CSC236 Worksheet 8 Solution

Hyungmo Gu

May 14, 2020

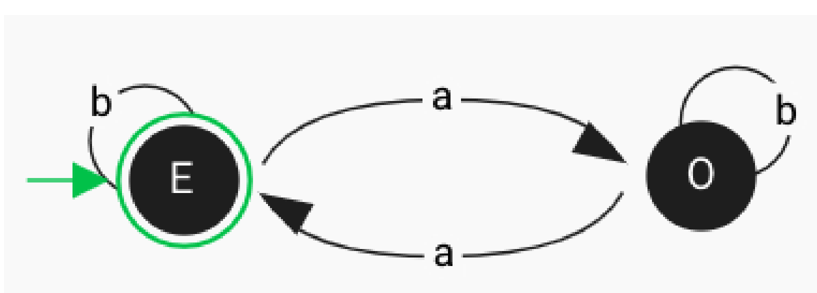
Question 1

- **Part 1 (Building L_1 and L_2):**

  L_1:

$$Q = \{E, O\}$$
$$\Sigma = \{a, b\}$$
$$\delta = \begin{array}{|c|c|c|} \hline & \text{a} & \text{b} \\ \hline {}^{*}\text{E} & \text{O} & \text{E} \\ \text{O} & \text{E} & \text{O} \\ \hline \end{array}$$
$$q_0 = E$$
$$F = \{E\}$$

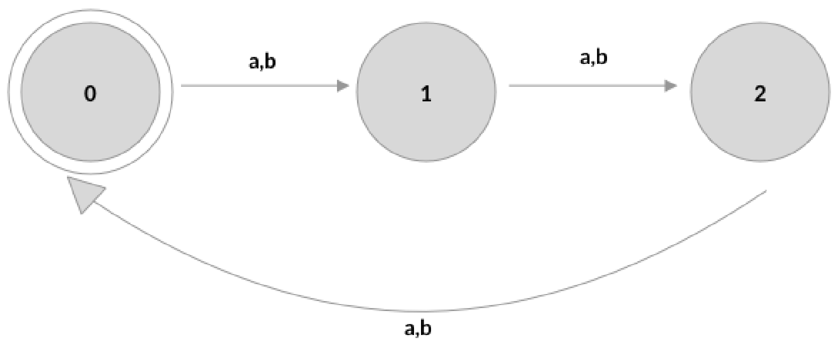
  Draw Diagram



  L_2:

$$Q = \{0, 1, 2\}$$
$$\Sigma = \{a, b\}$$
$$\delta = \begin{array}{|c|c|c|} \hline & \text{a} & \text{b} \\ \hline {}^{*}0 & 1 & 1 \\ 1 & 2 & 2 \\ 2 & 0 & 0 \\ \hline \end{array}$$
$$q_0 = 0$$
$$F = \{0\}$$

Draw Diagram



Part 1 (Building $L_1 \cap L_2$):

$$Q = \{(E, 0), (E, 1), (E, 2), (O, 0), (O, 1), (O, 2)\}$$
$$\Sigma = \{a, b\}$$
$$\delta = \begin{array}{|c|c|c|} \hline & \text{a} & \text{b} \\ \hline {}^{*}\text{(E,0)} & 1 & 1 \\ \text{(E,1)} & 2 & 2 \\ \text{(E,2)} & 0 & 0 \\ \text{(O,0)} & 1 & 1 \\ \text{(O,1)} & 2 & 2 \\ \text{(O,2)} & 0 & 0 \\ \hline \end{array}$$
$$q_0 = (E, 0)$$
$$F = \{(E, 0)\}$$

Correct Solution:

Part 1 (Building L_1 and L_2):

L_1:

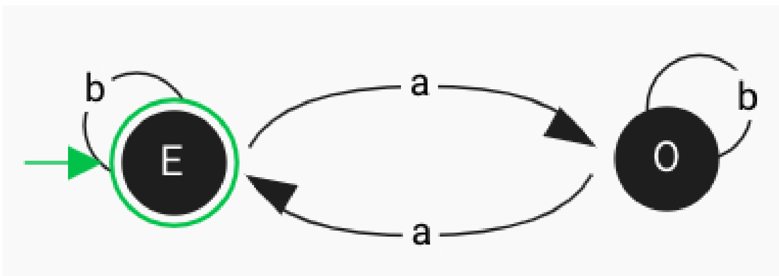
$$Q = \{E, O\}$$
$$\Sigma = \{a, b\}$$

$\delta =$

	a	b
*E	O	E
O	E	O

$$q_0 = E$$
$$F = \{E\}$$

Draw Diagram



L_2:

