Generalized Transition Graphs

A generalized transition graph (GTG) is a collection of three things

1) Finite number of states, at least one of which is start state and some (maybe none) final states.
2) Finite set of input letters (Σ) from which input strings are formed.
3) Directed edges connecting some pair of states labeled with regular expression.

It may be noted that in GTG, the labels of transition edges are corresponding regular expressions

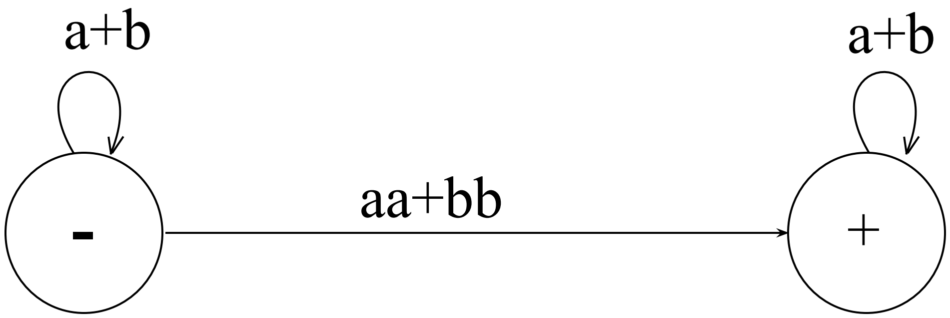
Example

Consider the language L of strings, defined over $\Sigma=\{a,b\}$, containing **double a or double b**. The language L can be expressed by the following regular expression

$$(a+b)^* (aa + bb) (a+b)^*$$

The language L may be accepted by the following GTG.

Example continued ...



Example

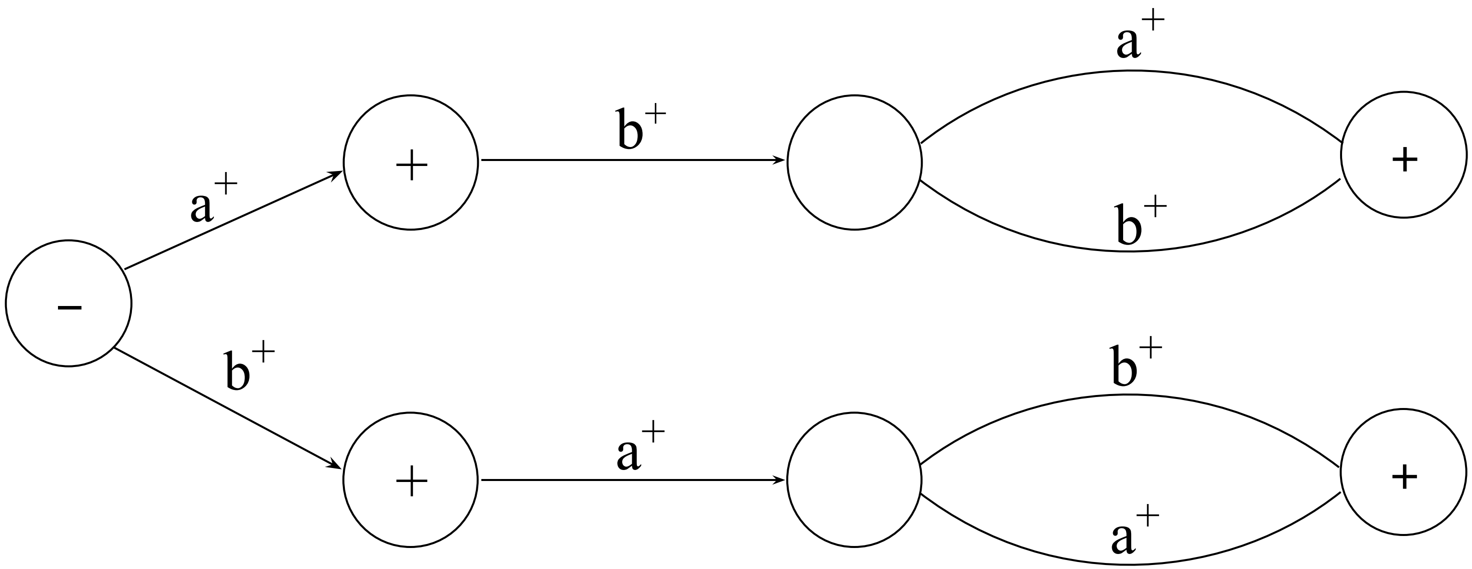
- Consider the Language L of strings, defined over $\Sigma = \{a, b\}$, **beginning with and ending in same letters.**

  The language L may be expressed by the following regular expression

  $$(a+b)+ \quad a(a + b)^*a + b(a + b)^*b$$

  This language may be accepted by the following GTG

Example



Example

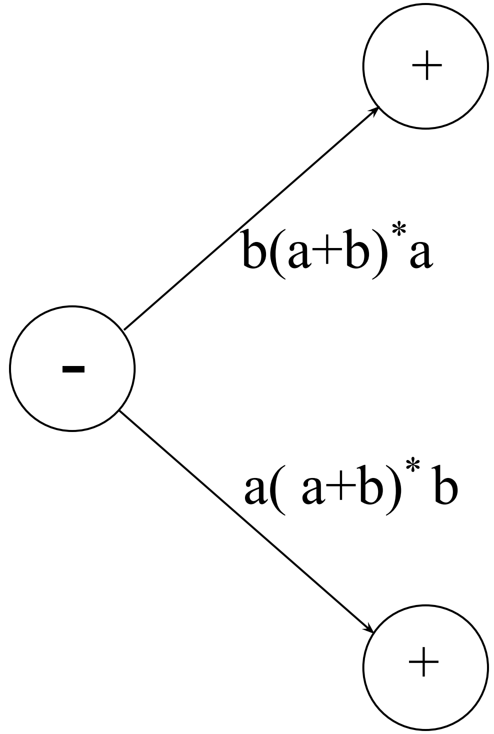
Consider the language L of strings of, defined over $\Sigma = \{a, b\}$, **beginning and ending in different letters**.

The language L may be expressed by RE

$$a(a + b)^*b + b(a + b)^*a$$

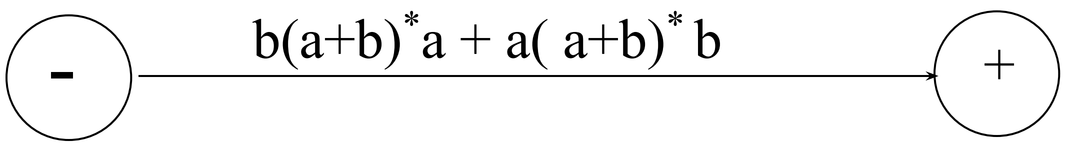
The language L may be accepted by the following GTG

Example Continued ...



The language L may be accepted by the following GTG as well

Example Continued ...



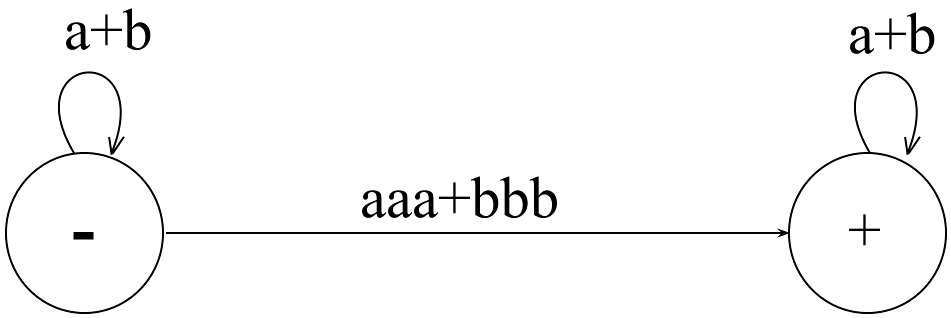
Example

Consider the language L of strings, defined over $\Sigma=\{a, b\}$, **having triple a or triple b.** The language L may be expressed by RE

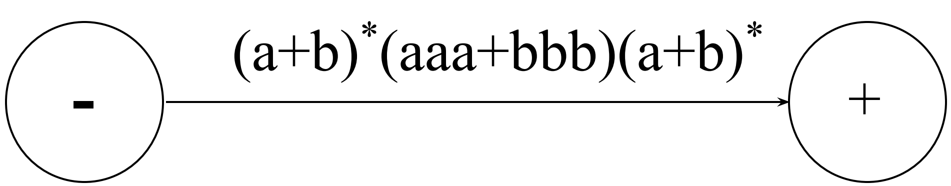
$$(a+b)^* (aaa + bbb) (a+b)^*$$

This language may be accepted by the following GTG

Example Continued ...



OR



Nondeterminism

- TGs and GTGs provide certain relaxations *i.e.* there may exist more than one path for a certain string or there may not be any path for a certain string, this property creates **nondeterminism** and it can also help in differentiating TGs or GTGs from FAs. Hence an FA is also called a Deterministic Finite Automaton (DFA).

Kleene's Theorem

- **If** a language can be expressed by

1. FA or
2. TG or
3. RE

then it can also be expressed by other two as well.

It may be noted that the theorem is proved, proving the following three parts

Kleene’s Theorem continued …

Kleene’s Theorem Part I

If a language can be accepted by an FA then it can be accepted by a TG as well.

Kleene’s Theorem Part II

If a language can be accepted by a TG then it can be expressed by an RE as well.

Kleene’s Theorem Part III

If a language can be expressed by a RE then it can be accepted by an FA as well.

Kleene’s Theorem continued …

Proof(Kleene’s Theorem Part I)

Since every FA can be considered to be a TG as well, therefore there is nothing to prove.

Kleene’s Theorem continued …

<u>Proof(Kleene’s Theorem Part II)</u>

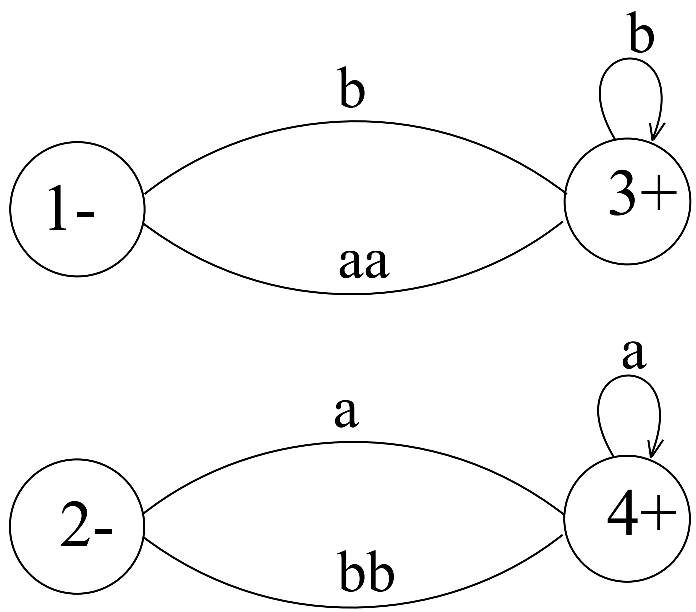
To prove part II of the theorem, an algorithm consisting of different steps, is explained showing how a RE can be obtained corresponding to the given TG. For this purpose the notion of TG is changed to that of GTG *i.e.* the labels of transitions are corresponding REs.

Kleene's Theorem part II continued …

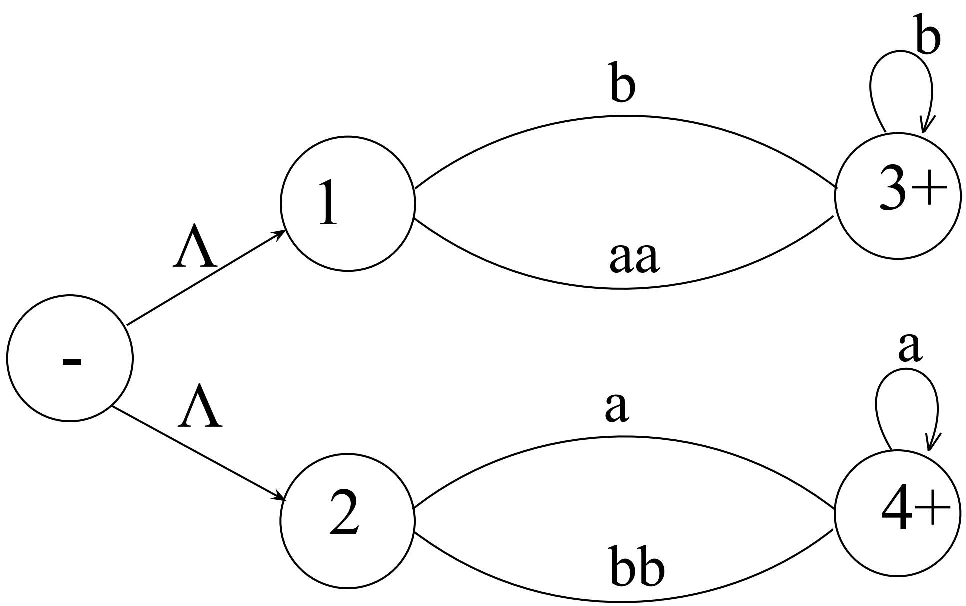
Basically this algorithm converts the given TG to GTG with one initial state along with a single loop, or one initial state connected with one final state by a single transition edge. The label of the loop or the transition edge will be the required RE.

Step 1 If a TG has more than one start states, then introduce a new start state connecting the new state to the old start states by the transitions labeled by Λ and make the old start states the non-start states. This step can be shown by the following example

Example



Example Continued ...



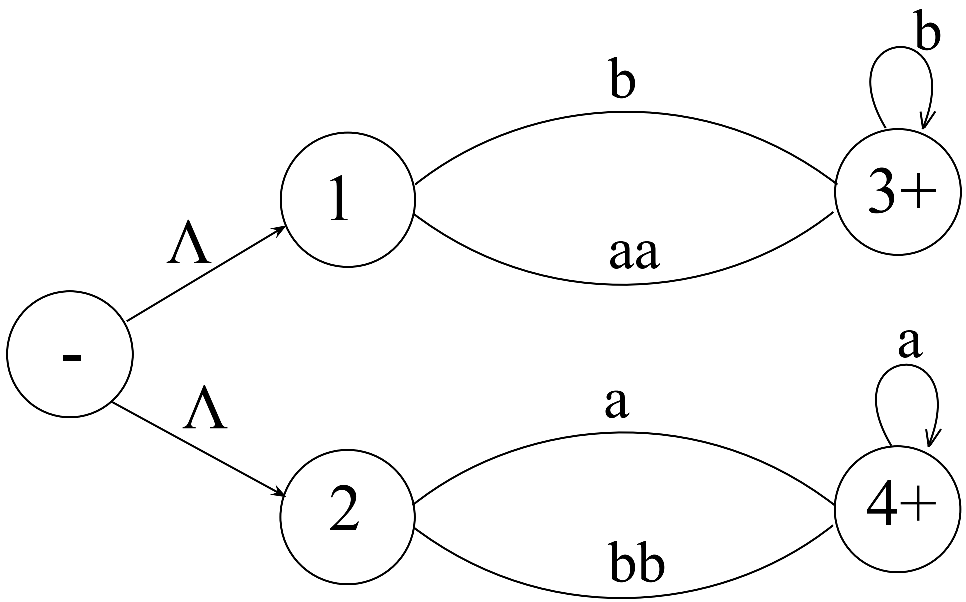
Kleene’s Theorem part II continued …

Step 2:

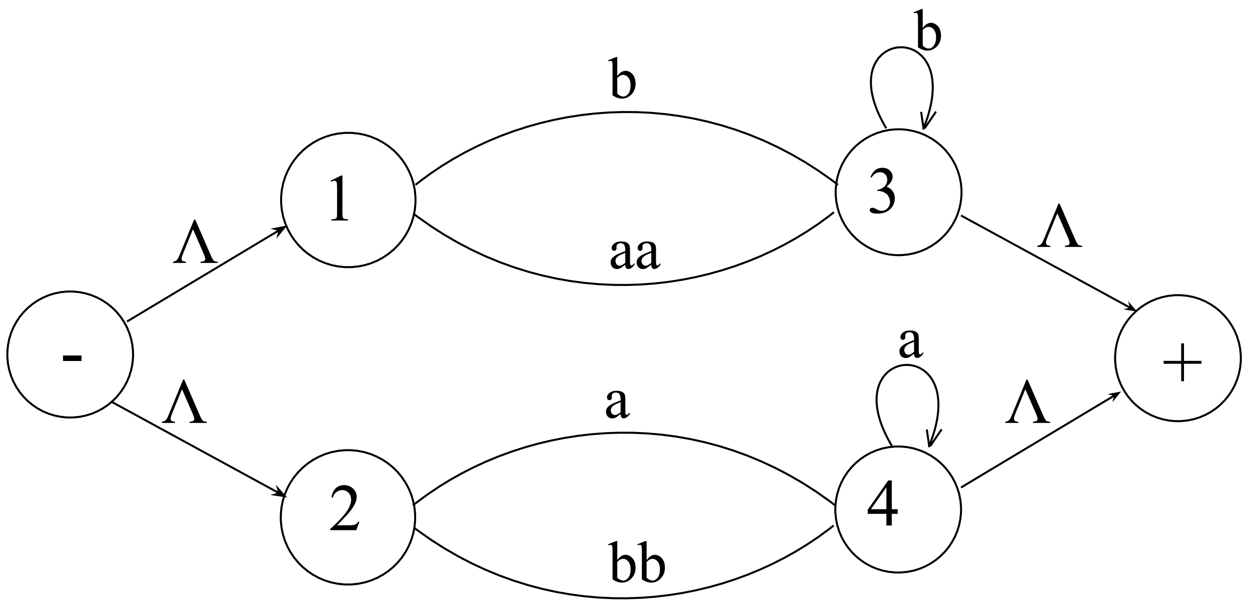
If a TG has more than one final states, then introduce a new final state, connecting the old final states to the new final state by the transitions labeled by Λ.

