# NFAs and Transition Graphs



Course: Theory Of Automata
Topic: NFAs And Transition Graphs
Instructor: Ahmed Mateen

# Deterministic FA (DFA)



- The FAs that we have studied so far are DFA in that
  - At every state there is exactly one outgoing transition for a character and the machine can follow the transition deterministically
    - No duplicates
    - No missing edges

# Nondeterministic Finite Automata (NFA)



- The FA where a state can have more than one transition for the same character.
- This puts the machine in an indecisive state for which transition to follow
  - Has duplicate transitions
  - Can miss transitions for some characters
- Reduces number of states and transitions

# Nondeterministic Finite Automata (NFA)



- Costly execution
  - Needs concurrent processing to find a successful path
- An NFA can have a successful and unsuccessful path for the same input
- If an NFA has at least one successful path for an input it is considered to be valid
- Machine crashes for an undefined transition thus causing implicit reject

# NFA Language recognition



- Acceptance
  - If at least one successful path exists
- Rejection
  - Either machine crashes on input
  - No successful path exists

# Nondeterministic Finite Automata (NFA)



- Examples
  - An NFA that accepts the language {*bb*, *bbb*}
  - All words that contain *bb* in them
  - All words contains a double letter

# Epsilon Transitions

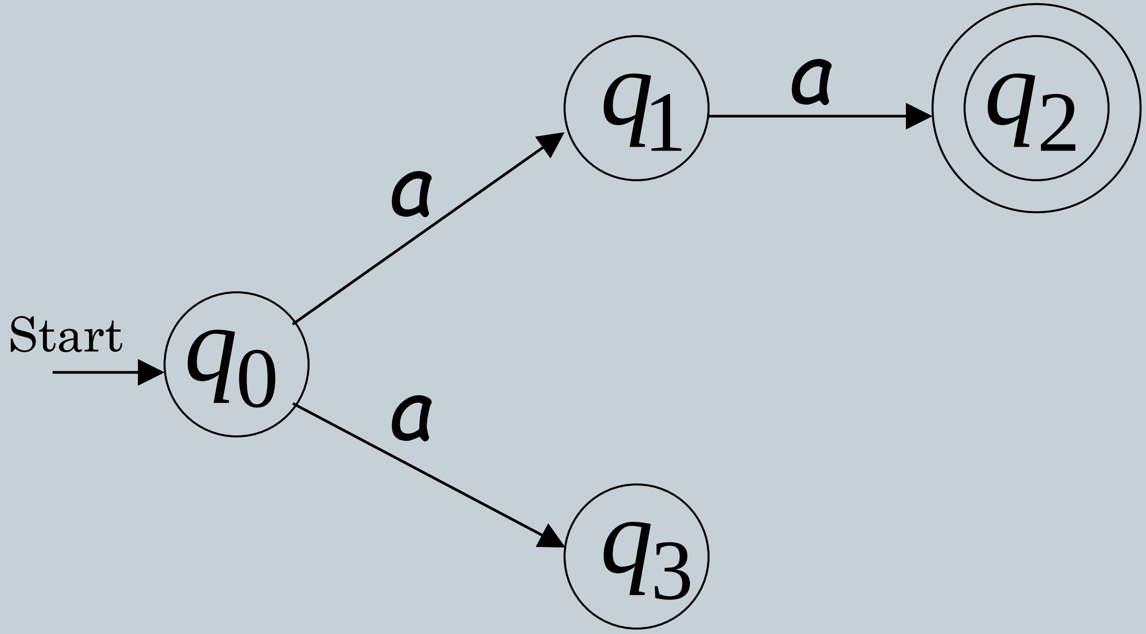


- ϵ- Transitions
- A null transition that changes state but doesn't consume any character
- Possible with NFAs and Transition Graphs (discussed next)

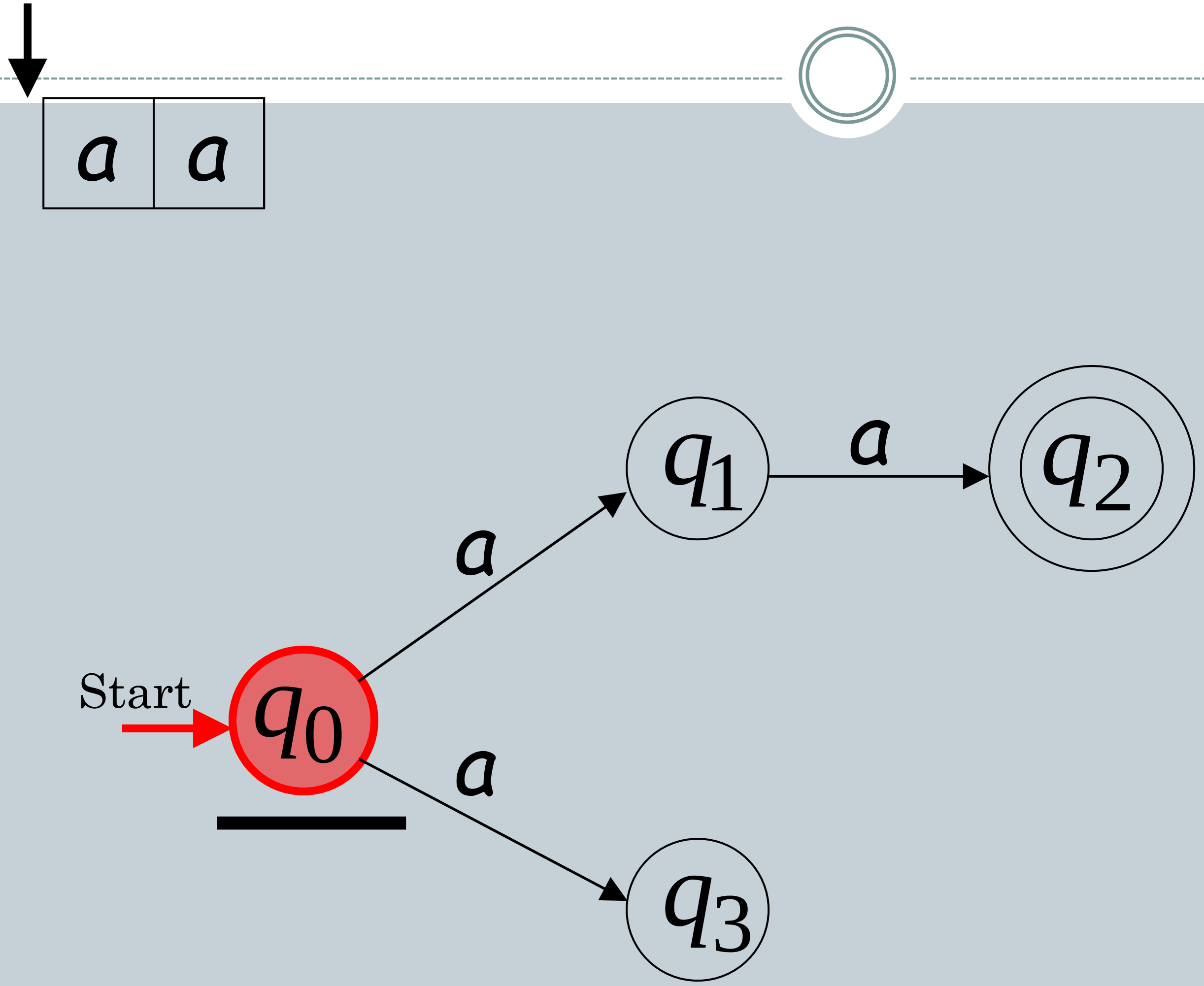
# Nondeterministic Finite Automata (NFA)



