NFAs accept the Regular Languages

$L(M_1) = \{\lambda, a^*b\}$

λ

Equivalence of Machines

Definition:

Machine M_1 (NFA) is equivalent to machine M_2 (FA)

if $L(M_1) = L(M_2)$

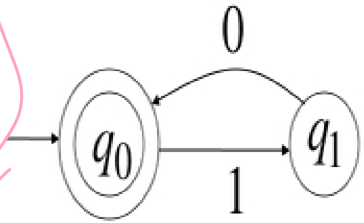
Proof by
- contradiction
- construction (more useful)

Example of equivalent machines

whenever there is an empty string you need q0 as an accepting state

NFA M_1

$$L(M_1)=\{10\}^*$$



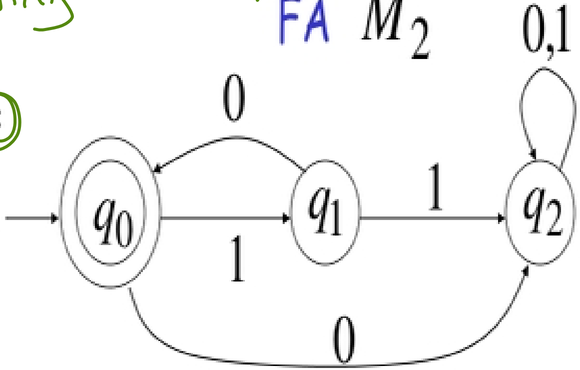
Will be in midterm this spesific info

can ask star or (+) or something else

he will ask writing these equation

FA M_2

$$L(M_2)=\{10\}^*$$



We will prove:

$\left\{ \begin{array}{l} \text{Languages} \\ \text{accepted} \\ \text{by NFAs} \end{array} \right\} = \left\{ \begin{array}{l} \text{Regular} \\ \text{Languages} \end{array} \right\}$

Languages accepted by DFAs

NFAs and DFAs have the same computation power

We will show:

$$\left\{\begin{array}{l}\text{Languages}\\ \text{accepted}\\ \text{by NFAs}\end{array}\right\} \supseteq \left\{\begin{array}{l}\text{Regular}\\ \text{Languages}\end{array}\right\}$$

$$\left\{\begin{array}{l}\text{Languages}\\ \text{accepted}\\ \text{by NFAs}\end{array}\right\} \subseteq \left\{\begin{array}{l}\text{Regular}\\ \text{Languages}\end{array}\right\}$$

Proof-Step 1

$$\left\{\begin{array}{l}\text{Languages}\\ \text{accepted}\\ \text{by NFAs}\end{array}\right\} \supseteq \left\{\begin{array}{l}\text{Regular}\\ \text{Languages}\end{array}\right\}$$

Proof: Every DFA is trivially an NFA

– you can convert the states to each other

⇓

Any language L accepted by a DFA is also accepted by an NFA

Proof-Step 2

$$\left\{\begin{array}{l}\text{Languages}\\ \text{accepted}\\ \text{by NFAs}\end{array}\right\} \subseteq \left\{\begin{array}{l}\text{Regular}\\ \text{Languages}\end{array}\right\}$$

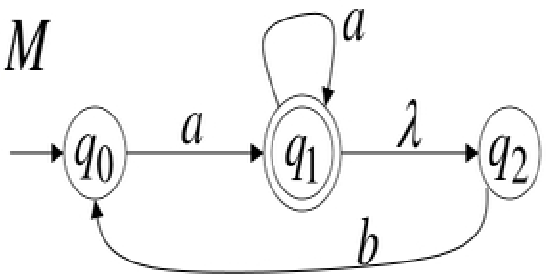
Proof: Any NFA can be converted to an equivalent DFA

