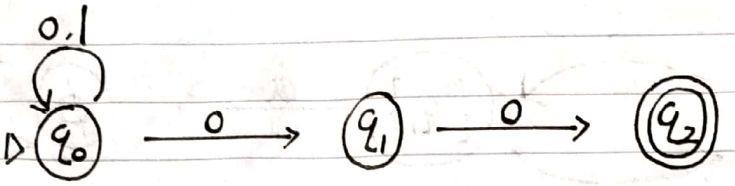
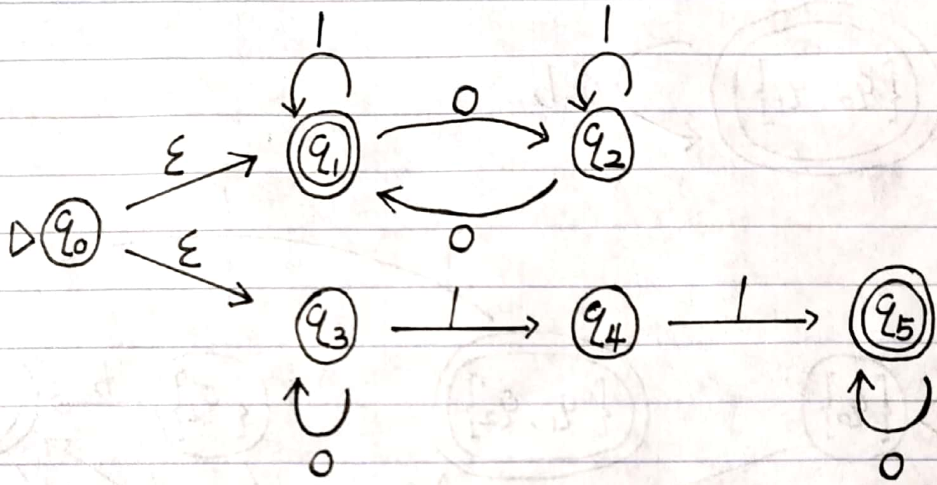
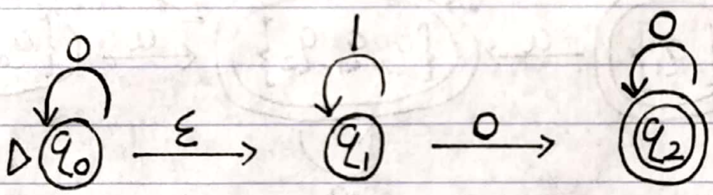
1(a).


0,1
q0
0
q1
0
q2

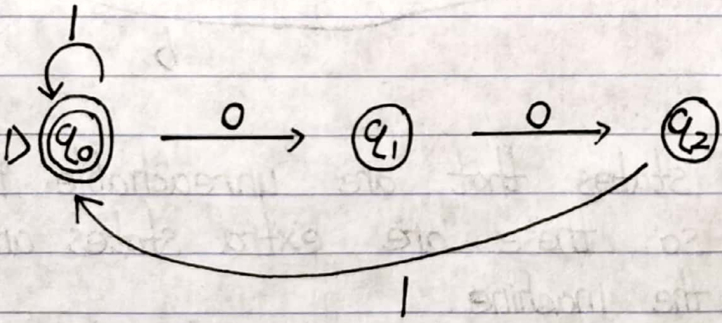
1(b).


ε
ε
q0
1
q1
0
0
1
q2
q3
0
1
q4
1
q5
0

1(c).


