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 higholevel 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 $\{\mathit{stuff} \mid \mathit{condition}\}$, 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 $\{\mathit{foo}(w) \mid \mathit{bar}(w) \in L\}$ takes an ***arbitrary string*** x as input, *guesses* a string w such that $x = \mathit{foo}(w)$, and determines whether the guessed string w satisfies the condition $\mathit{bar}(w) \in L$.

---

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 want to build a machine M' that accepts InsertAny1s(L).

   We construct a new ***NFA with ε-transitions*** $M' = (Q', s', A', \delta')$ that accepts InsertAny1s(L) as follows. The input to M' is the *result* of inserting 1s into some string w, and M' needs to determine whether that string w is in L. Intuitively, M' *guesses* the original string w, by nondeterministically guessing which 1s were inserted, and simulates M running on that original string w.

   M' has the same states and start state and accepting states as M, but it has a different transition function.

   $$Q' = Q$$
   $$s' = s$$
   $$A' = A$$
   $$\delta'(q, 0) = \{\ \delta(q, 0)\ \}$$
   $$\delta'(q, 1) = \left\{\ \boxed{q,\ \delta(q, 1)}\ \right\}$$
   $$\delta'(q, \varepsilon) = \left\{\ \boxed{}\ \right\}$$

   Yes, the last box is empty. Each time M' reads a 1, M' guesses whether to ignore that 1 (because it was inserted) or to pass that 1 to M (because it was a symbol in w). In hindsight, we don't need any ε-transitions, so we can safely delete the final transition $\delta'(q, \varepsilon) = \varnothing$ and the phrase "with ε-transitions" at the beginning. ■

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:

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

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. The input to M' is the *result* of deleting 1s from some string w, and M' needs to determine whether that string w is in L. Intuitively, M' *guesses* the original string w, by nondeterministically guessing where 1s were deleted, and simulates M running on that original string w.

M' has the same states and start state and accepting states as M, but a different transition function.

$$Q' = Q$$
$$s' = s$$
$$A' = A$$

$$\delta'(q, 0) = \{\ \delta(q, 0)\ \}$$

$$\delta'(q, 1) = \left\{ \boxed{\ \delta(q, 1)\ } \right\}$$

$$\delta'(q, \varepsilon) = \left\{ \boxed{\ \delta(q, 1)\ } \right\}$$

M' passes each of its input symbols to M. Between input symbols M' uses ε-transitions to guess where 1s have been deleted and passes those deleted 1s to M. ∎

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

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

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. The input to M' is the *result* $x\mathtt{1}y$ of inserting exactly one 1 into some string xy, and M' needs to determine whether xy is in L. If the input string for M' does not contain a 1, then M' rejects it as invalid. Otherwise, intuitively, M' *guesses* which 1 in its input string was inserted, and simulates M on the rest of the input string.

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.

$$Q' = Q \times \{\textsc{True}, \textsc{False}\}$$
$$s' = (s, \textsc{False})$$
$$A' = \boxed{A \times \{\textsc{True}\} = \big\{(q, \textsc{True}) \bigm| q \in A\big\}}$$

$$\delta'((q, \textsc{False}), \mathtt{0}) = \{\ (\delta(q, \mathtt{0}), \textsc{False})\ \}$$
$$\delta'((q, \textsc{False}), \mathtt{1}) = \left\{\boxed{(\delta(q, \mathtt{1}), \textsc{False}),\ (q, \textsc{True})}\right\}$$
$$\delta'((q, \textsc{False}), \varepsilon) = \left\{\boxed{}\right\}$$

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

Whenever M' reads a 1 while its "skipped" flag is $\textsc{False}$, M' guesses whether to ignore that 1 and sets the "skipped" flag to $\textsc{True}$ (because that 1 was inserted into M's input string) or to pass it to the simulation of M. Otherwise, M' passes all input symbols directly to M. Finally, M' accepts if and only if its simulation of M accepts and M' has skipped a 1.

In hindsight, we don't need ε-transitions, so we can safely delete those transitions from our solution, along with the phrase "with ε-transitions" at the beginning. ■

4. Let $\text{DELETEONE1}(L) := \{xy \mid x1y \in L\}$ denote the set of all strings that can be obtained from strings in L by deleting exactly one 1. For example:

$$\text{DELETEONE1}(\{\varepsilon, 00, 101101\}) = \{01101, 10101, 10110\}$$

Prove that the language $\text{DELETEONE1}(L)$ 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 $\text{DELETEONE1}(L)$ as follows. The input to M' is the *result* of deleting one 1 from some string w, and M' needs to determine whether that string w is in L. Intuitively, M' *guesses* the original string w, by nondeterministically guessing where the 1 was deleted, and simulates M running on that original string w. Equivalently, M' simulates the original DFA M on the prefix x before the deleted 1, then the deleted 1 (which is *not* part of M''s input), and finally the suffix y after the deleted 1.