This step can be shown by the previous example of TG, where the step 1 has already been processed

Example



Example continued …



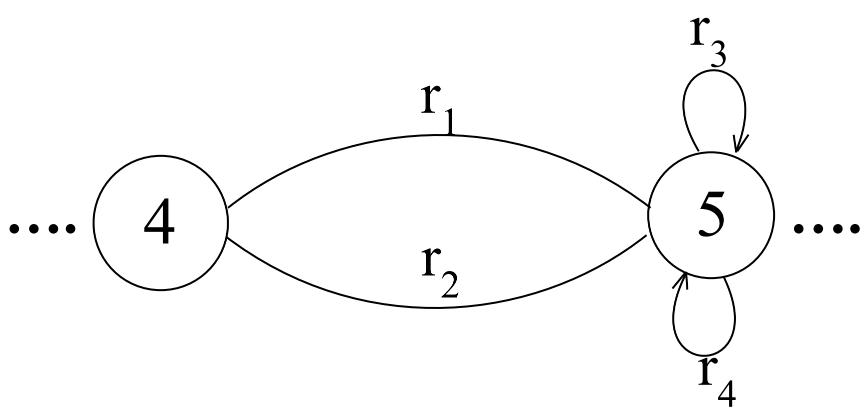
Kleene's Theorem part II continued ...

Step 3:

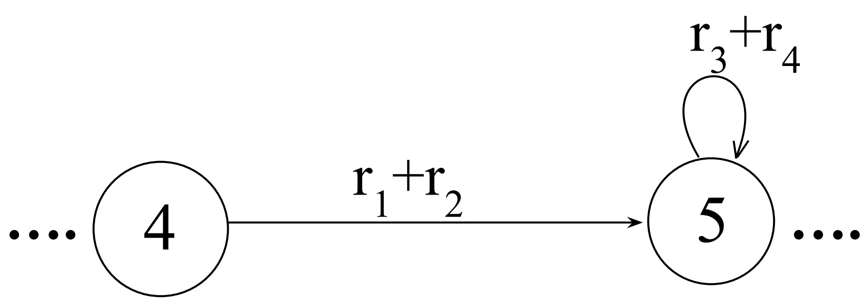
If a state has two (more than one) incoming transition edges labeled by the corresponding REs, from the same state (including the possibility of loops at a state), then replace all these transition edges with a single transition edge labeled by the sum of corresponding REs.

This step can be shown by a part of TG in the following example

Example

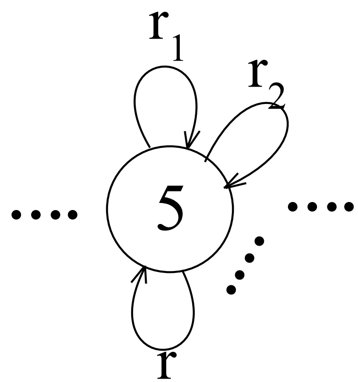


The above TG can be reduced to

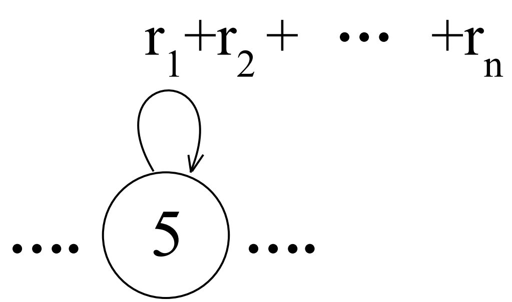


Note

- The step 3 can be generalized to any finite number of transitions as shown below



The above TG can be reduced to

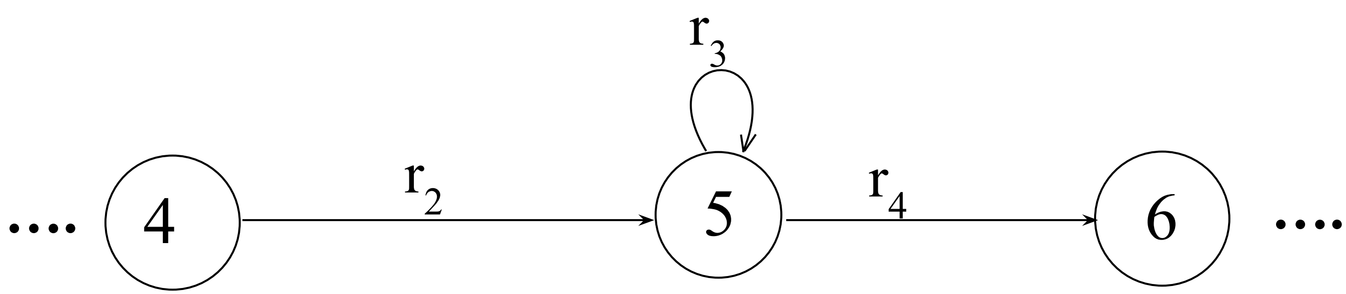


Kleene’s Theorem part II continued …

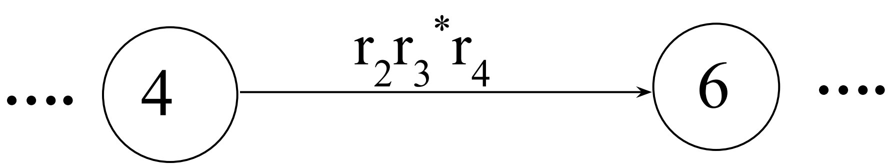
<u>Step 4 (bypass and state elimination)</u>

If three states in a TG, are connected in sequence then eliminate the middle state and connect the first state with the third by a single transition (include the possibility of circuit as well) labeled by the RE which is the concatenation of corresponding two REs in the existing sequence. This step can be shown by a part of TG in the following example

Example



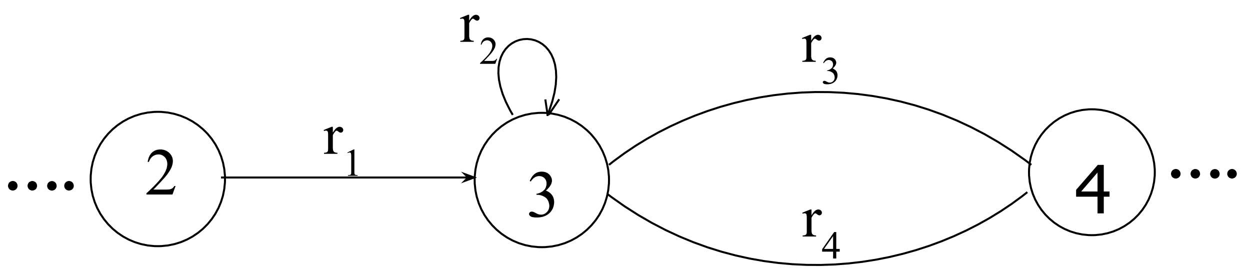
To eliminate state 5 the above can be reduced to



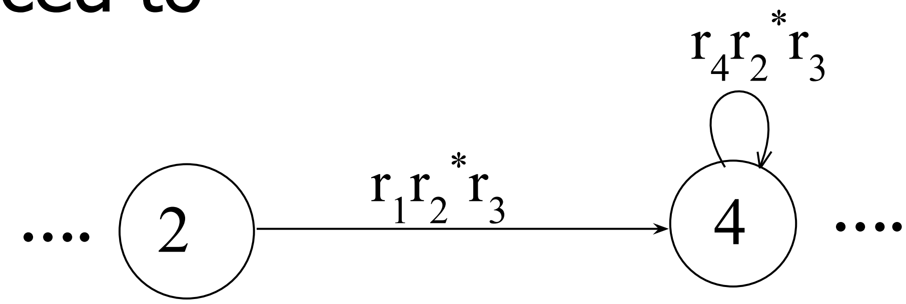
Consider the following example containing a circuit

Example

Consider the part of a TG, containing a circuit at a state, as shown below

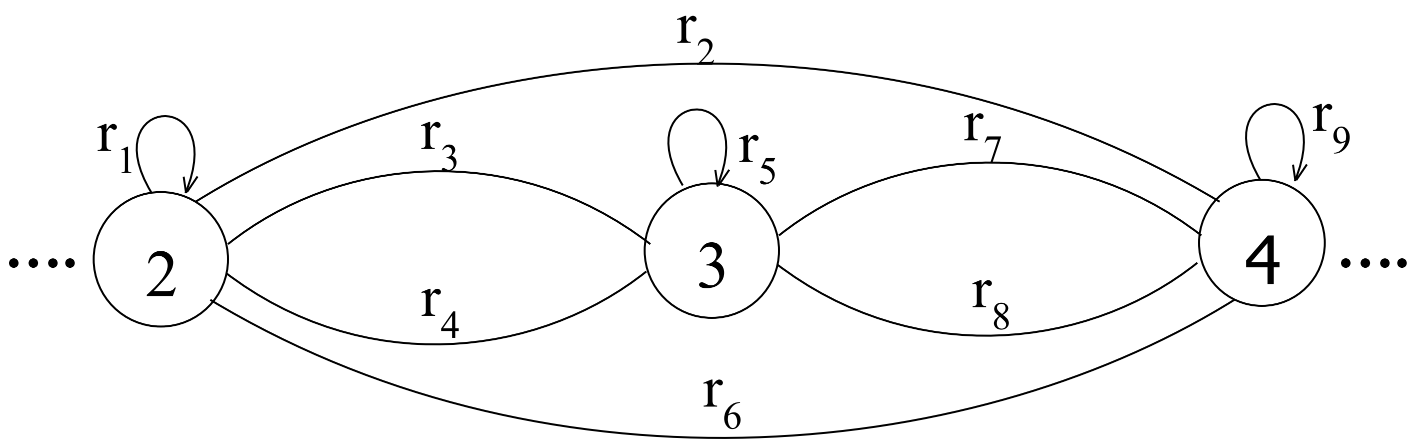


To eliminate state 3 the above TG can be reduced to



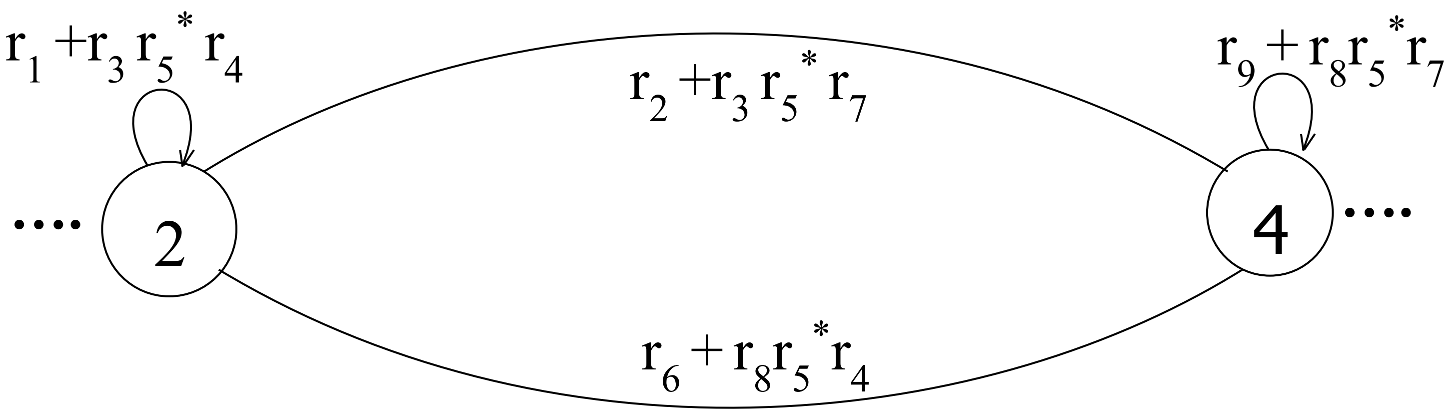
Example

Consider a part of the following TG



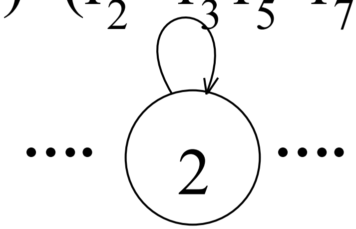
To eliminate state 3 the above TG can be reduced to

Example continued …



To eliminate state 4 the above TG can be reduced to

$$(r_1 + r_3 r_5^* r_4) + (r_2 + r_3 r_5^* r_7)(r_9 + r_8 r_5^* r_7)^*(r_6 + r_8 r_5^* r_4)$$

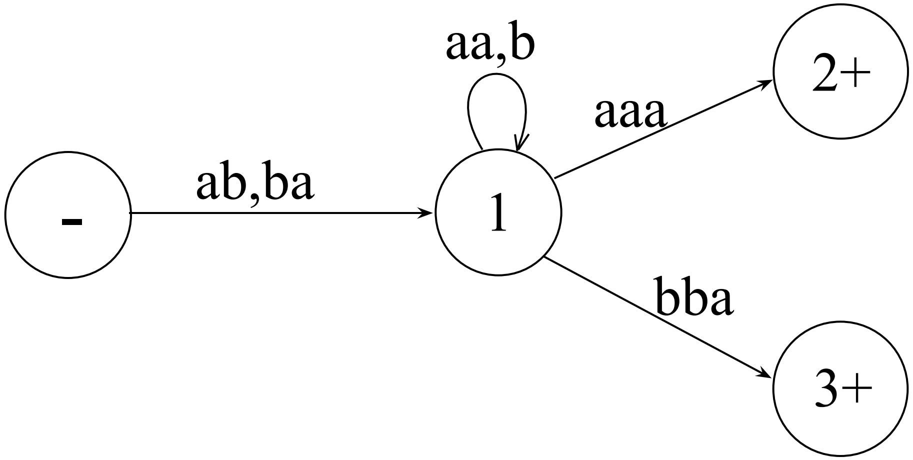


Note

- It is to be noted that to determine the RE corresponding to a certain TG, four steps have been discussed. This process can be explained by the following particular examples of TGs

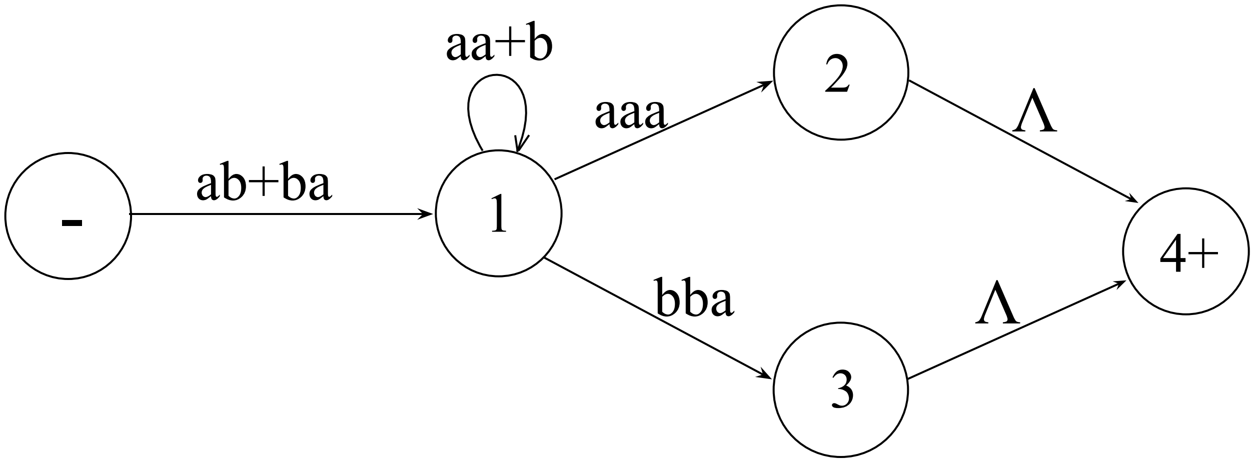
Example

- Consider the following TG

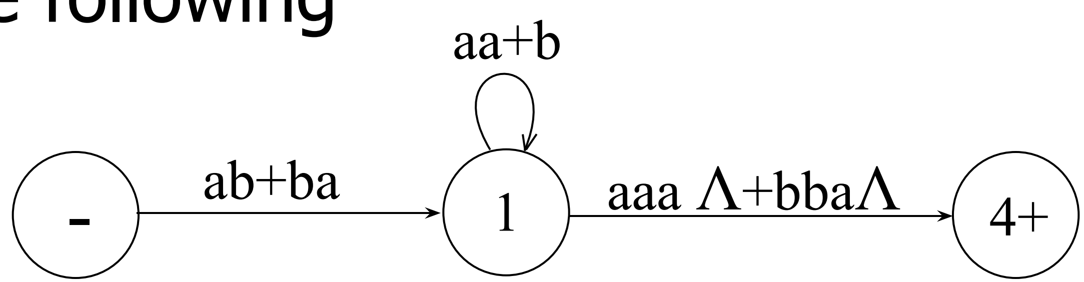


To have single final state, the above TG can be reduced to the following

Example continued …



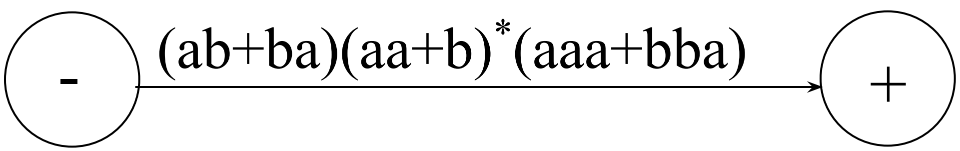
To eliminate states 2 and 3, the above TG can be reduced to the following