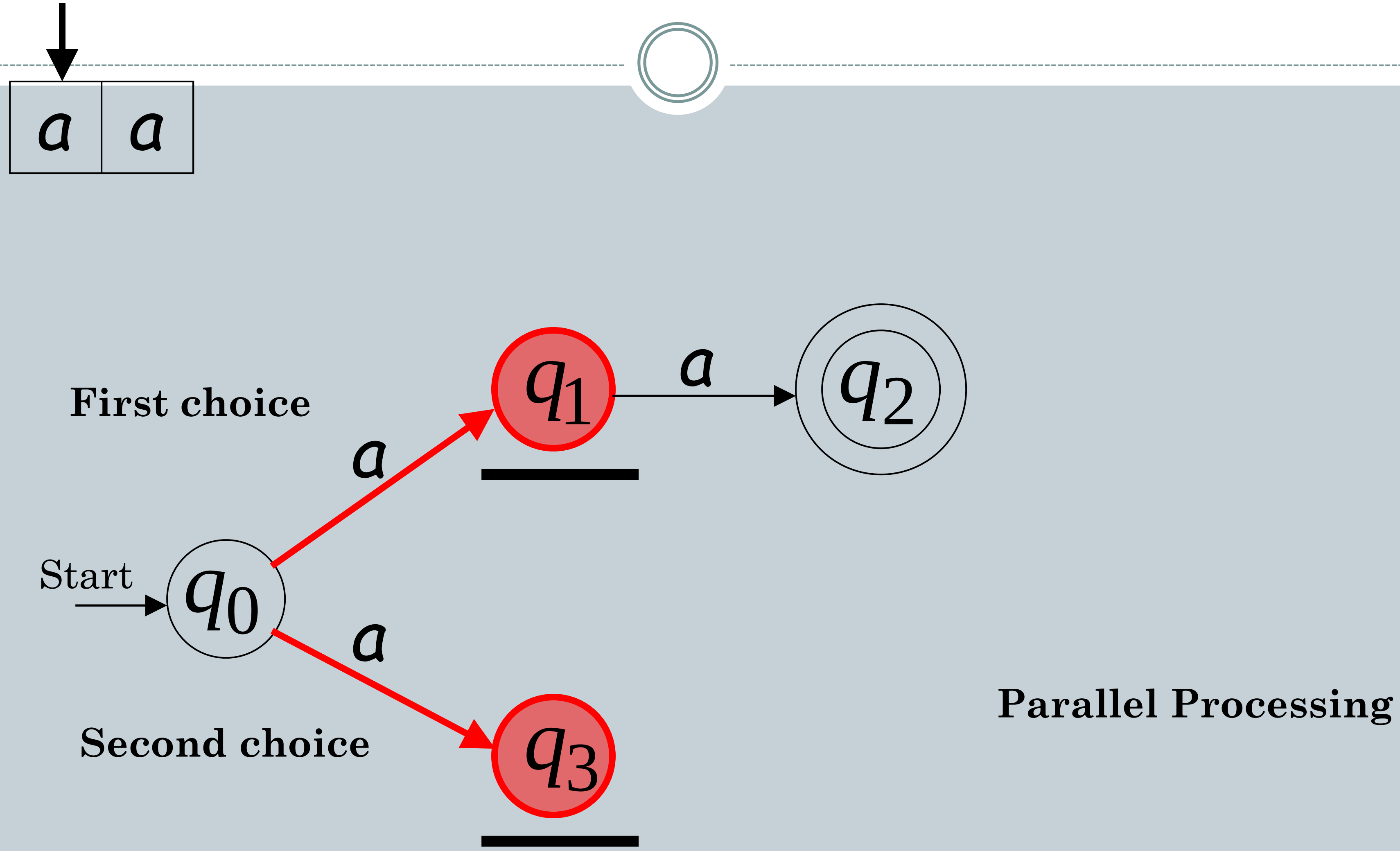
**Alphabet =** $\{a\}$



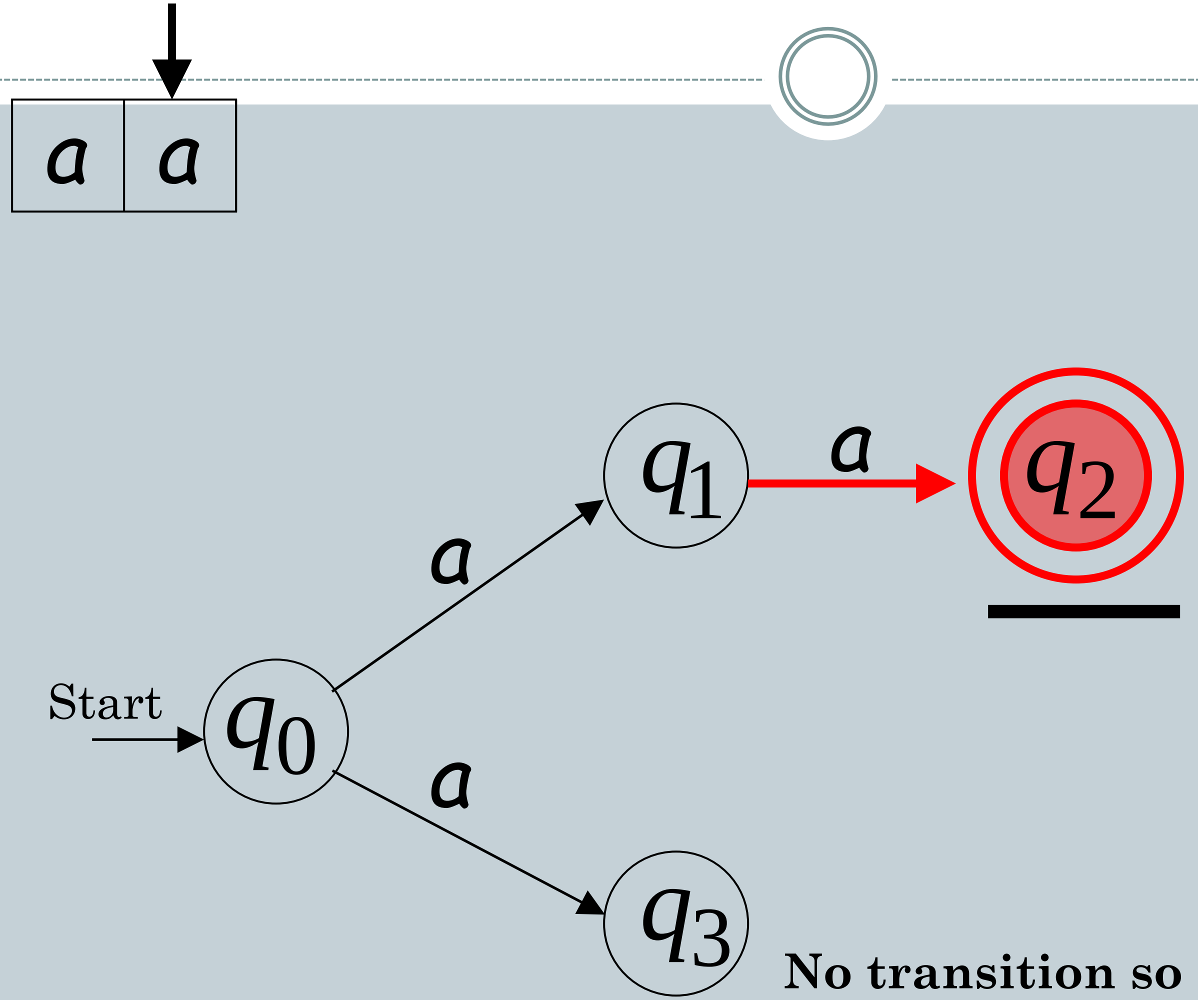
# Example: Accepting



# Example: Accepting

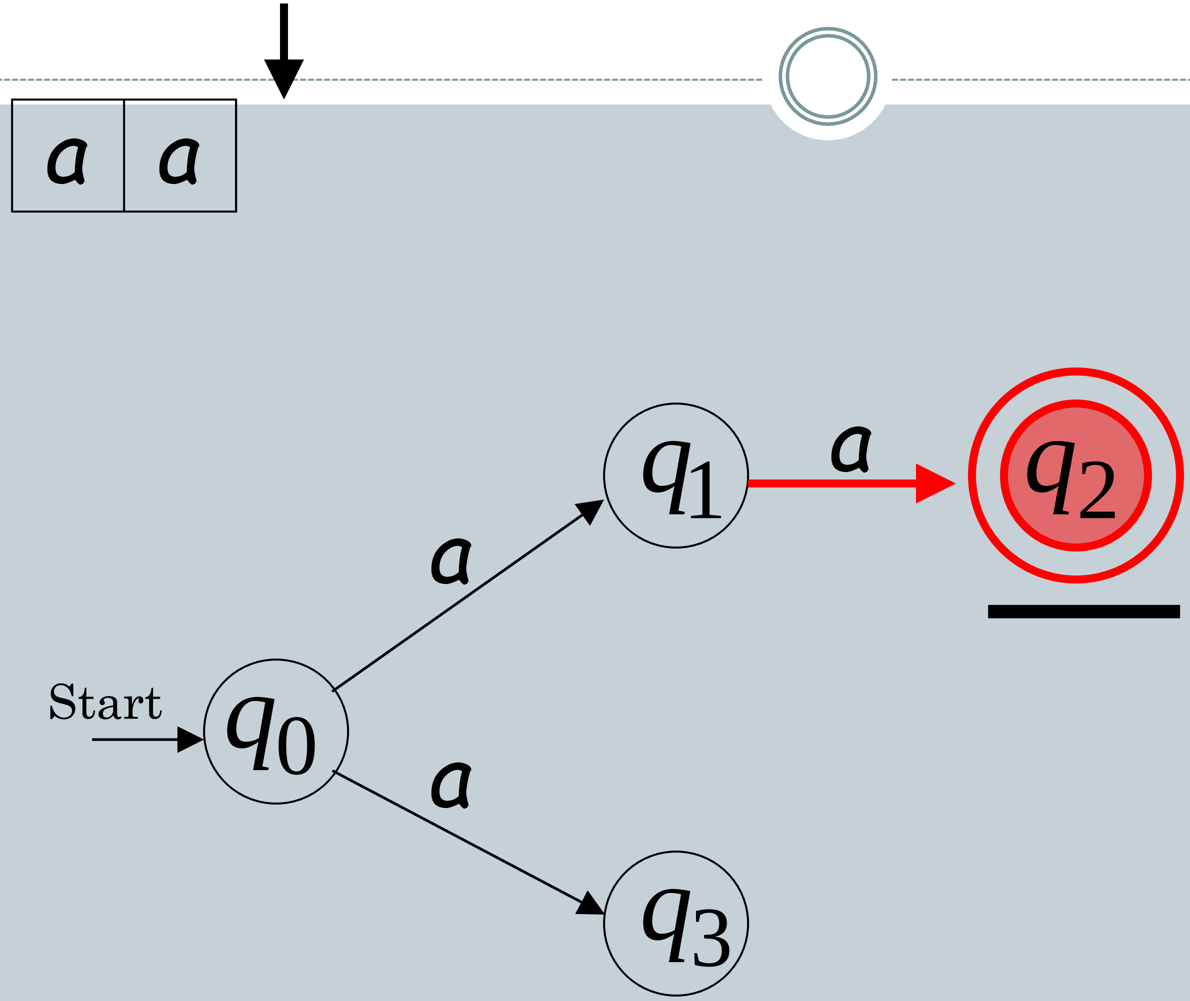


# Example: Accepting



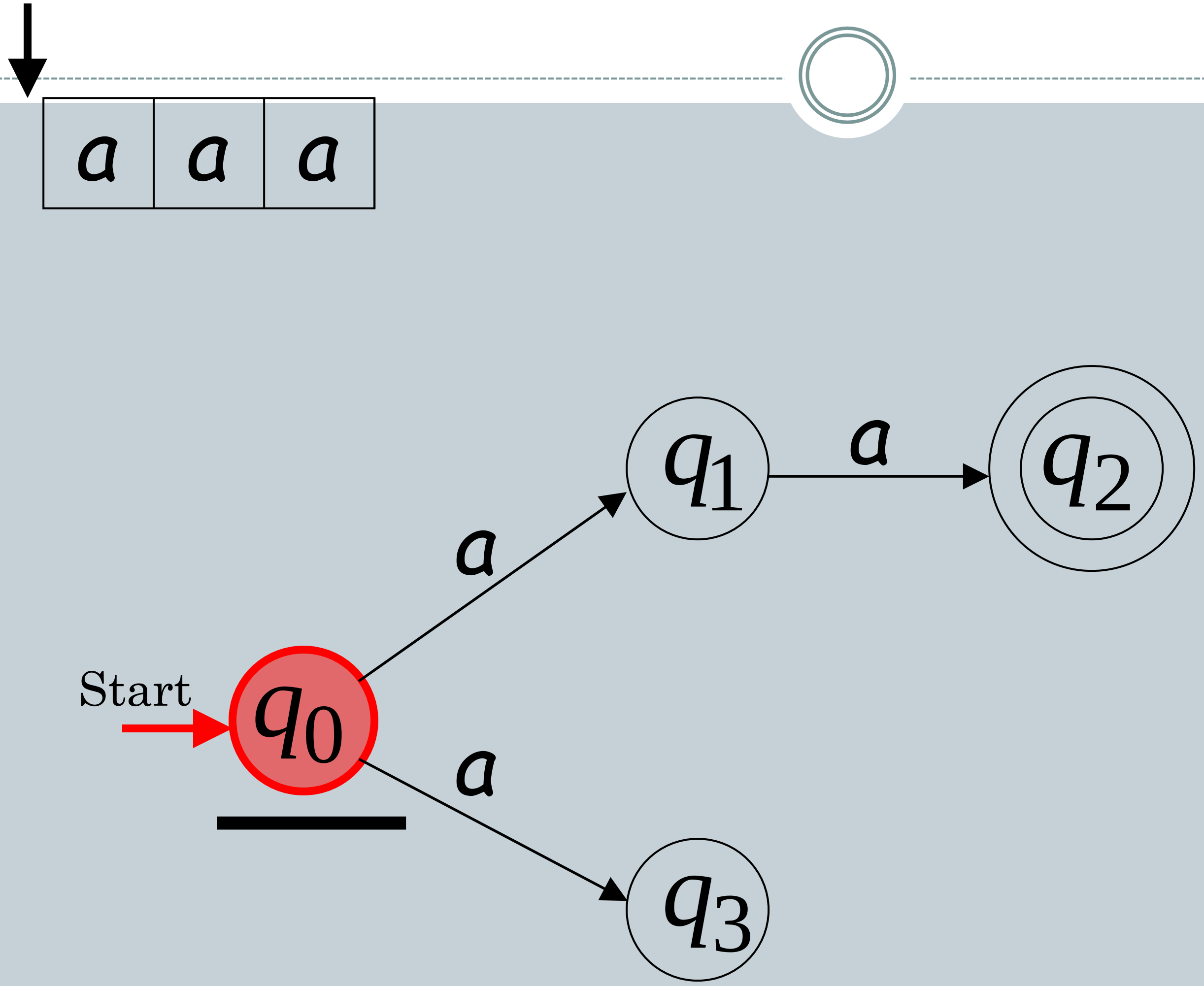

**No transition so leave it**

# Example: Accepting

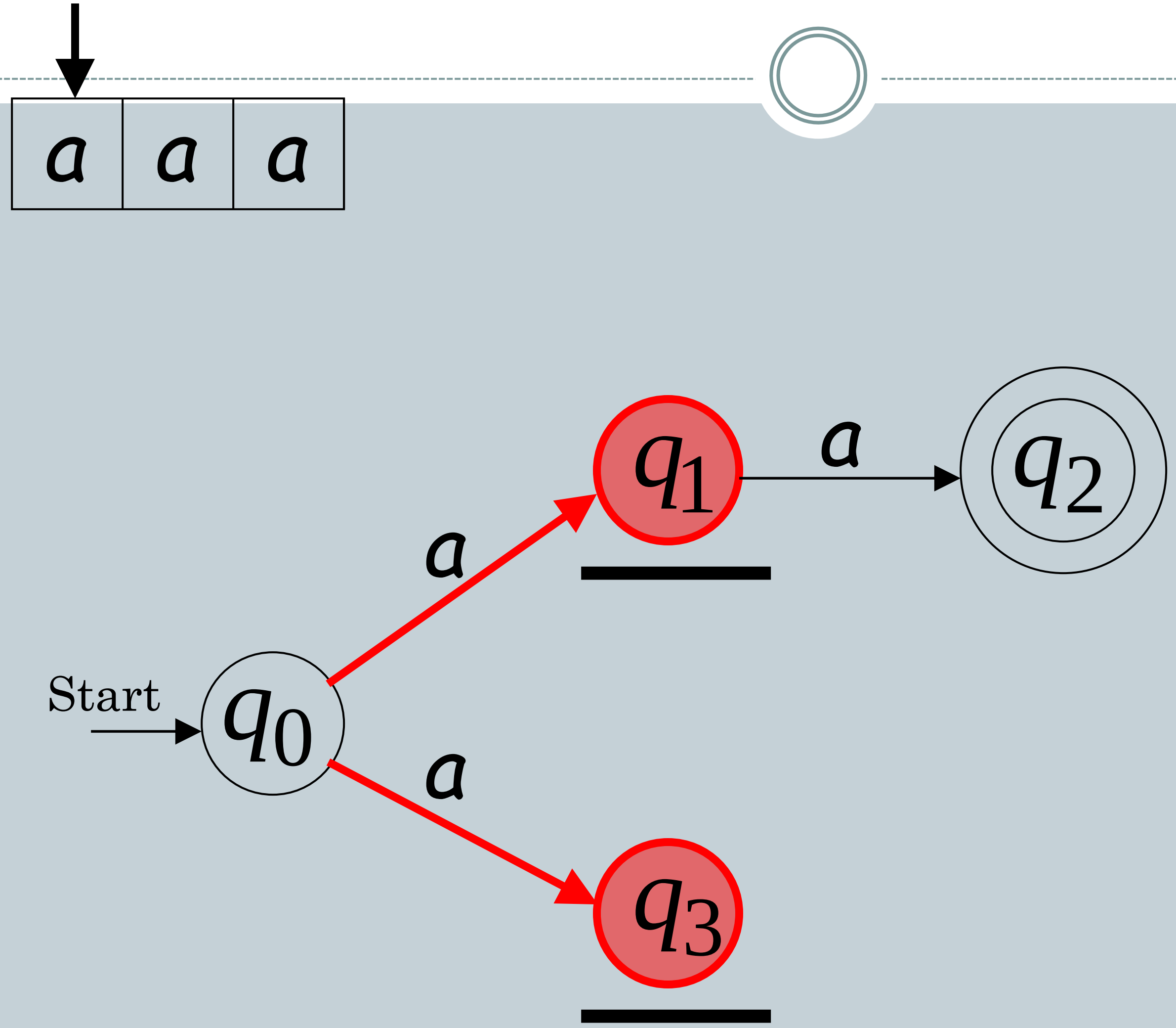



Since there is no more symbol to read and it is an accepting state Therefore, the NFA will "**Accept**"