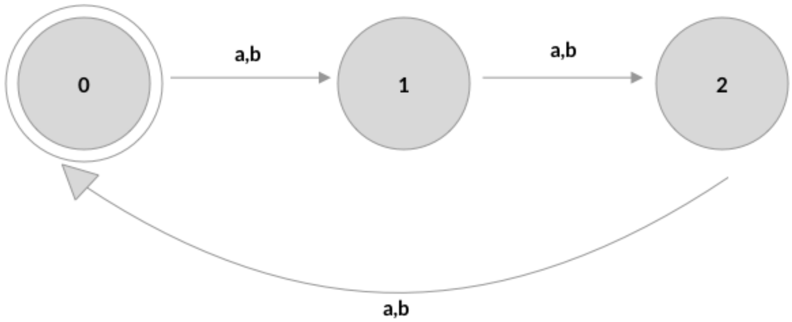
$$Q = \{0, 1, 2\}$$
$$\Sigma = \{a, b\}$$

$\delta =$

	a	b
*0	1	1
1	2	2
2	0	0

$$q_0 = 0$$
$$F = \{0\}$$

Draw Diagram



Part 1 (Building $L_1 \cap L_2$):

$$Q = \{(E,0),(E,1),(E,2),(O,0),(O,1),(O,2)\}$$
$$\Sigma = \{a,b\}$$

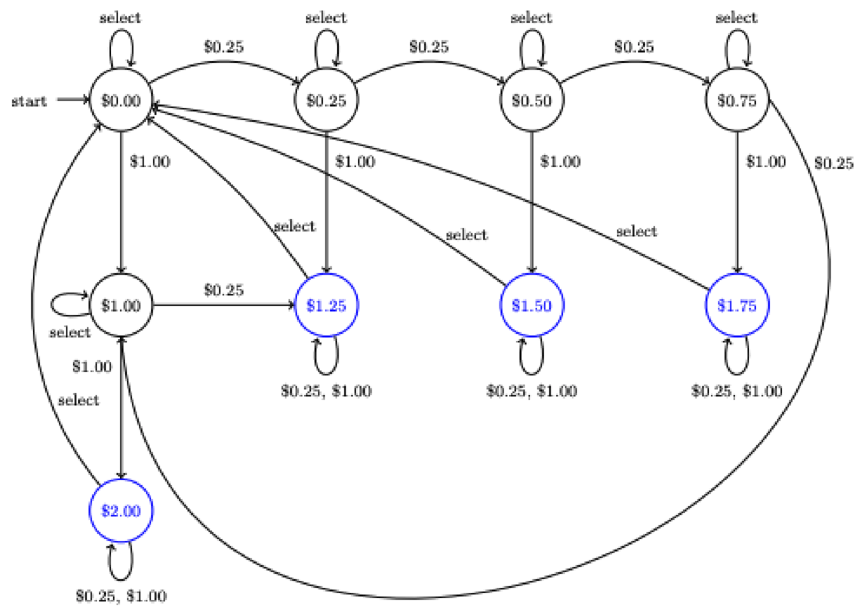
$\delta =$

	a	b
*(E,0)	(O,1)	(E,1)
(E,1)	(O,2)	(E,2)
(E,2)	(O,0)	(E,0)
(O,0)	(E,1)	(O,1)
(O,1)	(E,2)	(O,2)
(O,2)	(E,0)	(O,0)

$$q_0 = (E,0)$$
$$F = \{(E,0)\}$$

Notes:

- **Deterministic Finite State Automaton (DFSA)**: is a mathematical method of machine which, given any input string x, **accepts** or **rejects** x.
- Applications of DFSA
  1. Vending Machine