The above TG can be reduced to the following

Example continued …

To eliminate state 1 the above TG can be reduced to the following

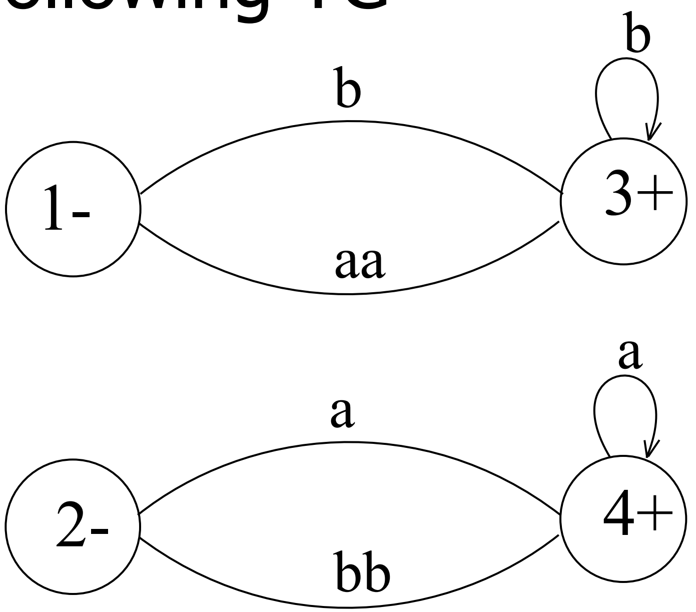


Hence the required RE is

$(ab+ba)(aa+b)^*(aaa+bba)$

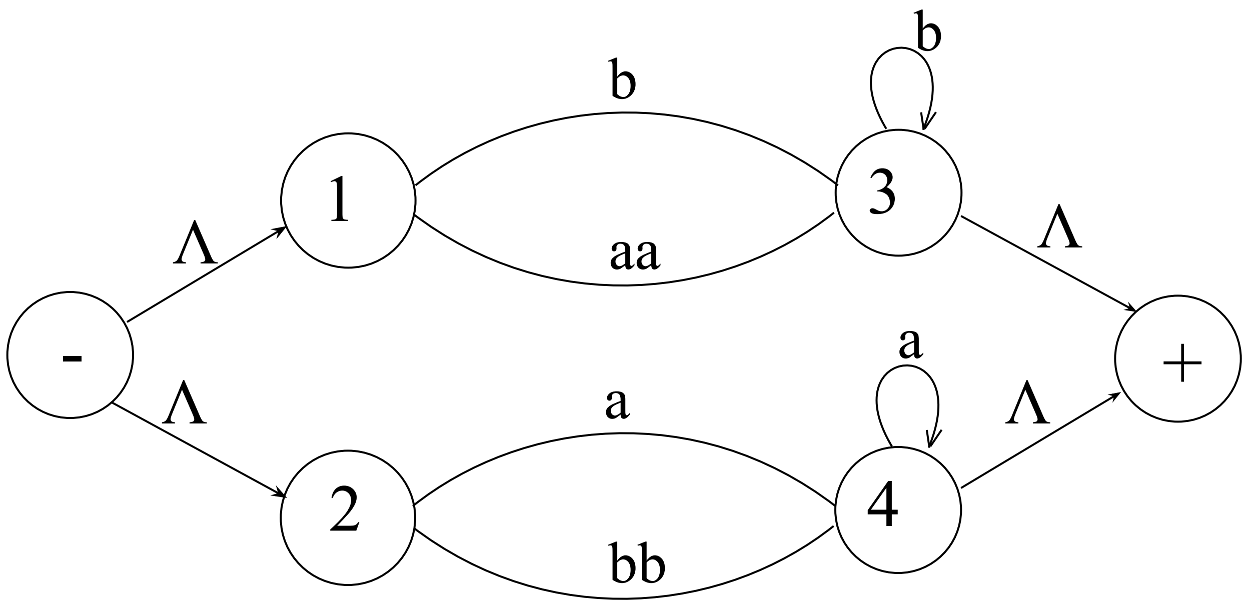
Example

- Consider the following TG



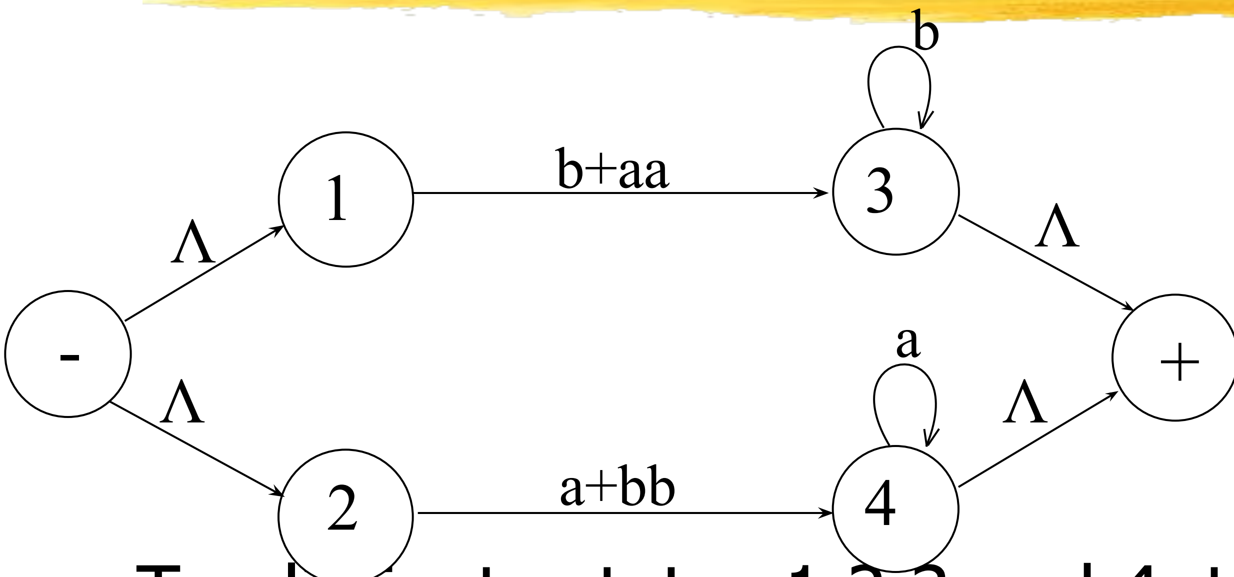
To have single initial and single final state the above TG can be reduced to the following

Example continued …

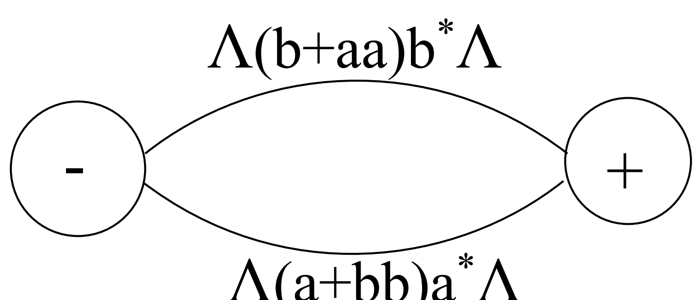


To obtain single transition edge between 1 and 3; 2 and 4, the above can be reduced to the following

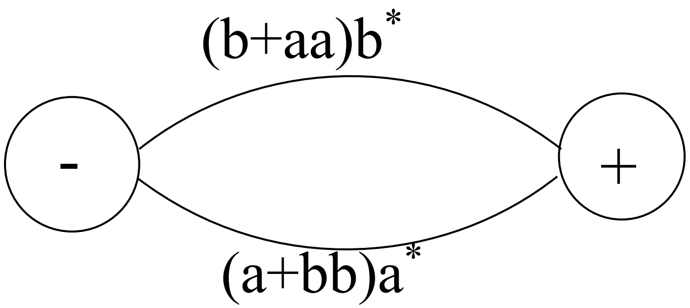
Example continued ...



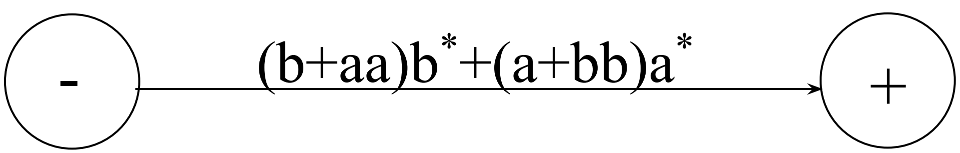
To eliminate states 1,2,3 and 4, the above TG can be reduced to the following TG



Example continued …



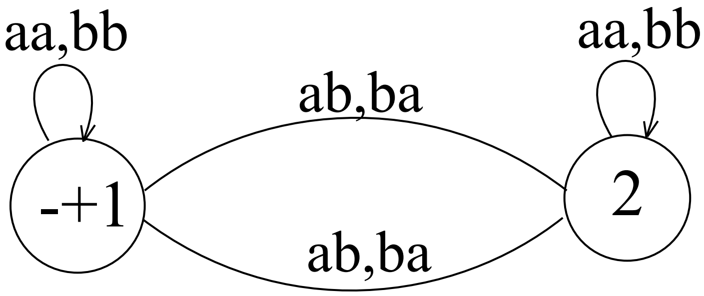
To connect the initial state with the final state by single transition edge, the above TG can be reduced to the following



Hence the required RE is $(b+aa)b^*+(a+bb)a^*$

Example

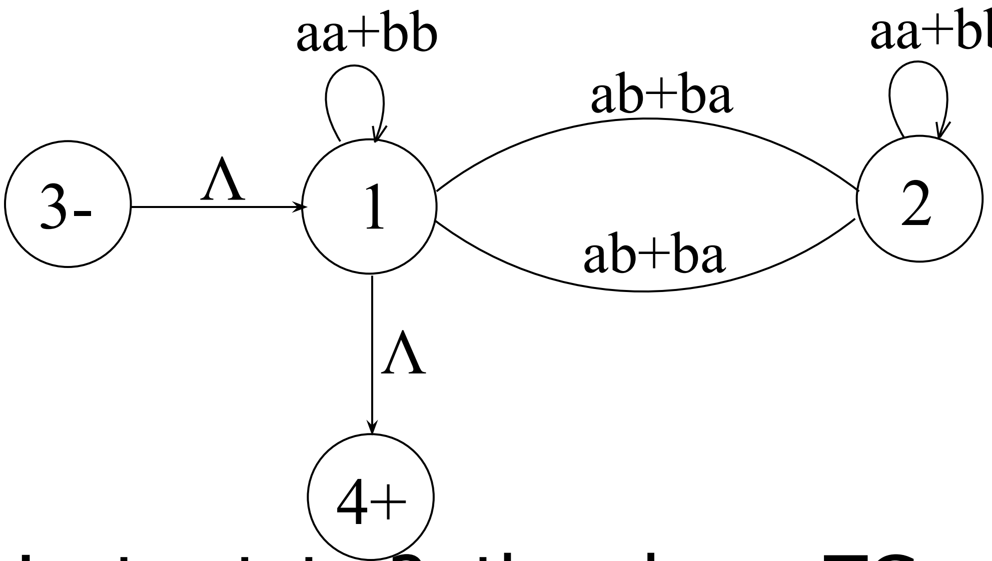
Consider the following TG, accepting EVEN-EVEN language



Example continued …

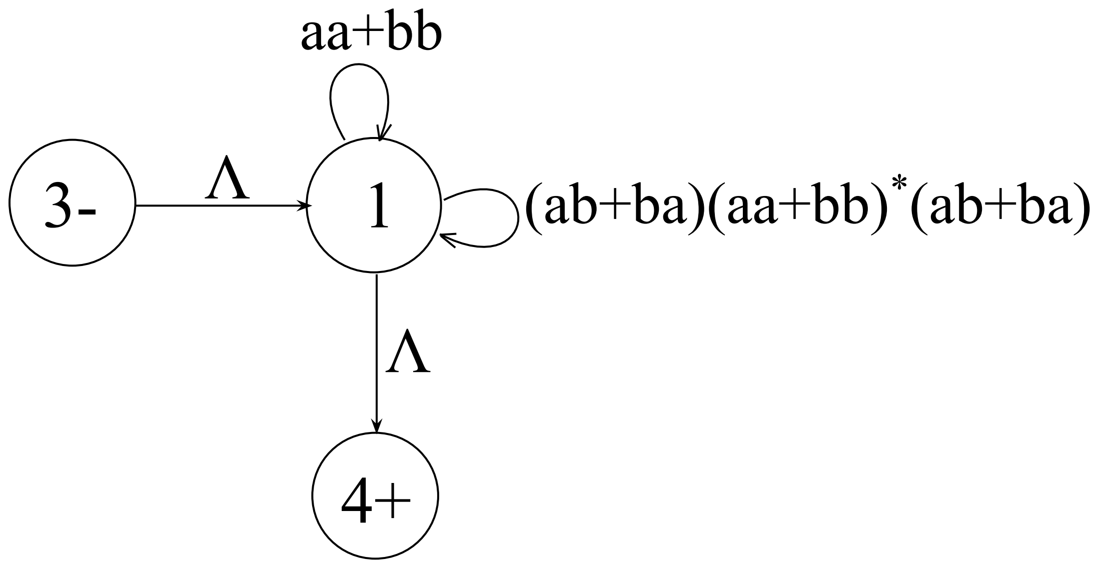
It is to be noted that since the initial state of this TG is final as well and there is no other final state, so to obtain a TG with single initial and single final state, an additional initial and a final state are introduced as shown in the following TG

Example continued …



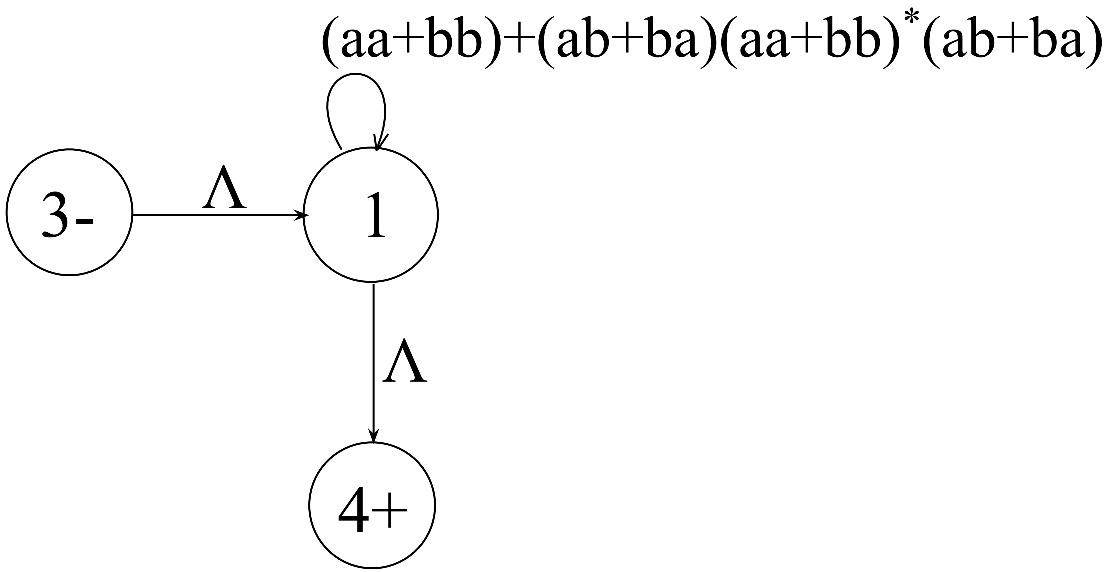
- To eliminate state 2, the above TG may be reduced to the following

Example continued …



To have single loop at state 1, the above TG may be reduced to the following

Example continued …



To eliminate state 1, the above TG may be reduced to the following

Example continued …

$$3- \xrightarrow{\Lambda(aa+bb+(ab+ba)(aa+bb)^*(ab+ba))^*\Lambda} 4+$$

Hence the required RE is

$(aa+bb+(ab+ba)(aa+bb)^*(ab+ba))^*$

Kleene's Theorem Part III

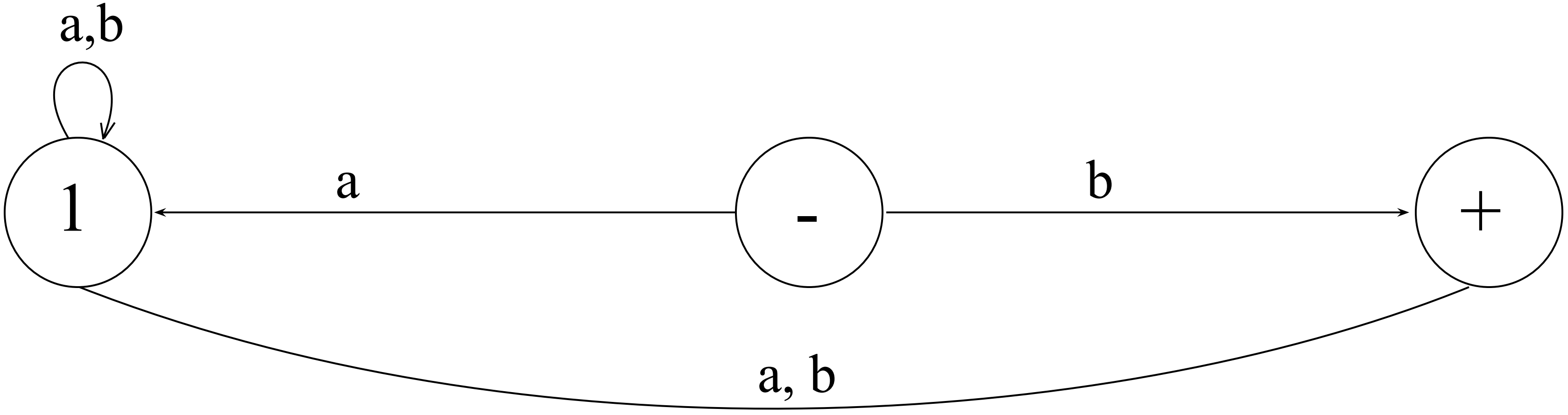
Statement:

If the language can be expressed by a RE then there exists an FA accepting the language.

