Formal Languages & Automata Theory

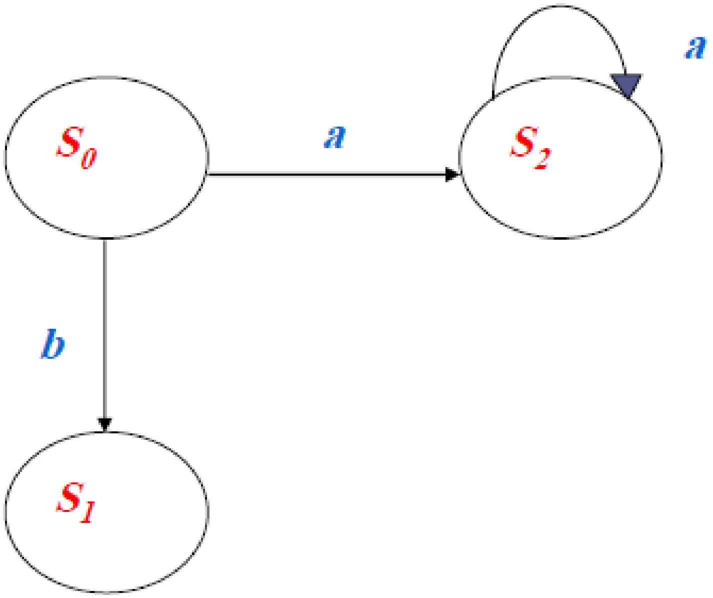
Lecture 3 :

Non-deterministic Finite Automata (FNA)

Presented by: Dr. Ebtsam Abdelhakam Sayed
Faculty of Computers and Information
Minia University

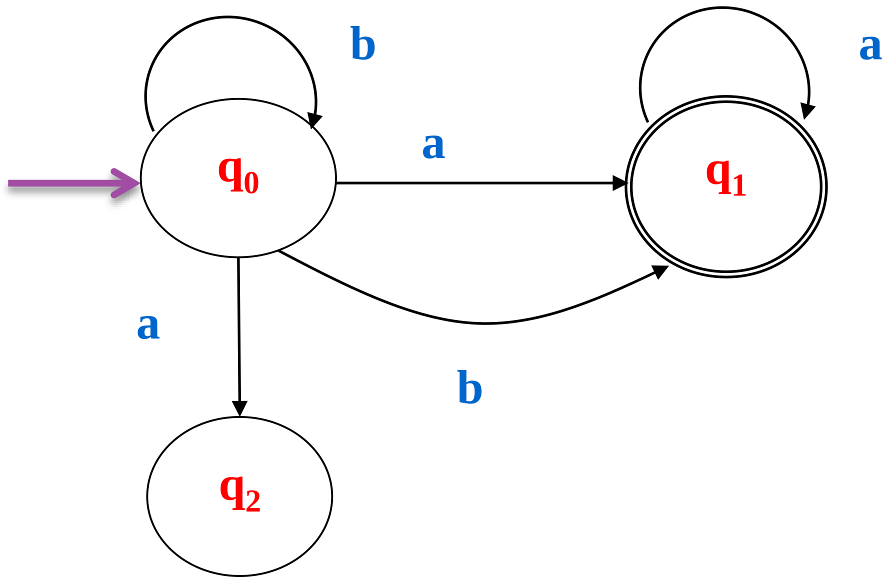
Deterministic finite Automata (DFA)

The finite Automata is called Deterministic. Finite Automata if there is only one path for a specific input from current state to next state. For example, the DFA can be shown as below.



From state S_0 for input ' a ' there is only one path, going to S_2. Similarly, from S_0 there is only one path for input b going to S_1.

Non-deterministic Finite Automata (FNA)



The concept of Non- deterministic finite Automata is exactly reverse of Deterministic Finite Automata. The Finite Automata is called NFA when there exist many paths for a specific input from current state to next state.

Note that the NFA shows from q_0 for input a there are two next states q_1 and q_2 similarly, from q_0 for input b the next states are q_0 and q_1. Thus, it is not fixed or determined that with a particular input where to go next Hence this FA is called non-deterministic finite automata.

Consider the string bba. This string can be derived as.

Input	b	b	a
Path	q_0	q_0	q_1

or

Input	b	b	a
Path	q_0	q_0	q_2

or

Input	b	b	a
Path	q_0	q_1	q_1

Thus, you cannot take the decision of which path has to be followed for deriving the given string.

Deference between NFA and DFA.

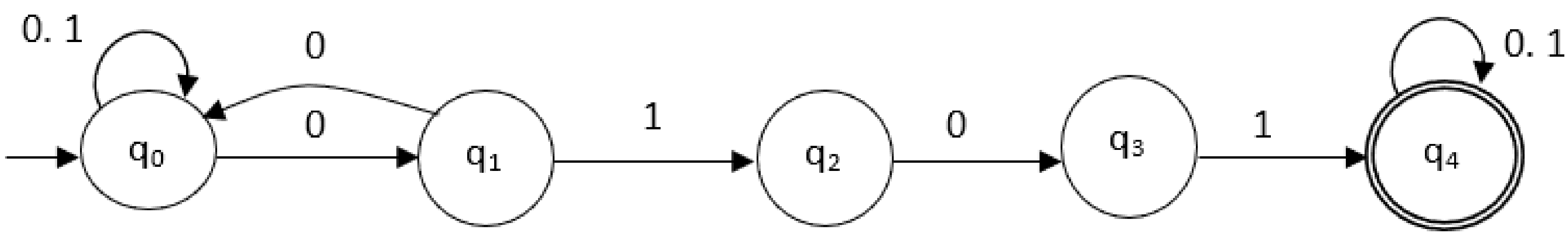
- The DFA is a deterministic finite automaton whereas NFA is a non – deterministic finite automata.
- **In DFA, for a given state, on a given input we reach to a deterministic and unique state.**
- On the other hand, **in NFA we may lead to more than one states for given input.**
- **The DFA is a subset of NFA**. We need to convert NFA to DFA in the design of computer.

Example 1.24: Construct a NFA for the language.

L1 = {consisting a substring 0 1 0 1}.

L2 = $\{a^n \cup b^n\}$.

Solution: We will consider L1 first to design NFA. There can be any combination of 0 and 1 the language but a substring 0101 must be present. We will get such a substring then it leads to a final state or accept state.



This is a NFA as for 0 input we have two different paths one going to q_0 and other is going to q_1.

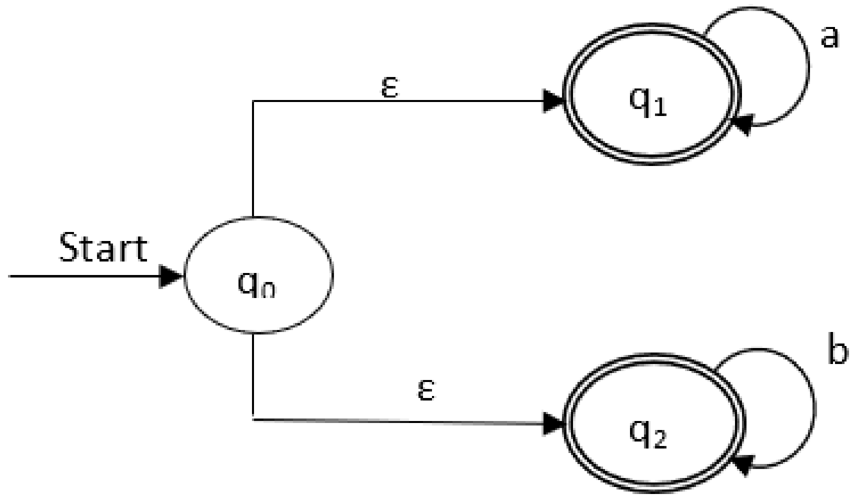
The string 00010101 is acceptable by above given NFA and it is as shown below.

$$
\begin{array}{ll}
00010101 & 0q_0 \vdash 010101 \\
 & 00q_0 \vdash 010101 \\
 & 000q_1 \vdash 10101 \\
 & 0001q_2 \vdash 0101 \\
 & 00010q_3 \vdash 101 \\
 & 000101q_4 \vdash 01 \\
 & 0001010q_4 \vdash 1 \\
 & 00010101q_4 \vdash \varepsilon
\end{array}
$$

00010101 is accepted as q_4 is final state.

Now we will build a NFA for L_2. The language L_2 is a language in which there be any number of as or any number of b's. It accepts (a, b, aa, bb, aaa, bbb, …)

Hence the NFA will be.



The NFA Shows two different states q_1 and q_2 for the input ε (pronounced as epsilon) is a null move i.e. a move carrying no symbol from input set ∑. But a state change occurs from one state to other.