2. Protocol analysis
3. Text parsing
4. Video game character behavior

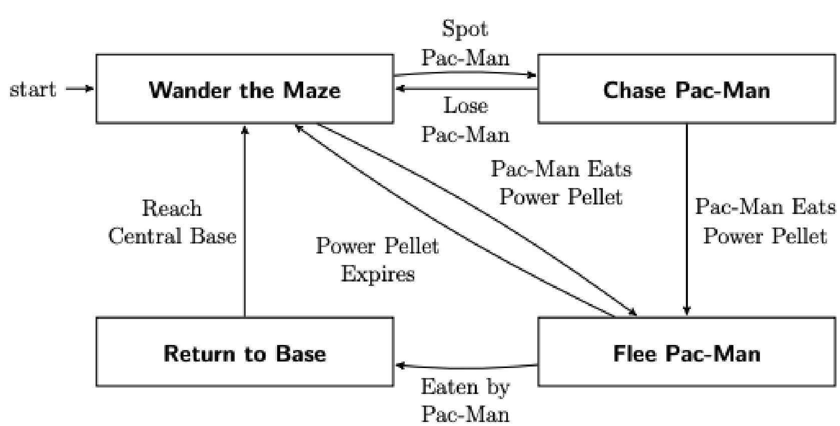
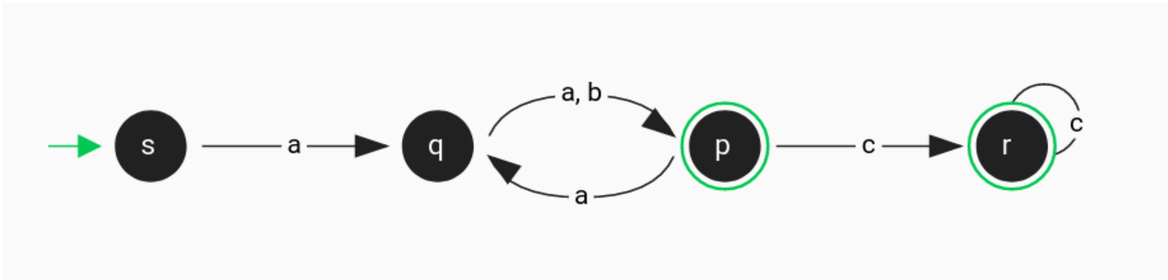


Figure 3: Behavior of a Pac-Man Ghost

5. Security Analysis
6. CPU control units (**)
7. Natural Language Processing (**)
8. Speech Recognition (**)

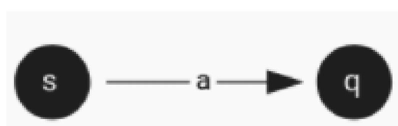
- Definitions and Syntax



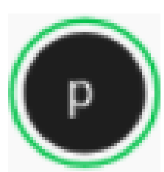
- *DFSA M* is a quintuple $M = (Q, \Sigma, q_0, F, \delta)$, where
  - Q : a finite set of **states**.
    - Represents status of system
    - Is represented by a black circle, i.e. s,q

    - i.e. automatic sliding door at walmart has two states: either close or open
    - i.e. traffic light has three states: red, yellow, green
  - Σ : a finite non-empty alphabet
    - is set of symbols in each transition, i.e. a, b, c

  - $q_0 \in Q$: the start or initial state
  - $\delta : Q \times \sigma \to Q$: a transition function
    - is a connection between two states.
    - is represented by an arrow



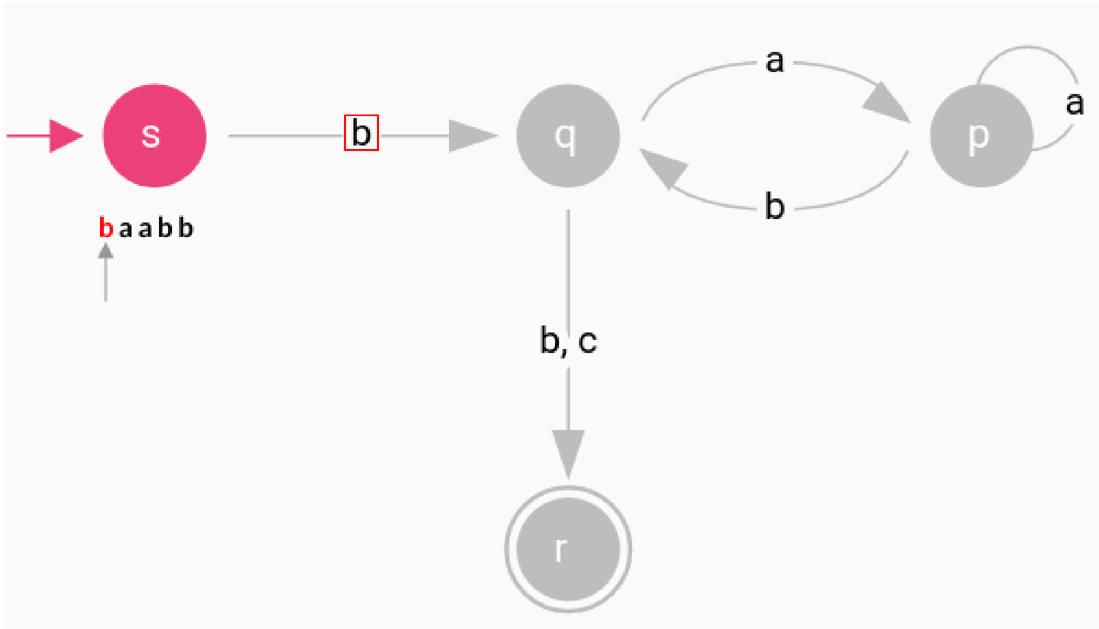
  - $F \subseteq Q$: the set of accepting or final states
    - Is represented by a double circle



