Lecture Nine

Topics that must be covered in this lecture:

- ***Regular language.***
- ***Properties of Language Classes***
- ***Representation of Languages***
- ***closure properties of regular languages***
- ***Decision properties for regular sets (membership, emptiness, finiteness).***

Regular language

Regular language: a language that can be defined by a regular expression is called a regular language.

- If L1 and L2 are regular language there are regular expression r1 and r2 that defined these language, then (r1+r2) is a regular expression that defines the language L1+L2.
- The language L1 L2 can be defined by the regular expression r1 r2.
- The language L1* can be defined by the regular expression r1*.

Therefore, all these set of words are definable by regular expressions and so are themselves regular languages.

Languages Accepted by FAs

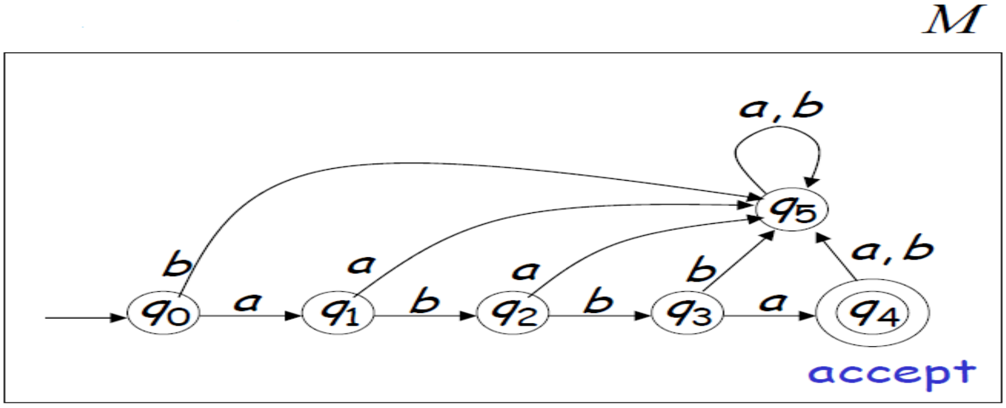
Definition:

The language L(M) contains all input strings accepted by FA M.

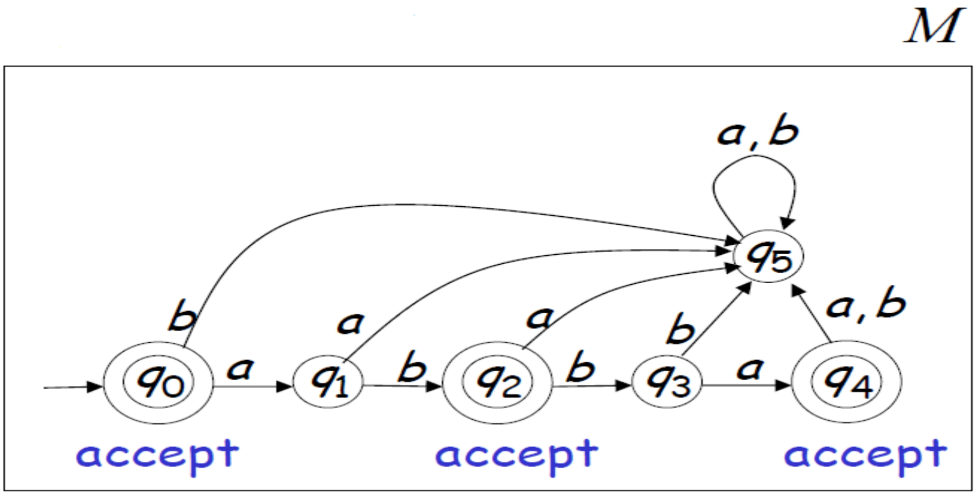
L(M)= { strings that drive to a final state}

Example:

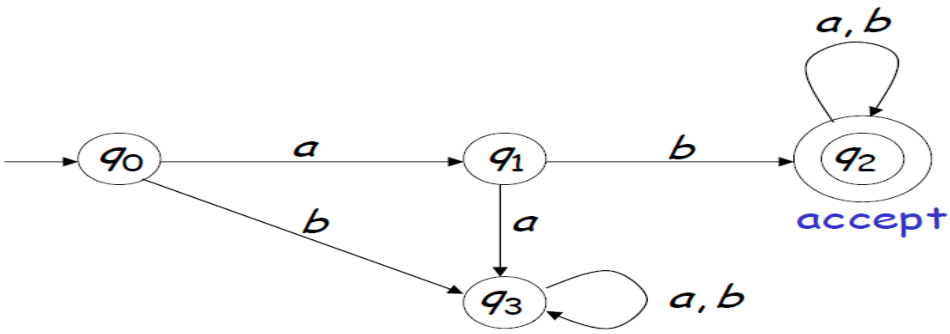
1- Let L(M)={abba}



2- L(M)={λ,ab,abba}



3- L(M)= { all strings with prefix ab}



Representation of Languages

□Representations can be formal or informal.
□Example (formal): represent a language by a RE or FA defining it.
□Example: (informal): a logical or prose statement about its strings:
□ {0^n1^n | n is a nonnegative integer}
"The set of strings consisting of same number of 0's followed by the same number of 1's."

