Lecture 5

Converting Regular Expressions into Finite Automata

Assoc. Prof. Osama fathy

Overview

- Questions
- Kleene’s Theorem
- Consequences
- Convert Regular Expressions to NFA-Λ
- Convert NFA-Λ to FA

Questions

- Can every language which is represented by a **regular expression** be described by a **finite automaton**?
- Can every language which is described by a **finite automaton** be represented by a **regular expression**?
- Can every language be represented by a **regular expression** or a **finite automaton**?

Kleene's Theorem

Any language which can be defined by

- **Regular Expressions**
- **Finite Automaton**
- **Nondeterministic Finite Automaton (NFA)**
- **NFA-Λ**
- **Transition Graphs**
- **Generalised Transistion Graphs**

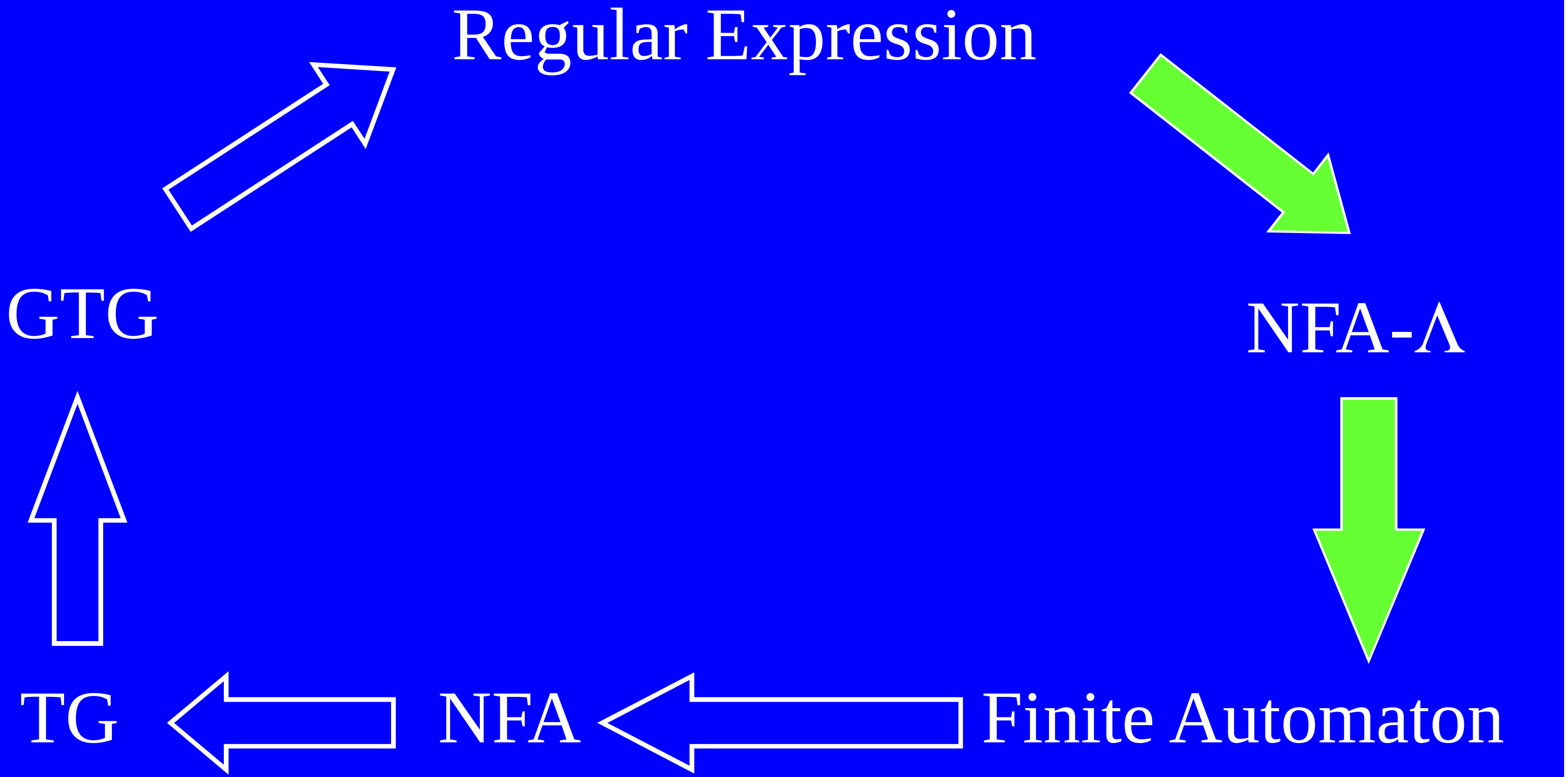
can be defined by any of the other methods

