DFAs can't compute every language

A non-regular language!

$$L = \{ a^n b^n \mid n \in \mathbb{N} \}$$

$$= \{ \varepsilon, ab, a^2b^2, a^3b^3, \dots \}$$

No DFA can accept L:

For a contradiction, assume $L = L(M)$ where

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

Consider $\delta^*(q_0, a^i)$ for various i.

Since there are finitely many states,

$$\exists\, i,j \quad \text{with} \quad \delta^*(q_0, a^i) = \delta^*(q_0, a^j) \quad i \neq j.$$

$$\text{Then} \quad \underset{\text{accepted}}{\delta^*(q_0, a^ib^i)} = \underset{\text{rejected}}{\delta^*(q_0, a^jb^i)}$$

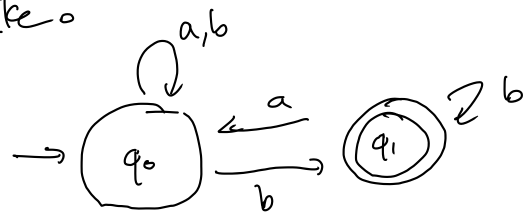
This is a contradiction since the same state can't be both accepting & rejecting.

New machine: NFA

Nondeterministic Finite Automaton

The NFA can produce several different results for the same input string.

Like:

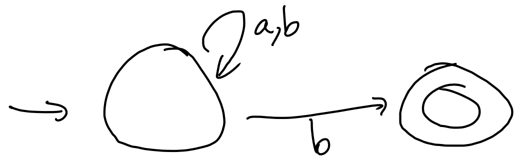


b is accepted,
bb also
a is rejected
ba also rejected.

Here, in q_0 if we read b,
we can go to q_1, or stay at q_0.

We say a string is accepted by a NFA if any of the possible final states is accepting.

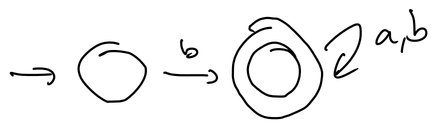
An NFA can have any number of outgoing arrows of each letter (even 0)



Accepts any string ending with b.

abaaab is accepted

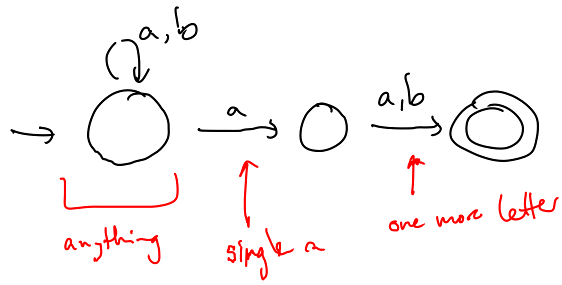
abaa is rejected.



Accepts any string starting with b.

$$\{ bx \mid x \in \{a,b\}^* \}$$

An NFA for strings with 2^{nd}-to-last letter a.



Make an NFA to accept:

$$\{ abxa \mid x \in \{a,b\}^* \}$$

$\{xb \mid x \text{ has an even \# of a's}\}$

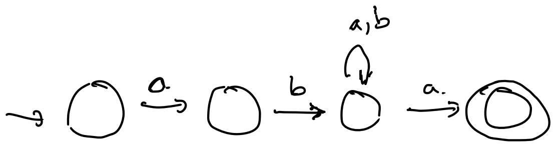
$\{a, b^2\}^*$ ← any string on a & b^2

like ab^2ab^2a or a^2b^2

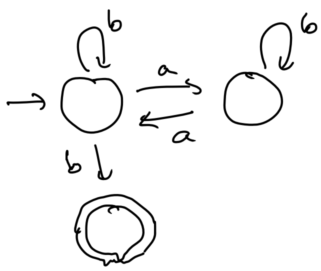
or b^6

not bab

$\{abxa \mid x \in \{a,b\}^*\}$



$\{xb \mid x \text{ has even \# of a's}\}$



$\{a, b^2\}^*$