Properties of Language Classes

□ A language class is a set of languages such as the regular languages.
□ Language classes have two important kinds of properties:

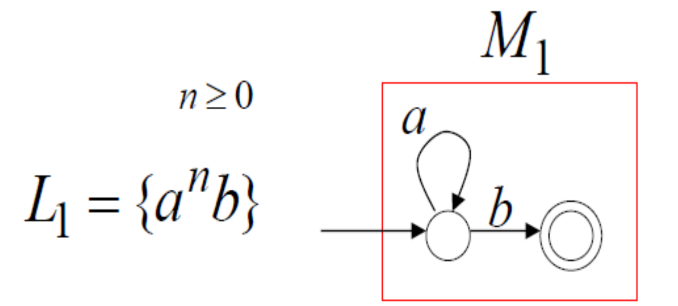
1. Closure properties.
2. Decision properties.

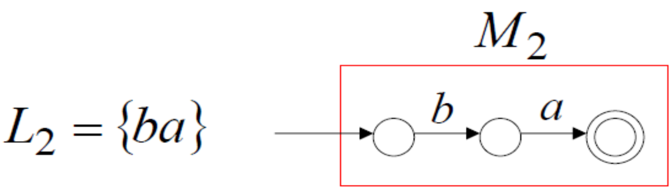
Closure Properties of Regular Language:

- A closure property of a language class says that given languages in the class, an operator (e.g., union) produces another language in the same class.
- Example: the regular languages are obviously closed under union, concatenation, and (Kleene) closure.
- We say: Regular languages are **closed under** these properties.

Regular language L1(M1)=L1

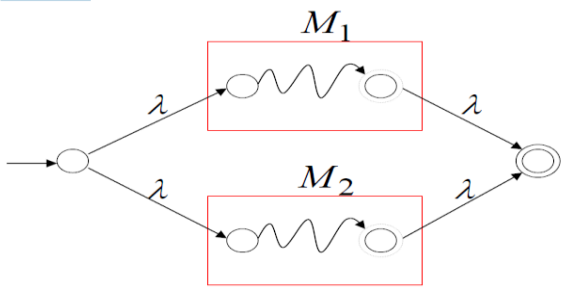
Example: suppose M1 recognize L1 and M2 recognize L2:





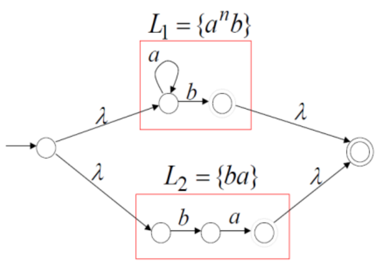
Union

NFA for L1∪ L2



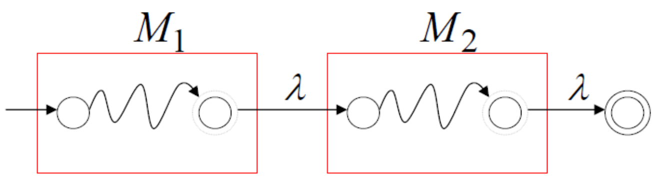
Example:

NFA with L1∪ L2=$\{a^n b\}$∪{ba}



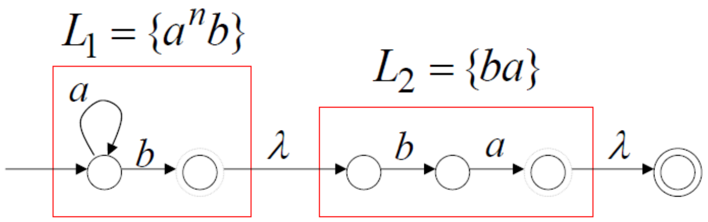
Concatenation

NFA for $L1L2$



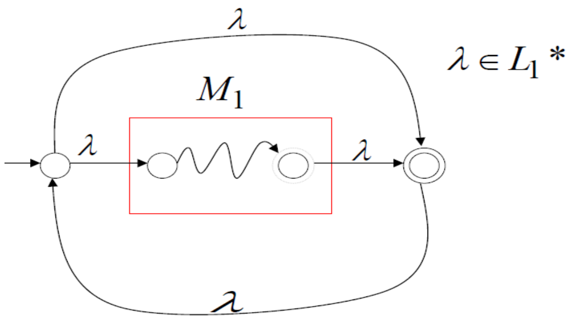
Example

NFA for L1L2=$\{a^nb\}\{ba\}=\{a^nbba\}$



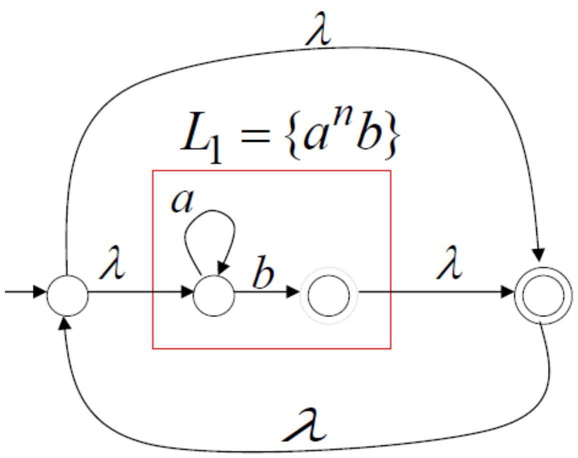
Star Operation

NFA for L1*

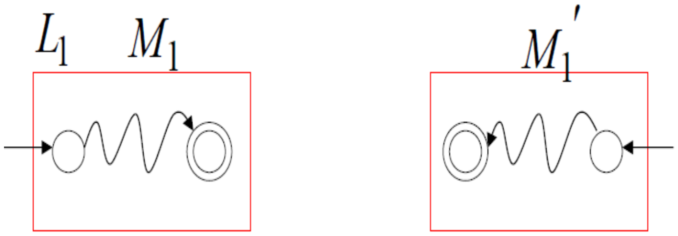


Example

NFA for L1*=$\{a^nb\}$*



Reverse: NFA $L1^R$

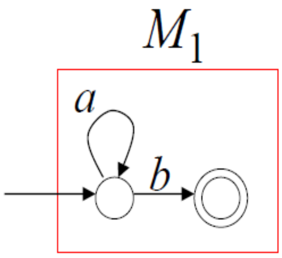


1.Reverse all transitions

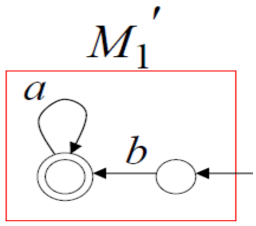
2. Make initial state final state and vice versa

Example

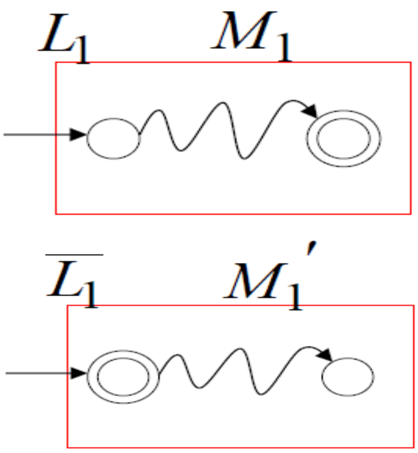
$$L_1 = \{a^n b\}$$



$$L_1^R = \{ba^n\}$$



Complement

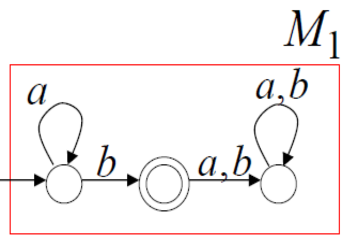


1.Take the **FA** that accepts *L*1

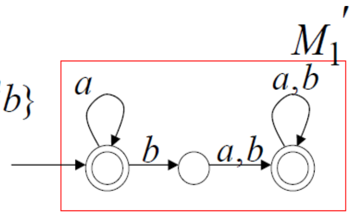
2. Make final states non-final, and vice-versa

Example

$L_1 = \{a^n b\}$



$\overline{L_1} = \{a,b\}^* - \{a^n b\}$



Intersection

NFA for L1∩ L2

Example

$L_1 = \{a^n b\}$ regular

$L_2 = \{ab, ba\}$ regular

⇒ $L_1 \cap L_2 = \{ab\}$ regular

Decision properties for regular sets (membership, emptiness, finiteness):

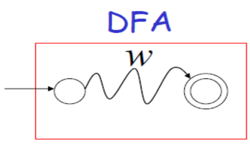
A decision property for a class of languages is an algorithm that takes a formal description of a language (e.g., a DFA) and tells whether or not some property holds.

□Example: Is language L empty?

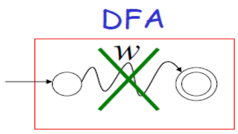
1. **Membership:**

   Question: given regular language L and string w how can we check if w ∈ L?

   Answer: take the DFA that accept L and check if w is accepted.



$w \in L$



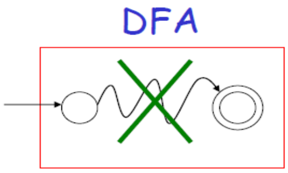
$w \notin L$

2- emptiness:

Question: given regular language L how can we check if L is empty: L=Φ?

Answer: take the DFA that accept L and check if there is any path from the initial state to a final state.

$L \neq \varnothing$

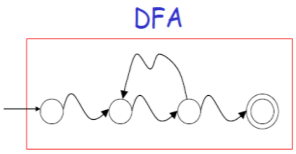


$L = \varnothing$

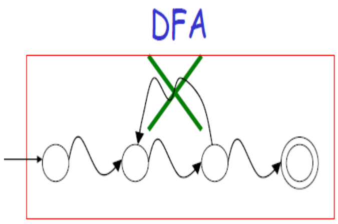
3- finiteness:

Question: given regular language L how can we check if L is finite?

Answer: take the DFA that accept L and check if there is a walk without cycle from the initial state to a final state.



L is infinite



L is finite