Problem 1 (DFA Minimization — Essential)
Consider the following DFA $A$ over the alphabet $\{0, 1\}$:

1. Give the table $T$ of distinguishabilities for $A$. Show only the proper lower triangle of $T$, that is, fill in the following table with $X$ in each entry corresponding to a pair of distinguishable states.

2. Draw the minimal (4-state) DFA equivalent to $A$.

Problem 2 (NFA Conversion)
Consider the following clean NFA $N$:

1. Using the state-elimination method described in the book or in class, convert $N$ into an equivalent regular expression.

2. Using the set-of-states method, convert $N$ into an equivalent DFA, only giving those states that are reachable from the start state $S$. Do not perform any simplifications such as merging indistinguishable states.
Problem 3 (Closure Properties of Regular Languages — Essential)
The alphabet in this problem contains the symbol 0. For any word \( w \) on our alphabet let \( \text{NoZero}(w) \) result from \( w \) by deleting all the 0's in \( w \) and then closing up the gaps. For any language \( L \) let \( \text{NoZero}(L) = \{ \text{NoZero}(w) \mid w \in L \} \). Prove that if \( L \) is a regular language, then so is \( \text{NoZero}(L) \).

Problem 4 (The Pumping Lemma)
Show that the language
\[
L = \{ w \in \Sigma^* \mid \text{no prefix of } w \text{ has more 0's than 1's} \}
\]
is not pumpable (hence not regular). [Note: a string \( w \) is always a prefix of itself.]

Problem 5 (Turing Machines — Essential)
Consider a standard, one-tape Turing machine \( M \) with input alphabet \( \{0, 1\} \), tape alphabet \( \{0, 1, \sqcup\} \) where \( \sqcup \) is the blank symbol, and the following transition diagram:

![Turing Machine Diagram]

Give the sequence of IDs (configurations) of the complete computation of \( M \) on each of the three inputs 011, 10, and 0. (Hint: \( M \) decides the language of all binary strings of odd length whose middle symbol is 1, so the first input is accepted while the other two are rejected.)

Problem 6 (Decidability and Computability)
Fix some enumerator \( E \) that enumerates an infinite language. Let \( A \) be the language of all strings \( w \) so that
- \( E \) prints \( w \) at least once, and
- when \( w \) is first printed, it is longer than any string \( E \) has printed before.
Show that \( A \) is infinite and decidable. For the latter, give an explicit decision procedure for \( A \).

Problem 7 (Undecidability and Noncomputability — Essential)
Let \( f \) be a function that has the following property: for any TM \( M \) and string \( w \) such that \( M \) accepts \( w \), \( f(\langle M, w \rangle) \) outputs a number \( t \) such that \( M \) accepts \( w \) in less than \( t \) steps. (If \( M \) does not accept \( w \), then \( f(\langle M, w \rangle) \) could be any natural number.)
Show that no such \( f \) can be computable. [Hint: Show that if \( f \) were computable, then one can decide \( A_{\text{TM}} \).]

Problem 8 (Enumerability/Turing Recognizability)
Let \( f \) be a computable function. Show that range(\( f \)) is Turing-recognizable. (Here, range(\( f \)) is defined as \( \{ y \mid (\exists x) \ f(x) = y \} \).

Problem 9 (Reducibility — Essential)
Let \( B := \{ \langle M \rangle \mid M \text{ is a TM and } 010 \in L(M) \text{ and } 011 \notin L(M) \} \). Construct a mapping reduction from \( \overline{A_{\text{TM}}} \) to \( B \).

Problem 10 (Polynomial-Time Computability)
Let \( L := \{ w \mid \langle M, w \rangle \mid M \text{ is a DFA that accepts } w^n \text{ for all natural numbers } n \} \).
Show that \( L \in \text{P} \) by giving a polynomial time decision procedure for \( L \).
Problem 11 (NP and NP-Completeness — Essential)
The language

\[ \text{VERTEXCOVER} := \{ \langle G, k \rangle \mid G \text{ is a graph that has a vertex cover of size } k \}. \]

Show that VERTEXCOVER is NP-complete.

Problem 12 (Space-Bounded Computation)
Prove that if TBFQ $\in$ NP, then $\text{NP} = \text{PSPACE}$. 