⇓

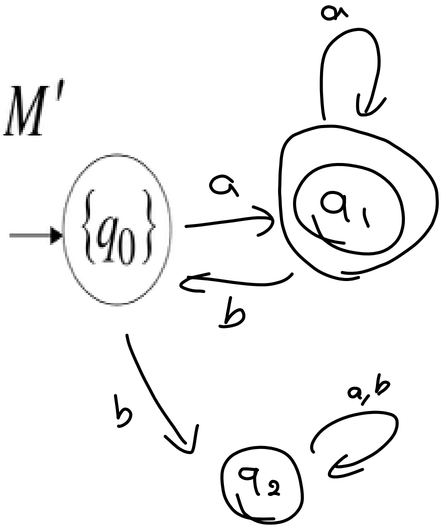
Any language L accepted by an NFA is also accepted by a DFA

Convert NFA to FA

NFA M



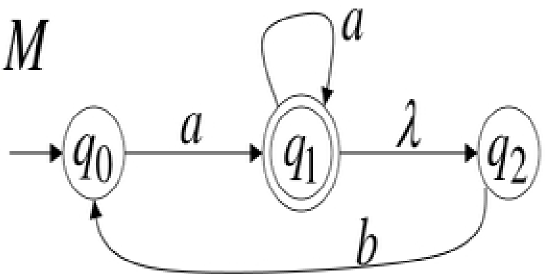
FA M'



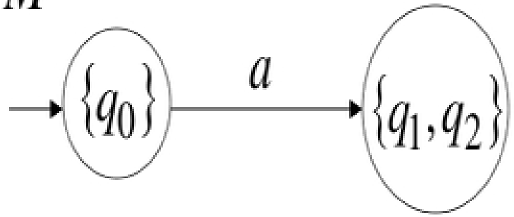
aba

Convert NFA to FA

NFA M

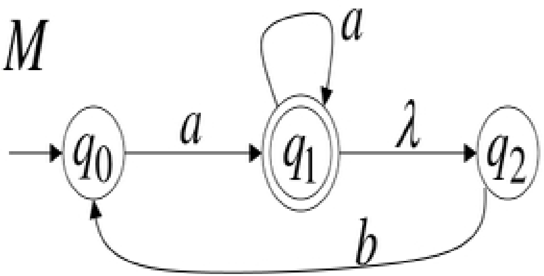


FA M'

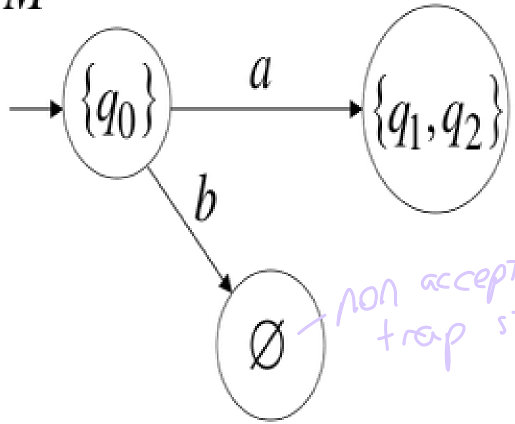


Convert NFA to FA

NFA M



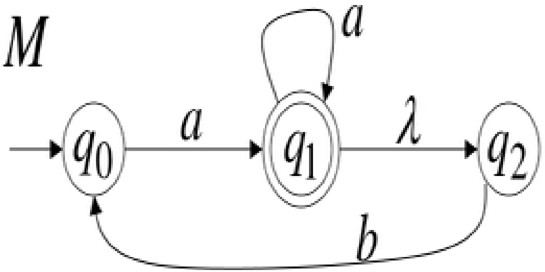
FA M'