    - Multiple accepting states may exists
    - Purpose: When processing ends, the output is either *accept* or *reject*

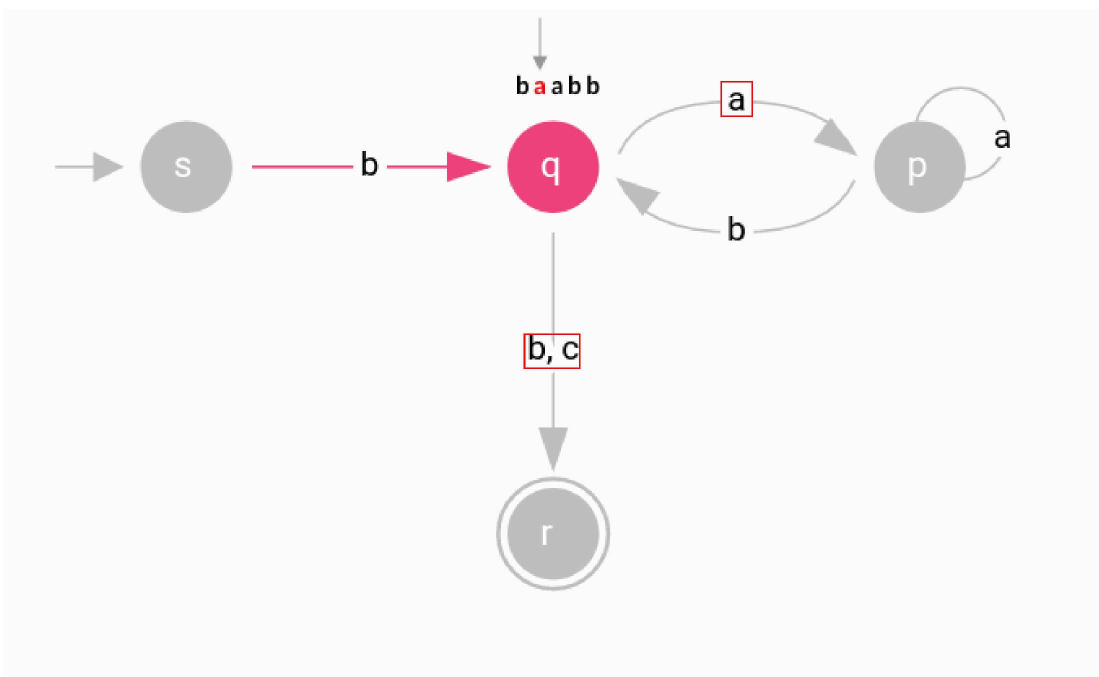
- Simple Example

– Step 1



1. First symbol of the input **baabb** is **b** and the current state is s.
2. Ask, is there any exiting transition from s that contains the symbol **b**?
3. The answer is yes, so move to q