0
q0
ε
1
q1
0
0
q2

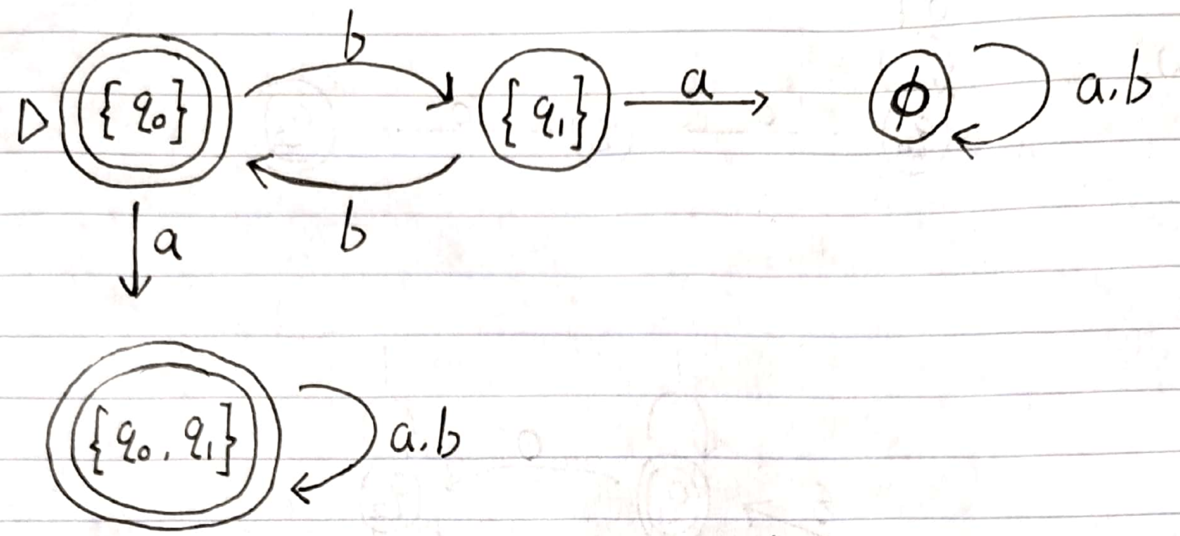
1(d).


1
q0
0
q1
0
q2
1

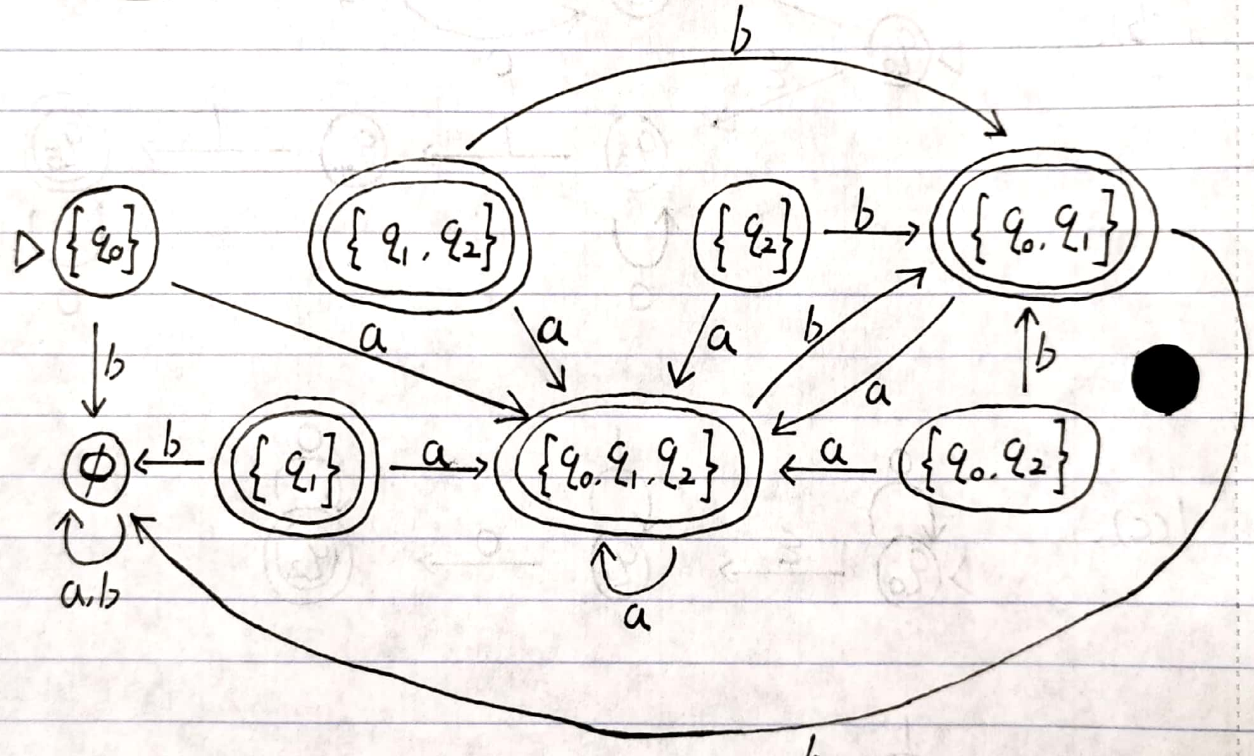
1(e).


