Dep. Matem. e Aplic., Univ. Minho — 11 de Novembro de 2011

Resolução do 1º Teste de

Computabilidade

Lic. Ciências Computação — Duração: 2 horas

1. Seja $A = \{a, b\}$. Considere a máquina de Turing

$$\mathcal{T} = (\{0, 1, 2, 3, 4\}, A, A \cup \{\Delta\}, \delta, 0, 4, \Delta)$$

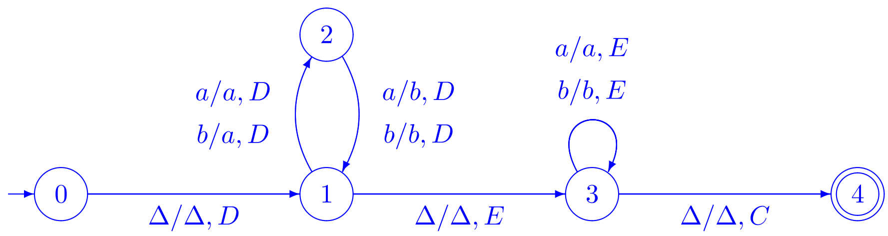
onde a função transição δ é definida pela tabela seguinte:

δ	a	b	Δ
0			$(1, \Delta, D)$
1	$(2, a, D)$	$(2, a, D)$	$(3, \Delta, E)$
2	$(1, b, D)$	$(1, b, D)$	
3	$(3, a, E)$	$(3, b, E)$	$(4, \Delta, C)$

A máquina $\mathcal{T}$ calcula uma função parcial $g : A^* \to A^*$.

a) Represente $\mathcal{T}$ graficamente.

R: *$\mathcal{T}$ pode ser representada graficamente da seguinte forma:*



b) Indique a sequência de configurações que podem ser computadas a partir da configuração $(0, \underline{\Delta}aaaabaab)$.

R: *A sequência de configurações de $\mathcal{T}$ que podem ser computadas a partir da configuração $(0, \underline{\Delta}aaaabaab)$ é a seguinte:*

$$\begin{aligned}
&(0, \underline{\Delta}aaaabaab) \longrightarrow (1, \Delta\underline{a}aaabaab) \longrightarrow (2, \Delta a\underline{a}aabaab) \longrightarrow (1, \Delta ab\underline{a}abaab) \\
&\longrightarrow (2, \Delta aba\underline{a}baab) \longrightarrow (1, \Delta abab\underline{b}aab) \longrightarrow (2, \Delta ababa\underline{a}ab) \longrightarrow (1, \Delta ababab\underline{a}b) \\
&\longrightarrow (2, \Delta abababa\underline{b}) \longrightarrow (1, \Delta abababab\underline{\Delta}) \longrightarrow (3, \Delta abababa\underline{b}) \stackrel{*}{\longrightarrow} (3, \underline{\Delta}abababab) \\
&\longrightarrow (4, \underline{\Delta}abababab).
\end{aligned}$$

c) Identifique o domínio D da função g.

R: *O domínio D da função g é a linguagem reconhecida por $\mathcal{T}$. Ora, partindo da configuração inicial $(0, \underline{\Delta}u)$ de uma palavra $u \in A^*$, a máquina $\mathcal{T}$ chega ao estado final se e só se u tem comprimento par (quando u tem comprimento ímpar, $\mathcal{T}$ pára no estado 2). Logo,*

$$D = L(\mathcal{T}) = \{u \in A^* : |u| = 2k, k \in \mathbb{N}_0\}.$$

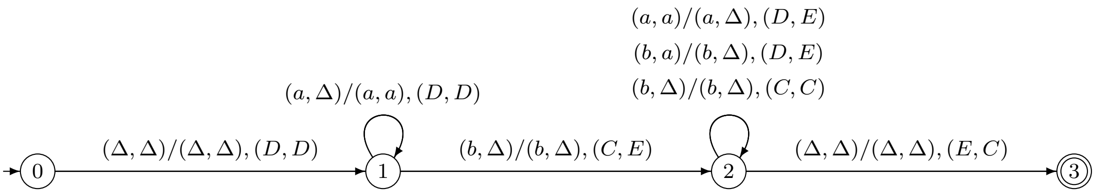
d) Para cada elemento $u \in D$, determine a palavra $g(u)$.