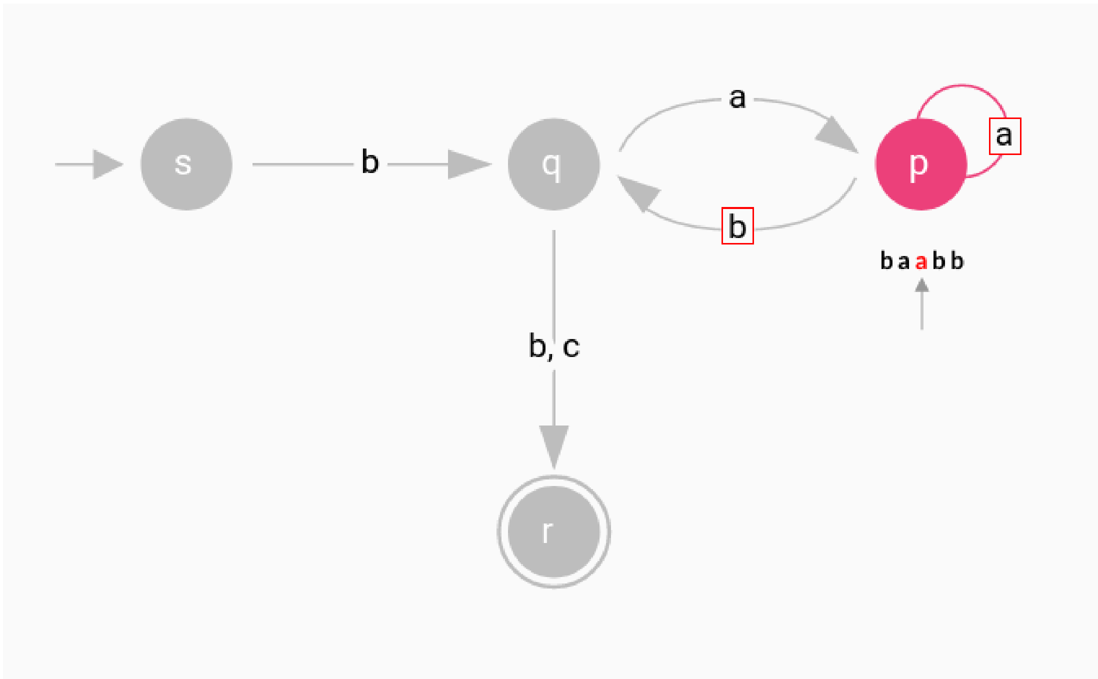
– Step 2



1. Next symbol of the input **baabb** is **a** and the current state is q.
2. Ask, is there any exiting transition from q that contains the symbol **a** or **b,c**?

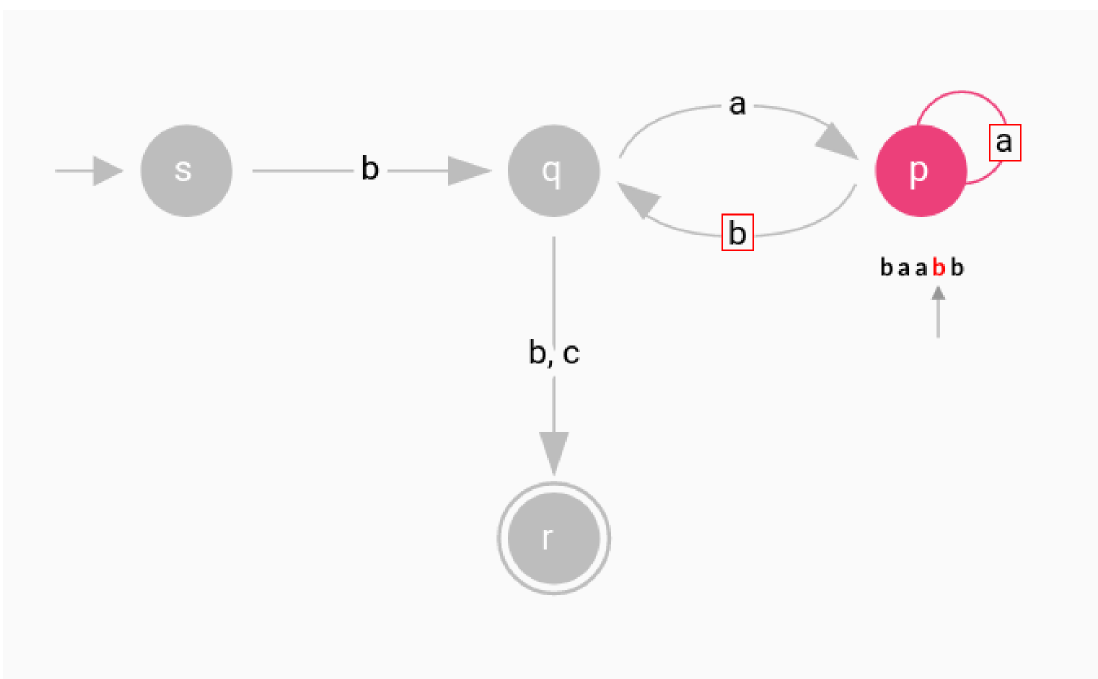
3. The answer is yes, and it's **a**. So move to p