Example 1.25: Design the NFA transition diagram for the transition table as given below:

	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
q_1	$\{q_3\}$	-
q_2	$\{q_2, q_3\}$	$\{q_3\}$
q_3	$\{q_3\}$	$\{q_3\}$

Here the NFA is $M = (\{q_0, q_1, q_2, q_3\}, \{0,1\}, \delta, q_0, \{q_3\})$

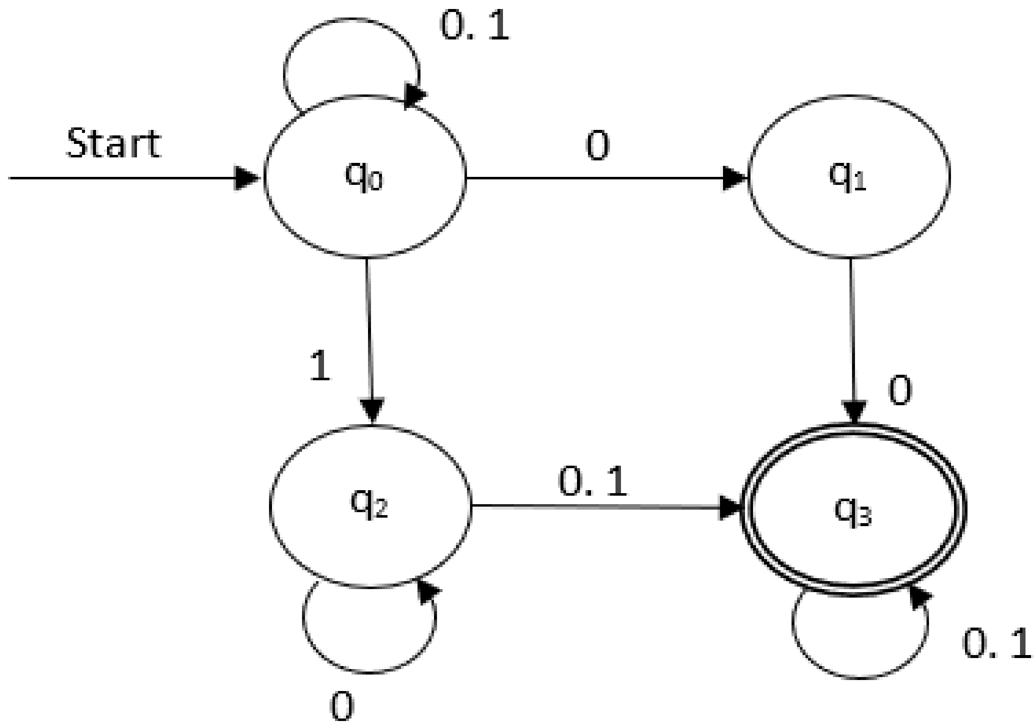
Transition function δ

Solution: The transition diagram ca be drawn by using the mapping function as given in table.

	$\delta(q_0,0)$	=	$\{q_0, q_1\}$
	$\delta(q_0,1)$	=	$\{q_0, q_2\}$
Then,	$\delta(q_1,0)$	=	$\{q_3\}$
Then,	$\delta(q_2,0)$	=	$\{q_2, q_3\}$
	$\delta(q_2,1)$	=	$\{q_3\}$
Then,	$\delta(q_3,0)$	=	$\{q_3\}$
	$\delta(q_3, 1)$	=	$\{q_3\}$

	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
q_1	$\{q_3\}$	-
q_2	$\{q_2, q_3\}$	$\{q_3\}$
q_3	$\{q_3\}$	$\{q_3\}$

	0	1
→ q_0	$\{q_0, q_1\}$	$\{q_0, q_2\}$
q_1	$\{q_3\}$	-
q_2	$\{q_2, q_3\}$	$\{q_3\}$
q_3	$\{q_3\}$	$\{q_3\}$



Example 1.26: Construct NFA for the language

$$L = \{0101^n \cup 0100 \mid n \geq 0\}$$

Over $\Sigma = \{0, 1\}$

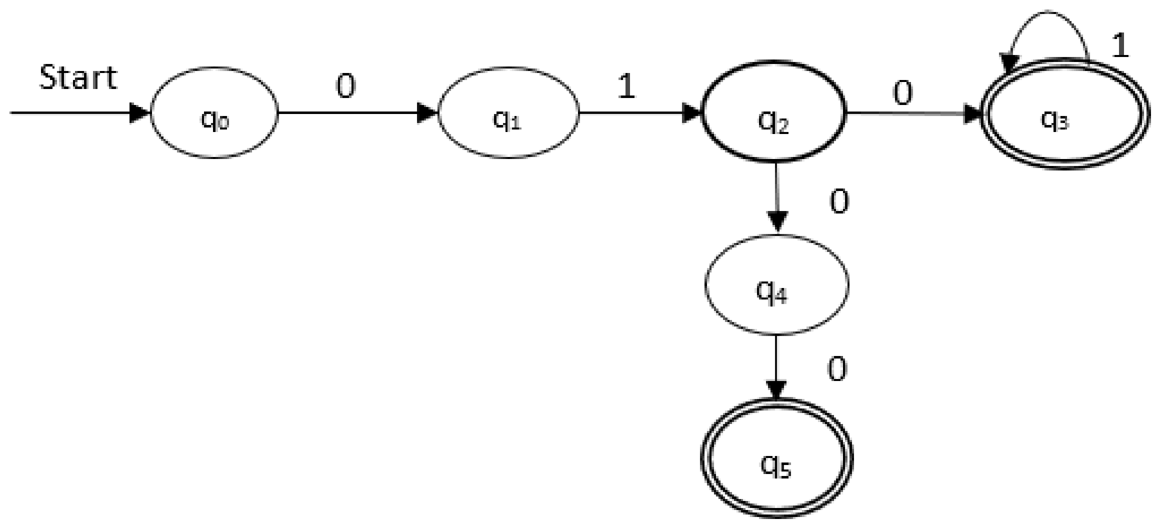
Solution: Here in language L first three symbols are common i.e. 010. Hence the NFA is

Example 1.26: Construct NFA for the language

$$L = \{0101^n \cup 0100 \mid n \geq 0\}$$

Over $\Sigma = \{0, 1\}$

Solution: Here in language L first three symbols are common i.e. 010. Hence the NFA is



The states q_3 and q_5 are final states accepting 0101^n and 0100 respectively. The NFA then can be denoted by, $M = (\{q_0, q_1, q_2, q_3, q_4, q_5\}, \delta, q_0, \{q_3, q_5\})$

Example 1.27: Construct a transition diagram for the NFA

$M = (\{q_1, q_2, q_3), \delta, q_1, \{q_3\})$ where δ is given by

$\delta\ (q_1, 0) = \{q_2, q_3\}$ $\qquad$ $\delta\ (q_1, 1) = \{q_1\}$

$\delta\ (q_2, 0) = \{q_1, q_2\}$ $\qquad$ $\delta\ (q_2, 1) = \phi$

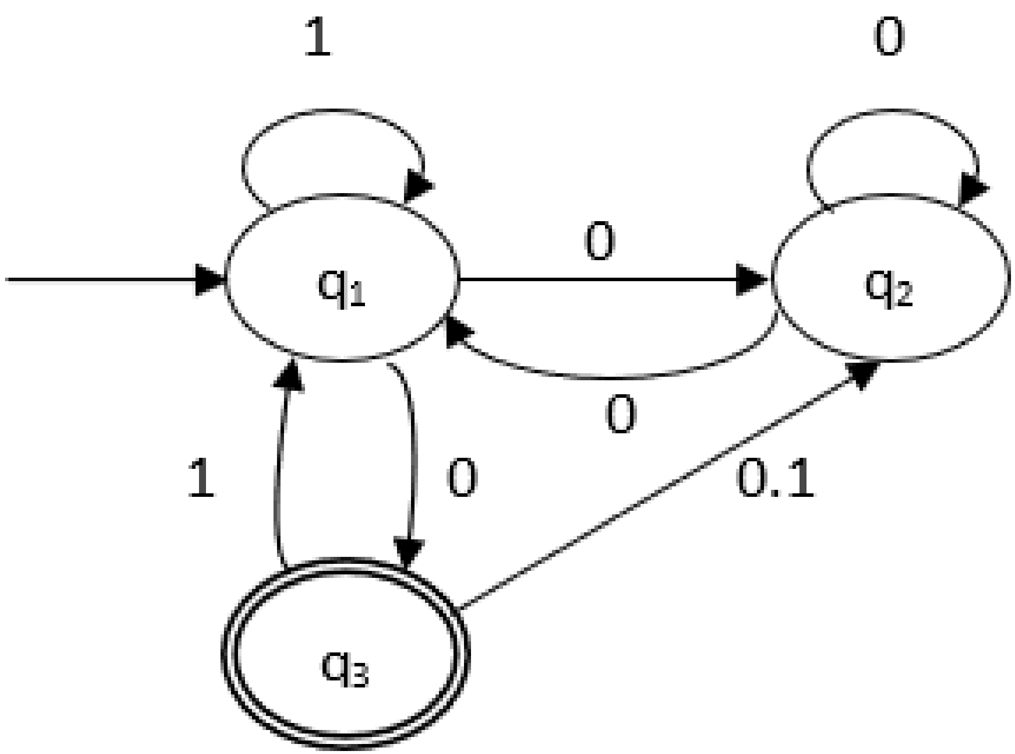
$\delta\ (q_3, 0) = \{q_2\}$ $\qquad$ $\delta\ (q_3, 1) = \{q_1, q_2\}$

Solution: Firstly, we will design a transition table using the given mapping function.

States \ Input	0	1
$\rightarrow q_1$	$\{q_2, q_3\}$	q_1
q_2	$\{q_1, q_2\}$	ϕ
q_3	q_2	$\{q_1, q_2\}$

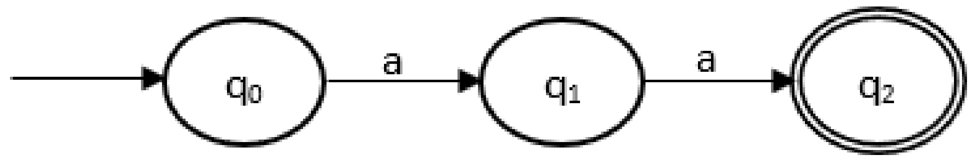
The NFA will be.

The NFA will be.

