1. **Decidability/Recognizability (60 points)**

**Height**
For a Turing machine $M$, let $\mathsf{height}(M)$ denote the length of the shortest string accepted by $M$ (or $\infty$ if $M$ does not accept any string). Consider the following language

$$L_{\mathsf{height}} = \{\langle M\rangle \mid \mathsf{height}(M) < |\langle M\rangle|\},$$

that is the language of Turing machines that accept a string that is shorter than their own description.

**a) (15 points)** Show that $L_{\mathsf{height}}$ is undecidable.

**b) (15 points)** Show that $L_{\mathsf{height}}$ is recognizable.

*For all parts, remember to first write your intuition before writing your formal solution.*

**Trump Machine**
A Turing machine $M$ over alphabet $\Sigma = \{0, 1, \$\}$ and tape alphabet $\Gamma = \{\sqcup, 0, 1, \$, \sharp\}$ is a Trump machine if for all $x \in \Sigma^*$ such that $x$ contains at least 5.7 billion \$ symbols, during the execution of $M$ on input $x$, $M$ erases all \$ symbols from its tape (replaces them with blanks), writes a single wall symbol $\sharp$, and then shuts down (either accepts or rejects).

Let

$$L_{\mathsf{Trump}} = \{\langle M\rangle \mid M \text{ is a Trump machine}\}.$$

**c) (15 points)** Show that $L_{\mathsf{Trump}}$ is undecidable.

**d) (15 points)** Show that $L_{\mathsf{Trump}}$ is unrecognizable.

2. **Inconclusive Decider (40 points)**
An Inconclusive Decider is like a decider except that on some set of inputs, instead of accepting or rejecting, it may instead halt and output "INCONCLUSIVE".

Formally, we say that a language $L$ is $k$-Inconclusive-Decidable for some $k \in \mathbb{N} \cup \{\infty\}$ if there exists a Turing machine $M$ such that

- For all $x$ such that $x \in L$, $M$ either halts and outputs "ACCEPT" or halts and outputs "INCONCLUSIVE".
- For all $x$ such that $x \notin L$, $M$ either halts and outputs "REJECT" or halts and outputs "INCONCLUSIVE".
- $M$ halts and outputs "INCONCLUSIVE" on at most $k$ distinct inputs $x$.

(Note that L is decidable if and only if it is 0-Inconclusive-Decidable)

a) **(5 points)** Show that all languages are $\infty$-Inconclusive-Decidable.

b) **(20 points)** Show that $L_{\mathsf{Halt}_\varepsilon} = \{\langle M\rangle \mid M(\varepsilon) \text{ halts}\}$ is **not** 1-Inconclusive-Decidable.

c) **(15 points)** Show that $L_{\mathsf{Halt}_\varepsilon} = \{\langle M\rangle \mid M(\varepsilon) \text{ halts}\}$ is **not** 2-Inconclusive-Decidable.

*For all parts, remember to first write your intuition before writing your formal solution.*

3. **Perpetual Machines (70 points)**

An perpetual machine $P$ is like a Turing machine, except that it never halts on any input. Instead of containing accept and reject states, $q_{accept}$ and $q_{reject}$, it contains a non-halting accepting state $q^*$. When $P$ enters $q^*$, instead of terminating its computation, it continues executing and can transition out of this state.

A perpetual machine $P$ computes a language $L$ if

- For all $x \in L$, $P(x)$ enters $q^*$ an infinite number of times
- For all $x \notin L$, $P(x)$ does not enter $q^*$ an infinite number of times

If there does not exist any perpetual machine $P$ that computes a language $L$, we say that $L$ is uncomputable by perpetual machines.

**a) (25 points)** Show that the language

$$L_{\mathsf{loop}} = \{\langle M \rangle \mid M \text{ is a Turing machine and } M(\varepsilon) \text{ loops}\}$$

is computable by perpetual machines.

**b) (20 points)** Construct a language $L \subseteq \Sigma^*$ that is uncomputable by perpetual machines and prove that this is the case. *(Hint: Use diagonalization)*

**c) (25 points)** Show that the language

$$L_{\mathsf{Empty}} = \{\langle M \rangle \mid M \text{ is a Turing machine and } L(M) = \emptyset\}$$

is computable by perpetual machines.

**d) (Extra Credit, 40 points)** Show that the language

$$L_{\mathsf{EQ}} = \{(\langle M_1 \rangle, \langle M_2 \rangle) \mid M_1 \text{ and } M_2 \text{ are Turing machines and } L(M_1) = L(M_2)\}$$

is computable by perpetual machines.

4. **Card Shuffling ([illegible] points)**

Let $C = (N, R)$ be a card shuffling game, described by a natural number $N$ and a set of allowed card shuffling rules $R$. We introduce some notation and define the game below.

- Normal Deck: A deck of $N$ cards labeled $(1, 2, 3, ..., N)$.
- Ace Deck: A normal deck but with the first card replaced by an Ace. In other words, a deck of $N$ cards labeled $(A, 2, 3, ..., N)$.
- **Shuffling Rule:** A shuffling rule specifies an allowed shuffle (permutation) of any finite number of cards. In particular, each rule is a tuple $([A, 1, 2, ..., N]^*, [A, 1, 2, ..., N]^*)$, where the left side of the tuple specifies the card sequence that can be shuffled with this rule, and the right side of the tuple specifies the new card sequence after the shuffle. For example, $([2, 7, 4, 1], [1, 2, 4, 7])$ means that if we see cards labeled 2, 7, 4, 1 next to each other in this order, then we can shuffle/rearrange them to be in the new order of 1, 2, 4, 7. $R$ is the set of allowed shuffling rules and is a subset of the set of all possible shuffles.
- Game Setup: In a row, we place face-up the cards of an Ace deck in order and then place face-up the cards of a normal deck in order.


- Gameplay: Each turn, we may do one of the following:
  - Add a normal deck: Place face-up the cards of a normal deck in order to the right of all cards currently placed.
  - Perform a shuffle: Choose any number of consecutive cards and shuffle them according to some allowed shuffle rule in $R$. In other words, choose any number of consecutive cards that are in the order specified in the left hand side of some rule in $R$, and then reorder them to be in the order specified by the right hand side of the same rule.
- Win Condition: You win if you can get a card labeled $N$ into the leftmost card position by playing this game. You can use as many turns as you want. (See example on next page.)

Let $L = \{C = (N, R) \mid \text{It is possible to win game } C\}$. Prove that $L$ is undecidable.
*Hint: Consider shuffles of 2N or 3N cards.*

You may assume that every TM can be converted into an equivalent TM that will never try to move left on the leftmost tape position, and if this new TM halts, it always halts when the head is at the leftmost tape position.

*Remember to first write your intuition before writing your formal solution.*

**Examples**

Let $C = (3, \{$[illegible]$\})$.
Then, you can win game C after four turns.

Setup

Turn 1: Use rule
((A, 2, 3, 1, 2, 3), (A, 2, 3, 2, 1, 3))

Turn 2: Add a new deck

Turn 3: Use rule
((2, 1, 3, 1, 2, 3), (3, 2, 1, 3, 2, 1))

Turn 4: Use rule
((A, 2, 3, 3, 2, 1), (3, 2, 1, 3, 2, A))

Win! (3 is in the leftmost spot)