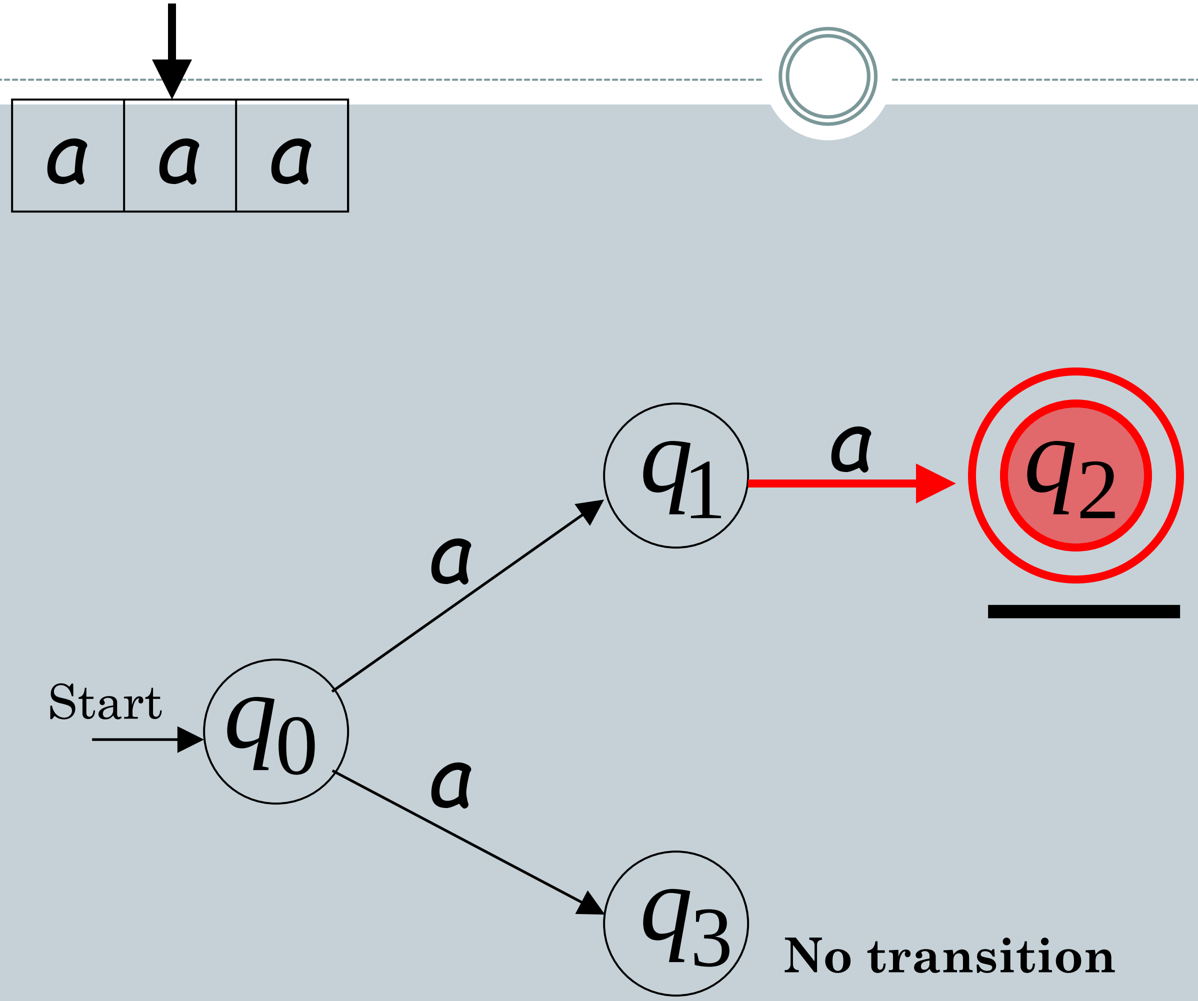
# Example: Rejecting



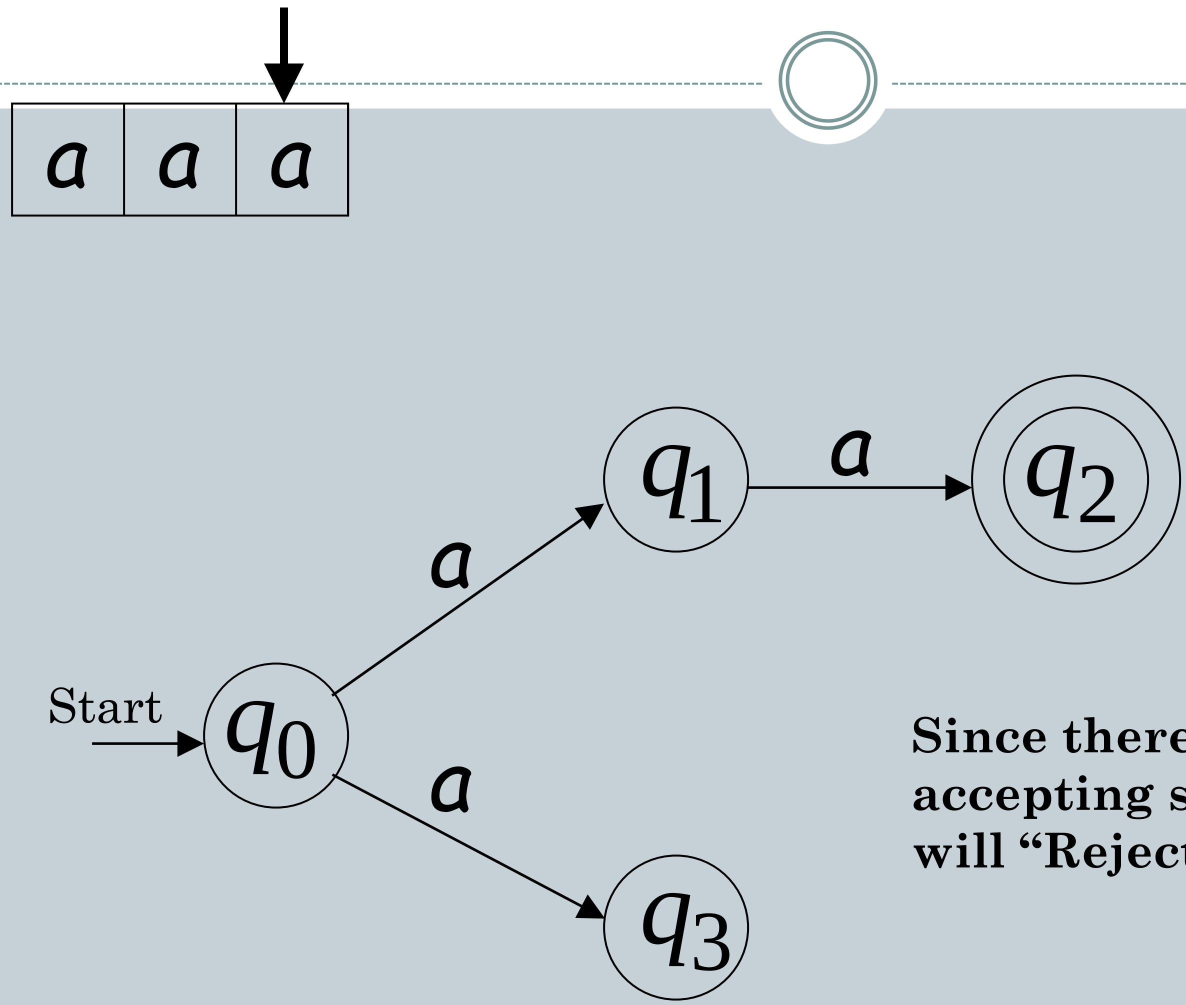
# Example: Rejecting



# Example: Rejecting



# Example: Rejecting

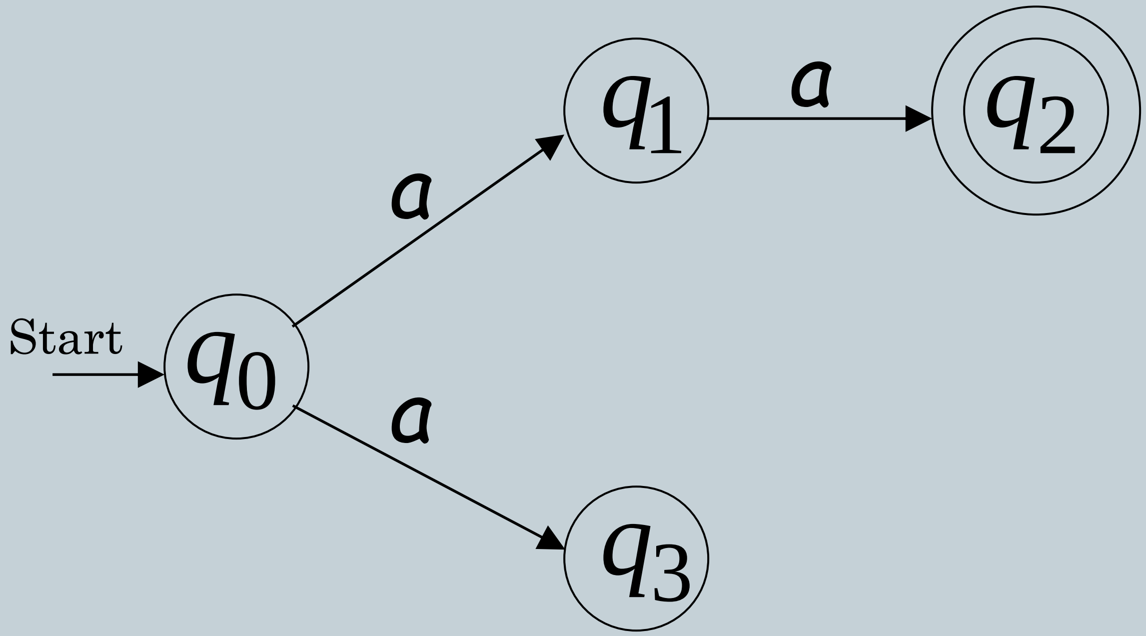



**No transition so leave the state**

**Since there is no current state or accepting state therefore, NFA will "Reject" the input string.**

# Language of the NFA

$$L = \{aa\}$$



# Transition Graph (TG)



- Relaxed input conditions
- Can read multiple characters before making a transition
- Thus every edge can be labeled with a substring instead of a single character
- Can have multiple start states

# Transition Graph (TG)



- **Definition:-** A transition graph is a combination of the following:
  - Finite number of states, at least one of which is start state and some (may be none) final states.
  - Finite set of input letters (Σ) from which input strings are formed.
  - Finite set of transitions that show how to go from one state to another based on reading specified substrings of input letters, possibly even the null string.

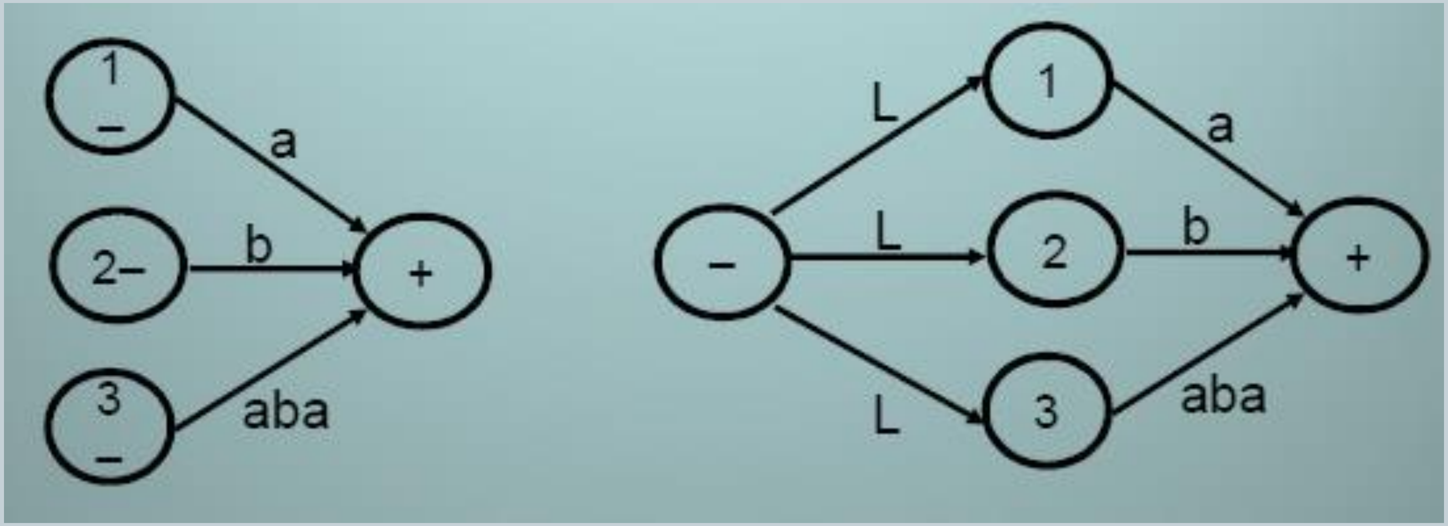
# Transition Graph (TG)



- It is to be noted that in TG there may exist more than 1 paths for certain string, while there may not exist any path for certain string as well. If there exists at least one path for a certain string, starting from initial state and ending in final state, the string is supposed to be accepted by the TG, otherwise the string is supposed to be rejected. Obviously, collection of accepted strings is the language accepted by the TG.