- Step 3



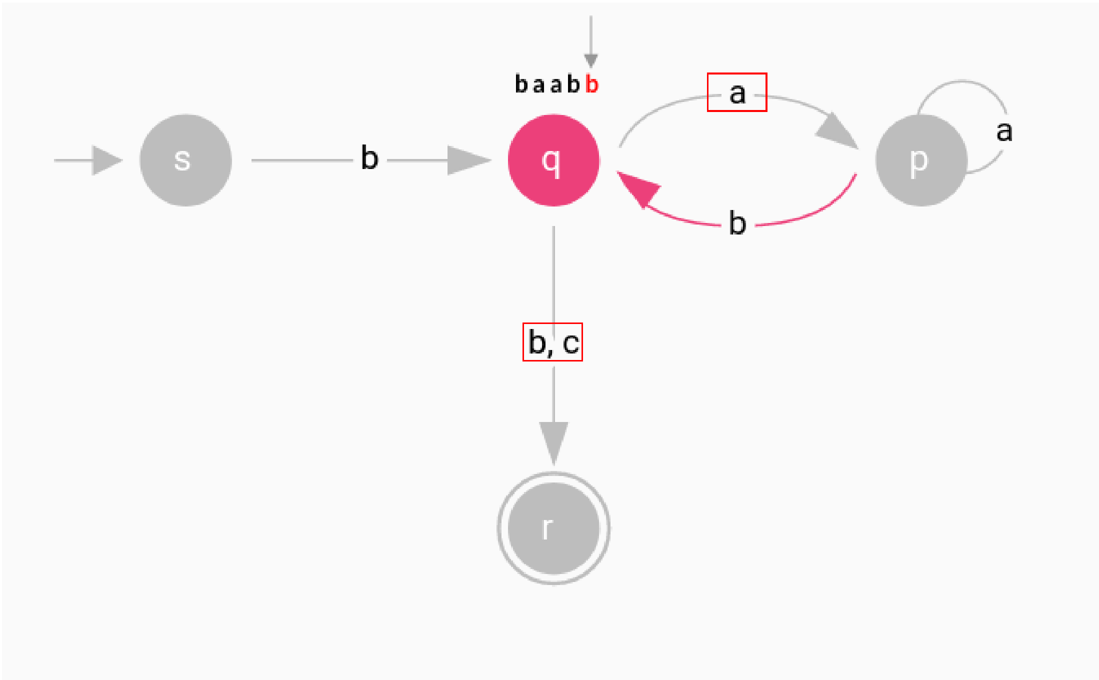
1. Next symbol of the input **baabb** is **a** and the current state is p.
2. Ask, is there any exiting transition from p that contains the symbol **a** or **b**?
3. The answer is yes, and it's **a**. So move to p

- Step 4



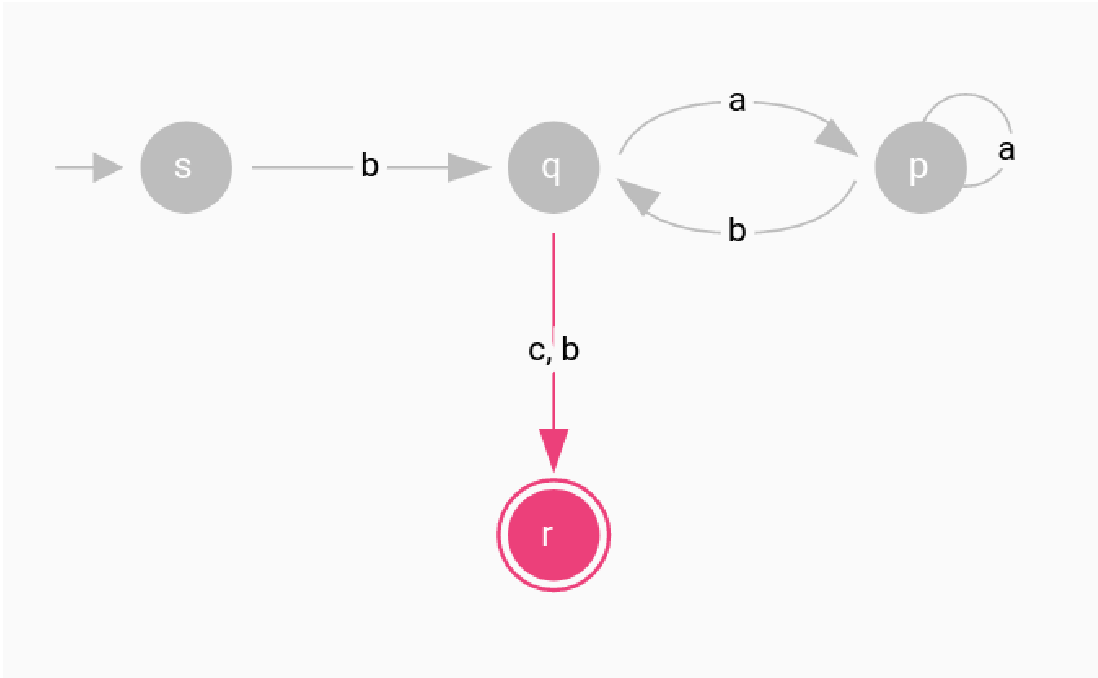
1. Next symbol of the input **baabb** is **b** and the current state is p.
2. Ask, is there any exiting transition from p that contains the symbol **a** or **b**?
3. The answer is yes, and it's **b**. So move to q

