1 Find a string of minimal length among those that belong to (the language represented by the regular expression) \((10 + 0)^{*}0\) but do not belong to (the language represented by the regular expression) \(0(1*01^*0)^{+}1(0 + 1)^{*}\).

00

2 Give a regular expression for the language over \(\{a, b, c\}\) consisting of the strings that begin with \(a\) or don’t end with \(c\).

\[a(a + b + c)^{*} + \Lambda + (a + b + c)^{*}(a + b)\]

3 Give a regular expression for the language over \(\{a, b\}\) consisting of the strings that do not end with \(aaa\).

\[\Lambda + a + aa + (a + b)^{*}(b + ba + baa)\]

4 In proving things about DFAs it is convenient to use the following inductive definition of \(\Sigma^*\).

- **Basis**: \(\Lambda \in \Sigma^*\).
- **Induction**: If \(x \in \Sigma^*\) and \(a \in \Sigma\), then \(xa \in \Sigma^*\).

In accordance with this definition, we can define the reverse \(x^R\) of a string \(x\) recursively, as follows.

\[
\begin{align*}
\Lambda^R &= \Lambda \\
(xa)^R &= ax^R \\
(x \in \Sigma^*, a \in \Sigma)
\end{align*}
\]

Prove that for all \(a \in \Sigma\), \(a^R = a\).

\[
\begin{align*}
a^R &= (\Lambda a)^R \\
&= a\Lambda^R \quad \text{(recursive defn of reverse)} \\
&= a\Lambda \quad \text{(recursive defn of reverse)} \\
&= a
\end{align*}
\]
5 Prove that for all $x \in \Sigma^*$,
\[(x^R)^R = x\]
by structural induction on $x$ (using the inductive definition of $\Sigma^*$ shown in the previous problem).

You may use the following two lemmas without proof.

Lemma 1. For all $a \in \Sigma$, $a^R = a$. (Result from previous problem.)

Lemma 2. For all $x, y \in \Sigma^*$, $(xy)^R = y^Rx^R$.

Claim: For all $x \in \Sigma^*$, $(x^R)^R = x$.

Proof by structural induction on $x$.

Basis: $(\Lambda^R)^R = (\Lambda)^R = \Lambda$.

Induction: $x \in \Sigma^*$, $a \in \Sigma$.
IH: $(x^R)^R = x$.
NTS: $((xa)^R)^R = xa$.

\[
\begin{align*}
((xa)^R)^R &= (ax^R)^R \\
&= (x^R)^R a^R \quad \text{(defn reverse)} \\
&= xa^R \quad \text{(Lemma 2)} \\
&= xa \quad \text{(IH)} \\
&= xa \quad \text{(Lemma 1)}
\end{align*}
\]

6 Recall the inductive definition of the set of regular languages. The set of regular languages over an alphabet $\Sigma$ is the least set of languages over $\Sigma$ s.t.
1. $\emptyset$ is a regular language,
2. $\{\Lambda\}$ is a regular language,
3. For all $a \in \Sigma$, $\{a\}$ is a regular language,
4. If $A$ is a regular language, so is $A^*$,
5. If $A$ and $B$ are regular languages, so are $A \cup B$ and $AB$.

(Here, conditions 1–3 are the basis part of the inductive definition, and conditions 4 & 5 are the induction part.)

Prove by structural induction (on the inductive definition of the set of regular languages) that for every regular language $L$, the language $L^R = \{x^R \mid x \in L\}$ is regular.
You may use without proof the following two lemmas.

**Lemma 1:** For any language $L$, $(L^*)^R = (L^R)^*$.

**Lemma 2:** For any languages $L_1, L_2$, $(L_1 L_2)^R = L_2^R L_1^R$.

**Claim:** For every regular language $L$, the language $L^R$ is regular.

**Proof by structural induction on $L$** (that is, on the inductive definition of the set of regular languages).

For convenience, we divide the presentation of the *basis* part of the proof into three cases (cases 1, 2 & 3 below). Similarly, we divide the presentation of the *induction* part of the proof into two cases (cases 4 & 5 below).

**Case 1:** $\emptyset^R = \emptyset$, which is regular (by condition 1 of the inductive defn of regular languages).

**Case 2:** $\{\Lambda\}^R = \{\Lambda^R\} = \{\Lambda\}$, which is regular (by condition 2 of the inductive defn of regular languages).

**Case 3:** For all $a \in \Sigma$, $\{a\}^R = \{a^R\} = \{a\}$, which is regular (by condition 3 of the inductive defn of regular languages).

**Case 4:** $L$ is a regular language.
   - IH: $L^R$ is regular.
   - NTS: $(L^*)^R$ is regular.

By IH, $L^R$ is regular. It follows by condition 4 of the inductive defn of regular languages that $(L^R)^*$ is regular. And by Lemma 1, $(L^R)^* = (L^*)^R$.

**Case 5:** $L_1$ and $L_2$ are regular languages.
   - IH: $L_1^R$ and $L_2^R$ are regular.
   - NTS: $(L_1 \cup L_2)^R$ and $(L_1 L_2)^R$ are regular.

By IH, $L_1^R$ and $L_2^R$ are regular. It follows by condition 5 of the inductive defn of regular languages that $L_1^R \cup L_2^R$ and $L_2^R L_1^R$ are regular. It is clear that $L_1^R \cup L_2^R = (L_1 \cup L_2)^R$, and by Lemma 2, $L_2^R L_1^R = (L_1 L_2)^R$. 