R: *Para $u \in D$, tem-se*

$$g(u) = (ab)^{\frac{|u|}{2}}.$$

De facto, $\mathcal{T}$ substitui, em u: cada letra numa posição ímpar por a (quando transita do estado 1 para o estado 2); cada letra numa posição par por b (quando transita do estado 2 para o estado 1). Dado que o comprimento de $|u|$ é par, o resultado é portanto o indicado.

2. Seja $A = \{a, b\}$ e seja $\mathcal{T}$ a seguinte máquina de Turing sobre A com duas fitas,



a) Indique a sequência de configurações que podem ser computadas a partir da configuração $(0, \underline{\Delta}aaabab, \underline{\Delta})$ e diga se a palavra $aaabab$ é aceite por $\mathcal{T}$.

R: *A sequência de configurações de $\mathcal{T}$ que podem ser computadas a partir da configuração $(0, \underline{\Delta}aaabab, \underline{\Delta})$ é a seguinte:*

$$(0, \underline{\Delta}aaabab, \underline{\Delta}) \longrightarrow (1, \Delta\underline{a}aabab, \Delta\underline{\Delta}) \xrightarrow{*} (1, \Delta aaa\underline{b}ab, \Delta aaa\underline{\Delta}) \longrightarrow (2, \Delta aaa\underline{b}ab, \Delta aa\underline{a})$$
$$\longrightarrow (2, \Delta aaab\underline{a}b, \Delta a\underline{a}) \longrightarrow (2, \Delta aaaba\underline{b}, \Delta\underline{a}) \longrightarrow (2, \Delta aaabab\underline{\Delta}, \underline{\Delta}) \longrightarrow (3, \Delta aaaba\underline{b}, \underline{\Delta}).$$

A palavra aaabab é portanto aceite por $\mathcal{T}$, já que, a partir da configuração inicial desta palavra se computou uma configuração de aceitação, isto é, uma configuração cujo estado é o estado final de $\mathcal{T}$ (neste caso o estado 3).

b) Identifique a linguagem L reconhecida por $\mathcal{T}$.

R: *A linguagem reconhecida por $\mathcal{T}$ é*

$$L = \{a^n bx : x \in A^*, n = |bx|\}.$$

Com efeito, partindo da configuração inicial de uma palavra $w \in A^*$, para se atingir o estado final da máquina $\mathcal{T}$ é necessário passar do estado 1 para o estado 2. Para isso tem que se ler uma (a primeira) ocorrência de b em w, depois de ter lido todas as ocorrências anteriores da letra a e de as copiar para a $2^{\underline{a}}$ fita. Então w tem de ser da forma $w = a^n bx$ para alguns $n \in \mathbb{N}_0$ e $x \in A^*$. Agora, no estado 2, por cada letra da palavra bx que a máquina $\mathcal{T}$ lê na $1^{\underline{a}}$ fita, apaga um a na $2^{\underline{a}}$ fita. Portanto, os cursores atingem simultaneamente o símbolo branco em cada fita se e só se $n = |bx|$, e só nesse caso é que $\mathcal{T}$ vai para o estado final.

c) Para que palavras $u \in A^*$, $(0, \underline{\Delta}u, \underline{\Delta})$ é uma configuração de ciclo?