– Step 5



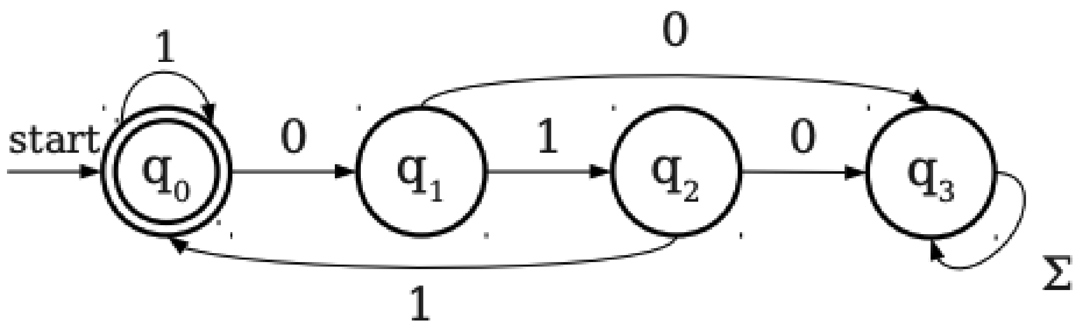
1. Next symbol of the input **baabb** is **b** and the current state is q.
2. Ask, is there any exiting transition from q that contains the symbol **a** or **b,c**?
3. The answer is yes, and it's **b**. So move to r

– Step 6



1. Next symbol of the input **baabb** is **b** and the current state is r.
2. Ask, if it satiesfies the accepting or final state (i.e, has the end of string been reached?). If so, the output is accept. Otherwise, it's reject.

- Formal Languages
  - is a <u>subset</u> of all possible words $\Sigma*$ formed by symbols of alphabet Σ.
    - $\Sigma*$ is set of all possible strings over the alphabet Σ.
    - i.e. $\Sigma = \{a, b\}$, $\Sigma* = \{a, b, aa, ab, ba, bb, aaa, aab, \cdots\}$
  - Example
    1. $L = \{w \mid w \text{ has at most seventeen 0's}\}$
    2. $L = \{w \mid w \text{ has equal number of 0's and 1's}\}$
    3. $L = \{x \in \{a, b\}^* \mid \text{the number of as in } x \text{ is even}\}$
       - $*$ in $\{a, b\}^*$ means all possible combinations
       - i.e. $\{a, b, aa, ab, ba, bb, aaa, baa, aba, \cdots\}$
- Tabular DFAs
  - Example



$\delta =$

	0	1
${}^*q_0$	q_1	q_0
q_1	q_3	q_2
q_2	q_3	q_0
q_3	q_3	q_3

Note: * means it's an accepting state

Question 2

•

Rough Works:

1. Prove that M_1 accepts L_1

First, define Σ^* as the smallest set such that

(a) $\epsilon \in \Sigma^*$

(b) $s \in \Sigma^* \Rightarrow sa \in \Sigma^* \wedge sb \in \Sigma^*$

I will prove that M_1 accepts L_1.

Define $P(s)$ as:

$$P(s) : \delta^*(E, s) = \begin{cases} E & \text{if } s \text{ has an even number of } as \\ O & \text{if } s \text{ has an even number of } as \end{cases} \tag{1}$$

I will prove $\forall s \in \Sigma^*$, $P(s)$ by structural induction.

