Problem 1. Assume that $M = (Q, \Sigma, \delta, q_0, F)$ is a DFA with $L(M) = L$.
We construct an NFA $N = (Q_N, \Sigma, \delta_N, q_{0_N}, F_N)$ as follows.

1. $Q_N = (\{0, 1\} \times \emptyset \times \emptyset) \cup \{ q_{0_N} \}$
   ($q_{0_N}$ is the start state of $N$ that is not in $\{0, 1\} \times \emptyset \times \emptyset$)

2. The set of accepting states of $N$:
   $F_N = \{ (1, q, q) \mid q \in Q \}$.

3. The 1-step nondeterministic transition function $\delta_N$ of $N$:
   $\delta_N : Q_N \times (\Sigma \cup \{\varepsilon\}) \to P(Q_N)$
   is defined as follows:
   (i) $\delta_N(q_{0_N}, \varepsilon) = \{ (0, q, q) \mid q \in Q \}$
   (ii) $\delta_N((0, s, q), a) = \{ (0, \delta(r, a), q) \}$
       for all $q, r \in Q$ and $a \in \Sigma$
   (iii) $\delta_N((0, r, q), \varepsilon) = \{ (1, r, q) \}$
       for any $r \in F$ and $q \in Q$
   (iv) $\delta_N((1, r, q), a) = \{ (1, \delta(r, a), q) \}$
       for all $q, r \in Q$ and $a \in \Sigma$
   (v) For all other combinations of argument-values of $\delta_N$, the $\delta_N$-values are $\emptyset$

(idea of the construction is given below.)
N starts out in the start state \( q_0 \), and immediately makes a nondeterministic guess for some state \( q \) of \( M \) from \((0, q, q)\) to jump to. The idea is that the "0" indicates that \( N \) is entering the first phase of its computation, in which it reads a portion (suffix) of its input string corresponding to \( q \) in the definition of \( K \). It jumps to any state \( q \) of \( M \) but it also remembers which state it jumped to. Every state \( N \) even moves to from this point on will have the form \((a, r, q)\) for some \( a \in \{0, 1\} \) and \( r \in \Sigma \), but \( q \) the same \( q \) that it has just jumped to. The 3rd coordinate \( q \) represents the memory \( \gamma \) where it first jumped and it will never forget or change this part of its state.

Intuitively, the state \( q \) is a guess made by \( N \) for the state that \( M \) would be on after reading \( x \) (which \( N \) has not seen yet — so it is just a guess).

Then, \( N \) starts reading symbols and essentially mimicking \( M \) on those input symbols — this corresponds to the transitions listed as (ii).
At some point, our deterministic choice, \( N \) decides that it may be time to move to the second phase of its computation, reading the second part of its input, which corresponds to the string \( x \) in the definition of \( L \).

It can only make this nondeterministic move to a state \( y \) in \( \text{Run} \) \((4, 9, 9)\), when \( y \) is an accepting state of \( M \).

The reason is that \( N \) only wants to accept \( y \) when \( M \) accepts \( y \), so \( M \) should be at an accepting state at this point.

This corresponds to the statements listed as (iii).

Finally, in the second phase of its computation, \( N \) simulates \( M \) on the second part of its input, which corresponds to the string \( x \). \( N \) accepts only for states of the form \((4, 9, 9)\), because those are the states that indicate that \( N \) made the correct guess on its first step for the state that \( M \) would be in on input \( x \).

Not a formal proof, but the intuitive description should give that \( L(N) = L \).
Problem 2.

(a) To prove the given statement, we substitute \( r = r_1 r_2^* \) into the "re-equation" \( r = r_1 + r r_2 \) and show that the two sides of r.e.s. are equal as r.e.s. The "equation" \( r = r_1 + r r_2 \) after substitution \( r = r_1 r_2^* \) is:

L.H.S. is \( r_1 r_2^* \); Are these two r.e.s. \( \Rightarrow \) equal as r.e.s.? That is, is \( L(r_1 r_2^*) = L(r_1 + r_1 r_2^* r_2) \)?

You can prove the above (equality in \( \equiv \)) by showing:

\[ L(r_1 r_2^*) \leq L(r_1 + r_1 r_2^* r_2) \]

and \( L(r_1 r_2^*) \equiv L(r_1 + r_1 r_2^* r_2) \) formally.

Or, observe that:

\[ r_1 + r_1 r_2^* r_2 = r_1 (1 + r_2^* r_2) \]

"left-distributively of r.e.s."

\[ = r_1 (1 + r_2^+) \Rightarrow \text{c.c. } r_2^* r_2 \]

\[ = r_1 r_2^* (c^+ r^+ = r \text{ for every r.e. r}) \]
(b) To prove the given statement, we assume that $r$ satisfies the "property", i.e., $r = r_1 + r r_2$
with $x \not\in L(r)$. What are $r$ (i.e., $L(r)$) look like?

Let $x \in L(r)$ be arbitrary. We show that

$x \in L(r_1, r_2^{n+1})$ by a (proof) induction argument.

Suppose that $x \not\in L(r_1, r_2^n)$

i.e., $x \not\in L(r_i, r_2^j)$ for any $j$. (1)

We can then show, using an induction, that

for every $n \geq 0$, $x \in L(r_1, r_2^n)$.

basis: $n = 0$. By assumption $x \in L(r)$

(since $x \in L(r_1, r_2^0)$)

Induction step. Assume that $n \geq 0$ and

$x \in L(r_1, r_2^n)$. (2)

Observe that:

$r_2^{n+1} = (r_1 + r r_2) r_2^n$

$= r_1 r_2^n + r r_2^{n+1}$

Now, $x \in L(r_2^n)$ and $x \not\in L(r_2^n)$,

(3) above $\quad$ (2) above

We must have $x \in L(r_2^{n+1})$, as desired.

