Let L be an arbitrary regular language over the alphabet $\Sigma = \{0, 1\}$. Prove that the following languages are also regular. For each language, we provide a high level intuitive sketch of the proof; your task is to fill in the remaining technical details. (You probably won't get to all of these during the lab session.)

Important note about notation: If a language L is described in set-builder notation as $\{stuff \mid condition\}$, then every finite-state machine that accepts L takes *stuff* as input, and checks whether *stuff* satisfies the stated *condition*.

In particular, every finite-state machine that accepts the language $\{foo(w) \mid bar(w) \in L\}$ takes an ***arbitrary string*** x as input, *guesses* a string w such that $x = foo(w)$, and determines whether the guessed string w satisfies the condition $bar(w) \in L$.

---

1. Let 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:

   $$\text{InsertAny1s}(\{\varepsilon,\ 1,\ 00\}) = \{\varepsilon,\ 1,\ 11,\ 111, \ldots,\ 00,\ 100,\ 0111110, 1110111111101111, \ldots\}$$

   Prove that the language 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 ε-transitions*** $M' = (Q', s', A', \delta')$ that accepts 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.

   $$\begin{aligned} Q' &= Q \\ s' &= s \\ A' &= A \\ \delta'(q, 0) &= \{\ \delta(q, 0)\ \} \\ \delta'(q, 1) &= \{\ \ \} \\ \delta'(q, \varepsilon) &= \{\ \ \} \end{aligned}$$

   ■

2. Let DeleteAny1s(L) is the set of all strings that can be obtained from strings in L by deleting ***any number of*** 1s anywhere in the string. For example:

$$\textsc{DeleteAny1s}(\{\varepsilon,\ \texttt{00},\ \texttt{1101}\}) = \{\varepsilon, \texttt{0}, \texttt{00}, \texttt{01}, \texttt{10}, \texttt{101}, \texttt{110}, \texttt{1101}\}$$

Prove that the language 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 ε-transitions*** $M' = (Q', s', A', \delta')$ that accepts 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.

$$\begin{aligned}
Q' &= Q \\
s' &= s \\
A' &= A \\
\delta'(q, \texttt{0}) &= \{\ \delta(q, \texttt{0})\ \} \\
\delta'(q, \texttt{1}) &= \{ \qquad\qquad\qquad\qquad\qquad\qquad \} \\
\delta'(q, \varepsilon) &= \{ \qquad\qquad\qquad\qquad\qquad\qquad \}
\end{aligned}$$

■

3. Let $\textsc{InsertOne1}(L) := \{x1y \mid xy \in L\}$ denote the set of all strings that can be obtained from strings in L by inserting ***exactly one*** 1. For example:

$$\textsc{InsertOne1}(\{\varepsilon, 00, 101101\}) = \{1, 100, 010, 001, 1101101, 1011101, 1011011\}$$

Prove that the language $\textsc{InsertOne1}(L)$ 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 ε-transitions*** $M' = (Q', s', A', \delta')$ that accepts $\textsc{InsertOne1}(L)$ as follows.

If the input string w does not contain a 1, then M' must rejects it; otherwise, intuitively, M' *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 $(q, \textsc{False})$ means (the simulation of) M is in state q and M' has not yet skipped over a 1. State $(q, \textsc{True})$ means (the simulation of) M is in state q and M' has already skipped over a 1.

$$Q' = Q \times \{\textsc{True}, \textsc{False}\}$$
$$s' = (s, \textsc{False})$$
$$A' =$$

$$\delta'((q, \textsc{False}), 0) = \{\ (\delta(q, 0), \textsc{False})\ \}$$
$$\delta'((q, \textsc{False}), 1) = \{\ \}$$
$$\delta'((q, \textsc{False}), \varepsilon) = \{\ \}$$

$$\delta'((q, \textsc{True}), 0) = \{\ \}$$
$$\delta'((q, \textsc{True}), 1) = \{\ \}$$
$$\delta'((q, \textsc{True}), \varepsilon) = \{\ \}$$

■

4. Let $\textsc{DeleteOne1}(L) := \{xy \mid x1y \in L\}$ denote the set of all strings that can be obtained from strings in L by deleting exactly one 1. For example:

$$\textsc{DeleteOne1}(\{\varepsilon, 00, 101101\}) = \{01101, 10101, 10110\}$$

Prove that the language $\textsc{DeleteOne1}(L)$ is regular.

Solution: Let $M = (\Sigma, Q, s, A, \delta)$ be a DFA that accepts the regular language L. We construct an ***NFA with ε-transitions*** $M' = (\Sigma, Q', s', A', \delta')$ that accepts $\textsc{DeleteOne1}(L)$ 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 $(q, \textsc{False})$ means (the simulation of) M is in state q and M' has not yet reinserted a 1. State $(q, \textsc{True})$ means (the simulation of) M is in state q and M' has already reinserted a 1.

$$Q' = Q \times \{\textsc{True}, \textsc{False}\}$$
$$s' = (s, \textsc{False})$$
$$A' =$$

$$\delta'((q, \textsc{False}), 0) = \{\ (\delta(q, 0), \textsc{False})\ \}$$
$$\delta'((q, \textsc{False}), 1) = \{\ \}$$
$$\delta'((q, \textsc{False}), \varepsilon) = \{\ \}$$

$$\delta'((q, \textsc{True}), 0) = \{\ \}$$
$$\delta'((q, \textsc{True}), 1) = \{\ \}$$
$$\delta'((q, \textsc{True}), \varepsilon) = \{\ \}$$

■

Work on these later: Consider the following recursive function on strings:

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

Intuitively, $evens(w)$ skips over every other symbol in w, starting with the first symbol. For example, $evens(\texttt{THE⋄SNAIL}) = \texttt{H⋄NI}$ and $evens(\texttt{GROB⋄GOB⋄GLOB⋄GROD}) = \texttt{RBGBGO⋄RD}$.

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

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

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