A) As the regular expression is obtained applying addition, concatenation and closure on the letters of an alphabet and the Null string, so while building the RE, sometimes, the corresponding FA may be built easily, as shown in the following examples

Example

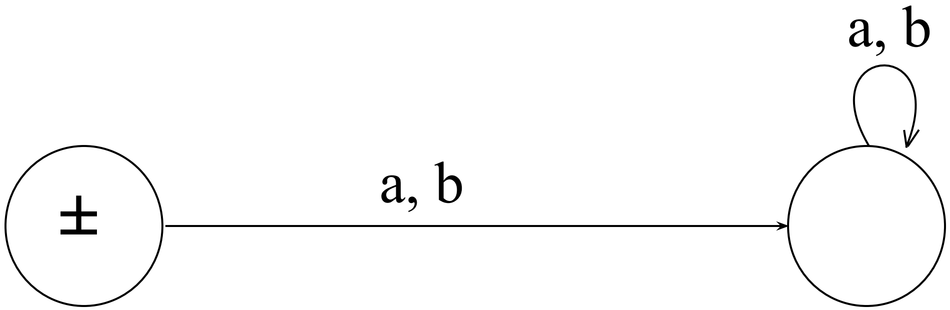
- Consider the language, defined over Σ={a,b}, **consisting of only b**, then this language may be accepted by the following FA



which shows that this FA helps in building an FA accepting only one letter

Example

- Consider the language, defined over Σ={a,b}, **consisting of only □**, then this language may be accepted by the following FA



Kleene's Theorem Part III Continued …

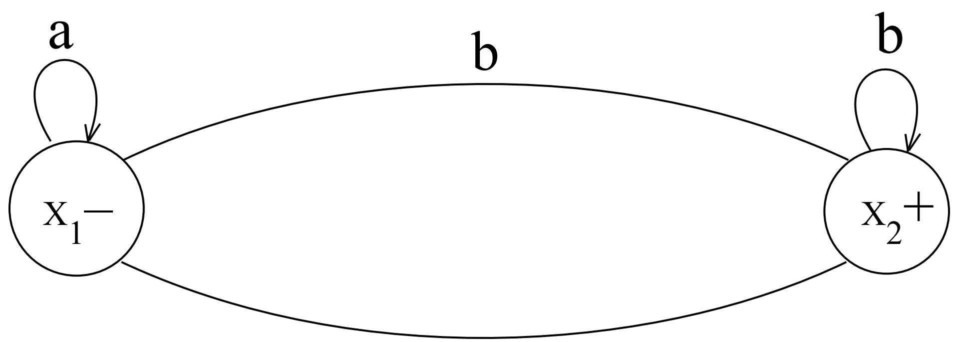
B) As, if r_1 and r_2 are regular expressions then their sum, concatenation and closure are also regular expressions, so an FA can be built for any regular expression if the methods can be developed for building the FAs corresponding to the sum, concatenation and closure of the regular expressions along with their FAs. These three methods are explained in the following discussions

Kleene's Theorem Part III Continued ...

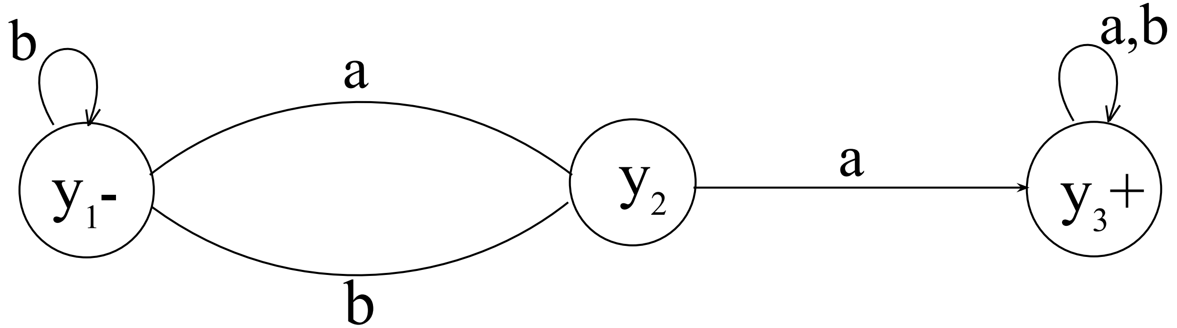
- **Method1 (Union of two FAs):** Using the FAs corresponding to r_1 and r_2 an FA can be built, corresponding to $r_1 + r_2$. This method can be developed considering the following examples

Example

Let $r_1=(a+b)^*b$ defines L_1 and the FA_1 be



and $r_2 = (a+b)^*aa(a+b)^*$ defines L_2 and FA_2 be



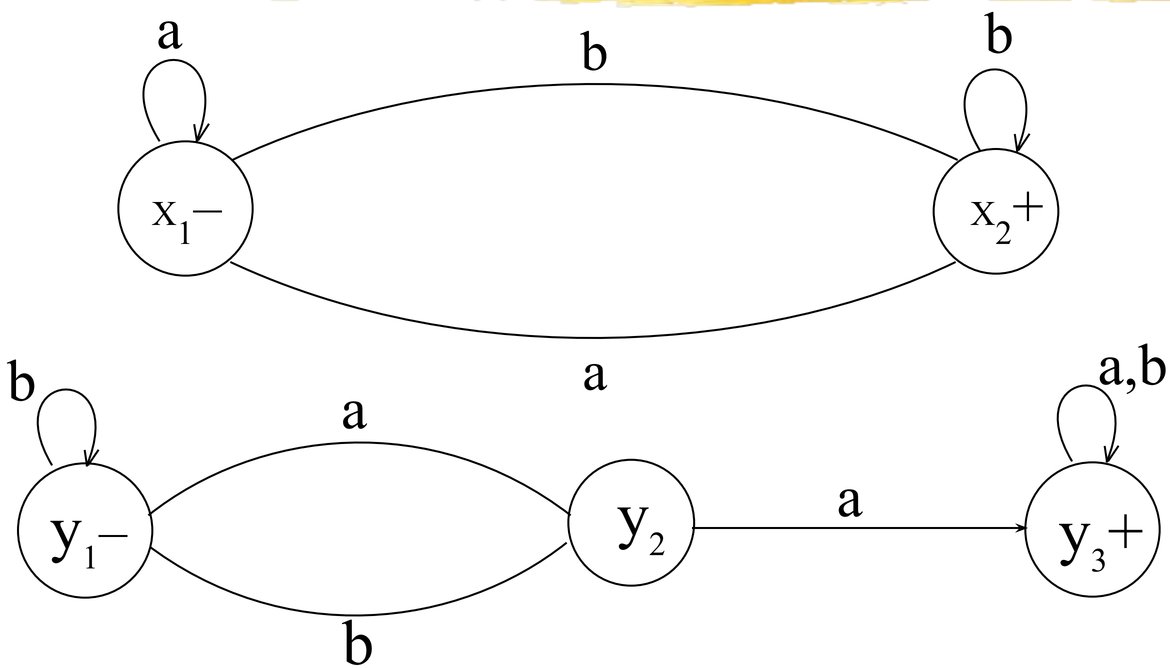
Sum of two FAs Continued …

Let FA_3 be an FA corresponding to $r_1 + r_2$, then the initial state of FA_3 must correspond to the initial state of FA_1 or the initial state of FA_2. Since the language corresponding to $r_1 + r_2$ is the union of corresponding languages L_1 and L_2, consists of the strings belonging to L_1 or L_2 or both, therefore a final state of FA_3 must correspond to a final state of FA_1 or FA_2 or both.

Sum of two FAs Continued …

Since, in general, FA_3 will be different from both FA_1 and FA_2, so the labels of the states of FA_3 may be supposed to be z_1,z_2, z_3, …, where z_1 is supposed to be the initial state. Since z_1 corresponds to the states x_1 or y_1, so there will be two transitions separately for each letter read at z_1. It will give two possibilities of states either z_1 or different from z_1. This process may be expressed in the following transition table for all possible states of FA_3.

Example continued …

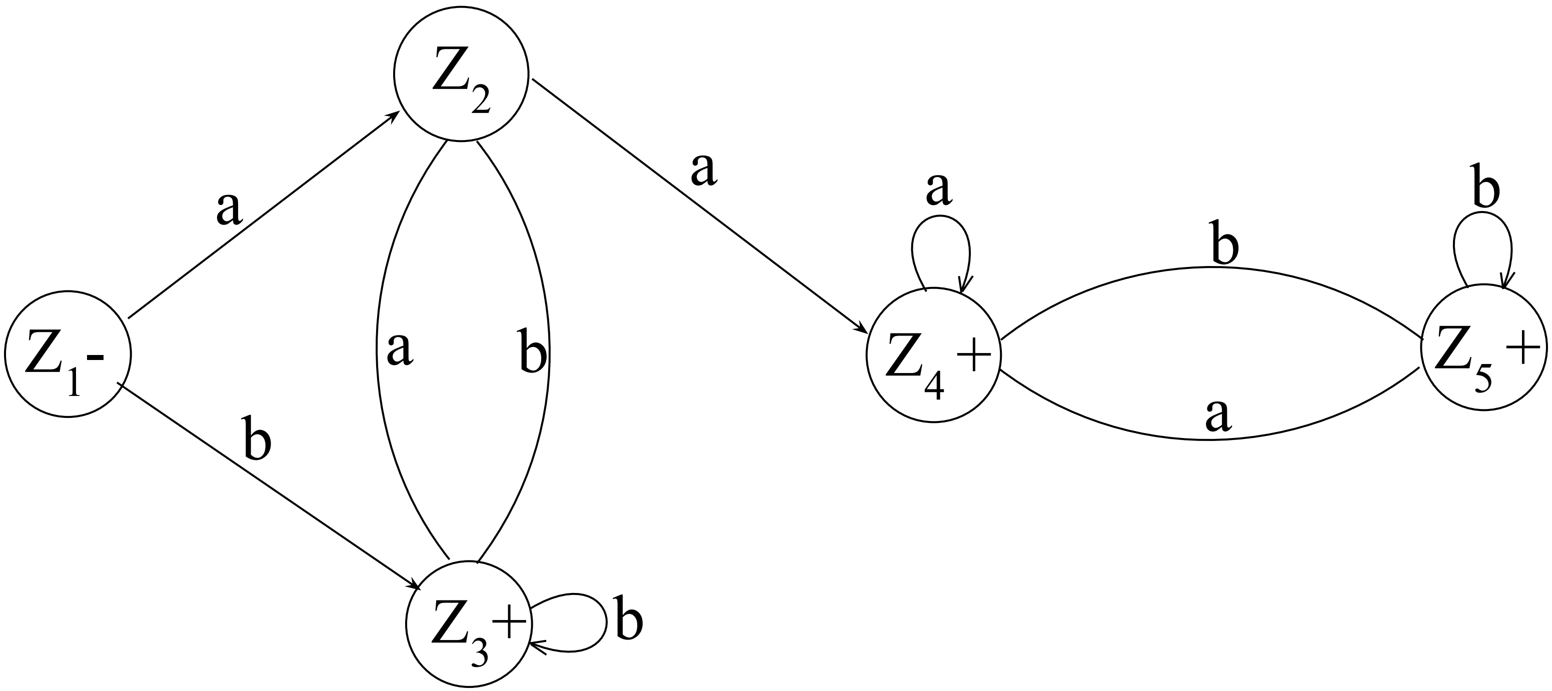


Old states	New states after reading	
	a	b
$z_1 - \square (x_1, y_1)$	$(x_1, y_2) \square z_2$	$(x_2, y_1) \square z_3$

Example continued …

Old States	New States after reading	
	a	b
$z_2 \square (x_1,y_2)$	$(x_1,y_3) \square z_4$	$(x_2,y_1) \square z_3$
$z_3+ \square (x_2,y_1)$	$(x_1,y_2) \square z_2$	$(x_2,y_1) \square z_3$
$z_4+ \square (x_1,y_3)$	$(x_1,y_3) \square z_4$	$(x_2,y_3) \square z_5$
$z_5+ \square (x_2,y_3)$	$(x_1,y_3) \square z_4$	$(x_2,y_3) \square z_5$

Example continued …



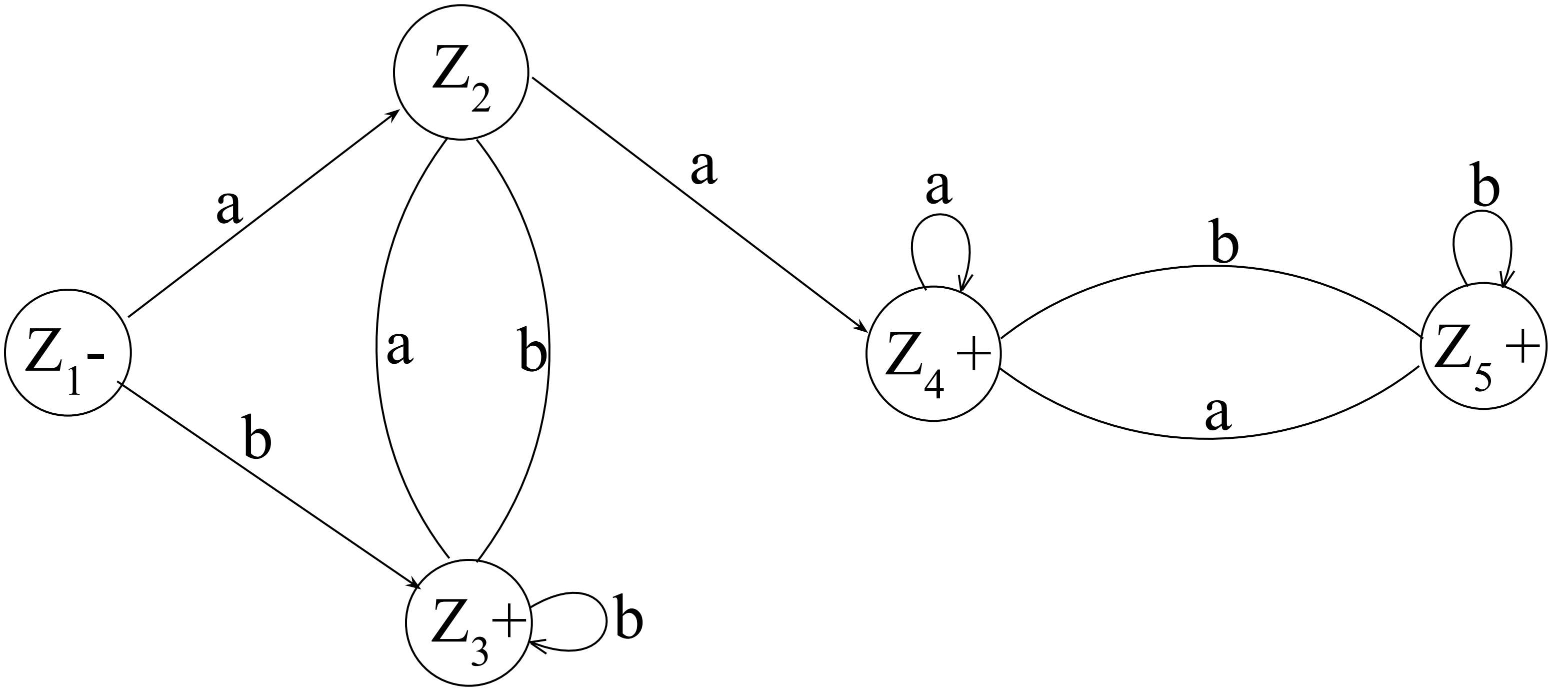
RE corresponding to the above FA may be

$r_1+r_2 = (a+b)^*b + (a+b\ )^*aa(a+b\ )^*$

Note

- It may be noted that in the previous example FA_1 contains two states while FA_2 contains three states. Hence the total number of possible combinations of states of FA_1 and FA_2, in sequence, will be six. For each combination the transitions for both a and b can be determined, but using the method in the example, number of states of FA_3 was reduced to five.