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

$$Q' = Q \times \{\text{TRUE}, \text{FALSE}\}$$
$$s' = (s, \text{FALSE})$$
$$A' = \boxed{A \times \{\text{TRUE}\} = \big\{(q, \text{TRUE}) \,\big|\, q \in A\big\}}$$

$$\delta'((q, \text{FALSE}), 0) = \{\, (\delta(q, 0), \text{FALSE}) \,\}$$
$$\delta'((q, \text{FALSE}), 1) = \left\{\boxed{(\delta(q, 1), \text{FALSE})}\right\}$$
$$\delta'((q, \text{FALSE}), \varepsilon) = \left\{\boxed{(\delta(q, 1), \text{TRUE})}\right\}$$
$$\delta'((q, \text{TRUE}), 0) = \left\{\boxed{(\delta(q, 0), \text{TRUE})}\right\}$$
$$\delta'((q, \text{TRUE}), 1) = \left\{\boxed{(\delta(q, 1), \text{TRUE})}\right\}$$
$$\delta'((q, \text{TRUE}), \varepsilon) = \left\{\boxed{}\right\}$$

M' passes each of its input symbols to M. Between input symbols, M' guesses where the single 1 has been deleted and passes that deleted 1 to M using an ε-transition. Finally, M' accepts if and only if its simulation of M accepts and M' has identified the deleted 1. ■

Work on these later: Consider the following recursively defined 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, $evens(\texttt{THE}\diamond\texttt{SNAIL}) = \texttt{H}\diamond\texttt{NI}$ and $evens(\texttt{GROB}\diamond\texttt{GOB}\diamond\texttt{GLOB}\diamond\texttt{GROD}) = \texttt{RBGBGO}\diamond\texttt{RD}$.

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.

Solution: Let $M = (\Sigma, Q, s, A, \delta)$ be a DFA that accepts the regular language L. We need to construct a new machine M' that accepts $\textsc{Unevens}(L)$. The input to M' is an arbitrary string w, and M' needs to determine whether $evens(w)$ is in L.

We construct a **DFA** $M' = (\Sigma, Q', s', A', \delta')$ that accepts $\textsc{Unevens}(L)$ as follows:

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

$$\delta'((q, 0), a) = (q, 1)$$
$$\delta'((q, 1), a) = (\delta(q, a), 0)$$

M' reads its input string w and simulates M running on $evens(w)$.

- State $(q, 0)$ means M is in state q and M' has read an even number of symbols, so M should ignore the next symbol (if any).
- State $(q, 1)$ means M is in state q and M' has read an odd number of symbols, so M should read the next symbol (if any).

As usual, I started by trying to construct an NFA with ε-transitions. But when I was finished, I noticed that all my ε-transitions led to $\varnothing$ and that all my real transitions led to a single state, which meant I had actually built a DFA! ■

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

Solution: Let $M = (\Sigma, Q, s, A, \delta)$ be a DFA that accepts the regular language L. We need to build a machine M' that accepts $evens(L)$. The input to M' is the *output* of $evens(w)$, and M' needs to determine whether the original string w is in L. So intuitively, M' guesses the odd-indexed symbols in w, and then passes the reconstituted string w to M.

We construct an NFA $M' = (\Sigma, Q', s', A', \delta')$ that accepts $evens(L)$ as follows:

$$\begin{aligned} Q' &= Q \\ s' &= s \\ A' &= A \cup \big\{q \in Q \bigm| \delta(q, 0) \in A\big\} \cup \big\{q \in Q \bigm| \delta(q, 1) \in A\big\} \\ \delta'(q, a) &= \big\{\delta\big(\delta(q, 0), a\big),\ \delta\big(\delta(q, 1), a\big)\big\} \end{aligned}$$

M' reads the input string $evens(w)$ and simulates M running on string w, while nondeterministically guessing the missing symbols in w.

- When M' reads the symbol a from $evens(w)$, it guesses a symbol $b \in \Sigma$ and simulates M reading ba from w.
- When M' finishes reading $evens(w)$, it guesses whether w has even or odd length, and in the odd case, it guesses the last symbol in w.

As usual, I started by trying to construct an NFA with ε-transitions. But when I was finished, I noticed that all my ε-transitions led to $\varnothing$, so I erased them. ■