23MT2014

THEORY OF COMPUTATION

Topic:

PUSH DOWN AUTOMATA PDA

Session - 1

AIM OF THE SESSION

The aim of this session is to provide an understanding of Pushdown Automata (PDA), their structure, working principles, and their applications in the field of theoretical compu

INSTRUCTIONAL OBJECTIVES

This Session is designed to:

•To introduce the concept of Pushdown Automata and their components, including the stack.
•To explain the behavior and transition rules of Pushdown Automata.
•To discuss the applications of Pushdown Automata in language recognition and parsing.

LEARNING OUTCOMES

At the end of this session, you should be able to:

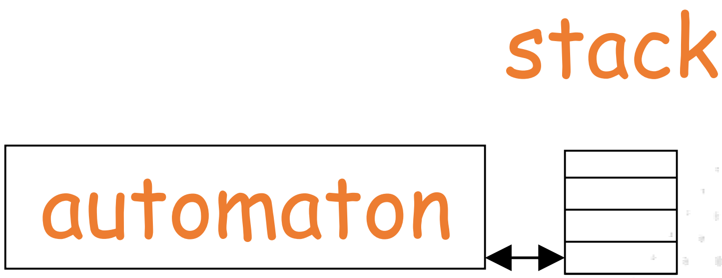
•Define what a Pushdown Automaton is and identify its key components: states, input alphabet, stack alphabet, transition function, and accepting states.
•Construct a Pushdown Automaton to recognize a given context-free language or solve a specific problem.
•Understand the relationship between Pushdown Automata and context-free grammars in the context of language recognition.

