4.2 Considere o problema de se testar se um AFD e uma expressão regular são equivalentes. Expresse esse problema como uma linguagem e mostre que ela é decidível.

Let $EQ_{\mathsf{DFA,REX}} = \{\langle A, R\rangle \mid A$ is a DFA, R is a regular expression and $L(A) = L(R)\}$. The following TM E decides $EQ_{\mathsf{DFA,REX}}$.

E = "On input $\langle A, R\rangle$:

1. Convert regular expression R to an equivalent DFA B using the procedure given in Theorem 1.28.
2. Use the TM C for deciding EQ_{DFA} given in Theorem 4.5, on input $\langle A, B\rangle$.

3. If R accepts, *accept*. If R rejects, *reject*."

4.3 Seja $TOD_{\mathsf{AFD}} = \{\langle A\rangle \mid A$ é um AFD que reconhece $\Sigma^*\}$. Mostre que TOD_{AFD} é decidível.

Let $ALL_{\mathsf{DFA}} = \{\langle A\rangle \mid A$ is a DFA that recognizes $\Sigma^*\}$. The following TM L decides ALL_{DFA}.

L = "On input $\langle A\rangle$ where A is a DFA:

1. Construct DFA B that recognizes $\overline{L(A)}$ as described in Exercise 1.10.
2. Run TM T from Theorem 4.4 on input $\langle B\rangle$, where T decides E_{DFA}.
3. If T accepts, *accept*. If T rejects, *reject*."

4.4 $A\varepsilon_{\mathsf{GLC}} = \{\langle G\rangle \mid G$ é uma GLC que gera $\varepsilon\}$. Mostre que $A\varepsilon_{\mathsf{GLC}}$ é decidível.

Let $A\varepsilon_{\mathsf{CFG}} = \{\langle G\rangle \mid G$ is a CFG that generates $\varepsilon\}$. The following TM V decides $A\varepsilon_{\mathsf{CFG}}$.

V = "On input $\langle G\rangle$ where G is a CFG:

1. Run TM S from Theorem 4.6 on input $\langle G, \varepsilon\rangle$, where S is a decider for A_{CFG}.
2. If S accepts, *accept*. If S rejects, *reject*."

4.5 Seja X o conjunto $\{1, 2, 3, 4, 5\}$ e Y o conjunto $\{6, 7, 8, 9, 10\}$. Descrevemos as funções $f\colon X \longrightarrow Y$ e $g\colon X \longrightarrow Y$ nas tabelas abaixo. Responda a cada item e dê uma razão para cada resposta negativa.

n	$f(n)$
1	6
2	7
3	6
4	7
5	6

n	$g(n)$
1	10
2	9
3	8
4	7
5	6

a. f é um-para-um?
b. f é sobrejetora?
c. f é uma correspondência?
d. g é um-para-um?
e. g é sobrejetora?
f. g é uma correspondência?

a. g is one-to-one. f is not one-to-one because $f(1) = f(3)$.
b. g is onto. f is not onto because there does not exist $x \in X$ such that $f(x) = 10$.
c. g is a correspondence because g is one-to-one and onto. f is not a correspondence because f is not one-to-one and onto.

4.6

Seja $\mathcal{B}$ o conjunto de todas as seqüências infinitas sobre $\{0, 1\}$. Mostre que $\mathcal{B}$ é incontável, usando uma prova por diagonalização.

Suppose $\mathcal{B}$ is countable and a correspondence $f\colon \mathcal{N} \longrightarrow \mathcal{B}$ exists. We construct x in $\mathcal{B}$ that is not paired with anything in $\mathcal{N}$. Let $x = x_1x_2\ldots$. Let $x_i = 0$ if $f(i)_i = 1$, and $x_i = 1$ if $f(i)_i = 0$ where $f(i)_i$ is the ith bit of $f(i)$. Therefore, we ensure that x is not $f(i)$ for any i because it differs from $f(i)$ in the ith symbol, and a contradiction occurs.

4.9

Seja $\mathrm{INFINITA}_{\mathrm{AFD}} = \{$ <A> | A é uma AFD e L(A) é uma linguagem infinita$\}$. Mostre que $\mathrm{INFINITA}_{\mathrm{AFD}}$ é decidível.

Let $INFINITE_{\mathsf{DFA}} = \{\langle A\rangle \mid L(A)$ is an infinite language$\}$. The following TM I decides $INFINITE_{\mathsf{DFA}}$.