R: *Como se pode verificar, analisando o comportamento da máquina $\mathcal{T}$, uma configuração $(0, \underline{\Delta}u, \underline{\Delta})$ é de ciclo se e só se a partir dela se chega a uma configuração em que a máquina está no estado 2, o cursor da $1^{\underline{a}}$ fita está a ler b e o cursor da $2^{\underline{a}}$ fita está a ler Δ (e está na $1^{\underline{a}}$ célula). Ou seja, se é possível fazer uma computação*

$$(0, \underline{\Delta}u, \underline{\Delta}) \xrightarrow{*} (2, u_1\underline{b}u_2, \underline{\Delta})$$

em $\mathcal{T}$. Note-se para além disso que, neste caso, $u = u_1 b u_2$ pois o conteúdo da $1^{\underline{a}}$ fita nunca é alterado. A palavra u_1 pode tomar duas formas:

- *$u_1 = \epsilon$. Neste caso, $u = bu_2 \in bA^*$ e chega-se à configuração $(2, u_1\underline{b}u_2, \underline{\Delta})$ logo que o estado 2 é atingido, ou seja, imediatamente depois de ler o b inicial da palavra u.*
- *$u_1 = a^n bx$ para alguns $n \in \mathbb{N}$ e $x \in A^*$, com $n = |bx|$. Neste caso, $u_1 \in L$ (donde $u \in LbA^*$) e chega-se à configuração $(2, u_1\underline{b}u_2, \underline{\Delta})$ porque depois de se esgotarem os n a's na $2^{\underline{a}}$ fita e de se ter avançado o factor bx em u, ainda "sobrou" em u o sufixo bu_2, começado pela letra b.*

Em resumo, $(0, \underline{\Delta}u, \underline{\Delta})$ é uma configuração de ciclo se e só se

$$u \in bA^* \cup LbA^*.$$

d) Para que palavras $v \in A^*$, a partir de $(0, \underline{\Delta}v, \underline{\Delta})$ pode ser computada uma configuração de rejeição?

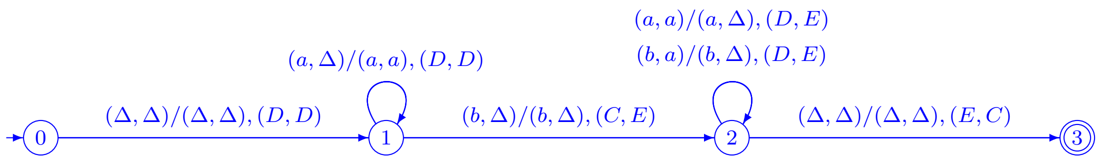
R: *Pode ser computada uma configuração de rejeição a partir de* $(0, \underline{\Delta}v, \underline{\Delta})$ *se e só se esta configuração não é de ciclo e a partir dela não pode ser computada uma configuração de aceitação, ou seja, se e só se*

$$v \in A^* \setminus (bA^* \cup LbA^* \cup L) = a^* \cup LaA^*.$$

Note-se que, nos casos em que $v \in a^*$, *a máquina* $\mathcal{T}$ *pára no estado* 1, *e quando* $v \in LaA^*$ *a máquina* $\mathcal{T}$ *pára no estado* 2 *(depois de ter lido o prefixo* $a^n bx \in L$ *de* u *e de ter ficado a ler a letra* a *que sucede a esse prefixo).*

e) Verifique que é possível fazer uma alteração (simples) na máquina $\mathcal{T}$ de modo a obter uma máquina de Turing $\mathcal{T}'$ que reconhece L e que nunca entra em ciclo. Conclua que L é recursiva.

R: *Basta eliminar em* $\mathcal{T}$ *a transição do estado* 2 *para* 2 *cuja etiqueta é* $(b, \Delta)/(b, \Delta), (C, C)$. *Esta eliminação não altera a linguagem reconhecida pela máquina, pois as palavras que originavam ciclos originam agora paragens no estado (não final)* 2. *A máquina* $\mathcal{T}'$



assim obtida reconhece, portanto, a linguagem L *e é um algoritmo. Ora, existe uma caracterização das linguagens recursivas como sendo aquelas que são reconhecidas por algoritmos. Pode-se então concluir que a linguagem* L *é recursiva.*

4. Indique, justificando, a veracidade ou falsidade das seguintes afirmações.

a) Existe uma linguagem não recursiva L sobre o alfabeto $\{x, y\}$ tal que $L \cap AA$ e $L \cap NAA$ são linguagens recursivas.

