Names: Omar Abdelmotaleb and Benjamin Singleton
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Omar Abdelmotaleb
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Benjamin Singleton

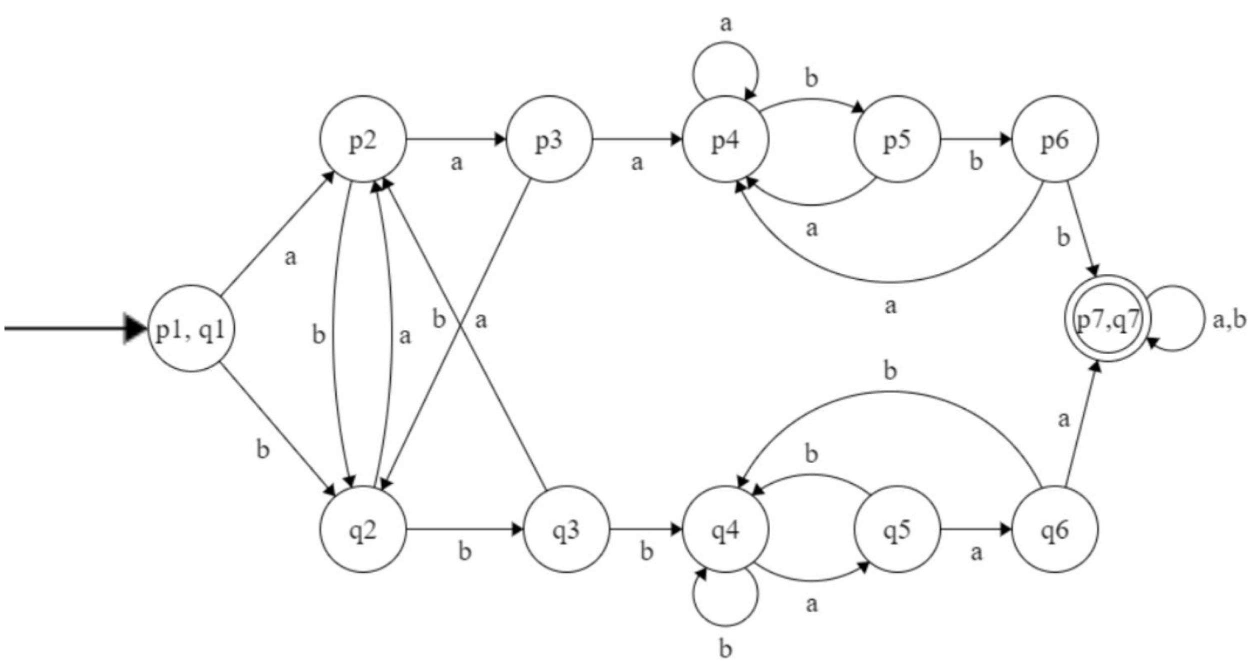
Problem 1. (30 points) Construct deterministic FSAs for each of the following languages over the alphabet $\{a, b\}$:

1. $L_1 = \{w: w\ contains\ the\ string\ aaa\ and\ the\ string\ bbb\}$.

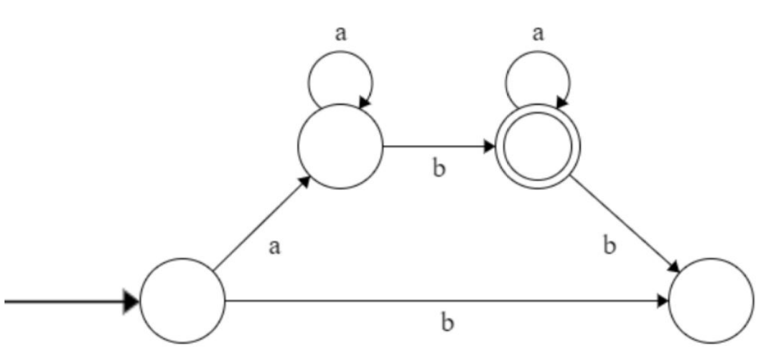
2. $L_2 = \{w: w\ starts\ with\ an\ a\ and\ has\ at\ most\ one\ b.\}$
   Hint: Express L_2 as the intersection of two languages.

3. $L_3 = \{w: the\ 4^{th}\ symbol\ from\ the\ end\ of\ w\ is\ a\}$.
   So, for example, $abbbaaba \in L_3$ but $abaababa \notin L_3$. To get started see how trivial it would be if the FSA had to check the last symbol of the input. Then try to design an FSA that must check only the 2nd last symbol. What does each state in this FSA represent? Now extend this idea to design an FSA that accepts L_3.

1.



2.



