Máquinas de Turing

Luís Felipe

UFF

07 de Junho de 2023

Luís Felipe
07/06/23

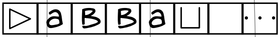
Máquina de Turing

- Até o momento trabalhamos com autômatos finitos, que são bons modelos para dispositivos com quantidade limitada de memória, e com autômatos com pilha, que são bons modelos para dispositivos que têm memória ilimitada mas que só pode ser utilizada com restrições tais quais a de uma pilha.
- A partir de agora, vamos trabalhar com uma máquina que pode fazer tudo que um computador real pode fazer: A Máquina de Turing.
- Proposta em 1936, por Alan Turing, o pai da computação.
- Ainda assim, as máquinas de Turing não resolvem qualquer problema.
- Esses problemas vão além dos limites teóricos da computação.

Luís Felipe

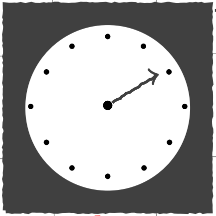
07/06/23

Esquema



fita infinita à direita
dividida em casas

cada casa contém
um símbolo ou está
em branco (só um n.
finito de casas não está
em branco em
qualquer momento)



cabeçote de
leitura e escrita

move-se para
esq. e dir. uma
casa de cada vez

conjunto finito
de estados

Luís Felipe

07/06/23

Definição formal

Formalmente, uma máquina de Turing (MT) é um 6-upla $M = (\Sigma_0, \Sigma, Q, q_0, F, \delta)$, onde:

1. Σ_0 é o alfabeto da entrada, satisfazendo $\Sigma_0 \cap \{\triangleright, \sqcup, \leftarrow, \rightarrow\} = \emptyset$
2. Σ é o alfabeto da fita, satisfazendo $\Sigma_0 \cup \{\triangleright, \sqcup\} \subseteq \Sigma$ e $\Sigma \cap \{\leftarrow, \rightarrow\} = \emptyset$
3. Q é o conjunto finito de estados
4. $q_0 \in Q$ é o estado inicial
5. $F \subseteq Q$ é o conjunto de estados finais
6. δ é a função de transição determinística $\delta : (Q \setminus F) \times \Sigma \rightarrow Q \times ((\Sigma \setminus \{\triangleright\}) \cup \{\leftarrow, \rightarrow\})$

Luís Felipe

07/06/23

OBS.:

1. A transição é definida apenas para estados que não são finais. Logo, a máquina SEMPRE para ao alcançar um estado final.
2. A máquina não pode escrever o símbolo $\triangleright$ na fita para não interferir em seu uso como marcador da ponta esquerda da fita.
3.
   - Se $\delta(q, a) = (q', \rightarrow)$, então a máquina muda de q para q' e move o cabeçote uma casa para a direita na fita.
   - Se $\delta(q, a) = (q', \leftarrow)$, então a máquina muda de q para q' e move o cabeçote uma casa para a esquerda na fita.
   - Se $\delta(q, a) = (q', b)$, então a máquina muda de q para q', não move o cabeçote, e substitui o a pelo b na casa atual da fita.
4. A função δ deve satisfazer: $\delta(q, \triangleright) = (q', \rightarrow)$, onde $q', q \in Q$.

Luís Felipe

07/06/23

Autômatos finitos vs máquinas de Turing

- A *MT* pode escrever e ler a fita
- O cabeçote pode se mover para a esquerda e para a direita
- A fita é infinita
- Os estados finais fazem efeito imediato

Luís Felipe
07/06/23

Exemplos:

1. $\Sigma_0 = \{a\}, \Sigma = \Sigma_0 \cup \{\triangleright, \sqcup\}, Q = \{q_0, q_1, h\}, F = \{h\}$

q	σ	$\delta(q, \sigma)$
q_0	a	$(q_1, \sqcup)$
q_0	$\sqcup$	$(h, \sqcup)$
q_0	$\triangleright$	$(q_0, \rightarrow)$
q_1	a	(q_0, a)
q_1	$\sqcup$	$(q_0, \rightarrow)$
q_1	$\triangleright$	$(q_1, \rightarrow)$

$\triangleright$	a	a	a	$\sqcup$	a	a

OBS.: Esta máquina move para a direita até achar um símbolo $\sqcup$.

Luís Felipe
07/06/23

Exemplos:

2. $\Sigma_0 = \{a\}, \Sigma = \Sigma_0 \cup \{\triangleright, \sqcup\}, Q = \{q_0, h\}, F = \{h\}$

q	σ	$\delta(q, \sigma)$
q_0	a	$(q_0, \leftarrow)$
q_0	$\sqcup$	$(h, \sqcup)$
q_0	$\triangleright$	$(q_0, \rightarrow)$

$\triangleright$	a	a	$\sqcup$	a	a

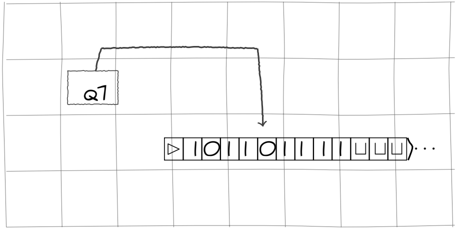
OBS.: Esta máquina move para a esquerda até achar um símbolo $\sqcup$. O problema é quando não encontrar.