R: *Seja* L *uma linguagem sobre o alfabeto* $\{x, y\}$. *Note-se que*

$$L = (L \cap AA) \cup (L \cap \overline{AA}) = (L \cap AA) \cup (L \cap NAA).$$

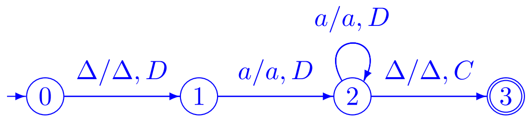
Suponhamos que $L \cap AA$ *e* $L \cap NAA$ *são linguagens recursivas. Então* L *é recursiva pois a união de duas linguagens recursivas ainda é uma linguagem recursiva. Logo, a afirmação é falsa.*

b) A linguagem reconhecida pela composição sequencial $\mathcal{T}_1 \longrightarrow \mathcal{T}_2$, de duas máquinas de Turing $\mathcal{T}_1$ e $\mathcal{T}_2$, está contida na linguagem reconhecida pela máquina $\mathcal{T}_1$.

R: *Para* $k \in \{1, 2\}$, *sejam* i_k *e* f_k *os estados, respetivamente, inicial e final da máquina* $\mathcal{T}_k$. *Então, por definição,* i_1 *é o estado inicial e* f_2 *é o estado final da máquina* $\mathcal{T}_1 \longrightarrow \mathcal{T}_2$.
Seja u *uma palavra aceite pela máquina* $\mathcal{T}_1 \longrightarrow \mathcal{T}_2$. *Então, a partir da configuração inicial* $(i_1, \underline{\Delta}u)$ *de* u *é possível computar uma configuração de aceitação, isto é, atinge-se* f_2, *o estado final da máquina* $\mathcal{T}_1 \longrightarrow \mathcal{T}_2$. *Ora, dado que a única "passagem" de* $\mathcal{T}_1$ *para* $\mathcal{T}_2$ *é do estado* f_1 *para o estado* i_2, *deduz-se que nessa computação é necessariamente atingido o estado* f_1. *Ora, isto significa que a palavra* u *é aceite pela máquina* $\mathcal{T}_1$.
Provou-se assim que $L(\mathcal{T}_1 \longrightarrow \mathcal{T}_2) \subseteq L(\mathcal{T}_1)$, *o que mostra que a afirmação é verdadeira.*

c) Existem um alfabeto A e uma máquina de Turing $\mathcal{T}$ de alfabeto de entrada A tais que $\mathcal{T}$ rejeita uma e uma só palavra de A^*.

R: *Por exemplo, a máquina de Turing*



de alfabeto de entrada $A = \{a\}$, rejeita unicamente a palavra vazia.

3. Considere a linguagem $L = a^+b^+ \cup b^+a^+$ sobre o alfabeto $A = \{a, b\}$.

a) Indique, justificando, os valores de $\chi_L(aabbba)$ e de $\chi_L(baaa)$, onde χ_L é a função característica de L.

