Problem 2.

(a) Basic idea: The machine nondeterministically guesses (when reading an input symbol 1) and verifies a string of 1 \((0+1)^* 1\) that is forthcoming.

\[ M : \text{start} \rightarrow q_{\text{start}} \rightarrow q_1 \rightarrow q_{\text{odd}} \rightarrow q_{\text{even}} \rightarrow q_{\text{accept}} \]

- \(q_{\text{start}} : \text{Nondeterministically wait or guess on an input symbol of 1} \)
- \(q_1, q_{\text{odd}}, q_{\text{even}}, q_{\text{accept}} : \text{Having encountered an input symbol of 1, verify if a string of } 1(0+1)^* 1(0+1)^* \text{ appears.} \)

Can verify that \(\forall x \in \{0,1\}^* \, M \text{ accepts } x \iff x \in (0+1)^* 1(0+1)^* 1(0+1)^* \).

(b) The given language is the disjoint union of the two languages:

\[ L_a = \{ x \in \{0,1,2\}^* \mid \#_a(x) \geq 3 \text{ and } 0 \leq c(x) \leq 2 \} \]

\[ L_b = \{ x \in \{0,1,2\}^* \mid \#_b(x) \geq 3 \text{ and } 0 \leq a(x) \leq 2 \} \]

Basic idea for constructing a DFA \(M_a\) accepting \(L_a\): each state has 3 components to record \(\#_a(x), \#_b(x), \text{ and } \#_c(x)\) of the input \(x\) consumed so far.

- \(Q = \{ (i,j,k) \in \mathbb{N}^3 \mid i \leq 3, j \leq 2, \text{ and } k \leq 2 \} \setminus \{0\} \)
- \(q_{\text{start}} : (0,0,0)\)
- Set of accepting states: \(\{(3,j,k) \mid 0 \leq j, k \leq 2\}\)
1-step transition function: $S: \mathbb{Q} \times \{a, b, c\} \rightarrow \mathbb{Q} \times \{a, b, c\}$
defined as:

$S((i, j, k), a) = \begin{cases} (i+1, j, k) & \text{if } i \leq 2 \\ (i, j, k) & \text{if } i = 3 \end{cases}$

$S((i, j, k), b) = \begin{cases} (i, j+1, k) & \text{if } j \leq 1 \\ \text{error} & \text{if } j = 2 \end{cases}$

$S((i, j, k), c) = \begin{cases} (i, j, k+1) & \text{if } k \leq 1 \\ \text{error} & \text{if } k = 2 \end{cases}$

For $d \in \{a, b, c\}$, $S((\text{error}, d)) = \text{error}$

A DFA $M_b$ accepting $L_b$ is similar.

A desired DFA accepting $L_a \cup L_b$ is:

Start $\rightarrow (q_0) \xrightarrow{\epsilon} M_4$

(c) Given that a FA $M$ accepting $L$, we construct an FA $M'$
accepting $\text{half}(L)$. The basic idea is that $M'$ keeps
track of two states in $M$ (using two coordinates/tracks in
a state of $M'$):

1. **Forward Simulation**: For each input symbol read in $M'$, $M'$ uses first
coordinate/back to simulate $M$ on that symbol.
   (At the same time, $M'$ simulates the backward simulation
   starting at $q_{accept}$ in $M$.)

2. **Backward Simulation**: $M'$ uses second coordinate/back to simulate $M$
   backwards on a guessed symbol.

$M'$ accepts an input $\alpha$ iff the forward simulation (on $\alpha$) and the backward simulation (on a
guessed $\beta$, $y_1 = \alpha x$) are in the same state of $M$. 
Finally, assume that NFA $M = (Q, \Sigma, \delta, q_0, \text{accept})$ accepts $L$.

Construct a NFA $M' = (Q', \Sigma, \delta', q'_0, F')$ as follows:

$$Q' = Q \times Q,$$

$$q'_0 = (q_0, \text{accept}),$$

$$F' = \{(q, q') \mid q \in Q\}.$$

and $\delta' : Q' \times \Sigma \rightarrow 2Q'$ is defined as:

$$\forall (p, q) \in Q \times Q, \forall a \in \Sigma,$$

$$\delta'(p, q, a) = \{ (rs) \in Q \times Q \mid r \in \delta(p, a) \land$$

$$\exists b \in \Sigma \exists q \in \delta(s, b) \}$$

Problem 3.