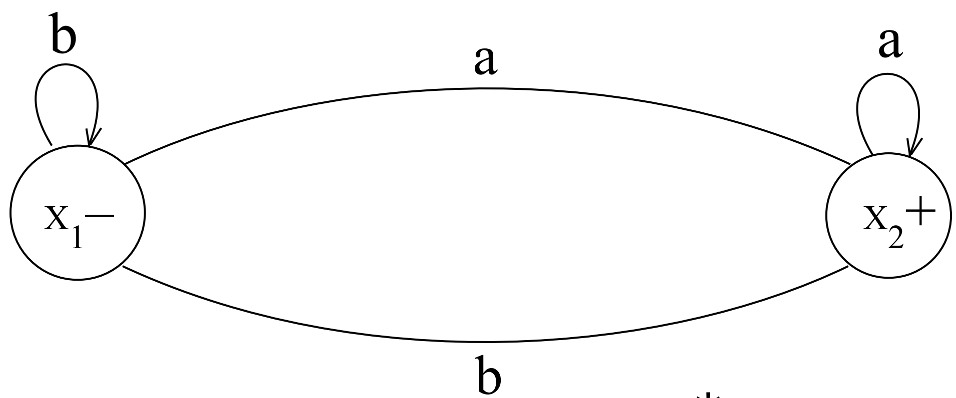
Task



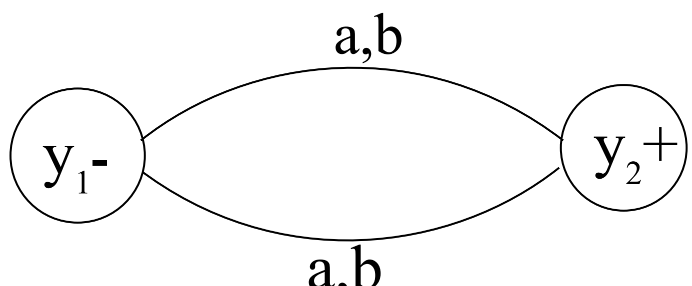
Build an FA equivalent to the previous FA

Example

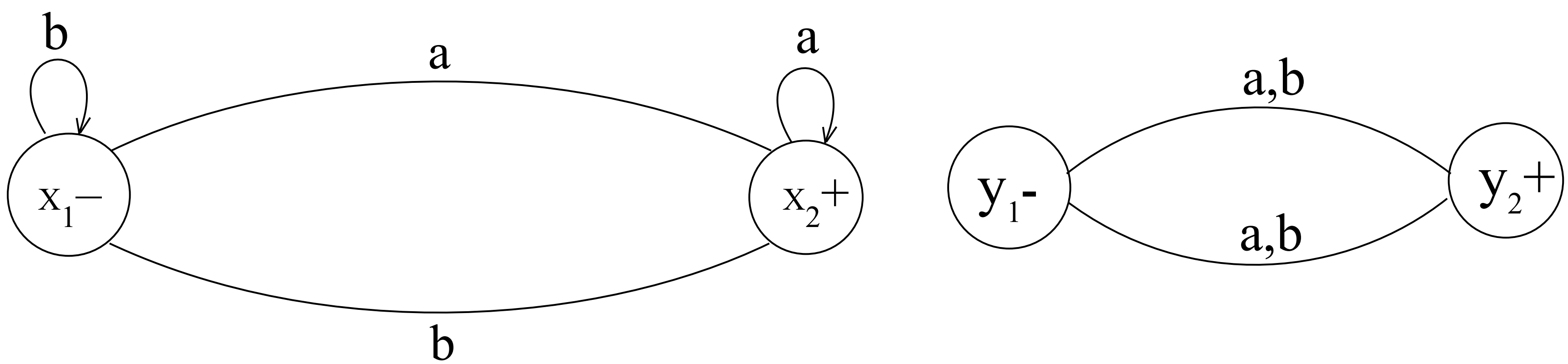
Let r_1=$(a+b)^*a$ and the corresponding FA_1 be



also r_2 = $(a+b)((a+b)(a+b))^*$ or $((a+b)(a+b))^*(a+b)$ and FA_2 be



Example continued …

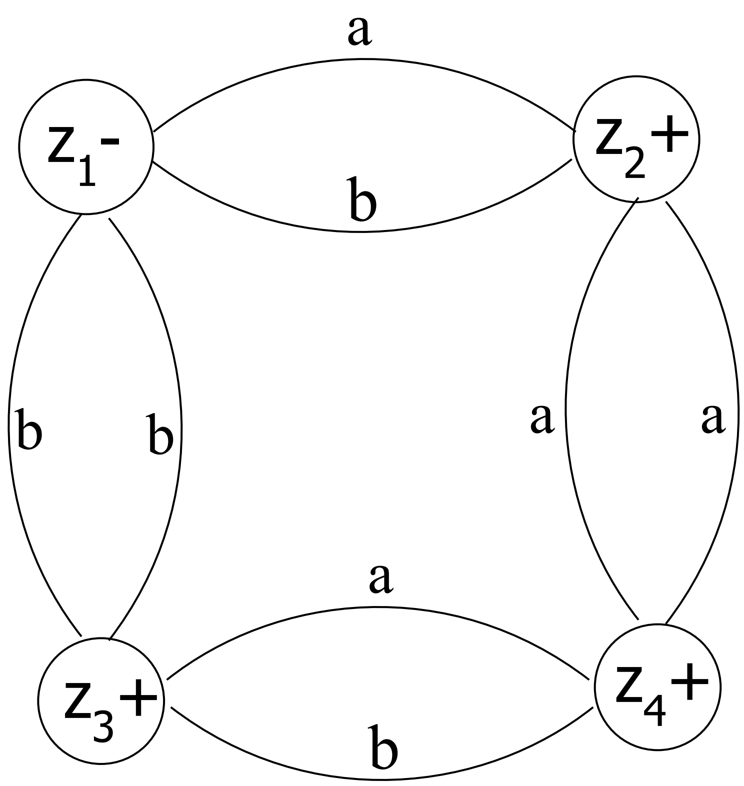


Old States	New States after reading	
	a	b
z_1- □ (x_1,y_1)	(x_2,y_2) □ z_2	(x_1,y_2) □ z_3

Example continued …

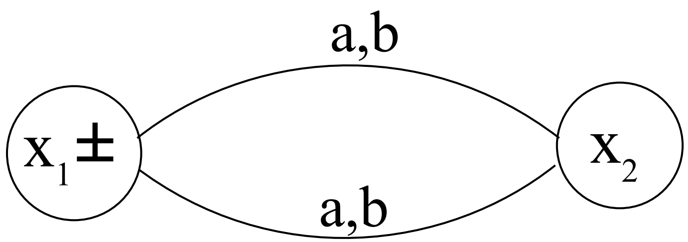
Old States	New States after reading	
	a	b
$z_2+\square(x_2,y_2)$	$(x_2,y_1)\ \square z_4$	$(x_1,y_1)\ \square\ z_1$
$z_3+\ \square(x_1,y_2)$	$(x_2,y_1)\ \square z_4$	$(x_1,y_1)\ \square\ z_1$
$z_4+\ \square(x_2,y_1)$	$(x_2,y_2)\ \square\ z_2$	$(x_1,y_2)\ \square\ z_3$

Example continued ...

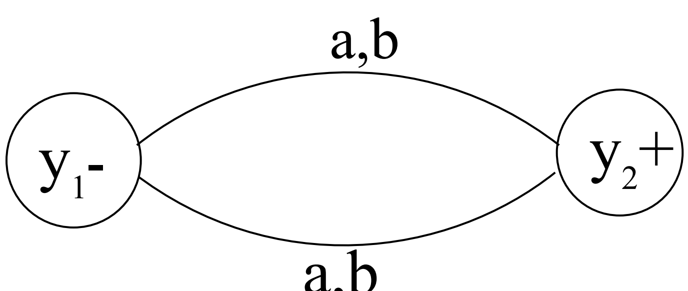


Example

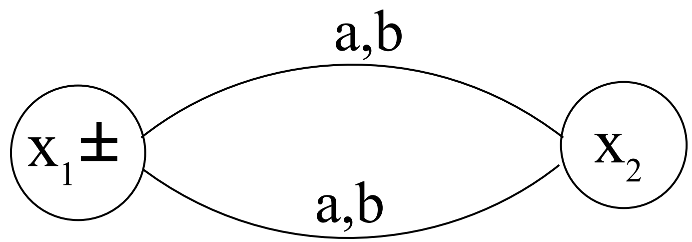
Let $r_1 = ((a+b)(a+b))^*$ and the corresponding FA_1 be

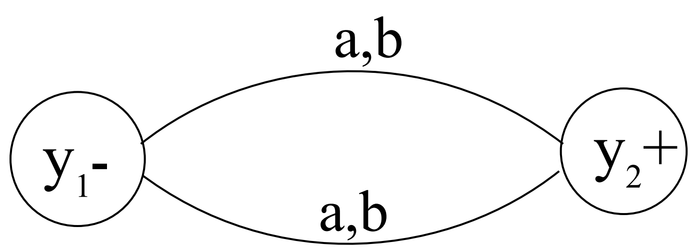


also $r_2 = (a+b)((a+b)(a+b))^*$ or $((a+b)(a+b))^*(a+b)$ and FA_2 be



Example continued ...



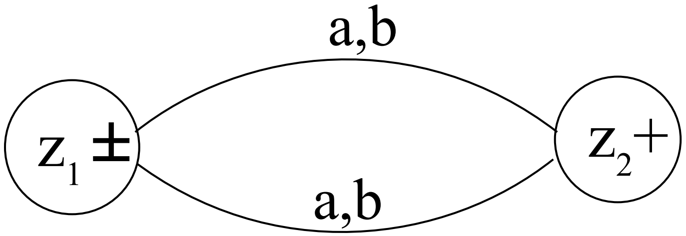


Old States	New States after reading	
	a	b
$z_1 \pm \square (x_1, y_1)$	$(x_2, y_2) \square z_2$	$(x_2, y_2) \square z_2$

Example continued …

Old States	New States after reading	
	a	b
z_2+□(x_2,y_2)	(x_1,y_1) □z_1	(x_1,y_1) □ z_1

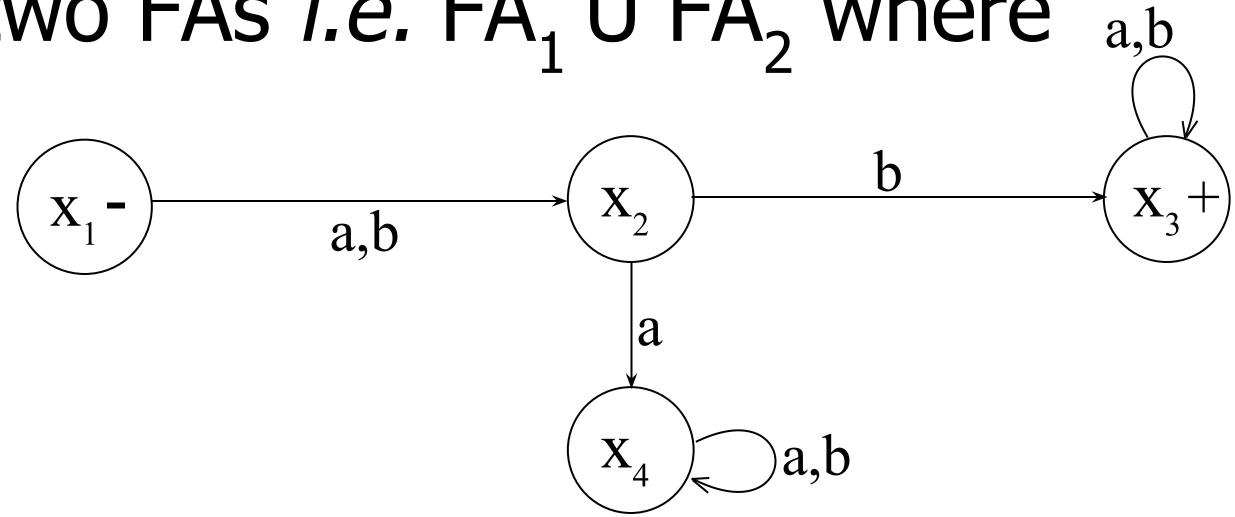
Example continued ...

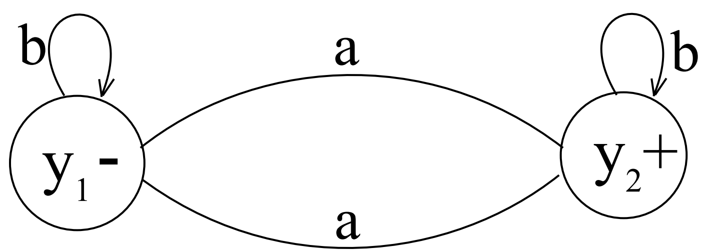


Task

Build an FA corresponding to the union of these two FAs *i.e.* FA_1 U FA_2 where

FA_1





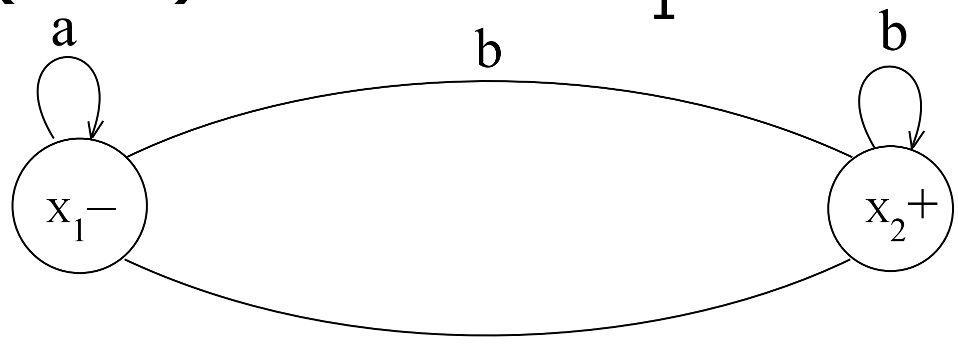
FA_2

Kleene's Theorem Part III Continued …

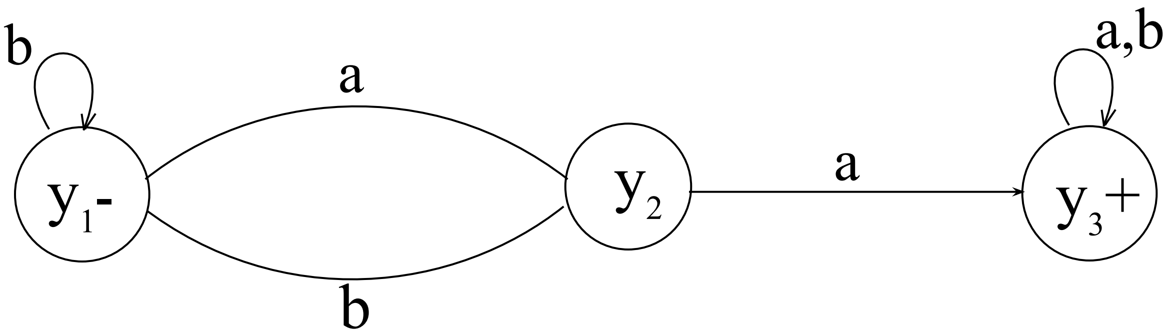
- **Method2 (Concatenation of two FAs):**
  Using the FAs corresponding to r_1 and r_2, an FA can be built, corresponding to r_1r_2. This method can be developed considering the following examples