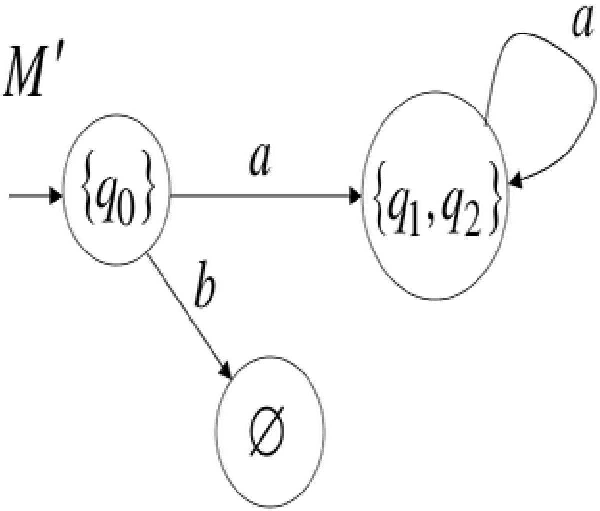
non accepting
trap state

Convert NFA to FA

NFA M

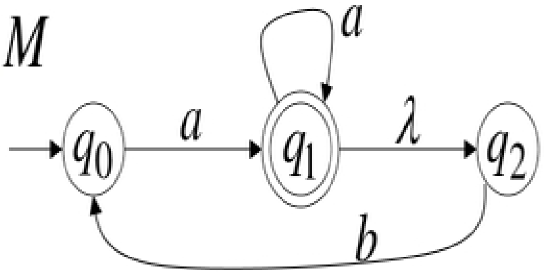


FA M'

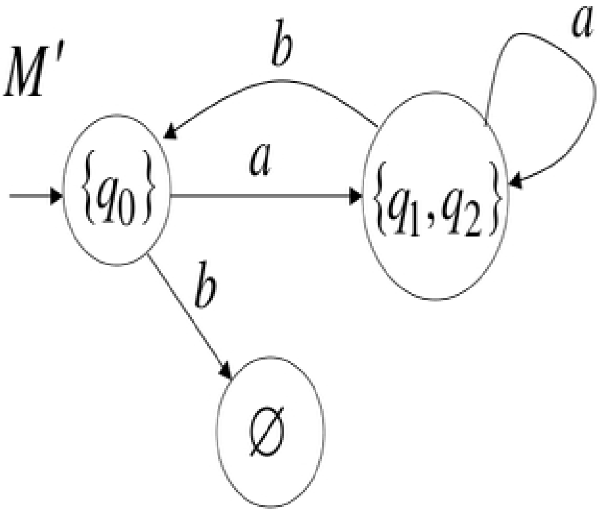


Convert NFA to FA

NFA M

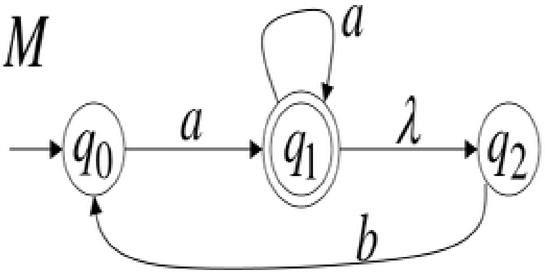


FA M'

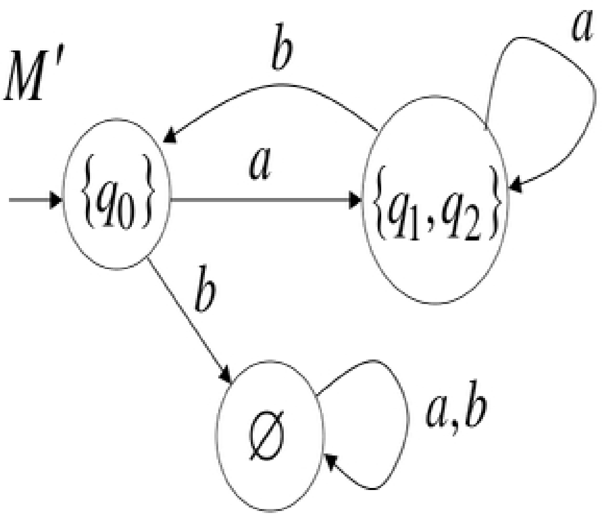


Convert NFA to FA

NFA M

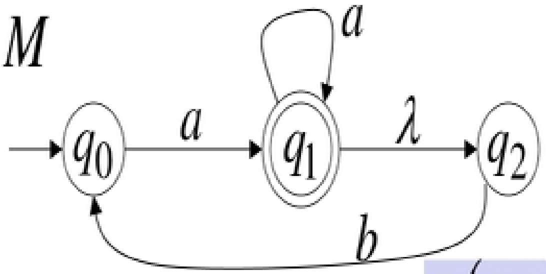


FA M'