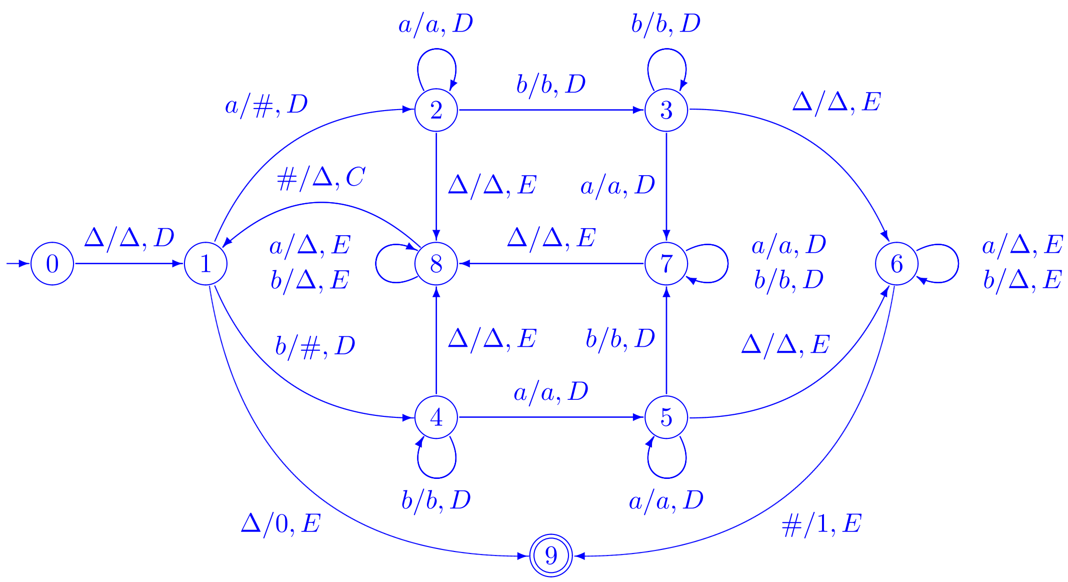
R: *Note-se que, para uma palavra u sobre o alfabeto A,*

$$\begin{aligned} u \in L \quad &\Leftrightarrow \quad u \in a^+b^+ \text{ ou } u \in b^+a^+ \\ &\Leftrightarrow \quad u = a^m b^n \text{ ou } u = b^m a^n, \text{ para alguns } m, n \in \mathbb{N}. \end{aligned}$$

Tem-se, portanto, que $aabbba \notin L$ e $baaa \in L$. Por definição da função χ_L, segue-se que $\chi_L(aabbba) = 0$ e $\chi_L(baaa) = 1$.

b) Construa uma máquina de Turing $\mathcal{T}$ que calcule a função χ_L, e descreva informalmente a estratégia dessa máquina.

R: *Uma máquina de Turing $\mathcal{T}$ que calcula a função χ_L é, por exemplo, a seguinte*



Informalmente, a estratégia de $\mathcal{T}$ é a seguinte. Começando com a configuração inicial $(0, \underline{\Delta}w)$ de uma palavra $w \in A^*$, esta máquina lê a palavra w até ao fim, marcando apenas a primeira posição com um símbolo auxiliar $\#$ (ou seja, substitui a letra inicial de w, na $2^{\underline{a}}$ célula, por $\#$). A seguir, volta à célula inicial apagando w e substituindo o símbolo $\#$ da $2^{\underline{a}}$ célula por 1 ou 0 conforme w pertença ou não à linguagem L. A verificação se w pertence ou não a L é feita do seguinte modo. Se $\mathcal{T}$ chega ao final da leitura de w:

- *nos estados 3 ou 5, então w é da forma aa^mbb^n ou bb^maa^n, donde $w \in a^+b^+ \cup b^+a^+ = L$. Então $\mathcal{T}$ transita para o estado 6 e depois para o estado final deixando 1 na fita.*
- *no estado 1, então $w = \epsilon \notin L$ e $\mathcal{T}$ transita para o estado final deixando 0 na fita.*
- *nos estados 2 ou 4, então w é da forma aa^m ou bb^n, donde $w \notin L$. Então $\mathcal{T}$ transita para o estado 1 e depois para o estado final deixando 0 na fita.*
- *no estado 7, então w é da forma aa^mbb^nau ou bb^maa^nbv com $u, v \in A^*$, donde $w \notin L$. Então $\mathcal{T}$ transita sucessivamente pelos estados 8, 1 e final deixando 0 na fita.*