

1. Máquinas de Turing

1.1 Considere a máquina de Turing

$$\mathcal{T} = (\{0, 1, 2, 3, 4, 5, 6, 7, 8\}, \{a, b\}, \{a, b, \Delta\}, \delta, 0, 8, \Delta)$$

onde a função transição δ é definida pela tabela seguinte:

δ	a	b	Δ
0			$(1, \Delta, D)$
1	$(1, a, D)$	$(1, b, D)$	$(2, \Delta, E)$
2	$(3, \Delta, D)$	$(5, \Delta, D)$	$(8, \Delta, D)$
3			$(4, a, D)$
4	$(4, a, D)$	$(4, b, D)$	$(7, a, E)$
5			$(6, b, D)$
6	$(6, a, D)$	$(6, b, D)$	$(7, b, E)$
7	$(7, a, E)$	$(7, b, E)$	$(2, \Delta, E)$

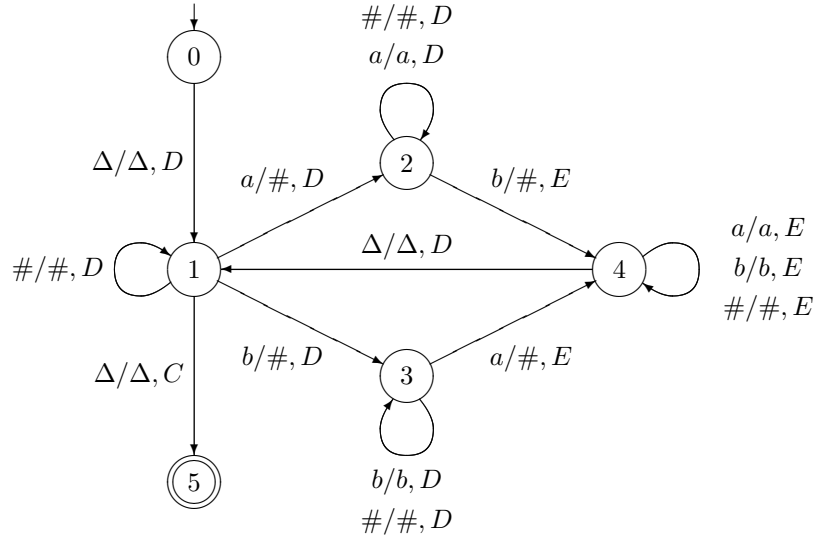
- Represente a máquina de Turing \mathcal{T} através de um grafo.
- Indique a sequência de configurações que podem ser computadas a partir da configuração $(0, \underline{\Delta}ab)$; e a partir da configuração $(0, \underline{\Delta}baa)$?
- Indique informalmente o comportamento de \mathcal{T} , quando a configuração inicial é $(0, \underline{\Delta}u)$, onde u é uma palavra de $\{a, b\}^*$.

1.2 Considere a máquina de Turing $\mathcal{T} = (\{0, 1, 2\}, \{a, b\}, \{a, b, \Delta\}, \delta, 0, 2, \Delta)$, onde a função transição δ é definida pela tabela seguinte:

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

- Indique a sequência de configurações que podem ser computadas a partir de $(0, \underline{\Delta}aab)$.
- Indique uma palavra $u \in \{a, b, \Delta\}^*$ tal que, a partir da configuração $(0, \underline{u})$ pode ser computada uma configuração de:
 - paragem;
 - ciclo;
 - aceitação;
 - rejeição.
- Descreva informalmente o comportamento de \mathcal{T} quando a configuração inicial é $(0, \underline{u})$, onde u é uma palavra sobre $\{a, b, \Delta\}$.
- Calcule a linguagem reconhecida por \mathcal{T} .

1.3 Considere a seguinte máquina de Turing \mathcal{T} de alfabeto de entrada $A = \{a, b\}$,



- Indique a sequência de configurações que podem ser computadas a partir da configuração $(0, \underline{\Delta}ababba)$.
- Identifique a linguagem reconhecida por \mathcal{T} .