q0
0

2(a).

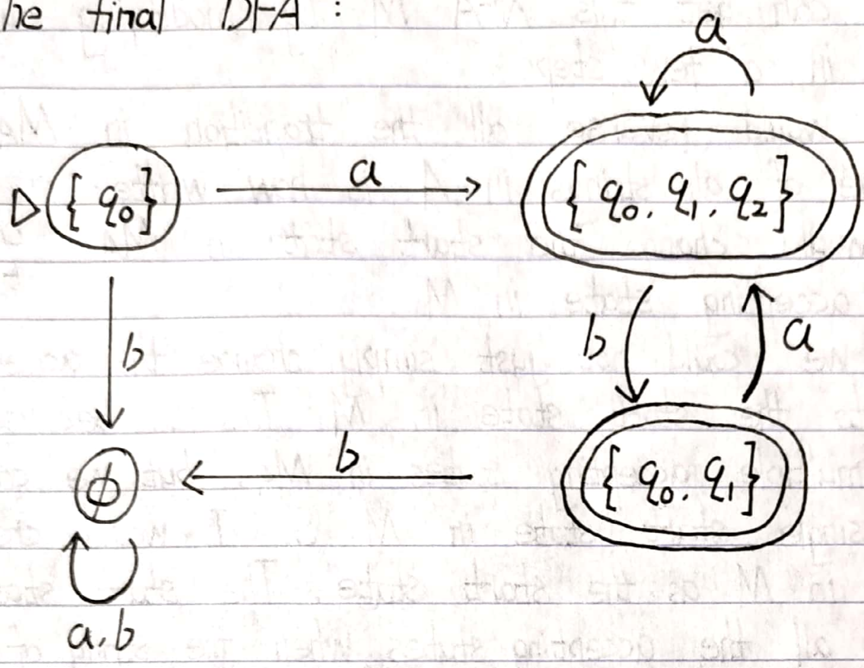


2(b).

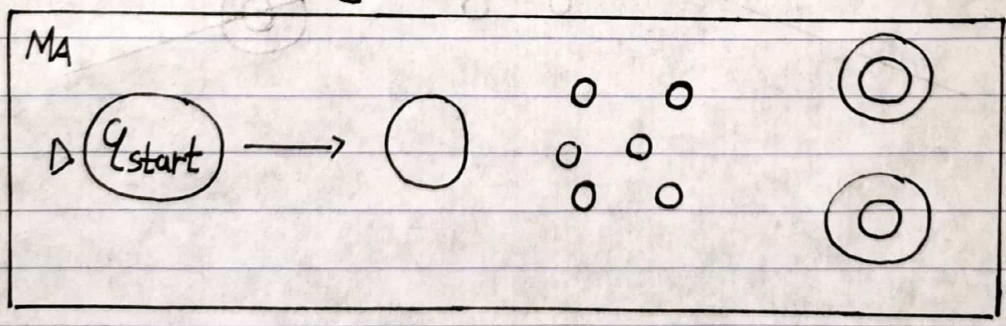


there are 4 states that are unreachable from the start state, so these are extra states and can be removed from the machine.