Pushdown Automata
PDAs

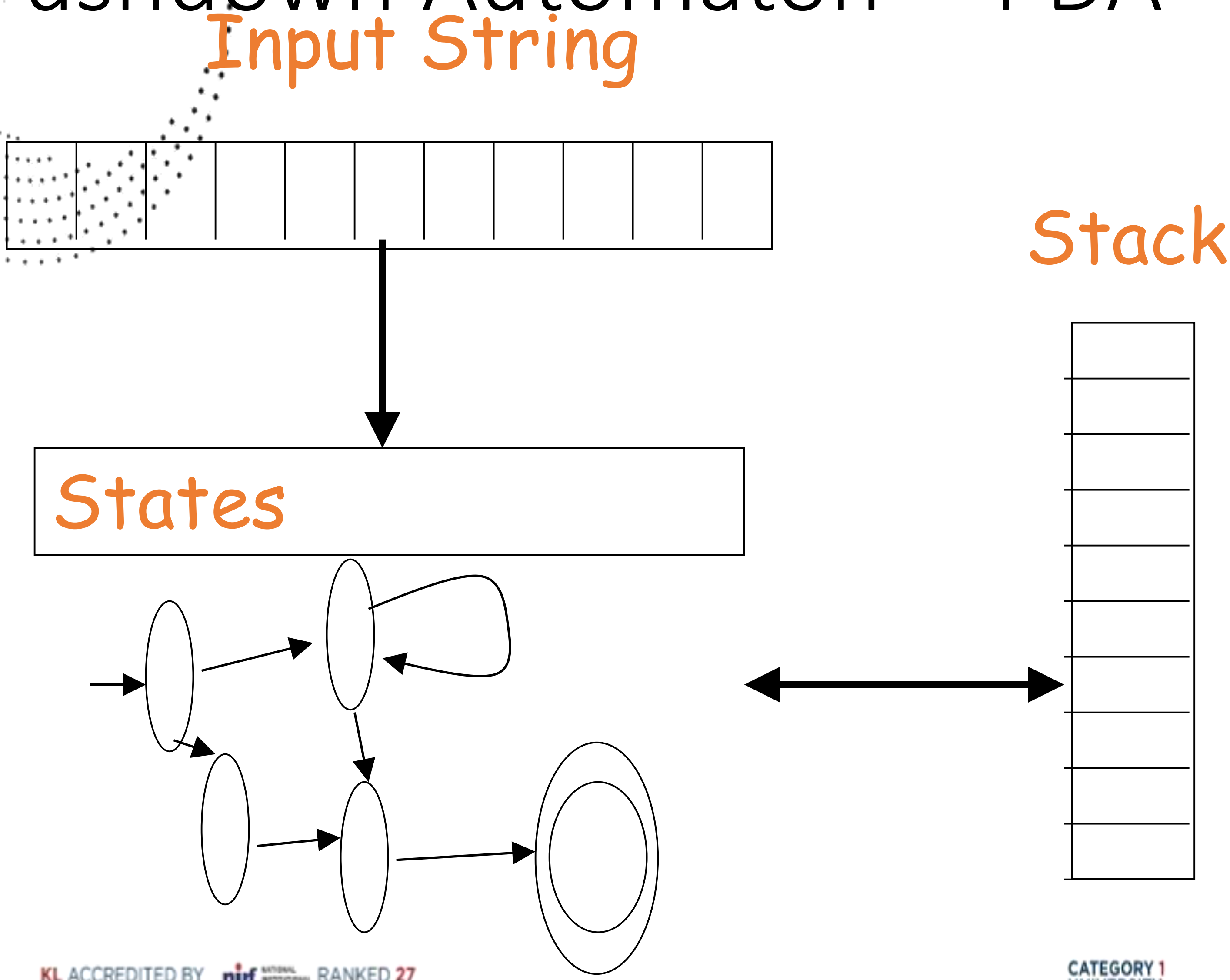
Context-Free Languages

Context-Free Grammars

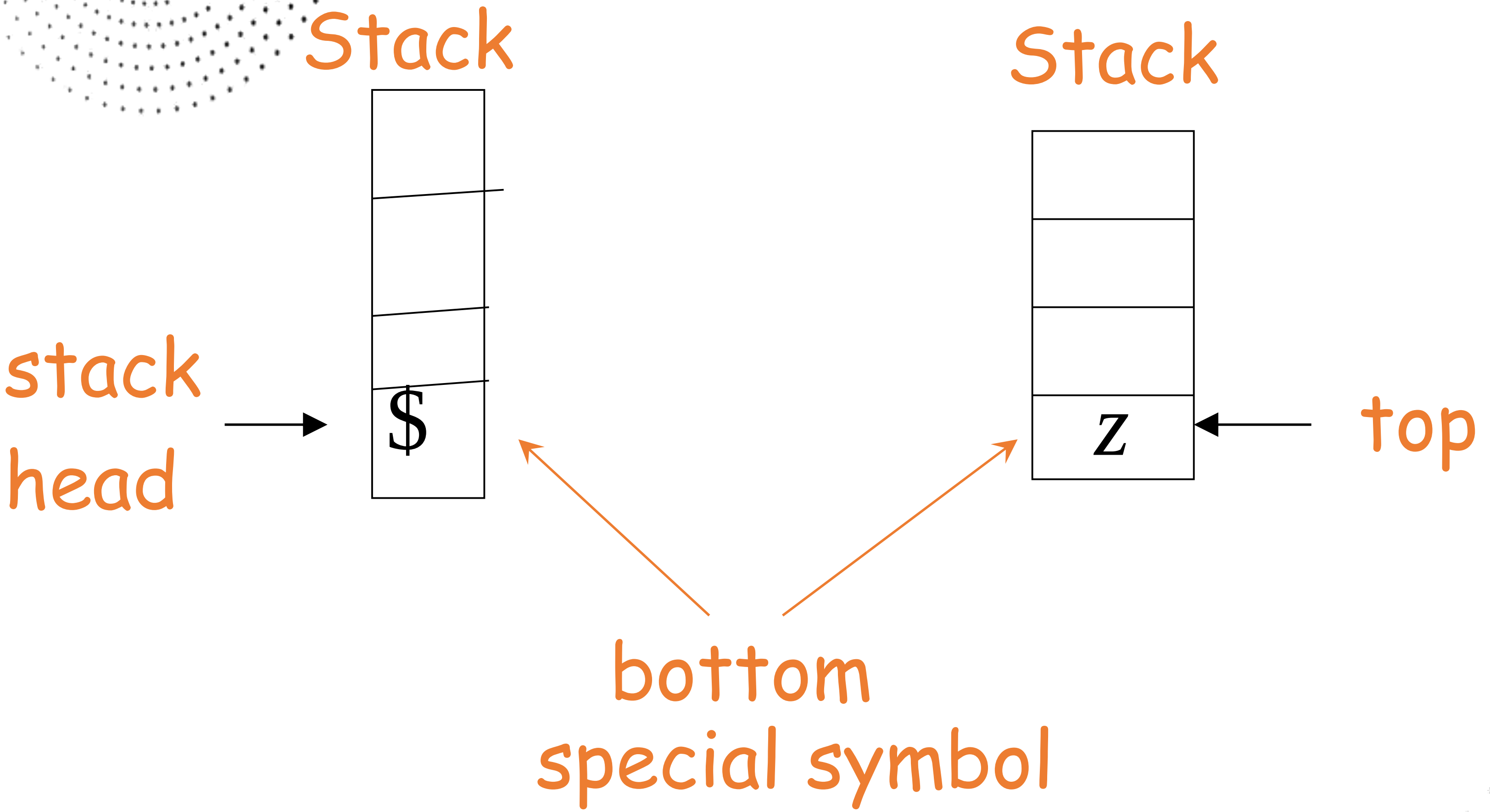
Pushdown Automata



Pushdown Automaton -- PDA



