

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

1. Let $\text{INSERTEDANY1s}(L)$ 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:

$$\text{INSERTEDANY1s}(\{\varepsilon, 1, 00\}) = \{\varepsilon, 1, 11, 111, \dots, 00, 100, 0111110, 111011111101111, \dots\}$$

Prove that the language $\text{INSERTEDANY1s}(L)$ 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 $\text{INSERTEDANY1s}(L)$ 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.

$$Q' = Q$$

$$s' = s$$

$$A' = A$$

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

$$\delta'(q, 1) = \{ \hspace{10em} \}$$

$$\delta'(q, \varepsilon) = \{ \hspace{10em} \}$$

■

2. Let $\text{DELETEDANY1s}(L)$ 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:

$$\text{DELETEDANY1s}(\{\epsilon, 00, 1101\}) = \{\epsilon, 0, 00, 01, 10, 101, 110, 1101\}$$

Prove that the language $\text{DELETEDANY1s}(L)$ 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 $\text{DELETEDANY1s}(L)$ 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.

$$Q' = Q$$

$$s' = s$$

$$A' = A$$

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

$$\delta'(q, 1) = \{ \hspace{15em} \}$$

$$\delta'(q, \epsilon) = \{ \hspace{15em} \}$$

■

3. Let $\text{INSERTEDONE1}(L) := \{x1y \mid xy \in L\}$ denote the set of all strings that can be obtained from strings in L by inserting *exactly one* 1. For example:

$$\text{INSERTEDONE1}(\{\epsilon, 00, 101101\}) = \{1, 100, 010, 001, 1101101, 1011101, 1011011\}$$

Prove that the language $\text{INSERTEDONE1}(L)$ 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 $\text{INSERTEDONE1}(L)$ as follows.

If the input string w does not contain a 1, then M' must reject 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, FALSE) means (the simulation of) M is in state q and M' has not yet skipped over a 1. State (q, TRUE) means (the simulation of) M is in state q and M' has already skipped over a 1.

$$Q' = Q \times \{\text{TRUE}, \text{FALSE}\}$$

$$s' = (s, \text{FALSE})$$

$$A' =$$

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

$$\delta'((q, \text{FALSE}), 1) = \{ \quad \quad \quad \}$$

$$\delta'((q, \text{FALSE}), \epsilon) = \{ \quad \quad \quad \}$$

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

$$\delta'((q, \text{TRUE}), 1) = \{ \quad \quad \quad \}$$

$$\delta'((q, \text{TRUE}), \epsilon) = \{ \quad \quad \quad \}$$

■

4. Let $\text{DELETEDONE1}(L) := \{xy \mid x1y \in L\}$ denote the set of all strings that can be obtained from strings in L by deleting exactly one 1. For example:

$$\text{DELETEDONE1}(\{\varepsilon, 00, 101101\}) = \{01101, 10101, 10110\}$$

Prove that the language $\text{DELETEDONE1}(L)$ 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 $\text{DELETEDONE1}(L)$ 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, 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.

$$Q' = Q \times \{\text{TRUE}, \text{FALSE}\}$$

$$s' = (s, \text{FALSE})$$

$$A' =$$

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

$$\delta'((q, \text{FALSE}), 1) = \{ \hspace{15em} \}$$

$$\delta'((q, \text{FALSE}), \varepsilon) = \{ \hspace{15em} \}$$

$$\delta'((q, \text{TRUE}), 0) = \{ \hspace{15em} \}$$

$$\delta'((q, \text{TRUE}), 1) = \{ \hspace{15em} \}$$

$$\delta'((q, \text{TRUE}), \varepsilon) = \{ \hspace{15em} \}$$

■

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

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

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

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

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