Example

Let $r_1=(a+b)^*b$ defines L_1 and FA_1 be



and $r_2 = (a+b)^*aa(a+b)^*$ defines L_2 and FA_2 be



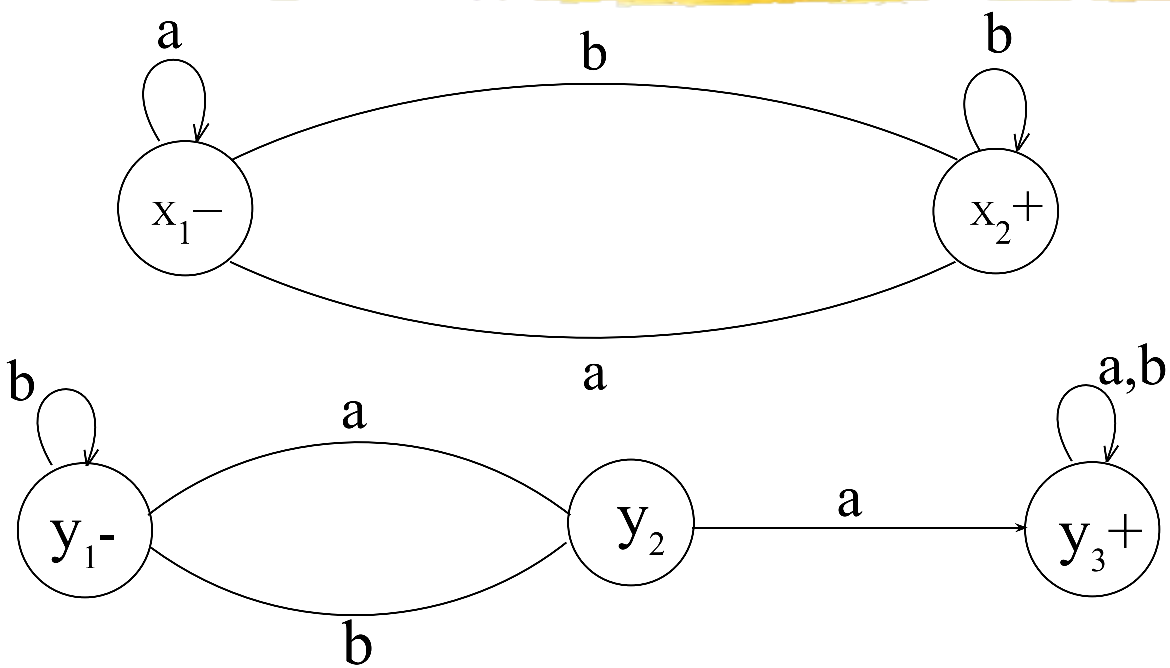
Concatenation of two FAs Continued …

Let FA_3 be an FA corresponding to r_1r_2, then the initial state of FA_3 must correspond to the initial state of FA_1 and the final state of FA_3 must correspond to the final state of FA_2. Since the language corresponding to r_1r_2 is the concatenation of corresponding languages L_1 and L_2, consists of the strings obtained, concatenating the strings of L_1 to those of L_2 , therefore ***the moment a final state of first FA is entered, the possibility of the initial state of second FA will be included as well.***

Concatenation of two FAs Continued …

Since, in general, FA_3 will be different from both FA_1 and FA_2, so the labels of the states of FA_3 may be supposed to be z_1,z_2, z_3, …, where z_1 stands for the initial state. Since z_1 corresponds to the states x_1, so there will be two transitions separately for each letter read at z_1. It will give two possibilities of states which correspond to either z_1 or different from z_1. This process may be expressed in the following transition table for all possible states of FA_3

Example continued …

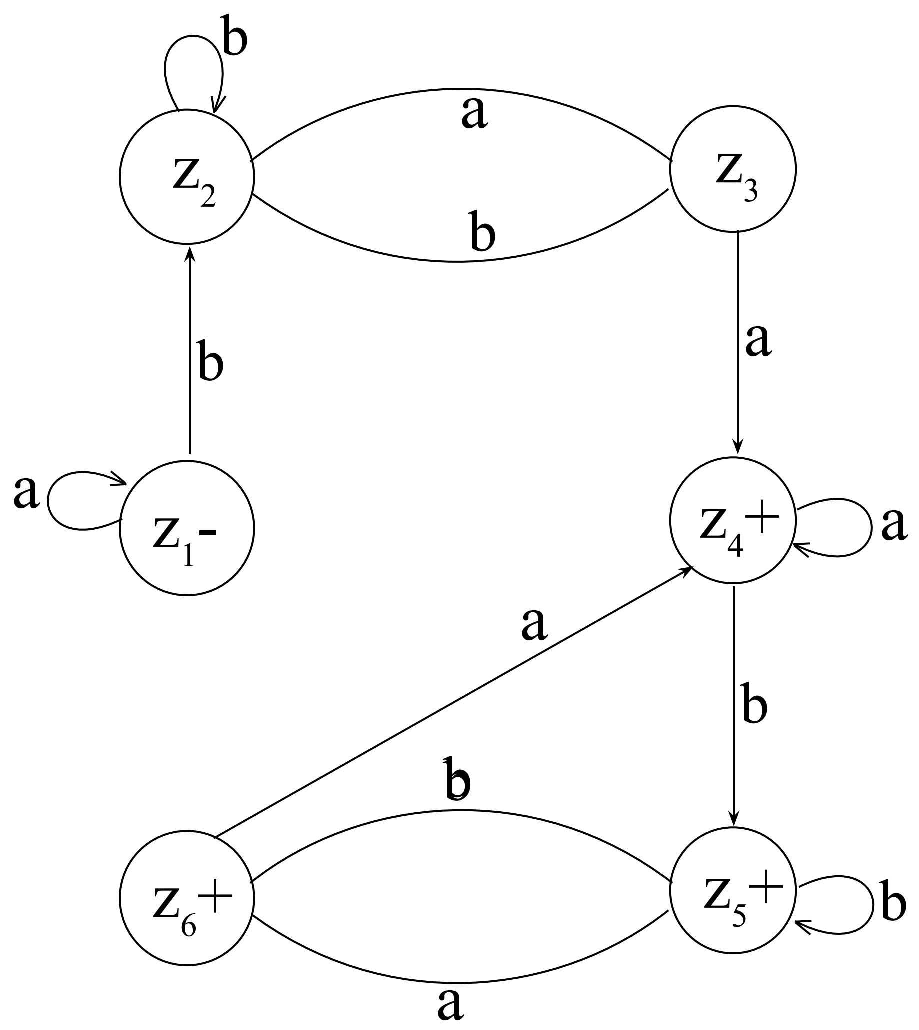


Old States	New States after reading	
	a	b
z_1-□x_1	x_1□z_1	(x_2,y_1)□ z_2

Example continued …

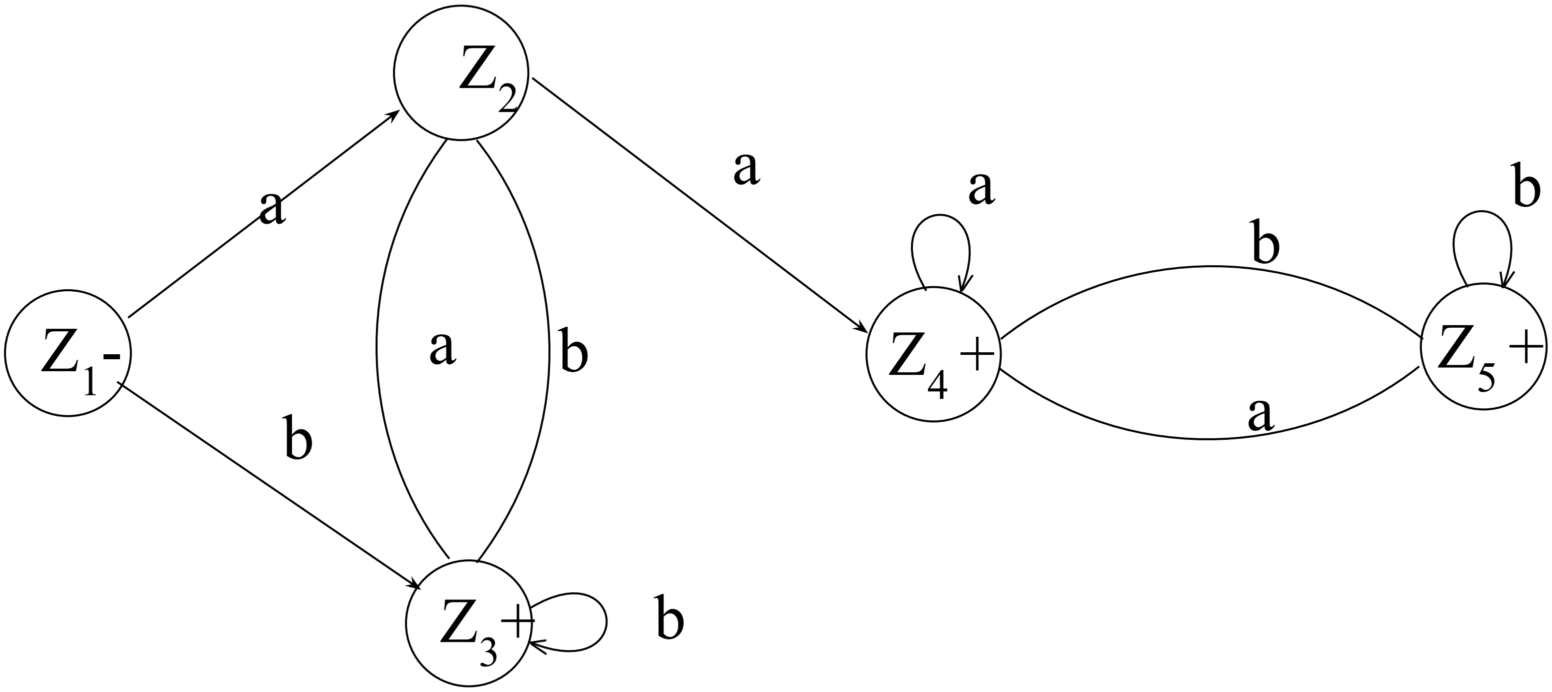
Old States	New States after reading	
	a	b
$z_2 \square (x_2,y_1)$	$(x_1,y_2) \square z_3$	$(x_2,y_1) \square\ z_2$
$z_3 \square (x_1,y_2)$	$(x_1,y_3) \square z_4$	$(x_2,y_1) \square\ z_2$
$z_4 + \square (x_1,y_3)$	$(x_1,y_3) \square\ z_4$	$(x_2,y_1,y_3) \square\ z_5$
$z_5 + \square (x_2,y_1,y_3)$	$(x_1,y_2\ ,y_3) \square\ z_6$	$(x_2,y_1,y_3) \square\ z_5$
$z_6 + \square (x_1,y_2,y_3)$	$(x_1,y_3) \square\ z_4$	$(x_2\ ,y_1,y_3) \square\ z_5$

Example continued ...

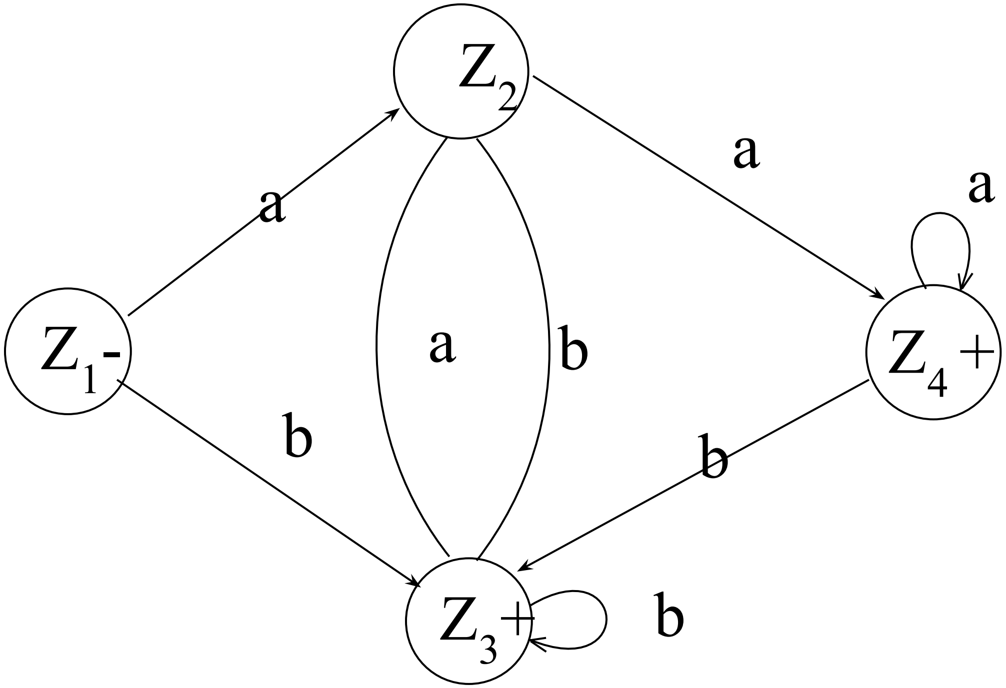


Task

- Build an FA equivalent to the following FA

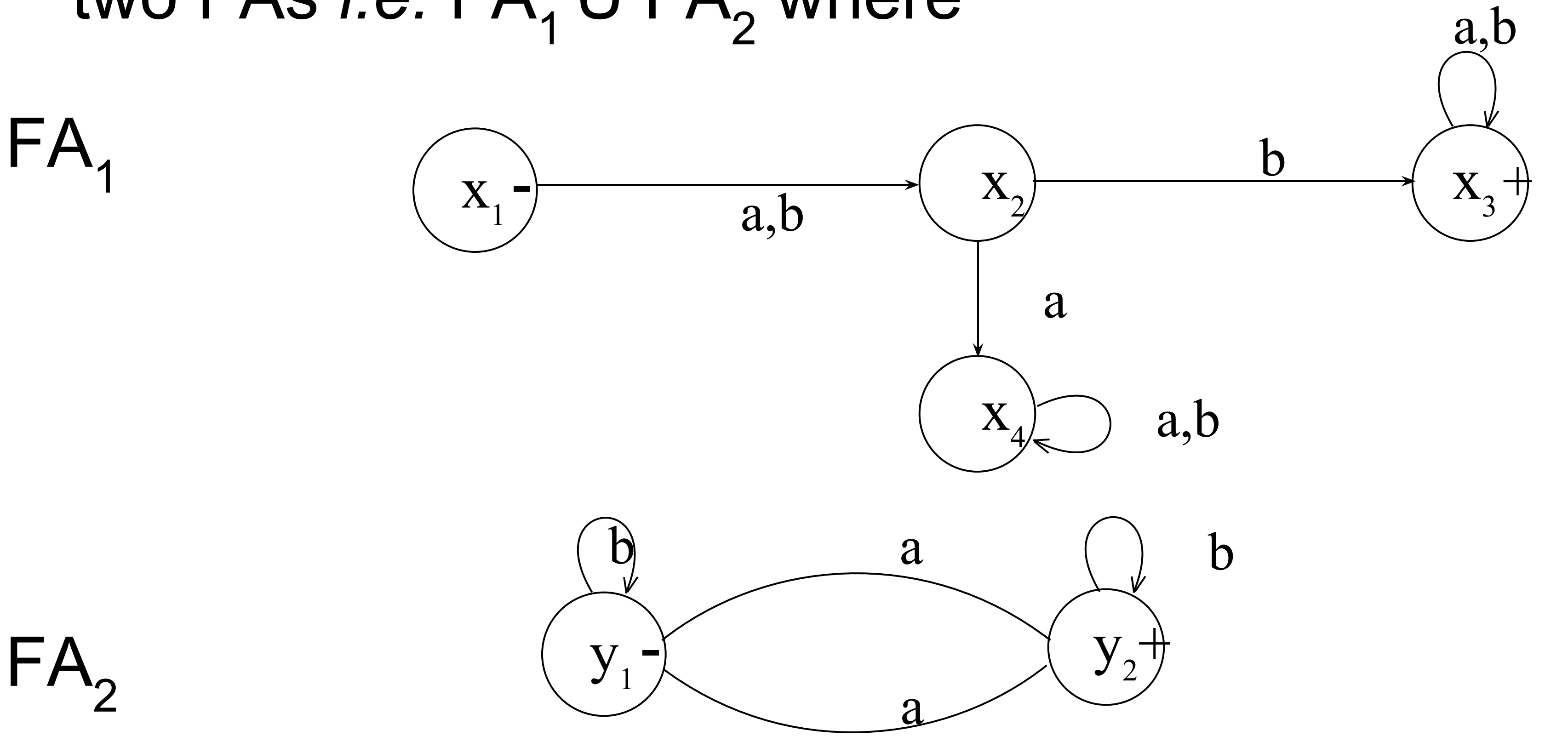


Solution of the Task



Task

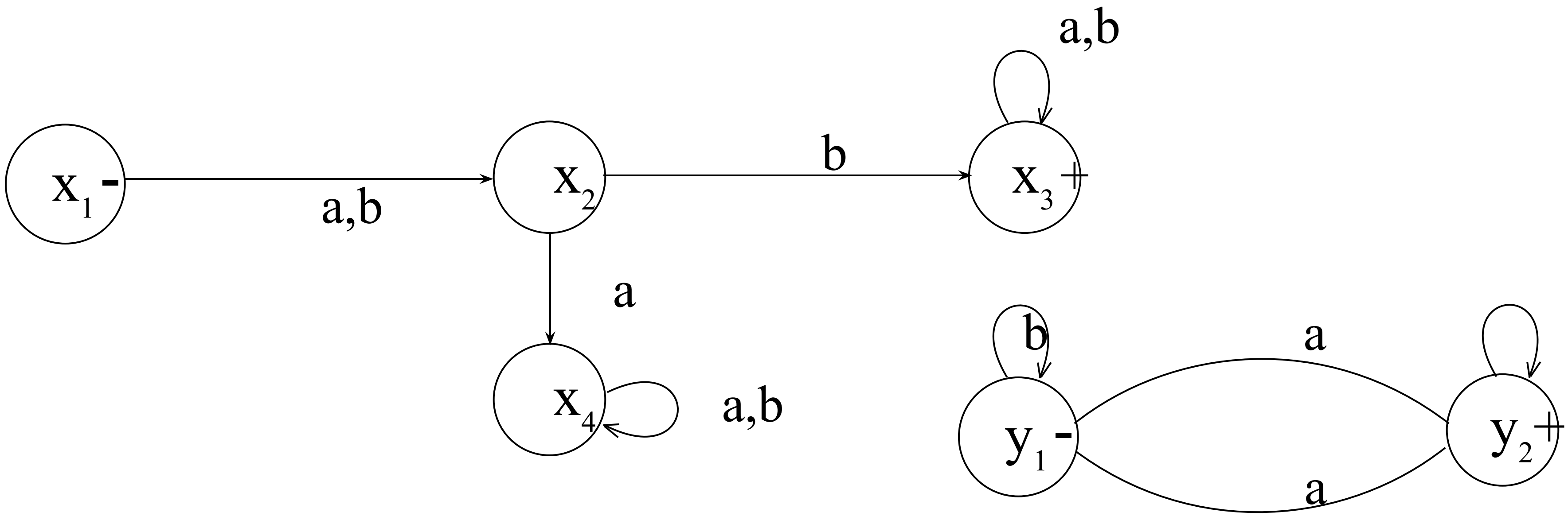
Build an FA corresponding to the union of these two FAs *i.e.* FA_1 U FA_2 where