1.4 Construa máquinas de Turing que reconheçam cada uma das seguintes linguagens:

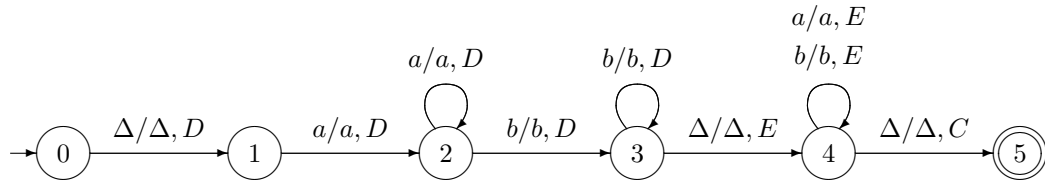
- ab^*a^+ , sobre o alfabeto $\{a, b\}$.
- $\{a^n b^{2n} \mid n \in \mathbb{N}_0\}$, sobre o alfabeto $\{a, b\}$.
- $\{a^n b^{2n} a^n \mid n \in \mathbb{N}_0\}$, sobre o alfabeto $\{a, b\}$.
- $\{a^m b^n \mid m, n \in \mathbb{N}_0, m < n\}$, sobre o alfabeto $\{a, b\}$.
- $\{a^n b^{mn} \mid m, n \in \mathbb{N}_0\}$, sobre o alfabeto $\{a, b\}$.
- $\{a^m b c^n : m + n \text{ é par}\}$, sobre o alfabeto $\{a, b, c\}$.
- $\{w c w^I \in A^* \mid w \in \{a, b\}^*\}$, sobre o alfabeto $\{a, b, c\}$.
- $\{abab^2ab^3 \cdots ab^n a : n \geq 1\}$, sobre o alfabeto $\{a, b\}$.

1.5 Mostre que, para toda a máquina de Turing $\mathcal{T} = (Q, A, T, \delta, i, f, \Delta)$, existe uma máquina de Turing \mathcal{T}' que reconhece a mesma linguagem que \mathcal{T} , e tal que \mathcal{T}' nunca rejeita uma palavra (ou seja, para qualquer palavra $w \in A^*$, \mathcal{T}' aceita w ou a configuração inicial $(i, \underline{\Delta}w)$ associada a w é uma configuração de ciclo).

1.6 Dada uma máquina de Turing \mathcal{T} , defina uma máquina de Turing \mathcal{T}_{aba} tal que:

$$\mathcal{T} \text{ aceita a palavra vazia } \epsilon \iff \mathcal{T}_{aba} \text{ aceita a palavra } aba.$$

1.7 Considere a seguinte máquina de Turing \mathcal{M} de alfabeto de entrada $A = \{a, b\}$,

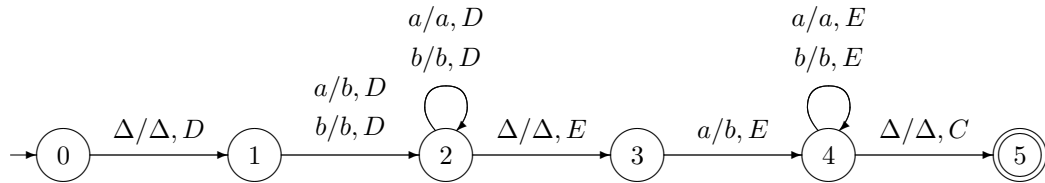


- a) Identifique a linguagem reconhecida pela máquina \mathcal{M} .
- b) Identifique a linguagem reconhecida pela máquina $\mathcal{M} \rightarrow \mathcal{T}$, onde \mathcal{T} é a máquina de Turing do Exercício 1.3.
- 1.8 Construa uma máquina de Turing $\mathcal{T} = (Q, A, T, \delta, i, f, \Delta)$, com alfabeto de entrada $A = \{a, b\}$, que insira uma letra $x \in A$ na célula onde o cursor está posicionado: ou seja, em rigor, que seja capaz de efetuar a computação

