4.15 Let $A = \{\langle R \rangle |\ R$ is a regular expression describing a language containing at least one string w that has `111` as a substring (i.e., $w = x\texttt{111}y$ for some x and y)$\}$. Show that A is decidable.

4.16 Prove that EQ_{DFA} is decidable by testing the two DFAs on all strings up to a certain size. Calculate a size that works.

***4.17** Let C be a language. Prove that C is Turing-recognizable iff a decidable language D exists such that $C = \{x |\ \exists y\ (\langle x, y \rangle \in D)\}$.

4.18 Let A and B be two disjoint languages. Say that language C ***separates*** A and B if $A \subseteq C$ and $B \subseteq \overline{C}$. Show that any two disjoint co-Turing-recognizable languages are separable by some decidable language.

4.19 Let $S = \{\langle M \rangle |\ M$ is a DFA that accepts $w^{\mathcal{R}}$ whenever it accepts $w\}$. Show that S is decidable.

4.20 A language is ***prefix-free*** if no member is a proper prefix of another member. Let $PREFIX\text{-}FREE_{\text{REX}} = \{R |\ R$ is a regular expression where $L(R)$ is prefix-free$\}$. Show that $PREFIX\text{-}FREE_{\text{REX}}$ is decidable. Why does a similar approach fail to show that $PREFIX\text{-}FREE_{\text{CFG}}$ is decidable?

A*4.21 Say that an NFA is ***ambiguous*** if it accepts some string along two different computation branches. Let $AMBIG_{\text{NFA}} = \{\langle N \rangle |\ N$ is an ambiguous NFA$\}$. Show that $AMBIG_{\text{NFA}}$ is decidable. (Suggestion: One elegant way to solve this problem is to construct a suitable DFA and then run E_{DFA} on it.)

4.22 A ***useless state*** in a pushdown automaton is never entered on any input string. Consider the problem of determining whether a pushdown automaton has any useless states. Formulate this problem as a language and show that it is decidable.

A*4.23 Let $BAL_{\text{DFA}} = \{\langle M \rangle |\ M$ is a DFA that accepts some string containing an equal number of 0s and 1s$\}$. Show that BAL_{DFA} is decidable. (Hint: Theorems about CFLs are helpful here.)

***4.24** Let $PAL_{\text{DFA}} = \{\langle M \rangle |\ M$ is a DFA that accepts some palindrome$\}$. Show that PAL_{DFA} is decidable. (Hint: Theorems about CFLs are helpful here.)

***4.25** Let $E = \{\langle M \rangle |\ M$ is a DFA that accepts some string with more 1s than 0s$\}$. Show that E is decidable. (Hint: Theorems about CFLs are helpful here.)

4.26 Let $C = \{\langle G, x \rangle |\ G$ is a CFG that generates some string w, where x is a substring of $w\}$. Show that C is decidable. (Suggestion: An elegant solution to this problem uses the decider for E_{CFG}.)

4.27 Let $C_{\text{CFG}} = \{\langle G, k \rangle |\ L(G)$ contains exactly k strings where $k \geq 0$ or $k = \infty\}$. Show that C_{CFG} is decidable.

4.28 Let A be a Turing-recognizable language consisting of descriptions of Turing machines, $\{\langle M_1 \rangle, \langle M_2 \rangle, \ldots\}$, where every M_i is a decider. Prove that some decidable language D is not decided by any decider M_i whose description appears in A. (Hint: You may find it helpful to consider an enumerator for A.)

5.14 Consider the problem of determining whether a Turing machine M on an input w ever attempts to move its head left when its head is on the left-most tape cell. Formulate this problem as a language and show that it is undecidable.

5.15 Consider the problem of determining whether a Turing machine M on an input w ever attempts to move its head left at any point during its computation on w. Formulate this problem as a language and show that it *is* decidable.

5.16 Let $\Gamma = \{\texttt{0}, \texttt{1}, \sqcup\}$ be the tape alphabet for all TMs in this problem. Define the ***busy beaver function*** $BB\colon \mathcal{N} \longrightarrow \mathcal{N}$ as follows. For each value of k, consider all k-state TMs that halt when started with a blank tape. Let $BB(k)$ be the maximum number of 1s that remain on the tape among all of these machines. Show that BB is not a computable function.

5.17 Show that the Post Correspondence Problem is decidable over the unary alphabet $\Sigma = \{\texttt{1}\}$.

5.18 Show that the Post Correspondence Problem is undecidable over the binary alphabet $\Sigma = \{\texttt{0,1}\}$.

5.19 In the *silly Post Correspondence Problem*, *SPCP*, in each pair the top string has the same length as the bottom string. Show that the *SPCP* is decidable.

5.20 Prove that there exists an undecidable subset of $\{\texttt{1}\}^*$.

5.21 Let $AMBIG_{\text{CFG}} = \{\langle G\rangle \mid G \text{ is an ambiguous CFG}\}$. Show that $AMBIG_{\text{CFG}}$ is undecidable. (Hint: Use a reduction from *PCP*. Given an instance

$$P = \left\{ \left[\frac{t_1}{b_1}\right], \left[\frac{t_2}{b_2}\right], \ \ldots\ , \left[\frac{t_k}{b_k}\right] \right\},$$

of the Post Correspondence Problem, construct a CFG G with the rules

$$\begin{aligned} S &\to T \mid B \\ T &\to t_1 T \texttt{a}_1 \mid \cdots \mid t_k T \texttt{a}_k \mid t_1 \texttt{a}_1 \mid \cdots \mid t_k \texttt{a}_k \\ B &\to b_1 B \texttt{a}_1 \mid \cdots \mid b_k B \texttt{a}_k \mid b_1 \texttt{a}_1 \mid \cdots \mid b_k \texttt{a}_k\ , \end{aligned}$$

where $\texttt{a}_1, \ldots, \texttt{a}_k$ are new terminal symbols. Prove that this reduction works.)