The Complement of Regular Language is a Regular Language

Outline of Proof:

- Suppose we have a Regular Language.
- Therefore we have a regular expression that defines the language.
- So, by Kleene’s Theorem, there is a FA that defines this language.
- We can convert this FA into one that defines the complement the language.
- So, by Kleene’s Theorem, there is a regular expression that defines the complement.

Kleene’s Theorem



Regular Expression

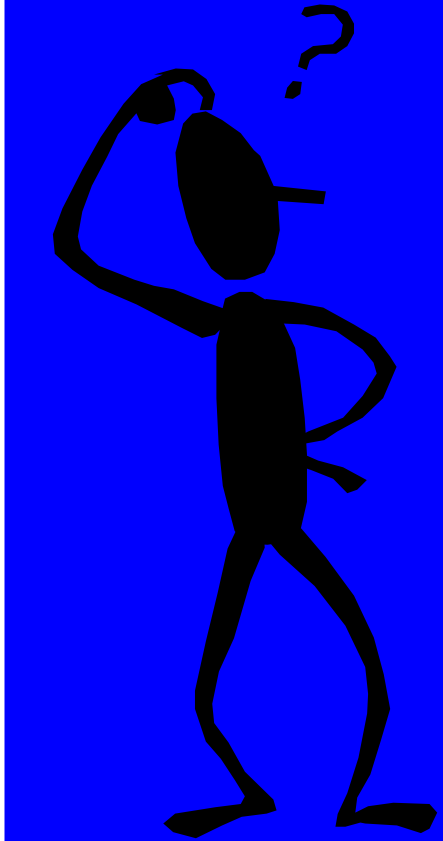
NFA-Λ

Finite Automaton

NFA

TG

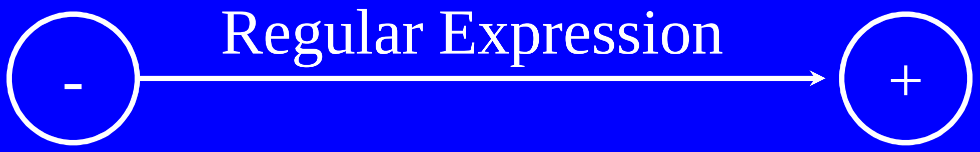
GTG



How to convert a Regular Expression into a NFA-Λ

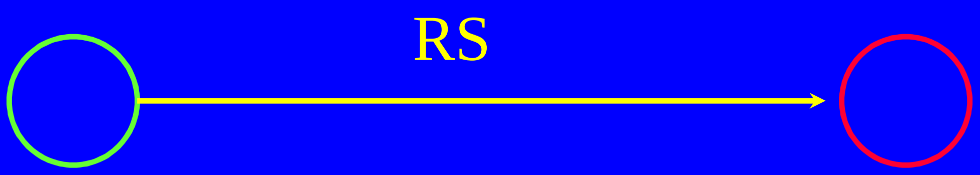
Converting Regular Expression to NFA-Λ

Start with the graph.