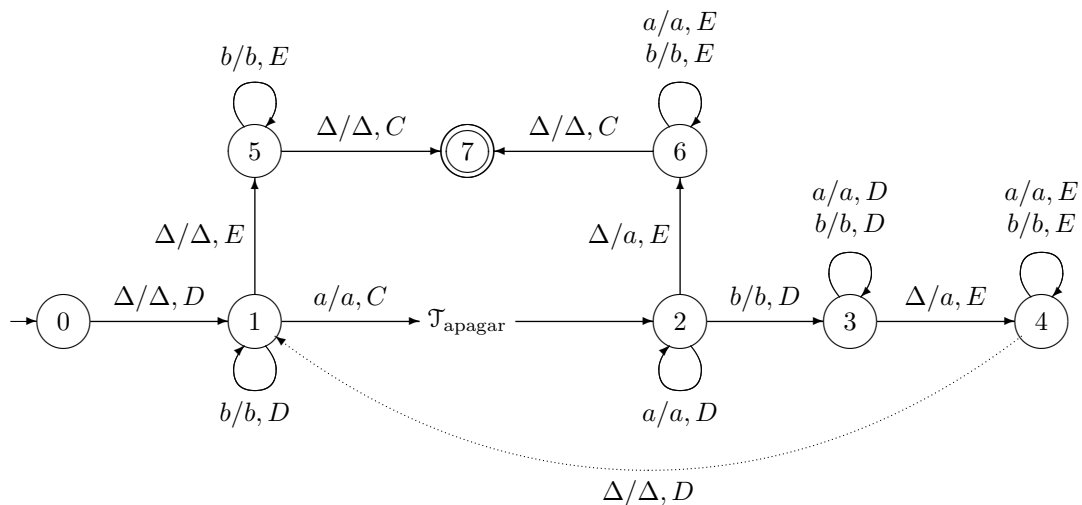
$$(i, uv) \xrightarrow{*} (f, u\underline{x}v)$$

para quaisquer palavras $u \in T^*$ e $v \in A^*$.

1.9 Considere a seguinte máquina de Turing \mathcal{T} de alfabeto de entrada $A = \{a, b\}$,



- a) Indique a sequência de configurações que podem ser computadas a partir da configuração $(0, \underline{\Delta}babba)$.
- b) Identifique o domínio D da função parcial $g : A^* \rightarrow A^*$ calculada por \mathcal{T} .
- c) Para cada palavra $u \in D$, determine a palavra $g(u)$.
- 1.10 A seguinte máquina de Turing calcula uma função g de $\{a, b\}^*$ para $\{a, b\}^*$:



Dada uma palavra $u \in \{a, b\}^*$, descreva a palavra $g(u)$.

1.11 Indique máquinas de Turing que calculem cada uma das seguintes funções:

$$\text{a) } g : \{a, b\}^* \longrightarrow \{1\}^* \\ u \longmapsto 1^{|u|_a}$$

$$\text{e) } g : \mathbb{N}_0 \longrightarrow \mathbb{N}_0 \\ n \longmapsto 2n$$

$$\text{b) } g : \{a, b\}^* \longrightarrow \{a, b\}^* \\ u \longmapsto u^I$$

$$\text{f) } g : \mathbb{N}_0 \longrightarrow \{0, 1, 2\} \\ n \longmapsto r, \text{ onde } n \equiv r \pmod{3}$$

$$\text{c) } g : \mathbb{N}_0 \longrightarrow \mathbb{N}_0 \\ n \longmapsto \begin{cases} 1 & \text{se } n = 3 \\ 0 & \text{senão.} \end{cases}$$

$$\text{g) } g : \mathbb{N}_0 \longrightarrow \mathbb{N}_0 \\ n \longmapsto \begin{cases} \frac{n}{2} & \text{se } n \text{ é par} \\ n.d. & \text{senão} \end{cases}$$

$$\text{d) } g : \mathbb{N}_0 \longrightarrow \mathbb{N}_0 \\ n \longmapsto n + 2$$

$$\text{h) } p_2 : \mathbb{N}_0^3 \longrightarrow \mathbb{N}_0 \\ (n_1, n_2, n_3) \longmapsto n_2$$

