Theory of Computing

Dr. Osama fathy

Regular Languages

A regular language is a language that can be defined by a regular expression.

We study some properties of the class of regular languages.

Example: Every finite language is regular

Theorem: Let L_1 and L_2 be two regular languages. The languages L_1+L_2, L_1L_2, and L_1^* are regular languages.

Proof 1: If L_1 and L_2 are regular languages, then there exist regular expressions $\mathbf{r_1}$ and $\mathbf{r_2}$ that define them.

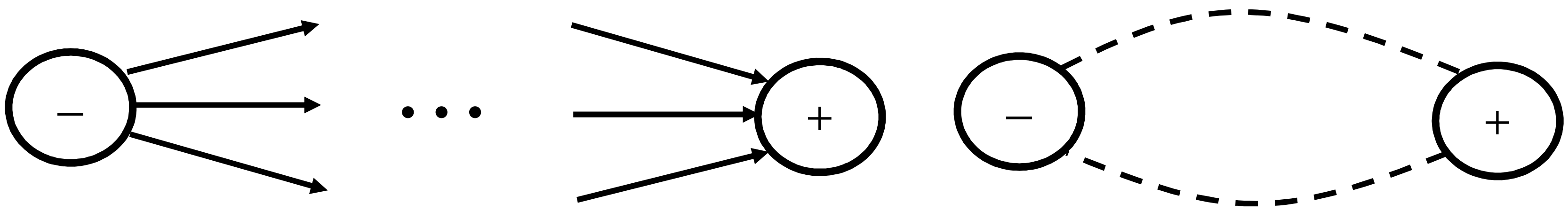
- The language associated with $\mathbf{r_1+r_2}$ is L_1+L_2.
- The language associated with $\mathbf{r_1r_2}$ is L_1L_2.
- The language associated with $\mathbf{r_1^*}$ is L_1^*.

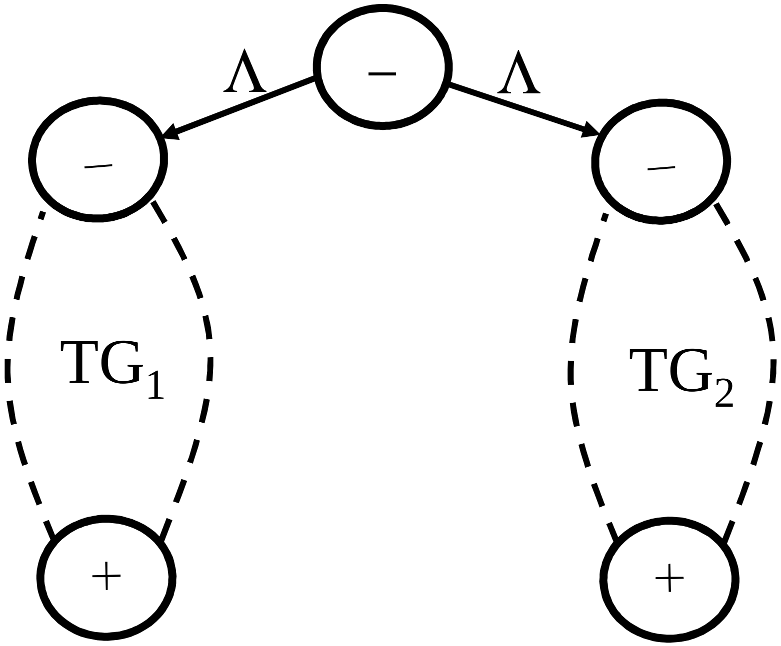
L_1+L_2, L_1L_2, and L_1^* can be defined by regular expressions

Thus, by definition, they are regular languages.

Proof 2: using Kleene's theorem

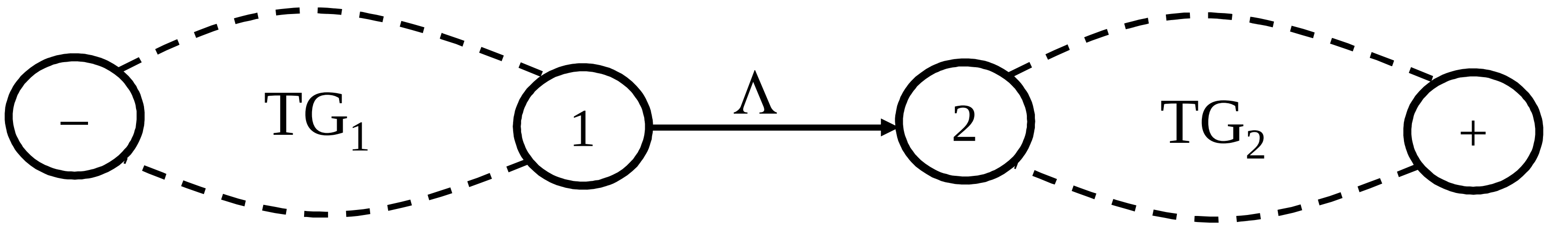
Because L_1 and L_2 are regular languages, there are regular expressions $\mathbf{r_1}$ et $\mathbf{r_2}$ that define them. By Kleene's theorem, there are transition graphs that accept them. We can transform these transition graphs into transition graphs with one start state and one final state. Let TG_1 and TG_2 be two transition graphs of this form that accept L_1 et L_2.



