"Solução" do exercício

▷ 1 1 1 1 "procurar 1 à direita"

▷ 1 1 1 1 "marcar o 1 como 'copiando'" ⇝ troca por 2

▷ 2 1 1 1 "escrever cópia no final"

⋮

▷ 2 1 1 1 3 "procurar novos 1s à esquerda"

▷ 2 1 1 1 3

⋮

▷ 2 1 1 2 3 3

D211233

⋮

D2122333

⋮

D2122333

⋮

D22223333

⋮

D22223333

⋮

D11111111 _

def: Uma <u>máquina de Turing</u> determinística é uma 6-upla

$$M = (\Sigma_0, \Sigma, Q, q, F, \delta)$$

δ: função de transição

F: estados finais ou terminais ou de parada

q: estado inicial

Q: estados

Σ: alfabeto da fita

Σ_0: alfabeto de entrada

Satisfazendo:

- Σ_0, Σ, Q, F são conjuntos

- $\Sigma_0 \subseteq \Sigma$
- $\triangleright \notin \Sigma_0$, $\triangleright \in \Sigma$
- $\sqcup \in \Sigma$, símbolo "em branco"
- $\leftarrow, \rightarrow \notin \Sigma$
- $q \in Q$
- $F \subseteq Q$
- δ é uma função, $\delta : [(Q \setminus F) \times \Sigma] \rightarrow [Q \times ((\Sigma \setminus \{\triangleright\}) \cup \{\leftarrow, \rightarrow\})]$

determinismo

estado atual da máquina (não-final)

símbolo lido pelo cabeçote

novo estado

símbolo a ser escrito

ou

movimento a ser feito

- $\forall q \in Q \setminus F \ \forall q' \in Q \ \forall s \in \Sigma \setminus \{\triangleright\} \ \left(\delta(q, \triangleright) \neq (q', \leftarrow) \wedge \delta(q, \triangleright) \neq (q', s)\right)$

ou, equivalente,

$\forall q \in Q \setminus F \ \exists q' \in Q \ \left(\delta(q, \triangleright) = (q', \rightarrow)\right)$

De agora em diante, seja $M = (\Sigma_0, \Sigma, Q, q_0, F, \delta)$ uma máq. de Turing.

<u>def</u>: Uma <u>palavra marcada</u> é um elemento $w \in \Sigma^* \setminus \{\varepsilon\}$ satisfazendo:

palavra vazia

1) O primeiro símbolo de w é $\triangleright$: $w = \triangleright w'$ para algum $w' \in \Sigma^*$

2) w tem alguma posição "marcada" (sublinhada), $w = w_1 \underline{a} w_2$ com $w_1, w_2 \in \Sigma^*$ e $a \in \Sigma$, de forma que se $w_2 \neq \varepsilon$, então o último símbolo de w_2 não é $\sqcup$

Ex: $\underbrace{\triangleright a}_{w_1} \underline{b} \underbrace{\sqcup \sqcup a}_{w_2}$

Não ex: $\underbrace{\triangleright a b}_{w_1} \underline{\sqcup} \underbrace{\sqcup \sqcup}_{w_2}$ $\quad\quad$ $\underbrace{\triangleright a b}_{w_1} \underline{\sqcup} \underbrace{\sqcup}_{w_2}$

Ex: $\underbrace{\triangleright a}_{w_1} b \underline{\sqcup} \underbrace{\ }_{w_2 = \varepsilon}$

def: Uma configuração de M é um par (q, w) onde $q \in Q$ e w é uma palavra marcada.

def: Dadas duas configurações (q, w) e (q', w'), dizemos que (q, w) produz (q', w'), denotado $(q, w) \vdash (q', w')$,
Yields
se $w = w_1 \underline{a} w_2$, $w' = w_1' \underline{a'} w_2'$, $q \in Q \setminus F$ e $\delta(q, a) = (q', x)$, com 1, 2 ou 3 satisfeito:

1) $x \in \Sigma$, $w_1' = w_1$, $w_2' = w_2$, $a' = x$

2) $x = \leftarrow$, $w_1 = w_1' a'$ e:

a) se $a = \sqcup$ e $w_2 = \varepsilon$, então $w_2' = \varepsilon$

b) caso contrário, $w_2' = a w_2$

3) $x = \rightarrow$, $w_1' = w_1 a$ e:

a) se $w_2 = \varepsilon$, então $a' = \sqcup$ e $w_2' = \varepsilon$

b) caso contrário, $w_2 = a' w_2'$

$\left\{ \begin{array}{l} w_1 \underline{a} w_2 \\ w_1' \underline{a'} w_2' \end{array} \right\}$ $\rhd x y \sqcup \sqcup \sqcup \underline{\sqcup}$

1) escrita

2) movimento p/ esq

3) movimento p/ dir

def: Uma <u>execução</u> de M é uma sequência (finita ou infinita) de configurações de M, começando em uma configuração do tipo (q_0, w), onde cada nova configuração
estado inicial
é produzida pela anterior e, caso termine, com última configuração do tipo (q_f, w') com $q_f \in F$.

$$(q_0, w_0) \vdash (q_1, w_1) \vdash (q_2, w_2) \vdash \dots$$

Exemplos

1) seja $M = \left(\overset{\Sigma_0}{\{a\}}, \overset{\Sigma}{\{a, \triangleright, \sqcup\}}, \overset{Q}{\{q_0, q_1, h\}}, \overset{q}{q_0}, \overset{F}{\{h\}}, \overset{\delta}{\delta} \right)$ onde

δ	a	$\triangleright$	$\sqcup$
q_0	$(q_1, \sqcup)$	$(q_0, \rightarrow)$	$(h, \sqcup)$
q_1	(q_0, a)	$(q_1, \rightarrow)$	$(q_0, \rightarrow)$

Executando em $w_0 = \underline{\triangleright}\, a\, a\, \sqcup\, a\, a\, a$

$w_1 = \triangleright\, \underline{a}\, a\, \sqcup\, a\, a\, a$

$w_2 = \triangleright\, a\, \underline{a}\, \sqcup\, a\, a\, a$

$w_3 = \triangleright\, a\, a\, \underline{\sqcup}\, a\, a\, a$

$w_0: (q_0, \underline{\triangleright} aa \sqcup aaa) \vdash (q_0, \triangleright \underline{a} a \sqcup aaa)$ caso 3b

$\vdash (q_1, \triangleright \underline{\sqcup} a \sqcup aaa)$ caso 1

$\vdash (q_0, \triangleright \sqcup \underline{a} \sqcup aaa)$ caso 3b

$\vdash (q_1, \triangleright \sqcup \underline{\sqcup} \sqcup aaa)$ caso 1

$\vdash (q_0, \triangleright \sqcup \sqcup \underline{\sqcup} aaa)$ caso 3b

$\vdash (h, \triangleright \sqcup \sqcup \underline{\sqcup} aaa)$

$w_1: (q_0, \triangleright \underline{a} a \sqcup aaa)$, mesma execução acima começando na 2ª config.

w_2: $(q_0, \triangleright a\underline{a}\sqcup aaa) \vdash (q_1, \triangleright a\underline{\sqcup}\sqcup aaa)$

$\vdash (q_0, \triangleright a\sqcup\underline{\sqcup}aaa)$

$\vdash (h, \triangleright a\sqcup\underline{\sqcup}aaa)$

w_3: $(q_0, \triangleright aa\underline{\sqcup}aaa) \vdash (h, \triangleright aa\underline{\sqcup}aaa)$

w_4: $(q_0, \triangleright aa\sqcup\underline{a}aa) \vdash$ .... exercício

$\vdash (h, \triangleright aa\sqcup\sqcup\sqcup\sqcup\underline{\sqcup})$

<u>def</u>: Um <u>decisor</u> é uma máquina de Turing

$M = (\Sigma_0, \Sigma, Q, q, F, \delta)$ satisfazendo

1) $F = \{s, n\}$

↳ rejeição

↳ aceitação

2) Existe $L \subseteq \Sigma_0^*$ tal que, sempre que a máquina M é executada com $w \in \Sigma_0^*$ na fita, imediatamente à direita de $\triangleright$, com o primeiro símbolo de w marcado, a execução para com:

a) se $w \in L$, então a máquina para no estado s

b) se $w \notin L$, então a máquina para no estado n

L é chamada linguagem decidida por M, ou M decide L.