Initial Stack Symbol



The States

Input symbol

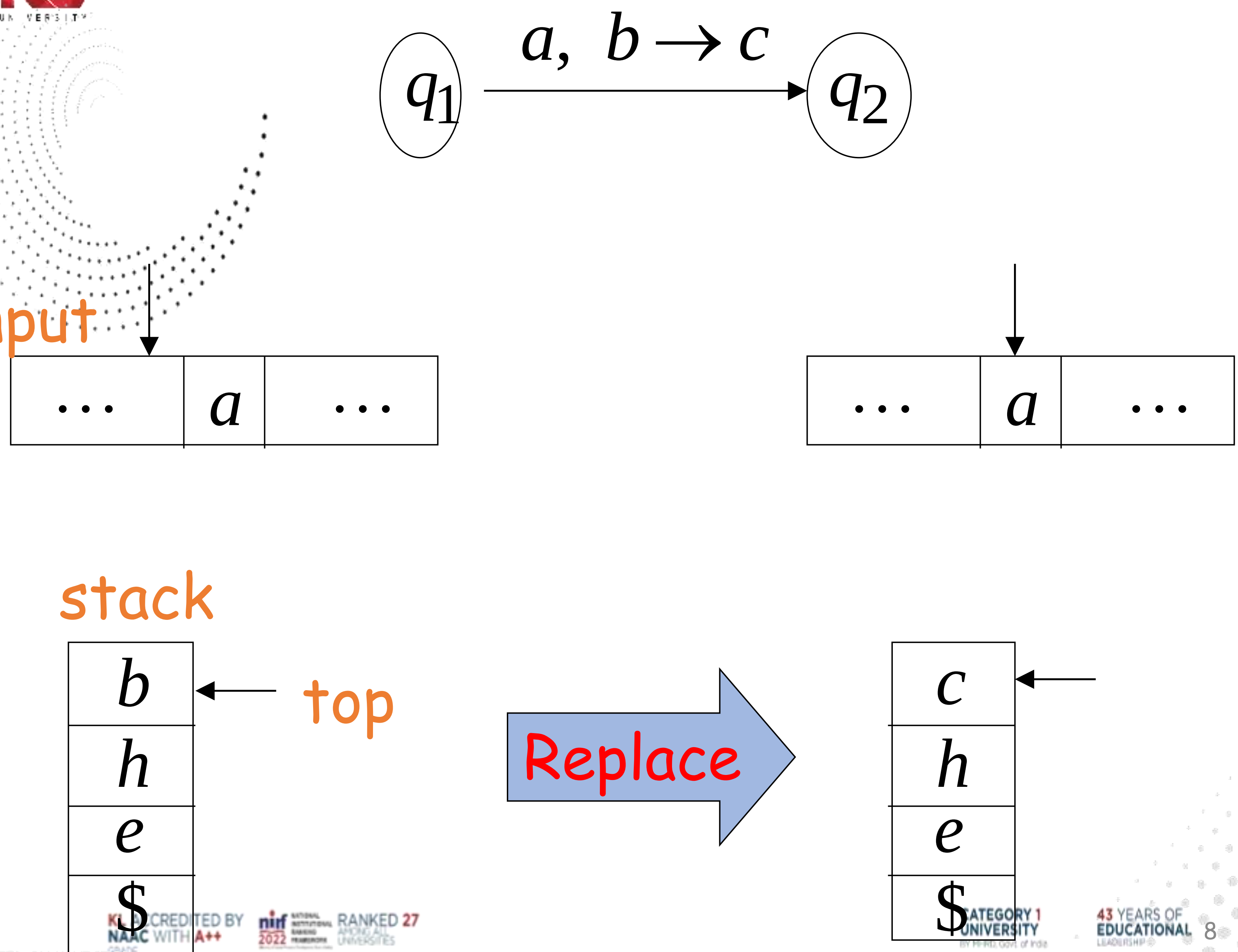
Pop symbol

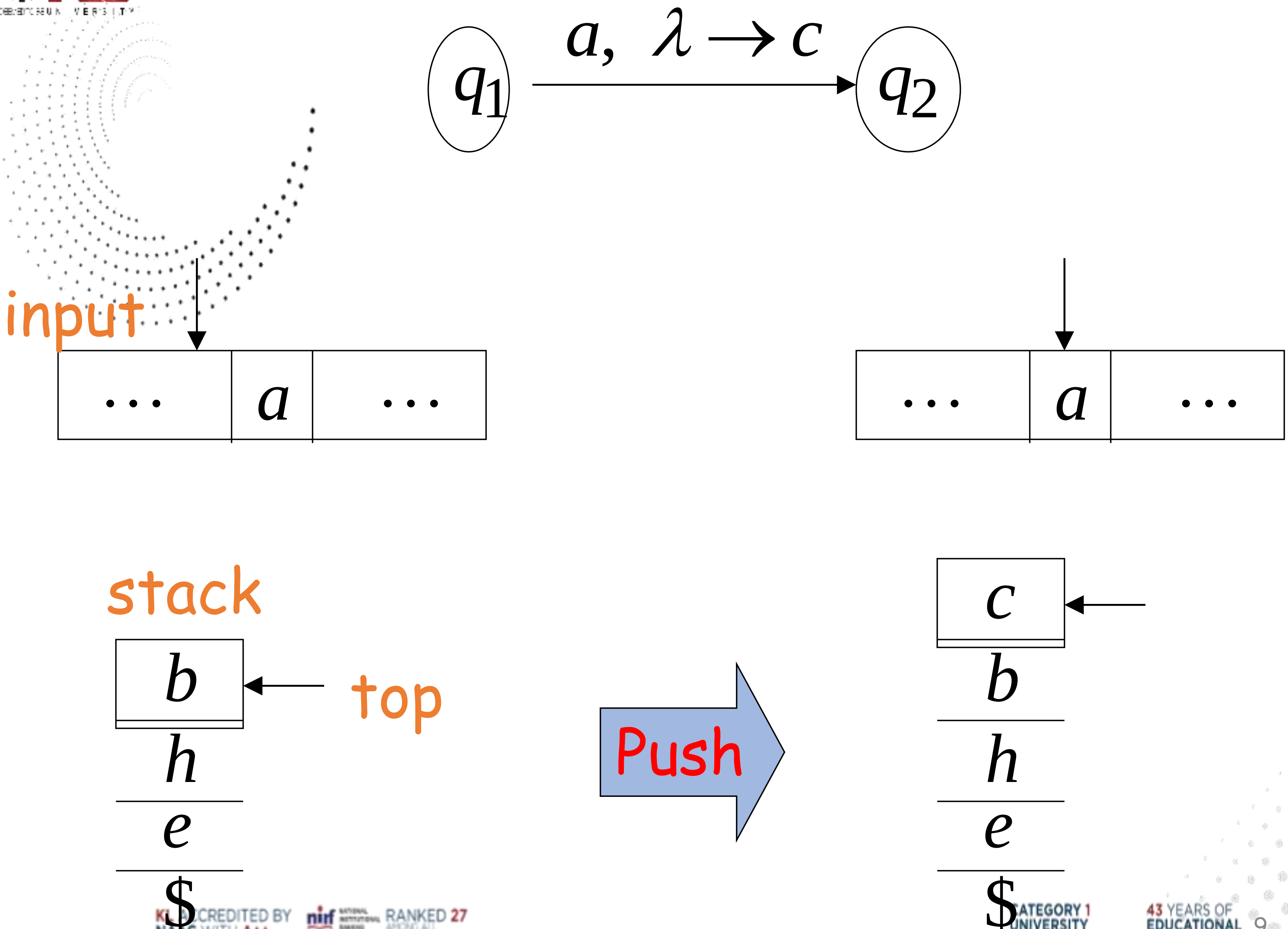
Push symbol

$$q_1 \xrightarrow{a,\ b \to c} q_2$$