5.22 Show that A is Turing-recognizable iff $A \leq_{\text{m}} A_{\text{TM}}$.

5.23 Show that A is decidable iff $A \leq_{\text{m}} \texttt{0}^*\texttt{1}^*$.

5.24 Let $J = \{w \mid \text{either } w = \texttt{0}x \text{ for some } x \in A_{\text{TM}}, \text{ or } w = \texttt{1}y \text{ for some } y \in \overline{A_{\text{TM}}}\ \}$. Show that neither J nor $\overline{J}$ is Turing-recognizable.

5.25 Give an example of an undecidable language B, where $B \leq_{\text{m}} \overline{B}$.

5.26 Define a ***two-headed finite automaton*** (2DFA) to be a deterministic finite automaton that has two read-only, bidirectional heads that start at the left-hand end of the input tape and can be independently controlled to move in either direction. The tape of a 2DFA is finite and is just large enough to contain the input plus two additional blank tape cells, one on the left-hand end and one on the right-hand end, that serve as delimiters. A 2DFA accepts its input by entering a special accept state. For example, a 2DFA can recognize the language $\{\texttt{a}^n\texttt{b}^n\texttt{c}^n \mid n \geq 0\}$.

a. Let $A_{\text{2DFA}} = \{\langle M, x\rangle \mid M \text{ is a 2DFA and } M \text{ accepts } x\}$. Show that A_{2DFA} is decidable.

b. Let $E_{\text{2DFA}} = \{\langle M\rangle \mid M \text{ is a 2DFA and } L(M) = \emptyset\}$. Show that E_{2DFA} is not decidable.

5.27 A ***two-dimensional finite automaton*** (2DIM-DFA) is defined as follows. The input is an $m \times n$ rectangle, for any $m, n \geq 2$. The squares along the boundary of the rectangle contain the symbol # and the internal squares contain symbols over the input alphabet Σ. The transition function is a mapping $Q \times \Sigma \longrightarrow Q \times \{\mathrm{L}, \mathrm{R}, \mathrm{U}, \mathrm{D}\}$ to indicate the next state and the new head position (Left, Right, Up, Down). The machine accepts when it enters one of the designated accept states. It rejects if it tries to move off the input rectangle or if it never halts. Two such machines are equivalent if they accept the same rectangles. Consider the problem of determining whether two of these machines are equivalent. Formulate this problem as a language, and show that it is undecidable.

A★**5.28** **Rice's theorem.** Let P be any nontrivial property of the language of a Turing machine. Prove that the problem of determining whether a given Turing machine's language has property P is undecidable.

In more formal terms, let P be a language consisting of Turing machine descriptions where P fulfills two conditions. First, P is nontrivial—it contains some, but not all, TM descriptions. Second, P is a property of the TM's language—whenever $L(M_1) = L(M_2)$, we have $\langle M_1 \rangle \in P$ iff $\langle M_2 \rangle \in P$. Here, M_1 and M_2 are any TMs. Prove that P is an undecidable language.

5.29 Show that both conditions in Problem 5.28 are necessary for proving that P is undecidable.

5.30 Use Rice's theorem, which appears in Problem 5.28, to prove the undecidability of each of the following languages.

A**a.** $INFINITE_{\mathsf{TM}} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is an infinite language}\}$.

b. $\{\langle M \rangle \mid M \text{ is a TM and } \texttt{1011} \in L(M)\}$.

c. $ALL_{\mathsf{TM}} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) = \Sigma^*\}$.

5.31 Let

$$f(x) = \begin{cases} 3x + 1 & \text{for odd } x \\ x/2 & \text{for even } x \end{cases}$$

for any natural number x. If you start with an integer x and iterate f, you obtain a sequence, $x, f(x), f(f(x)), \ldots$ Stop if you ever hit 1. For example, if $x = 17$, you get the sequence 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1. Extensive computer tests have shown that every starting point between 1 and a large positive integer gives a sequence that ends in 1. But, the question of whether all positive starting points end up at 1 is unsolved; it is called the $3x + 1$ problem.

Suppose that A_{TM} were decidable by a TM H. Use H to describe a TM that is guaranteed to state the answer to the $3x + 1$ problem.

5.32 Prove that the following two languages are undecidable.

a. $OVERLAP_{\mathsf{CFG}} = \{\langle G, H \rangle \mid G \text{ and } H \text{ are CFGs where } L(G) \cap L(H) \neq \emptyset\}$. (Hint: Adapt the hint in Problem 5.21.)

b. $PREFIX\text{-}FREE_{\mathsf{CFG}} = \{G \mid G \text{ is a CFG where } L(G) \text{ is prefix-free}\}$.

5.33 Let $S = \{\langle M \rangle \mid M \text{ is a TM and } L(M) = \{\langle M \rangle\}\,\}$. Show that neither S nor $\overline{S}$ is Turing-recognizable.

5.34 Consider the problem of determining whether a PDA accepts some string of the form $\{ww \mid w \in \{\texttt{0,1}\}^*\}$. Use the computation history method to show that this problem is undecidable.