universidade de aveiro

Teoria da Computação

Terceiro Teste

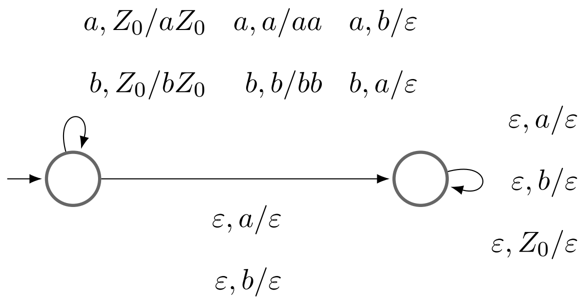
2019–2020

Data: 20 de Dezembro de 2019 Duração: 60 minutos

1. (8 valores) Considere a linguagem $L_1 = \{w \in \{a,b\}^\star : \#_a(w) \neq \#_b(w)\}$.

   (a) Construa um autómato de pilha que reconheça L_1 por pilha vazia.

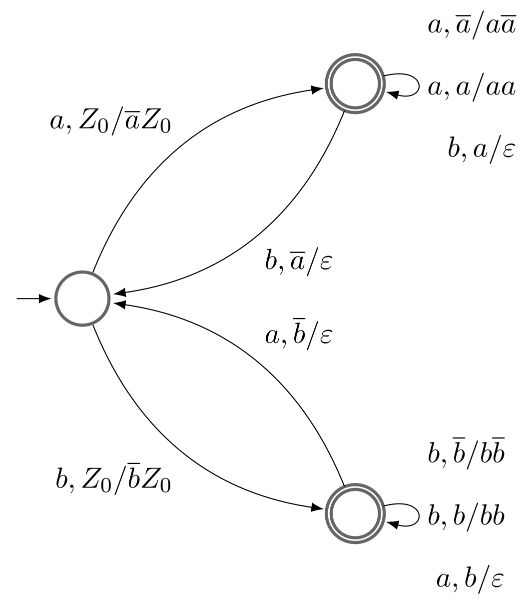
Solução: Se no final da leitura de w ainda restar algum a ou b na pilha então é porque $\#_a(w) \neq \#_b(w)$. O AP seguinte é não determinista e reconhece a linguagem na modalidade de pilha vazia.



   (b) Construa um autómato de pilha determinista que reconheça L_1. Indique a modalidade de reconhecimento.

Solução:

O AP seguinte é determinista e reconhece a linguagem na modalidade de estados de aceitação. A estratégia consiste em marcar a primeira letra lida (e guardada na pilha) para que se possa identificar quando o número de a's lidos é igual ao número de b's lidos.



2. (2 valores) Considere o problema da paragem

$$\text{HALT} = \{\langle M, w\rangle :\ M \text{ é uma máquina de Turing que pára com } w\}$$

(a) Mostre que HALT é recorrentemente enumerável.

Solução: Basta considerar uma pequena variação da Máquina de Turing Universal:

Dados $\langle M, w\rangle$ onde M é uma MT e w é uma palavra

- simula M com a entrada w;
- se M atinge um estado de aceitação então aceita;

  se M atinge um estado de rejeição então aceita.

Note-se que esta máquina de Turing pode não parar: basta que M não pare com w. No entanto esta nova máquina Turing pára e aceita todos os pares $\langle M, w\rangle$ tais que a máquina de Turing M pára com w, ou seja reconhece a linguagem HALT.

(b) Mostre que $\overline{\text{HALT}}$ não é recorrentemente enumerável.

Solução: Sempre que uma linguagem L e a sua linguagem complementar $\overline{L}$ são recorrentemente enumeráveis então também L e $\overline{L}$ são decidíveis (basta executar as máquinas M_L e $M_{\overline{L}}$ em paralelo).

Assim, uma vez que HALT é recorrentemente enumerável, se $\overline{\text{HALT}}$ fosse recorrentemente enumerável então HALT seria decidível o que sabemos ser falso. Logo $\overline{\text{HALT}}$ não pode ser recorrentemente enumerável.