1.12 Sejam \mathcal{T}_f , \mathcal{T}_g e \mathcal{T}_h máquinas de Turing que calculam funções $f : \mathbb{N}_0^2 \longrightarrow \mathbb{N}_0$ e $g, h : \mathbb{N}_0 \longrightarrow \mathbb{N}_0$ respetivamente. Mostre que as seguintes funções são ainda computáveis:

$$\text{a) } [\text{função composta}] \\ g \circ h : \mathbb{N}_0 \longrightarrow \mathbb{N}_0 \\ n \longmapsto g(h(n))$$

$$\text{d) } [\text{função troca de variáveis}] \\ t : \mathbb{N}_0^2 \longrightarrow \mathbb{N}_0 \\ (n, m) \longmapsto f(m, n)$$

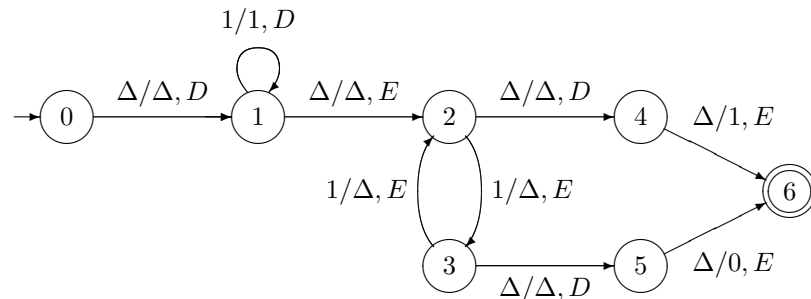
$$\text{b) } [\text{função soma}] \\ g + h : \mathbb{N}_0 \longrightarrow \mathbb{N}_0 \\ n \longmapsto g(n) + h(n)$$

$$\text{e) } [\text{função identificação de variáveis}] \\ i : \mathbb{N}_0 \longrightarrow \mathbb{N}_0 \\ n \longmapsto f(n, n)$$

$$\text{c) } [\text{função mínimo}] \\ \min(g, h) : \mathbb{N}_0 \longrightarrow \mathbb{N}_0 \\ n \longmapsto \min(g(n), h(n))$$

$$\text{f) } [\text{função parametrização da 2ª variável}] \\ f_k : \mathbb{N}_0 \longrightarrow \mathbb{N}_0 \\ n \longmapsto f(n, k), \text{ onde } k \in \mathbb{N}_0$$

1.13 A máquina de Turing \mathcal{T} seguinte, com alfabeto de entrada $A = \{1\}$, calcula a função característica χ_L de uma linguagem L sobre A .



a) Indique as configurações de \mathcal{T} que podem ser computadas a partir de $(0, \underline{\Delta}111)$.

b) Indique, justificando, o valor de $\chi_L(1111)$.

c) Diga qual é a linguagem L . Justifique.

d) Diga, justificando, qual é a linguagem reconhecida por \mathcal{T} .

e) Modifique a máquina \mathcal{T} de forma a obter uma máquina de Turing que reconheça L .

$$\begin{aligned} g: \quad \mathbb{N}_0^2 &\longrightarrow \mathbb{N}_0 \\ (m, n) &\longmapsto 2m + n. \end{aligned}$$

1.19 Considere a máquina de Turing não-determinista

$$\mathcal{T} = (\{q_0, q_1, q_2, q_3\}, \{1\}, \{1, \Delta\}, \delta, q_0, q_3, \Delta)$$

onde a função transição δ é definida pela tabela seguinte:

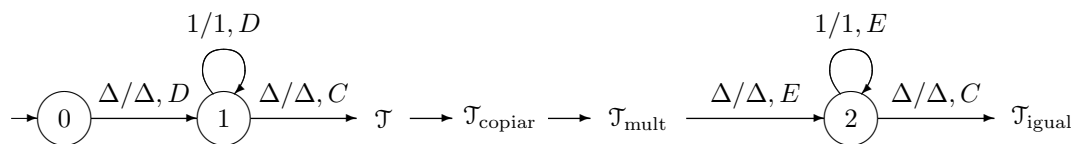
δ	1	Δ
q_0	\emptyset	$\{(q_1, \Delta, D)\}$
q_1	\emptyset	$\{(q_1, 1, D), (q_2, \Delta, E)\}$
q_2	$\{(q_2, 1, E)\}$	$\{(q_3, \Delta, C)\}$

Indique o comportamento de \mathcal{T} a partir da configuração inicial $(q_0, \underline{\Delta}u)$ associada a uma palavra $u \in \{1\}^*$.