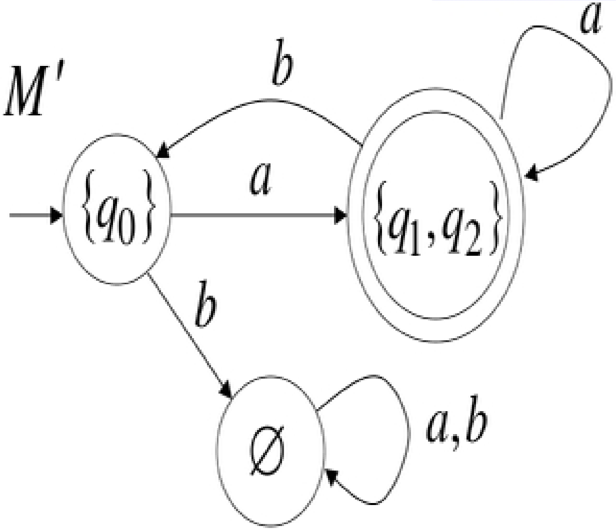
Convert NFA to FA

NFA M



$$L(M) = L(M')$$

FA M'



NFA to FA: Remarks

We are given an NFA M

We want to convert it
to an equivalent FA M'

With $L(M) = L(M')$

If the NFA has states

$$q_0, q_1, q_2, \ldots$$

the FA has states in the powerset

$$\emptyset, \{q_0\}, \{q_1\}, \{q_1, q_2\}, \{q_3, q_4, q_7\}, \ldots$$

Procedure NFA to FA

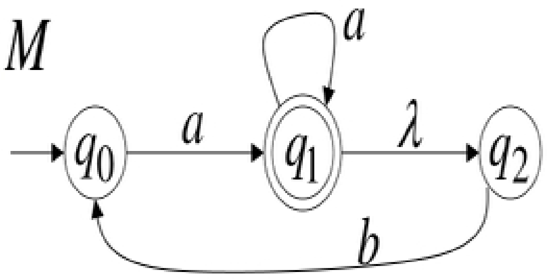
1. Initial state of NFA: q_0

Initial state of FA: $\{q_0\}$

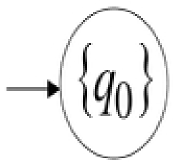
if there is lambda transition you'll add them to q_0

Example

NFA M



FA M'



Procedure NFA to FA

2. For every FA's state $\{q_i, q_j, \ldots, q_m\}$

Compute in the NFA

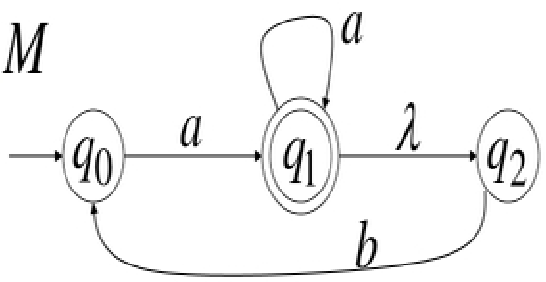
$$\left.\begin{array}{l} \delta^*(q_i, a), \\ \delta^*(q_j, a), \\ \ldots \end{array}\right\} = \{q'_i, q'_j, \ldots, q'_m\}$$

Add transition to FA

$$\delta(\{q_i, q_j, \ldots, q_m\},\ a) = \{q'_i, q'_j, \ldots, q'_m\}$$

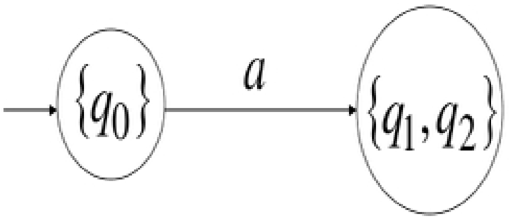
Example

NFA M



$$\delta^*(q_0, a) = \{q_1, q_2\}$$

FA M'



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

Procedure NFA to FA

Repeat Step 2 for all letters in alphabet,

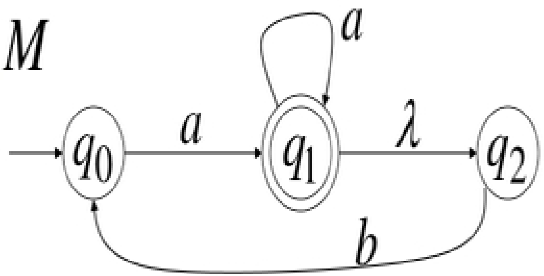
until

no more transitions can be added.