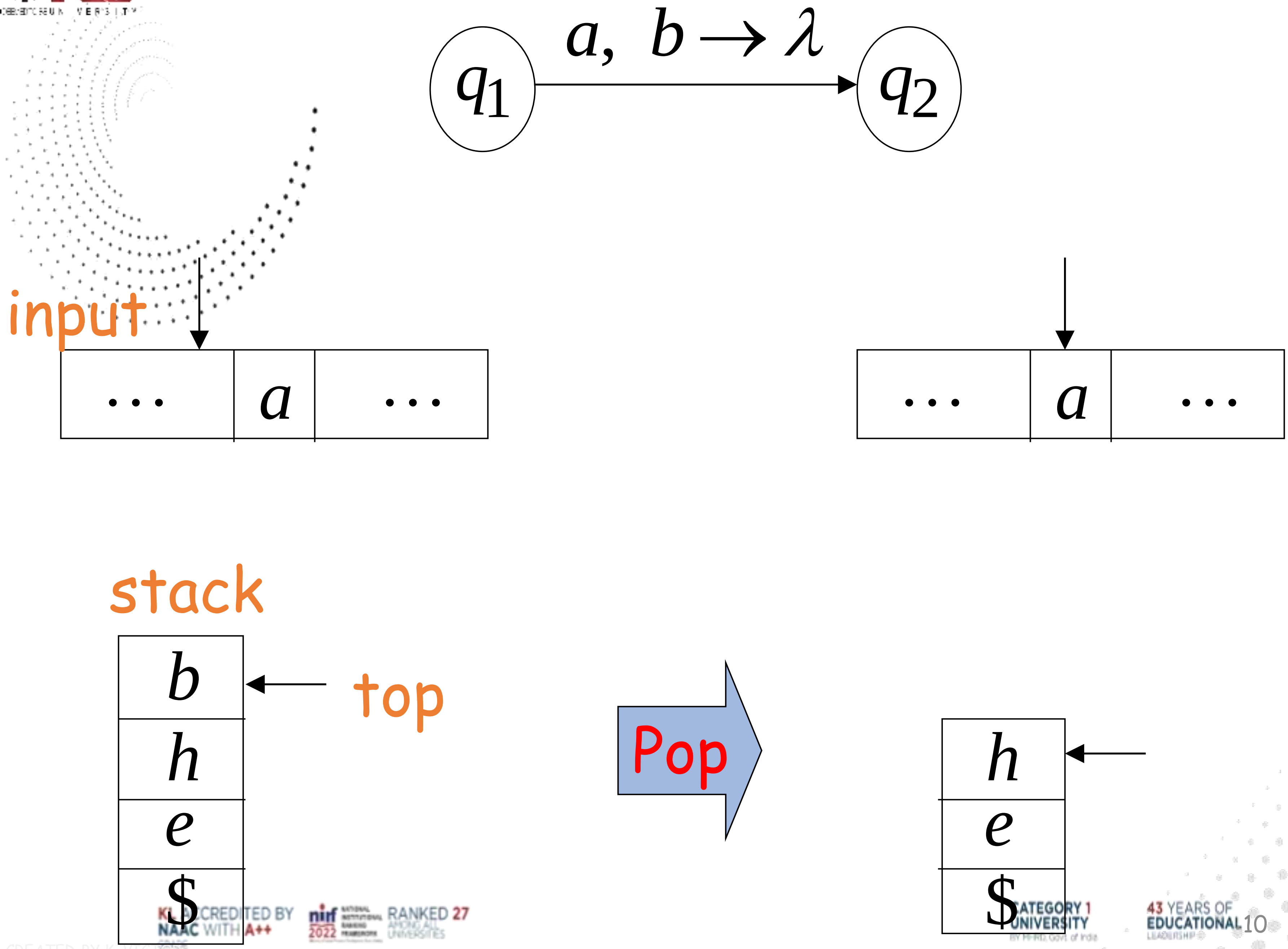
Alternatively

$$q_1 \xrightarrow{a, b / c} q_2$$

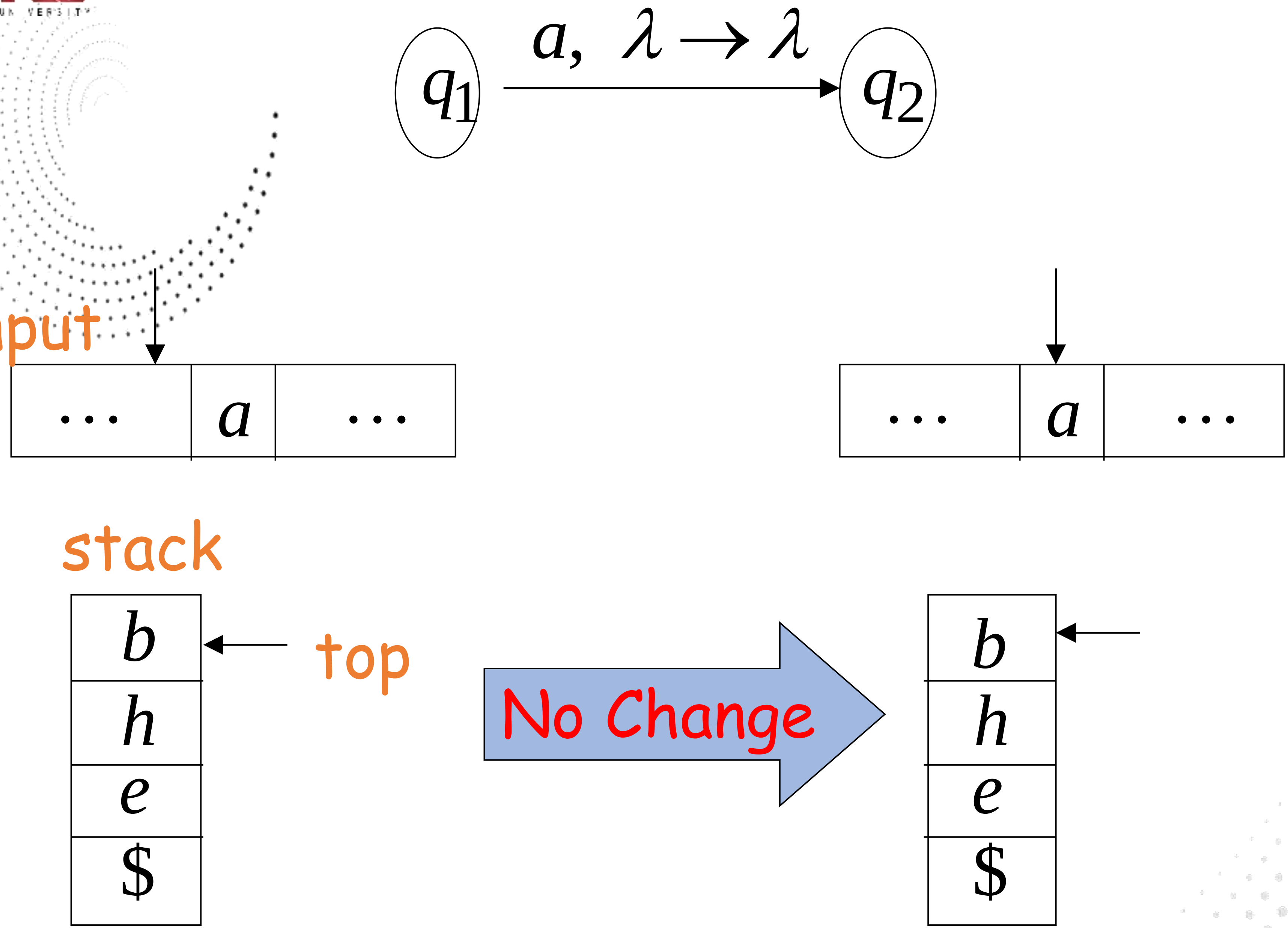

$a, \ b \to c$
q_1
q_2
input
$\cdots$ a $\cdots$
$\cdots$ a $\cdots$
