1 NFA and ε-NFA Solutions

1. Every DFA is an NFA in which the transition relation only returns subsets of size at most 1; in other words, there is a unique arrow exiting every state for every symbol in the alphabet.

Can give any NFA where a state has two transitions for the same symbol, as a counter-example of the converse.

2. (a) We want to create an NFA for strings over \{0, 1\} that end with 1011. We can start with a machine (DFA) that only recognizes the string "1011"

Recall in the DFA version of this question, we had to add multiple transitions for intermediary states. However, it is much simpler as an NFA. We only need to add transitions in state \(\varepsilon\) such that it can accept any number of 0 and 1 before entering the next state:

(b) The NFA for this question looks very similar to the previous question. It will take 2 different paths instead of 1.

(c) We can start building the NFA by accepting string 1000.
Now to accept any strings that starts with 1000, we simply need another transition in the final state that goes back to itself on any input $\Sigma$.

This is actually a DFA, since for every state in the diagram, there is at most one transition for each symbol. This is perfectly valid since all DFAs are also NFAs.

3. When we take the union of two language $A \cup B$ using $\varepsilon$-NFA, we create a new starting state, and have an $\varepsilon$ transition going to the starting states of both $A$ and $B$.

To concatenate two language, we add $\varepsilon$-transitions between all final states of machine of the language to the starting state of the second language. We also remove all final states of the first machine from the final states of the concatenate language’s machine.

There are various ways to simplified this $\varepsilon$-NFA, we will leave it as an exercise.
2 NFA to DFA Solution

1. First, we add \{0\} as starting states. (Step 1, 2, 3)

\[
\text{start} \rightarrow \{0\}
\]

2. Next, we noticed that 0 can transition a to either 0 or 1, and can transition b to 2, so we add those transitions and state. (Step 4)

\[
\text{start} \rightarrow \{0\} \rightarrow \{0, 1\} \rightarrow \{0, 1, 2\} \rightarrow \{0, 1, 2, 3\}
\]

3. Repeat step 4 for state \{0, 1\}. Notice that 0 can take transition on a and b like the previous steps, and 1 can transition a to 2 and transition b to 1 and 3. So we add transitions from \{0, 1\} to \{0, 1, 2\} with a and \{1, 2, 3\} with b.

\[
\text{start} \rightarrow \{0\} \rightarrow \{0, 1\} \rightarrow \{0, 1, 2\} \rightarrow \{0, 1, 2, 3\}
\]

4. Repeat step 4 for state \{2\}. (detail omitted)
5. repeat step 4 for state \{3\}. (detail omitted)

6. repeat step 4 for state \{1\}. (detail omitted)

7. repeat step 4 for state \{1, 3\}. (detail omitted)
8. repeat step 4 for state \{0, 1, 2\}. (detail omitted)

9. repeat step 4 for state \{1, 2, 3\}. (detail omitted)

10. repeat step 4 for state \{1, 2\}. (detail omitted)
11. We’ve finally added transitions to every states. To finish off, we will add final states to the DFA. We will label every states contains an element in $A$ (every states that contains 3 in this case) as final state. (Step 6)

![Diagram of DFA with transitions](image)

### 3 ε-NFA to NFA solution

1. We begin with drawing the starting state and any states that could be reached by at least one non-ε transition, namely state 0, 2 and 3. (Step 1, 2)

![Diagram of ε-NFA](image)

2. Starting with state 0, noticed that it could reach itself with an input $a$ and it could reach 2 with an input $b$ (with input sequence εb). (Step 3)

![Diagram of ε-NFA with transitions](image)
3. Moving on to state 2, noticed that it could reach both itself and 3 with an $a$ and 0 with a $b$. (Step 3)

![Diagram of state machine]

4. Moving on to state 3, noticed how it can reach everything with a $b$ and it can reach 2 and itself with an $a$. (Step 3)

![Diagram of state machine]

5. Finally, mark both 2 and 3 as accepting since they can both reach 2 without any input. (Step 4)

![Diagram of state machine]