3. (4 valores) Sabendo que o problema de decisão $\mathrm{E}_{\mathrm{MT}} = \{\langle M\rangle : \ \mathcal{L}(M) = \emptyset\}$ é indecidível, mostre que o problema $\mathrm{DIF}_{\mathrm{MT}} = \{\langle M_1, M_2\rangle : \ M_1$ e M_2 são máquinas de Turing e $\mathcal{L}(M_1) \neq \mathcal{L}(M_2)\}$ é também indecidível.

Solução: Admitamos que $\mathrm{DIF}_{\mathrm{MT}}$ é decidível. Então existe uma MT M_{DIF} tal que, para $\langle M_1, M_2\rangle$,

- Se $\mathcal{L}(M_1) \neq \mathcal{L}(M_2)$ então M_{DIF} pára e aceita $\langle M_1, M_2\rangle$;
- Se $\mathcal{L}(M_1) = \mathcal{L}(M_2)$ então M_{DIF} pára e rejeita $\langle M_1, M_2\rangle$.

Seja $M_\emptyset$ uma qualquer máquina de Turing tal que $\mathcal{L}(M_\emptyset) = \emptyset$.

Se existe a MT M_{DIF} então também existe a MT M_{E} tal que, para cada $\langle M\rangle$:

- simula o funcionamento da máquina M_{DIF} com $\langle M, M_\emptyset\rangle$.
- Se a simulação para num estado de aceitação então M_{E} rejeita $\langle M\rangle$;
- Se a simulação para num estado de rejeição então M_{E} aceita $\langle M\rangle$.

Esta máquina decide a linguagem E_{MT} uma vez que:

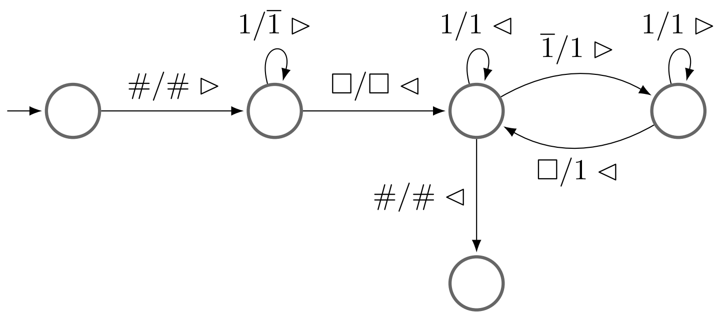
- Se $\mathcal{L}(M) = \emptyset$ então a máquina M_{DIF} rejeita $\langle M, M_\emptyset\rangle$ e a máquina M_{E} aceita $\langle M\rangle$;
- Se $\mathcal{L}(M) \neq \emptyset$ então a máquina M_{DIF} aceita $\langle M, M_\emptyset\rangle$ e a máquina M_{E} rejeita $\langle M\rangle$.

Mas a linguagem E_{MT} é indecidível, pelo que esta máquina não pode existir!

4. (6 valores) Em cada uma das seguintes alíneas construa uma máquina de Turing que satisfaça a especificação indicada:

(a) $\#1^k \vdash^\star \#1^{2k}$, $k \geq 1$;

Solução: A MT seguinte começa por marcar todos os 1's e em seguida vai acrescentado no final um 1 por cada $\overline{1}$ que encontra (desmarcando esse $\overline{1}$). A máquina pára exatamente antes do símbolo #.



(b) $1^n \vdash^{\star} 1^n\#1^{2^n}$, $n \geq 1$. (Dica: $2^n = 2 \times 2^{n-1}$)

Solução: Seguindo a sugestão, basta ir duplicando o número 1's após o separador # tantas vezes quantas o número de 1's antes do separador. A rotina DUPLICA representada na figura seguinte corresponde à MT implementada na alínea anterior. Note que no estado q_3, antes de executar a rotina DUPLICA, a máquina está sobre o símbolo # e que após executar a rotina a máquina fica sobre o símbolo que precede #. Inicialmente a máquina começa por colocar um 1 após # e voltar ao início da palavra.