1. $f: \Sigma^* \to \Sigma^*$, related language $L_f \subseteq \Gamma^*$

a) we will define alphabet Γ^*: $\Gamma^* = \Sigma \cup \mathbb{N} \cup$ ("(", ")", ","). we will define a related language:

$L_f = \{(x,y,z) \mid x \in \Sigma^*,\ y \in \mathbb{N},\ z \in \Sigma\}$.

↳ $L_f = (x,y,z)$ where $x \in \Sigma^*$ and is the input for f, $y \in \mathbb{N}$ and represents the index of $z \in \Sigma$, which is a character of the output string.

b) a computer that decides L_f will be able to compute f in the following steps:

1. The computer will accept the string if the character at the index y of the output string $f(x)$ matches z where $y \in [0, |x|)$.
2. The computer will accept (x,y,z) iff the character at the y-th index of $f(x)$ is z.
3. After looping through, we should have $|x|$ accepted strings.
4. We can construct $f(x)$ by concatenating z in order of the indices from $y=0$ up to $y=|x|-1$.
5. $\therefore$ we have a computer that decides L_f to compute f.

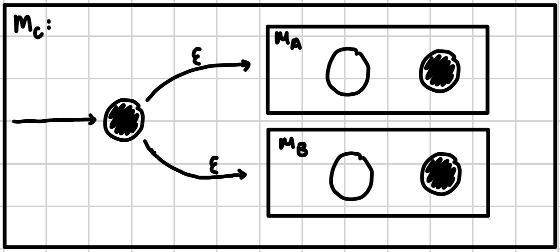
given f, we can decide L_f in the following steps:

1. If a string is not in the form (x,y,z), it will be rejected.
2. we can check if the y^{th} index of the known $f(x)$ output string is equal to z. If so, the computer accepts and rejects if not.
   ↳ if f(x)[y].equals(z)? accept : reject
3. $\therefore$ we can decide L_f given f.

2. a) $\Rightarrow$ given a regular language, we will show that it can be recognized by an all-paths-NFA. by the equivalence thm, if a language is regular then there is a FA that recognizes the language. A FA is an instance of an APNFA with only one path for an input, and inputs are only accepted if the path ends in the accept state. $\therefore$ all languages are recognized by an APNFA.

$\Leftarrow$ if L is recognized by an APNFA, L is a regular language.
we must show that an APNFA can be written as a normal NFA.
Let $N = (Q, \Sigma, \delta, q_0, F)$ where N represents an arbitrary APNFA that recognizes L. we will construct a standard NFA with $M = (Q', \Sigma', \delta', q_0', F') \mid$ M recognizes the same language as N. we need the powerset of Q | it consists of all subsets of Q: $Q' = P(Q)$. $\Sigma' = \Sigma$ because the alphabets of M and N must match. $q_0' = \{q_0\}$ | M and N have the same start state. $F' = P(F)$ since M should only accept if its accepting state consists of solely accepting states of N.
we will define $\delta'(r,w) = \begin{cases} 0, & \text{if } r \in R \mid \delta(r,w) = 0 \\ q \in Q \text{ and } q \in E(\delta(r,w)) \text{ for } r \in R, & \text{otherwise} \end{cases}$
A language that is recognized by a NFA is a regular language. Since we constructed a standard NFA M that recognizes L, which is also recognized by N, L must be a regular language.

2. b) from part a, we have proved that any regular language is accepted by an APNFA. Since A and B are regular languages, they will be recognized by APNFAs. NTS that $C = A \cap B$ will be recognized by an APNFA and thus is a regular language. An APNFA that recognizes C must accept strings that are accepted by both M_A and M_B, which will be our APNFAs that accept A and B. We will call M_C the APNFA that accepts $C = A \cap B$.



* colored nodes are accept states

A string is accepted by M_C iff all paths end in an accept state, which means M_A and M_B end in accept state. The start state of M_C must be an accept state since if neither epsilon transition is taken, but the string is recognized by M_A and M_B then it will also be accepted by M_C $\therefore$ M_C recognizes $A \cap B$ and it follows that C is a regular language.

c) NFAs may have several valid computations for an input, so machine M and M_{flip} are not necessarily complements, but L and L_{flip} are. If a string is not in L, then computations in M for that string cannot end in an accept state. non-accept states in M are accept states in M_{flip}, the string that led to non-except in M is a member of L_{flip}. $\therefore$ $L_{flip} = \overline{L}$ as L_{flip} contains the strings rejected by M.

