1. Define:

\[ A_{DFA} = \{ < M, w > \mid M \text{ is a DFA that accepts input string } w \} \]

\[ EQ_{DFA,NFA} = \{ < M, N > \mid M \text{ is a DFA, } N \text{ is an NFA, and } L(M) = L(N) \} \]

![Diagram of finite automata M1 and M2](image)

Figure 1: A finite automaton M1 and a finite automaton M2.

Let M1 and M2 be the finite automata with alphabet \{a, b\} given in the above figure. Recall that DFA (respectively, NFA) stands for deterministic (respectively, nondeterministic) finite automaton.

Answer the following questions and give reasons for your answers.

(a) Is < M1, abb > ∈ A_{DFA}? Yes. M1 accepts abb.

(b) Is < M2, aa > ∈ A_{DFA}? No. M2 is not a DFA.

(c) Is < M1, M2 > ∈ EQ_{DFA,NFA}? No. L(M1) ≠ L(M2).

(d) Is < M2, M1 > ∈ EQ_{DFA,NFA}? No. M2 is not a DFA.

(e) Is < M1, M1 > ∈ EQ_{DFA,NFA}? Yes. L(M1) = L(M1).

(f) Is < M2, M2 > ∈ EQ_{DFA,NFA}? No. M2 is not a DFA.
(a) (7 marks) Let \( T = \{ (i, k, m) \mid i, k, m \in \mathbb{N} \} \). Show that the set \( T \) is countable.

All elements of \( T \) can be listed as follows:

1. List \((1, 1, 1)\)
2. List triples \((i, k, m)\) where \(i + k + m = 4\)
3. List triples \((i, k, m)\) where \(i + k + m = 5\)

Stage \(x\): List triples \((i, k, m)\) where \(i + k + m = x + 2\)

All elements of \( T \) appear on above list.

(b) (3 marks) Are the following sets countable? For each case circle the correct answer — no explanation needed. If you circle both YES and NO, it is considered a wrong answer.

- The set \( \{0, 1\}^* \). [YES] [NO]
- The set of all subsets of \( \{0, 1\}^* \). [YES] [NO]
- The set of all finite subsets of \( \{0, 1\}^* \). [YES] [NO]
Give an implementation-level description of a deterministic Turing machine that decides the following language over the alphabet $\Sigma = \{b, c, d\}$:

$$\{ w \in \Sigma^+ | w \text{ contains an equal number of } b\text{'s and } c\text{'s, or } w \text{ contains an equal number of } c\text{'s and } d\text{'s } \}. $$

$$M = \{\text{On input } w:\}$$

1. Repeat the following as long as new $b$'s and $c$'s are found:
   - Scan the input and mark one $b$ and one $c$.

2. If in the last scan no unmarked $b$'s and no unmarked $c$'s were found, accept.
   Else continue from 3.

3. Remove the masks from all the $c$'s.

4. Repeat the following as long as new unmarked $c$'s and $d$'s are found:
   - Scan the input and mark one $c$ and one $d$.

5. If in the last scan no unmarked $c$'s and no unmarked $d$'s were found, accept.
   Else reject.
We define

\[ \text{NEVERHALTS}_{\text{TM}} = \{ <M> \mid M \text{ is a deterministic Turing machine and the computation of } M \text{ does not halt on any input} \} \]

Without using Rice’s theorem show that \text{NEVERHALTS}_{\text{TM}} is undecidable.

We reduce \text{A}_{\text{TM}} to \text{NEVERHALTS}. Suppose \text{TM} P decides \text{NEVERHALTS}. Construct a decider for \text{A}_{\text{TM}}:

\[ Q = "\text{On input } <M,w> \text{ where } M \text{ is a TM} \]

1. Construct \text{TM} N:

\[ N = "\text{On input } x \text{  } \star \text{ignore this input } \star \]

1. Run M on w

2. If M accepts, accept.
   If M rejects, enter into an infinite loop.
   \(* M \text{ accepts } w \text{ if } N \text{ halts on some input } *\)

2. Run P on <N>.
   If P accepts, reject.
   If P rejects, accept"
(a) (4 marks) Does the following instance of the Post Correspondence Problem have a match (= solution). Justify your answer.

\[
\{ \begin{array}{c}
\frac{a}{ba}, \frac{aa}{abbb}, \frac{b}{baaa}, \frac{abb}{ab} \\
\end{array} \}
\]

\[
\text{Match: } \begin{bmatrix}
\frac{abb}{ab} \\
\frac{a}{bq}
\end{bmatrix}
\]

(b) (6 marks) We define the following language

\[
PCP = \{ < P > | P \text{ is an instance of the Post correspondence problem with a match } \}
\]

Answer the following questions and justify your answers.

i. Does there exist a decidable language \( A \) such that \( PCP \leq_m A \)?

\[\text{No. } PCP \text{ is undecidable and } B \leq_m A \text{ implies that } B \text{ is decidable (when } A \text{ is decidable)}\]

ii. Does there exist an undecidable language \( B \) such that \( PCP \leq_m B \)?

\[\text{Yes. } PCP \text{ is undecidable and } PCP \leq_m PCP \text{ via the identity mapping.}\]

iii. Does there exist a decidable language \( C \) such that \( C \leq_m PCP \)?

\[\text{Yes. } \text{Let } w_1 \text{ be an encoding of } PCP \text{ instance in 4(a), and } w_2 \text{ is an encoding of instance } \{ \frac{aa}{a} \} \text{ (with no match). Define } f : \{0,1\}^* \rightarrow \{0,1\}^* \text{ by } f(x) = \begin{cases} 
w_1 & \text{if } x = 0 \\
w_2 & \text{if } x \neq 0
\end{cases}
\]

\[\text{Now } \{0\} \leq_m PCP \text{ via the function } f.\]