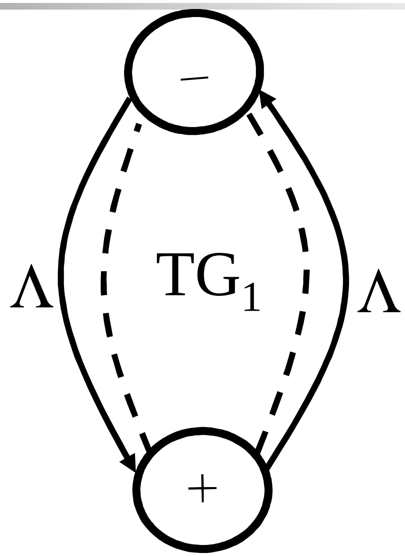


A transition graph that accepts L_1+L_2.

A transition graph that accepts L_1L_2.



A transition graph that accepts L_1^*



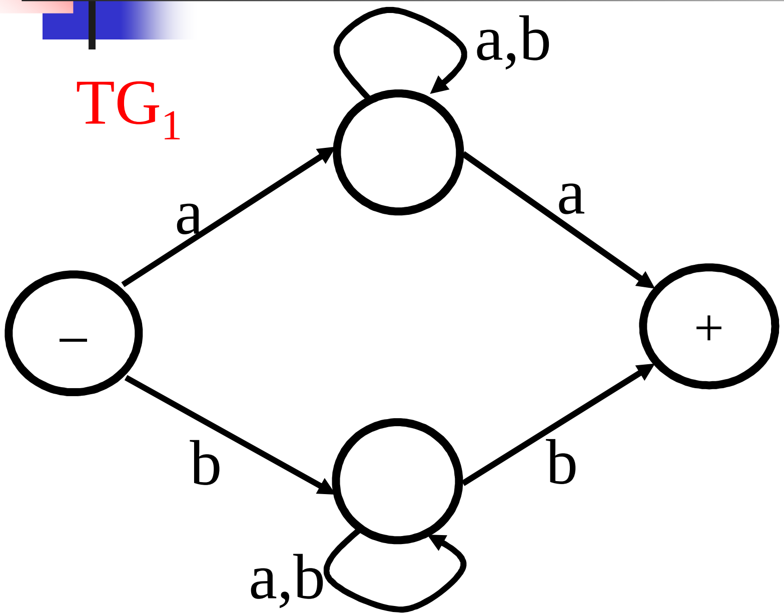
There exist transition graphs that accept L_1+L_2, L_1L_2, and L_1^*.

Thus, there are regular expressions that define them (by Kleene's theorem). Thus

$$L_1+L_2,\ L_1L_2 \text{ and } L_1^*$$

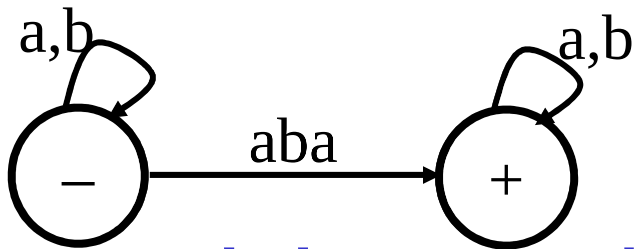
are regular languages (by definition).

TG$_1$



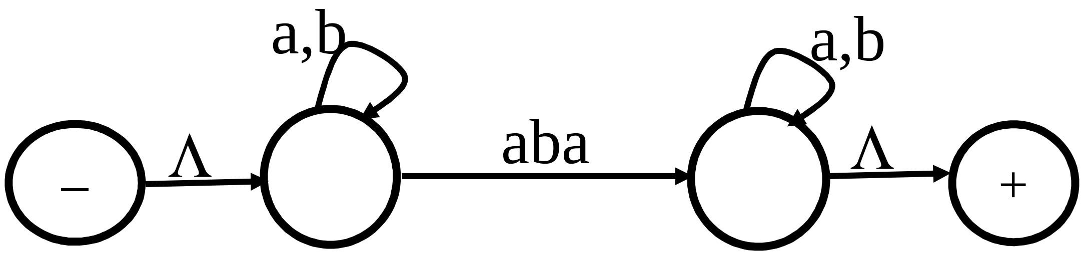
Words that begin and end with the same letter. **a(a+b)*a + b(a+b)*b**