I = "On input $\langle A\rangle$ where A is a DFA:

1. Let k be the number of states of A.
2. Construct a DFA D that accepts strings of length $\geq k$.
3. Construct a DFA M such that $L(M) = L(A) \cap L(D)$.
4. Run TM T from Theorem 4.4 on input $\langle M\rangle$, where T decides E_{DFA}.
5. If T accepts, *reject*. If T rejects, *accept*."

If A accepts a string with length at least k, this string can be must cause A to enter the same state at least twice, so it can be pumped up to obtain infinitely many strings that are accepted by A. Conversely, if A accepts infinitely many strings, it must accept strings of arbitrarily long lengths, in particular, a string of length at least k.

4.11

Seja A = {<M> | M é um AFD que não aceita nenhuma cadeia contendo um número ímpar de 1s}. Mostre que A é decidível.

The following TM X decides A.

X = "On input $\langle M\rangle$ where M is a DFA:

1. Construct a DFA O that accepts any string containing an odd number of 1s.
2. Construct DFA B such that $L(B) = L(M) \cap L(O)$.
3. Run TM T from Theorem 4.4 on input $\langle B\rangle$, where T decides E_{DFA}.
4. If T accepts, *accept*. If T rejects, *reject*."

4.12

Seja A = {<R,S> | R e S são expressões regulares e $L(R) \subseteq L(S)$}. Mostre que A é decidível.

We observe that $L(R) \subseteq L(S)$ if and only if $\overline{L(S)} \cap L(R) = \emptyset$. The following TM X decides A.

X = "On input $\langle R, S\rangle$ where R and S are regular expressions:

1. Construct DFA E such that $L(E) = \overline{L(S)} \cap L(R)$.
2. Run TM T from Theorem 4.4 on input $\langle E\rangle$, where T decides E_{DFA}.
3. If T accepts, *accept*. If T rejects, *reject*."

4.13 Seja $\Sigma = \{0,1\}$. Mostre que o problema de se determinar se uma GLC gera alguma cadeia em 1^* é decidível. Em outras palavras, mostre que

Let $A = \{v \mid v$ is of the form $ww^{\mathcal{R}}$ for $w \in \{0,1\}^*\}$. A is a CFL. Problem 2.17 showed that $B \cap A$ is context free if B is regular. The following TM X decides E.

X = "On input $\langle M\rangle$ where M is a DFA:

1. Let $C = L(M) \cap A$. Let G be the CFG of C.
2. Run TM R from Theorem 4.7 on input $\langle G\rangle$, where R decides E_{CFG}.
3. If R accepts, *reject*. If R rejects, *accept*."

X decides E because a DFA M accepts some string of the form $ww^{\mathcal{R}}$ if and only if $L(M) \cap A \neq \emptyset$.

4.15 Seja $A = \{\langle R\rangle \mid R$ é uma expressão regular que descreve uma linguagem contendo pelo menos uma cadeia w que tem 111 como uma subcadeia (isto é, $w = x111y$ para alguma x e alguma y)$\}$. Mostre que A é decidível.

The following TM X decides A.

X = "On input $\langle R\rangle$ where R is a regular expression:

1. Construct DFA E that accepts $\Sigma^*111\Sigma^*$.
2. Construct DFA B such that $L(B) = L(R) \cap L(E)$.
3. Run TM T from Theorem 4.4 on input $\langle B\rangle$, where T decides E_{DFA}.
4. If T accepts, *reject*. If T rejects, *accept*."

4.22 Um *estado inútil* em um autômato com pilha nunca é atingido sobre qualquer cadeia de entrada. Considere o problema de se determinar se um autômato com pilha tem quaisquer estados inúteis. Formule esse problema como uma linguagem e mostre que ele é decidível.

Let $U = \{\langle P\rangle \mid P$ is a PDA that has useless states$\}$. The following TM T decides U.

