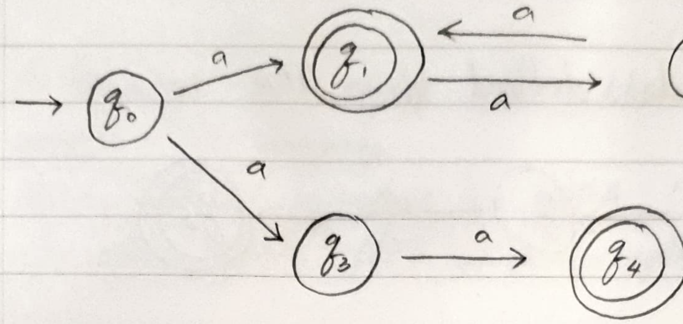
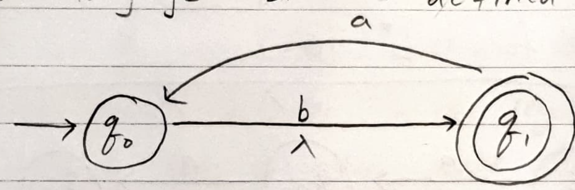
1.




2.

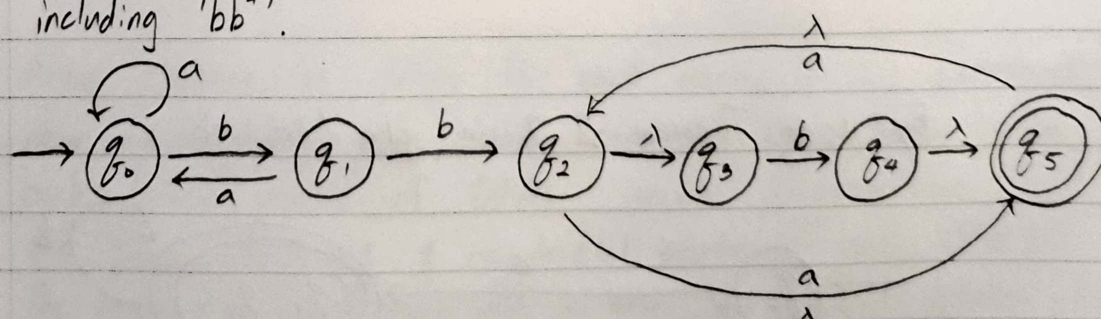
The language $L$ defined by NFA is...




The strings in $\bar{L}$ will be rejected by the NFA above.
To find $\bar{L}$, we have to find the cases when the strings are rejected.

The NFA rejects strings which include 'bb+'.
Thus, the set of all strings which include 'bb+' is $\bar{L}$.

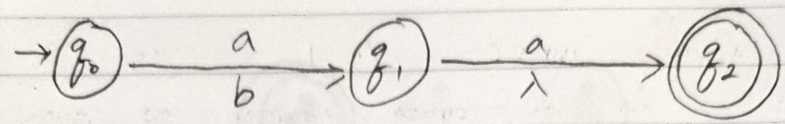
Now, we have to construct an NFA which accepts strings including 'bb+'.




The NFA accepts strings including 'bb+'.
Thus, the NFA accepts $\bar{L}$.

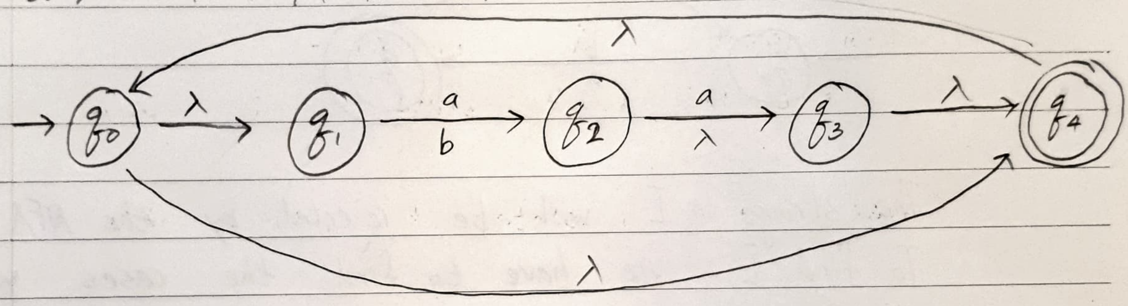
3.

The language L defined by NFA is...



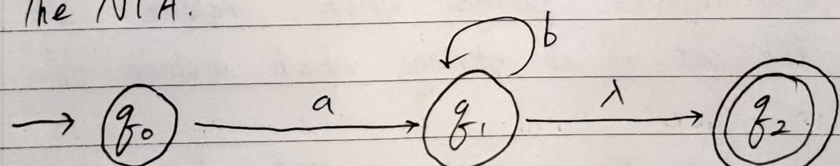

Let $L^* = L(\hat{M})$.