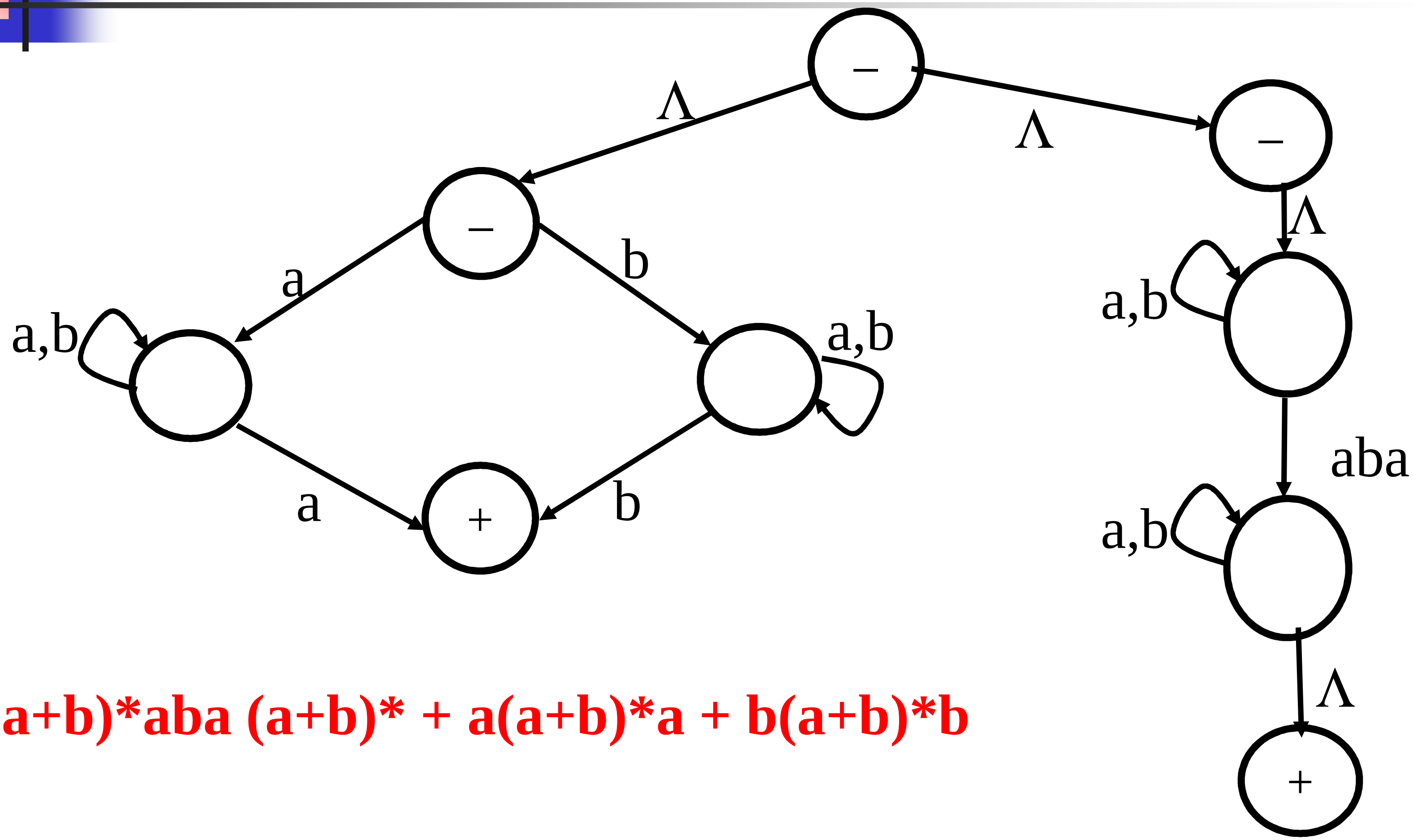
TG$_2$



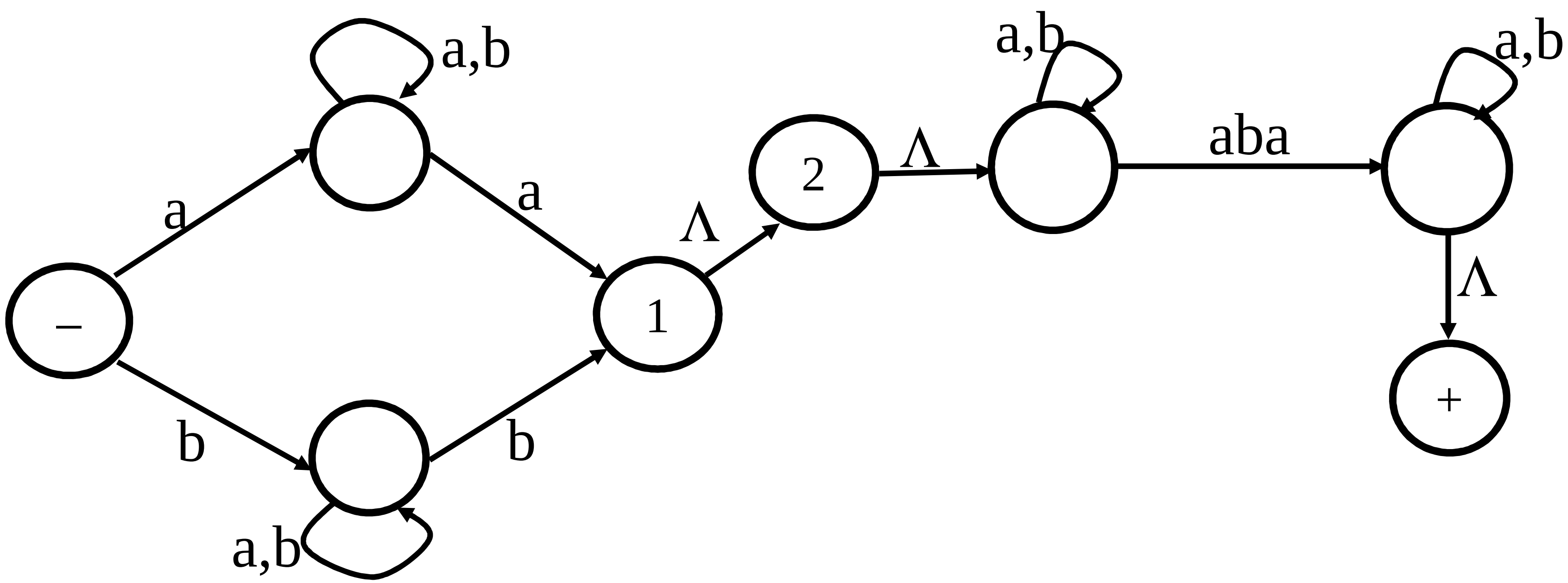
Words that contain aba.

(a+b)*aba(a+b)*