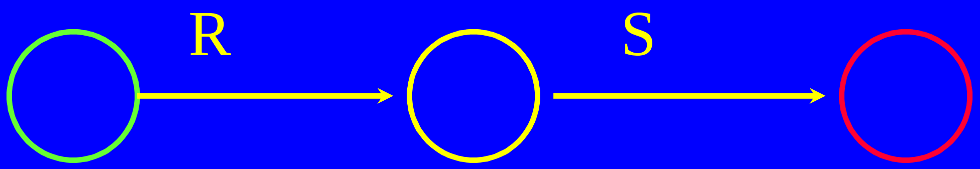


Apply the following rules until all edges are labeled with a letter or Λ.

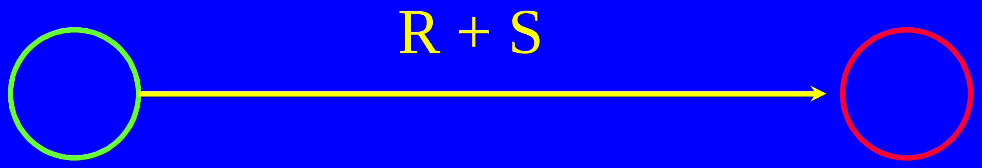
1. Delete any edge labeled with ϕ.
2. Transform any edge like



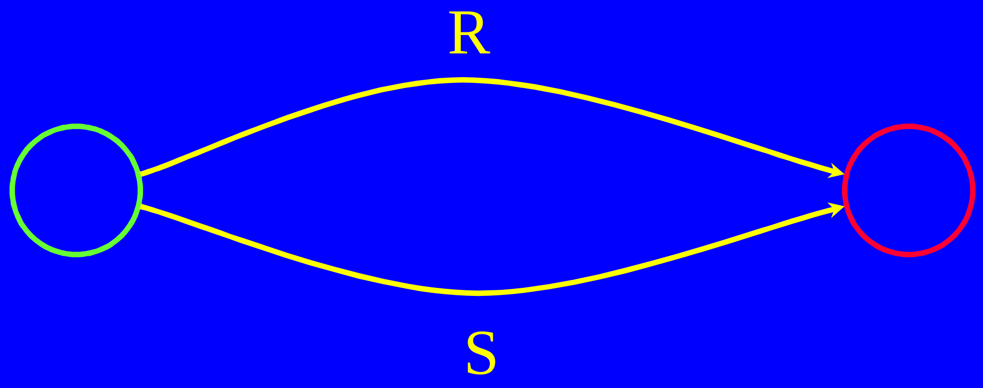
into



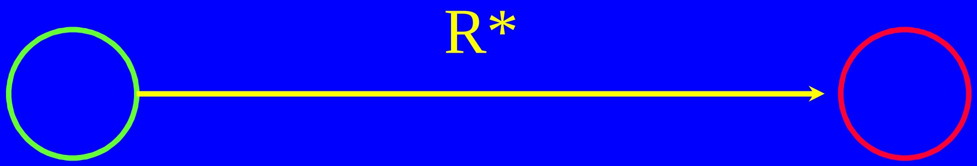
3. Transform any edge like:



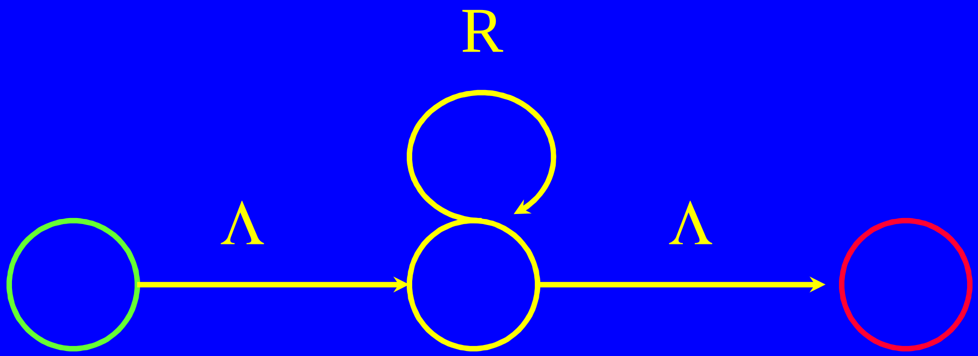
into



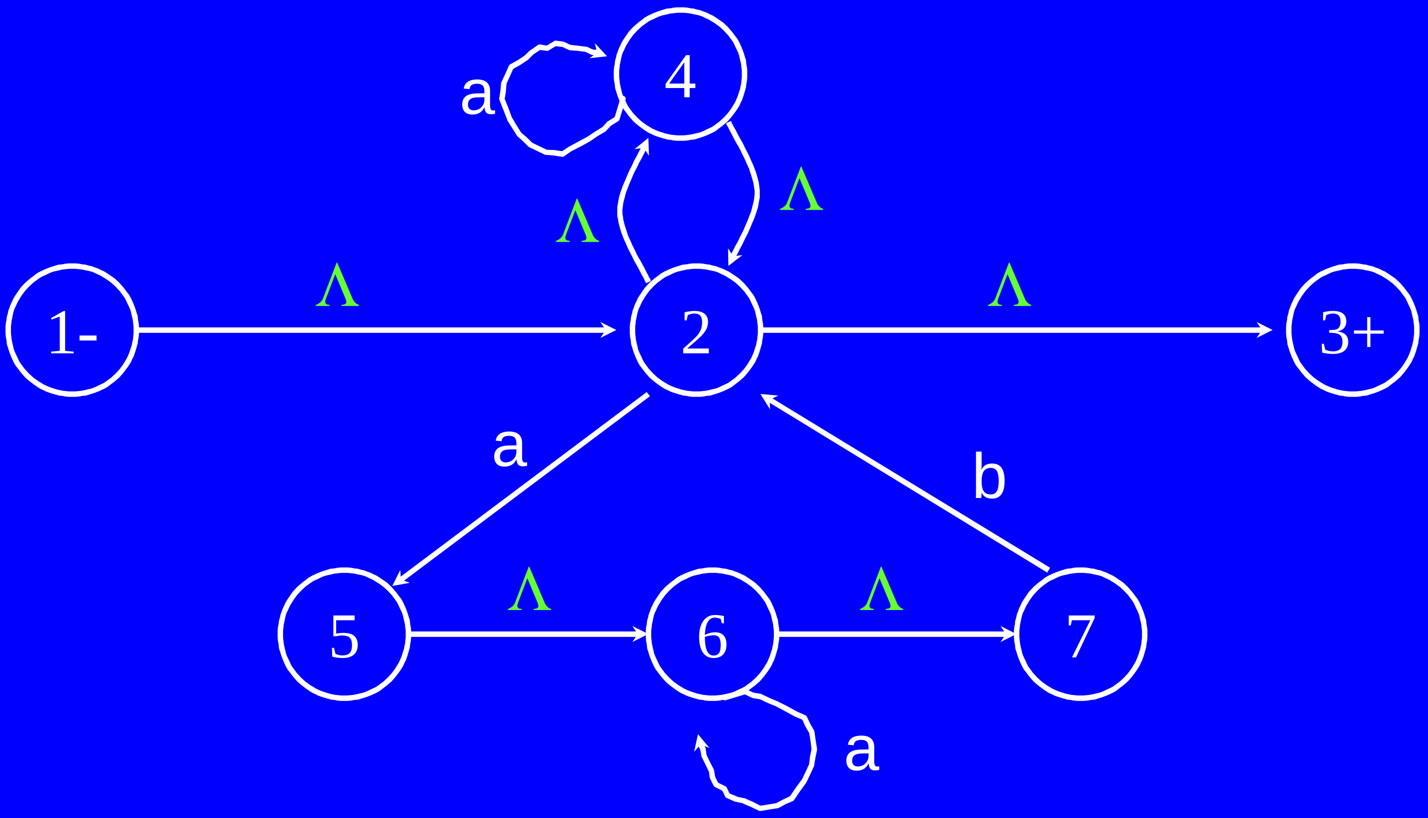
4. Transform any edge like:



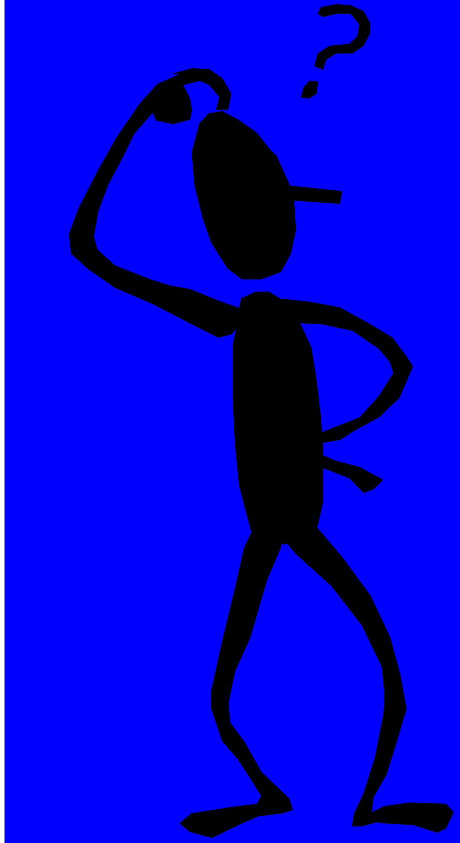
into



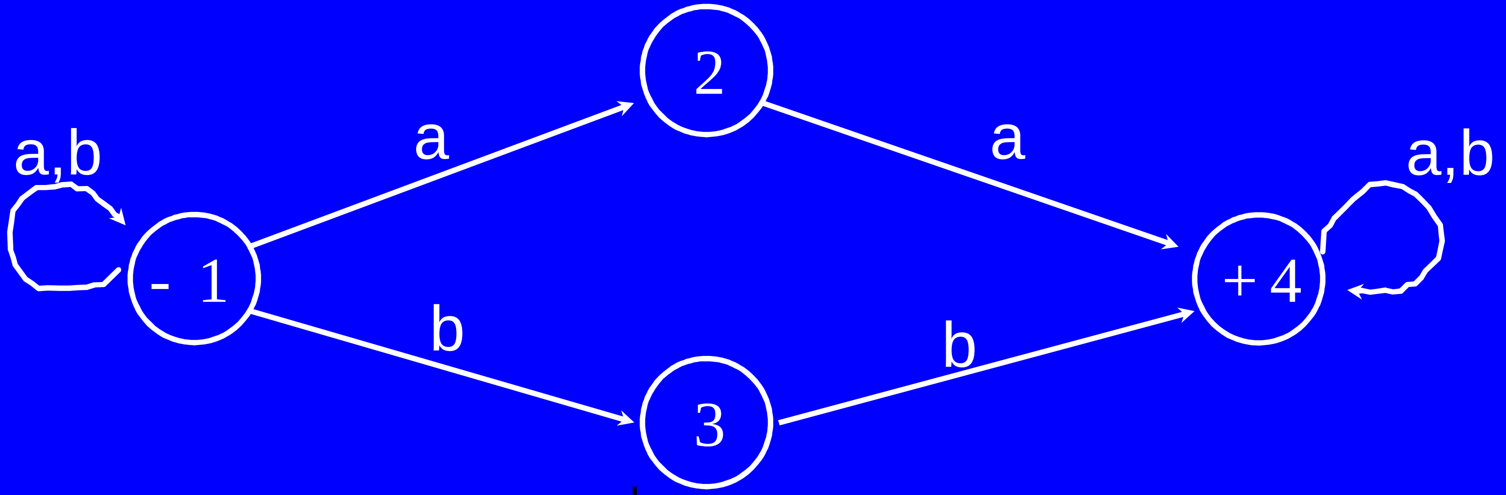
(a* + aa*b)*



How to convert a NFA-Λ into a FA

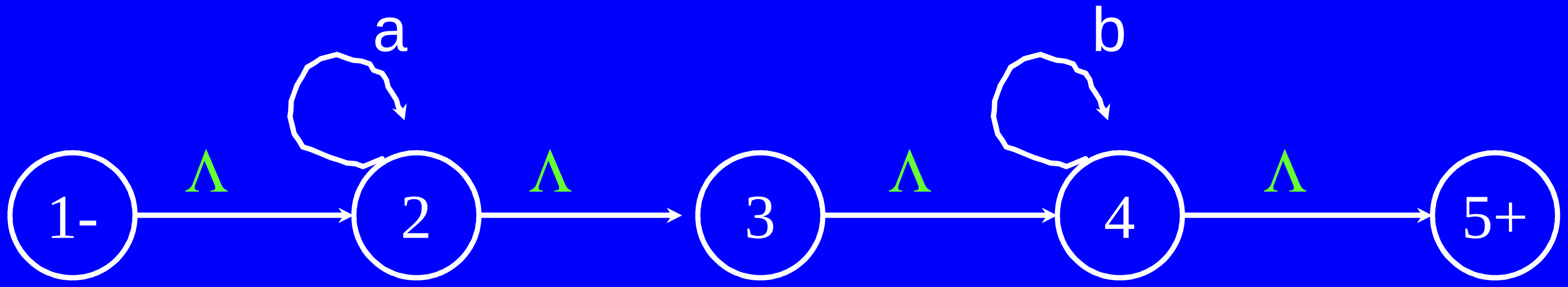