# Transition Graph (TG)

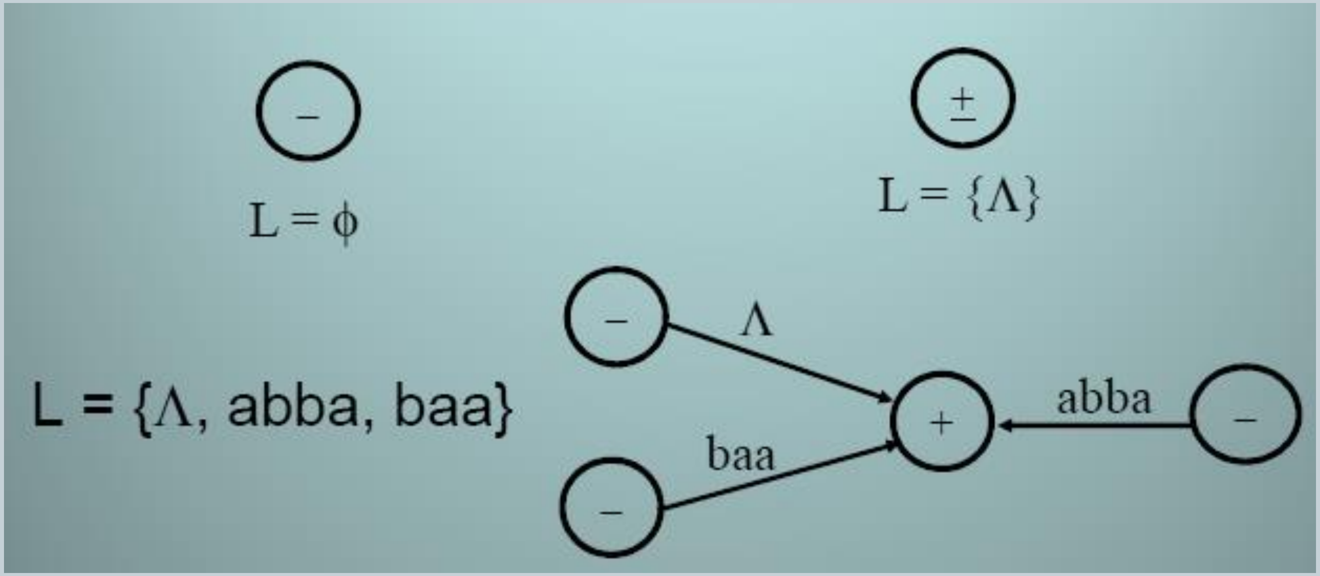





- Note:
  - These two machines are clearly equivalent.
  - **Remark:** *Every finite automaton is a transition graph.*

# Transition Graph (TG)

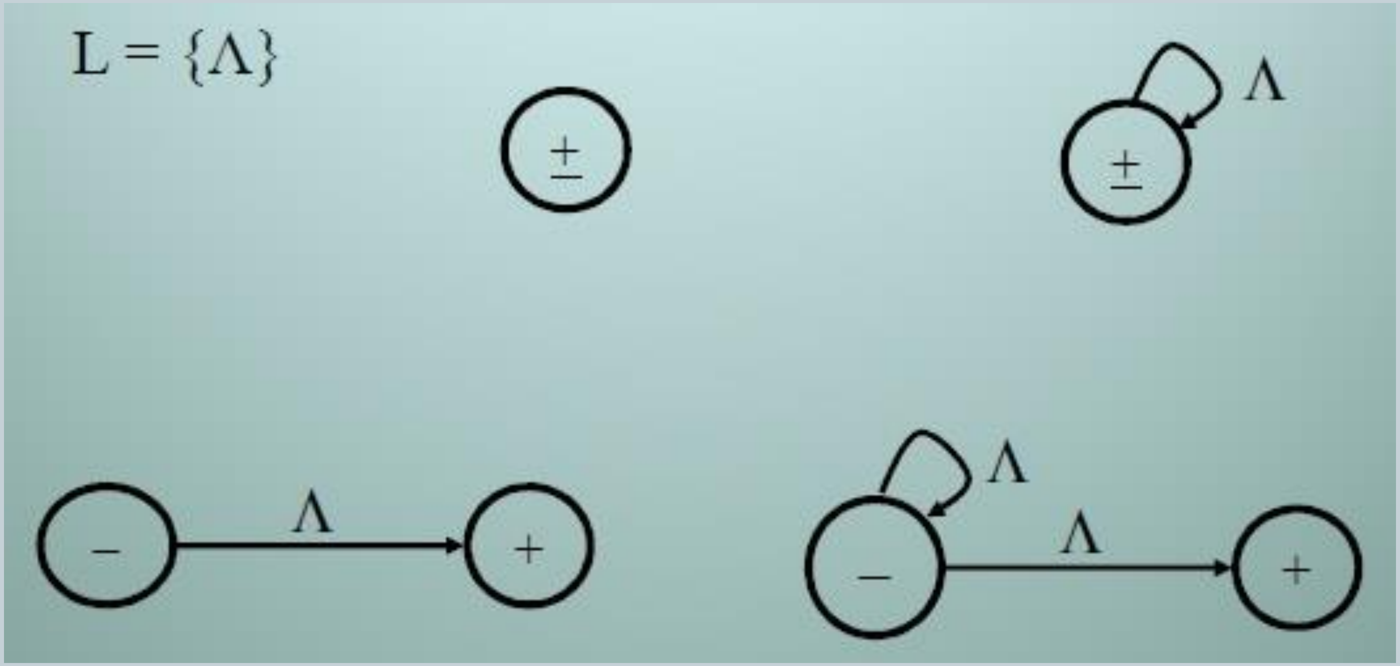





- Note: ε-Transitions is a null transition that changes state but doesn't consume any character
  - Possible with NFAs and Transition Graphs (discussed next)

# Transition Graph (TG)





# Transition Graph (TG)



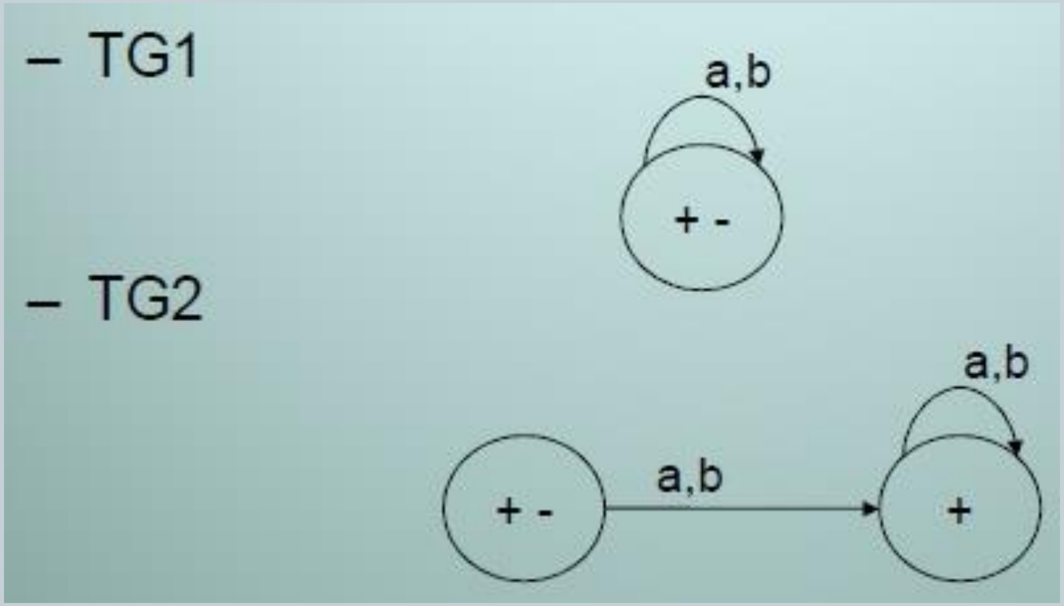
- Example:
  - Consider the language L of strings, defined over Σ = {a, b}, of all strings including null. The language L may be accepted by the following TG.


  - There are more than one paths for aa.
    - 1st path aa keeps at initial state. 2nd path aΛa takes to the final state. 3rd path aa from final state.

# Transition Graph (TG)



- Do we have more than 1 TG for a language?

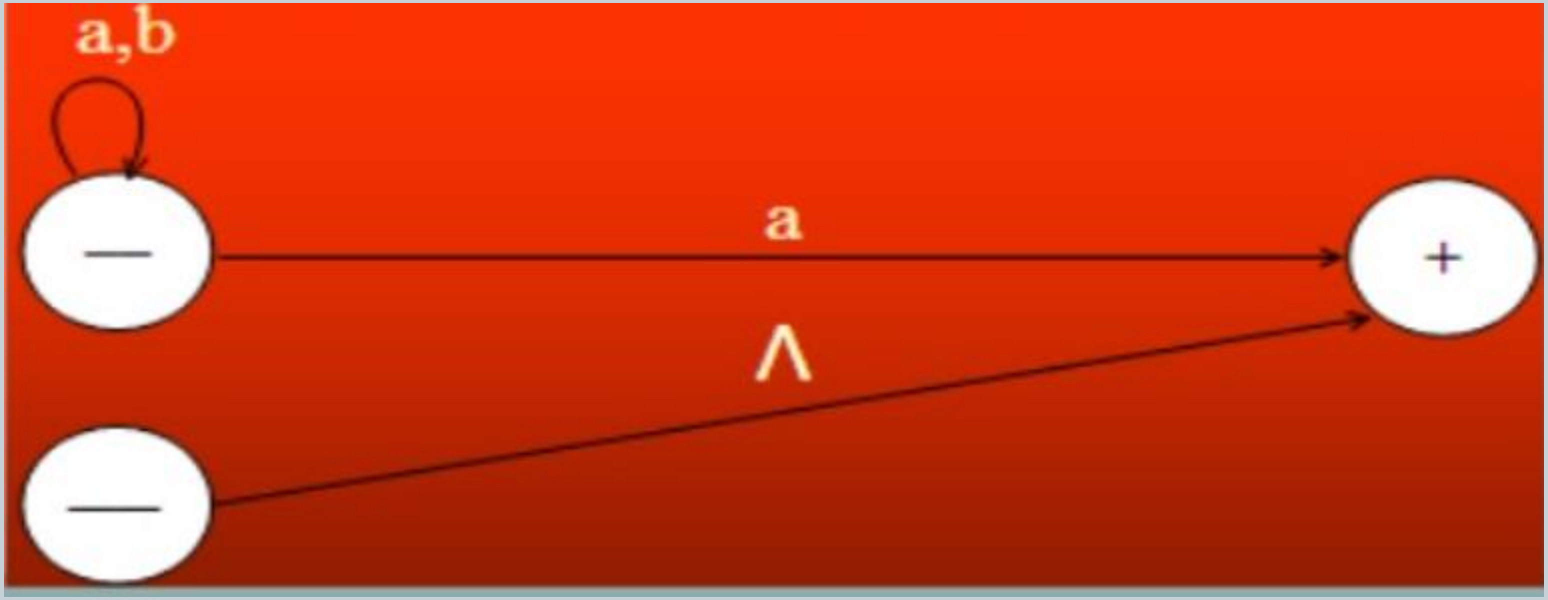



  - **Note:** We have got some relaxations in TGs. We are not bound to follow these relaxations, but we can use these relaxations.

# Transition Graph (TG)



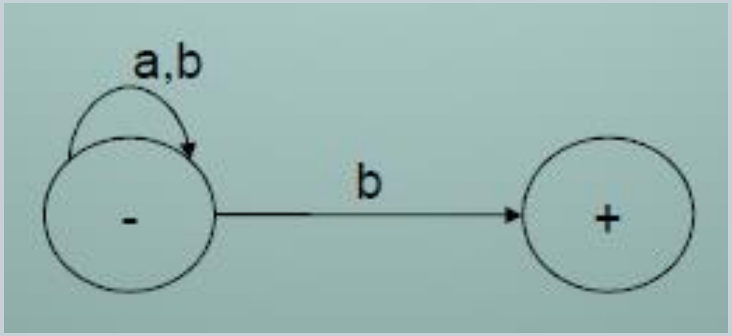
- Example:
  - Consider the language L of strings, defined over Σ = {a, b}, not ending in b. The language L may be expressed by the RE ʌ + (a+b)* a, may be accepted by the following TG.



# Transition Graph (TG)



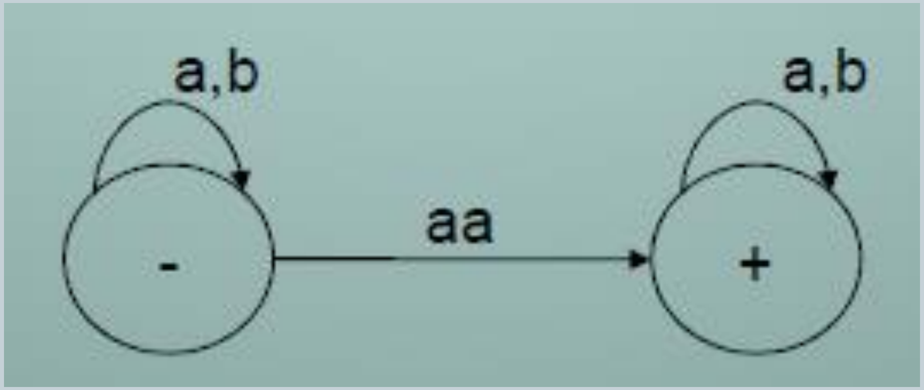
- Example:
  - Build a TG accepting the language of strings, defined over Σ = {a, b}, ending in b.
  - The language L may be expressed by the RE (a+b)* b, may be accepted by the following TG.