def: Uma linguagem $L \subseteq \Sigma_0^*$ é chamada recursiva ou decidível se existe um decisor $M = (\Sigma_0, \Sigma, Q, q, F, \delta)$ para L.

Teorema: Toda linguagem livre de contexto é decidível

Prova: No futuro!

Corolário: Toda linguagem regular é decidível.

Prova: Se L é regular, então L é aceita por um autômato finito $A = (\Sigma_f, Q_f, q_f, F_f, \delta_f)$,

definimos $M = (\Sigma_0, \Sigma, Q, q, F, \delta)$ com

- $\Sigma_0 = \Sigma_f$

- $\Sigma = \Sigma_0 \cup \{\triangleright, \sqcup, X\}$ com $\triangleright, \sqcup, X \notin \Sigma_f$
- $Q = Q_f \cup \{s, n\}$, com $s, n \notin Q_f$
- $q_r = q_{f}$
- $F = \{s, n\}$
- δ: $\delta(q, \sigma) = (\delta_f(q, \sigma), X)$ se $\sigma \in \Sigma_f$

  $\delta(q, X) = (q, \rightarrow)$

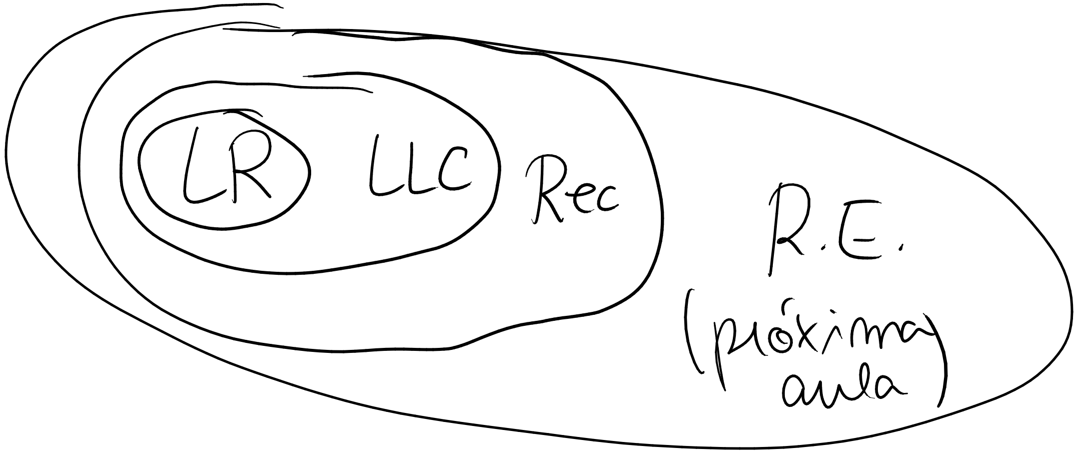
  $\delta(q, \triangleright)$ //

  $\delta(q, \sqcup) = (s, \sqcup)$, se $q \in F_f$

$$\delta(q, \sqcup) = (n, \sqcup) \quad , \text{ se } q \notin F_f$$

Tarefa: por que escrever X & depois $\rightarrow$, em vez de simplesmente $\rightarrow$?

Verificamos que M decide L. $\blacksquare$



Rascunho

$\triangleright\ a_0\ a_1\ a_2\ a_3\ a_4$

2 passos

$\triangleright\ X\ a_1\ a_2\ a_3\ a_4$