T = "On input $\langle P\rangle$, where U is a PDA:
1. **For each state q of P:**
2. **Modify P so that q is the only accept state.**
3. **Use the decider for E_{PDA} to test whether the modified PDA's language is empty. If it is, *accept*. If it is not, continue.**
4. **At this point, all states have been shown to be useful, so *reject*."**

5.1 Mostre que EQ_{GLC} é indecidível.

Suppose for a contradiction that EQ_{CFG} were decidable. We construct a decider M for $ALL_{\mathsf{CFG}} = \{\langle G\rangle \mid G$ is a CFG and $L(G) = \Sigma^*\}$ as follows:

M = "On input $\langle G\rangle$:
1. **Construct a CFG H such that $L(H) = \Sigma^*$.**
2. **Run the decider for EQ_{CFG} on $\langle G, H\rangle$.**
3. **If it accepts, *accept*. If it rejects, *reject*."**

M decides ALL_{CFG} assuming a decider for EQ_{CFG} exists. Since we know ALL_{CFG} is undecidable, we have a contradiction.

5.2 Mostre que EQ_{GLC} é co-Turing-reconhecível.

Here is a Turing Machine M which recognizes the complement of EQ_{CFG}:

M = "On input $\langle G, H\rangle$:
1. **Lexicographically generate the strings $x \in \Sigma^*$.**
2. **For each such string x:**
3. **Test whether $x \in L(G)$ and whether $x \in L(H)$, using the algorithm for A_{CFG}.**
4. **If one of the tests accepts and the other rejects, *accept*; otherwise, continue."**

5.4 Se $A \leq_m B$ e B é uma linguagem regular, isso implica que A seja uma linguagem regular? Por que ou por que não?

No, it does not imply that A is regular. For example, $\{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n \mid n \geq 0\} \leq_m \{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}$. The reduction first tests whether its input is a member of $\{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n \mid n \geq 0\}$. If so, it outputs the string ab, and if not, it outputs the string a.

[R]5.5 Mostre que A_{MT} não é redutível por mapeamento a V_{MT}. Em outras palavras, mostre que nenhuma função computável reduz A_{MT} a V_{MT}. (Dica: Use uma prova por contradição e fatos que você já conhece sobre A_{MT} e V_{MT}.)

Suppose for a contradiction that $A_{\mathsf{TM}} \leq_m E_{\mathsf{TM}}$ via reduction f. It follows from the definition of mapping reducibility that $\overline{A_{\mathsf{TM}}} \leq_m \overline{E_{\mathsf{TM}}}$ via the same reduction function f. Observe that $\overline{E_{\mathsf{TM}}}$ is Turing-recognizable, but, $\overline{A_{\mathsf{TM}}}$ is not Turing-recognizable, contradicting Theorem 5.22.

[R]5.6 Mostre que $\leq_m$ é uma relação transitiva.

5.6 Suponha que $A \leq_m B$ e $B \leq_m C$. Então, existem funções computáveis f e g tais que $x \in A \iff f(x) \in B$ e $y \in B \iff g(y) \in C$. Considere a função composta $h(x) = g(f(x))$. Podemos construir uma MT que computa h como segue: primeiro, simule uma MT para f (tal MT existe porque supusemos que f é computável) sobre a entrada x e chame a saída de y. Então, simule uma MT para g sobre y. A saída é $h(x) = g(f(x))$. Portanto, h é uma função computável. Além disso, $x \in A \iff h(x) \in C$. Logo, $A \leq_m C$ via a função de redução h.

[R]5.7 Mostre que se A é Turing-reconhecível e $A \leq_m \overline{A}$, então A é decidível.

Suppose $A \leq_m \overline{A}$. Then, $\overline{A} \leq_m A$, by the definition of mapping reducibility. Because A is Turing-recognizable, Theorem 5.22 implies that $\overline{A}$ is Turing-recognizable, and then Theorem 4.16 implies that A is decidable.

[R]5.8 Na prova do Teorema 5.15 modificamos a máquina de Turing M de modo que ela nunca tente mover sua cabeça além da extremidade esquerda da fita. Suponha que não fizéssemos essa modificação a M. Modifique a construção do PCP para lidar com esse caso.

5.8 Você tem que lidar com o caso em que a cabeça está na célula da extremidade esquerda da fita e tenta mover para a esquerda. Para fazer isso, acrescente dominós

$$\left[\frac{\#qa}{\#rb}\right]$$

5.9 Seja $T = \{\langle M\rangle \mid M$ é uma MT que aceita $w^{\mathcal{R}}$ sempre que ela aceita $w\}$. Mostre que T é indecidível.