Luís Felipe
07/06/23

Configurações

- Uma *MT* começa a computação com uma palavra $w = \triangleright w'$ onde $w' \in (\Sigma_0 \cup \{\sqcup\})^\star$, na extremidade esquerda da fita e o cabeçote sobre o primeiro símbolo de w'.

- Uma configuração de uma *MT* é uma expressão de três itens da computação:
  - estado atual que a *MT* encontra-se
  - conteúdo atual da fita, e
  - posição do cabeçote



Notação: $(q_7, \triangleright 1011\underline{0}1111)$, onde o cabeçote encontra-se sempre no símbolo sublinhado.

Luís Felipe

07/06/23

Configuração seguinte

Sejam C, C' configurações de uma MT. Dizemos que C' é uma configuração seguinte a C, i.e. $C \vdash C'$, nos seguintes casos:

1. $C = (q, \triangleright\sigma_1 \dots \underline{\sigma_i} \dots \sigma_k)$, $C' = (q', \triangleright\sigma_1 \dots \underline{\gamma} \dots \sigma_k)$ e $\delta(q, \sigma_i) = (q', \gamma)$
2. $C = (q, \triangleright\sigma_1 \dots \underline{\sigma_i} \dots \sigma_k)$, $C' = (q', \triangleright\sigma_1 \dots \underline{\sigma_{i-1}}\sigma_i \dots \sigma_k)$ e $\delta(q, \sigma_i) = (q', \leftarrow)$
3. $C = (q, \triangleright\sigma_1 \dots \underline{\sigma_i} \dots \sigma_k)$, $C' = (q', \triangleright\sigma_1 \dots \sigma_i\underline{\sigma_{i+1}} \dots \sigma_k)$ e $\delta(q, \sigma_i) = (q', \rightarrow)$

Uma computação em uma MT $M = (\Sigma_0, \Sigma, Q, q_0, F, \delta)$ é uma sequência de configurações

$$(q_0, \triangleright\underline{\sigma_1} \dots \sigma_i \dots \sigma_k) \vdash \dots$$

Luís Felipe
07/06/23

Aceitação

- Uma MT aceita uma palavra w onde $w \in (\Sigma_0 \cup \{\sqcup\})^\star$, se existe uma computação $(q_0, \triangleright\sigma_1 \ldots \sigma_k) \vdash \ldots \vdash (h, v)$, onde v é uma descrição qualquer da fita e $h \in F$.
- A linguagem aceita (ou reconhecida) por uma MT M é o conjunto de todas as palavras aceitas por M.
    - $w \in L \leftrightarrow M$ para com entrada w.
- Uma linguagem é Recursivamente Enumerável ou Turing Reconhecível ou Semi-decidível se é aceita por um MT.
    - Só temos resposta para o SIM.

OBS.: Em termos de aceitação de linguagens, toda MT é equivalente a uma MT com um único estado final.

Luís Felipe
07/06/23

Decisor

- Um decisor é um tipo especial de MT com 2 estados finais (S e N) tal que, com qualquer entrada, a MT sempre alcança um estado final.

- Um decisor aceita uma palavra w se ele para no estado S ao receber w como entrada.

- Um decisor rejeita uma palavra w se ele para no estado N ao receber w como entrada.

- Um decisor decide uma linguagem L se
  - $w \in L \leftrightarrow$ o decisor para em S ao receber w como entrada E
  - $w \notin L \leftrightarrow$ o decisor para em N ao receber w como entrada.

- Ou seja, temos resposta para o SIM e NÃO.

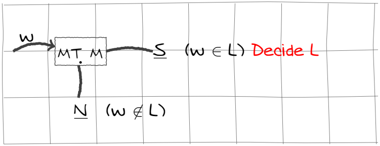
- Uma linguagem L é recursiva ou Turing-decidível ou decidível se existe uma MT que a decide.

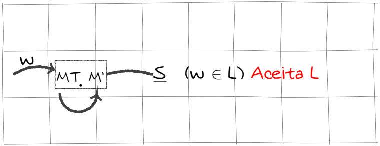
Luís Felipe

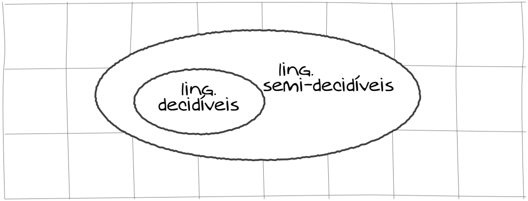
07/06/23

Esquema

OBS.: Se existe uma *MT* que decide *L*, então existe uma *MT* que aceita *L*.







Luís Felipe
07/06/23

Como será?

A linguagem $L = \{w\#w \mid w \in \{0,1\}^*\}$ não é LLC. Mostre pelo Lema do bombeamento. Possível palavra: $s = 0^p1^p\#0^p1^p$.

Será que existe um decisor para L?

- A ideia é emparelhar a cadeia antes do símbolo # com a cadeia que vem depois.
- Para tal, a máquina faz múltiplas varreduras na fita
- Ela lê o primeiro símbolo e marca
- Lê o primeiro símbolo após # e marca, se for o mesmo símbolo
  - Se não for, rejeita
- Quando todos os símbolos à esquerda de # tiverem sido marcados, ela verifica se tem algum desmarcado
  - Se houver, rejeita
  - Senão, aceita