Problem 1. (a) An NFA with a single transition accepting \( L = \{aa, bb\}^* \):

![NFA Diagram]

Figure 1: NFA for \( L = \{aa, ba\}^* \)

(b) Convert the NFA of question (a) to a DFA.

When we apply the subset construction, we get:

<table>
<thead>
<tr>
<th>Transition</th>
<th>States</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>A {0}, B, C</td>
</tr>
<tr>
<td>b</td>
<td>A {0}, B, C</td>
</tr>
<tr>
<td>a</td>
<td>A {0}, B, C</td>
</tr>
<tr>
<td>b</td>
<td>A {0}, B, C</td>
</tr>
<tr>
<td>(\epsilon)</td>
<td>A {0}, B, C</td>
</tr>
<tr>
<td>(\epsilon)</td>
<td>A {0}, B, C</td>
</tr>
<tr>
<td>(\epsilon)</td>
<td>B {1}, D, E</td>
</tr>
<tr>
<td>(\epsilon)</td>
<td>B {1}, D, E</td>
</tr>
<tr>
<td>(\epsilon)</td>
<td>C {2}, E, D</td>
</tr>
<tr>
<td>(\epsilon)</td>
<td>C {2}, E, D</td>
</tr>
<tr>
<td>(\epsilon)</td>
<td>D {0, 3}, B, C</td>
</tr>
<tr>
<td>(\epsilon)</td>
<td>D {0, 3}, B, C</td>
</tr>
<tr>
<td>(\epsilon)</td>
<td>E \emptyset, E, E</td>
</tr>
<tr>
<td>(\epsilon)</td>
<td>E \emptyset, E, E</td>
</tr>
</tbody>
</table>

The final states are A and D and the start state is A.
Problem 2. (a) Given any DFA, $D = (Q, \Sigma, \delta, q_0, F)$, let $D'$ be the DFA, $D' = (Q \cup \{q'_0\}, \Sigma, \delta', q'_0, F')$, where $q'_0$ is a new state not in $Q$, with $F' = F$ if $q_0 \notin F$ else $F' = F \cup \{q'_0\}$, and with the transition function $\delta'$ defined as follows:

For all $a \in \Sigma$, if $p \in Q$ then

$$\delta'(p, a) = \delta(p, a)$$

else

$$\delta'(q'_0, a) = \delta(q_0, a).$$

Clearly, there are no incoming transitions into $q'_0$ and since the transitions from $q'_0$ are identical to the transitions from $q_0$ and all the other transitions are the same as in $D$, we have $L(D') = L(D)$.

(b) It is false that a DFA accepts a finite language iff its contains no underlying cycle. This is because, given any DFA, there must be a transition from every state on every input and as a DFA is finite, every DFA has a cycle! For example, the following DFA over the alphabet \{a\} only accepts $\epsilon$, yet it has a cycle:

Figure 3: DFA for $\{\epsilon\}$

Problem 3.
Let \( \Sigma = \{a_1, \ldots, a_m\} \) be an alphabet. Recall that the sets of languages \( R(\Sigma)_n \) are defined inductively as
\[
R(\Sigma)_0 = \{\{a_1\}, \ldots, \{a_m\}, \emptyset, \{\epsilon\}\}, \\
R(\Sigma)_{n+1} = R(\Sigma)_n \cup \{L_1 \cup L_2, L_1 L_2, L^* \mid L_1, L_2, L \in R(\Sigma)_n\}.
\]
and that the set of regular expressions \( \mathcal{R}(\Sigma)_n \) are defined inductively as
\[
\mathcal{R}(\Sigma)_0 = \{a_1, \ldots, a_m, \emptyset, \epsilon\}, \\
\mathcal{R}(\Sigma)_{n+1} = \mathcal{R}(\Sigma)_n \cup \{(R_1 + R_2), (R_1 \cdot R_2), R^* \mid R_1, R_2, R \in \mathcal{R}(\Sigma)_n\}.
\]