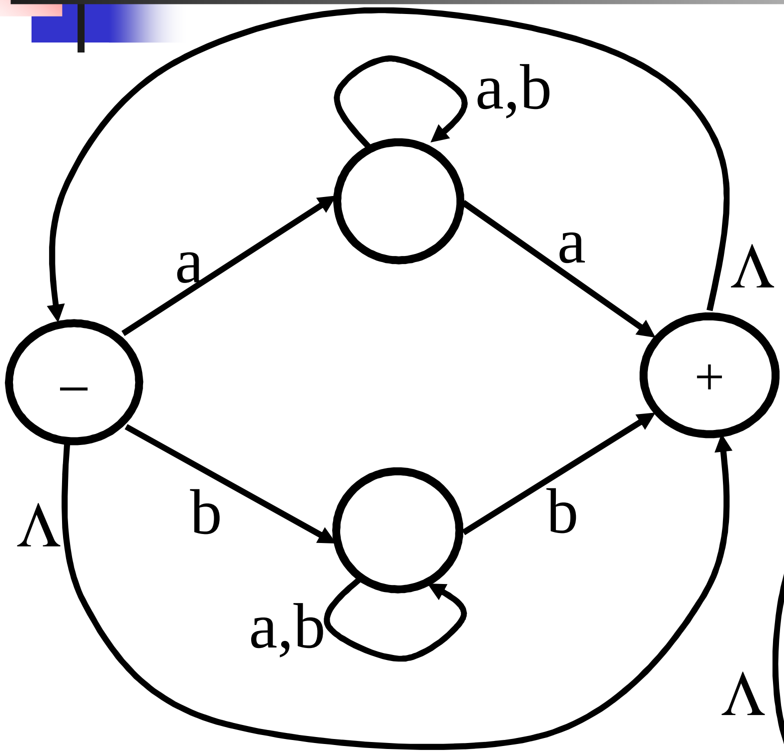




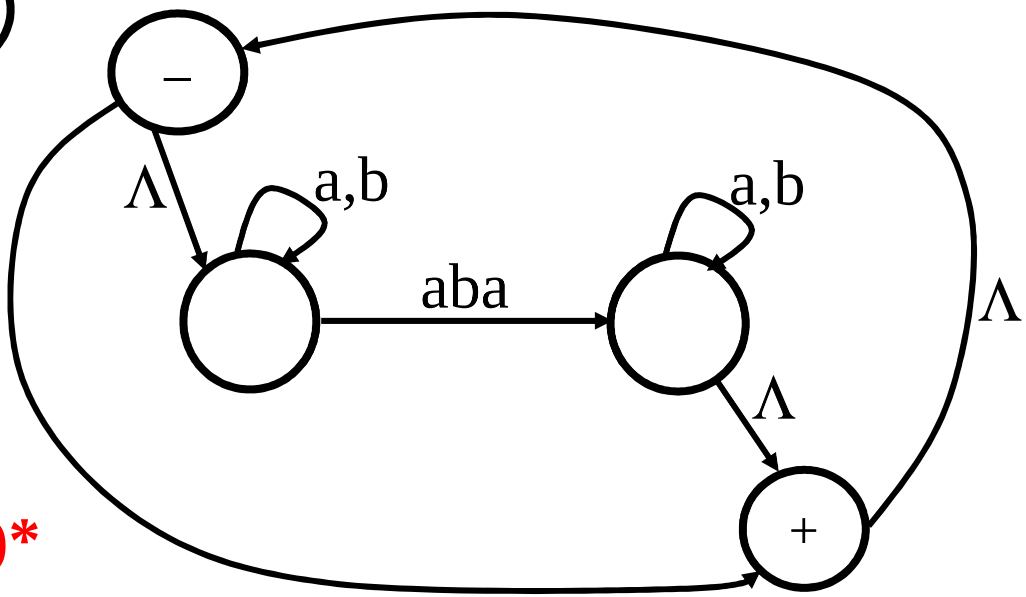
(a+b)*aba (a+b)* + a(a+b)*a + b(a+b)*b



(a(a+b)*a + b(a+b)*b)((a+b)*aba(a+b)*)