The final DFA :



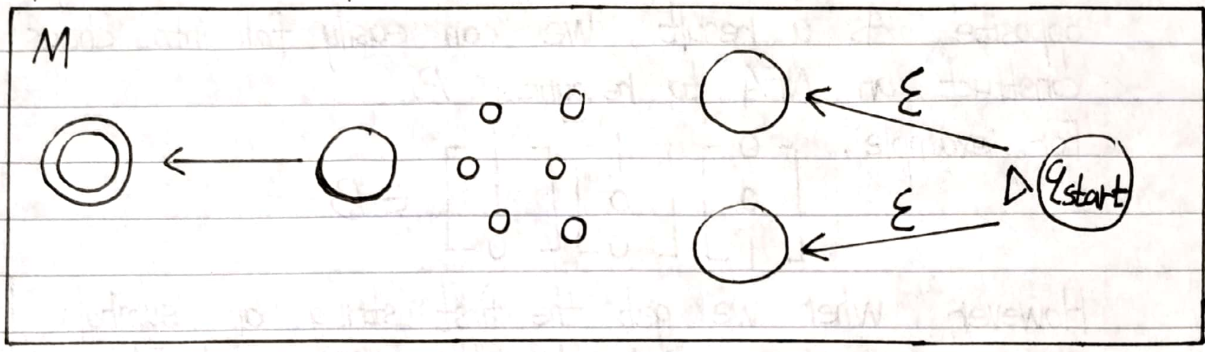
3(a). If language A is regular, then there is an NFA M_A that recognizes it. (A language is regular if and only if there exists an NFA that recognizes it).
Our M_A in a general case looks like :



Now, we need to prove reverse(A) is also regular. All we need to do is to find a NFA M that recognizes reverse(A). if we do have such NFA M, we get a successful proof.
(Since a language is regular if and only if there exists an NFA that recognizes it).

Here, we can get this NFA M by modifying our original NFA M_A in a few steps.
First, I would reverse all the transitions in M_A since the language of all strings in A is now written in reverse.
Then I would change the start state in M_A, q_{start}, into the accepting state in M.
However, we could not just simply change the accepting states in M_A into the start state in M. This is because there could be multiple accepting states in M_A, but we could only have one single start state in M. So I would create a new state in M as the start state. The start state could transit to all the accepting states in M_A when the string of symbols is empty string ε.

After such modification, our M looks like :



Compared to our original NFA M_A, we created a new state in M as the start state. The alphabet does not change.
All the transitions are now get reversed. The start state in M_A becomes the accepting state in M.
Now we have an NFA M that recognizes reverse (A), so that reverse (A) is proved also to be regular.
Thus, the set of languages which are regular is closed under the reverse operation.

3(b). ~~$B = \{ w$~~
$B = \{ w \in \Sigma_3 \mid$ the bottom row of w is the sum of the top two rows $\}$

It is difficult for us to straightforward estimate whether B is regular, I mean it is hard to describe an NFA M that recognizes B. This is because when we justify whether $w \in B$, we need to keep adding the numbers in the top two rows and see whether the sum is equal to the number in the bottom row, and we would continue this process from left (the first string of symbols) to right (the last string of symbols).
However, when we do binary operation, we would calculate from right (the least significant number) to left (the most significant number), and it is also the direction our carry number goes.