A seguir, perguntas 5.10, 5.11, 5.28 e 5.30:

5.10 Considere o problema de se determinar se uma máquina de Turing de duas fitas em algum momento escreve um símbolo não-branco sobre sua segunda fita quando ela é executada sobre a entrada w. Formule esse problema como uma linguagem e mostre que ele é indecidível.

5.11 Considere o problema de se determinar se uma máquina de Turing de duas fitas em algum momento escreve um símbolo não-branco sobre sua segunda fita durante o curso de sua computação sobre qualquer cadeia de entrada. Formule esse problema como uma linguagem e mostre que ele é indecidível.

5.28 **Teorema de Rice.** Seja P qualquer propriedade não-trivial da linguagem de uma máquina de Turing. Prove que o problema de determinar se a linguagem de uma dada máquina de Turing tem a propriedade P é indecidível.

Em termos mais formais, seja P uma linguagem constituída de descrições de máquinas de Turing, em que P satisfaz duas condições. Primeiro, P é não-trivial — ela contém alguma descrição, mas não todas as descrições de MTs. Segundo, P é uma propriedade da linguagem da MT — quando $L(M_1) = L(M_2)$, temos que $\langle M_1\rangle \in P$ sse $\langle M_2\rangle \in P$. No caso, M_1 e M_2 são quaisquer MTs. Prove que P é uma linguagem indecidível.

5.30 Use o teorema de Rice, que aparece no Problema 5.28, para provar a indecidibilidade de cada uma das seguintes linguagens.

[R]a. $INFINITA_{\mathsf{TM}} = \{\langle M\rangle \mid M$ é uma MT e $L(M)$ é uma linguagem infinita$\}$.
b. $\{\langle M\rangle \mid M$ é uma MT e $1011 \in L(M)\}$.
c. $TUDO_{\mathsf{TM}} = \{\langle M\rangle \mid M$ é uma MT e $L(M) = \Sigma^*\}$.

A seguir, respostas 5.10, 5.11, 5.28 e 5.30:

5.10 Seja $B = \{\langle M, w\rangle \mid M$ é uma MT de duas fitas que escreve um símbolo não-branco na sua segunda fita quando roda $w\}$. Mostre que A_{MT} se reduz a B. Suponha, para o propósito de obter uma contradição, que a MT R decide B. Então, construa a MT S que usa R para decidir A_{MT}.

S = "Sobre a entrada $\langle M, w\rangle$:
1. Use M para construir a seguinte MT de duas fitas, T.
   T = "Sobre a entrada x:
   1. Simule M sobre x usando a primeira fita.
   2. Se a simulação indicar que M aceita, escreva um símbolo não-branco na segunda fita."
2. Execute R sobre $\langle T, w\rangle$ para determinar se T sobre a entrada w escreve um símbolo não-branco na sua segunda fita.
3. Se R aceita, M aceita w; portanto, *aceite*. Caso contrário, *rejeite*."

5.11 Seja $C = \{\langle M\rangle \mid M$ é uma MT de duas fitas que escreve um símbolo não-branco na sua segunda fita quando roda sobre alguma entrada$\}$. Mostre que A_{MT} se reduz a C. Suponha, para o propósito de obter uma contradição, que a MT R decide C. Então, construa a MT S que usa R para decidir A_{MT}.

S = "Sobre a entrada $\langle M, w\rangle$:
1. Use M e w para construir a seguinte MT de duas fitas, T_w.
   T_w = "Sobre qualquer entrada:
   1. Simule M sobre w usando a primeira fita.
   2. Se a simulação indicar que M aceita, escreva um símbolo não-branco na segunda fita."
2. Execute R sobre $\langle T_w\rangle$ para determinar se T_w, em algum momento, escreve um símbolo não-branco na sua segunda fita.
3. Se R aceita, M aceita w; portanto, *aceite*. Caso contrário, *rejeite*."

5.28 Suponha, para o propósito de obter uma contradição, que P é uma linguagem decidível satisfazendo as propriedades, e R_P uma MT que decide P. Mostramos como decidir A_{MT} usando R_P através da construção da MT S. Primeiro, seja $T_\emptyset$ uma MT que sempre rejeite; assim, $L(T_\emptyset) = \emptyset$. Você pode supor que $\langle T_\emptyset\rangle \notin P$ sem perda de generalidade, visto que pode prosseguir com $\overline{P}$ em vez de P se $\langle T_\emptyset\rangle \in P$. Como P não é trivial, existe uma MT T tal que $\langle T\rangle \in P$. Conceba S para decidir A_{MT} usando a capacidade de R_P de distinguir entre $T_\emptyset$ e T.

