# Regular Language Review Questions

**1.** Consider the alphabet $\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\}$. We shall use strings in this alphabet to describe two integers: one using the top row of bits and one using the bottom row. Each integer is represented in binary. For example, to represent the two integers 13 and 7 (whose binary representations are and 1101 and 111), we would use the string $\begin{pmatrix} 1 \\ 0 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$: the top row is 1101, and the bottom row is 111 (the extra 0 at the beginning of the bottom row is just padding to make the two rows the same length).

**(a)** Let *LESS* be the language of all strings where the integer represented in the top row is less than the integer represented by the bottom row. For example, the string $\begin{pmatrix} 1 \\ 0 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ is not in *LESS*, since 13 is not less than 7. Give a regular expression for the language *LESS*.

**(b)** Let *LESS2* be the language of all strings where the integer represented in the top row is less than double the integer represented by the bottom row. For example, the string $\begin{pmatrix} 1 \\ 0 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ is in *LESS2*, since $13 < 2 \cdot 7$. Give a regular expression for the language *LESS2*.

Hint: It might be easier to find the regular expression by drawing a finite automaton for *LESS2* first.

**2.** Let $L$ be the language described by the regular expression $1^*((0 \cup 00)11^*)^*(\varepsilon \cup 0 \cup 00)$.

**(a)** List five strings that are in $L$.

**(b)** List five strings that are not in $L$.

**(c)** Give a precise English description of the language $L$.

**3.** If $L \subseteq \Sigma^*$ is any language, we define $DROP(L)$ to be the set of all strings that can be obtained by removing a single character from a string in $L$. More formally,

$$DROP(L) = \{xy : x, y \in \Sigma^* \text{ and } \exists a \in \Sigma \text{ such that } xay \in L\}.$$

**(a)** Let $L_a = \{\texttt{010}, \varepsilon, \texttt{011}, \texttt{0}, \texttt{0010}\}$. List all the elements of $DROP(L_a)$.

**(b)** Let $L_b$ be the language described by the regular expression $\texttt{a}(\texttt{bc})^*$. Write a regular expression for the language $DROP(L_b)$. You do not have to prove your answer is correct.

**(c)** Your goal is to define a function $D$ that maps regular expressions to regular expressions. Give a recursive definition of $D(R)$ so that if $R$ is a regular expression for a language $L$, then $D(R)$ is a regular expression for the language $DROP(L)$.

You can write your answer by filling in the blanks in the following. (I have given you the answer for two of the six cases needed for the recursive definition.)

$$\begin{aligned} D(\emptyset) &= \emptyset \\ D(\varepsilon) &= \underline{\hspace{4cm}} \\ D(a) &= \underline{\hspace{4cm}}, \text{ for any } a \in \Sigma \end{aligned}$$

If $R_1$ and $R_2$ are regular expressions then,

$$\begin{aligned} D(R_1 \cup R_2) &= D(R_1) \cup D(R_2) \\ D(R_1 \circ R_2) &= \underline{\hspace{4cm}} \\ D(R_1^*) &= \underline{\hspace{4cm}} \end{aligned}$$

**(d)** Prove that for all regular expressions $R$, if $L$ is the language represented by $R$, then the language represented by $D(R)$ is $DROP(L)$.

In other words, explain why your answer to part (c) is correct for all $R$.

Hint: use mathematical induction on the number of operators $(\cup, ^*, \circ)$ that appear in the regular expression $R$.

**(e)** Prove that if $L$ is any regular language then $DROP(L)$ is also a regular language.

**4.** Let $L_1 = \{\mathtt{0}^j\mathtt{1}^k\mathtt{2}^\ell : j \geq k \text{ or } j \leq \ell\}$. Is $L_1$ regular? Prove your answer is correct.

**5.** Let $\Sigma = \{), (\}$. Let $L_3 \subseteq \Sigma^*$ be the set of strings of correctly balanced parentheses. For example, $(())()$ is in $L_3$ and $(()))($ is not in $L_3$. Formally, $L_3$ is defined recursively as follows.

- $\varepsilon \in L_3$.
- A string $x \neq \varepsilon$ is in $L_3$ if and only if $x$ is of the form $(y)z$, where $y$ and $z$ are in $L_3$.

Prove that $L_3$ is not regular.

**6.** If $L$ is a language, let $PAL(L)$ be the language consisting of all palindromes in $L$. More formally, $PAL(L) = \{x \in L : x = x^R\}$.

**(a)** Let $L_2$ be the language described by the regular expression $\mathtt{0}^*\mathtt{1}^*\mathtt{0}$. Give a regular expression for the language $PAL(L_2)$. You do not have to prove your answer is correct.

**(b)** Is the following claim true or false?
**Claim:** For all languages $L$, if $L$ is regular, then $PAL(L)$ is regular.
Prove your answer is correct.