}(n) \leq O(n^{2ab})$ pelo que a linguagem está em NPSPACE.

Pelo Teorema de Savitch sabemos ainda que $NPSPACE \subseteq PSPACE$, ou melhor que existe N'' equivalente a N' com N'' determinista e $space_{N''}(n) \leq O(n^{4ab})$, pelo que a linguagem está em PSPACE.

(1+3 valores)

O teorema de Rice é muito útil para demonstrar a indecidibilidade de linguagens, mas é inconclusivo para linguagens às quais não se possa aplicar. Nesse sentido, fixe um alfabeto Σ e dê exemplos de linguagens $\emptyset \subsetneq L_1, L_2 \subsetneq \mathcal{M}^\Sigma$ às quais o teorema de Rice não se aplique tais que:

(a) L_1 é decidível,
(b) L_2 é indecidível,

Justifique cuidadosamente.

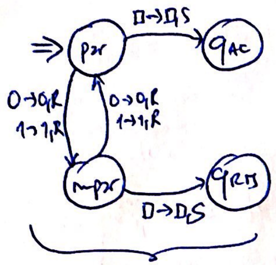
Tome-se $\Sigma = \{0,1\}$.

(a) Considere-se $L_1 = \{M \in \mathcal{M}^{\{0,1\}} : M \text{ tem 4 estados}\}$.

Obviamente, $L_1 \subseteq \mathcal{M}^{\{0,1\}}$.

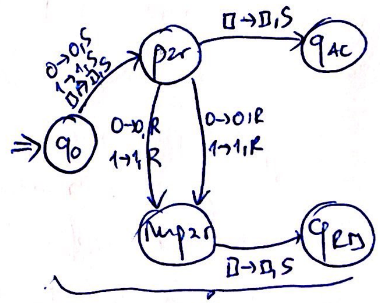
Facilmente se verifica que $\emptyset \subsetneq L_1 \subsetneq \mathcal{M}^{\{0,1\}}$.

Tomem-se as máquinas



A

e



B

Facilmente $A \in L_1$ e $B \notin L_1$.

No entanto o teorema de Rice não se aplica pois $A \equiv B$.

L_1 é decidível, pois o número de estados da máquina é dado explicitamente na sua representação canónica.

(b) Considere-se

$L_2 = \{M \in \mathcal{M}^{\{0,1\}} :$ computação de M sobre M é infinita$\}$

Como na terceira questão desta resolução, onde já vimos que L_2 é indecidível.

Por definição, $L_2 \subseteq \mathcal{M}^{\{0,1\}}$, e facilmente se verifica que $\emptyset \subsetneq L_2 \subsetneq \mathcal{M}^{\{0,1\}}$.

Tomem-se as máquinas

$$\underbrace{M_{inf}: \begin{array}{l}\text{input } x\\ \text{loop infinito}\end{array}} \quad \text{e} \quad \underbrace{M_{ALL}: \begin{array}{c}\text{input } x\\ \downarrow\\ q_{AC}\end{array}}$$

Facilmente, $M_{inf} \in L_2$ e $M_{ALL} \notin L_2$.

No entanto o teorema de Rice não é aplicável.

Note-se que se $A \equiv B$ e $A \in L_2$

então A sobre A é infinita

e dada a equivalência

B sobre A é infinita <u>ou aborta</u>

e isso não nos diz nada sobre

a computação de B sobre B.