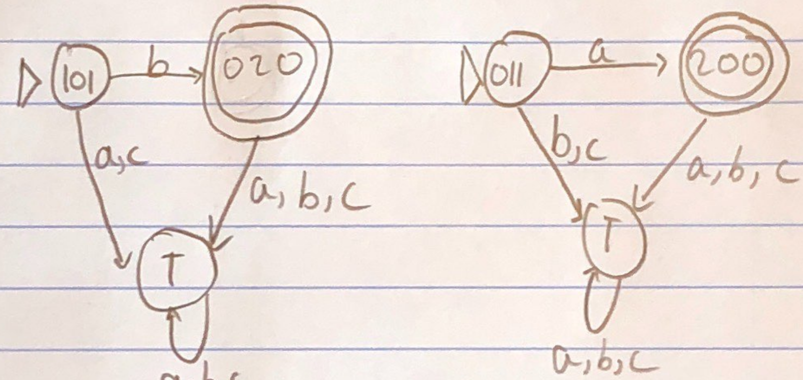
Assignment 2 Gabriele Nicula

1)a) The alphabet Σ is $\{a, b, c\}$

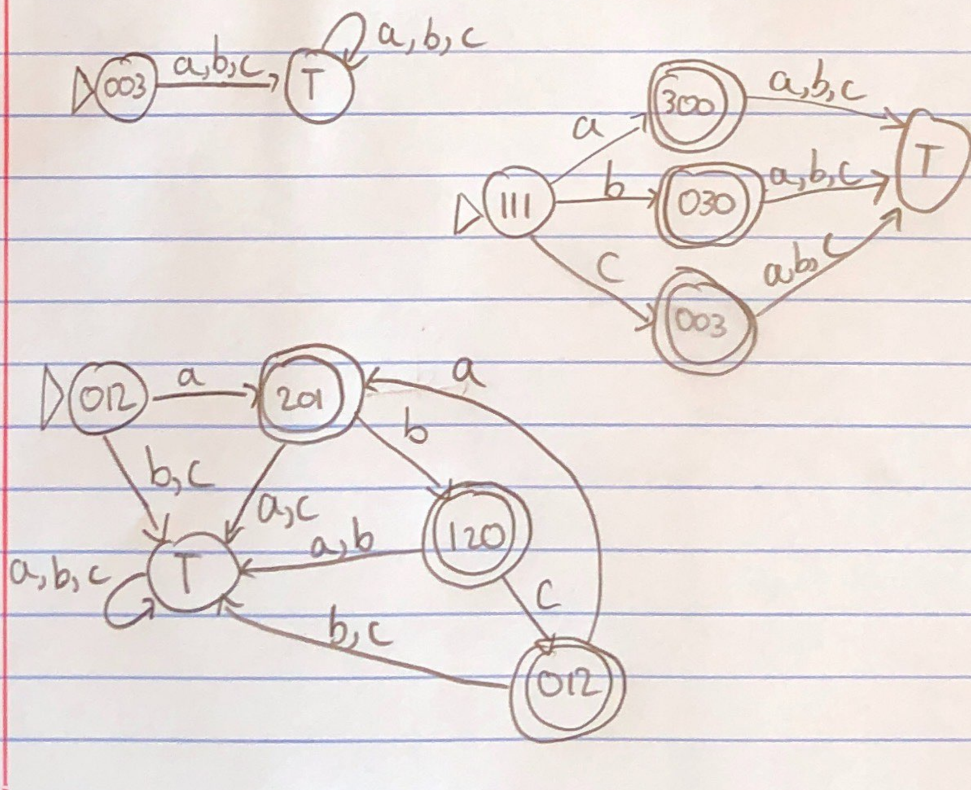
b)

	a	b	c
110	T	T	002
002	T	T	T
T	T	T	T

c)



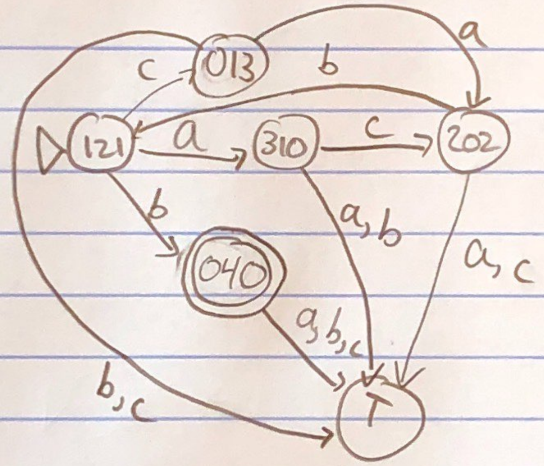
d)



e) List all must-fail, might-fail, and cannot-fail states from previous DFA's

must fail: 111, 003, 030, 300
might fail: none
cannot fail: 120, 210, 012, 021, 201, 102

