Homework Problems

H2.1 Design finite automata that recognise the following languages:

(i) $\{w \in \{a,b\}^* \mid w$ begins with the substring $aba\}$;

(ii) $\{w \in \{a,b\}^* \mid w$ contains aba as a substring$\}$;

(iii) $\{w \in \{a,b\}^* \mid w$ has substring aba at its end$\}$;

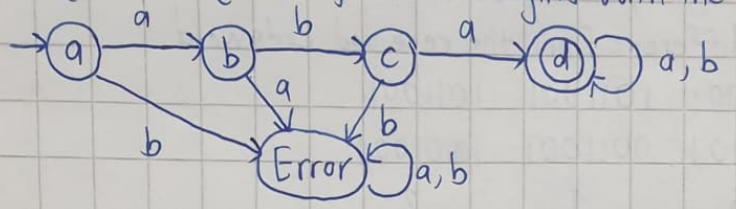
(iv) $\{w \in \{a,b\}^* \mid w$ contains both ab and ba as (possibly overlapping) substrings$\}$.

State (say) whether your automata are complete or not.

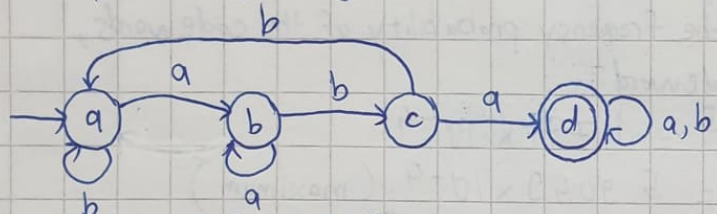
Your automata **don't have to be** complete (being complete is of course allowed, but not required).

H2.1 :

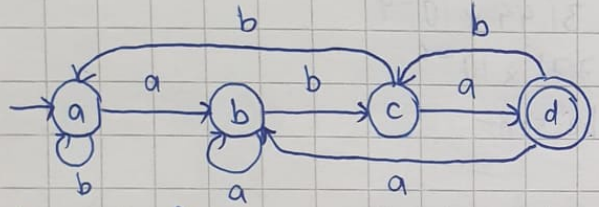
(i) { w ∈ {a,b}* | w begins with the substring aba };



(ii) { w ∈ {a,b}* | w contains aba as a substring };



(iii) { w ∈ {a,b}* | w has substring aba at its end }



All of the finite automata designed above are complete with final state "d" and all the entries in the transition diagram has been filled in

H2.2 Design finite automata that recognise the following languages:

(i) $\{w \in \{0,1\}^* \mid$ the number of 1's in w is divisible by three (or possibly zero)$\}$;

(ii) $\{w \in \{0,1\}^* \mid$ the number of substrings 01 in w is odd$\}$;

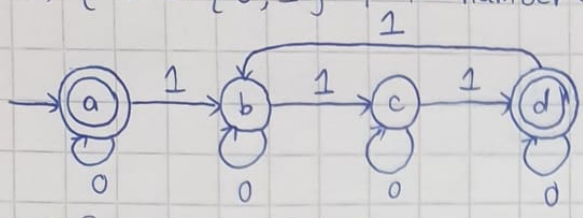
(iii) $\{w \in \{0,1\}^* \mid$ the number of 0's in w is odd and the number of 1's in w is divisible by three (or possibly zero)$\}$.

State (say) whether your automata are complete or not.

Your automata **don't have to be** complete (being complete is of course allowed, but not required).

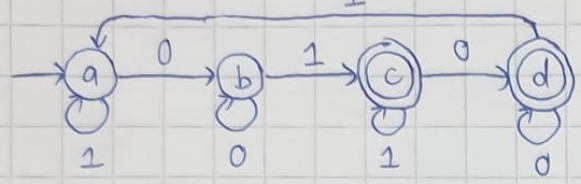
H2.2

(i) $\{w \in \{0,1\}^* \mid$ The number of 1's in w is divisible by 3 (0 included)$\}$;



(Complete automaton)

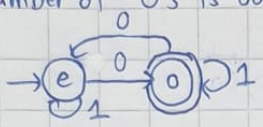
(ii) $\{w \in \{0,1\}^* \mid$ The number of substrings 01 in w is odd$\}$;



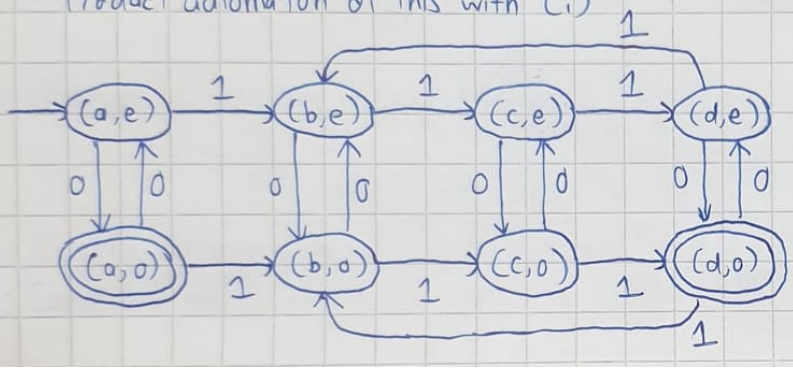
(Complete automaton)

(iii) $\{w \in \{0,1\}^* \mid$ Number of 0's is odd and number of 1's is divisible by 3 (0 included)$\}$;

Number of 0's is odd:



Product automaton of this with (i)



(Complete automaton)

H2.3 Design a finite automaton that models the behaviour of a lift moving between two floors. The lift can be either up or down. Both floors have a simple 'call here' button for the lift, and inside the lift there are buttons for going 'up' and 'down'. In addition, the lift has a door that can be opened and closed; the lift only moves when the door is closed. The time required for the lift to travel between the two floors does not need to be taken into account, and any possible service requests occurring during this interval can be ignored. The automaton does not need to have any distinct "final states".

H2.3 : A deterministic finite automaton is a 5 tuple $M = (K, \Sigma, \delta, s, F)$

Since the automaton for the lift does not have final state $\Rightarrow M_{lift} = (k, \Sigma, \delta, s)$

- Set of states: $K = \{bc, bo, ac, ao\}$
  where b: below floor, a: above floor, o: door opened, c: door closed
- Alphabet: $\Sigma = \{u, d, csf, cdf, e\}$
  where: u: go up, d: go down, csf: call elevator at same floor
  e: enter the lift cdf: ~ different floor
- Initial state: $s = bc$
- Diagram of transition function of the lift δ could be as below