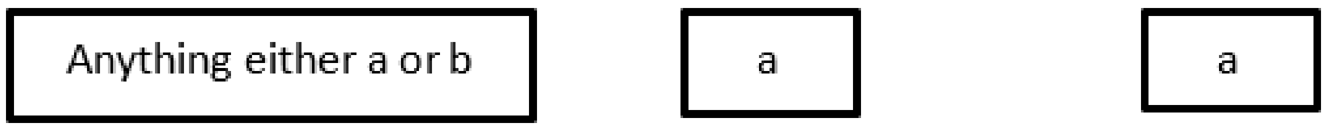


Design NFA accepting all strings ending with aa over {a,b}

Solution: The simple FA which accepts a string with ' aa' is

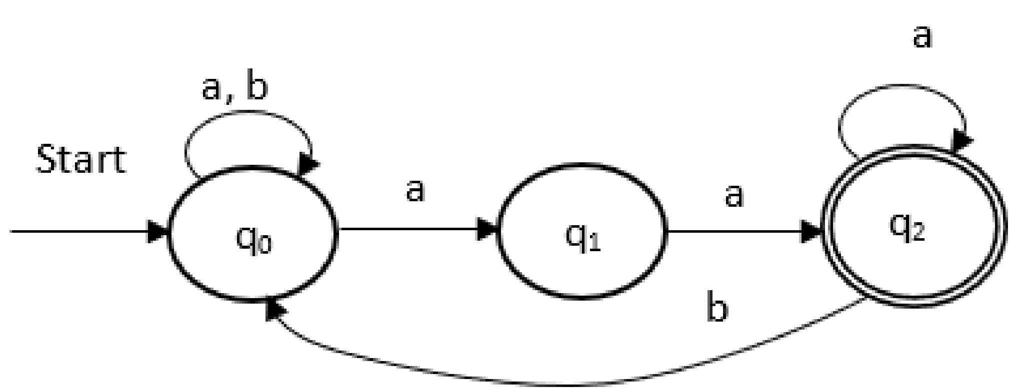


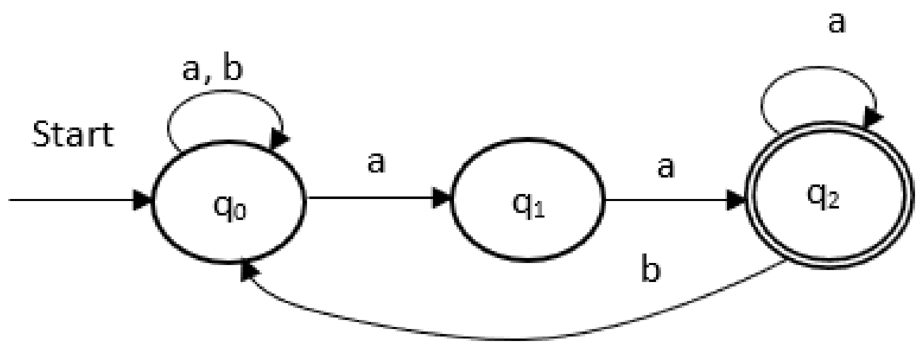
Now there can be a situation where in:



Hence, we can design a required NFA as. It can be denoted by,

$M = (\{q_0, q_1, q_2\}, \delta, \{q_0\}, \{q_2\})$





We can test some strings for above drawn NFA.

Consider.

$$\begin{aligned}\delta(q_0, aaa) &\vdash \delta(q_0, aa)\\ &\vdash \delta(q_1, a)\\ &\vdash \delta(q_2, \varepsilon)\end{aligned}$$

i.e. we are reaching to accept state.

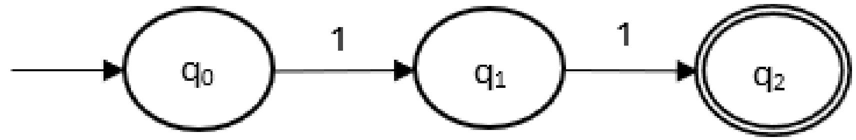
$$\begin{aligned}\delta(q_0, aaa) &\vdash \delta(q_0, aa)\\ &\vdash \delta(q_0, a)\\ &\vdash \delta(q_1, \varepsilon)\end{aligned}$$

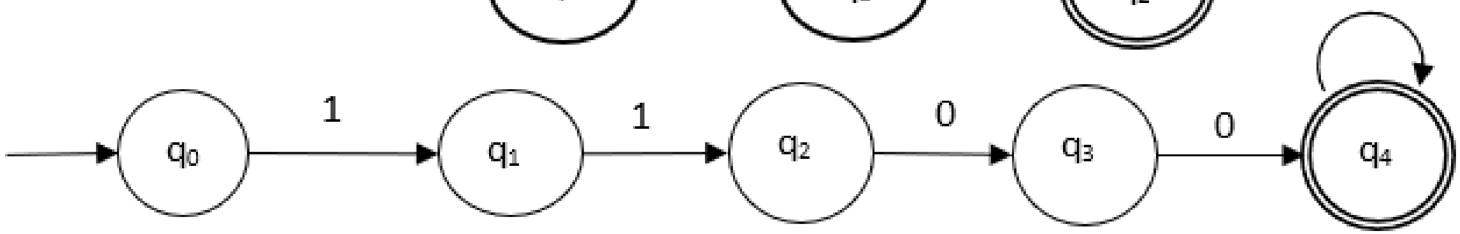
i.e. We are not in final state.

Thus, there are two possibilities by which we move with string ' aaa ' in above given NFA.

Example 1. 31: Construct a NFA in which double '1' is followed by double '0', over $\sum = (0, 1)$

Solution: The FA with double 1 is as drawn below.

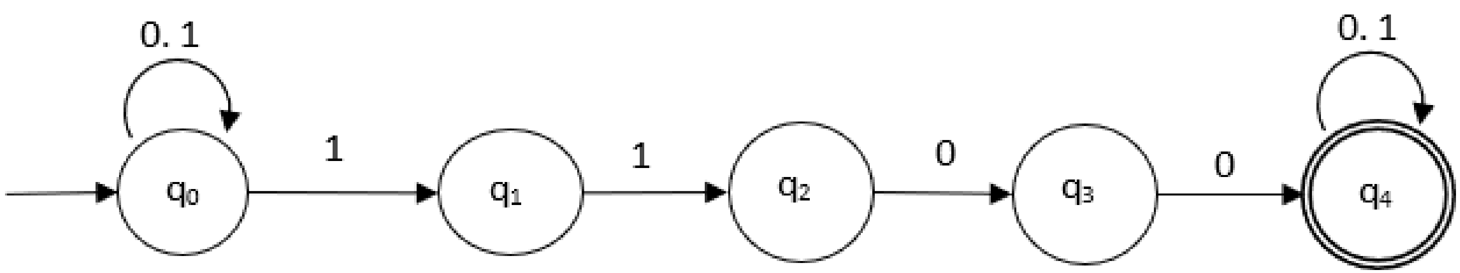




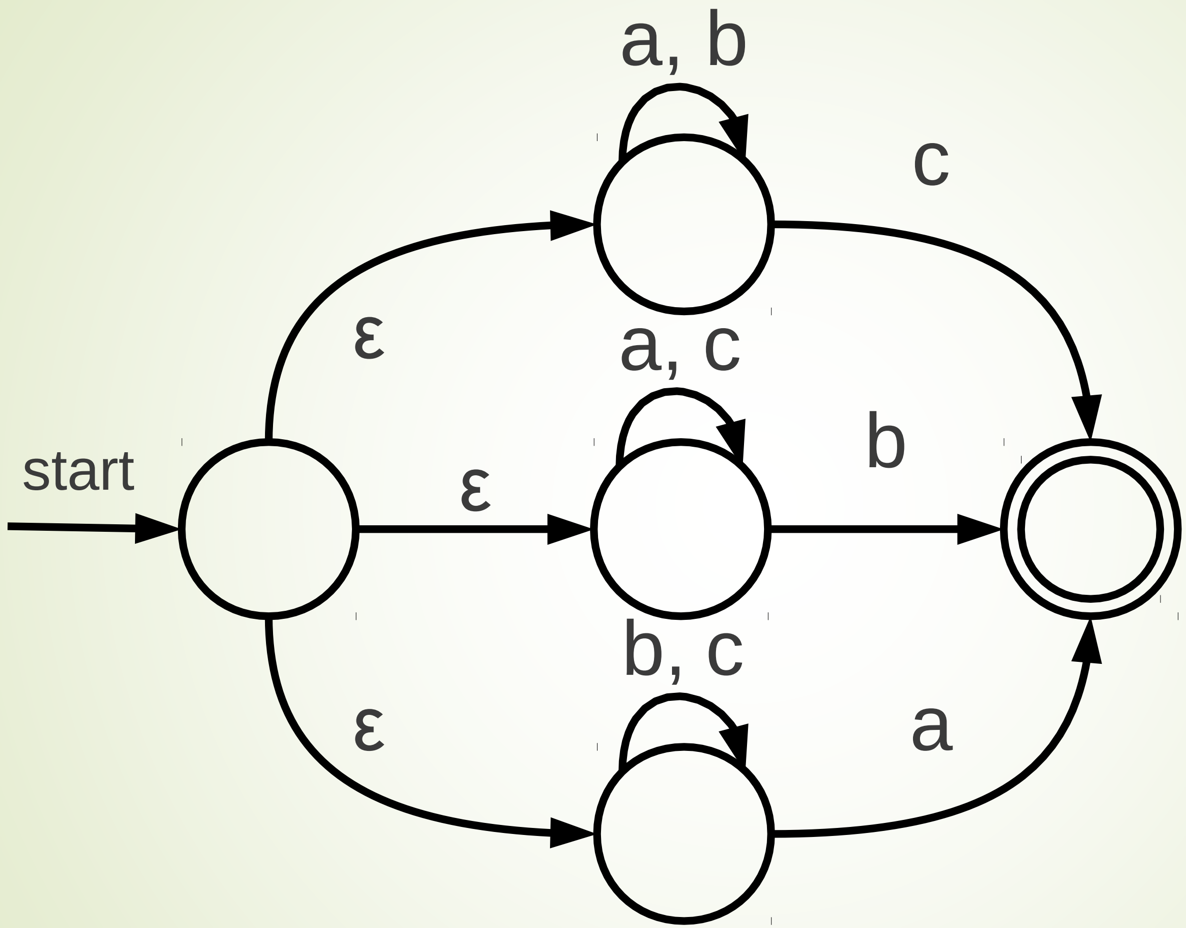
It should be immediately followed by double 0. Then,

Now before double 1 there can be any string of 0 and 1. Similarly after double there can be any string of 0 and 1.

