All solutions should be neatly written or type set. All major steps in proofs must be justified.

1. (10 points) For this problem you will practice converting a NFA to a DFA. Convert the following NFA to a DFA. Show only the reachable states of the DFA. The transition function should be given in a table.

![NFA Diagram]

2. (10 points) For this problem you will have practice in showing that regular languages are closed under more operations using finite automata constructions. We define the simple interleaving of two languages $A$ and $B$ over $\Sigma$ by

$$A \mid B = \{x_1y_1 \cdots x_ny_n : x_i, y_i \in \Sigma, x_1x_2 \cdots x_n \in A, \text{ and } y_1y_2 \cdots y_n \in B\}.$$ 

For example if $A = \{a, ab, aa\}$ and $B = \{01, 11\}$ then $A \mid B = \{a0b1, a1b1, a0a1, a1a1\}.$

(a) Start with DFA’s $M_1$ and $M_2$ that accept $L_1$ and $L_2$, respectively. Then construct an DFA that accepts $L_1 \mid L_2$. A cross product type construction will be useful.

(b) State without proof a behavioral lemma for your construction that describes how your new machine behaves relative to the original machines.
(c) Use the behavioral lemma to prove that $M$ accepts $L_1 \mid L_2$.

3. (10 points) For this problem you will have more practice in showing that regular languages are closed under more operations using finite automata constructions. We define the reversal of a language as follows:

$$L^R = \{x^R : x \in L\}.$$

That is the reversal of a language is the set of reversals of all strings in the language.

(a) Given a DFA $M$ that accepts $L$ construct an NFA $M'$ such that $M'$ accepts $L^R$.

(b) State without proof a behavioral lemma for your construction that describes how your new machine behaves relative to the original machines.