stack
top
Replace
b
h
e
$\$$
c
h
e
$\$$



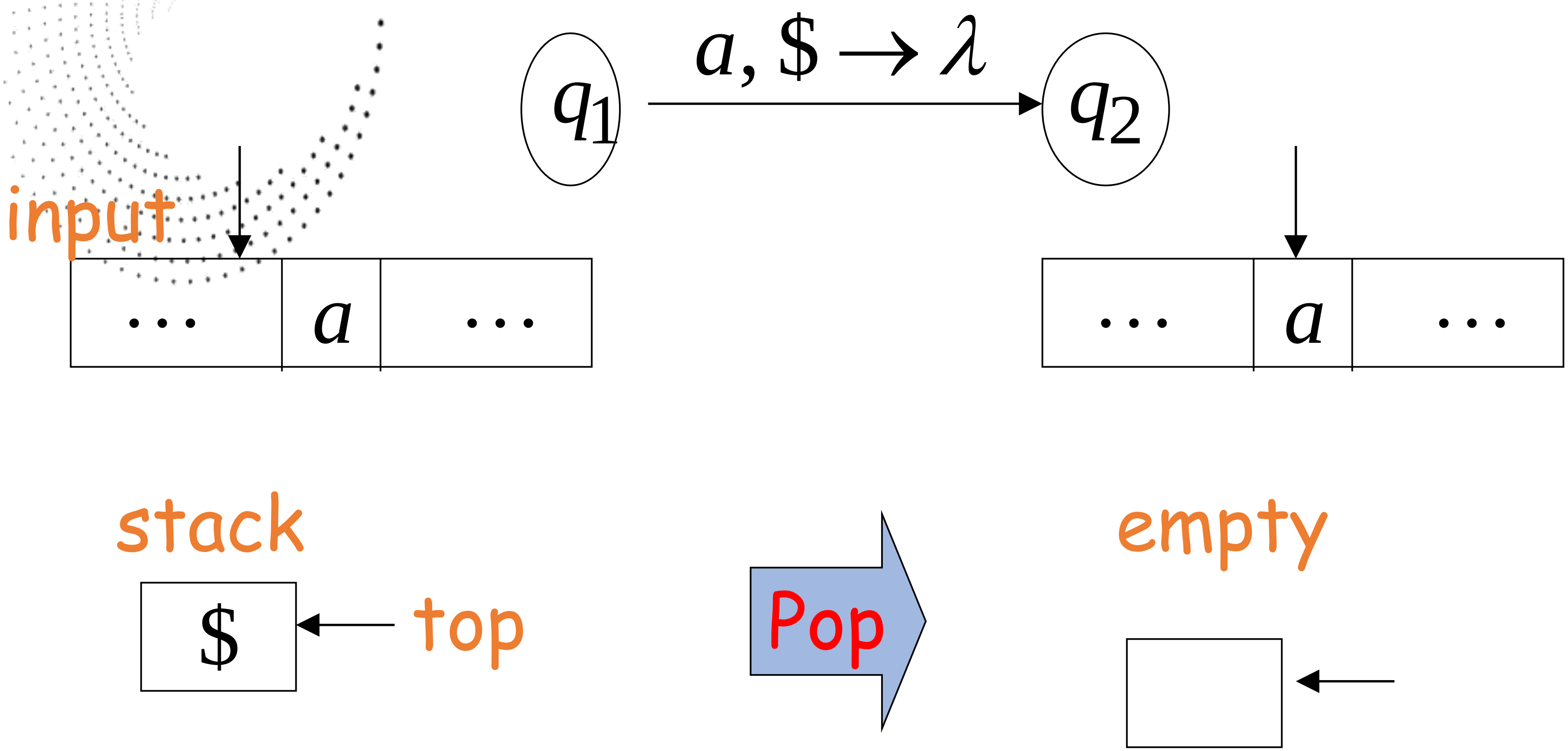
$a, \lambda \to c$
q_1
q_2
input
$\cdots$ a $\cdots$
$\cdots$ a $\cdots$
stack
top
b
h
e
$\$$
Push
c
b
h
e
$\$$



