Last time: DFAs.

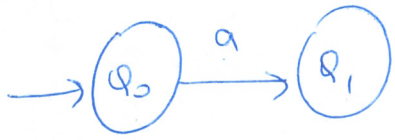
deterministic finite automata.

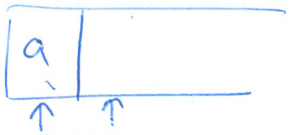
↓

for every state / character pair

there is one action (state transition)

M_1

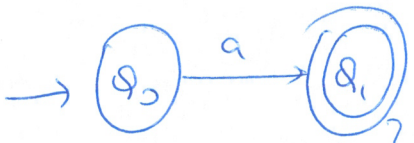




would reject "b"

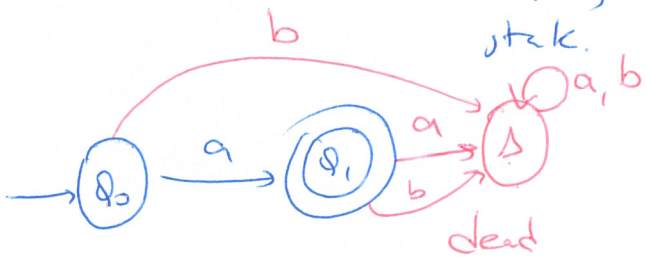
→ always reads in the forward direction (no idling, no going back)

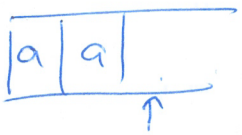
initially: state = q_0



would accept "a"

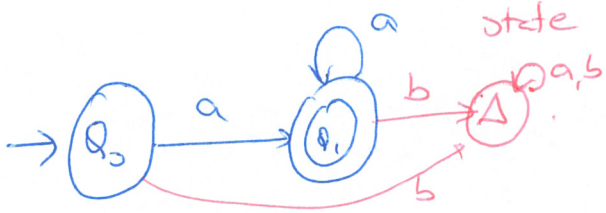
accepting state.

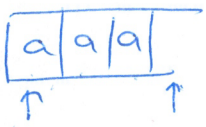




$\Sigma = \{a, b\}$

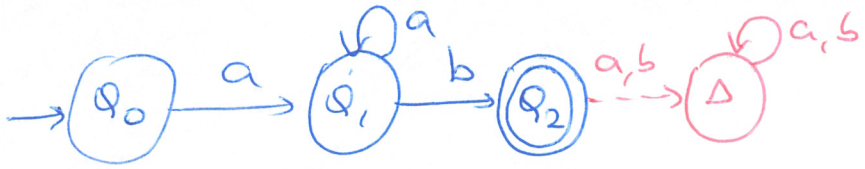
dead state





It has to read the entire input to be able to make a decision.

Δ, the dead state is a sink state.

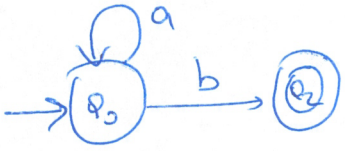


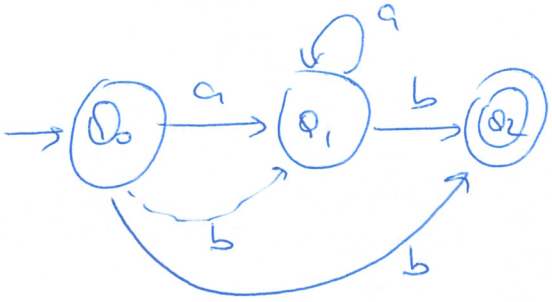
a^+b

a	a	b
↑		

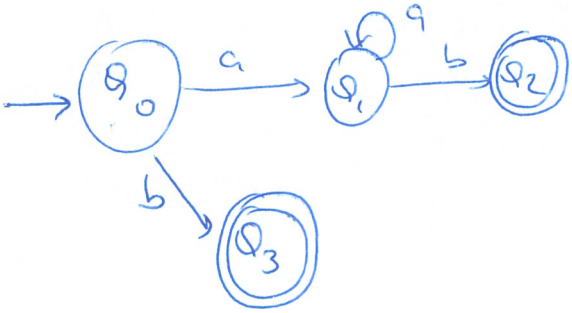
a	a	b	a
		↑	

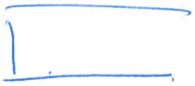
a^*b





$\lambda \notin a^*b$





λ: nothing on the tape

if q_0 is an accepting state then
λ is accepted

if q_0 is not an accepting state then
λ is not accepted.