(a(a+b)*a + b(a+b)*b)*



((a+b)*aba(a+b)*)*

Definition: If L is a language over the alphabet Σ, the complement of L, written L' is the language of all words on Σ that are not words in L. ($L' = \Sigma^* - L$).

Example:

S={a,b}

L= all words containing aa.

L'=?

$b^*(abb^*)^*(a+\Lambda)$

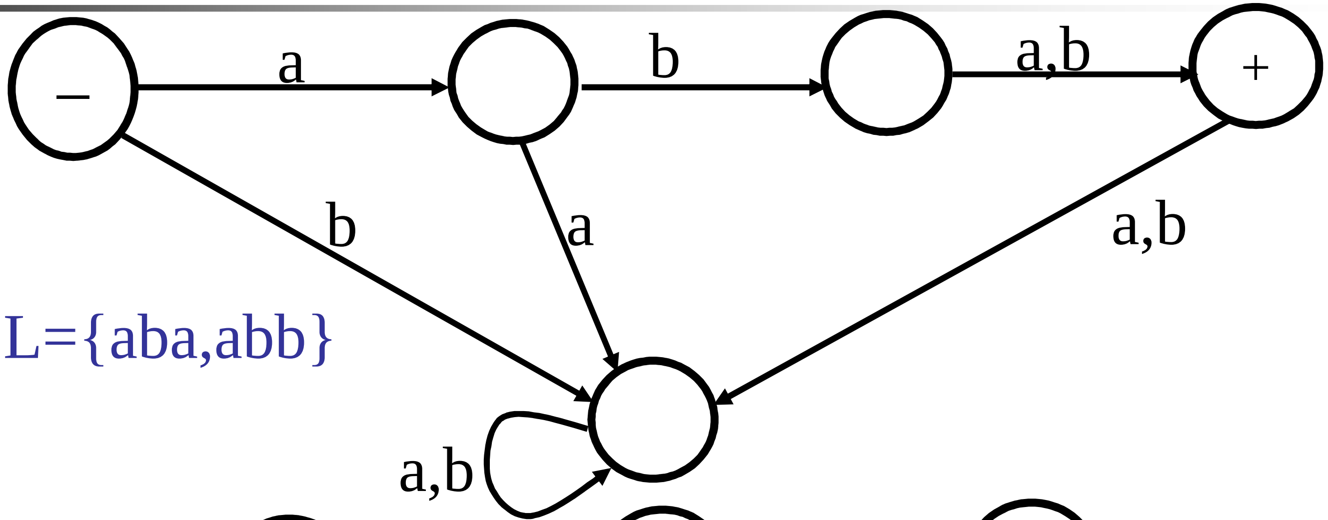
Remark: (L')' = L.

Theorem: If L is a regular language, then L’ is a regular language.

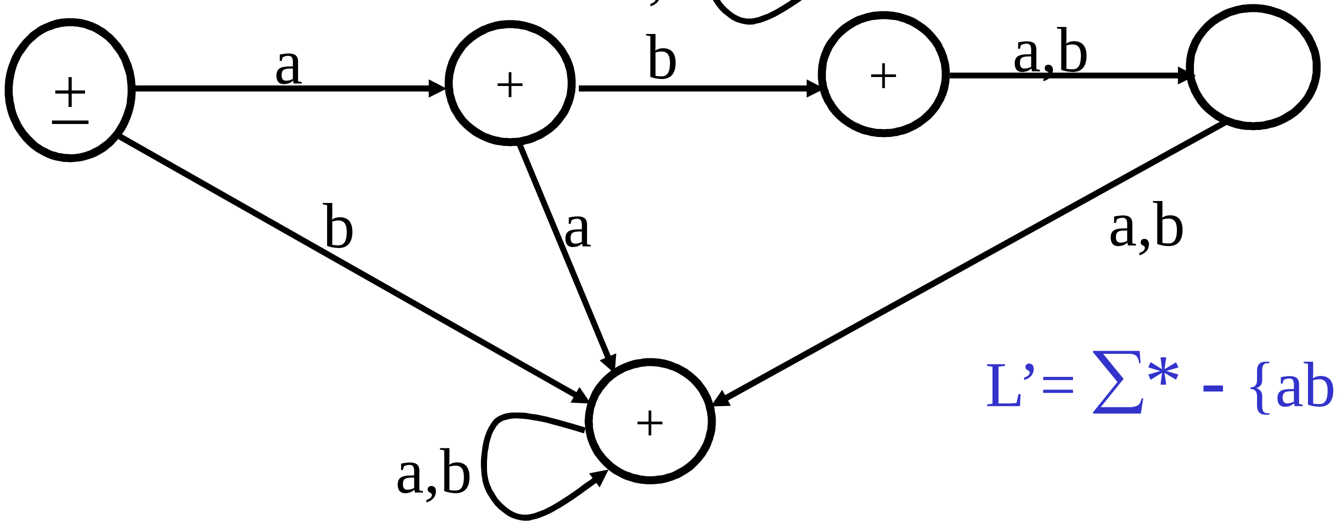
Proof: There exists a finite automaton that accepts L (by Kleene’s theorem). All words accepted by this FA end in a final state. All words that are not accepted end in a state that is not a final state.

