Module 13

Intro Finite State Machines (FSM) & Regular Language

Language: A language, L is a set of strings composed from alphabet (Σ).

Alphabet (Σ): It is a finite set of symbols. $\Sigma = \{a, b\}$, $\Sigma = \{0, 1\}$

String: It is a sequence of symbols which could be used to model language, codes, etc.

Finite State Machine (FSM)

State: Condition of a machine defined by

- → Inputs that brought the machine to this state
- → current output of the machine.
- → response of machine to inputs.

FSM (Finite State Automata), $M = \{S, \Sigma, q_0, F, \delta\}$

S: a finite set states.

δ = the transition function. $(1 \xrightarrow{a} 2)$

Σ: the alphabet

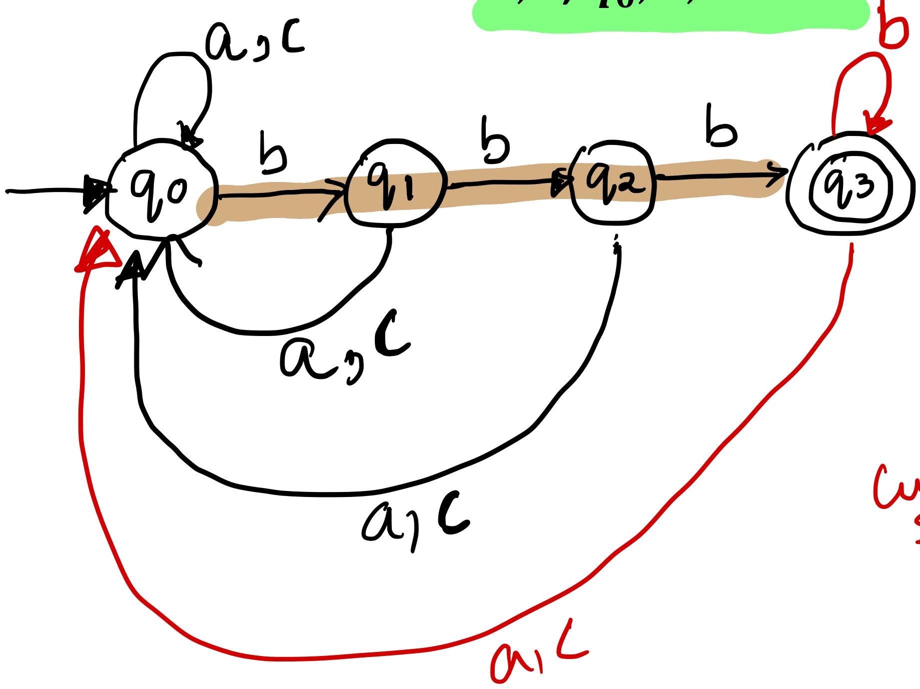
q_0: q_0 → q_0

F: final states $\subseteq S$ q_F

<u>Ex 1</u>: Let $\Sigma = \{a, b, c\}$. Let L_1 = The set of strings ending in bbb.

Draw machine. Define S, Σ, q_0, F, and δ

bbbb



acceptable strings:

bbb, a bbb, aa bbb

bbbb, ac bbb, c bbb

Not acceptable: a ba bbb

bbba x

bbbac x

$S = \{q_0, q_1, q_2, q_3\}$

$\Sigma = \{a, b, c\}$

$q_0 = q_0$

$F = q_3$

Current state δ	a	b	c
q_0	q_0	q_1	q_0
q_1	q_0	q_2	q_0
q_2	q_0	q_3	q_0
q_3	q_0	q_3	q_0

Current state

+

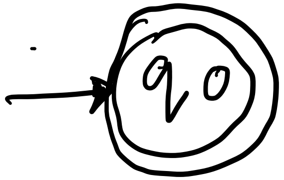
Input

$\Downarrow$

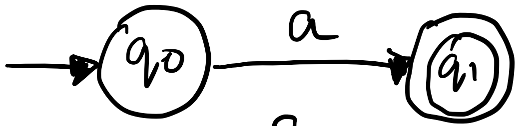
Next state.

Regular Language (Regular Expression) is composed of strings defined by regular expression.

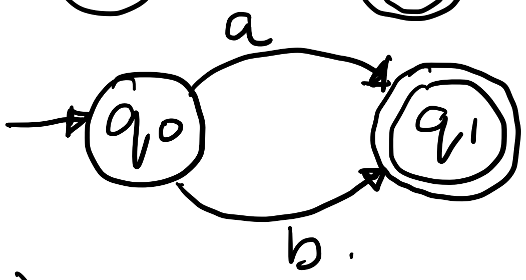
① λ (empty string) is Regular expression.