A deterministic finite automaton is (DFA)

a quintuple $M = (Q, \Sigma, \delta, q_0, F)$

- $F \subseteq Q$: set of accepting states
- q_0 : initial state
- δ : transition function
- Σ : the input alphabet
- Q : set of states

$(\{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta, q_0, \{q_2, q_3\})$

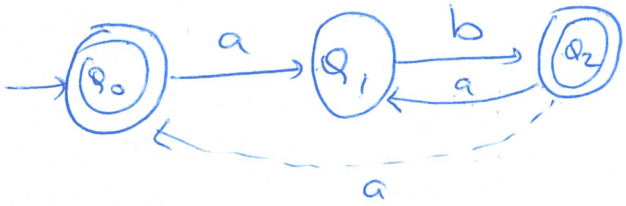
$\delta(q_i, a) = q_j \quad : Q \times \Sigma \to Q$

δ:

state	a	b
q_0	q_1	q_3
q_1	q_1	q_2
q_2	q_d	q_d
q_3	q_d	q_d

Can an accepting state transition to another state besides the dead state? Yes.

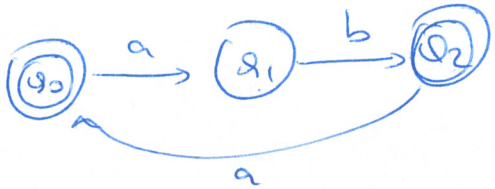
$(ab)^*$ $\quad \lambda, ab, abab, \ldots$



abab ✓

aba ✗

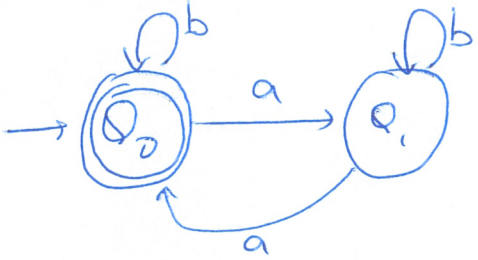
$ab \cup (aba)^*$



$aba \in L$?

$(aba)^*(ab \mid \lambda)$

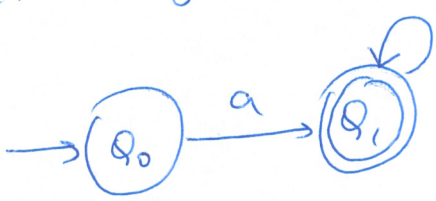
* Design a machine M with $\Sigma = \{a, b\}$ that accepts the strings with an even number of "a"s (0 is even).



bb

ab

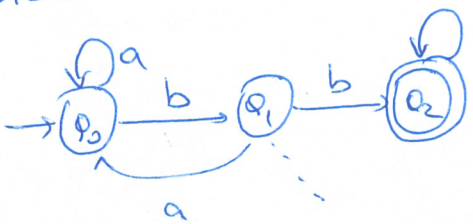
* Design a machine that accepts all strings that begin with an "a". $\Sigma = \{a, b\}$



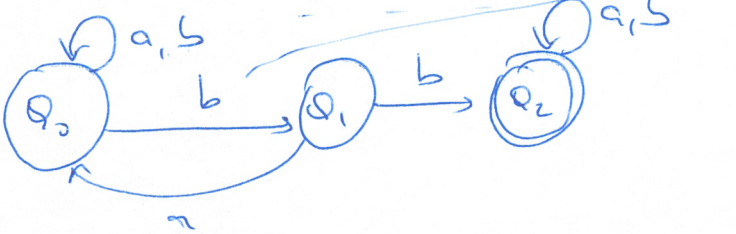
$a(a|b)^*$

} multiple symbols here represent an "or" transition on a or b to q_0.

* Design a machine that accepts all strings that contain a "bb". $\Sigma = \{a, b\}$



Does q_0 need a self loop with b? two "b" transitions from q_0: not a deterministic machine.



$\{\lambda\} \neq \{\}$

(hint for the homework)