# Transition Graph (TG)



- Example:
  - Consider the language L of strings, defined over Σ = {a, b}, containing double aa.
  - The language L may be expressed by the RE (a+b)* aa (a+b)*, may be accepted by the following TG.



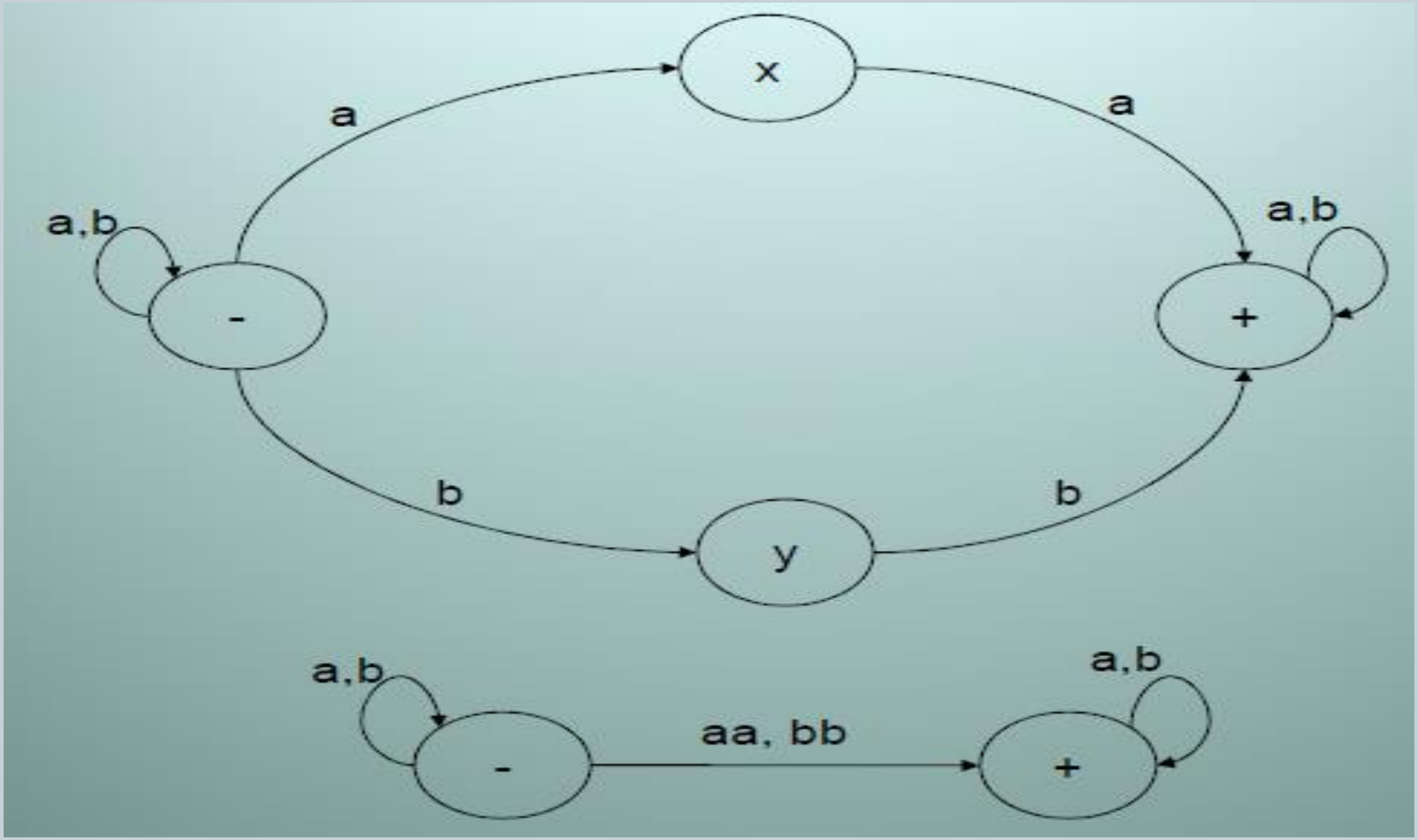
# Transition Graph (TG)



- Example:
  - Consider the language L of strings, defined over Σ = {a, b}, having double aa or doubld bb.
  - The language L may be expressed by the Regular Expression (a+b)* (aa+bb) (a+b)*.
  - The language L may be accepted by the following TG.

# Transition Graph (TG)

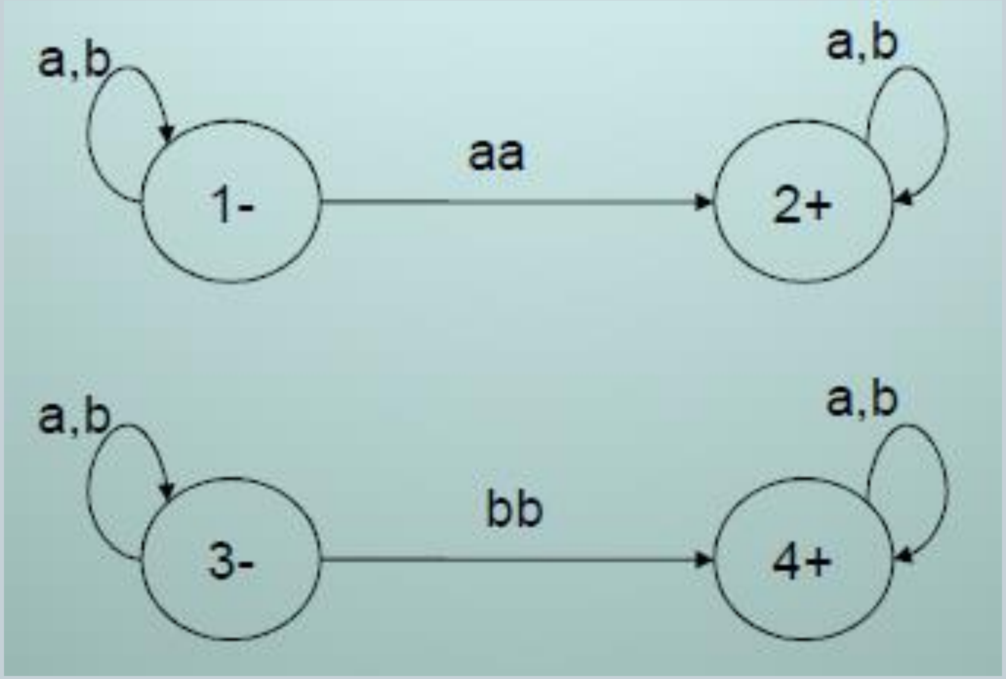




# Transition Graph (TG)



- OR

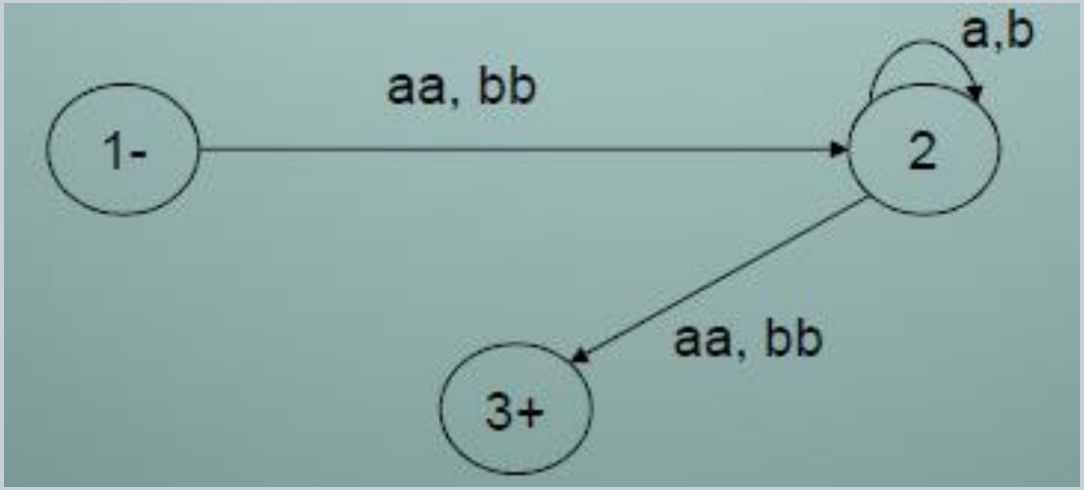



- Note: In the above TG if the states are not labeled then it may not be considered to be a single TG.

# Transition Graph (TG)



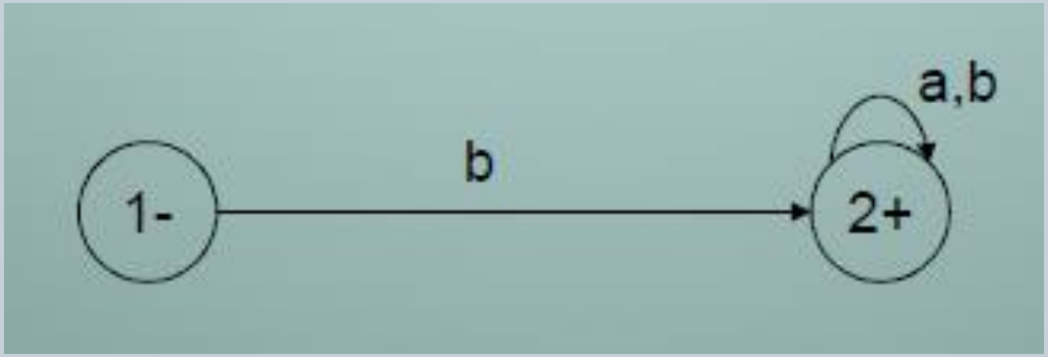
- Example:
  - Consider the language L of strings, defined over Σ = {a, b}, that start and end in double letter .
  - The language L may be expressed by the RE aa(a+b)*aa + aa(a+b)*bb + bb(a+b)*aa + bb(a+b)*bb.
  - The language L may be accepted by the following TG.



# Transition Graph (TG)



- Example:
  - Consider the language L of strings, defined over Σ = {a, b}, start with b .
  - The language L may be expressed by the RE b(a+b)*.
  - The language L may be accepted by the following TG.



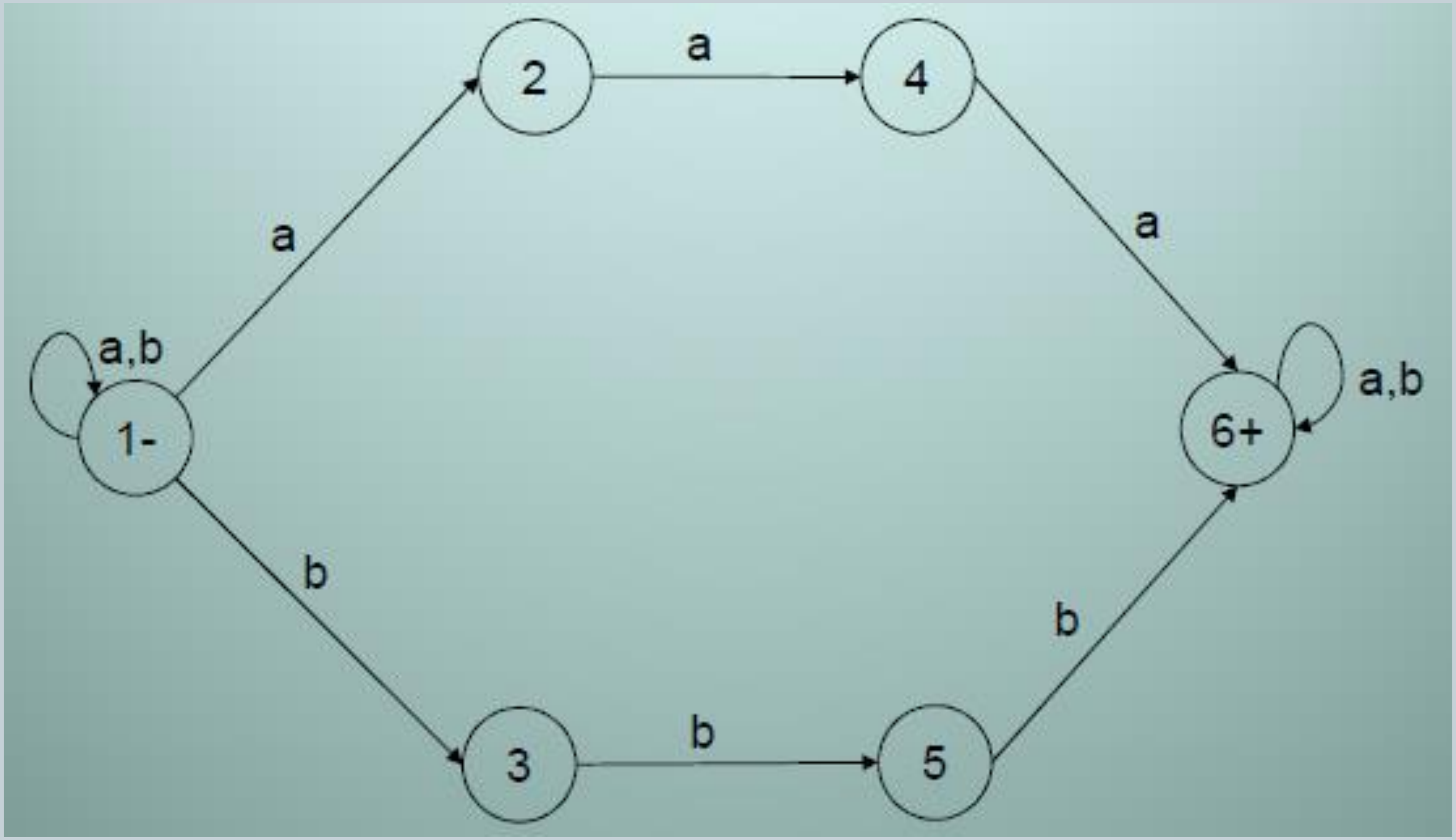
# Transition Graph (TG)



- Example:
  - Consider the language L of strings, defined over $\Sigma = \{a, b\}$, containing aaa, bbb .
  - The language L may be expressed by the RE $(a+b)^*(aaa+bbb)(a+b)^*$.
  - The language L may be accepted by the following TG.