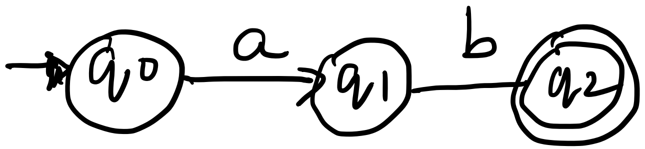
② $\forall a \in \Sigma$, a is regular expression.



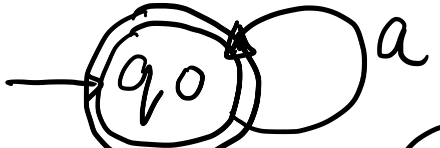
③ a + b is a regular expression
"OR" "a or b"



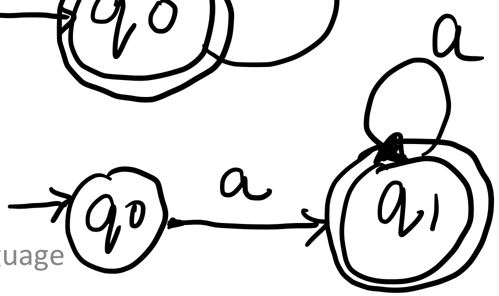
④ a b is a regular expression ("a AND b")
(AND)



⑤ a^* is a regular expression, * means zero or more repetation of a



⑥ a^+ is a regular expression, + means 1 or more repetation of a



λ
a
aa
aaa
aaaa
⋮

a^+ a^*

Regular Expression Examples   Let $\Sigma = \{a, b, c\}$

Ex 2: Let L_2 be the set of strings having zero or more as followed by zero or more bs.

Regular Expression: $L_2 = a^* b^*$

Potential Strings: λ, a, b, ab, aab, aa, aaaa..., bbb...,

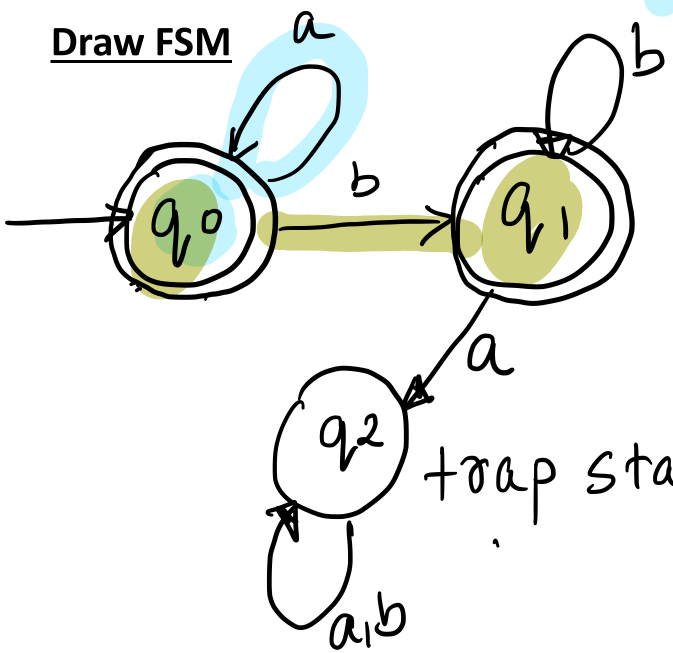
ba ✗

baaa.... ✗

babaa ✗

bbbb

Draw FSM



trap state/deadstate

$M_2 = \{S, \Sigma, q_0, F, \delta\}$

$S = \{q_0, q_1, q_2\}$

$\Sigma = \{a, b, c\}$

$q_0 = \{q_0\}$

$F = \{q_0, q_1\}$

Current State	a	b
q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_2	q_2

Ex 3: Let $L_3 = ab^+a$

Description: The language L_3 consists of strings begining with single a followed by one or more b and ending in single a.