We reverse the final status of each state: all final states become non-final states, and all non-final states become final states.

The new finite automaton accepts exactly those words that are not in L. By Kleene’s theorem, L’ is regular.



Example: L={aba,abb}



L’= ∑* - {aba,abb}

Theorem: Let L_1 and L_2 be two regular languages. Then $L_1 \cap L_2$ is a regular language.

Proof:

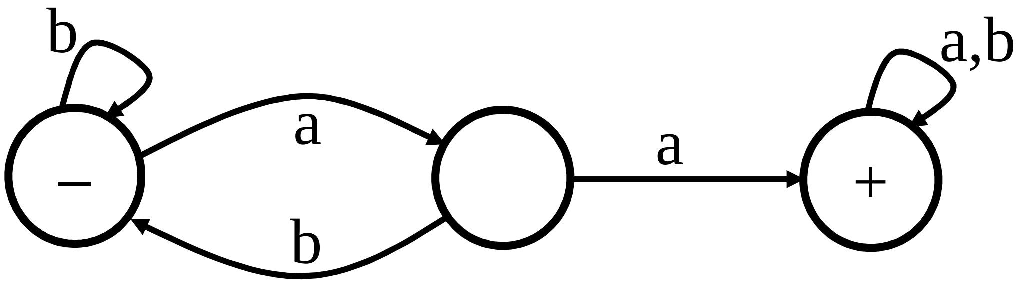
$L_1 \cap L_2 = (L_1' + L_2')'$.

If L_1 and L_2 are regular, then L_1' et L_2' are also regular.
If L_1' and L_2' are regular, then $L_1' + L_2'$ is regular.
If $L_1' + L_2'$ is regular, then $(L_1' + L_2')'$ is regular.

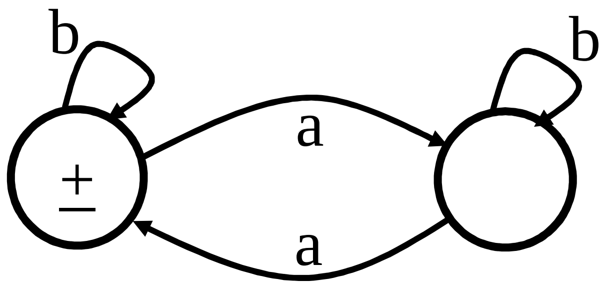
Thus, $L_1 \cap L_2$ is a regular language.

Example 1: $L_1 \cap L_2 = (L_1' + L_2')'$

L_1: words with double a

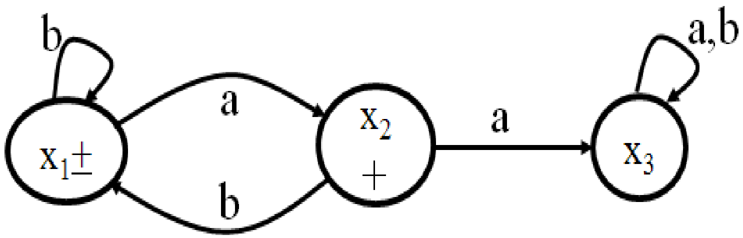


L_2: words with an even number of a's

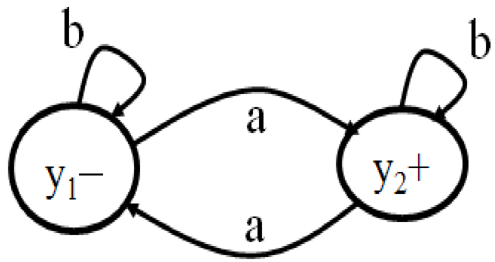


$aaa \in L_1 \quad aba \in L_2$

L'$_1$



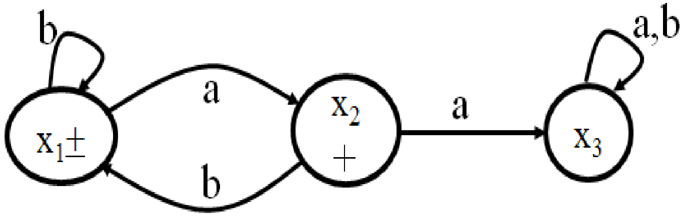
L$_2$'