# Transition Graph (TG)

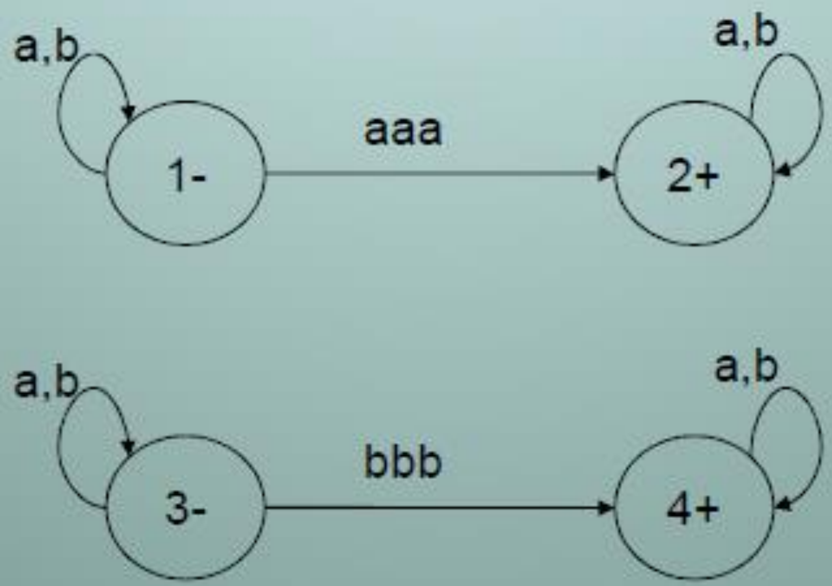




# Transition Graph (TG)




- OR



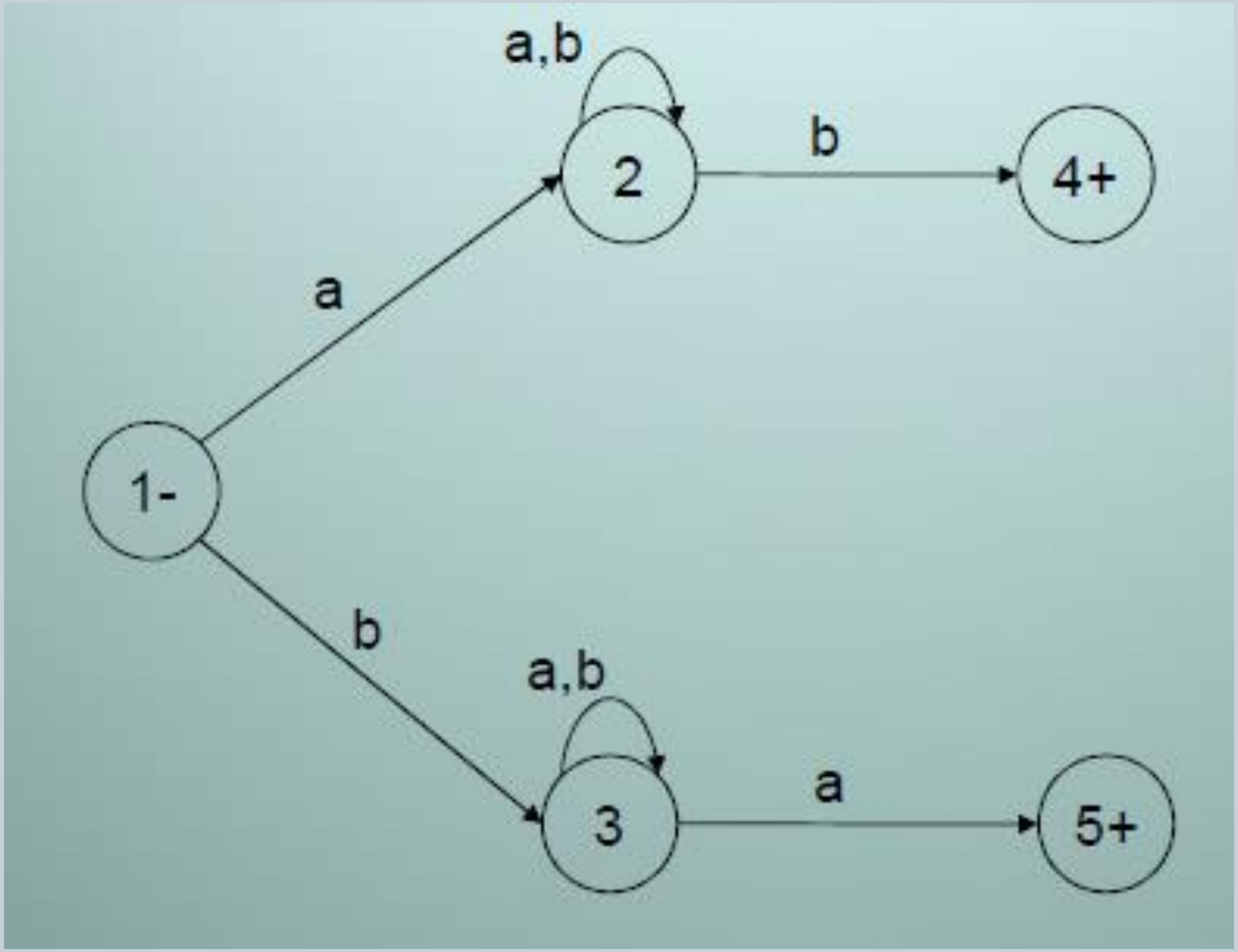
# Transition Graph (TG)



- Example:
  - Consider the language L of strings, defined over $\Sigma = \{a, b\}$, beginning and ending in different letters .
  - The language L may be expressed by the regular expression $a(a+b)^*b + b(a+b)^*a$.
  - The language L may be accepted by the following Transition Graph.

# Transition Graph (TG)





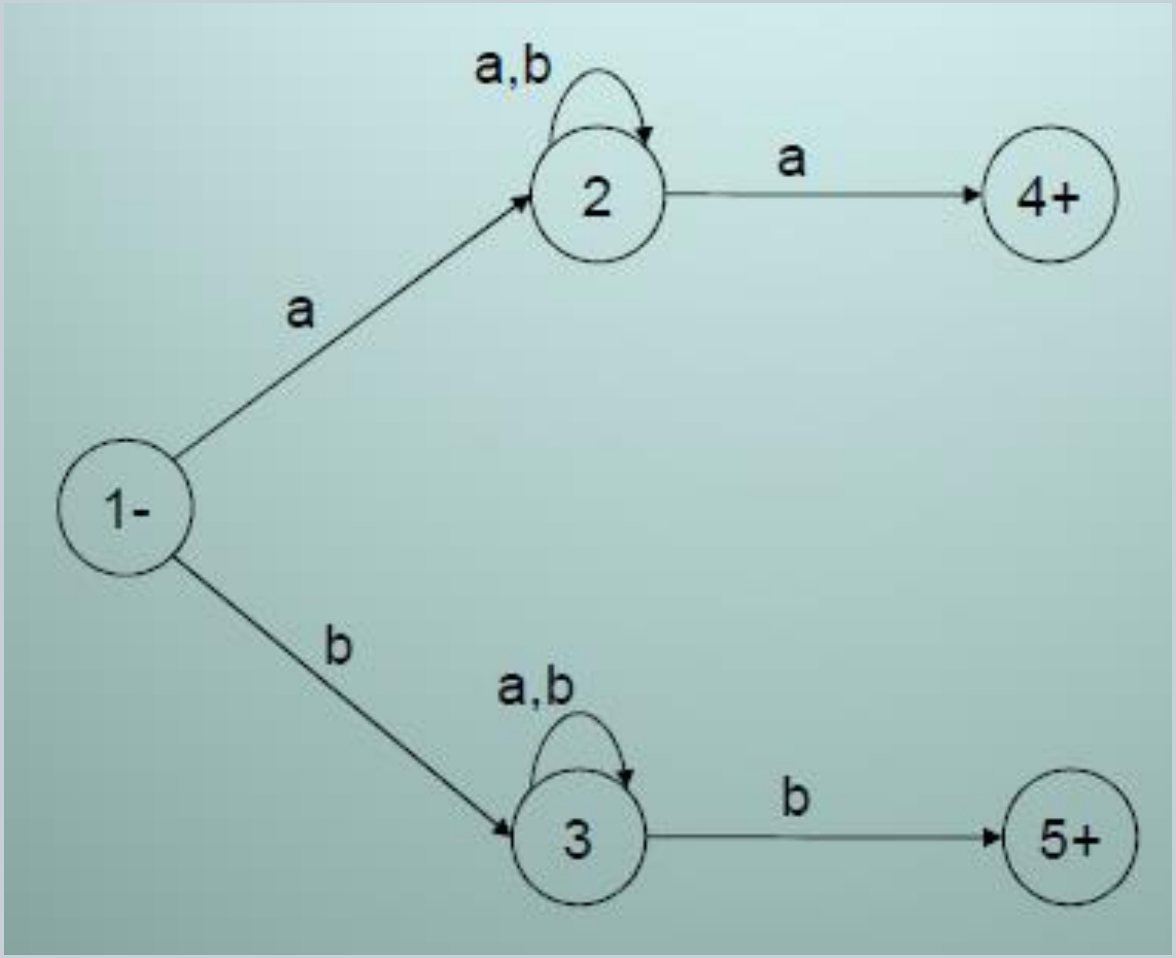
# Transition Graph (TG)



- Example:
  - Consider the language L of strings of length two or more, defined over Σ = {a, b}, beginning and ending in same letters .
  - The language L may be expressed by the regular expression a(a+b)*a+ b(a+b)*b.
  - The language L may be accepted by the following Transition Graph.

# Transition Graph (TG)

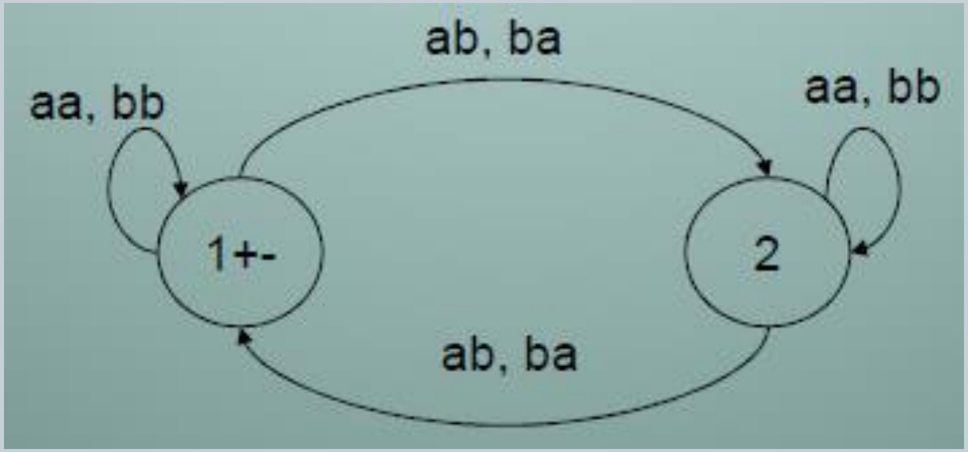




# Transition Graph (TG)



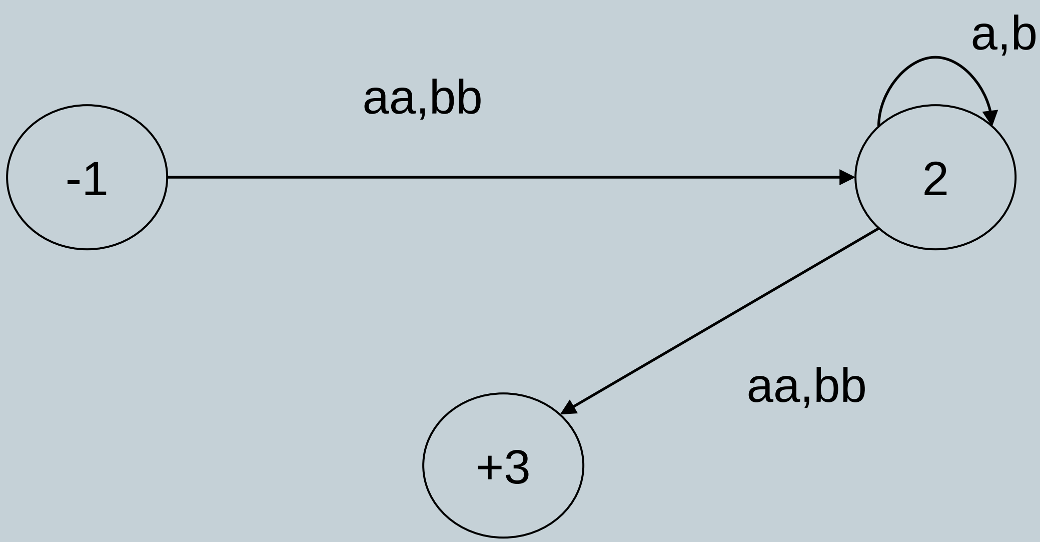
- Example:
  - Consider the EVEN-EVEN language, defined over Σ = {a, b}.
  - The language may be expressed by the following RE (aa+bb+(ab +ba) (aa+bb)*(ab+ba)) *
  - The language L may be accepted by the following TG