1.20 Seja \mathcal{T} a máquina de Turing do exercício anterior e sejam:

- $\mathcal{T}_{\text{copiar}}$ a máquina de Turing capaz de copiar uma palavra, ou seja, de transformar o conteúdo da fita de $\underline{\Delta}u$ em $\underline{\Delta}u\Delta u$;
- $\mathcal{T}_{\text{mult}}$ a máquina de Turing capaz de multiplicar dois números, ou seja, de transformar o conteúdo da fita de $\underline{\Delta}1^m\Delta 1^n$ em $\underline{\Delta}1^{mn}$;
- $\mathcal{T}_{\text{igual}}$ a máquina de Turing capaz de testar a igualdade entre palavras, ou seja, começando com a fita em $\underline{\Delta}u\Delta v$, atinge uma configuração de aceitação se e só se $u = v$.

Considere a seguinte máquina de Turing não-determinista.



Qual é a linguagem que esta máquina de Turing reconhece?

1.21 Seja

$$L = \{1^n : n > 1 \text{ é um natural não primo}\}.$$

Usando a ideia do exercício anterior, construa uma máquina de Turing que reconheça a linguagem L .

1.22 Prove que a linguagem $L = \{wa^n : w \in A^*, n \in \mathbb{N}_0, |w|_b = n\}$ sobre o alfabeto $A = \{a, b\}$ é recursiva.

1.23 Suponha que L_1, \dots, L_k são linguagens recursivamente enumeráveis que formam uma partição de A^* . Mostre que cada L_i é uma linguagem recursiva.

1.24 Esboce uma prova de que, se L_1 e L_2 são linguagens recursivamente enumeráveis, então L_1L_2 e L_1^* são também recursivamente enumeráveis, construindo máquinas de Turing não-deterministas que aceitem estas linguagens.

$$\begin{aligned} & \Delta \underline{x}xyxyxxxxyxyxxxxyyxxxxyxxxxyxxxxyxyxxy \cdots \\ & \Delta xy\underline{x}xyxxxxyxxy\Delta \cdots \\ & \Delta xxxxx\Delta \cdots \end{aligned}$$

2. Problemas de decisão

2.1 Seja $A = \{a, b\}$. Mostre que as seguintes propriedades de palavras $w \in A^*$ são decidíveis.

- a) w tem comprimento ímpar.
- b) ab não é um fator de w .
- c) w tem o mesmo número de ocorrências das letras a e b .

2.2 Indique quais das afirmações seguintes sobre palavras $w \in \{x, y\}^*$ são decidíveis. Indique ainda quais das afirmações indecidíveis são semi-decidíveis.

- a) $w = c(\mathcal{T})$ para alguma máquina de Turing \mathcal{T} e \mathcal{T} aceita w .
- b) $w = c(\mathcal{T})$ para alguma máquina de Turing \mathcal{T} e \mathcal{T} não aceita w .
- c) $w = c(\mathcal{T})$ para alguma máquina de Turing \mathcal{T} .
- d) $w \neq c(\mathcal{T})$ para toda a máquina de Turing \mathcal{T} .

2.3 Sejam P e Q predicados de domínio D , e sejam $\neg P$, $P \wedge Q$ e $P \vee Q$ os predicados definidos, para cada $d \in D$, por:

$$\begin{aligned}(\neg P)(d) &= \neg P(d) \\ (P \wedge Q)(d) &= P(d) \wedge Q(d) \\ (P \vee Q)(d) &= P(d) \vee Q(d).\end{aligned}$$

Mostre que:

- a) se P e Q são decidíveis, então $P \wedge Q$ e $P \vee Q$ são decidíveis;
- b) se P e Q são semi-decidíveis, então $P \wedge Q$ e $P \vee Q$ são semi-decidíveis;
- c) P é decidível se e só se $\neg P$ é decidível;
- d) P é decidível se e só se P e $\neg P$ são semi-decidíveis.
- e) P ser semi-decidível não implica que $\neg P$ seja semi-decidível.