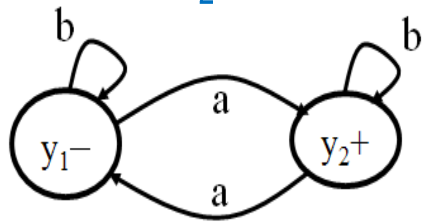
L'$_1$ + L$_2$'

	State	a	b
±Z1	X1orY1	X2orY2	X1orY1
+Z2	X1orY2	X2orY1	X1orY2
+Z3	X2orY1	X3orY2	X1orY1
+Z4	X2orY2	X3orY1	X1orY2
Z5	X3orY1	X3orY2	X3orY1
+Z6	X3orY2	X3orY1	X3orY2

L'_1



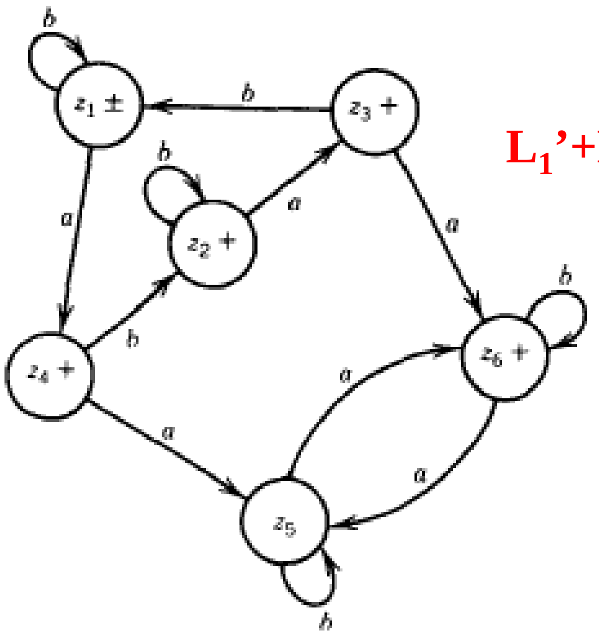
L_2'



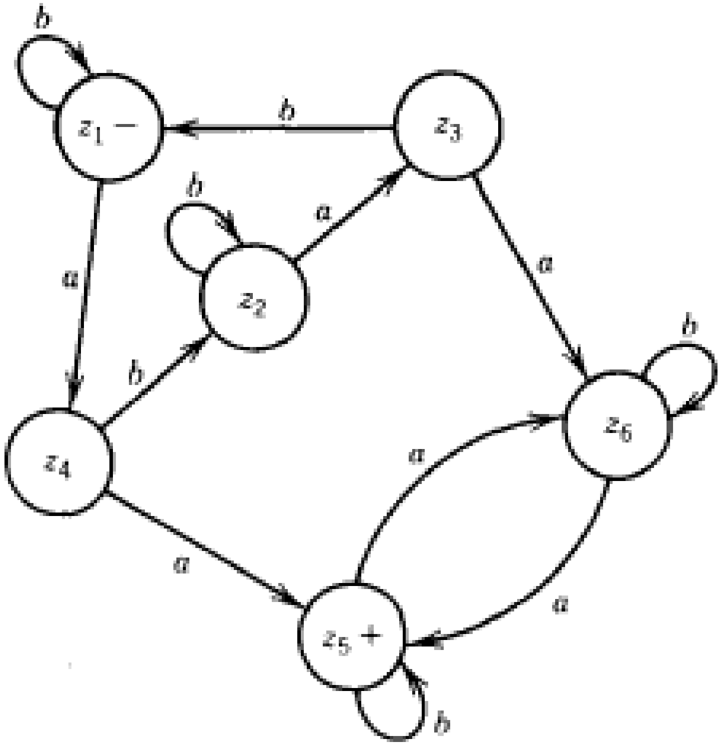
	State	a	b
±Z1	X1orY1	X2orY2	X1orY1
+Z2	X1orY2	X2orY1	X1orY2
+Z3	X2orY1	X3orY2	X1orY1
+Z4	X2orY2	X3orY1	X1orY2
Z5	X3orY1	X3orY2	X3orY1
+Z6	X3orY2	X3orY1	X3orY2

$L'_1 + L_2'$

$(L_1'+L_2')'=L_1\cap L_2$



$L_1'+L_2'$



	a	b
±Z1	Z4	Z1
+Z2	Z3	Z2
+Z3	Z6	Z1
+Z4	Z5	Z2
Z5	Z6	Z5
+Z6	Z5	Z6

Example 2:

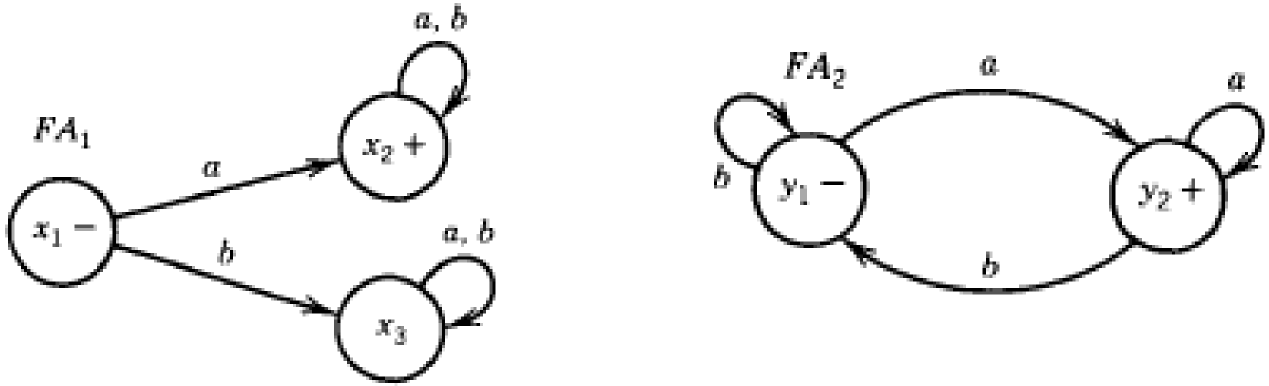
L_1 = all words that begin with an a

L_2 = all words that end with an a

$\mathbf{r}_1 = \mathbf{a}\ (\mathbf{a} + \mathbf{b})^*$

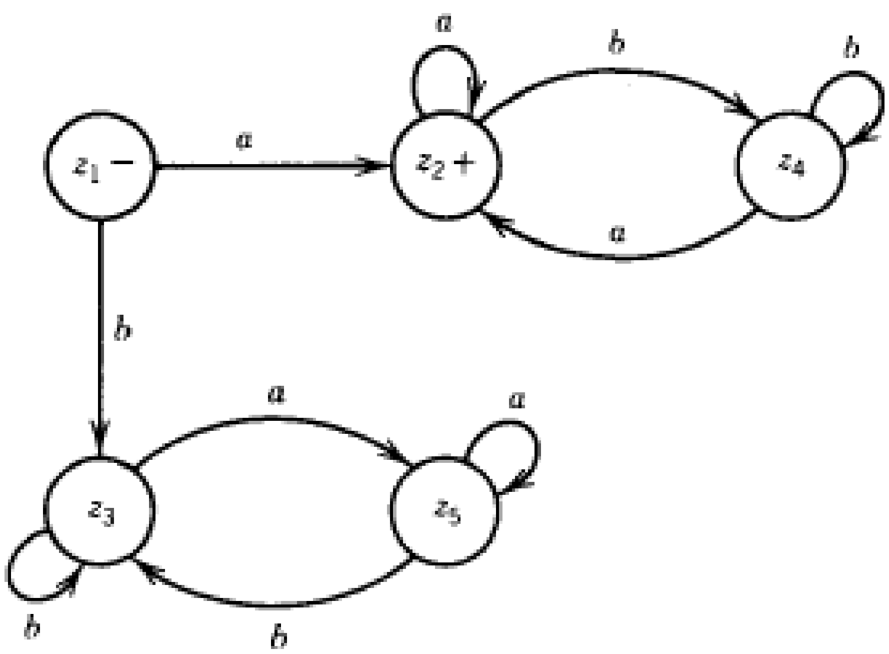
$\mathbf{r}_2 = (\mathbf{a} + \mathbf{b})^*\ \mathbf{a}$

$L_1 \cap L_2$ = all words that begin and end with the letter a



Example 2:

State	Read a	Read b	New names
x_1 or y_1	x_2 or y_2	x_3 or y_1	$-z_1$
x_2 or y_2	x_2 or y_2	x_2 or y_1	$+z_2$
x_3 or y_1	x_3 or y_2	x_3 or y_1	z_3
x_2 or y_1	x_2 or y_2	x_2 or y_1	z_4
x_3 or y_2	x_3 or y_2	x_3 or y_1	z_5



Theorem: Let L_1 and L_2 be two regular languages. Then $L_1 \cap L_2$ is also a regular language.

Proof 2: By constructive algorithm. There exist finite automata that accept L_1 and L_2 (by Kleene's theorem). Recall the constructive algorithm for building a finite automaton for L_1+L_2 from the finite automata for L_1 and L_2. We build the same finite automaton except for the final states. Each state in the new automaton represents a pair of states, one from each of the original finite automata, $\{x_i, y_j\}$. A state in the new machine is final if both of these states are final states in the original machines.

What is difference in building

- Union machine $L1 \cup L2$
- Intersection machine $L1 \cap L2$
- Machine for $L1' \cap L2$
- Machine for $L1 \cap L2'$
- Machine for $(L1' \cap L2) \cup (L1 \cap L2')$
- (what is the last one for?)
- Build FA3 using xi OR yj states