# Transition Graph (TG)



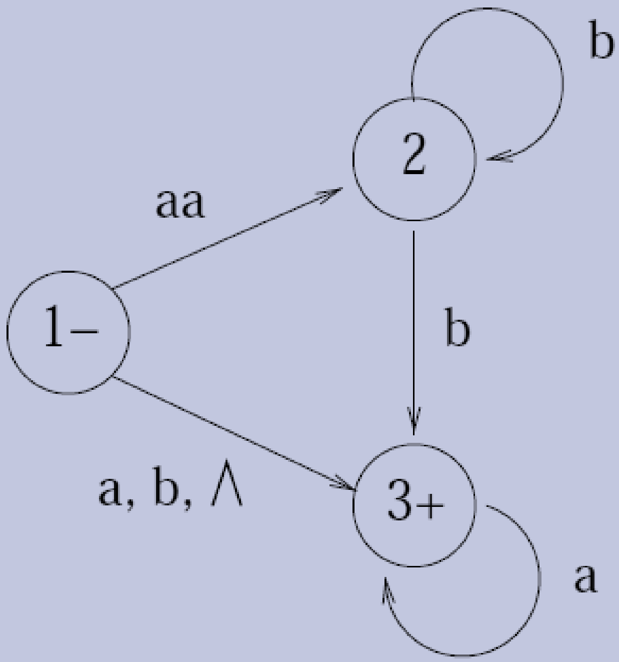
- Examples
  - All words that start and end with a double letter



# Transition Graph (TG)



**Example:** Consider the following machine that processes strings over the alphabet $\Sigma = \{a, b\}$:




Note that this machine is not a finite automaton:

# Transition Graph (TG)

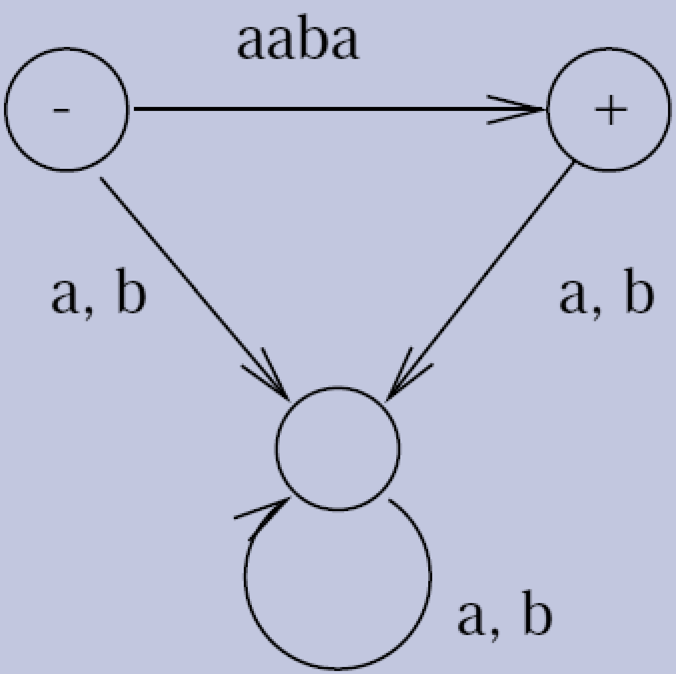


- The arc from state 1 to state 2 is labeled with the string aa, which is not a single letter.
- There are two arcs leaving state 2 labeled with b.
- There is no arc leaving state 2 labeled with a.
- There is an arc from state 1 to state 3 labeled with , which is not a letter from .
- There is no arc leaving state 3 labeled with b.

# Transition Graph (TG)



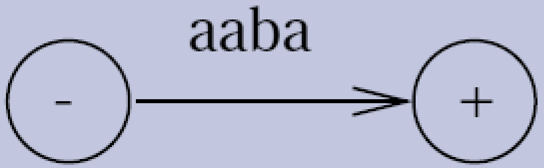
**Example:** Only accepts the word *aaba*



# Transition Graph (TG)



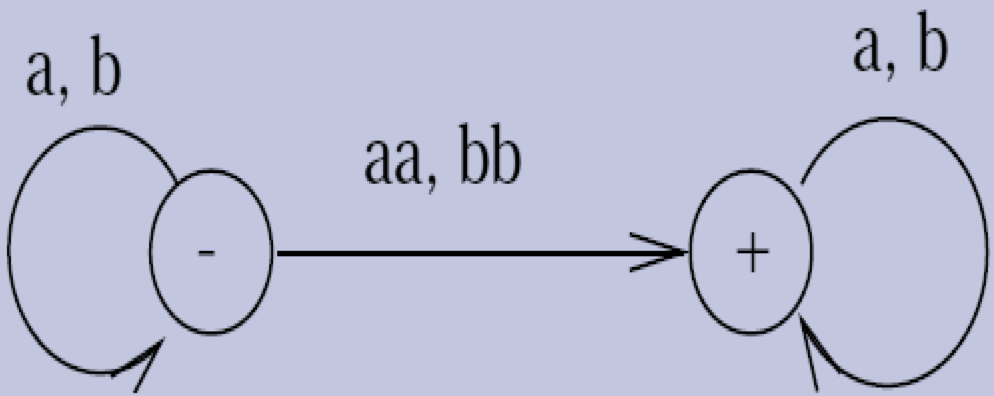
or



# Transition Graph (TG)



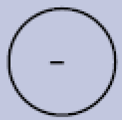
Example: Accepts all words that contain a doubled letter.



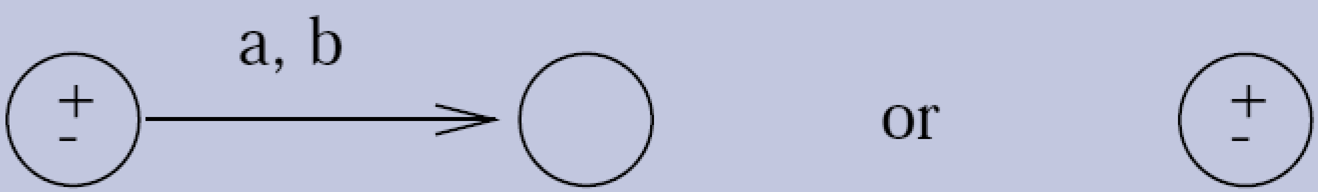
# Transition Graph (TG)



**Example:** this TG accepts nothing, not even Λ.



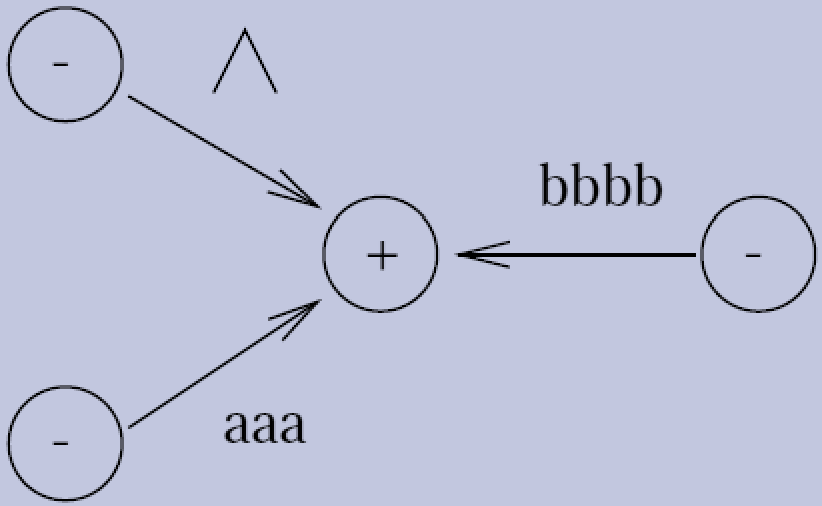

**Example:** this TG accepts only the string Λ.



# Transition Graph (TG)



**Example:** This TG accepts only the words $\Lambda$, $aaa$ and $bbbb$.

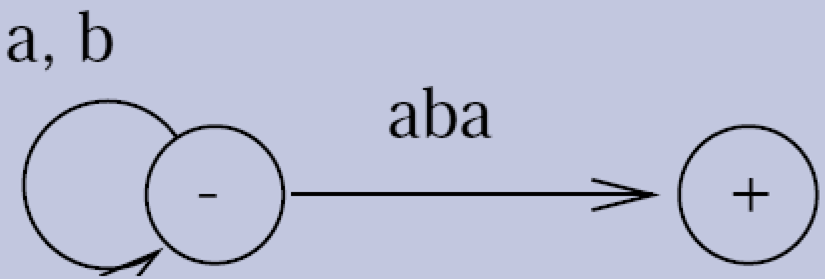


# Transition Graph (TG)



**Example:** this TG accepts only words that end in *aba*; i.e., the language generated by the regular expression

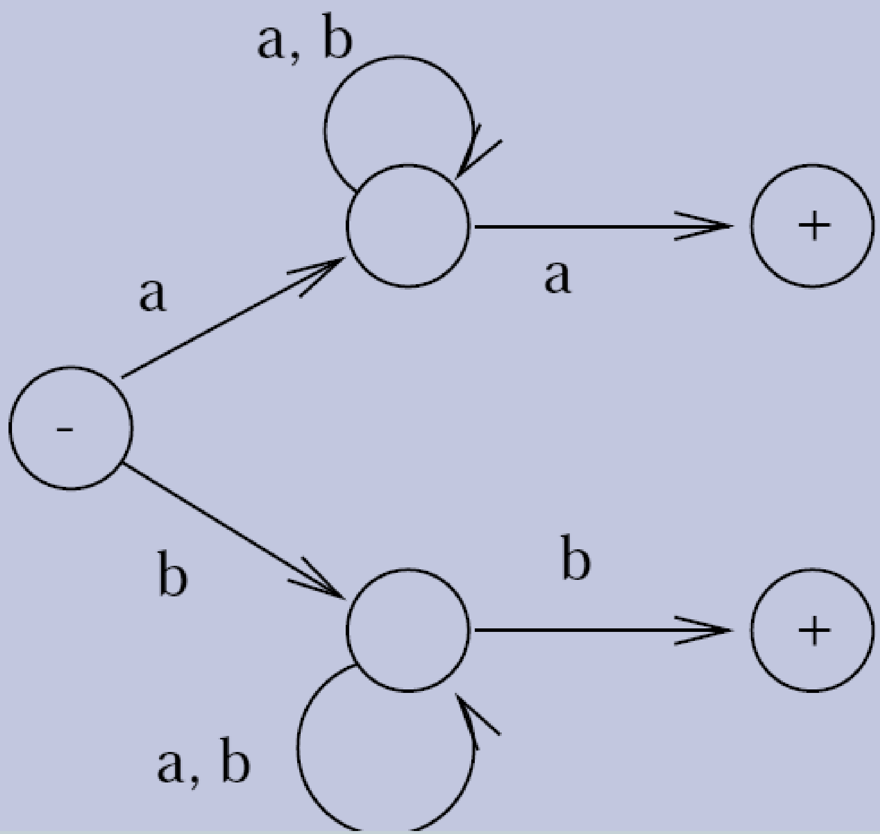
$$(\mathbf{a}+\mathbf{b})^*\mathbf{aba}$$



# Transition Graph (TG)



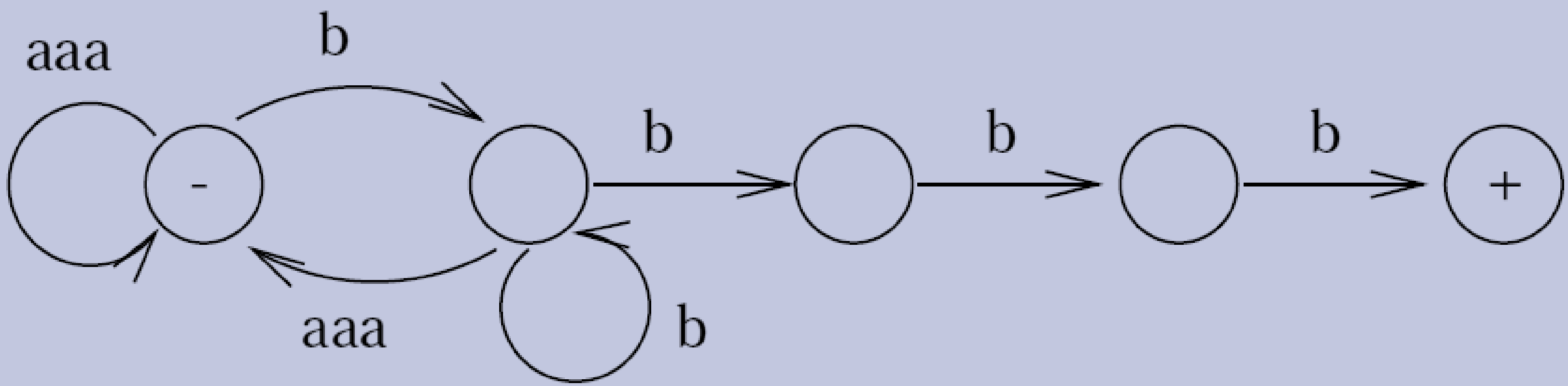
**Example:** this TG accepts the language of all words that begin and end with the same letter and have at least two letters.