$S =$ "Sobre a entrada $\langle M, w \rangle$:
1. Use M e w para construir a seguinte MT M_w.
   $M_w =$ "Sobre a entrada x:
   1. Simule M sobre w. Se ela pára e rejeita, *rejeite*. Se ela aceita, prossiga no estágio 2.
   2. Simule T sobre x. Se ela aceita, *aceite*."
2. Use a MT R_P para determinar se $\langle M_w \rangle \in P$. Se SIM, *aceite*. Se NÃO, *rejeite*."

A MT M_w simula T se M aceita w. Logo, $L(M_w)$ é igual a $L(T)$ se M aceita w e é igual a $\emptyset$, em caso contrário. Portanto, $\langle M, w \rangle \in P$ sse M aceita w.

5.30 (a) $INFINITA_{TM}$ é uma linguagem de descrições de MTs. Ela satisfaz as duas condições do teorema de Rice. Primeiro, ela é não-trivial porque algumas MTs têm linguagens infinitas e outras não têm. Segundo, ela depende apenas da linguagem. Se duas MTs reconhecem a mesma linguagem, então ou ambas têm descrições em $INFINITA_{TM}$ ou nenhuma delas tem. Conseqüentemente, o teorema de Rice implica que $INFINITA_{TM}$ é indecidível.

$F =$ "Sobre a entrada $\langle A, B \rangle$, onde A e B são AFDs:
1. Construa o AFD C conforme descrito.
2. Rode a MT T do Teorema 4.4 sobre a entrada $\langle C \rangle$.
3. Se T aceita, *aceite*. Se T rejeita, *rejeite*."

TEOREMA 5.28

Se $A \leq_m B$ e B é Turing-reconhecível, então A é Turing-reconhecível.

A prova é a mesma que aquela do Teorema 5.22, exceto que M e N são reconhecedores em vez de decisores.

TEOREMA 5.22

Se $A \leq_m B$ e B é decidível, então A é decidível.

MT que decide um AFD (4.4)

$M =$ "Sobre a entrada $\langle B, w \rangle$, onde B é um AFD, e w, uma cadeia:
1. Simule B sobre a entrada w.
2. Se a simulação termina em um estado de aceitação, *aceite*. Se ela termina em um estado de não-aceitação, *rejeite*."

MT que decide uma AFN

$N =$ "Sobre a entrada $\langle B, w \rangle$ onde B é um AFN, e w, uma cadeia:
1. Converta AFN B para um AFD equivalente C, usando o procedimento para essa conversão dado no Teorema 1.39.
2. Rode a MT M do Teorema 4.1 sobre a entrada $\langle C, w \rangle$.
3. Se M aceita, *aceite*; caso contrário, *rejeite*."

EXR é uma linguagem decídivel

$P =$ "Sobre a entrada $\langle R, w \rangle$ onde R é uma expressão regular e w é uma cadeia:
1. Converta a expressão regular R para um AFN equivalente A usando o procedimento para essa conversão dado no Teorema 1.54.
2. Rode a MT N sobre a entrada $\langle A, w \rangle$.
3. Se N aceita, *aceite*; se N rejeita, *rejeite*."

VAFD (vacuidade) é decidivel

$T =$ "Sobre a entrada $\langle A \rangle$ onde A é um AFD:
1. Marque o estado inicial de A.
2. Repita até que nenhum estado novo venha a ser marcado:
3. Marque qualquer estado que tenha uma transição chegando nele a partir de qualquer estado que já está marcado.
4. Se nenhum estado de aceitação estiver marcado, *aceite*; caso contrário, *rejeite*."

MT que decide uma GLC (4.6)

PROVA A MT S para A_{GLC} segue.

$S =$ "Sobre a entrada $\langle G, w \rangle$, onde G é uma GLC, e w, uma cadeia:
1. Converta G para uma gramática equivalente na forma normal de Chomsky.
2. Liste todas as derivações com $2n - 1$ passos, onde n é o comprimento de w, exceto se $n = 0$; nesse último caso, liste todas as derivações com 1 passo.
3. Se alguma dessas derivações gera w, *aceite*; se não, *rejeite*."

EQ AFD é uma linguagem decidível (4.5)