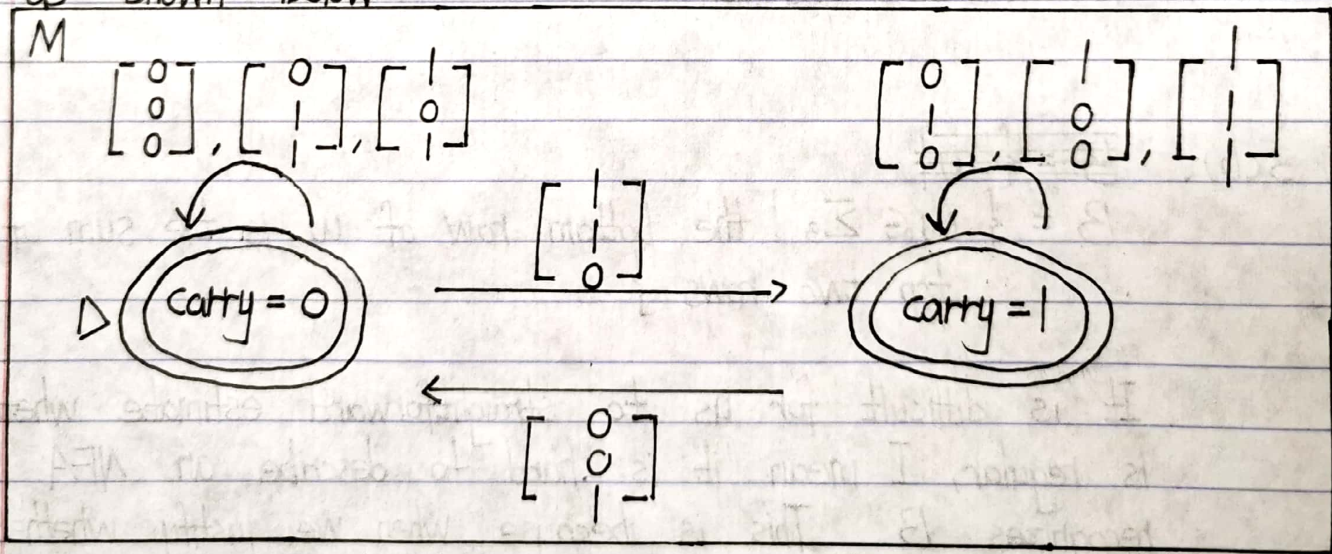
Obviously, these two directions of calculation are completely opposite. As a result, we can easily fall into chaos when construct an NFA to recognize B.

For example, $\begin{bmatrix}0\\0\\1\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix}\begin{bmatrix}1\\1\\0\end{bmatrix} \in B$

However, when we get the first string of symbols $\begin{bmatrix}0\\0\\1\end{bmatrix}$, $0+0$ is not equal to 1, the NFA might die immediately, which does not meet our expectation.

Since it is difficult to find an NFA to recognize B (though it does exists), we can try to find an NFA M that recognizes reverse(B), then reverse(B) would be regular.

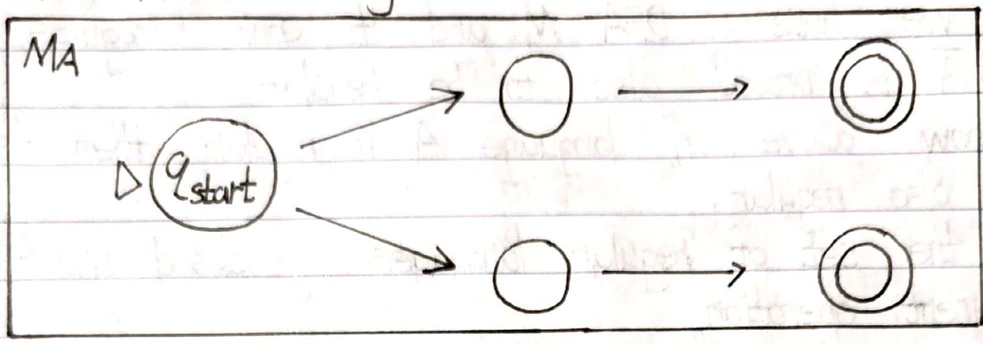
In reverse(B), we would add numbers from right to left, which matches the pattern of binary operation. So that we can easily get our NFA M that recognizes reverse(B), as shown below:



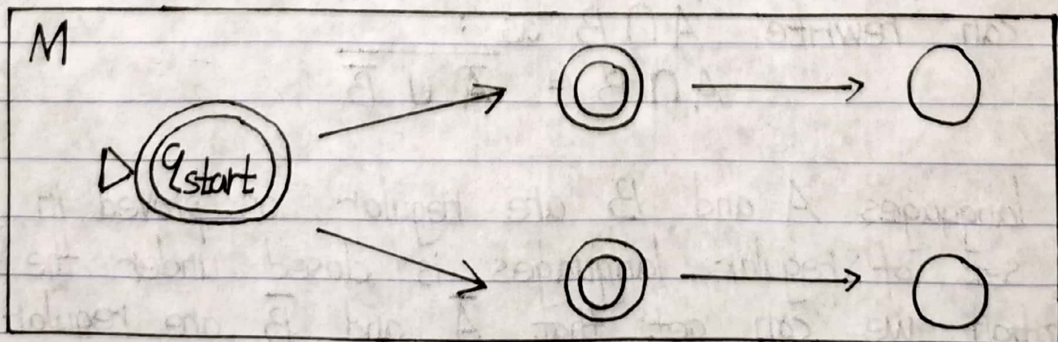
reverse(B) could be recognized by NFA M, so reverse(B) is regular.
As proved in 3(a), the set of regular languages is closed under the reverse operation. Therefore, reverse(reverse(B)) = B is also regular.

We successfully show that B is regular.

4(a). If language A is regular, then there is a DFA M_A that recognizes it. (A language is regular if and only if there exists a DFA that recognizes it).
Our M_A in a general case looks like :



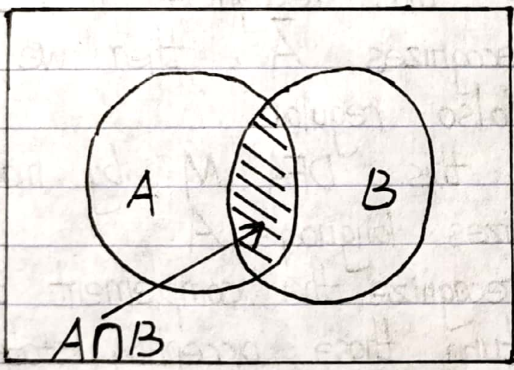
Now we need to prove language A's complement $\bar{A}$ is also regular. Since a language is regular if and only if there exists a DFA that recognizes it, If we can find a DFA M that recognizes $\bar{A}$, then we could successfully prove that $\bar{A}$ is also regular.
Here we can get this DFA M by modifying our original DFA M_A that recognizes language A.
Since we need to recognize the complement of language A, we would naturally turn those accepting states in M_A into rejecting states, and those ~~[illegible]~~ original rejecting states would become accepting states in our DFA M.
After such modification, our M looks like :



Compared to the original DFA M_A, all the states Q, the alphabet Σ, transition function δ and the start state q_0 do not change in DFA M. The only difference between these

two DFAs is that all the accepting states in M are rejecting states in M_A, and all the rejecting states in M are accepting states in M_A, and vice versa. They are complement to each other.
Now we have a DFA M and it could recognize $\bar{A}$, so that $\bar{A}$ is proved also to be regular.
As shown above, if language A is regular, then its complement $\bar{A}$ is also regular.
Thus, the set of regular languages is closed under the complement operation.

4(b). we first draw a Venn diagram to show $A \cap B$:



According to DeMorgan's Laws:

$$(!A) \& (!B) = !(A | B)$$

we can rewrite $A \cap B$ as:

$$A \cap B = \overline{\bar{A} \cup \bar{B}}$$

If languages A and B are regular, as proved in 4(a), the set of regular languages is closed under the complement operation, we can get that $\bar{A}$ and $\bar{B}$ are regular.
Since regular languages are closed under the union operation, now we have both $\bar{A}$ and $\bar{B}$ regular, so that $\bar{A} \cup \bar{B}$ is also regular.

Again, as proved in 4(a), the set of regular languages is closed under the complement operation, now we have $\bar{A} \cup \bar{B}$ regular, so that its complement $\overline{\bar{A} \cup \bar{B}}$ is also regular.

As mentioned before, $A \cap B = \overline{\bar{A} \cup \bar{B}}$, finally we can get that $A \cap B$ is regular.

Till here, we have successfully proved that if languages A and B are regular, then $A \cap B$ is also regular.

Thus, the set of regular languages is closed under the intersection operation.