The function \( \mathcal{L} : \mathcal{R}(\Sigma) \rightarrow R(\Sigma) \) is defined recursively by
\[
\mathcal{L}[a_i] = \{a_i\}, \\
\mathcal{L}[\emptyset] = \emptyset, \\
\mathcal{L}[\epsilon] = \{\epsilon\}, \\
\mathcal{L}[(R_1 + R_2)] = \mathcal{L}[R_1] \cup \mathcal{L}[R_2], \\
\mathcal{L}[(R_1 R_2)] = \mathcal{L}[R_1] \mathcal{L}[R_2], \\
\mathcal{L}[R^*] = \mathcal{L}[R]^*.
\]

We prove by induction on \( n \geq 0 \) that for every regular expression \( R \in \mathcal{R}(\Sigma)_n \), we have \( \mathcal{L}[R] \in R(\Sigma)_n \).

For the base case \( n = 0 \), since \( R(\Sigma)_0 = \{\{a_1\}, \ldots, \{a_m\}, \emptyset, \{\epsilon\}\} \) and \( \mathcal{R}(\Sigma)_0 = \{a_1, \ldots, a_m, \emptyset, \epsilon\} \), the claim is true by definition of \( \mathcal{L} \).

For the induction step, pick any regular expression \( R \in \mathcal{R}(\Sigma)_{n+1} \). By the definition of \( \mathcal{R}(\Sigma)_{n+1} \), either \( R \in \mathcal{R}(\Sigma)_n \) or there are regular expressions \( R_1, R_2, R_3 \in \mathcal{R}(\Sigma)_n \) such that either \( R = (R_1 + R_2) \) or \( R = (R_1 R_2) \) or \( R = R_3^* \).

If \( R \in \mathcal{R}(\Sigma)_n \), then by the induction hypothesis \( \mathcal{L}[R] \in R(\Sigma)_n \).

Otherwise, by the induction hypothesis \( L_1 = \mathcal{L}[R_1], L_2 = \mathcal{L}[R_2], \) and \( L_3 = \mathcal{L}[R_3] \) all belong to \( R(\Sigma)_n \). By the recursive definition of \( \mathcal{L} \), we have
\[
\mathcal{L}[(R_1 + R_2)] = \mathcal{L}[R_1] \cup \mathcal{L}[R_2] = L_1 \cup L_2 \\
\mathcal{L}[(R_1 R_2)] = \mathcal{L}[R_1] \mathcal{L}[R_2] = L_1 L_2 \\
\mathcal{L}[(R_3^*)] = (\mathcal{L}[R_3])^* = L_3^* 
\]
and by the definition of \( R(\Sigma)_{n+1} \) we have \( L_1 \cup L_2, L_1 L_2, L_3^* \in R(\Sigma)_{n+1} \), establishing the induction hypothesis. Therefore, \( \mathcal{L}(\mathcal{R}(\Sigma)_n) \subseteq R(\Sigma)_n \). Since
\[
R(\Sigma) = \bigcup_{n \geq 0} R(\Sigma)_n \quad \text{and} \quad \mathcal{R}(\Sigma) = \bigcup_{n \geq 0} \mathcal{R}(\Sigma)_n,
\]

3
every $R \in \mathcal{R}(\Sigma)$ belongs to $\mathcal{R}(\Sigma)_n$ for some $n \geq 0$, and since $\mathcal{L}(\mathcal{R}(\Sigma)_n) \subseteq R(\Sigma)_n \subseteq R(\Sigma)$, we also have $\mathcal{L}(\mathcal{R}(\Sigma)) \subseteq R(\Sigma)$.