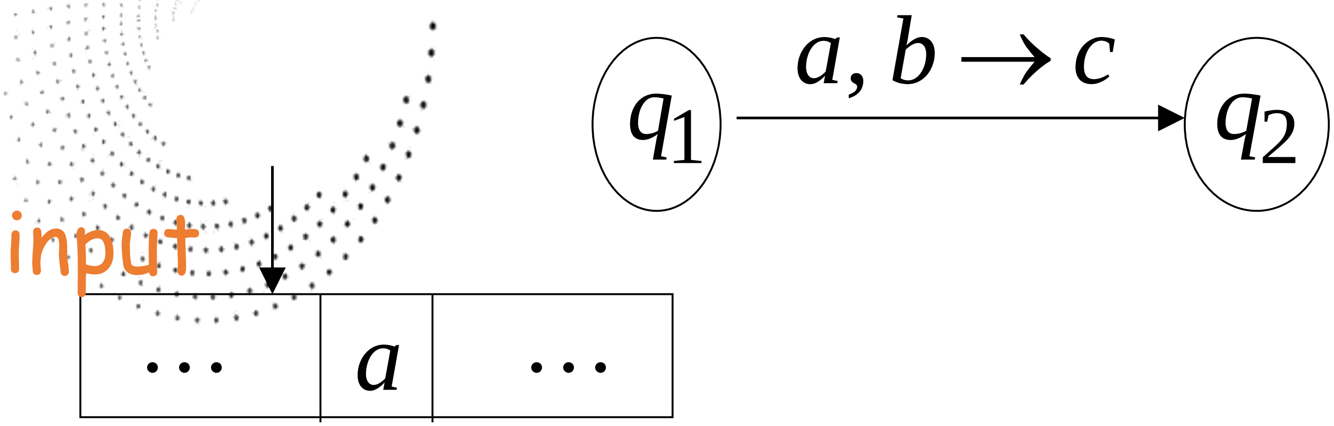
q_1
$a, \ b \rightarrow \lambda$
q_2
input
$\cdots$ a $\cdots$
$\cdots$ a $\cdots$
stack
top
b
h
e
$\$$
Pop
h
e
$\$$


q_1
$a, \lambda \rightarrow \lambda$
q_2
input
... a ...
... a ...
stack
b
h
e
$
top
No Change
b
h
e
$

