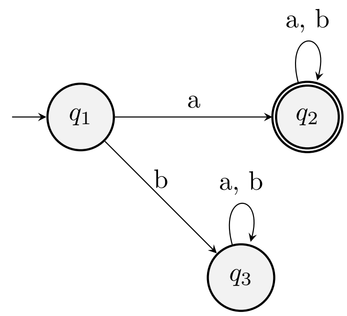
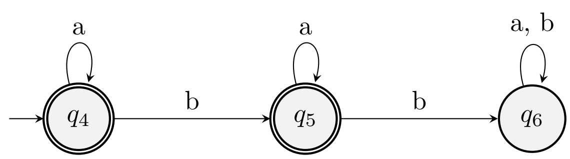
Theory of Computation: Assignment 2 Solutions

Arjun Chandrasekhar

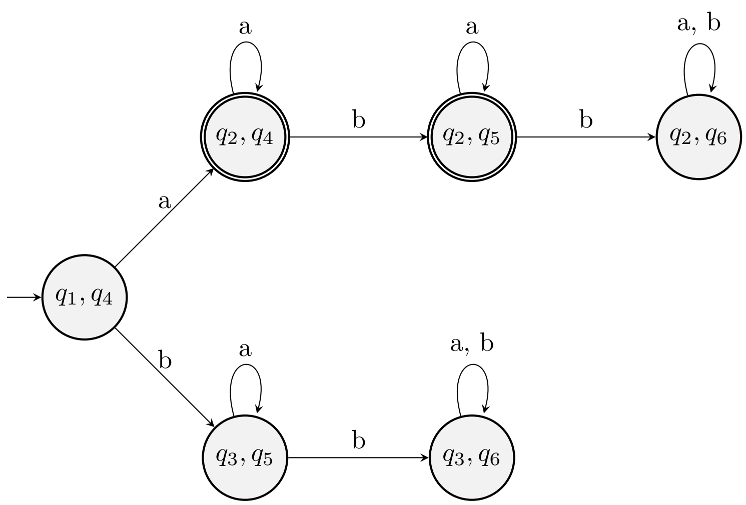
1. First. we'll make a DFA for $L = \{w | w \text{ starts with an a}\}$



Next we'll make a DFA for $L = \{w | w \text{ has at most one b}\}$



Finally, we'll combine them into a DFA for L using the intersection construction technique from class.

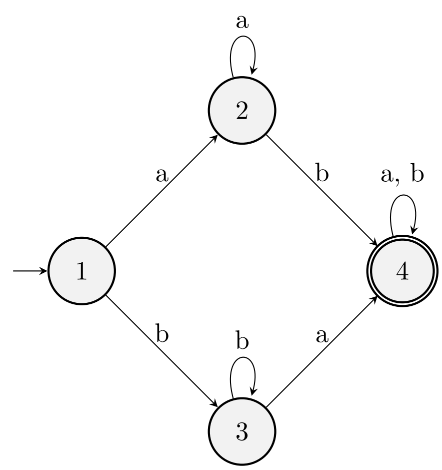


Some notes:

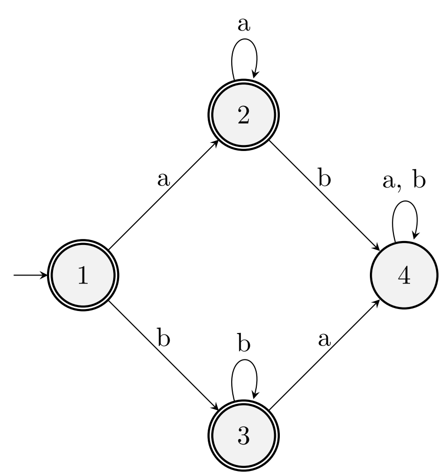
- Not all states are shown - just the ones that are reachable from the start state. If we were to show the full DFA, there would be $3 \times 3 = 9$ states.

- The start state is (q_1, q_4) because the fist machine starts in q_1 and the second machine starts in q_4
- The accept states are (q_2, q_4) and (q_2, q_5), because the first machine accepts in q_2, and the second machine accepts in either q_4 or q_5