Task solution

- RE corresponding to FA_1 may be $(a+b)b(a+b)^*$ which generates the language of strings, defined over $\Sigma=\{a,b\}$, **with b as second letter.**
- RE corresponding to FA_2 may be $b^*a(b+ab^*a)^*$ which generates the language of strings, defined over $\Sigma=\{a,b\}$, **with odd number of a's.**

Solution continued …

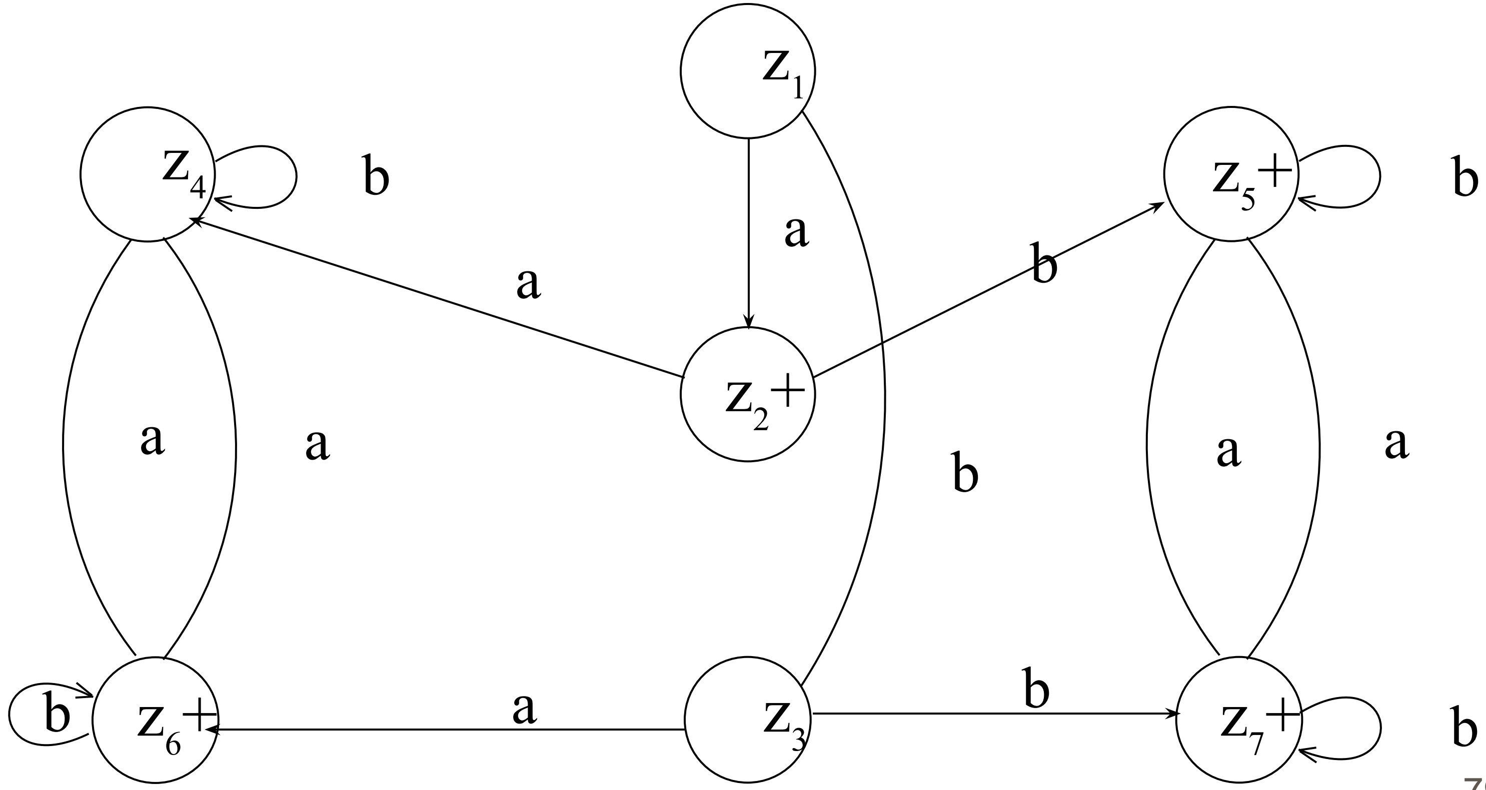


Old States	New States after reading	
	a	b
$\mathbf{z_1}$- □ $(\mathbf{x_1,y_1})$	$(\mathbf{x_2,y_2})$ □ $\mathbf{z_2}$	$(\mathbf{x_2,y_1})$ □ $\mathbf{z_3}$

Solution continued …

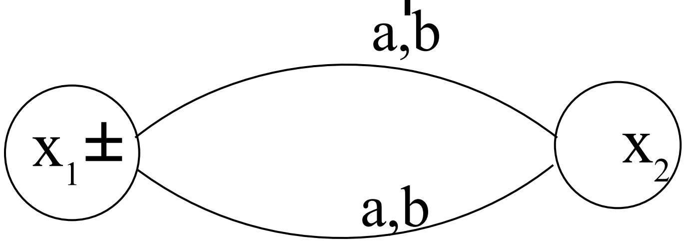
Old States	New States after reading	
	a	b
z_2+□(x_2 , y_2)	(x_4,y_1)□z_4	(x_3,y_2)□ z_5
z_3 □(x_2,y_1)	(x_4,y_2)□z_6	(x_3,y_1)□ z_7
z_4□(x_4,y_1)	(x_4,y_2)□ z_6	(x_4,y_1)□ z_4
z_5+□(x_3,y_2)	(x_3,y_1)□ z_7	(x_3,y_2)□ z_5
z_6+□(x_4,y_2)	(x_4,y_1)□z_4	(x_4,y_2)□ z_6
z_7+□(x_3,y_1)	(x_3,y_2)□z_5	(x_3,y_1)□ z_7

Solution continued …

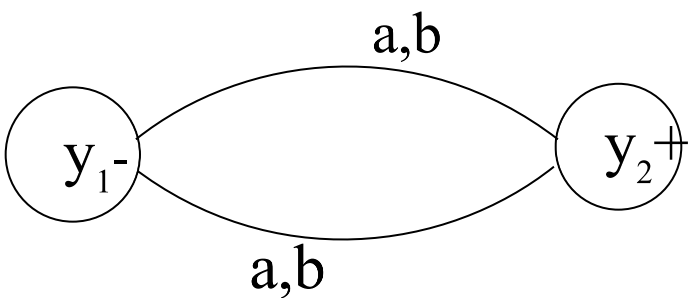


Example

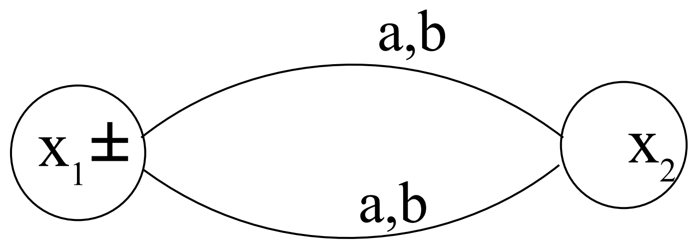
Let $r_1=((a+b)(a+b))^*$ and the corresponding FA_1 be

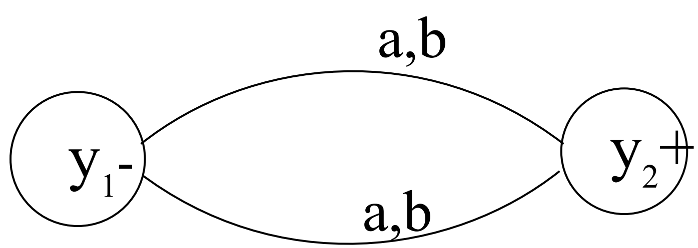


also $r_2 = (a+b)((a+b)(a+b))^*$ or $((a+b)(a+b))^*(a+b)$ and FA_2 be



Example continued …



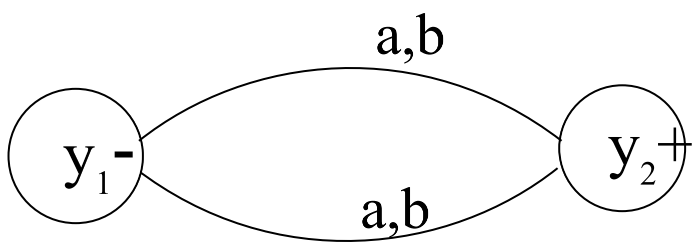


Old States	New States after reading	
	a	b
$z_1-\square(x_1,y_1)$	$(x_2,y_2)\ \square z_2$	$(x_2,y_2)\ \square\ z_2$

Example continued …

Old States	New States after reading	
	a	b
$z_2 + \square (x_2, y_2)$	$(x_1, y_1) \; \square z_1$	$(x_1, y_1) \; \square \; z_1$

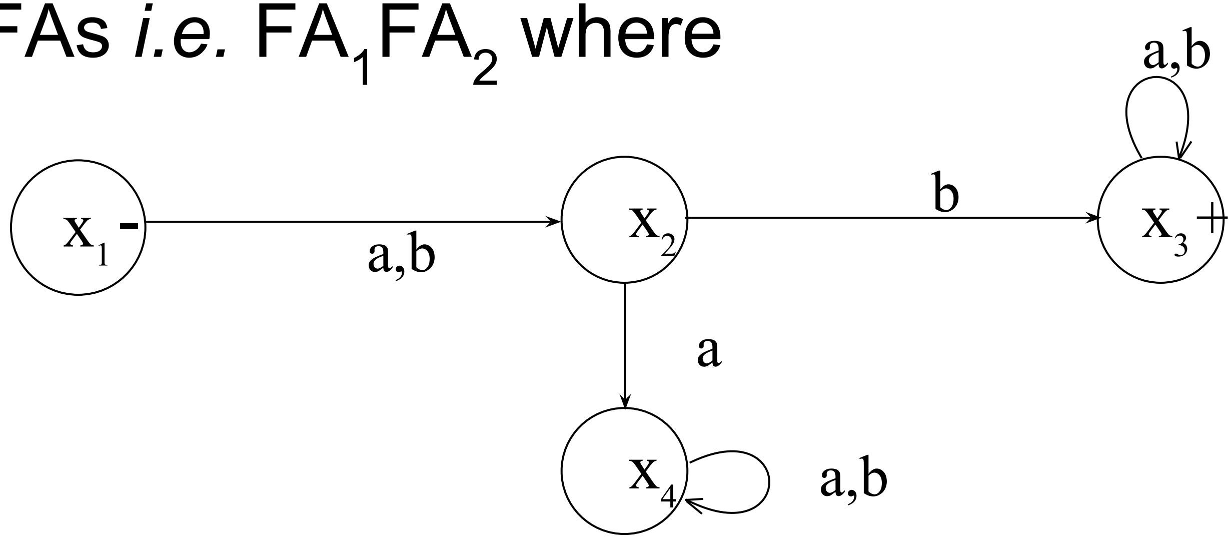
Example continued …

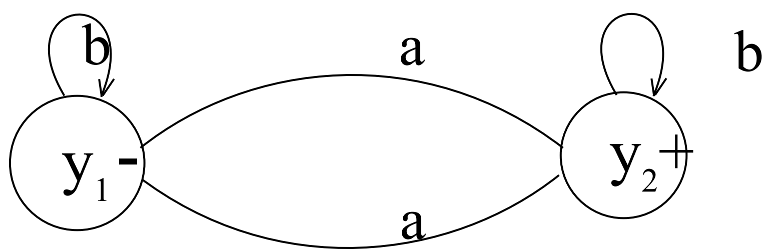


Task

Build FA corresponding to the concatenation of these two FAs *i.e.* FA_1FA_2 where

FA_1





FA_2

Kleene's Theorem Part III Continued …

- **Method3: (Closure of an FA)**

  Building an FA corresponding to r^*, using the FA corresponding to r.

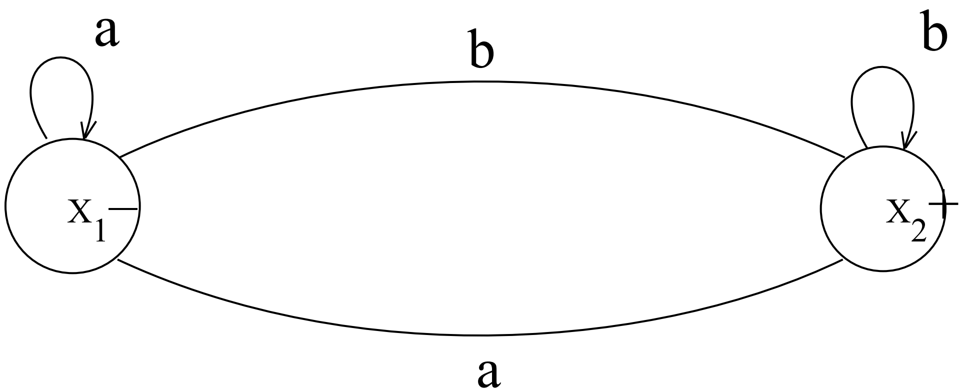
  It is to be noted that if the given FA already accepts the language expressed by the closure of certain RE, then the given FA is the required FA. However the method, in other cases, can be developed considering the following examples

Closure of FA Continued …

Closure of an FA, is same as concatenation of an FA with itself, except that the initial state of the required FA is a final state as well. Here the initial state of given FA, corresponds to the initial state of required FA and a non final state of the required FA as well.