So, the NFA $\hat{M}$ for $L^*$ is...

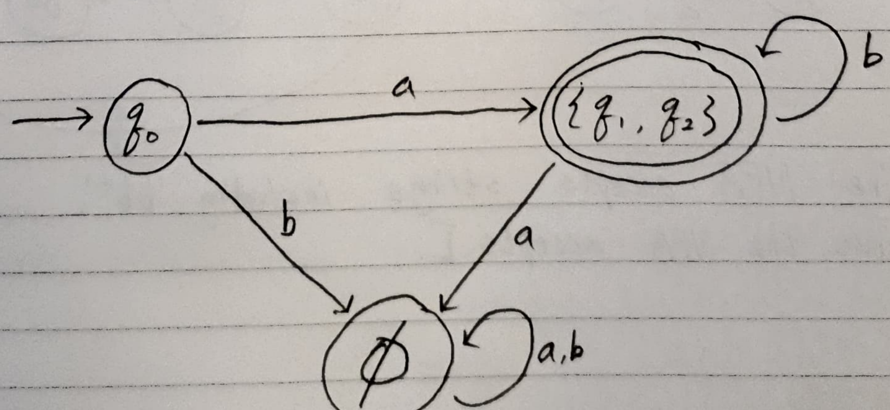



4.

The NFA:




By using Theorem 2.2 in the textbook...



Date . . No.

5.

A language is regular if there exists DFA which accepts the language.
Since DFA is equivalent to NFA, a language is regular if there exists NFA which accepts the language.

Let language L is finite language.

If $L = \phi$ or $L = \{\lambda\}$, then there exist NFA or DFA which accept L. So, L is regular.
$\because$ For $L = \{\lambda\}$, the initial state of NFA or DFA is finalstate, and for $L = \phi$, the state $\phi$ is both initial state and final state of NFA or DFA.

Assume $L = \{w_1, w_2, \cdots, w_i, \cdots, w_n\}$.
($\forall n \in N$, and n is a finite number)

We can construct DFAs for each $w_i$ in L. $(i = 1, 2, \cdots, n)$.
Then there are 'n' numbers of DFAs, and each DFA accepts each $w_i$ in L. $(i = 1, 2, 3, \cdots, n)$.

After constructing DFAs for each string in L, construct new initial state $q_{NFAi}$, and connect the $q_{NFAi}$ to each initial state of DFAs with $\lambda$-transition.
$(\delta(q_{NFAi}, \lambda) = \{q : q$ is initial state of each DFA$\})$.
The initial state of each DFA is now non-initial state of NFA.
Then construct new final state $q_{NFAf}$, and connect each final state of DFAs with $\lambda$-transition.
$(\delta(q_f, \lambda) = \{q_{NFAf}\}$, $q_f$ is final state of each DFA$)$
The final state of each DFA is now non-final state of NFA.

MOOKEUK

Now there exists an NFA which accepts $L$.
Thus, $L$ is regular.

Then let's add new string $w_{n+1}$ ($w_{n+1} \notin L$) to $L$.
Similar to previous steps, construct a DFA which accepts $w_{n+1}$. Then connect initial state of NFA to the initial state of DFA for $w_{n+1}$ with $\lambda$-transition, and the final state of DFA for $w_{n+1}$ to the final state of NFA with $\lambda$-transition.

$f(q_{NFAi}, \lambda) = \{q_{w_{n+1}i}\}$, $q_{w_{n+1}i}$ : initial state of DFA for $w_{n+1}$

$f(q_{w_{n+1}f}, \lambda) = \{q_{NFAf}\}$, $q_{w_{n+1}f}$ : final state of DFA for $w_{n+1}$.

Now, initial state and final state of DFA for $w_{n+1}$ are non-initial state and non-final state of NFA.

There exists an NFA which accepts $L \cup \{w_{n+1}\}$.
Thus, $L \cup \{w_{n+1}\}$ is regular.

Since $n$ can be any finite number, and there exists NFA for $L$, we proved that finite languages are regular by induction.

Therefore, all finite languages are regular. ■

Date . . No.

6. If the language $L$ is regular, there exist NFAs or DFAs which accept $L$.

If DFA for $L$ has only one final state, we set the initial state of DFA for $L$ to final state, and the final state of DFA for $L$ to initial state. Then we reverse the direction of each transition in the DFA.
The renewed DFA accepts $L^R$.

If DFA for $L$ has multiple final states, we connect multiple final states to a new final state with $\lambda$-transition.
Then we set the initial state of NFA to final state, and the final state of NFA to initial state. We reverse the direction of each transition in NFA.
The renewed NFA accepts $L^R$.

Since we constructed NFA or DFA which accept $L^R$, $L^R$ is regular.

Therefore, if $L$ is regular, then $L^R$ is regular. ■