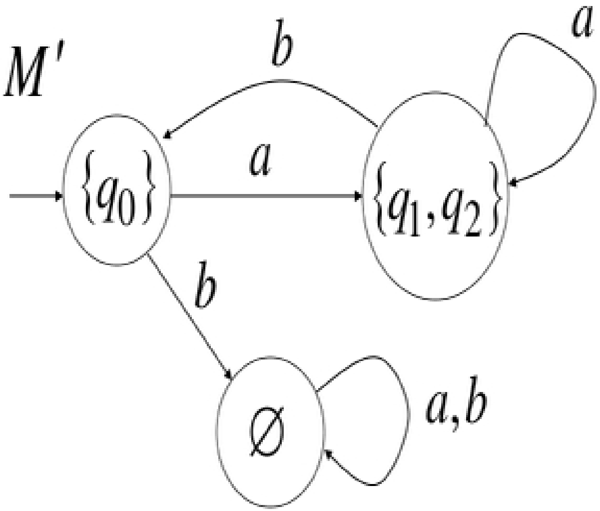
Be careful with lambda transitions, they are the things that makes it complicated

Example

NFA M



FA M'



Procedure NFA to FA

3. For any FA state $\{q_i, q_j, \ldots, q_m\}$

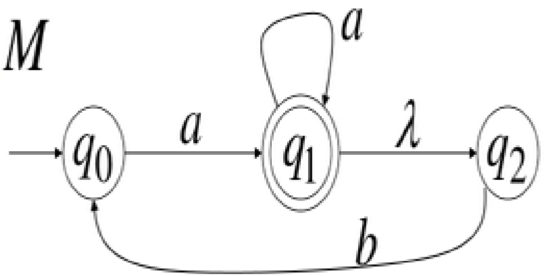
If q_j is accepting state in NFA

Then, $\{q_i, q_j, \ldots, q_m\}$

is accepting state in FA

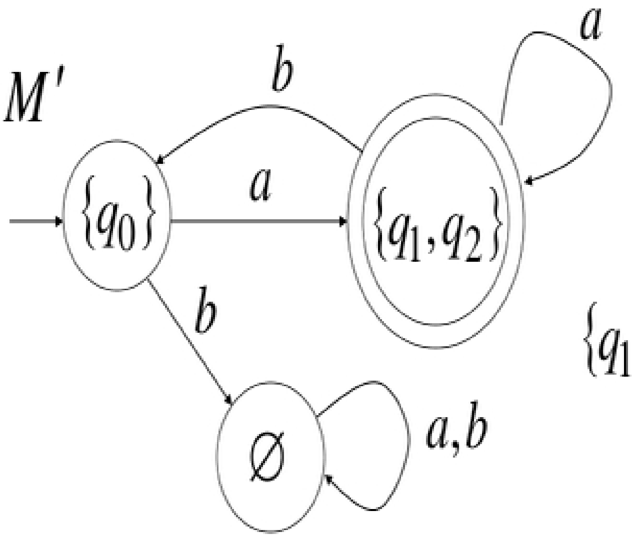
Example

NFA M



$q_1 \in F$

FA M'



$\{q_1, q_2\} \in F$

Theorem

Take NFA M

Apply procedure to obtain DFA M'