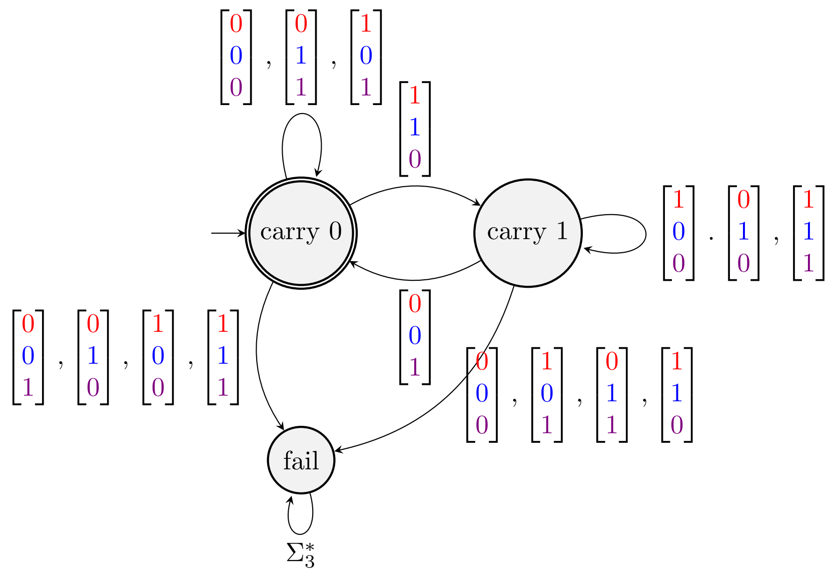
2. First we'll make a DFA for the (simpler) complement language $L = \{w|$ contains either ab or ba as a substring$\}$



Then, we use the complement construction from class (i.e. flipping the accept/reject states) to construct the DFA for the original language:



3. We'll use the technique from class, in which we use the state to keep track of the current carry value. Transitions will represent adding the digits of a column. If we ever find an inconsistency, we go to a reject state. At the end, we accept if and only if we are in the carry 0 state; this indicates that all of the columns added up properly and we are not missing any leading digits.



Note that it is very important to be able to read the characters in reverse, because this lets us read the digits from least significant to most significant digit.

4. For this problem, we need to show that if A and B are regular, then $A \oplus B$ is also regular.

   Approach 1: We'll construct a DFA to recognize $A \oplus B$. We know that A and B are regular. Let $M_A = (Q_A, \Sigma, \delta_A, S_A, F_A)$ be a DFA that recognizes A, and let $M_B = (Q_B, \Sigma, \delta_B, S_B, F_B)$ be a DFA that recognizes B. The following DFA $M = (Q, \Sigma, \delta, S, F)$ will recognize $A \oplus B$:

   - $Q = Q_A \times Q_B$ - each state is a combination of a state from M_A and a state from M_B
   - $\Sigma = \Sigma$ - the alphabet is the same
   - $\delta((q_A, q_B), \sigma) = (\delta_A(q_A, \sigma), \delta_B(q_B, \sigma))$. Some notes on this:
     - The input to the transition function is always a state, and a symbol. The state is (q_A, q_B) and the symbol is σ. In this case, each state is a combination of a state from M_A and a state from M_B
     - The output of the transition function is always a state; in this case, each state is a state from M_A and a state from M_B.
     - $\delta(q_A, \sigma)$ applies M_A's transition function to M_A's current state q_A and the current symbol σ. Similarly, $\delta(q_B, \sigma)$ applies M_B's transition function to M_B's current state q_B and the current symbol σ
   - $S = (S_A, S_B)$ - the starting state is a combination of M_A's start state and M_B's start state.
   - $F = \{(q_A, q_B) | q_A \in F_A \text{ or } q_B \in F_B \ \underline{\text{but not both}}\}$ - we accept any combination of states for which exactly one of the two machines is accepting.

   Approach 2: We note that

   $$A \oplus B = (A \cap B^c) \cup (A^c \cap B)$$

   Regular languages are closed under complement, union, and intersection; therefore, regular languages are closed under XOR.

5. We need to show that if A is regular, then EVERY-OTHER(A) is also regular.

   Approach 1: We'll construct a DFA to recognize EVERY-OTHER(A). We know that A is regular. Let $M_A = (Q_A, \Sigma, \delta_A, S_A, F_A)$ be a DFA that recognizes A. The following DFA $M = (Q, \Sigma, \delta, S, F)$ will recognizes EVERY-OTHER(A):

- $Q = Q_A \times \{\text{ODD}, \text{EVEN}\}$. Every state is a combination of a state from M_A, and a "counter" that keeps track of whether we are reading an even or an odd character.
- $\Sigma = \Sigma$ - the alphabet is the same
- $\delta((q_A, \text{ODD}), \sigma) = (\delta_A(q_A, \sigma), \text{EVEN})$ - if it's an odd character, then we transition M_A's state according to the transition function, and move the counter to an even character.
  $\delta((q_A, \text{EVEN}), \sigma) = (q_A, \text{ODD})$. If it's an even character we ignore it; we don't transition M_A's state, and we move the counter back to an odd character.
- $S = (S_A, \text{ODD})$
- $F = F_A \times \{ODD\}$

Approach 2: We note that

$$\text{EVERY-OTHER}(A) = \text{PERFECT-SHUFFLE}(A, \Sigma^*)$$

Regular languages are closed under PERFECT-SHUFFLE, and Σ^* is a regular language. Therefore EVERY-OTHER(A) is regular.