1. Basis Case

   $|\epsilon| = 0$, an even number, and $\delta^*(E, \epsilon) = E$ so the implication in the first line of the invariant is true in this case. Also since $|\epsilon|$ is not odd, the implication in the second line of the invariant is vacuously true. So $P(\epsilon)$ holds.

2. Inductive Step

   Let $s \in \Sigma^*$ and assume $P(s)$. I will show that $P(sa)$ and $P(sb)$ follow. There are two cases to consider:

   1. Case sa

      Then,

$$\delta^*(E, sa) = \delta(\delta^*(E, s), a) = \begin{cases} \delta(E, a) & \text{if } s \text{ has even number of } as \\ \delta(O, a) & \text{if } s \text{ has odd number of } as \end{cases} \qquad \text{[By } P(s)\text{]} \tag{2}$$

$$= \begin{cases} O & \text{if } sa \text{ has odd number of } as \\ E & \text{if } sa \text{ has even number of } as \end{cases} \qquad \text{[One more } a\text{]} \tag{3}$$

   2. Case sb (Let's first start with this)

      Then,

$$\delta^*(E, sb) = \delta(\delta^*(E, s), b) = \begin{cases} \delta(E, b) & \text{if } s \text{ has even number of } as \\ \delta(O, b) & \text{if } s \text{ has odd number of } as \end{cases} \qquad \text{[By } P(s)\text{]} \tag{4}$$

$$= \begin{cases} E & \text{if } sb \text{ has odd number of } as \\ O & \text{if } sb \text{ has even number of } as \end{cases} \qquad \text{[One more } b\text{]} \tag{5}$$

   So $P(sa)$ and $P(sb)$ follow.

   The first line of the invariant ensures that all strings with an even number as are accepted. The contrapositive of the second line of the invariant ensures that any string that does not drive the machine to state O does not have an odd number of as, in other words all strings that drive the machine to state E have an even number of as. So M_1 accepts L_1.

2. Prove that M_2 accepts L_2

   Define $P(s)$ as:

$$P(s) : \delta^*(0, s) = \begin{cases} 0 & \text{if } |s| \equiv 0 \mod (3) \\ 1 & \text{if } |s| \equiv 1 \mod (3) \\ 2 & \text{if } |s| \equiv 2 \mod (3) \end{cases} \tag{6}$$

   I prove $\forall s \in \Sigma^*$, $P(s)$ by structural induction.

   1. Basis Case (Let's give this a shot)

   2. Inductive Step (Let's also try this)

3. Prove that $M_{1\wedge 2}$ accepts $L_1 \cap L_2$

   Denote the states for M_1 as Q_1, the states for M_2 as Q_2, their respective transition functions as δ_1 and δ_2, and the transition function for $M_{1\wedge 2}$ and the transition function for $M_{1\wedge 2}$ as $\delta_{1\wedge 2}$. Inspection of $\delta_{1\wedge 2}$ shows that if $(q_1, q_2, c) \in Q_1 \times Q_2 \times \Sigma$, then $\delta_{1\wedge 2}((q_1, q_2), c) = (\delta_1(q_1, c), \delta_2(q_2, c))$. Thus, the following invariant follows by simplying taking conjunctions of the invariants of the component machines, for any $s \in \Sigma^*$.

$$P(s) : \delta^*((E, 0), s) = \begin{cases} (E, 0) & \text{if } s \text{ has an even number of } as \wedge |s| \equiv 0 \mod 3 \\ (E, 1) & \text{if } s \text{ has an even number of } as \wedge |s| \equiv 1 \mod 3 \\ (E, 2) & \text{if } s \text{ has an even number of } as \wedge |s| \equiv 2 \mod 3 \\ (O, 0) & \text{if } s \text{ has an even number of } as \wedge |s| \equiv 0 \mod 3 \\ (O, 1) & \text{if } s \text{ has an even number of } as \wedge |s| \equiv 1 \mod 3 \\ (O, 2) & \text{if } s \text{ has an even number of } as \wedge |s| \equiv 2 \mod 3 \end{cases} \tag{7}$$

   The implication on the first line ensures that all strings with an even number of as and a length that is a multiple of 3 end up in state $(e, 0)$. The contrapositive of the implications on the other lines ensure that nay string that does not derive the machine to one of those 5 states must have an even number of as and a length that is a multiple of 3. Hence $M_{1\wedge 2}$ accepts $L_1 \cap L_2$.