1. $M \rightarrow M_i$:
   - Follow lecture notes: proof of the equivalence between NFAs with E-transitions and NFAs without E-transitions.
   - Compute E-closures for individual states $\delta$ in $M$.
   - Compute E-closures for subsets of states.
   - Relate the transition function $\delta_2$ of $M_i$ to the multi-step transition function $\delta$ of $M$.

2. $M_i \rightarrow M_2$, using Subset Construction:
   - Relate the transition function $\delta_2$ of $M_2$ to the transition function $\delta_1$ of $M_i$.
Problem 4.
Consider a restricted form of the language \( J = \{ x \in \Sigma^* \mid I_{x \in L}(x = \text{disc}(y)) \} \):
\[
J' = \{ uv \mid u, v \in \Sigma^* \land \exists \epsilon \in L \}
\]

We show that the language \( J' \) is regular.

Then, you can modify the construction for \( J' \)

to an NFA for \( J \).
Can you "reverse" my ideas to prove non-regularity of \( K \)?

Without loss of generality, we assume that \( L = L(M) \) for some
DFA \( M = (Q, \Sigma, \delta, q_0, F) \), and we construct an
NFA \( M' = (Q', \Sigma, \delta', q_0', F') \) as follows.

The idea of constructing \( M' \):

1. Define \( Q' = Q \times \{ 0, 1 \} \)
   use two labels:
   \( 0 \)-labels for simulating \( M \),
   before nondeterministically guessing
   of the deleted symbol.
   \( 1 \)-labels for simulating \( M' \)
   after such guessing.

2. Define \( q_0' = (q_0, 0) \), and
   \( F' = \{ (q, 1) \mid q \in F \} \)
3. The transition function $S'$ of $M'$ is defined as:

\[ S': Q \times \Sigma \to Q \] 

\[ S'(q, a) = \begin{cases} S(q, a), & a \in \Sigma \\ S(q, 0), & a \in \Sigma \end{cases} \]

- Simulate the transition of $S$ (using 0-labels) before a nondeterministic guess, then a deleted symbol
- Nondeterministically guess a deleted symbol, and
- Enter the simulation of the transition of $S$ (using 1-labels)

\[ S'(q, 0) = \begin{cases} S(q, 1), & a \in \Sigma \\ S(q, 0), & a \in \Sigma \end{cases} \]

- Simulate the transition of $S$ after such a nondeterministic guess

\[ S'(q, 1) = \begin{cases} S(q, a), & a \in \Sigma \\ \emptyset, & a \in \Sigma \end{cases} \]

- Accept with 1-labels

\[ S'(q, 1) = \emptyset \quad \forall q \in Q \]

**Why are $M'$ work?** \( L(M') = \Gamma' \) 

**Problem 5:** Exercise 1.18 (b) and (i)

h. \( \Sigma^* \cup 1 \cup \Sigma^* \cup 1 \cup \varepsilon \)

i. \( (1 \Sigma^*)(1 \cup \varepsilon) \)

**Problem 5:** Exercise 1.21 (a)

1.21 In both parts we first add a new start state and a new accept state. Several solutions are possible, depending on the order states are removed.

a. Here we remove state 1 then state 2 and we obtain

\[ a^*b(a \cup b)*b^* \]

**Exercise 1.22 (b)** 1.22 b. \( \#(a \cup b)^* \# \)
Problem 7.

(a) The language $L_1$ is not regular by Pumping Lemma (for regular languages). Suppose $L_1$ were regular. Let $p \in \mathbb{N}$ be a positive integer constant given in the pumping lemma.

Consider $z = a^n b^n a^n \in L_1$ and $|z| = n + n + n^2 \geq n$.

