**INSTRUCTIONS**

Upload a **single file** to Gradescope for each group. All group members’ names and PIDs should be on **each** page of the submission.

Your assignments in this class will be evaluated not only on the correctness of your answers, but on your ability to present your ideas clearly and logically. You should always explain how you arrived at your conclusions, using mathematically sound reasoning. Whether you use formal proof techniques or write a more informal argument for why something is true, your answers should always be well-supported. Your goal should be to convince the reader that your results and methods are sound.

**READING** Sipser Chapter 1.1-1.3

**KEY CONCEPTS** DFA, NFA, regular languages, closure properties on regular languages

---

**1.1** **(5 points)** In this problem, we will use the construction in the first part of the proof of Theorem 1.39 in Sipser which shows that every NFA (without ε transitions) has an equivalent DFA. Use the construction to create an equivalent DFA $D$ from the following NFA $N$ and provide the state diagram of the result. You may leave out unreachable states. (An automaton $A$ is equivalent to $B$ if $L(A) = L(B)$)

![State Diagram](image)

**1.2** **(5 points)** Describe $L(N)$ and provide the state diagram of a DFA $D'$ equivalent to $N$ (not necessarily using the construction).
2. **(10 points)** In this problem we will prove that every Multi-Start-State-NFA (MSSNFA) has an equivalent NFA with a single start state.
   For this problem we require you to:

   1. Provide the construction part of the proof from an MSSNFA $M$ to an NFA $N$ using the formal definition of an MSSNFA and the formal definition of an NFA.
   2. Perform the construction on the following MSSNFA $M$ and provide the state diagram of the result.

   ![State Diagram](image)

   3. Use the formal definition of computation to show that $L(M) = L(N)$.
      This requires two parts: $w \in L(M) \rightarrow w \in L(N)$ and $w \in L(N) \rightarrow w \in L(M)$.

See Sipser for the formal definition of an NFA.
The following is the definition of a Multi-Start-State-NFA and its definition of computation.

A Multi-Start-State-NFA is a 5-tuple $(Q, \Sigma, \delta, Q_0, F)$ where:

1. $Q$ is a finite set of states
2. $\Sigma$ is a finite alphabet
3. $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$ is a transition function
4. $Q_0 \subseteq Q$ is the set of start states
5. $F \subseteq Q$ is the set of final states

The formal definition of computation for a Multi-Start-State-NFA is similar to that of an NFA. Let $M = (Q, \Sigma, \delta, Q_0, F)$ be a Multi-Start-State-NFA and $w$ a string over the alphabet $\Sigma$. Then we say that $M$ accepts $w$ if we can write $w$ as $w = y_1y_2...y_m$, where $y_i$ is a member of $\Sigma_\epsilon$ and a sequence of states $r_0, r_1, ..., r_m$ exists in $Q$ with three conditions:

1. $r_0 \in Q_0$
2. $r_{i+1} \in \delta(r_i, y_{i+1})$, for $i = 0, ..., m - 1$
3. $r_m \in F$. 
3. (10 points) Prove that regular languages are closed under the reverse operation.

Use the following definitions:

- The reverse of $w$, written $w^R$, is the string obtained by writing $w$ in the opposite order (i.e., $w_nw_{n-1}...w_1$)
- $reverse(L) = \{w^R \mid w \in L\}$

For this problem we require you to:

1. Provide the construction part of the proof from a NFA $N$ to an NFA $N'$ using formal definitions.
2. Perform the construction on the following NFA $N$ and provide the state diagram of the result.

![State Diagram](image)

(You may construct a MSSNFA $M$ instead if you want.)

(The proof of correctness is not required.)
4. **(10 points)** Prove that regular languages are closed under the *stutter* operation.

Use the following definition: 

\[ \text{stutter}(L) = \{ a_1a_1a_2a_2...a_na_n \mid w = a_1a_2...a_n \in L \text{ and } a_i \in \Sigma \} \]

For this problem we require you to:

1. Provide the construction part of the proof from a DFA \( D \) to an NFA \( N \) using formal definitions.

2. Perform the construction on the following state diagram and provide the state diagram of the result.

\[ \text{start} \quad q_0 \quad q_1 \]

(You may construct a DFA \( D' \) instead if you want.)

(The proof of correctness is not required.)
5. (10 points) Prove that regular languages are closed under the \textit{skip} operation.
Use the following definition: \[ \text{skip}(L) = \{ a_1 a_3 a_5 \ldots a_n \mid w = a_1 a_2 \ldots a_n \in L \text{ and } n \text{ is odd and } a_i \in \Sigma \} \]
For this problem we require you to:

1. Provide the construction part of the proof from a DFA \( D \) to an NFA \( N \) using formal definitions.
2. Perform the construction on the following DFA \( D \) and provide the state diagram of the result.

(The proof of correctness is not required.)