NFA ->DFA (EXAMPLE-1)



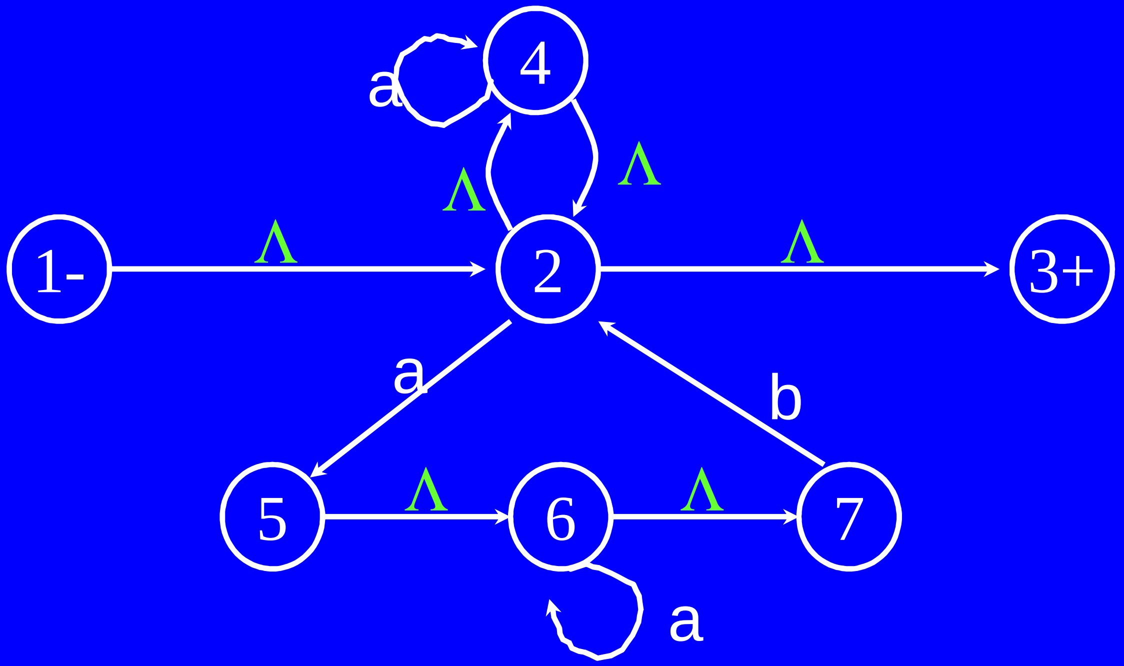
	a	b
Start {1}	{1,2}	{1,3}
{1,2}	{1,2,4}	{1,3}
{1,3}	{1,2}	{1,3,4}
Final {1,2,4}	{1,2,4}	{1,3,4}
Final {1,3,4}	{1,2,4}	{1,3,4}