Then, we need to consider all possible $u, v, w \in \{0, 1\}^*$. Such that:

- $|z| = uvw$
- $|uv| \leq n$
- $|v| \geq 1$

Notice that, since $|uv| \leq n$, we have:

- $u = a^\alpha$
- $v = a^\beta$ with $\alpha + \beta \leq n$ and $\beta \geq 1$
- $w = a^{n - (\alpha + \beta)}b^n a^n$

Now, for $i = 0$, the string $uu^i w = uw = a^{n - (\alpha + \beta)}b^n a^n$

The string $uu^i w = a^{n - \beta} b^n a^n \in L_1$ if $(n - \beta) \cdot n = n^2$.

This is not possible since $\beta \geq 1$.

Thus, $uu^i w \notin L_1$.

(b) $L_2 = \{ b^2 a^n b^n a^3 | n \geq 3 \}$ is regular. A regular expression denoting $L_2$ is:

$\{ b^2 a^n b^n a^3 \}$
(c) \( L_3 = \{ a^{3k} \mid k \geq 0 \} \) is not regular by Pumping Lemma.

The proof is similar to the non-regularity of \( \{ a^{2k} \mid k \geq 0 \} \) by Pumping Lemma.

Sketch: Suppose \( L_3 \) were regular. Let \( n \) be the "Pumping Lemma constant".

The choice of \( z = a^{n^3} \) will contradict the Pumping Lemma.

Consider all \( u, v, w \in \Sigma^* \) such that \( z = uvw \) with \( |v| \neq 1 \) and \( |uv| \leq n \).

Then, consider \( z = u^i v^j w \). We claim that \( u^i v^j w \notin L_3 \).

What is \( |uv^2w| \)? (Is it a "perfect cube"?)

We want to show that \( |uv^2w| < (n+1)^3 \).

\[ |uv| = |u| + |v| = n^3 + |v| \]

\[ |uv^2w| = |uv| + |v^2| = n^3 + |v|^2 \]

Observe: \( |v|^2 \leq n \) since \( |v| \geq 1 \).

Also true: \( |v| \leq n \) since \( |v| \leq n \).
(a) The language $L_1 = \{a^i b^j | i, j \geq 0, j = i^3 \}$ is not regular. Apply the pumping lemma on $L_1$ to prove the non-regularity of $L_1$. 

Suppose $L_1$ were regular. 
What is $L_2 = \{a^i b^{i^3} | i \geq 0, j \neq i^3 \}$? 

(6) Use the closure properties for regular languages to show that the language $L_2 = \{a^i b^{i^3} | i \geq 0, j \neq i^3 \}$ is not regular. 

Suppose $L_2$ were regular. 
What is $L_2^c = \{a^i b^j | j \neq i^3 \}$? 

(Be careful, $L_2$ is not regular yet)$\{a^i b^{i^3} | j \neq i^3 \}$. 
The complementation needs to be applied 
so: - the ordering of $a$s and $b$s 
- the condition $\#(b) \neq \#(a)^3$ 

So, we consider $L_2 \cap a^* b^*$ 

(i.e., $L_2 \cap \{a^* b^* \}^{\leq3} \{a^* b^* \}^{\neq3} \{a^* b^* \}$ 

Then, we can see that 

$L_2 \cap a^* b^*$ 

is not regular, why? 

$L_1$ is proven to be non-regular in part (a).
Problem 9.

(a) True.

Let $\Sigma = \{0, 1, 3\}$ and $A$ be any non-regular language over $\Sigma$ (for instance, $A = \{0^i \mid i \geq 3\}$).

Then, let $B = A = \Sigma^* - A$.

Note that $B$ is not regular. (Why?)

Now, $A \cup B = \emptyset$, which is regular.

(b) True.

Let $\Sigma = \{0, 3\}$ and $A$ be any non-regular language over $\Sigma$ such that $0 \in A$.

For example, $A = \{0^i \mid i \geq 3\}$.

Then, let $B = A \cup \{3\}$. (Important)

Note that $B$ is not regular. (Why?)

Now, $AB = A \cdot (A \cup \{3\}) = A^* - A$.

Can you see the set-equality or language-equality?

(c) True.

Let $\Sigma = \{0, 3\}$ and $A$ be any non-regular language over $\Sigma$ such that $0 \in A$.

Then, $A^* = \Sigma^*$. (Important)

Can you see the set-equality or language-equality?