4.4 Let \( A e_{CFG} = \{\langle G \rangle \mid G \text{ is a CFG that generates } \varepsilon \} \). Show that \( A e_{CFG} \) is decidable.

4.13 Let \( A = \{\langle R, S \rangle \mid R \text{ and } S \text{ are regular expressions and } L(R) \subseteq L(S)\} \). Show that \( A \) 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 \( A \).)

5.16 Let \( \Gamma = \{0, 1, \omega\} \) be the tape alphabet for all TMs in this problem. Define the busy beaver function \( BB : N \rightarrow N \) 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.

+ Prove that \( BB(k) \) grows faster than any computable function \( f(k) \), \( \lim_{k \to \infty} f(k)/BB(k) = 0 \).

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 \( \{1\}^* \).

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

\[
P = \left\{ \left[ \begin{array}{c} t_1 \\ b_1 \end{array} \right], \left[ \begin{array}{c} t_2 \\ b_2 \end{array} \right], \ldots, \left[ \begin{array}{c} t_k \\ b_k \end{array} \right] \right\},
\]

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

\[
S \rightarrow T | B \\
T \rightarrow t_1 a_1 | \cdots | t_k a_k | t_1 a_1 | \cdots | t_k a_k \\
B \rightarrow b_1 a_1 | \cdots | b_k a_k | b_1 a_1 | \cdots | b_k a_k 
\]

where \( a_1, \ldots, a_k \) are new terminal symbols. Prove that this reduction works.)
5.31 Let

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

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

6.19 Recall the Post correspondence problem that we defined in Section 5.2 and its associated language \( PCP \). Show that \( PCP \) is decidable relative to \( A_{\text{TM}} \).

7.21 Let \( \text{DOUBLE-SAT} = \{ \langle \phi \rangle \mid \phi \text{ has at least two satisfying assignments} \} \). Show that \( \text{DOUBLE-SAT} \) is NP-complete.

7.27 A **coloring** of a graph is an assignment of colors to its nodes so that no two adjacent nodes are assigned the same color. Let

\[ 3\text{COLOR} = \{ \langle G \rangle \mid \text{the nodes of } G \text{ can be colored with three colors such that no two nodes joined by an edge have the same color} \} \]

Show that \( 3\text{COLOR} \) is NP-complete. (Hint: Use the following three subgraphs.)

![Diagram](https://via.placeholder.com/150)

**palette**  **variable**  **OR-gadget**