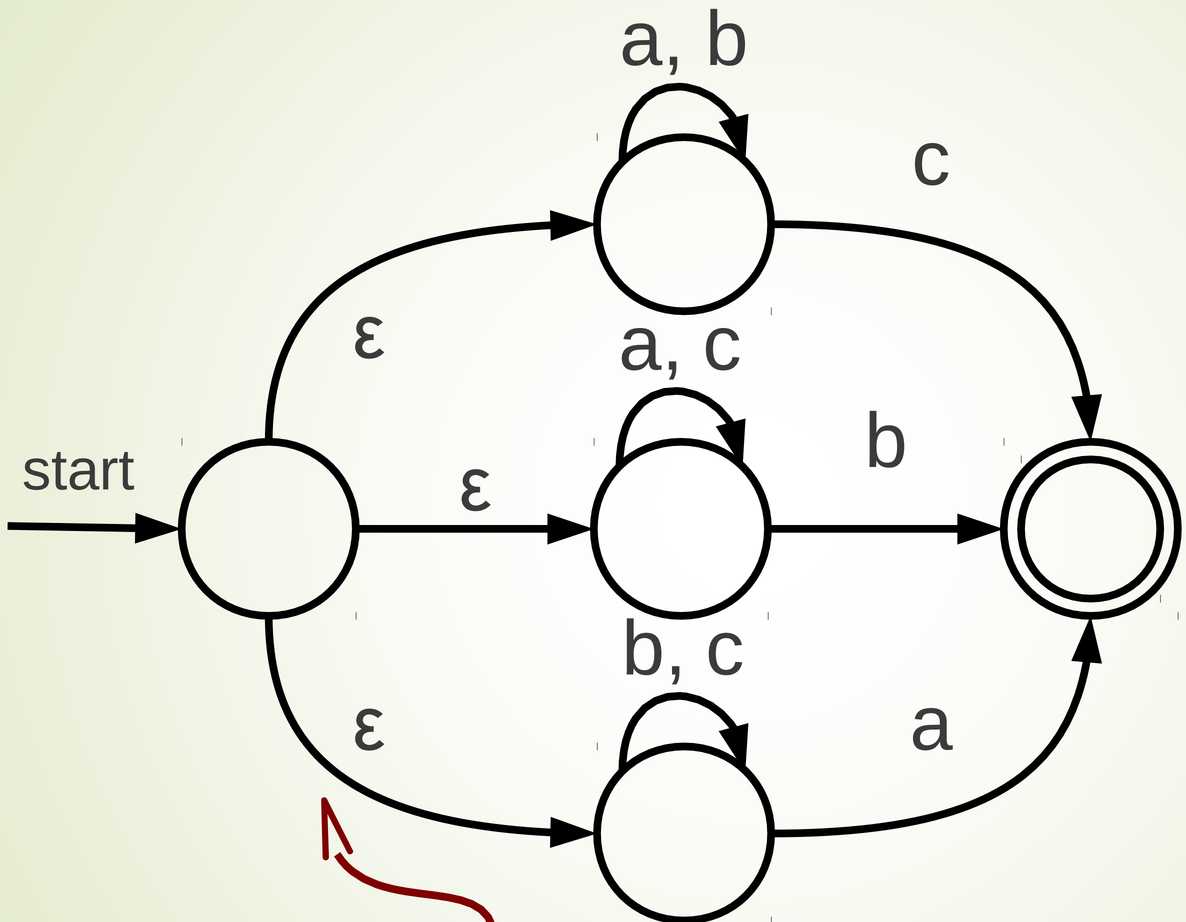
Hence, the NFA becomes:



An ε-NFA example

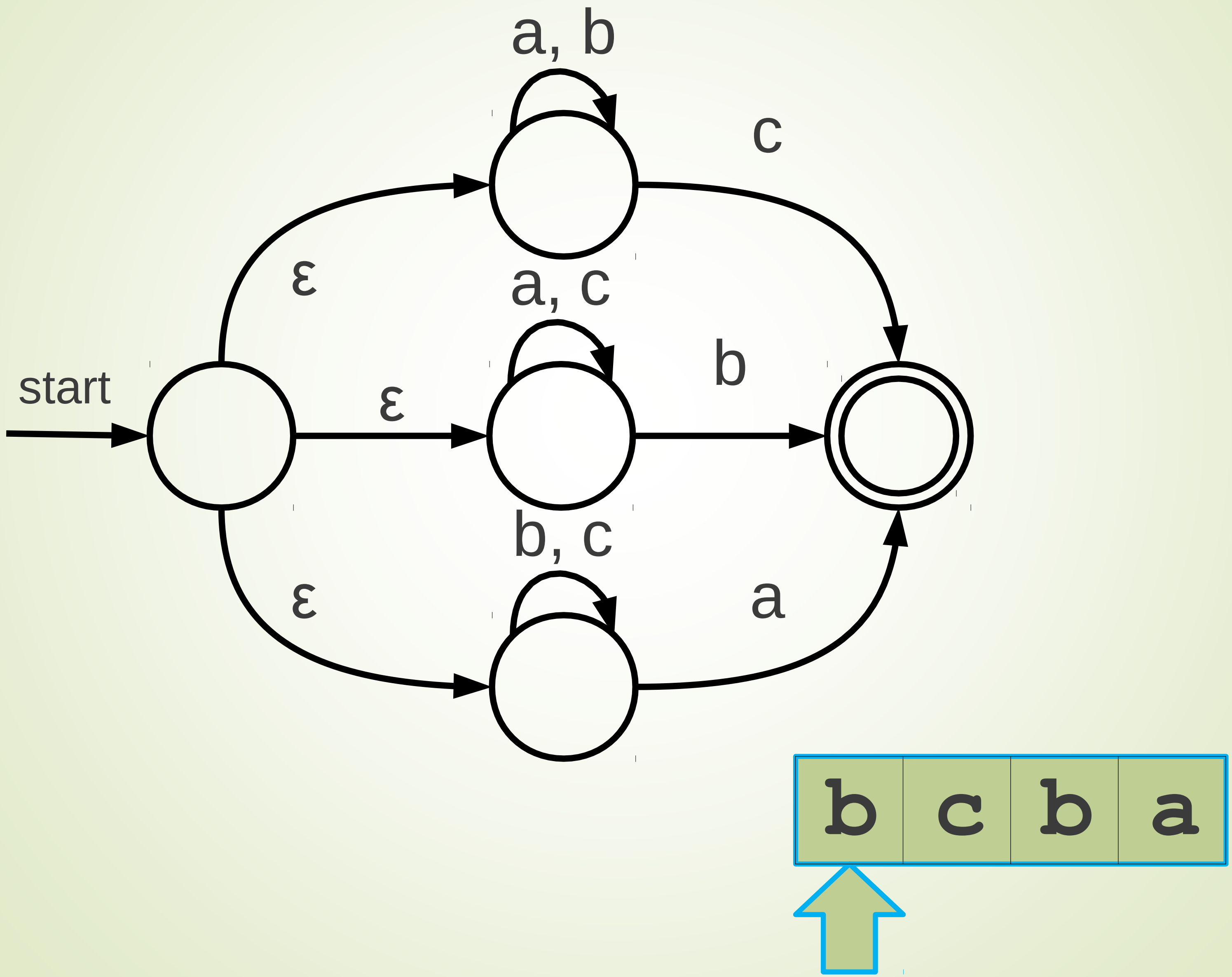


ε-NFA Automaton

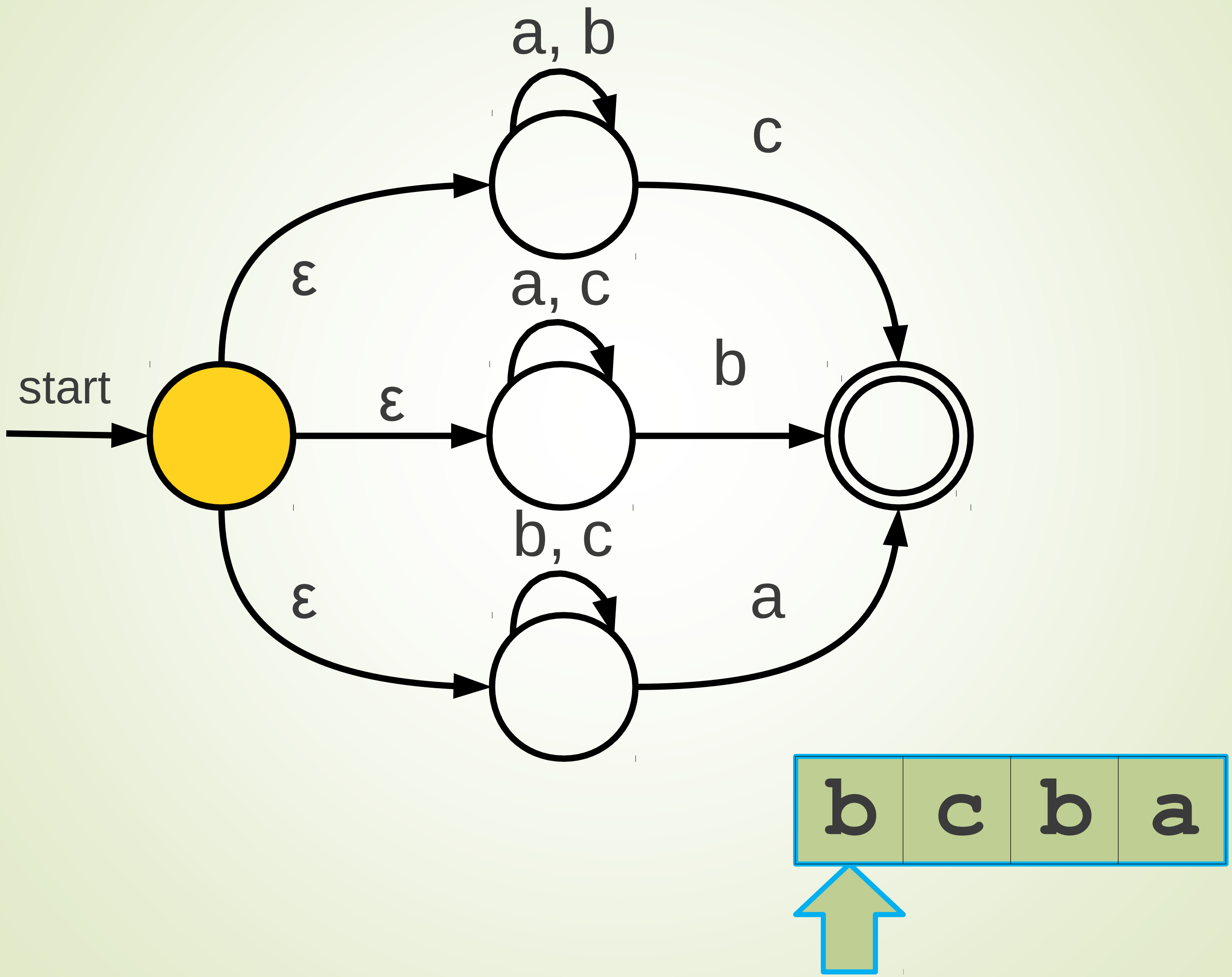


These are called **ε-transitions**. These transitions are followed automatically and without consuming any input.

