1. NFAs to DFAs

Convert the following NFA to a DFA.

Solution:

- Start with the NFA.
- Create states for each subset of the NFA’s states.
- For each state and each input symbol, create transitions to the appropriate subset of the next state.
- Include a special state for the empty set ("\emptyset") to handle \lambda transitions.

The resulting DFA is shown in the solution diagram.
2 State minimization
Use the state minimization algorithm from lecture to minimize the following finite state machine.

Solution:
Regular Expressions to NFAs
Using the constructions given in lecture, find nondeterministic finite-state automata that recognize each of these sets:

a) $0^*1^*$

Solution:
1) First, we create the machine for the regular expression $0$:

2) Next, we apply the construction for $A^*$ with our machine $A = 0$ from the previous step:

3) Similarly, we can create the machine for $1^*$:

4) Lastly, we use the construction for concatenation to connect $0^*$ and $1^*$. Note that since our start state is also a final state in $0^*$, we must connect it to the start state for $1^*$ with lambda and then make sure that it becomes a non-final state as well.
b) \((0 \cup 11)^*\)
Solution:

![DFA Diagram]

4 FSAs to Regular Expressions
Convert the following DFA to a regular expression.

Solution:
\[ [11 \cup 00 \cup (10 \cup 01)(00 \cup 11)^*(01 \cup 10)]^* \]