NFA ->DFA (EXAMPLE-2)



		a	b
Start/Final	{1,2,3,4,5}	{2,3,4,5}	{4,5}
Final	{2,3,4,5}	{2,3,4,5}	{4,5}
Final	{4,5}	ϕ	{4,5}
	ϕ	ϕ	ϕ

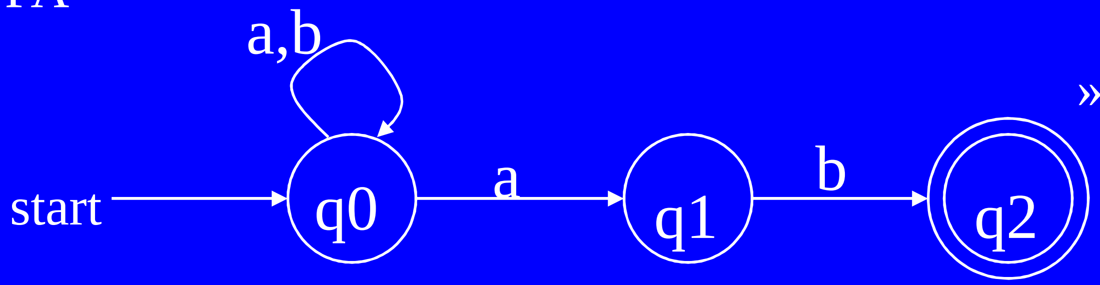
NFA ->DFA (EXAMPLE-3)



	a	b
Start /Final{1,2,3,4}	{5,4,6,7,2,3}	ϕ
Final {2,3,4,5,6,7}	{5,4,6,7,2,3}	{2,3,4}
Final {2,3,4}	{5,4,6,7,2,3}	ϕ
ϕ	ϕ	ϕ

NFA ->DFA (EXAMPLE-4)