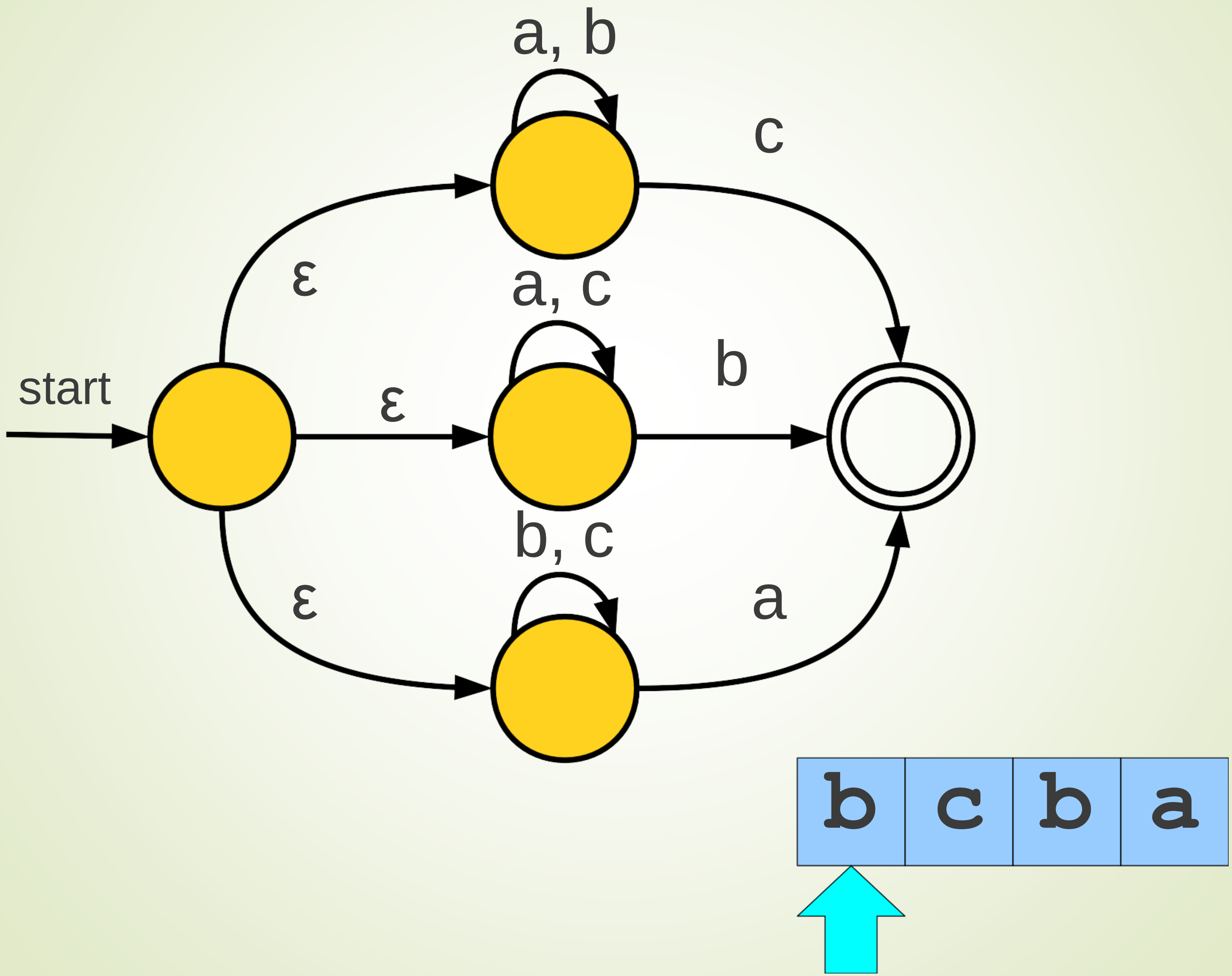
ε-NFA Automaton



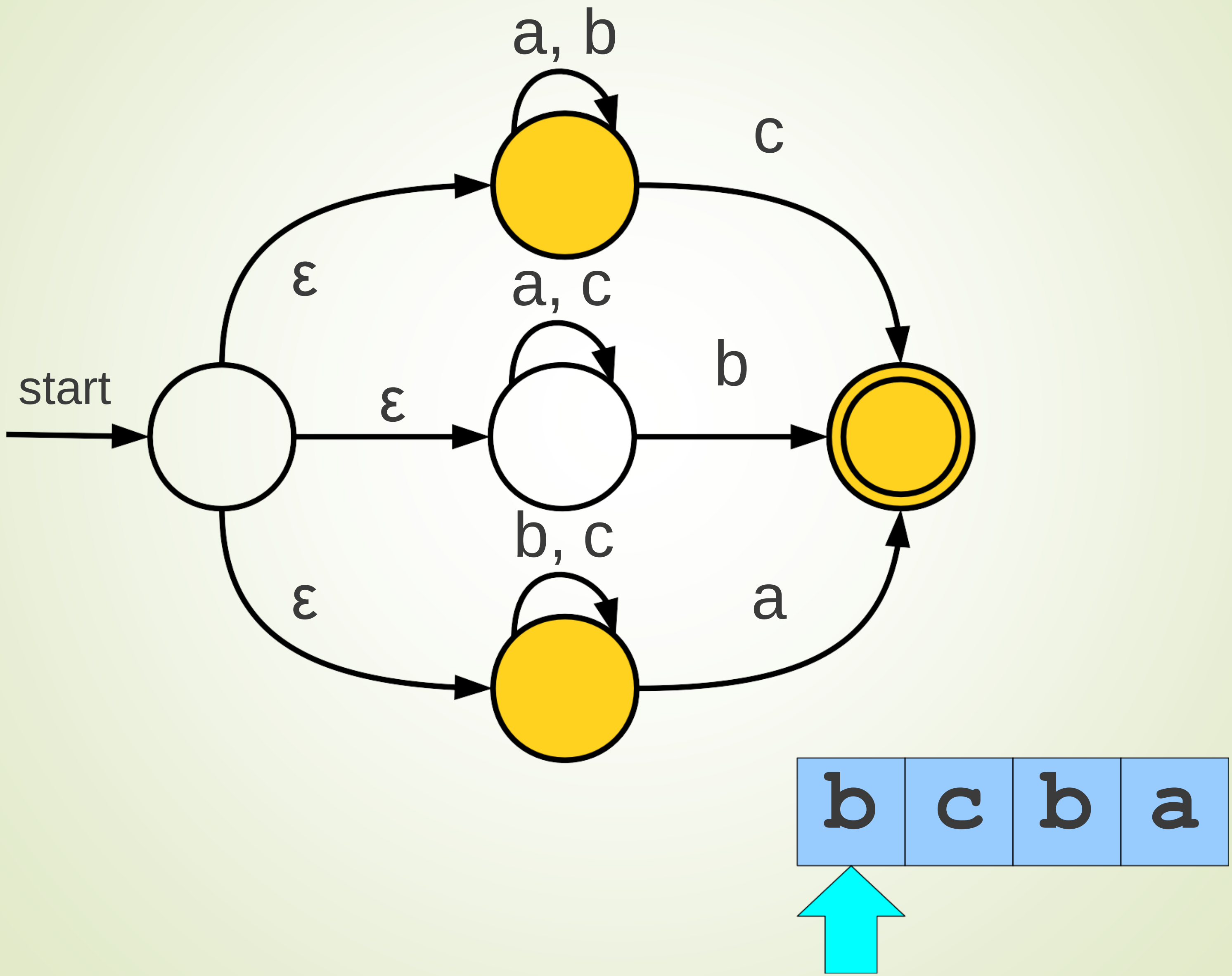
ε-NFA Automaton



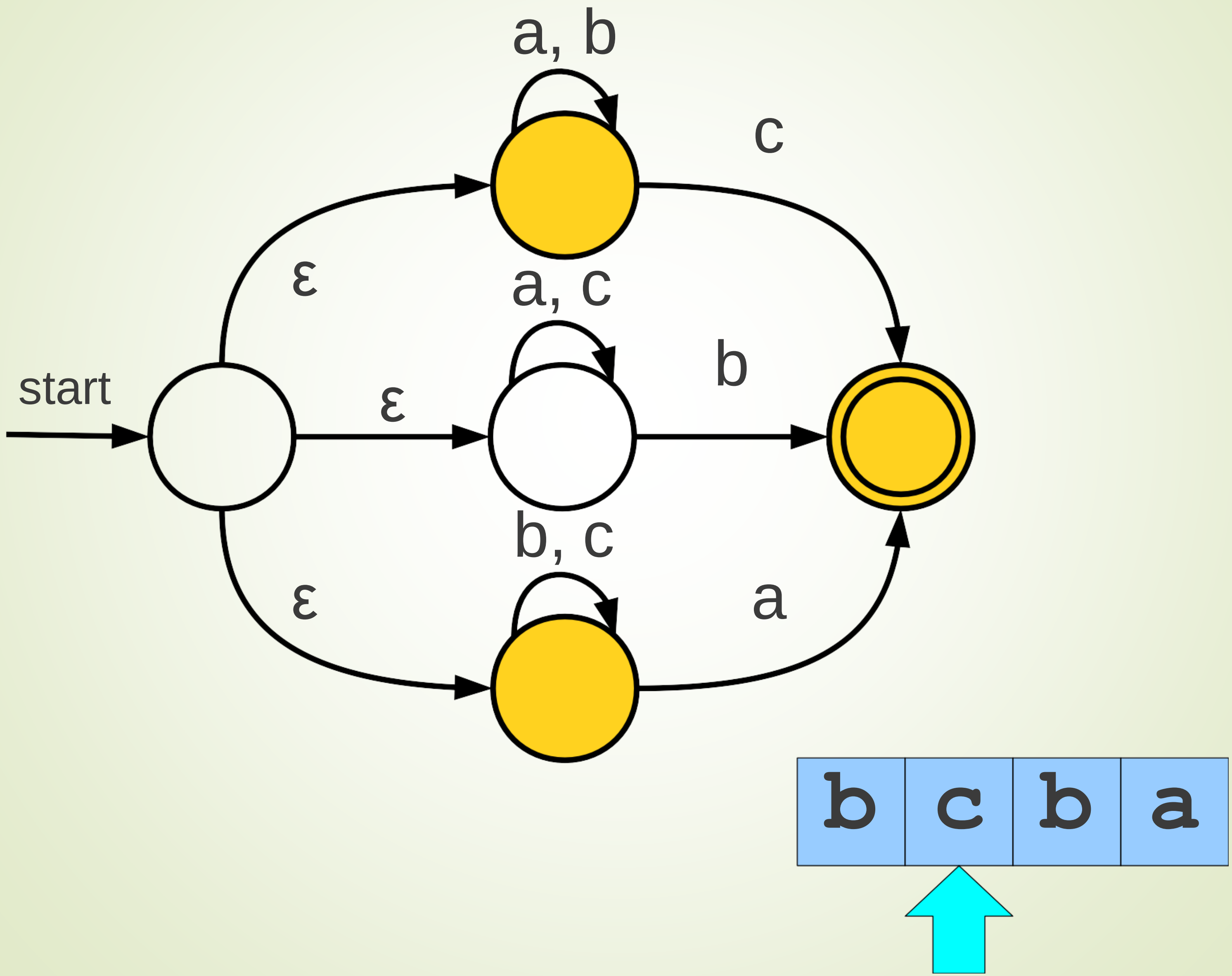
ε-NFA Automaton



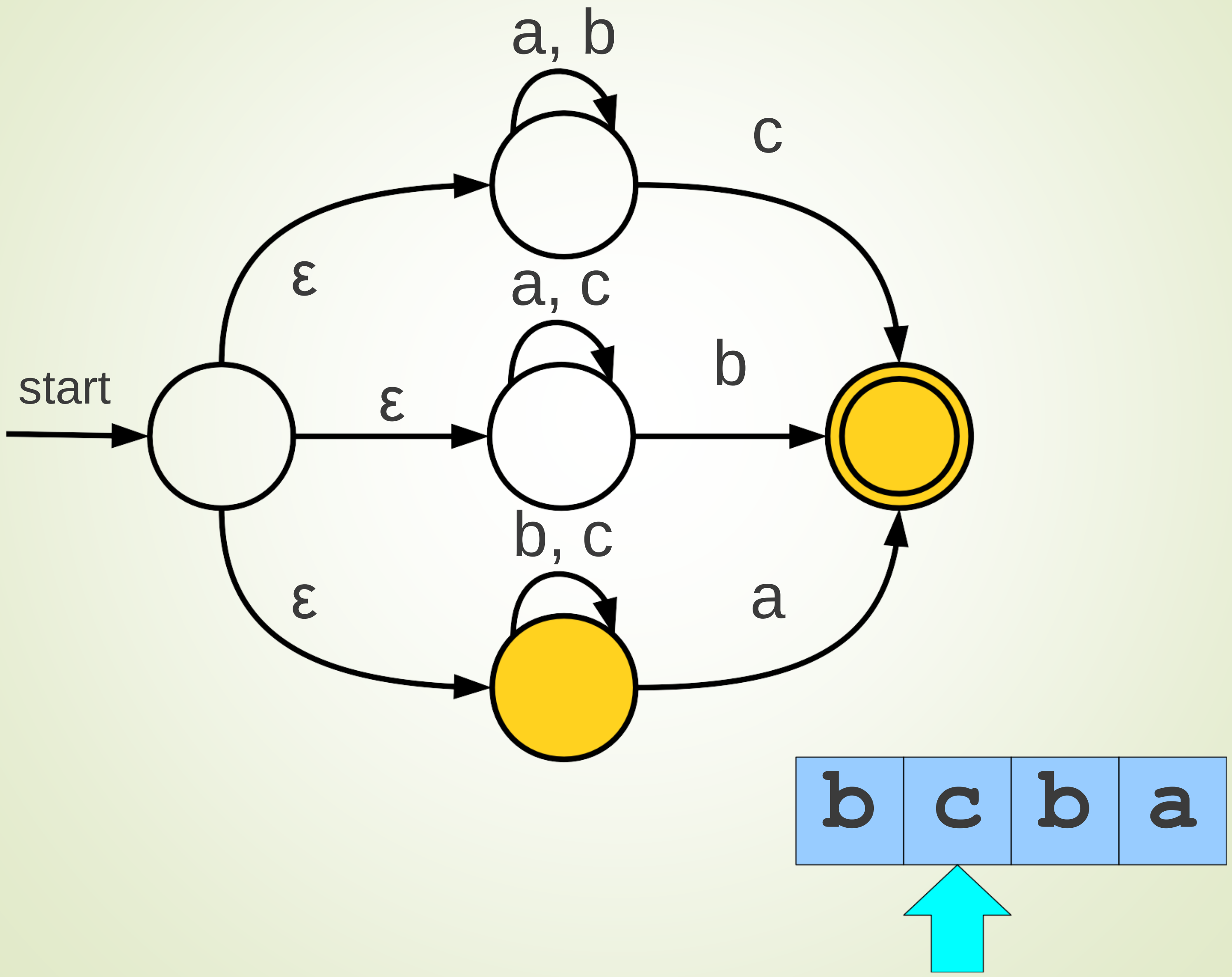
ε-NFA Automaton



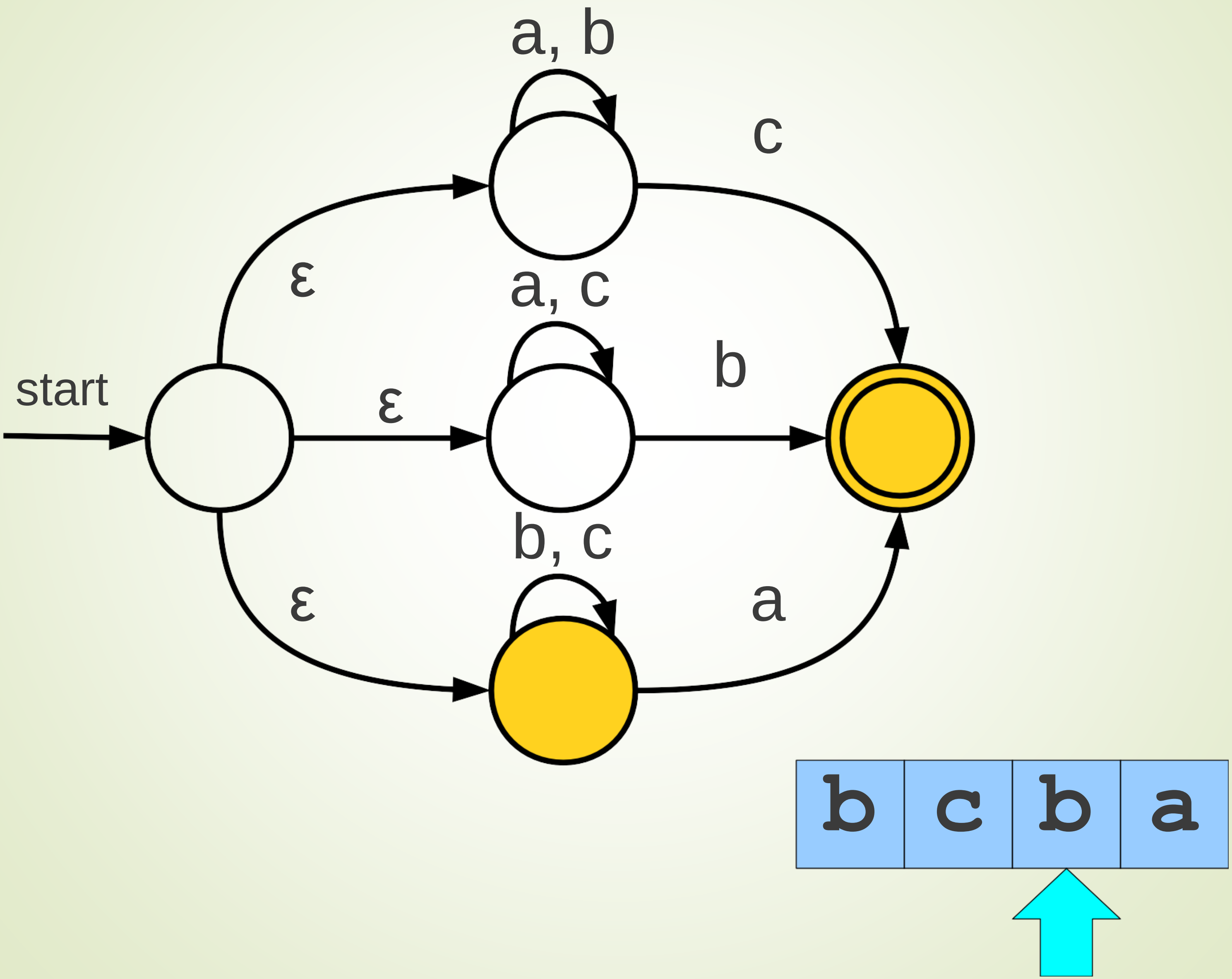
ε-NFA Automaton



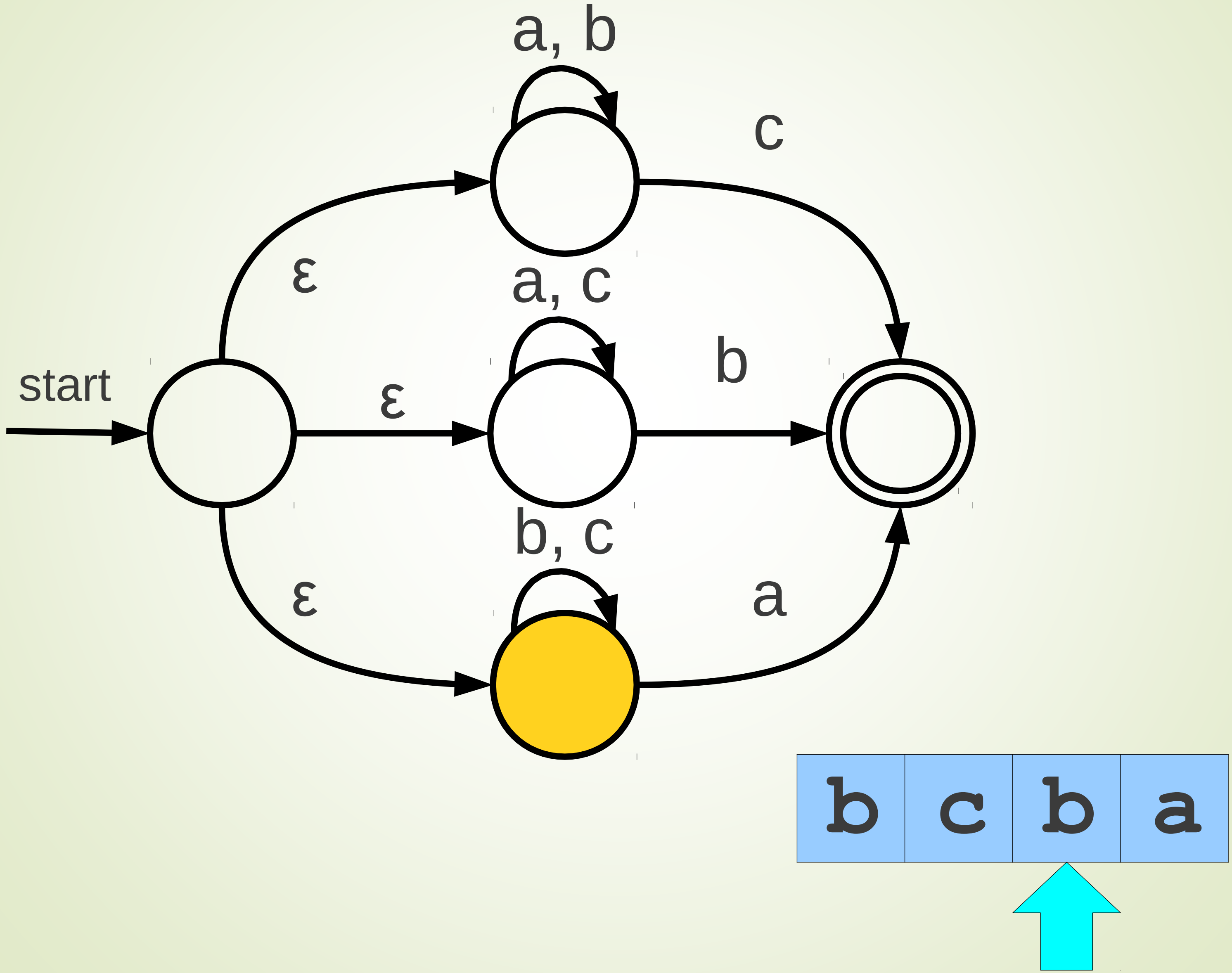
ε-NFA Automaton



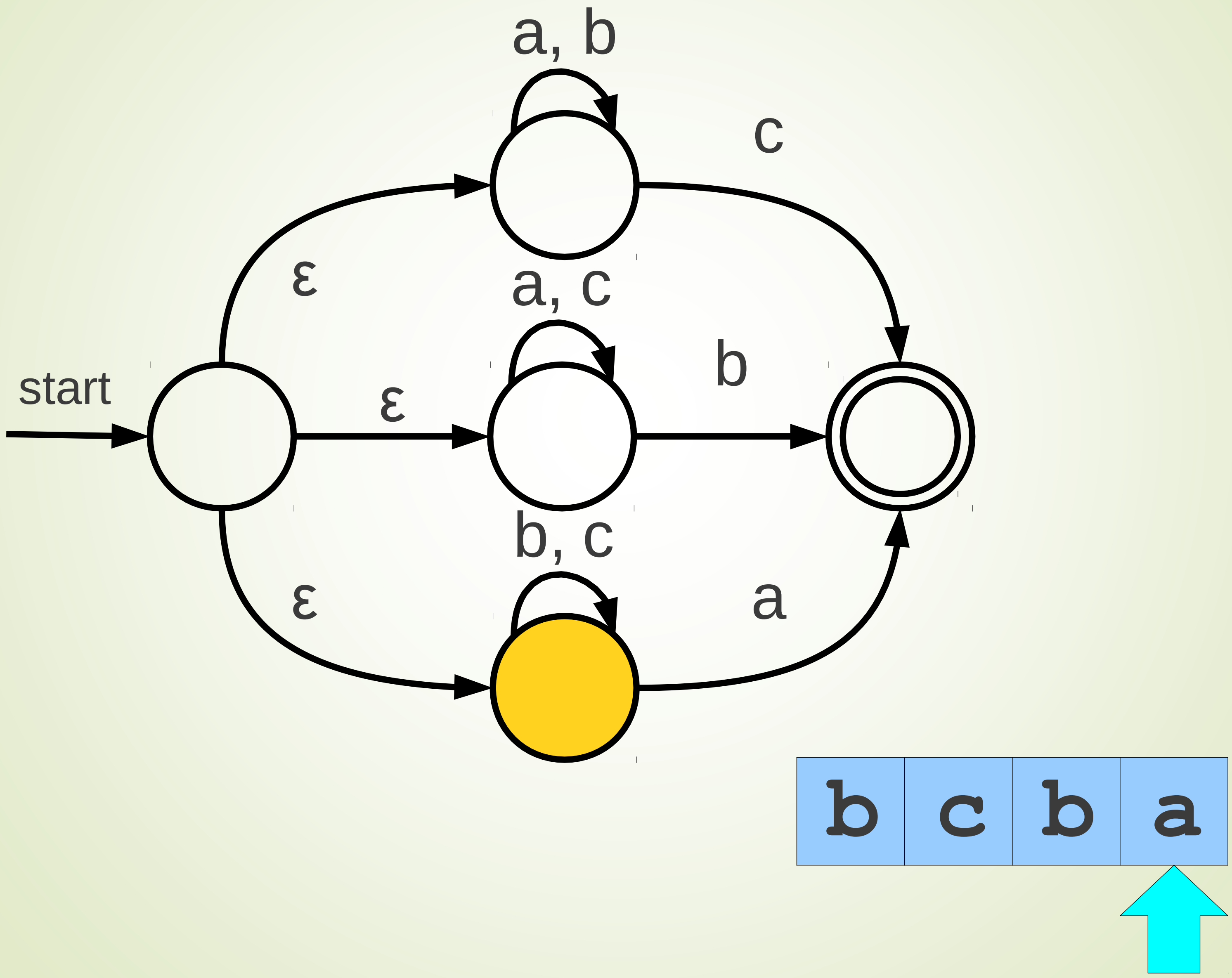
ε-NFA Automaton



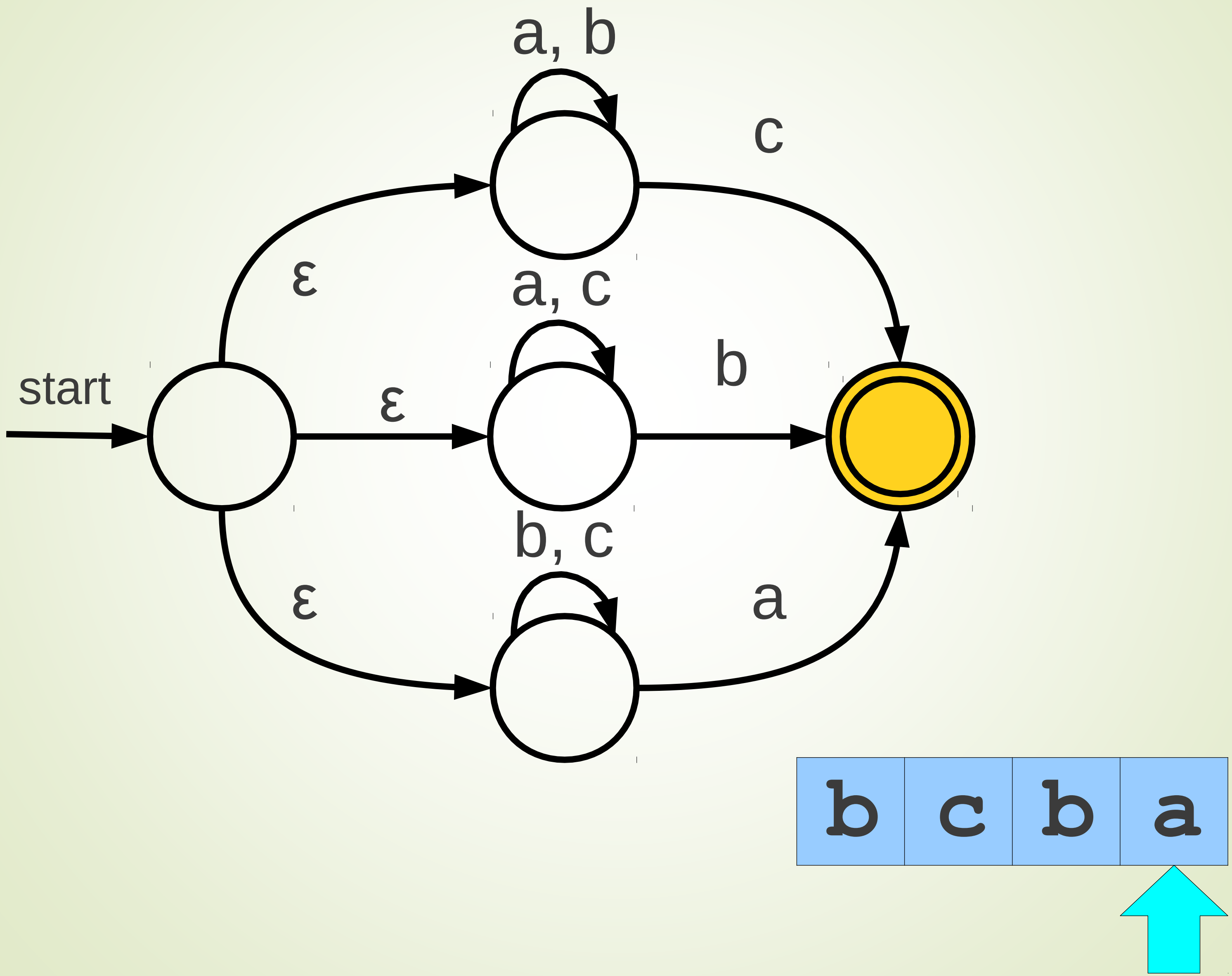
ε-NFA Automaton



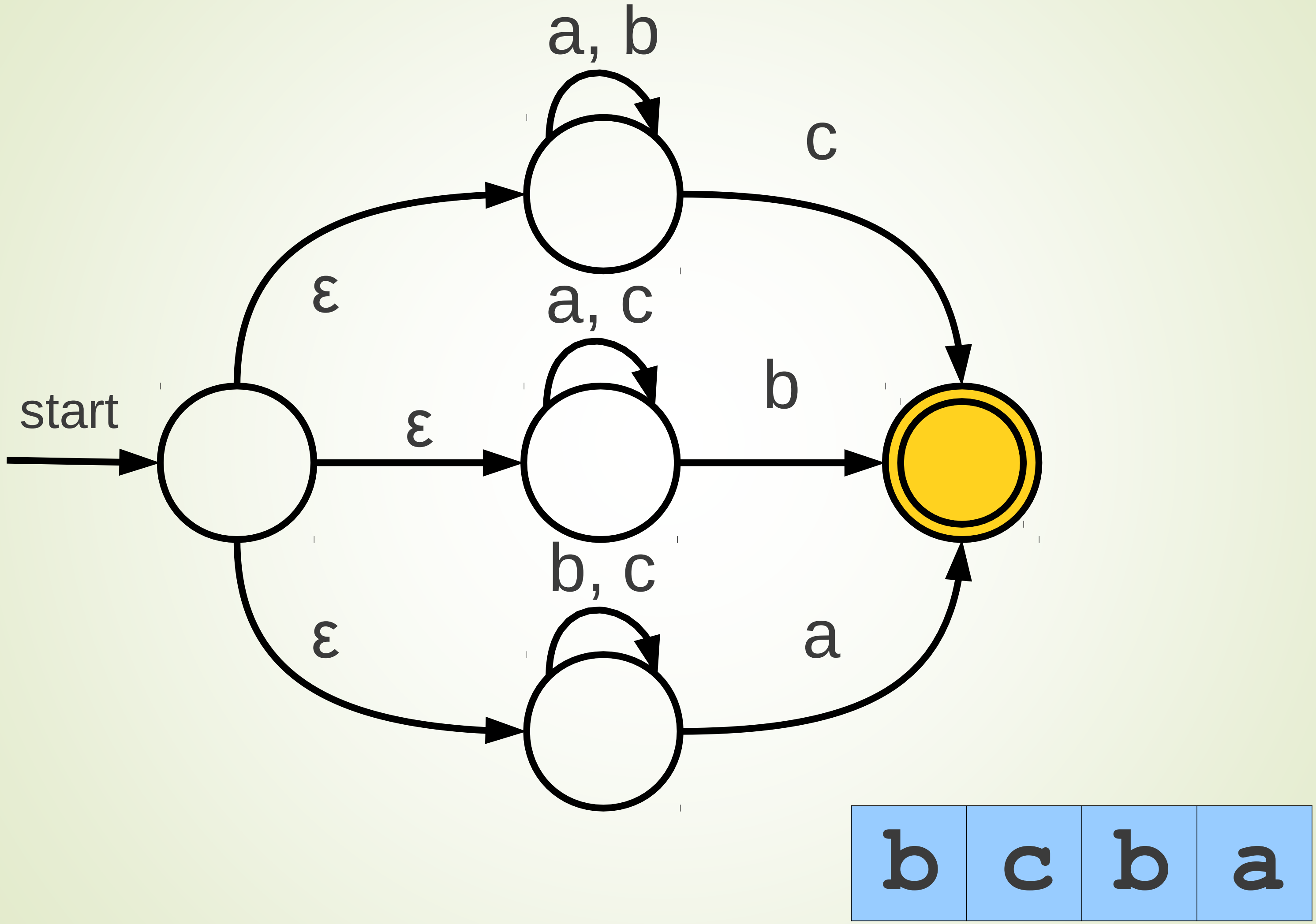
ε-NFA Automaton



ε-NFA Automaton



ε-NFA Automaton



DFA vs. NFA

DFA	NDFA