2.4 O objetivo deste exercício é fazer uma redução do problema da aceitação ao problema da paragem. Seja A um alfabeto.

- a) Dada uma máquina de Turing \mathcal{T} , defina uma máquina de Turing \mathcal{T}' tal que, para toda a palavra $w \in A^*$, \mathcal{T} aceita w se e só se \mathcal{T}' pára com w .
- b) Mostre que existe uma máquina de Turing \mathcal{R} que calcula a função $r : c(\mathcal{T}) \mapsto c(\mathcal{T}')$.
- c) Conclua que $Aceitação \leq Paragem$.

2.5 O objetivo deste exercício é fazer outra demonstração da indecidibilidade do problema da paragem.

- a) Mostre que a linguagem $\{w : w = c(\mathcal{T}) \text{ e } \mathcal{T} \text{ não pára com } w, \text{ para alguma MT } \mathcal{T}\}$ não é recursiva.
- b) Conclua que o problema $Q(\mathcal{T})$: “ \mathcal{T} pára com $c(\mathcal{T})$ ” é indecidível.
- c) Reduza Q ao problema da paragem e conclua que o problema da paragem é indecidível.

2.6 Das afirmações seguintes selecione as que são verdadeiras, sejam quais forem os problemas de decisão P_1 , P_2 , Q_1 e Q_2 tais que $P_1 \leq Q_1$, $Q_1 \leq P_2$ e $Q_2 \leq P_2$.

- a) Se P_2 é decidível, então P_1 , Q_1 e Q_2 são também decidíveis.
- b) Se P_1 é indecidível, então P_2 , Q_1 e Q_2 são também indecidíveis.
- c) $P_1 \leq P_2$.
- d) Se Q_1 é semi-decidível, então P_1 e P_2 são também semi-decidíveis.

2.7 Considere os problemas de decisão

- $Pára_\epsilon$: dada uma máquina de Turing \mathcal{T} , será que \mathcal{T} pára com ϵ ?
- $Pára_{ab}$: dada uma máquina de Turing \mathcal{T} , será que \mathcal{T} pára com a palavra ab ?

- a) Mostre que $Pára_\epsilon \leq Pára_{ab}$.
- b) Conclua que o problema $Pára_{ab}$ é indecidível.

2.8 Considere o problema *Equivalência*: dadas máquinas de Turing \mathcal{T}_1 e \mathcal{T}_2 , será que \mathcal{T}_1 e \mathcal{T}_2 reconhecem a mesma linguagem?

- a) Mostre que o problema *AceitaTudo* se reduz a *Equivalência*.
- b) Conclua que o problema da equivalência é indecidível.

2.9 Mostre que os seguintes problemas de decisão são indecidíveis.

- a) Dadas máquinas de Turing \mathcal{T}_1 e \mathcal{T}_2 , será que $L(\mathcal{T}_1) \subseteq L(\mathcal{T}_2)$?
- b) Dadas máquinas de Turing \mathcal{T}_1 e \mathcal{T}_2 , será que $L(\mathcal{T}_1) \cap L(\mathcal{T}_2) = \emptyset$?
- c) Dada uma máquina de Turing \mathcal{T} e um estado não final q , será que \mathcal{T} atinge o estado q quando iniciada com a fita vazia?

2.10 Porque é que o seguinte argumento é incorreto?

O problema da aceitação da palavra vazia é um subproblema do problema da aceitação, que é indecidível, e portanto é ele próprio indecidível.

2.11 Mostre que as seguintes propriedades de máquinas de Turing \mathcal{T} são decidíveis.

- a) O estado inicial de \mathcal{T} é q_6 .
- b) \mathcal{T} tem 4 estados.
- c) O símbolo s_{12} pertence ao alfabeto da fita de \mathcal{T} .
- d) $\delta(q_3, \Delta) = (q_5, \Delta, E)$ é uma transição de \mathcal{T} .
- e) O código de \mathcal{T} é w_0 , onde $w_0 \in \{x, y\}^*$ é uma palavra fixa.