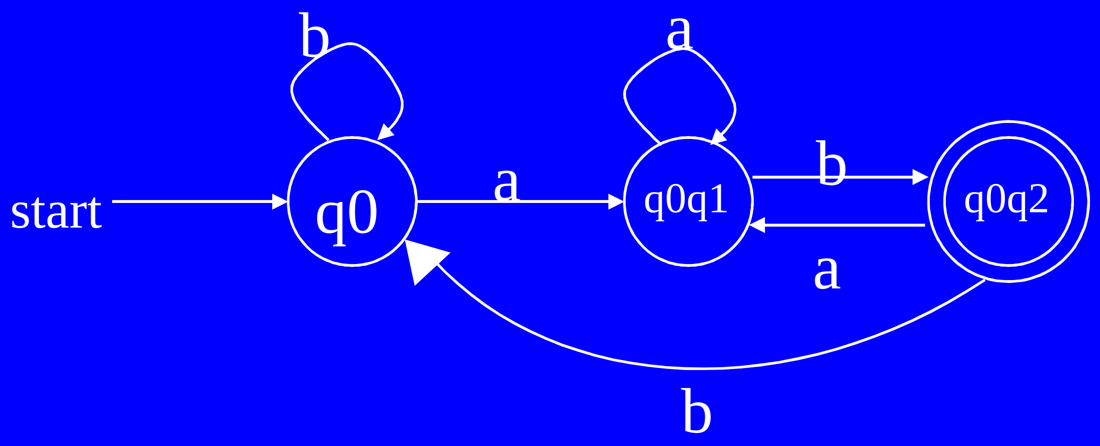
» NFA



» Transaction table

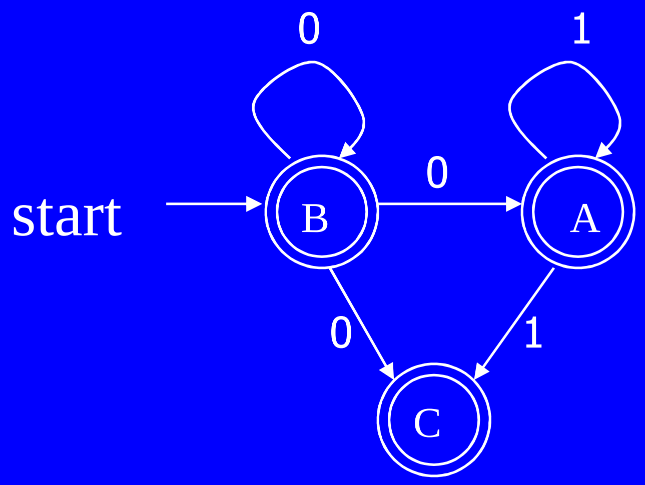
States/ input	**a**	**b**
q0	q0q1	q0
q1	ϕ	q2
q2	ϕ	ϕ
q0q1	q0q1	q0q2
q0q2	q1q2	q0

» Equivalent DFA



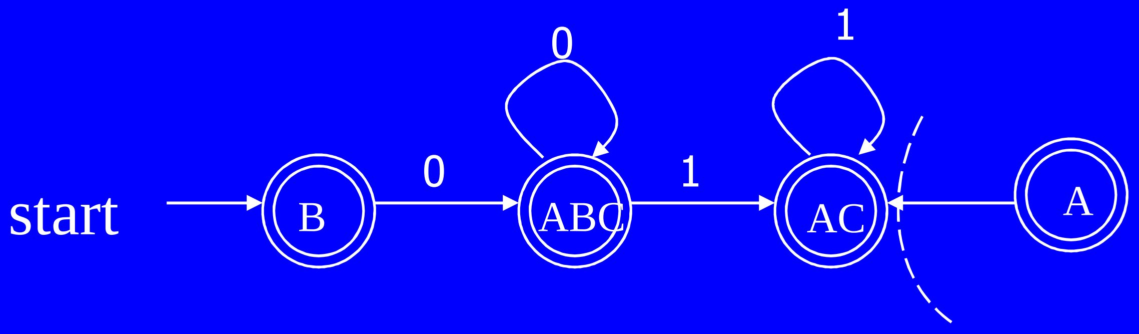
NFA ->DFA (EXAMPLE-5)

» NFA

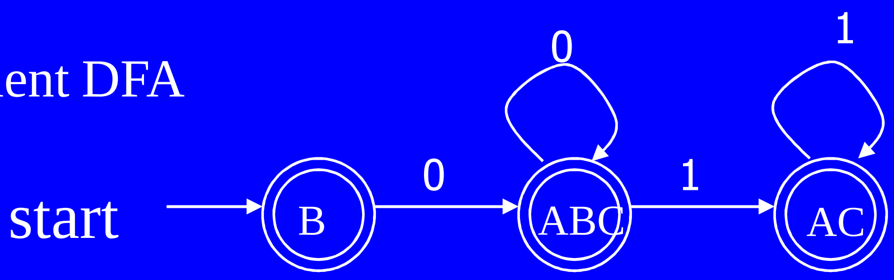


» Transaction table:

States/ input	0	1
B	BAC	ϕ
A	ϕ	AC
C	ϕ	ϕ
ABC	ABC	AC
AC	ϕ	AC



» Equivalent DFA



start
B
ABC
AC
0
0
1
1