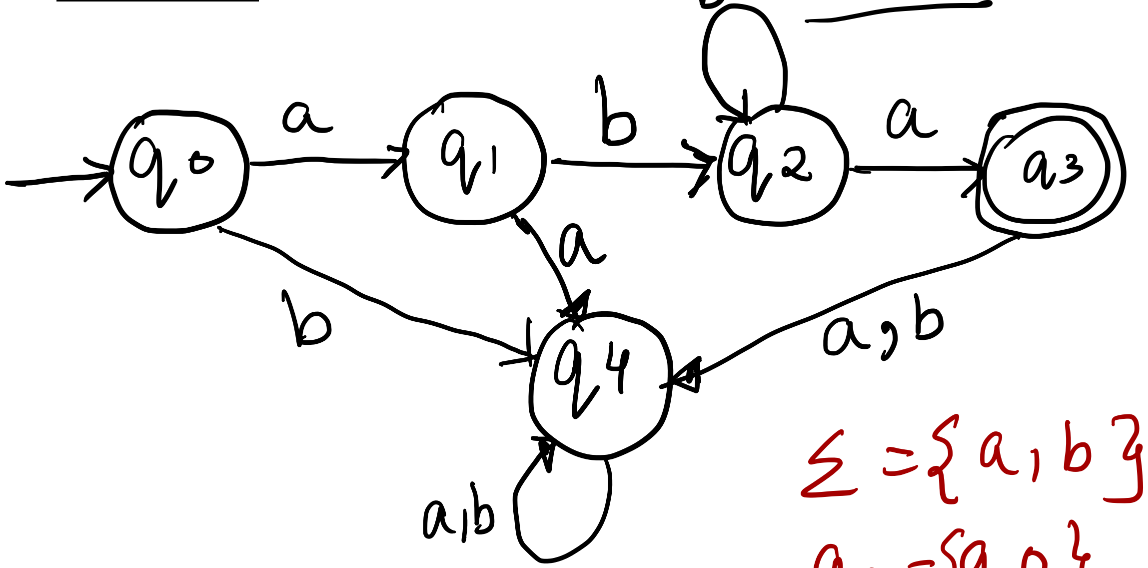
Potential Strings:

aba, abba, abbba, abb...ba.

aba

aa (Invalid), bbbbb, bab

Draw FSM



$\Sigma = \{a, b\}$

$q_0 = \{q_0\}$

$F = \{q_3\}$

$M = \{S, \Sigma, q_0, F, \delta\}$

$S = \{q_0, q_1, q_2, q_3, q_4\}$

δ	a	b
q_0	q_1	q_4
q_1	q_4	q_2
q_2	q_3	q_2
q_3	q_4	q_4
q_4	q_4	q_4

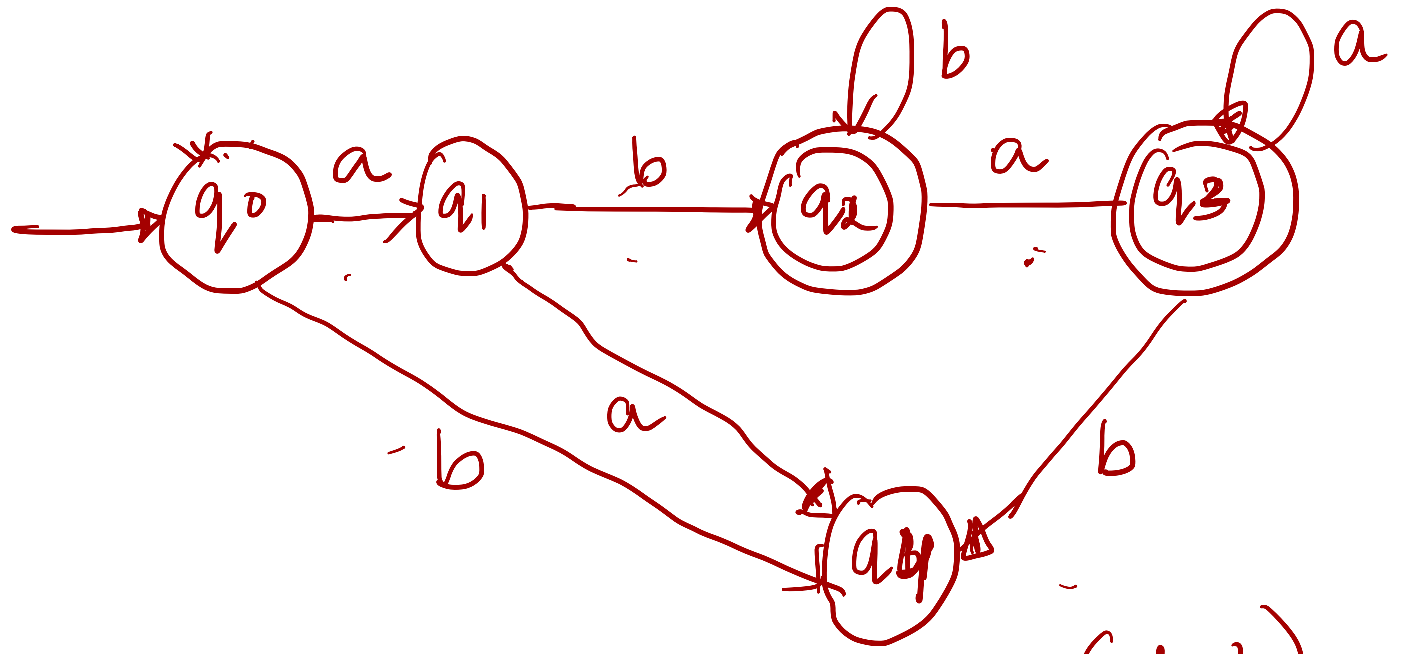
Relationship between FSM and Regular Languages