This completes the induction step.

By induction, we have shown that

$\forall n \geq 0, x \in L(r_2^n)$. (4)
Now, we examine the outcome of:
\[ \forall n \geq 0 \ x \in L(r^n) \].

Recall no assumption that \( r \notin L(r_2) \), so every string in \( L(r_2) \) must have length \( \geq 1 \). It follows that every string in \( L(r^n) \) must have length \( \geq n \) for all \( n \).

How can the string \( x \) (\( x \in L(r^n r_2^n) \)) for \( \forall n \geq 0 \)?

\( x \) must have length \( \geq n \) for every \( n \).

impossible.

So, (supposition in (2) is false).

That is, \( x \in L(\bar{r} r_2^\ast) \).

Thus, we have shown that:
\[ \forall x, x \in L(r) \Rightarrow x \in L(\bar{r} r_2^\ast) \].

Combining with part (a), we have:
\[ \forall x, x \in L(r) \Leftrightarrow x \in L(\bar{r} r_2^\ast) \]
\( \Rightarrow \), \( r = \bar{r} r_2^\ast \) as required.
3. the language $L_2$ is not regular by using the pumping lemma.

Suppose $L_2$ were regular.

Let $n$ be the pumping lemma constant.

Consider $z = 0^n 2^n 1^n \in L_2$ with $121 = 2n + 1 \geq n$.

Then, consider, for all $u, v, w \in \{0, 1, 2\}^*$

with $z = uvw$ Satisfy $|uv| \leq n$ and $|v| \geq 1$,

i.e., $u = 0^\alpha$, $v = 0^\beta$,

$w = 0^{n-\alpha-\beta} 2^n \ eta \geq 1$

$\underline{0^n 2^n 1^n} = \underline{u} \underline{v} \underline{w}$

Then, let $i = 0$,

$uv^n w = uw = 0^\alpha 0^{n-\alpha-\beta} 2^n$

$= 0^{n-\beta} 2^n$

$\alpha_1 \#_2 \beta \#_3 1^n$

$n-\beta < n$ (since $\beta \geq 1$)

so $uv^n w \notin L_2$.

What about $L_1$?

As discussed in Lecture, $L_1 = \{0, 13^* \}$

(since $L_1$ is regular).

How do prove that $L_1 = \{0, 13^* \}$?

Suffice to show that $\{0, 13^* \} \subseteq L_1$

i.e., $\forall x \in \{0, 13^* \}, x \in L_1$

i.e., $\forall x \in \{0, 13^* \}, x = uv^n w$ for some $u, v \in \{0, 13^* \}$

with $\#_2(u) = \#_2(v)$.
Again, as hinted in lectures,
for every $x = a_1 a_2 \cdots a_n \in \{0, 1\}^n$
for some $n \geq 0$ and $a_i \in \{0, 1\}$ for $i = 1, 2, \ldots, n$.
(Clearly, when $n = 0$, we have $x = \varepsilon$, and $\varepsilon \subseteq \{0, 1\}$)

Consider:
1. Place two indices $i$ and $j$ as shown

\[ \begin{array}{c}
\text{index } i \\
 i = 0 \quad 1 \quad a_1 \quad a_{n-1} \quad a_n \quad 1 \quad 0 \quad j
\end{array} \]

2. Define two functions $l_i$ (left-to-right for country $0$s) and $r_i$ (right-to-left for country $1$s) on indices $i$ and $j$, respectively.

Observe that:
\[ \begin{cases} l_i(i = 0) = 0 & \text{and} \quad r_i(j = 0) = 0 \\ l_i(i = n) \geq 0 & \text{and} \quad r_i(j = n) \geq 0 \end{cases} \]

2.2 \begin{align*}
\text{both } l_i & \text{- and } r_i \text{- functions are} \\
\text{"step"-functions: increase (or decrease) by one unit.}
\end{align*}

2.3 \begin{align*}
\text{both } l_i & \text{- and } r_i \text{- functions are} \\
\text{increasing functions (and stabilized "eventually")}
\end{align*}

These observations yield that there exists (a unique) index value where these two functions "meet".
That is, where can decompose $x$ into $U \cup V$ with $\#_0(U) = \#_1(V)$. Can you provide a formal argument from above?
Problem 4.

We know that every finite language is regular, so it suffices to prove that if $K$ is regular, then $K$ is finite; equivalently, if $K$ is infinite, then $K$ is non-regular.

So, we assume that $K$ is infinite. To prove that $K$ is non-regular, we apply the pumping lemma on $K$.

Suppose that $K$ were regular.

Let $n$ be the pumping lemma constant.

Choose $z \in K$ with $|z| \geq n$.

How? Yes, $K \subseteq L (= \{ a^i | i \geq 3 \})$ but, how do we know that $\exists z \in K$ with $|z| \geq n$?

From (1) above that $K$ is infinite,

so for the $n$ specified, $\exists z \in K$ with $|z| \geq n$.

As $K \subseteq K$, such $z$ (with $|z| \geq n$) must be of the form $z = a^i$ with $121 = i^2 \geq n$.

Now, follow the pumping lemma (lecture notes—examples, Problem 3, Suggested Example 1.76 in text) to produce a contradiction.