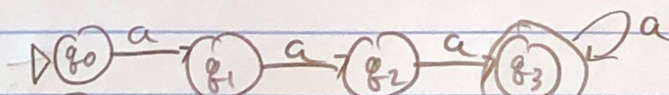
f) Draw the automaton with initial state 121



must fail: 040
cannot fail: 013, 310, 202
might fail: 121

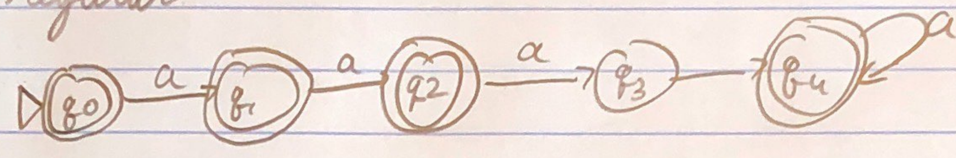
2) Show that $L = \{a^n : n \geq 3\}$ is regular

A language is regular if there exists a valid DFA for it.



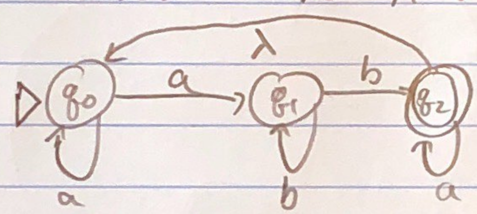
DFA that accepts the language exists so L is regular.

3) Show that $L = \{a^n : n \geq 0, n \neq 3\}$ is regular

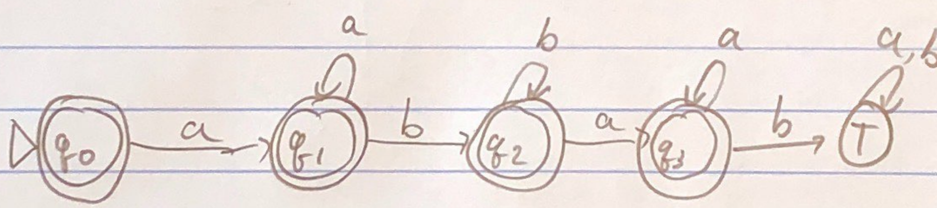


Since there exists a valid DFA that accepts the language L is regular

4) Convert the NFA to a DFA



NFA accepts $\{\lambda, a, aab, aaba, abbaa, \ldots\}$



DFA accepts: $\{\lambda, a, aab, aaba, abbaa, \ldots\}$

5) Is it true that for every NFA $M = (Q, \Sigma, \delta, q_0, F)$ the complement of $L(M)$ is equal to

$$\{w \in \Sigma^* : \delta^*(q_0, w) \cap (Q - F) \neq \emptyset\}$$

Yes it is true.

Let $L' = U - L(M)$ be the complement of $L(M)$

$L_2 = \{w \in \Sigma^* : \delta^*(q_0, w) \cap (Q - F) \neq \emptyset\}$

$L_2 \subseteq L'$

Consider a word $w \in L_2$, then
$\delta^*(q_0, w) \cap (Q - F) = Q_2 \neq \emptyset$

For each state $q \in Q_2 \Rightarrow q \in (Q - F)$
that means that q is not a final state of M, which means $w \in L'$ because $w \notin L(M)$

$L' \subseteq L_2$
Consider $w' \in L'$, that means w' is not accepted by $L(M)$ so $\delta^*(q_0, w')$ contains only non-final states of M.

There exists a non-final state q' such that

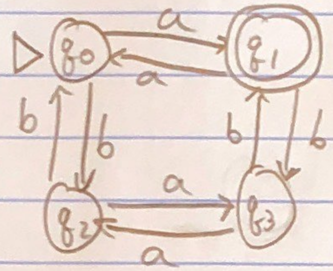
$$q' \in \delta(q_0, w') \cap (Q-F)$$

which proves that:

$$\delta^*(q_0, w') \cap (Q-F) \neq \emptyset \Rightarrow w' \in L_2$$

$$\left.\begin{aligned} &\text{Therefore } L' \subseteq L^2 \\ &L^2 \subseteq L' \end{aligned}\right\} \Rightarrow L' = L_2$$

6) DFA for $L = \{ w \in \{a, b\} \mid a \% 2 = 1, b \% 2 = 0 \}$



7) Let "truncate" remove the rightmost symbol from any string.

$$\text{truncate}(L) = \{\text{truncate}(w) : w \in L\}$$

Prove that if L is a regular language then truncate (L) is also regular.

Given L is a regular language there exists a DFA_L that accepts it.

We can modify this DFA_L in such a way that the new DFA accepts truncate (L).

Let Q_F be the set of final states in DFA_L. For each state $q_f \in Q_F$ we do the following:

a) If q_f has a transition to self (is self-looping), then we keep q_f as final, and make all the states q_i that transition to q_f as final.

b) If q_f does not transition to itself (is not self-looping), we make q_f non-final and make

every state q_i that transitions to q_f
a final state.

After applying this procedure to all the states in Q_F, we obtain a DFA that accepts any truncated word from L because they end up in a final state in this new DFA.

Therefore, if L is regular truncate(L) is also regular.