Then M and M' are equivalent :

$$L(M) = L(M')$$

Proof

$$L(M) = L(M')$$

$$\Updownarrow$$

$L(M) \subseteq L(M')$ AND $L(M) \supseteq L(M')$

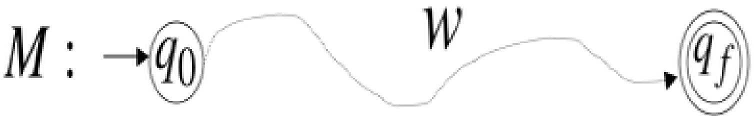
(no over this
proof

First we show: $L(M) \subseteq L(M')$

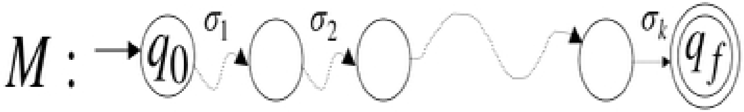
Take arbitrary: $w \in L(M)$

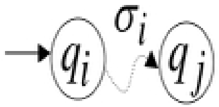
We will prove: $w \in L(M')$

$$w \in L(M)$$

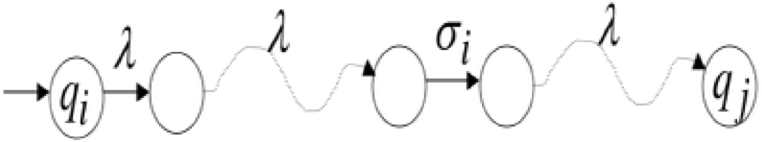


$$w = \sigma_1 \sigma_2 \cdots \sigma_k$$





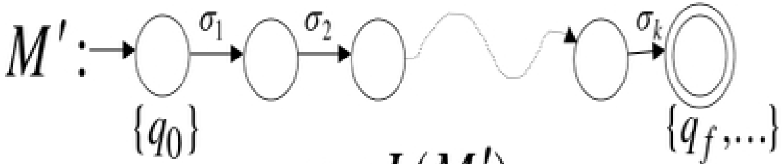
denotes



prop until here

We will show that if $w \in L(M)$

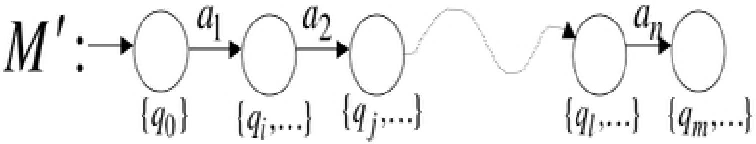
$$w = \sigma_1 \sigma_2 \cdots \sigma_k$$



$$w \in L(M')$$

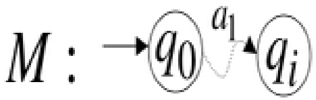
More generally, we will show that if in M:

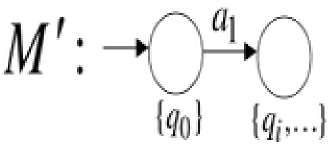
(arbitrary string) $v = a_1 a_2 \cdots a_n$



Proof by induction on $|v|$

Induction Basis: $v = a_1$

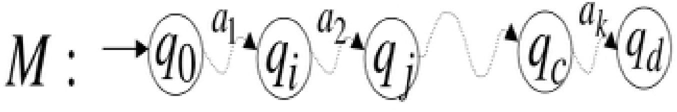


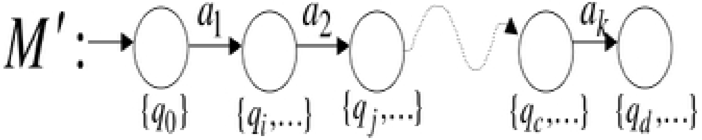


Is true by construction of M':

