Question 1: Close

a. $A = \{a\}^*$ for some $a \in \Sigma$

b. $A = \Sigma^*$

c. We know A is regular, so we can assume that we have a DFA $M = \{Q, \Sigma, \delta, s, F\}$ such that L(M) = A. Now we want to construct NFA $M' = \{Q', \Sigma, \delta', s', F'\}$ where $L(M') = Close(A)$

Let

$$\begin{aligned} Q' &= Q \times \{0,1\} \\ \delta'((p,0),a) &= \begin{cases} (\delta(p,a),0) & \text{for all } p \in Q, a \in \Sigma \\ (q,1) & \text{for some } q = \delta(p,b) \text{ where } b \in \Sigma \text{ and } q \neq \delta((p,a) \end{cases} \\ \delta'((p,1),a) &= (\delta(p,a),1) \quad \text{for all } p \in Q, a \in \Sigma \\ s' &= (s,0) \\ F' &= F \times \{1\} \end{aligned}$$

Suppose we have a string x in NFA M'. States $(p,0) \in Q \times \{0\}$ represent the state it ended up every transition follows transition function of NFA M. State $(q,1) \in Q \times \{1\}$ represent the state it ended up if exactly one transition does not follow transition function of δ NFA M.
It should be noted that in transition from $(p,0) \in Q \times \{0\}$ to $(q,1) \in Q \times \{1\}$, q should be reachable from p(This is because the result have to be close). Suppose we are taking character a in that transition, we would also have $q \neq \delta(p,a)$, this is because Close(A) does not necessarily include A.

Question 2:   Close But Not Quite

Given a language A over alphabet Σ,
define $CloseButNot(A) = \{$x|x and are close, $y \in A, x \notin A\}$

Let Not(A) = $\{x|x \notin A\}$
Then, $CloseButNot(A) = Not(A) \cap Close(A)$ Since we know that if A is regular then Close (A) is regular and intersection of two regular language is also regular. We only need to show that Not(A) is regular.

If A is regular, then we have DFA M = $(Q, \Sigma, \delta, s, F)$ that accepts A. Then DFA $M' = (Q, \Sigma, \delta, s, F')$ where $F' = Q \backslash F$ accepts Not(A). This means that if A is regular language, then Not(A) is also regular language.

In conclusion, if A is regular language, then $CloseButNot(A)$ is also regular language.

Question 3: All-NFA

An all-NFA M is a 5-tuple $(Q, \Sigma, \delta, s, F)$ that accepts $x \in \Sigma^*$ if every possible state that M could be in after reading input x is a state from F.

We first prove that every all-NFA recognizes a regular language:

We construct DFA $M' = (Q', \Sigma, \delta', s', F')$ such that

$$\begin{aligned} Q' &= P(Q)\text{(power set of Q)} \\ \delta'(q', a) &= \{\delta(q, a) | q \in q'\} \text{ for all } a \in \Sigma, q' \in Q' \\ s' &= s \\ F' &= P(F) \end{aligned}$$

Claim: $L(M) = L(M')$

Proof. By definition we have $\delta'(q', a) = \{\delta(q, a) | q \in q'\}$ for all$a \in \Sigma, q' \in Q'$ therefore,

$$\hat{\delta}'(q', x) = \{\hat{\delta}(q, x) | q \in q'\} \text{ for all } x \in \Sigma^*, q' \in Q' \tag{1}$$

□

therefore, the following statements are equivalent:

$x \in L(M) \Leftrightarrow \{\hat{\delta}(s, x)\} \subseteq F \Leftrightarrow \hat{\delta}'(s, x) \in F' \Leftrightarrow x \in L(M') \Leftrightarrow$ x is a regular language

Now we prove every regular language is recognized by some all-NFA:

A DFA is a all-NFA, and every regular language is recognized by some DFA. Therefore every regular language is recognized by some all-NFA.

In conclusion, we have shown that all-NFAs recognize the class of regular languages