

Let L be an arbitrary regular language over the alphabet $\Sigma = \{0, 1\}$. Prove that the following languages are also regular.

1. $\text{SUPERSTRINGS}(L) := \{xyz \mid y \in L \text{ and } x, z \in \Sigma^*\}$. This language contains all superstrings of strings in L . For example:

$$\text{SUPERSTRINGS}(\{10010\}) = \{\underline{10010}, 010\underline{10010}, \underline{10010}11, 100\underline{10010}010, \dots\}$$

[Hint: This is much easier than it looks.]

2. $\text{SUBSTRINGS}(L) := \{y \mid x, y, z \in \Sigma^* \text{ and } xyz \in L\}$. This language contains all substrings of strings in L . For example:

$$\text{SUBSTRINGS}(\{10010\}) = \{\varepsilon, 0, 1, 00, 01, 10, 001, 010, 100, 0010, 1001, 10010\}$$

3. $\text{CYCLE}(L) := \{xy \mid x, y \in \Sigma^* \text{ and } yx \in L\}$. This language contains all strings that can be obtained by splitting a string in L into a prefix and a suffix and concatenating them in the wrong order. For example:

$$\text{CYCLE}(\{00K!, 00K00K\}) = \{00K!, 0K!0, K!00, !00K, 00K00K, 0K00K0, K00K00\}$$

Work on these later.

5. $\text{SUBSEQUENCES}(L)$ is the set of all *subsequences* of strings in L . A subsequence of a string w is the result of deleting zero or more symbols from w , leaving the remaining symbols in order. For example:

$$\text{SUBSEQUENCES}(\{10010\}) = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 100, 101, 110, 0010, 1000, 1001, 10010\}$$

6. $\text{FLIPODDS}(L) := \{\text{flipOdds}(w) \mid w \in L\}$, where the function flipOdds inverts every other bit in w , starting with the first bit. For example:

$$\text{flipOdds}(0000\underline{1111010100}) = \underline{101001011111110}$$

7. $\text{UNFLIPODD1S}(L) := \{w \in \Sigma^* \mid \text{flipOdd1s}(w) \in L\}$, where the function flipOdd1s inverts every other **1** bit of its input string, starting with the first **1**. For example:

$$\text{flipOdd1s}(0000\underline{111100101010}) = 00000\underline{10100001000}$$

8. $\text{FLIPODD1S}(L) := \{\text{flipOdd1s}(w) \mid w \in L\}$, where the function flipOdd1s is defined in the previous problem.

9. $\text{STUTTER}(L) = \{\text{stutter}(w) \mid w \in L\}$, where the function stutter duplicates every symbol in the input string:

$$\text{stutter}(w) := \begin{cases} \varepsilon & \text{if } w = \varepsilon \\ aa \cdot \text{stutter}(x) & \text{if } w = ax \end{cases}$$

For example, $\text{stutter}(1001) = 11000011$.

10. $\text{UNSTUTTER}(L) = \{w \mid \text{stutter}(w) \in L\}$, where the function stutter is defined in the previous problem.