Induction hypothesis: $1 \leq |v| \leq k$

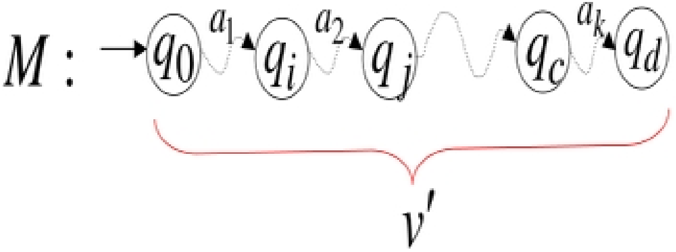
$$v = a_1 a_2 \cdots a_k$$

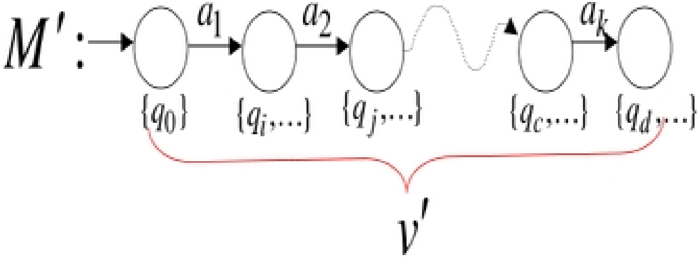




Induction Step: $|v| = k + 1$

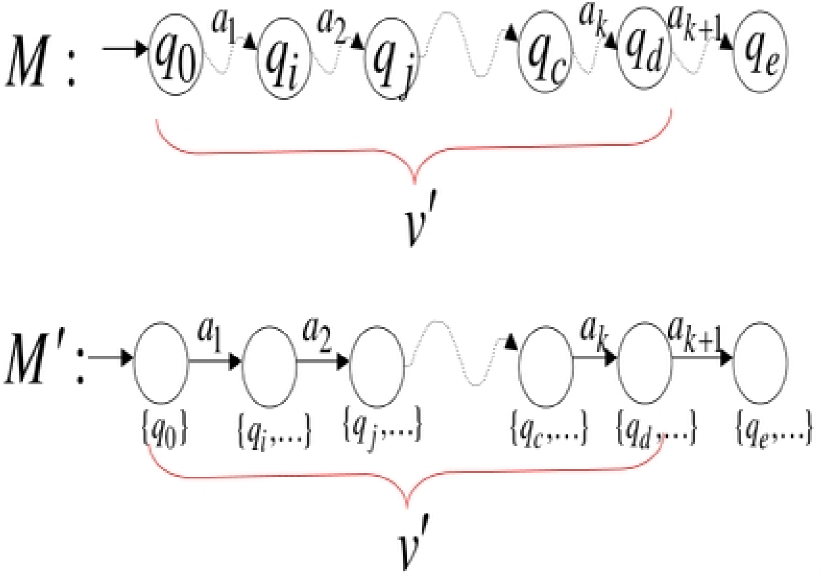
$$v = \underbrace{a_1 a_2 \cdots a_k}_{v'} a_{k+1} = v' a_{k+1}$$





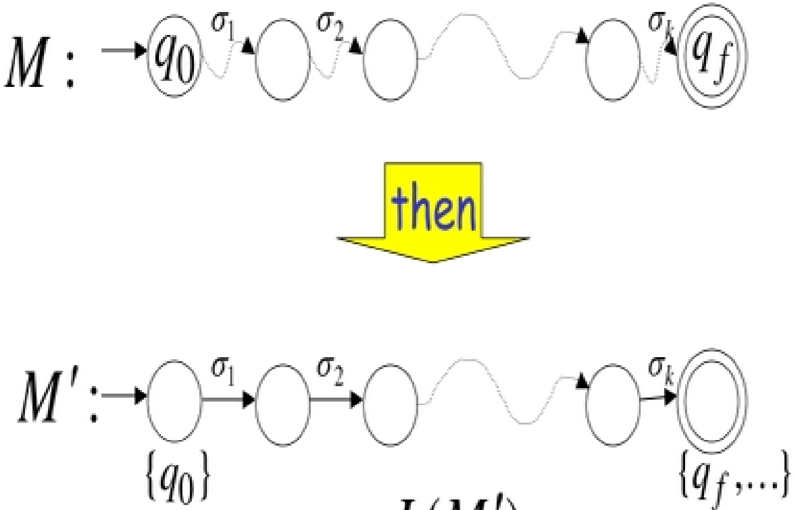
Induction Step: $|v| = k+1$

$$v = \underbrace{a_1 a_2 \cdots a_k}_{v'} a_{k+1} = v' a_{k+1}$$



Therefore if $w \in L(M)$

$$w = \sigma_1 \sigma_2 \cdots \sigma_k$$



$$w \in L(M')$$

We have shown: $L(M) \subseteq L(M')$

We also need to show: $L(M) \supseteq L(M')$

(proof is similar)

will ask in midterm

* can ask sorting in midterm?
search

Single Accepting State for NFAs

* You can have multiple accepting state in DFA's.
* Any NFA can be transform into multiple accepting state by adding another q and then doing lambda transitions.