A Possible Transition



A Bad Transition

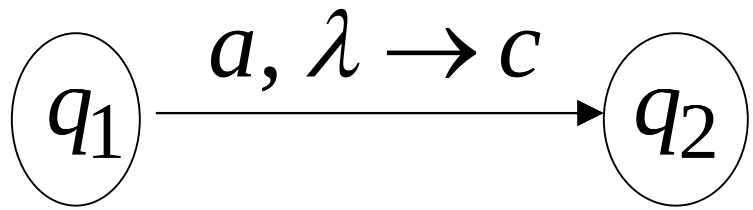


Empty stack

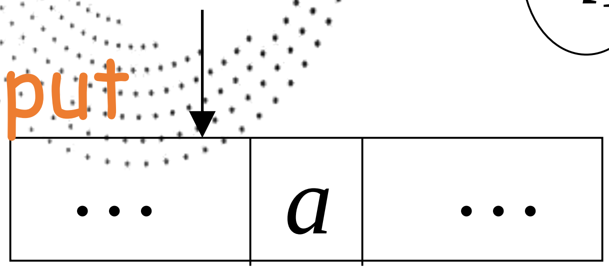
HALT

The automaton **Halts** in state q_1
and **Rejects** the input string

A Bad Transition



input



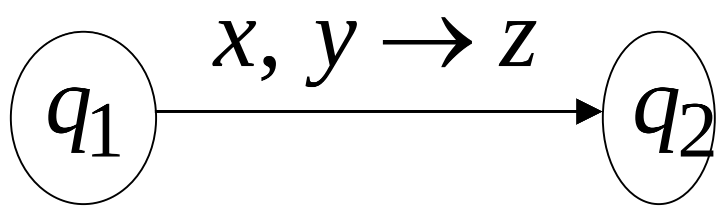
Empty stack

HALT

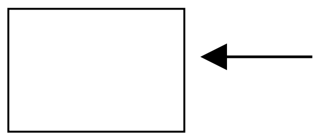
The automaton **Halts** in state q_1

and **Rejects** the input string

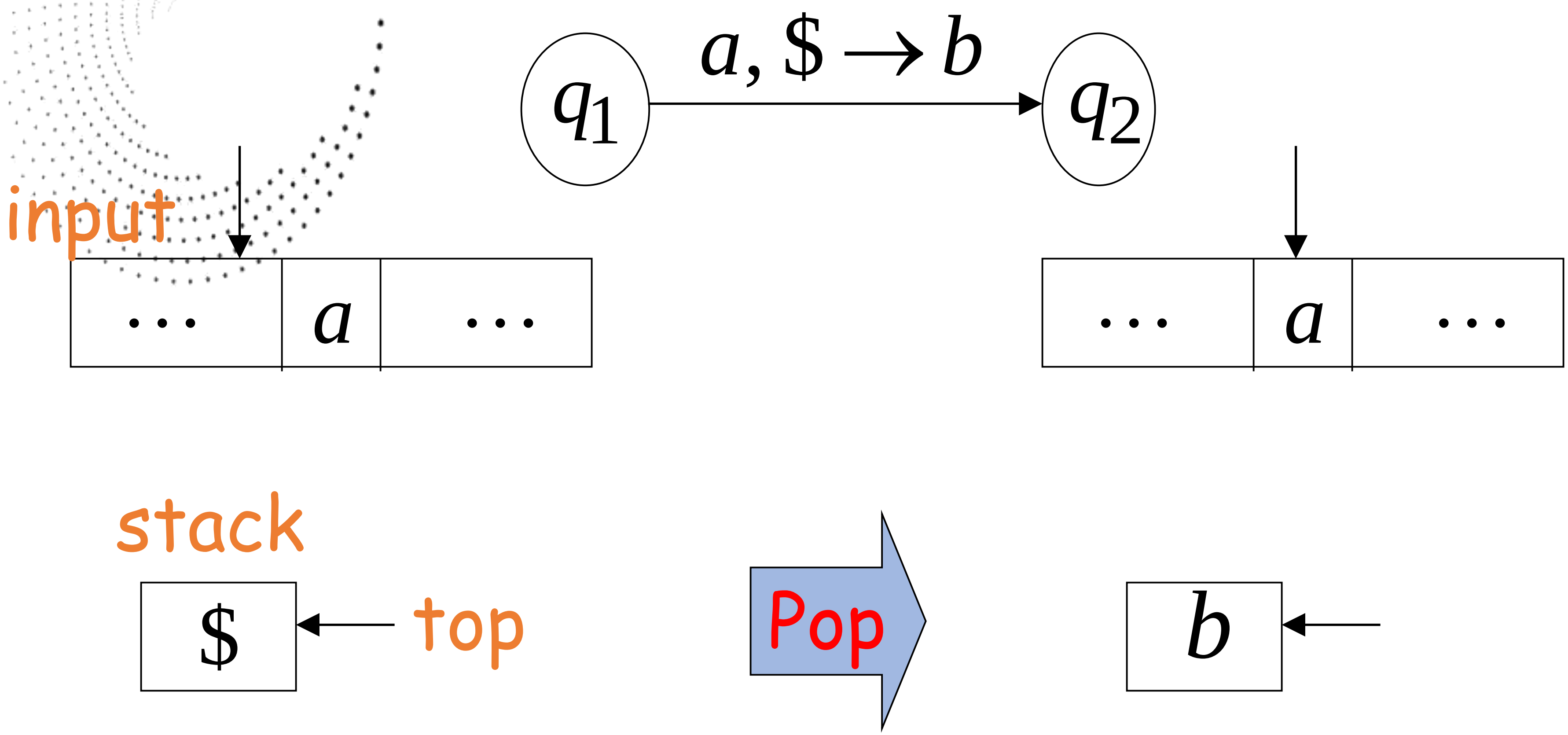
No transition is allowed to be followed
When the stack is empty



