1. Let $\Gamma = \{0, 1, \sqcup\}$ be the tape alphabet of all TMs in this problem. Define the busy beaver function $BB : \mathbb{N} \rightarrow \mathbb{N}$ 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.

2. Let $AMBIG_{CFG} = \{\langle G \rangle \mid G$ is an ambiguous CFG$\}$. Show that $AMBIG_{CFG}$ is undecidable. (Hint: Use a reduction from PCP. Given an instance $P = \{[t_1 \ b_1], [t_2 \ b_2], \ldots , [t_k \ b_k]\}$, of the Post Correspondence Problem, construct a CFG $G$ with the rules

$$S \rightarrow T | B$$
$$T \rightarrow t_1 Ta_1 | \cdots | t_k Ta_k | t_1 a_1 | \cdots | t_k a_k$$
$$B \rightarrow b_1 Ba_1 | \cdots | b_k Ba_k | b_1 a_1 | \cdots | b_k a_k,$$

where $a_1, \ldots , a_k$ are new terminal symbols. Prove that this reduction work.)

3. 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 $\{a^n b^n c^n \mid n \geq 0\}$.

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

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

4. Let $J = \{w \mid$ either $w = 0x$ for some $x \in A_{TM}$, or $w = 1y$ for some $y \notin A_{TM}\}$. Show that neither $J$ nor the complement of $J$ is Turing-recognizable.

5. 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 $\langle M_1 \rangle \in P$ if and only if $\langle M_2 \rangle \in P$. Here, $M_1$ and $M_2$ are any TMs. Prove that $P$ is an undecidable language.