The job of FSM is to accept strings in the language and reject strings not in the language.

FSM and languages are equivalent.

① $a\, b^+ a^*$

Possible : ab , aba, abb, abaaa ...



b a b a

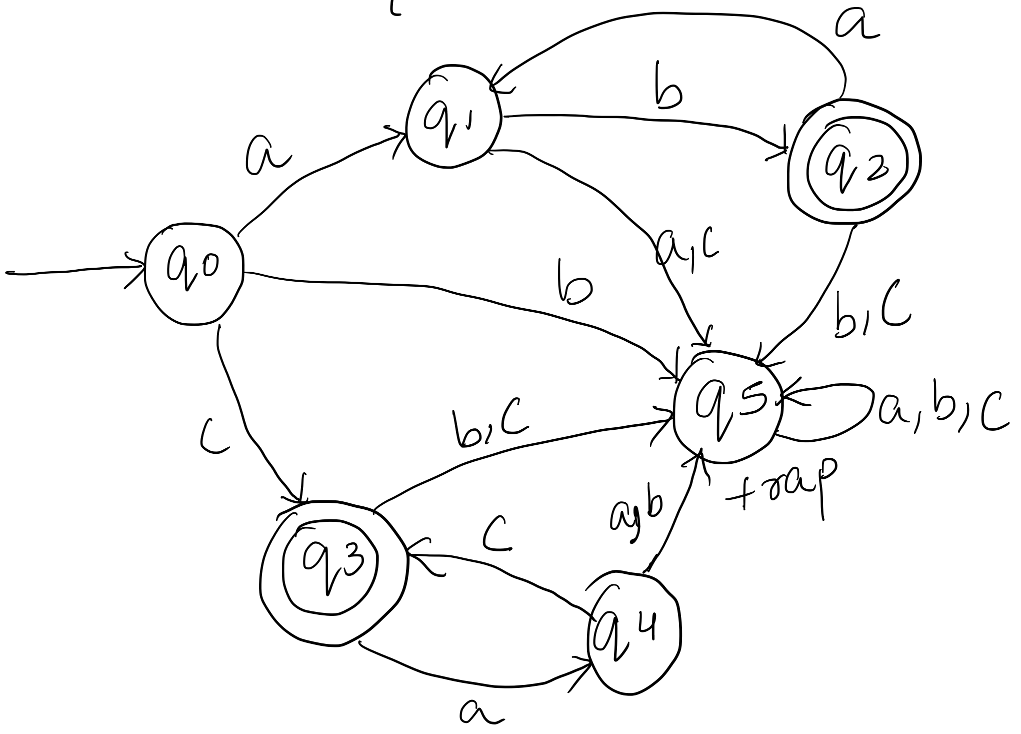
a a b a a x ...

$M : \{S, \Sigma, F, q_0, \delta\}$ (H.W)

② $(ab)^{+}$ ~~+~~ OR $c\,(ac)^{*}$ (record).

Valid string examples: ab, abab, ababab..., c, cac, cacac, ...

$M = \{S, \Sigma, q_0, F, \delta\}$



$S = \{q_0, q_1, q_2, q_3, q_4, q_5\}$

$\Sigma = \{a, b, c\}$

$q_0 = \{q_0\}$

$F = \{q_2, q_3\}$

δ / current state	a	b	c
q_0	q_1	q_5	q_3
q_1	q_5	q_2	q_5
q_2	q_1	q_5	q_5
q_3	q_4	q_5	q_5
q_4	q_5	q_5	q_3
q_5	q_5	q_5	q_5