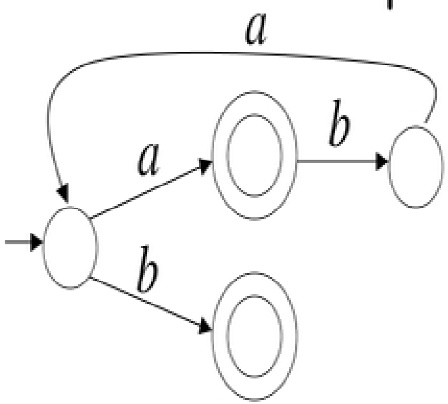
* search for a character in Java phyton c string?
midterm question posibilli.

Any NFA can be converted

to an equivalent NFA

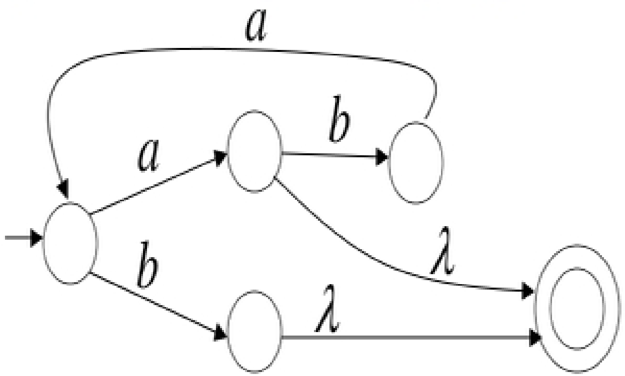
with a single accepting state

Example



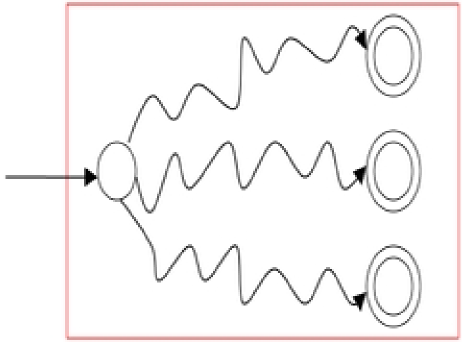
NFA

Equivalent Single Accepting State NFA

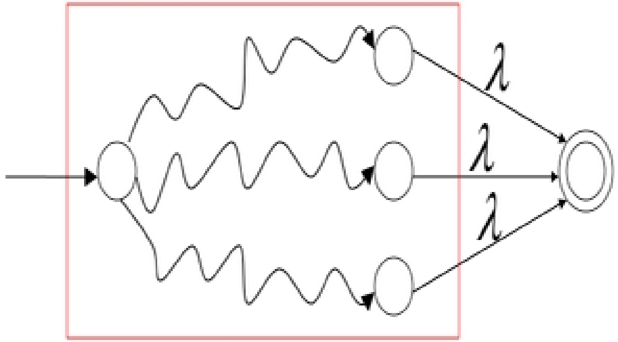


In General

NFA



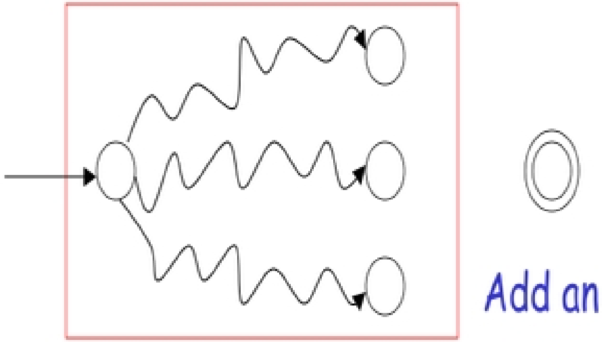
Equivalent NFA



Single accepting state

Extreme Case

NFA without accepting state



Add an accepting state without transitions