Let $L$ be an arbitrary regular language over the alphabet $\Sigma = \{0, 1\}$. Prove that the following languages are also regular. (You probably won’t get to all of these during the lab session.)

1. Let $\text{INSERTAny1s}(L)$ is the set of all strings that can be obtained from strings in $L$ by inserting any number of 1s anywhere in the string. For example:

$$\text{INSERTAny1s}(\{\varepsilon, 1, 00\}) = \{\varepsilon, 1, 11, 111, \ldots, 00, 100, 011110, 11101111101111, \ldots\}$$

Prove that the language $\text{INSERTAny1s}(L)$ is regular.

**Solution:** Let $M = (Q, s, A, \delta)$ be an arbitrary DFA that accepts the regular language $L$. We construct a new NFA with $\varepsilon$-transitions $M' = (Q', s', A', \delta')$ that accepts $\text{INSERTAny1s}(L)$ as follows.

Intuitively, $M'$ guesses which 1s in the input string have been inserted, skips over those 1s, and simulates $M$ on the original string $w$. $M'$ has the same states and start state and accepting states as $M$, but it has a different transition function.

- $Q' = Q$
- $s' = s$
- $A' = A$
- $\delta'(q, \emptyset) = \{ \delta(q, \emptyset) \}$
- $\delta'(q, 1) = \{ \}$
- $\delta'(q, \varepsilon) = \{ \}$
2. Let $\text{DELETEAny1s}(L)$ is the set of all strings that can be obtained from strings in $L$ by inserting any number of 1s anywhere in the string. For example:

$$\text{DELETEAny1s}([\epsilon, \ 00, 1101]) = [\epsilon, 0, 00, 01, 10, 101, 110, 1101]$$

Prove that the language $\text{DELETEAny1s}(L)$ is regular.

**Solution:** Let $M = (Q, s, A, \delta)$ be an arbitrary DFA that accepts the regular language $L$. We construct a new NFA with $\epsilon$-transitions $M' = (Q', s', A', \delta')$ that accepts $\text{DELETEAny1s}(L)$ as follows.

Intuitively, $M'$ guesses where 1s have been deleted from its input string, and simulates the original machine $M$ on the guessed mixture of input symbols and 1s. $M'$ has the same states and start state and accepting states as $M$, but a different transition function.

- $Q' = Q$
- $s' = s$
- $A' = A$
- $\delta'(q, \epsilon) = \{ \delta(q, \epsilon) \}$
- $\delta'(q, 0) = \{ \}$
- $\delta'(q, 1) = \{ \}$
- $\delta'(q, \epsilon) = \{ \}$

$\blacksquare$
3. Let $\text{InsertOne}_1(L) := \{ x1y \mid xy \in L \}$ denote the set of all strings that can be obtained from strings in $L$ by inserting \textit{exactly one} 1. For example:

$\text{InsertOne}_1(\{\epsilon, 00, 101101\}) = \{ 1, 100, 010, 001, 1101101, 101101, 101111 \}$

Prove that the language $\text{InsertOne}_1(L)$ is regular.

\textbf{Solution:} Let $M = (Q, s, A, \delta)$ be an arbitrary DFA that accepts the regular language $L$. We construct a new \textit{NFA with \texttt{\textepsilon}-transitions} $M' = (Q', s', A', \delta')$ that accepts $\text{InsertOne}_1(L)$ as follows.

If the input string $w$ does not contain a 1, then $M'$ must reject it; otherwise, intuitively, $M'$ \textit{guesses} which 1 was inserted into $w$, skips over that 1, and simulates $M$ on the remaining string $xy$.