2.12 Mostre que as seguintes propriedades de linguagens recursivamente enumeráveis L são indecidíveis.

- a) L contém w_0 , onde w_0 é uma palavra fixa.
- b) L é regular.
- c) L é finita.
- d) $L \neq \{a, b\}^*$.

2.13 Dê exemplo de uma propriedade de linguagens que nenhuma linguagem recursivamente enumerável verifica.

3. Funções parciais recursivas

3.1 Determine a função $\text{Rec}(f, g)$ definida recursivamente pelas funções $f : \mathbb{N}_0 \rightarrow \mathbb{N}_0$ e $g : \mathbb{N}_0^3 \rightarrow \mathbb{N}_0$ tais que:

- a) $f(x) = x$ e $g(x, y, z) = z + 2$.
- b) $f(x) = x$ e $g(x, y, z) = (y + 1)z$.

3.2 Seja $h : \mathbb{N}_0^3 \rightarrow \mathbb{N}_0$ a função definida, para cada $(x, y, z) \in \mathbb{N}_0^3$, por $h(x, y, z) = x + yz + 1$.

- a) Defina recursivamente a função h . Ou seja, determine funções $f : \mathbb{N}_0^2 \rightarrow \mathbb{N}_0$ e $g : \mathbb{N}_0^4 \rightarrow \mathbb{N}_0$ tais que $h = \text{Rec}(f, g)$.
- b) Mostre que h é uma função recursiva primitiva.

3.3 Sejam $qd : \mathbb{N}_0 \rightarrow \mathbb{N}_0$ e $h : \mathbb{N}_0^2 \rightarrow \mathbb{N}_0$ funções definidas por $qd(x) = x^2$ e $h(x, y) = (x + y)^2$.

- a) Determine funções f e g tais que $qd = \text{Rec}(f, g)$.
- b) Mostre que a função h é recursiva primitiva.

3.4 Mostre que as seguintes funções são recursivas primitivas:

- a) $\text{mult}(x, y) = x \cdot y$.
- b) $\text{exp}(x, y) = x^y$.
- c) $\text{fat}(x) = \begin{cases} 1 & \text{se } x = 0 \\ x \cdot (x - 1) \cdot \dots \cdot 2 \cdot 1 & \text{se } x > 0. \end{cases}$
- d) $\text{ad}^{(k)}(x_1, \dots, x_k) = x_1 + \dots + x_k$.
- e) $h(x, y) = 3x + 2^y + 5$.

3.5 Seja $f : \mathbb{N}_0^2 \rightarrow \mathbb{N}_0$ uma função recursiva primitiva. Mostre que as seguintes funções são recursivas primitivas:

- a) $g : \mathbb{N}_0^{k+2} \rightarrow \mathbb{N}_0$ definida por $g(x, y, z_1, \dots, z_k) = f(x, y)$. *[adição de variáveis]*
- b) $g : \mathbb{N}_0^2 \rightarrow \mathbb{N}_0$ definida por $g(y, x) = f(x, y)$. *[troca de variáveis]*
- c) $g : \mathbb{N}_0 \rightarrow \mathbb{N}_0$ definida por $g(x) = f(x, x)$. *[identificação de variáveis]*

3.6 Seja $A : \mathbb{N}_0^2 \rightarrow \mathbb{N}_0$ a função de Ackermann.

- a) Determine $A(3, 0)$ e $A(2, 2)$.
- b) Prove que $A(1, y) = y + 2$ para todo o $y \in \mathbb{N}_0$.
- c) Prove que $A(2, y) = 2y + 3$ para todo o $y \in \mathbb{N}_0$.