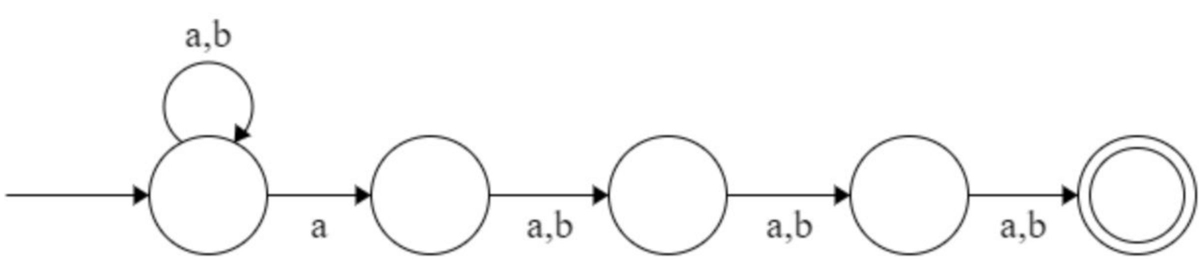
Names: Omar Abdelmotaleb and Benjamin Singleton
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Omar Abdelmotaleb
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Benjamin Singleton

3. We converted the NFA to a DFA using transition tables.

state	a	b
->A	A,B	A
B	C	C
C	D	D
D	E	E
E	-	-

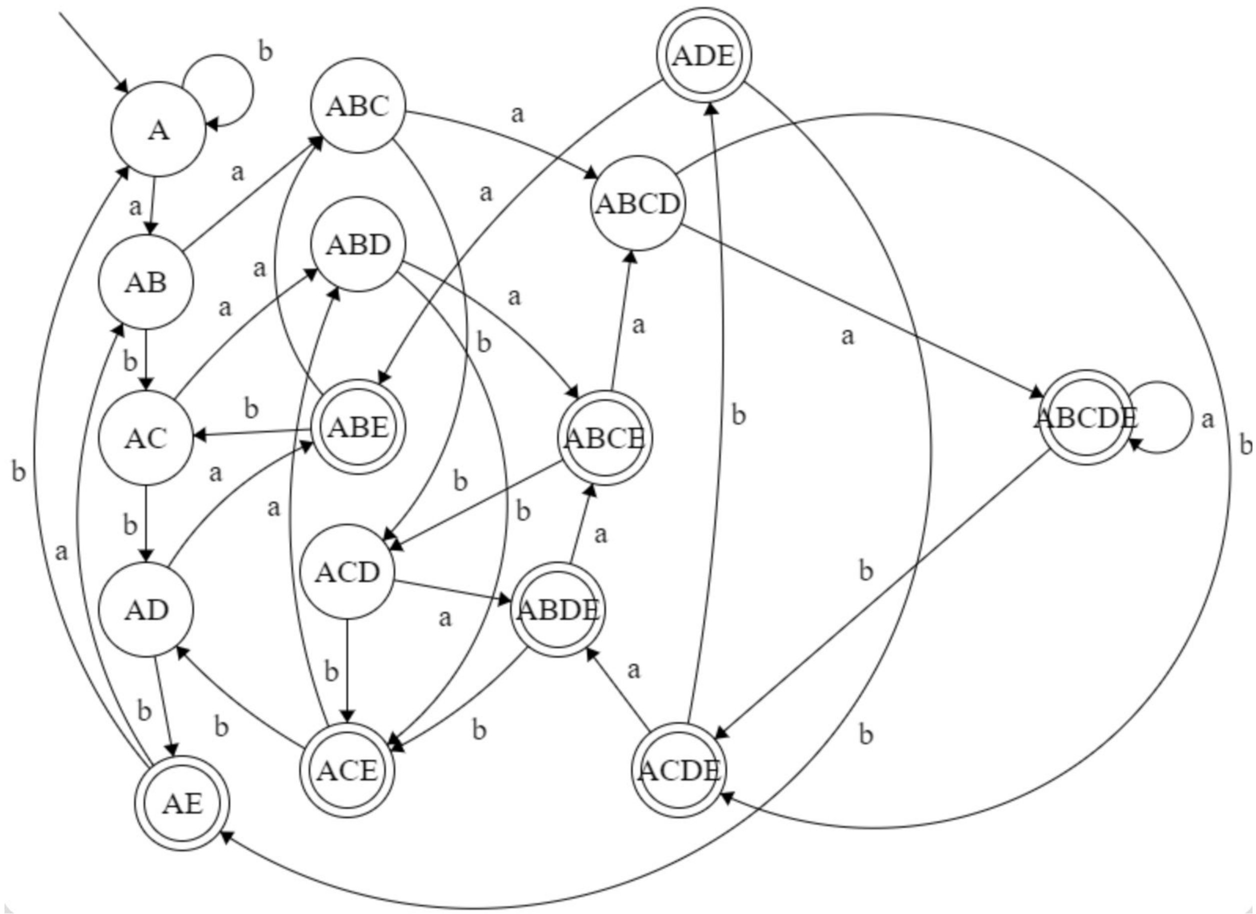
state	a	b
->A	AB	A
AB	ABC	AC
AC	ABD	AD
AD	ABE	AE
AE	AB	A
ABC	ABCD	ACD
ABD	ABCE	ACE
ABE	ABC	AC
ACD	ABDE	ADE
ACE	ABD	AD
ADE	ABE	AE
ABCD	ABCDE	ACDE
ABCE	ABCD	ACD
ABDE	ABCE	ACE
ACDE	ABDE	ADE
ABCDE	ABCDE	ACDE

NFA:



Names: Omar Abdelmotaleb and Benjamin Singleton
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Omar Abdelmotaleb
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Benjamin Singleton

DFA:



All FSAs shown were designed using Finite State Machine Designer:
http://madebyevan.com/fsm/

Names: Omar Abdelmotaleb and Benjamin Singleton
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Omar Abdelmotaleb
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Benjamin Singleton

Problem 2. (10 points) Modify the proof of Theorem 1.25 in the textbook to cover the case when the machines M_1, M_2 have *different* input alphabets Σ_1, Σ_2. Hint: the machine M that recognizes the union of the languages of M_1, M_2 will have input alphabet $\Sigma = \Sigma_1 \cup \Sigma_2$. Take care when defining $\delta((r_1, r_2), a)$ as the symbol a could belong to one alphabet but not the other!