$M'$ consists of two copies of $M$, one to process the prefix $x$ and the other to process the suffix $y$. State $(q, \text{False})$ means (the simulation of) $M$ is in state $q$ and $M'$ has not yet skipped over a 1. State $(q, \text{True})$ means (the simulation of) $M$ is in state $q$ and $M'$ has already skipped over a 1.

\begin{align*}
Q' &= Q \times \{\text{True}, \text{False}\} \\
s' &= (s, \text{False}) \\
A' &= \\
\delta'(q, \text{False}), \text{\theta} &= \{ (\delta(q, \text{\theta}), \text{\texttt{\textepsilon}}) \} \\
\delta'(q, \text{False}), 1 &= \{} \\
\delta'(q, \text{False}), \text{\texttt{\textepsilon}} &= \{} \\
\delta'(q, \text{True}), \text{\theta} &= \{} \\
\delta'(q, \text{True}), 1 &= \{} \\
\delta'(q, \text{True}), \text{\texttt{\textepsilon}} &= \{}
\end{align*}
4. Let \( \text{DELETEONE}1(L) := \{xy \mid x1y \in L\} \) denote the set of all strings that can be obtained from strings in \( L \) by deleting exactly one 1. For example:

\[
\text{DELETEONE}1(\{\varepsilon, 00, 101101\}) = \{01101, 10101, 10110\}
\]

Prove that the language \( \text{DELETEONE}1(L) \) is regular.

**Solution:** Let \( M = (\Sigma, Q, s, A, \delta) \) be a DFA that accepts the regular language \( L \). We construct an NFA with \( \varepsilon \)-transitions \( M' = (\Sigma, Q', s', A', \delta') \) that accepts \( \text{DELETEONE}1(L) \) as follows.

Intuitively, \( M' \) guesses where the 1 was deleted from its input string. It simulates the original DFA \( M \) on the prefix \( x \) before the missing 1, then the missing 1, and finally the suffix \( y \) after the missing 1.

\( M' \) consists of two copies of \( M \), one to process the prefix \( x \) and the other to process the suffix \( y \). State \((q, \text{FALSE})\) means (the simulation of) \( M \) is in state \( q \) and \( M' \) has not yet reinserted a 1. State \((q, \text{TRUE})\) means (the simulation of) \( M \) is in state \( q \) and \( M' \) has already reinserted a 1.

\[
\begin{align*}
Q' &= Q \times \{\text{TRUE}, \text{FALSE}\} \\
s' &= (s, \text{FALSE}) \\
A' &= \\
\delta'(\langle q, \text{FALSE} \rangle, 0 &= \{(q, 0, \text{FALSE}) \} \\
\delta'(\langle q, \text{FALSE} \rangle, 1 &= \{} \\
\delta'(\langle q, \text{FALSE} \rangle, \varepsilon &= \{} \\
\delta'(\langle q, \text{TRUE} \rangle, 0 &= \{} \\
\delta'(\langle q, \text{TRUE} \rangle, 1 &= \{} \\
\delta'(\langle q, \text{TRUE} \rangle, \varepsilon &= \{}
\end{align*}
\]
Work on these later: Consider the following recursively defined function on strings:

\[
evens(w) := \begin{cases} 
\epsilon & \text{if } w = \epsilon \\
\epsilon & \text{if } w = a \text{ for some symbol } a \\
b \cdot evens(x) & \text{if } w = abx \text{ for some symbols } a \text{ and } b \text{ and some string } x 
\end{cases}
\]

Intuitively, \( evens(w) \) skips over every other symbol in \( w \), starting with the first symbol. For example, \( evens(\text{THE\textbullet\textsc{SNAIL}}) = \text{H\textbullet\textsc{NI}} \) and \( evens(\text{GROB\textbullet\textsc{GOB\textbullet\textsc{GLOB\textbullet\textsc{GROD}}} = \text{RBGBG\textbullet\textsc{GO}}.} \)

Let \( L \) be an arbitrary regular language over the alphabet \( \Sigma = \{0, 1\} \).

5. Prove that the language \( \text{Unevens}(L) := \{w \mid evens(w) \in L\} \) is regular.

6. Prove that the language \( \text{Evens}(L) := \{evens(w) \mid w \in L\} \) is regular.