The transition from a state is to a single particular next state for each input symbol. Hence it is called *deterministic*.	The transition from a state can be to multiple next states for each input symbol. Hence it is called *non-deterministic*.
Empty string transitions are not seen in DFA.	NDFA permits empty string transitions.
Requires more space.	Requires less space.
A string is accepted by a DFA, if it transits to a final state.	A string is accepted by a NDFA, if at least one of all possible transitions ends in a final state.

Lab tutorial (2)

1: Design NFA which accepts the string containing either ' 01 ' or ' 10 ' Over ∑ (0, 1).

2: Construct a NFA in which double '1' is followed by double '0', over ∑ = (0, 1)

3: Construct a non-deterministic finite automata accepting (0 1, 1 0) and use it to find a deterministic finite automata.

Lab tutorial (3)

4: For the given transition table, obtain the translation diagram, find out whether following strings are accepted by this machine or not.

a) 101101
b) 00000

Σ State	0	1
q_0	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2

5: Construct a deterministic finite automata equivalent to

$M = (\{q_0, q_1, q_2, q_3\}, \{0, 1\}, \delta, q_0, \{q_3\})$

Σ State	0	1
q_0	q_0, q_1	q_0
q_1	q_2	q_1
Q_2	q_3	q_3
q_3	φ	q_2

Any Questions?