# Transition Graph (TG)



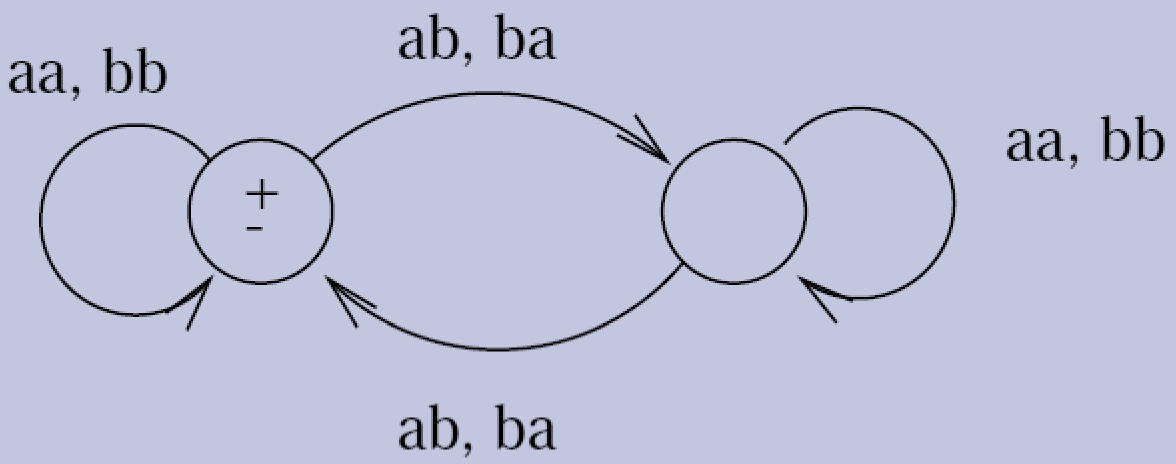
**Example:** this TG accepts the language of all words in which the $a$'s occur in clumps of three and that end in four or more $b$'s.



# Transition Graph (TG)



**Example:** this is the TG for EVEN-EVEN




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

**Example:** Is the word *baaabab* accepted by this machine?

# Transition Graph (TG)

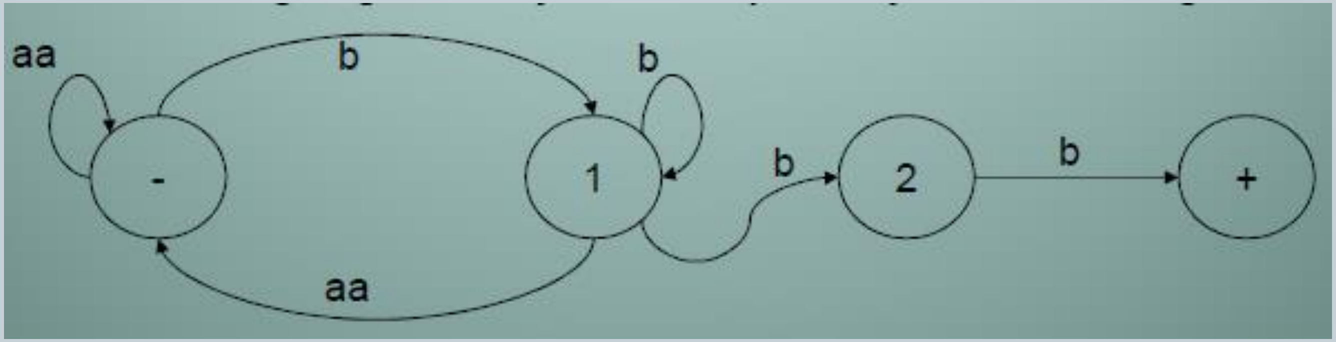


- Examples
  - All words that have al least one double letter in them
  - All words that begin and end with different letters
  - All words in which *a* occurs only in even clumps and that end in three or more *b*s
  - All words that have even number of letters

# Transition Graph (TG)



- Example:
  - Consider the language L, defined over Σ = {a, b}, in which ***a's occur only in even clumps and that ends in three or more b's.***
  - The language may be expressed by the following RE (aa)*b (b*+(aa(aa)*b)*) bb
  - The language L may be accepted by the following TG



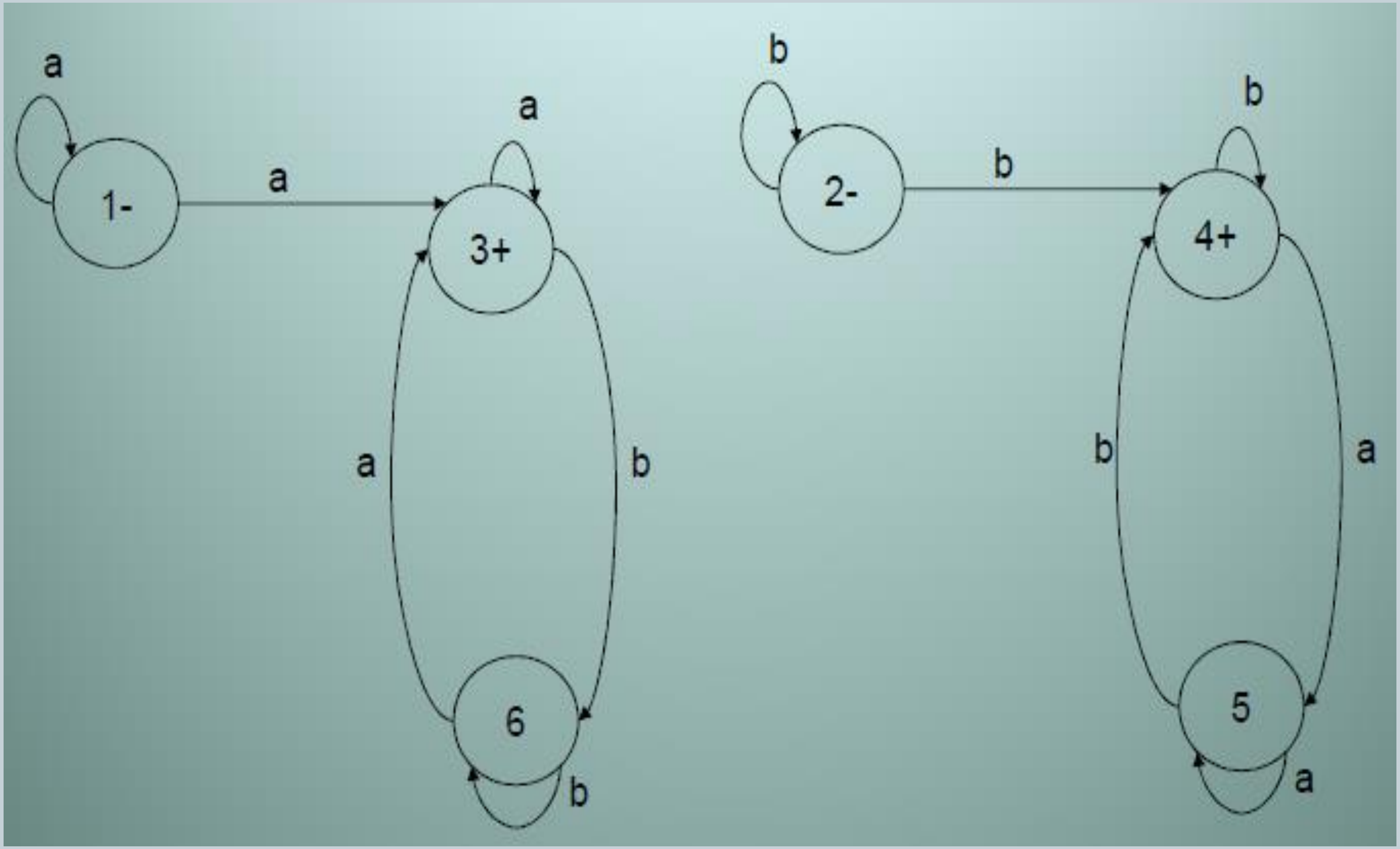
# Transition Graph (TG)



- Example:
  - Build a TG accepting the language L of strings, defined over Σ = {a, b}, beginning with and ending in the same letters .
  - The language may be expressed by the following regular expression a + b + a(a+b)*a + b(a+b)*b
  - The language L may be accepted by the following TG

# Transition Graph (TG)

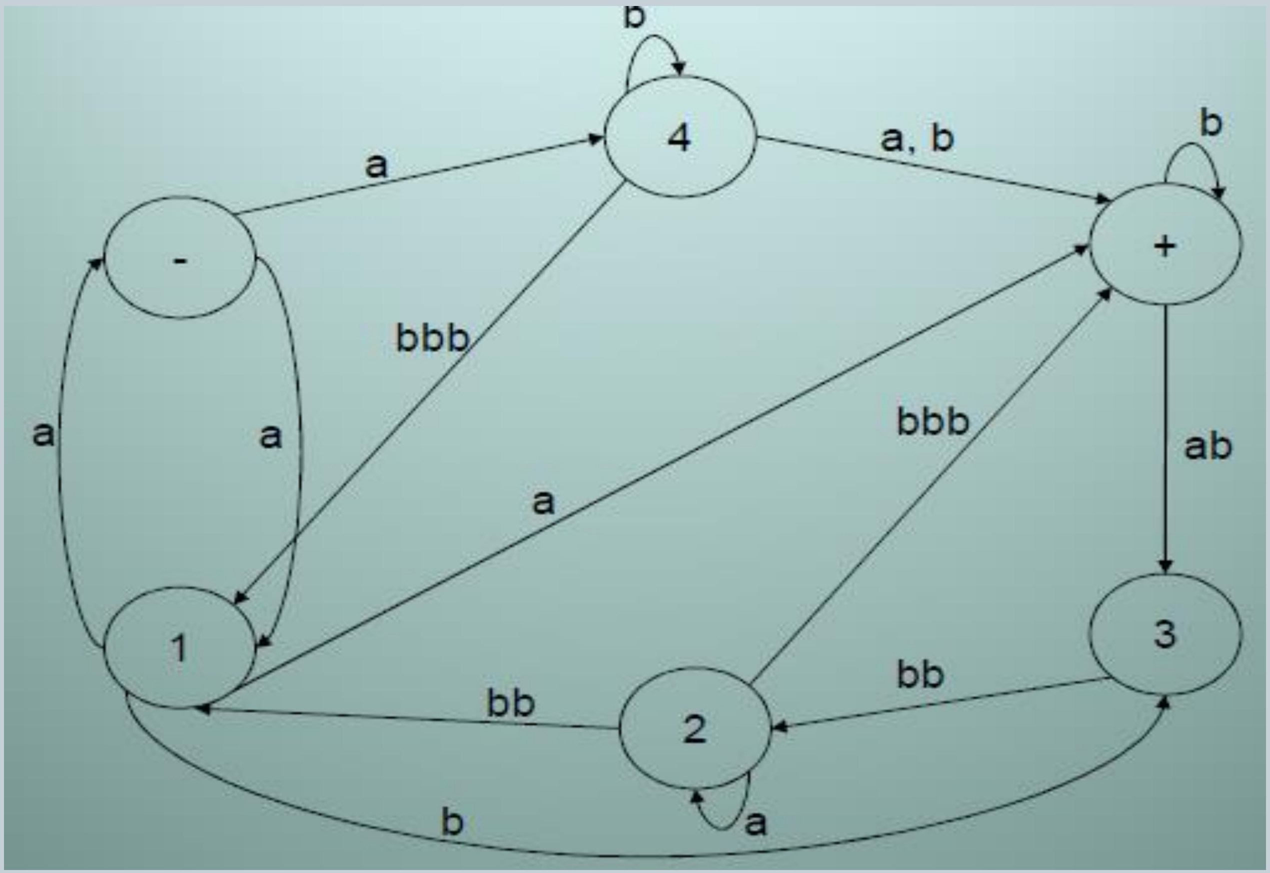




# Transition Graph (TG)



- Example: Consider the following TG



# Transition Graph (TG)



- Consider the string abbbabbbabba. It may be observed that the above string traces the following three paths:
- (a) (b) (b) (b) (ab) (bb) (a) (bb) (a)
  (-) (4) (4) (+) (+) (3) (2) (2) (1) (+)
- (a) (b) ((b)(b)) (ab) (bb) (a) (bb) (a)
  (-) (4) (+) (+) (+) (3) (2) (2) (1) (+)
- (a) ((b)(b)) (b) (ab) (bb) (a) (bb) (a)
  (-) (4) (4) (4) (+) (3) (2) (2) (1) (+)