3. Proof by contradiction. Let $L = \{x : x \in \Sigma^*, x = x^R\}$ be a regular language that represents all palindromic strings. Since we assume L is regular, it has a pumping length p. We will select string $w \in L \mid w = a^p b a^p$ where $a, b \in \Sigma$ and $|w| = 2p+1 > p$. Let $w = xyz$. There are 3 scenarios: $(k, m, n \in \mathbb{Z})$

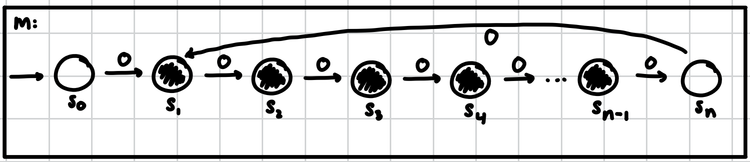
1. $x = a^k$, $y = a^m$, $z = a^n b a^{k+m+n}$ where $k, m > 0$, $n \geq 0$, and $k+m+n = p$
2. $x = a^k b a^n$, $y = a^m$, $z = a^{k-n-m}$ where $k, m > 0$, $n \geq 0$, and $k = p$
3. $x = a^k$, $y = a^n b a^m$, $z = a^r$ where $k > 0$, and $n, m \geq 0$, and $k+n = m+r+p$

We can show that $xy^iz \in L$ does not hold for all values of i.

1. Pumping on scenario 1 for $i = 2 \mid xy^2z$ yields $xy^2z = a^k a^{2m} a^n b a^{k+m+n} = a^{k+2m+n} b a^{k+m+n}$.
   $\rightarrow (xy^2z)^R = a^{k+m+n} b a^{k+2m+n} \neq xy^2z$, so $xy^2z \notin L$.
2. The pumping lemma has conditions $|xy| \leq p$ and $|y| > 0$. Scenario 2 does not fulfill $|xy| \leq p$ since $|xy| = k+n+m+1 > k > p$, so this scenario is not possible.
3. The pumping lemma has condition $|xy| \leq p$. Since $w = a^p b a^p$, $|xy| > p$, so scenario 3 is not possible.

∴ only scenario 1 works, but pumping y yields a strong not in L $\Rightarrow\Leftarrow$. L is not regular.

4. a) A language is regular if it can be represented as a DFA. Suppose $M = (Q, \Sigma, \delta, q_0, F)$ is a DFA. Let $Q = s_0, s_1, s_2 \ldots s_n \mid$ there are $n+1$ states.



* colored nodes are accept states

$\Sigma = \{0\}$. $q_0 = s_0$. $F = \{s_1, s_2, \ldots s_{n-1}\}$. $\delta(s_0, 0) = \{s_1\}$, $\delta(s_1, 0) = \{s_2\}$, $\delta(s_2, 0) = \{s_3\} \ldots$ $\delta(s_{n-1}, 0) = \{s_n\}$, $\delta(s_n, 0) = \{s_1\}$. Since we could construct a DFA for L_n, L_n is a regular language.

b) Proof by contradiction. Let $L = \{x : x \in \Sigma^*, |x| = \text{primes}\}$ represent a language containing all strings whose length is a prime number. Assume L is regular with pumping length p. Since there are infinite prime numbers, we can select a string $w \in L$ where $|w|$ is prime and $|w| > p$. Let $|w| = a$ and $w = xyz$. Since the alphabet only has one 0, there is only one scenario with a possibility of pumping by assigning a random number of 0s to x, y, z while fulfilling $|y| > 0$, $|xy| \leq p$. Consider $w = xyz \mid xy^iz \in L$ does not hold for some values of i. Assume that $|y| = n > 0$. Let $i = a+1$ with $xy^{a+1}z$ such that $|xy^{a+1}z| = |w| + y^n = a + a|y| = a + an = a(1+n)$. $xy^{a+1}z \notin L$ since its length $a(1+n)$ is a factor of a and is not prime. $\Rightarrow\Leftarrow$. L is not regular.

not 100% sure this works