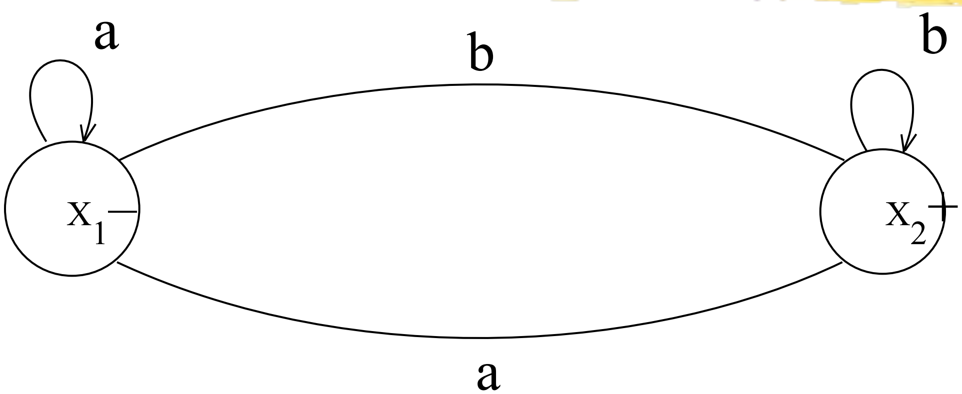
Example

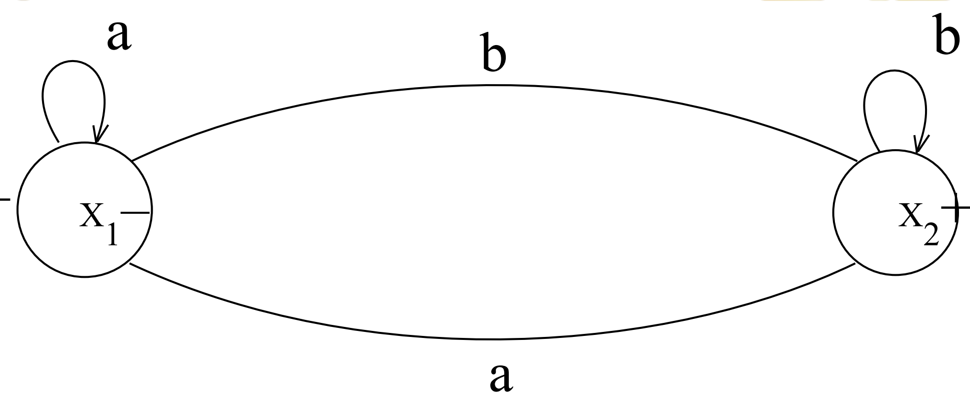
Let $r=(a+b)^*b$ and the corresponding FA be



then the FA corresponding to r^* may be determined as under

Example continued …





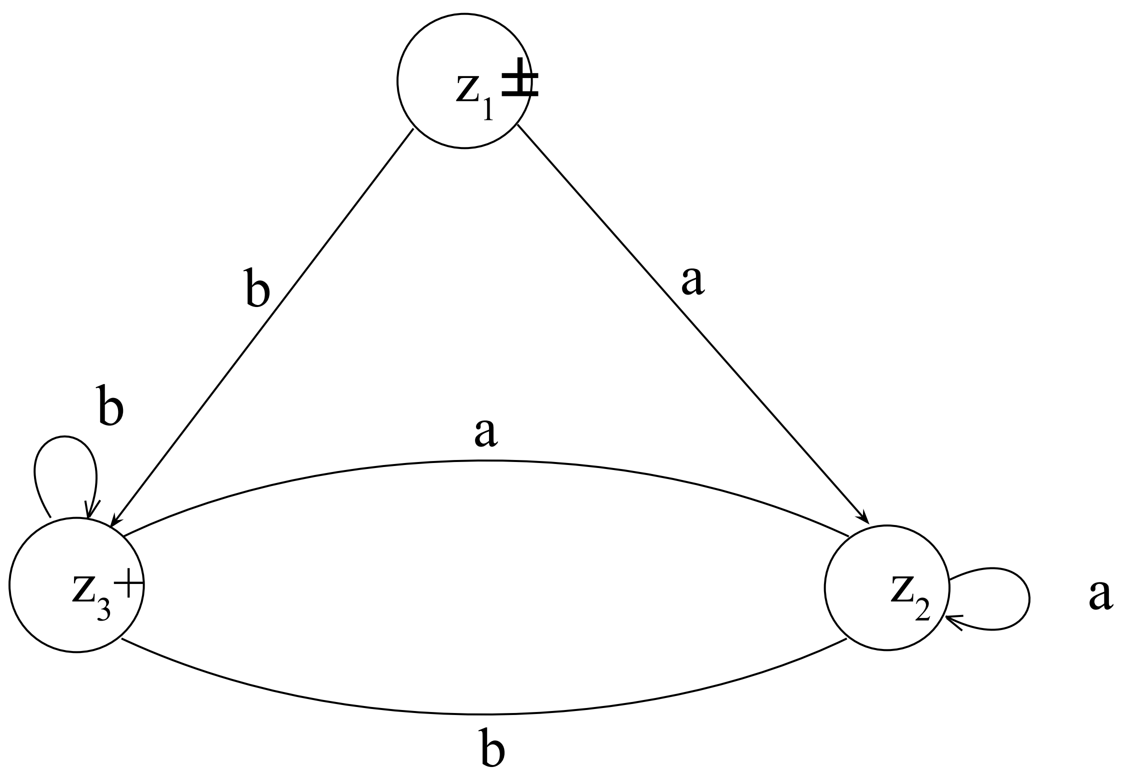
Old States	New States after reading	
	a	b
Final $z_1 \pm \square x_1$	$x_1 \square z_2$	$(x_2,x_1) \square\ z_3$

Example continued …

Old States	New States after reading	
	a	b
Non-final $z_2 \square x_1$	$x_1 \square z_2$	$(x_2,x_1) \square z_3$
$z_3+ \square (x_2,x_1)$	$x_1 \square z_2$	$(x_2,x_1) \square z_3$

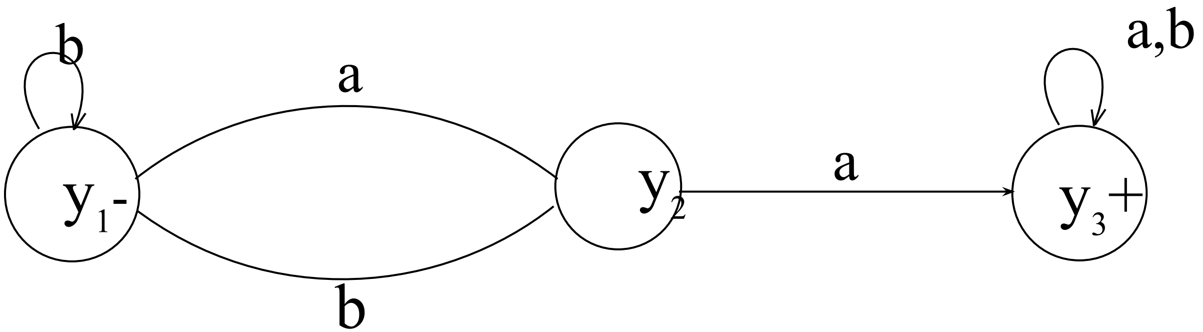
The corresponding transition diagram may be as under

Example continued …

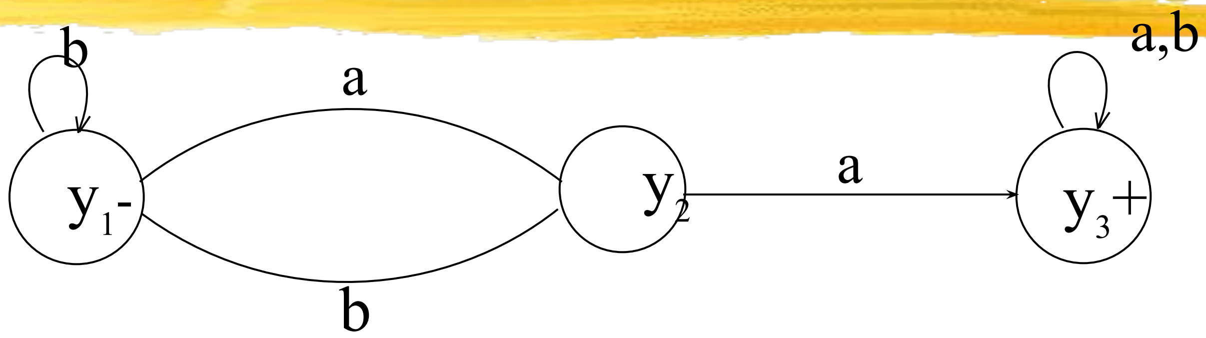


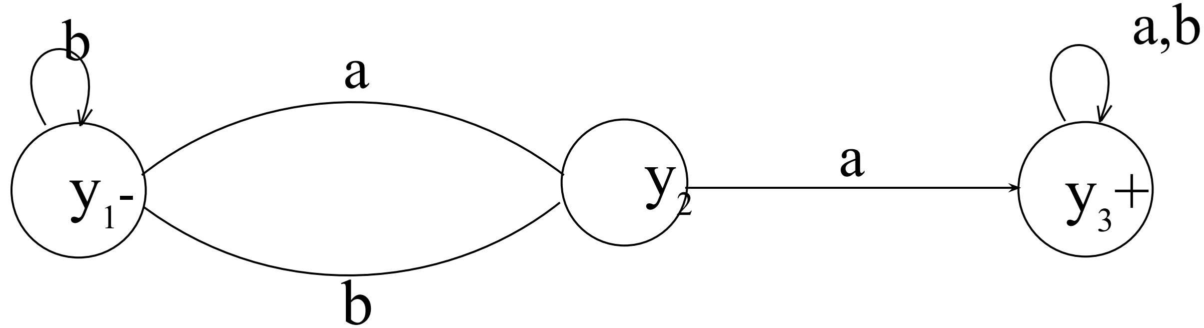
Example

Let $r=(a+b)^{*}aa(a+b)^{*}$ and the corresponding FA be



Example continued …



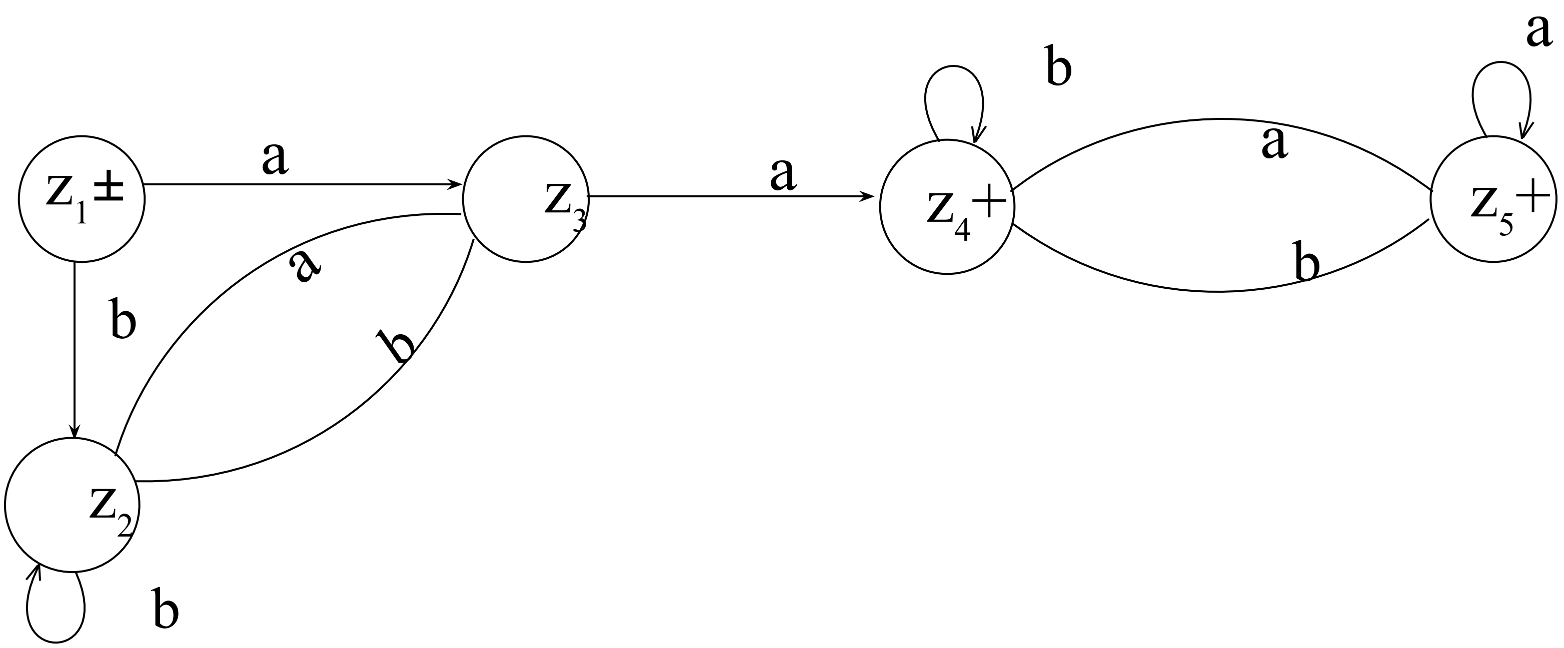


Old States	New States after reading	
	a	b
Final $z_1 \pm$ □ y_1	y_2 □ z_3	y_1 □ z_2

Example continued …

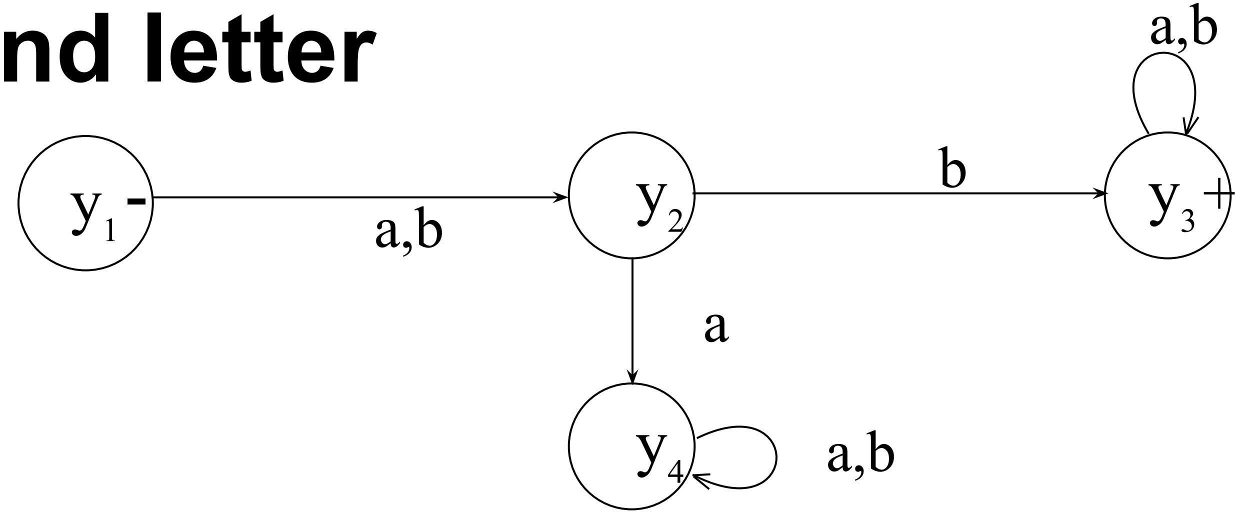
Old States	New States after reading	
	a	b
$z_2 \equiv y_1$	$y_2 \equiv z_3$	$y_1 \equiv z_2$
$z_3 \equiv y_2$	$(y_3, y_1) \equiv z_4$	$y_1 \equiv z_2$
$z_4^+ \equiv (y_3, y_1)$	$(y_3, y_1, y_2) \equiv z_5$	$(y_3, y_1) \equiv z_4$
$z_5^+ \equiv (y_3, y_1, y_2)$	$(y_3, y_1, y_2) \equiv z_5$	$(y_3, y_1) \equiv z_4$

Example continued …

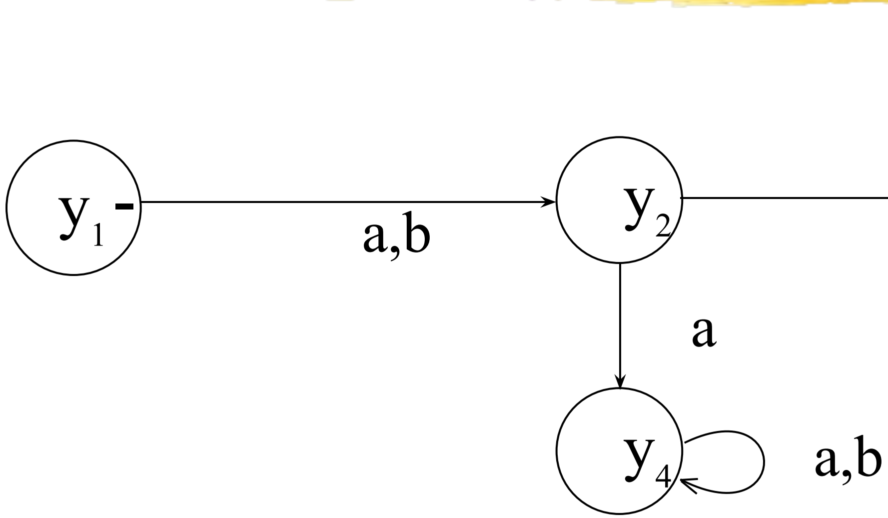


Example

Consider the following FA, accepting the language of strings with **b as second letter**



Example continued …



Old States	New States after reading	
	a	b
$z_1 \pm \square y_1$	$y_2 \square z_2$	$y_2 \square\ z_2$

Example continued ...

Old States	New States after reading	
	a	b
$z_2 \square y_2$	$y_4 \square z_3$	$(y_3,y_1) \square\ z_4$
$z_3 \square y_4$	$y_4 \square z_3$	$y_4 \square z_3$
$z_4 + \square (y_3,y_1)$	$(y_3 ,y_1 ,y_2) \square\ z_5$	$(y_3 ,y_1 ,y_2) \square\ z_5$
$z_5 + \square (y_3,y_1,y_2)$	$(y_3,y_1 ,y_2 ,y_4) \square\ z_6$	$(y_3,y_1,y_2) \square\ z_5$
$z_6 \square (y_3,y_1 ,y_2 ,y_4)$	$(y_1,y_1 ,y_2 ,y_4) \square z_6$	$(y_1,y_1 ,y_2 ,y_4) \square z_6$