3.7 Determine a função M_f de minimização da função $f : \mathbb{N}_0^2 \rightarrow \mathbb{N}_0$ tal que:

- a) $f(x, y) = (x + y)^2$.
- b) $f(x, y) = \begin{cases} 0 & \text{se } x^2 = y + 9 \\ 1 & \text{senão.} \end{cases}$

3.8 Mostre, sem construir máquinas de Turing, que as seguintes funções são computáveis:

- a) $f(x) = \begin{cases} \sqrt{x} & \text{se } x \text{ é um quadrado perfeito} \\ n.d. & \text{senão.} \end{cases}$
- b) $f(x) = \begin{cases} \frac{x}{2} & \text{se } x \text{ é par} \\ n.d. & \text{senão.} \end{cases}$

4. Introdução à Teoria da Complexidade

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

- a) n^2 é $\mathcal{O}(n^4)$.
- b) $2n^3 + n^2 + 7n + 3$ é $\mathcal{O}(n^3)$.
- c) n^4 é $\mathcal{O}(n^2)$.

4.2 Considere as relações definidas, para funções $f, g : \mathbb{N}_0 \rightarrow \mathbb{R}$, por:

$$f(n)R_1 g(n) \text{ se e só se } f(n) \text{ é } \mathcal{O}(g(n))$$

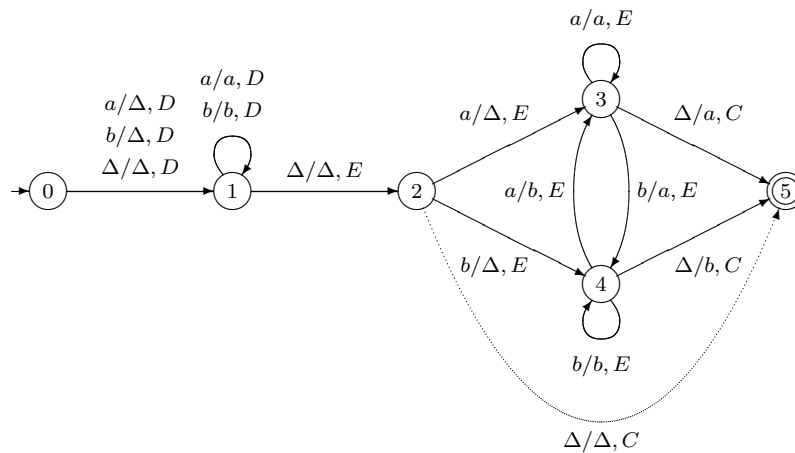
$$f(n)R_2 g(n) \text{ se e só se } f(n) \text{ é } \mathcal{O}(g(n)) \text{ e } g(n) \text{ não é } \mathcal{O}(f(n))$$

$$f(n)R_3 g(n) \text{ se e só se } f(n) \text{ é } \mathcal{O}(g(n)) \text{ e } g(n) \text{ é } \mathcal{O}(f(n)).$$

Mostre que:

- a) R_1 é reflexiva e transitiva;
- b) R_2 é transitiva e assimétrica (i.e., se $f(n)R_2 g(n)$, então $g(n) \not R_2 f(n)$);
- c) R_3 é uma relação de equivalência.

4.3 Determine a função de complexidade temporal da máquina $\mathcal{T}_{\text{apagar}}$, ou seja, da seguinte máquina de Turing



4.4 Seja $L = \{uu : u \in \{a, b\}^*\}$. Mostre que $L \in DTIME(n)$ e que $L \in DSPACE(n)$.

4.5 Mostre que a classe das linguagens regulares é uma subclasse de $DSPACE(1)$.

4.6 Em cada uma das alíneas seguintes, mostre que $L_1 \leq_p L_2$ e $L_2 \leq_p L_1$.

- a) $L_1 = \{a^n b^n : n \geq 0\}$ e $L_2 = \{a^{n+1} b^n : n \geq 0\}$;
- b) $L_1 = \{u \in A^* : u = u^I\}$ e $L_2 = \{uu : u \in A^*\}$, onde A é um alfabeto não singular qualquer.

4.7 Considere linguagens $L_1 \subseteq A_1^*$, $L_2 \subseteq A_2^*$ e $L_3 \subseteq A_3^*$. Mostre que:

- a) Se $L_1 \leq_p L_2$ e $L_2 \leq_p L_3$, então $L_1 \leq_p L_3$;
- b) Se $L_1 \leq_p L_2$ e $L_2 \in P$, então $L_1 \in P$;
- c) Se $L_1 \leq_p L_2$ e L_1 é NP -completa, então L_2 é NP -difícil.