1. Construct an NFA that accepts all binary strings that have a 1 as the third last symbol. I.e., 0000100 and 1111 are in the language, 1010 is not.

Solution:

$$
\begin{array}{c}
q_0 \quad 0, 1 \\
q_1 \\
q_2 \quad 1, 0 \\
q_3 \\
\end{array}
$$

2. Given an NFA $N = (\Sigma, Q, \delta, s, A)$, construct an NFA $N'$ that accepts all prefixes of $L(N)$, i.e., $w \in L(N') \Leftrightarrow wx \in L(N)$ for some $x \in \Sigma^*$.

Solution: Let $S$ be the set of states in $Q$ that are both reachable from $s$ and can reach an accepting state. Make every state in $S$ accepting; i.e., $N' = (\Sigma, Q, \delta, s, S)$.

3. Given an NFA $N = (\Sigma, Q, \delta, s, A)$, construct an NFA $N'$ that accepts all suffixes of $L(N)$, i.e., $w \in L(N') \Leftrightarrow xw \in L(N)$ for some $x \in \Sigma^*$.

Solution: Let $S$ be the set of states in $Q$ that are both reachable from $s$ and can reach an accepting state. Create a new start state $s_0$ and create an $\epsilon$-transition from $s_0$ to every state in $S$. I.e., $N' = (\Sigma, Q \cup \{s_0\}, \delta', s_0, A')$ where:

$$
\begin{align*}
\delta'(s_0, \epsilon) &= S \\
\delta'(s_0, c) &= \emptyset & \text{for any } c \in \Sigma \\
\delta'(s, c) &= \delta(s, c) & \text{for any } c \in \Sigma \cup \{\epsilon\}, s \in Q
\end{align*}
$$

Note that the addition of the extra state $s_0$ is necessary to avoid the $\epsilon$ transition being taken after the NFA takes a series of steps and returns to $s$.

4. Given an NFA $N = (\Sigma, Q, \delta, s, A)$, construct an NFA $N'$ that accepts $\text{insert1}(L(N))$, i.e., strings from $L(N)$ with a 1 inserted somewhere. In other words, $x \in L(N')$ if $x = y1z$ and some $y, z \in \Sigma^*$ and $yz \in L(N)$.

Solution: We can essentially have two copies of the state $Q$, with a transition between them when we see a 1. Formally:

$$
N' = (\Sigma, Q \times \{0, 1\}, \delta', (s, 0), A')
$$

where

$$
A' = \{(q, 1) | q \in A\}
$$

$$
\begin{align*}
\delta'((q, 0), 1) &= \{(r, 0) | r \in \delta(q, 1)\} \cup \{(q, 1)\} & \text{for } q \in Q \\
\delta'((q, 0), c) &= \{(r, 0) | r \in \delta(q, c)\} & \text{for } q \in Q, c \in \Sigma \cup \{\epsilon\} - \{1\} \\
\delta'((q, 1), c) &= \{(r, 1) | r \in \delta(q, c)\} & \text{for } q \in Q, c \in \Sigma \cup \{\epsilon\}
\end{align*}
$$
5. Given an NFA $N = (\Sigma, Q, \delta, s, A)$, construct an NFA $N'$ that accepts the reverse of $L(N)$, i.e., $w \in L(N') \iff w^R \in L(N)$.

**Solution:** This can be done by reversing every transition, and adding an extra starting state $s_0$ with an $\epsilon$-transition to every accepting state. The original start state $s$ becomes the sole accepting state.

$N' = (\Sigma, Q \cup s_0, \delta', s_0, \{s\})$ where:

- $\delta'(s_0, \epsilon) = A$
- $\delta'(s_0, c) = \emptyset$ for any $c \in \Sigma$
- $\delta'(s, c) = \{t \in Q | s \in \delta(t, c)\}$ for any $c \in \Sigma \cup \{\epsilon\}, s \in Q$