**Problem 4.** By definition, $L^R = \{w^R \mid w \in L\}$. We have

$$w \in (L_1 \cup L_2)^R \iff w^R \in L_1 \cup L_2$$

$$\text{iff } w^R \in L_1 \text{ or } w^R \in L_2$$

$$\text{iff } w \in L_1^R \text{ or } w \in L_2^R$$

$$\text{iff } w \in L_1^R \cup L_2^R,$$

which proves that

$$(L_1 \cup L_2)^R = L_1^R \cup L_2^R.$$

Recall that it was proved that

$$(uv)^R = v^R u^R \text{ and } (w^R)^R = w,$$

for all $u, v, w \in \Sigma^*$. We have

$$w \in (L_1 L_2)^R \iff w^R \in L_1 L_2$$

$$\text{iff } (\exists u \in L_1)(\exists v \in L_2)(w^R = uv)$$

$$\text{iff } (\exists u \in L_1)(\exists v \in L_2)(w = v^R u^R)$$

$$\text{iff } (\exists x \in L_1^R)(\exists y \in L_2^R)(w = yx)$$

$$\text{iff } w \in L_1^R L_2^R,$$

which proves that

$$(L_1 L_2)^R = L_1^R L_2^R.$$

We claim that

$$(L^n)^R = (L^R)^n, \quad \text{for all } n \geq 0.$$

This is proved by induction. For $n = 0$, we have

$$(L^0)^R = \{\epsilon\}^R = \{\epsilon\} = (L^R)^0,$$

so the base case holds.

Assume the induction hypothesis holds for any $n \geq 0$. Using $(L_1 L_2)^R = L_2^R L_1^R$, we get

$$(L^{n+1})^R = (L^n L)^R = L^R (L^n)^R = L^R (L^R)^n = (L^R)^{n+1},$$

establishing the induction step.

Then, we get

$$(L^*)^R = \left( \bigcup_{n \geq 0} L^n \right)^R = \bigcup_{n \geq 0} (L^n)^R = \bigcup_{n \geq 0} (L^R)^n = (L^R)^*,$$
so

\[(L^*)^R = (L^R)^*,\]

as claimed.

**Problem 5.** Let \( \Sigma = \{a, b\} \).

(a) A DFA accepting

\[L_1 = \{ w \in \Sigma^* \mid w \text{ contains an even number of } a's \}\]

![Figure 4: DFA for \( L_1 \)]

(b) A DFA accepting

\[L_2 = \{ w \in \Sigma^* \mid w \text{ contains a number of } b's \text{ divisible by } 3 \}\]

![Figure 5: DFA for \( L_2 \)]

(c) A DFA accepting \( L_3 = L_1 \cap L_2 \).

The cross-product construction (for intersection) yields:

<table>
<thead>
<tr>
<th></th>
<th>( a )</th>
<th>( b )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 0, A )</td>
<td>( 1, A )</td>
<td>( 0, B )</td>
</tr>
<tr>
<td>( 0, B )</td>
<td>( 1, B )</td>
<td>( 0, C )</td>
</tr>
<tr>
<td>( 0, C )</td>
<td>( 1, C )</td>
<td>( 0, A )</td>
</tr>
<tr>
<td>( 1, A )</td>
<td>( 0, A )</td>
<td>( 1, B )</td>
</tr>
<tr>
<td>( 1, B )</td>
<td>( 0, B )</td>
<td>( 1, C )</td>
</tr>
<tr>
<td>( 1, C )</td>
<td>( 0, C )</td>
<td>( 1, A )</td>
</tr>
</tbody>
</table>
The start state \((0, A)\) is also the only final state.

![Diagram of DFA for \(L_1 \cap L_2\)](image)

**Figure 6: DFA for \(L_1 \cap L_2\)**

**Problem 6.** Let \(D_1\) and \(D_2\) be two DFA’s over the same alphabet, \(\Sigma\). Using the hint, we have

\[L(D_1) \neq L(D_2) \text{ iff } L(D_1) - L(D_2) \neq \emptyset \text{ or } L(D_2) - L(D_1) \neq \emptyset.\]

However, the cross-product construction yields a DFA accepting \(L(D_1) - L(D_2)\) (or \(L(D_2) - L(D_1)\)) so all we have to do is to come up with a method for testing whether some arbitrary DFA \(D\) accepts a nonempty language. But, we have an algorithm for computing the set \(Q_r\) of states reachable from the start state \(q_0\) of \(D\), and observe that

\[L(D) \neq \emptyset \text{ iff } F \cap Q_r \neq \emptyset,\]

and the latter condition can obviously be tested (\(F\) is the set of final states of \(D\)).