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

1. $\textsc{FlipOdds}(L) := \{\mathit{flipOdds}(w) \mid w \in L\}$, where the function *flipOdds* inverts every odd-indexed bit in w. For example:

$$\mathit{flipOdds}(\texttt{0000111101010101}) = \texttt{1010010111111111}$$

Solution: Let $M = (Q, s, A, \delta)$ be a DFA that accepts L. We construct a new DFA $M' = (Q', s', A', \delta')$ that accepts $\textsc{FlipOdds}(L)$ as follows.

Intuitively, M' receives some string $\mathit{flipOdds}(w)$ as input, restores every other bit to obtain w, and simulates M on the restored string w.

Each state (q, flip) of M' indicates that M is in state q, and we need to flip the next input bit if $\mathit{flip} = \textsc{True}$.

$$\begin{aligned} Q' &= Q \times \{\textsc{True}, \textsc{False}\} \\ s' &= (s, \textsc{True}) \\ A' &= \\ \delta'((q, \mathit{flip}), a) &= \end{aligned}$$

■

2. $\textsc{UnflipOdd}\texttt{1}\textsc{s}(L) := \{w \in \Sigma^* \mid \mathit{flipOdd}\texttt{1}\mathit{s}(w) \in L\}$, where the function $\mathit{flipOdd}\texttt{1}$ inverts every other $\texttt{1}$ bit of its input string, starting with the first $\texttt{1}$. For example:

$$\mathit{flipOdd}\texttt{1}\mathit{s}(\texttt{0000}\underline{\texttt{1}}\texttt{1}\underline{\texttt{1}}\texttt{10}\underline{\texttt{1}}\texttt{010}\underline{\texttt{1}}\texttt{01}) = \texttt{0000}\underline{\texttt{0}}\texttt{1}\underline{\texttt{0}}\texttt{10}\underline{\texttt{0}}\texttt{010}\underline{\texttt{0}}\texttt{01}$$

Solution: Let $M = (Q, s, A, \delta)$ be a DFA that accepts L. We construct a new DFA $M' = (Q', s', A', \delta')$ that accepts $\textsc{UnflipOdd}\texttt{1}\textsc{s}(L)$ as follows.

Intuitively, M' receives some string w as input, flips every other $\texttt{1}$ bit, and simulates M on the transformed string.

Each state (q, flip) of M' indicates that M is in state q, and we need to flip the next $\texttt{1}$ bit of and only if $\mathit{flip} = \textsc{True}$.

$$\begin{aligned} Q' &= Q \times \{\textsc{True}, \textsc{False}\} \\ s' &= (s, \textsc{True}) \\ A' &= \\ \delta'((q, \mathit{flip}), a) &= \end{aligned}$$

■

3. $\textsc{FlipOdd1s}(L) := \{flipOdd1s(w) \mid w \in L\}$, where the function *flipOdd1* is defined as in the previous problem.

Solution: Let $M = (Q, s, A, \delta)$ be a DFA that accepts L. We construct a new **NFA** $M' = (Q', s', A', \delta')$ that accepts $\textsc{FlipOdd1s}(L)$ as follows.

Intuitively, M' receives some string $flipOdd1s(w)$ as input, ***guesses*** which 0 bits to restore to 1s, and simulates M on the restored string w. No string in $\textsc{FlipOdd1s}(L)$ has two 1s in a row, so if M' ever sees 11, it rejects.

Each state $(q, flip)$ of M' indicates that M is in state q, and we need to flip a 0 bit before the next 1 if $flip = \textsc{True}$.

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

■

4. Prove that the language $insert1(L) := \{x1y \mid xy \in L\}$ is regular.

Intuitively, $insert1(L)$ is the set of all strings that can be obtained from strings in L by inserting exactly one 1. For example, if $L = \{\varepsilon, \texttt{OOK!}\}$, then $insert1(L) = \{\texttt{1}, \texttt{1OOK!}, \texttt{O1OK!}, \texttt{OO1K!}, \texttt{OOK1!}, \texttt{OOK!1}\}$.

Solution: Let $M = (\Sigma, Q, s, A, \delta)$ be a DFA that accepts L. We construct an **NFA** $M' = (\Sigma, Q', s', A', \delta')$ that accepts $insert1(L)$ as follows.

Intuitively, M' nondeterministically chooses a 1 in the input string to ignore, and simulates M running on the rest of the input string.

- The state $(q, before)$ means (the simulation of) M is in state q and M' has not yet skipped over a 1.
- The state $(q, after)$ means (the simulation of) M is in state q and M' has already skipped over a 1.

$$Q' = Q \times \{before, after\}$$
$$s' = (s, before)$$
$$A' =$$
$$\delta'((q, before), a) =$$
$$\delta'((q, after), a) =$$

■

Work on these later:

5. Prove that the language $delete\mathtt{1}(L) := \{xy \mid x\mathtt{1}y \in L\}$ is regular.

   Intuitively, $delete\mathtt{1}(L)$ is the set of all strings that can be obtained from strings in L by deleting exactly one $\mathtt{1}$. For example, if $L = \{\mathtt{101101}, \mathtt{00}, \varepsilon\}$, then $delete\mathtt{1}(L) = \{\mathtt{01101}, \mathtt{10101}, \mathtt{10110}\}$.

6. Consider the following recursively defined function on strings:

$$stutter(w) := \begin{cases} \varepsilon & \text{if } w = \varepsilon \\ aa \bullet stutter(x) & \text{if } w = ax \text{ for some symbol } a \text{ and some string } x \end{cases}$$

   Intuitively, $stutter(w)$ doubles every symbol in w. For example:

   - $stutter(\mathtt{PRESTO}) = \mathtt{PPRREESSTTOO}$
   - $stutter(\mathtt{HOCUS{\diamond}POCUS}) = \mathtt{HHOOCCUUSS{\diamond}{\diamond}PPOOCCUUSS}$

   (a) Prove that the language $stutter^{-1}(L) := \{w \mid stutter(w) \in L\}$ is regular.

   (b) Prove that the language $stutter(L) := \{stutter(w) \mid w \in L\}$ is regular.

7. 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. For example:

   - $evens(\mathtt{EXPELLIARMUS}) = \mathtt{XELAMS}$
   - $evens(\mathtt{AVADA{\diamond}KEDAVRA}) = \mathtt{VD{\diamond}EAR}$.

   (a) Prove that the language $evens^{-1}(L) := \{w \mid evens(w) \in L\}$ is regular.

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

You may find it helpful to imagine these transformations concretely on the following DFA for the language specified by the regular expression $\mathtt{0}\mathtt{0}^*\mathtt{1}\mathtt{1}^*$.