Proof by Construction

Let M_1 recognize A_1, where $M_1 = (Q, \Sigma_1, \delta_1, q_2, F_1)$, and
M_2 recognize A_2, where $M_2 = (Q, \Sigma_2, \delta_2, q_2, F_2)$, and

Construct M to recognize $A_1 \cup A_2$, where $M = (Q, \sum, \delta, q, F)$.

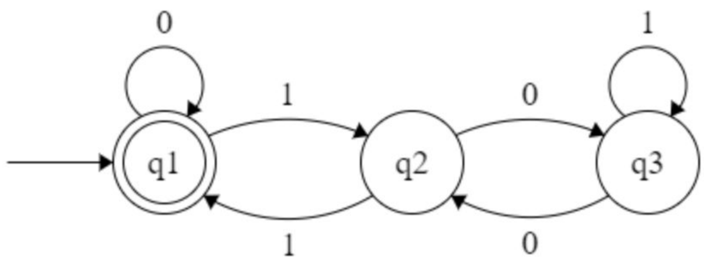
1. $Q = \{(r_1, r_2) \mid r_1 \in Q_1 \ and\ r_2 \in Q_2\}$,
   This set is the cartesian product of sets Q_1 and Q_2 and is written $Q_1 \times Q_2$. It is the set of all pairs of states, the first from Q_1 and the second from Q_2.
2. Σ, the alphabet and is the union of Σ_1 and Σ_2.
3. δ, the transition function is defined as follows. For each $(r_1, r_2) \in Q$ and for each $a \in \Sigma_1$ and $b \in \Sigma_2$, let
   $$\delta((r_1, r_2),\ a) \ = \ (\delta_1(r_1, a), \delta_2(r_1, b), \delta_3(r_2, a), \delta_4(r_2, b))$$
   Hence δ gets a state of M (which actually is a pair of states from M_1 and M_2), together with an input symbol, and returns M's next state.
4. q_0 is the pair (q_1, q_2).
5. F is the set of pairs in which either member is an accept state of M_1 or M_2. We can write it as
   $$F = \{(r_1, r_2) \mid r_1 \in F_1 \ or\ r_2 \in F_2\}.$$
   This expression is the same as $F = (F_1\ x\ Q_1) \cup (F_2\ x\ Q_2)$.

Names: Omar Abdelmotaleb and Benjamin Singleton
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Omar Abdelmotaleb
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Benjamin Singleton

Problem 3. (10 points) Let D_k denote the set of binary strings that represent numbers divisible by k. For example, input strings $0, 00, 000, 10, 010, 0010, 0100010$ are all divisible by 2 (the least significant bit is the rightmost, or last symbol in the sequence), and therefore are all in the language D_2.

1. Construct a deterministic FSA to recognize the language D_3. Hint: Use states to represent remainders of the input seen thus far when divided by 3.
2. Prove that D_k is regular, for every $k \geq 1$.

1.



2. Construct M_k such that $M = (Q, \Sigma, \delta, q, F)$.

Q, being the number of states, is equal to k, where k ≥ 1. So, $Q = \{q_0, q_1, \ldots, q_k\}$

$\Sigma = \{0,1\}$.

δ, the transition function where $\delta: (q_{k-1}, (0,1)) \to q_k$

q_0 , the start and accept state.

F , the set of accept states $\{q_0\}$.

Let ω be the input string over Σ. Let r be the remainder such that ω % k = r. The accept state is reached if r = 0. Each state will represent a remainder r for all of k. All states except q_0 are end states for all of r from 1 up to k-1. This constructs the FSA for M_k.

Since D_k denotes a set of binary strings that follow the alphabet $\Sigma = \{0,1\}$, its alphabet for the input is accepted over M_k. Since the alphabet is accepted over M_k, an FSA, D_k is regular.

Names: Omar Abdelmotaleb and Benjamin Singleton
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Omar Abdelmotaleb
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Benjamin Singleton

Optional Exercise. (Present your ideas anytime this term during Sandeep's office hours.) You have been asked to design an FSA to operate a building elevator. How would you go about this design process? What would a state of the FSA correspond to? What should be the alphabet? What constraints would be reasonable to impose on the movement of the elevator? This is an open-ended problem – we don't expect a full design, but rather things you considered and what constraints you found difficult to design for. To get started, imagine a 2-storey building, then a 3-storey building, etc. How do your ideas scale for a very tall building?