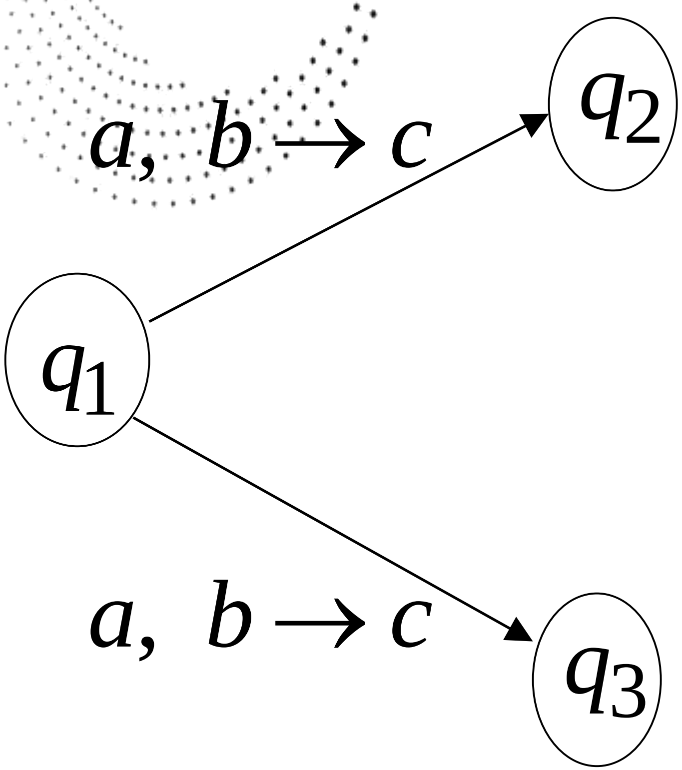
Empty stack

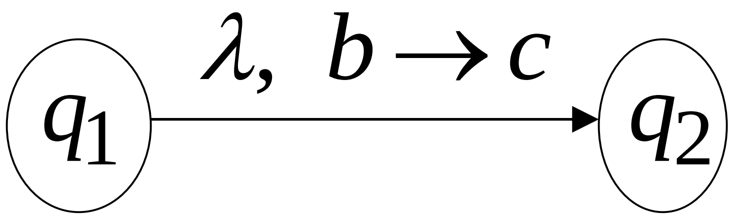


A Good Transition



Non-Determinism



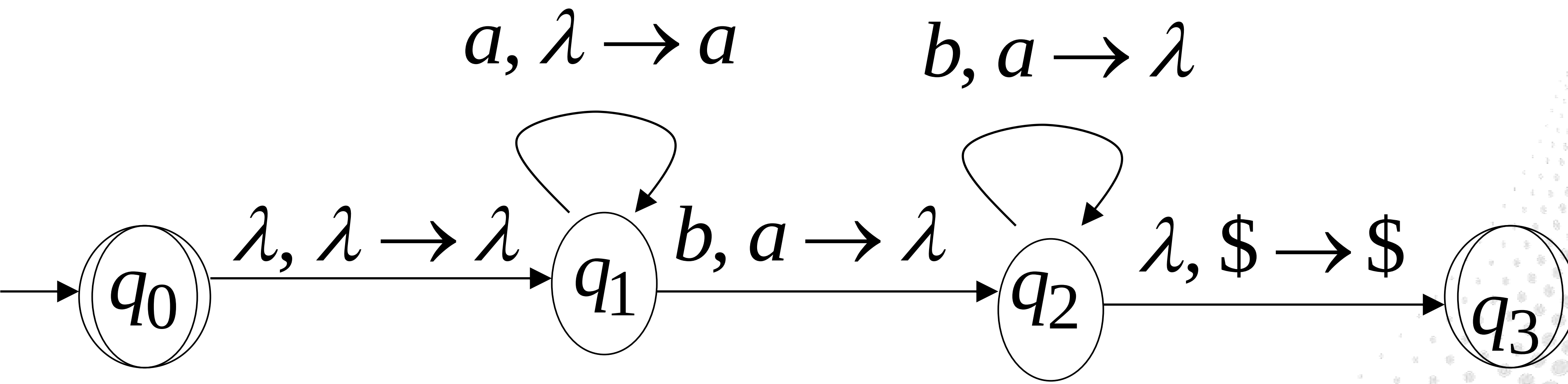


λ – transition

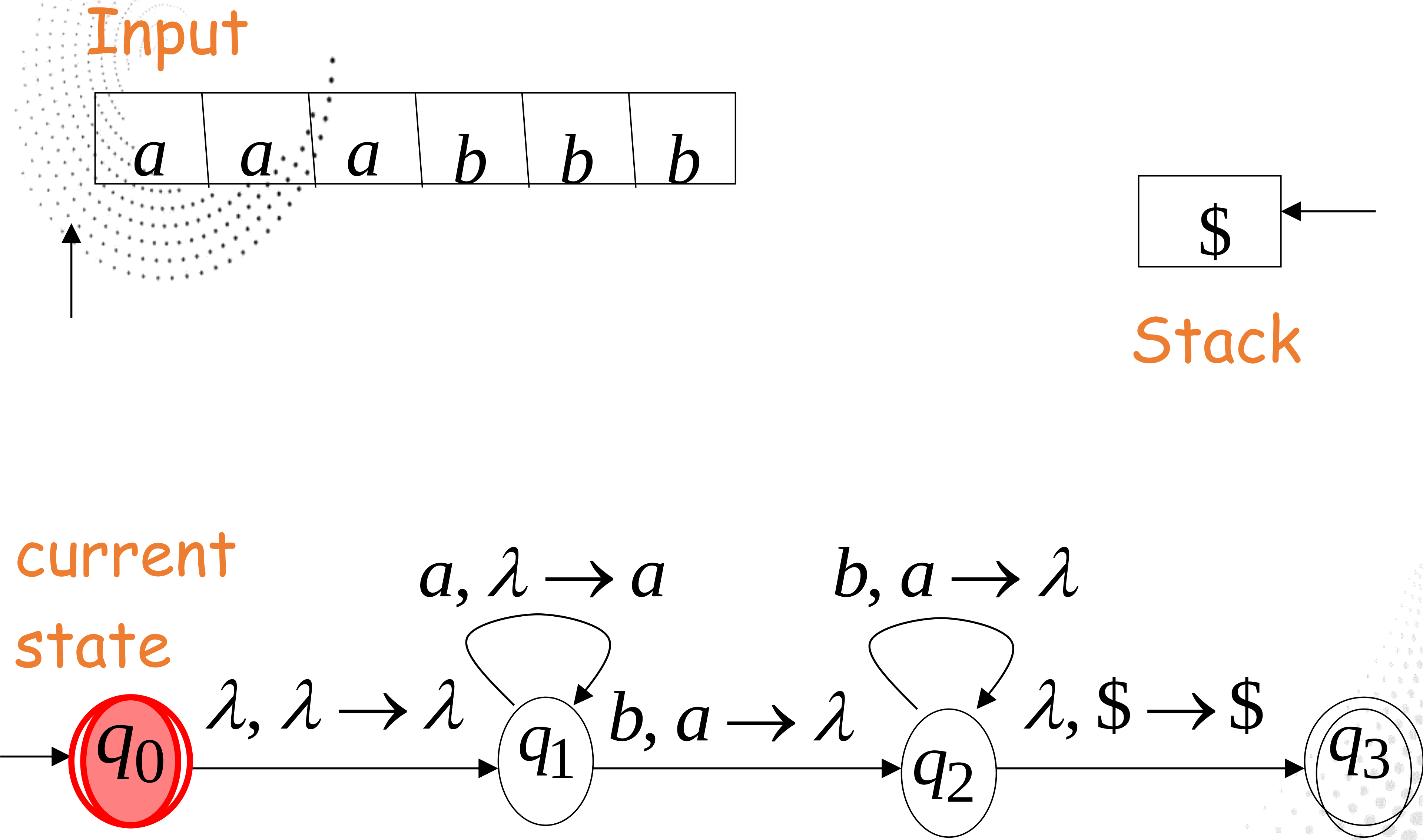
These are allowed transitions in a
Non-deterministic PDA (NPDA)

NPDA: Non-Deterministic PDA

Example:



Execution Example: Time 0



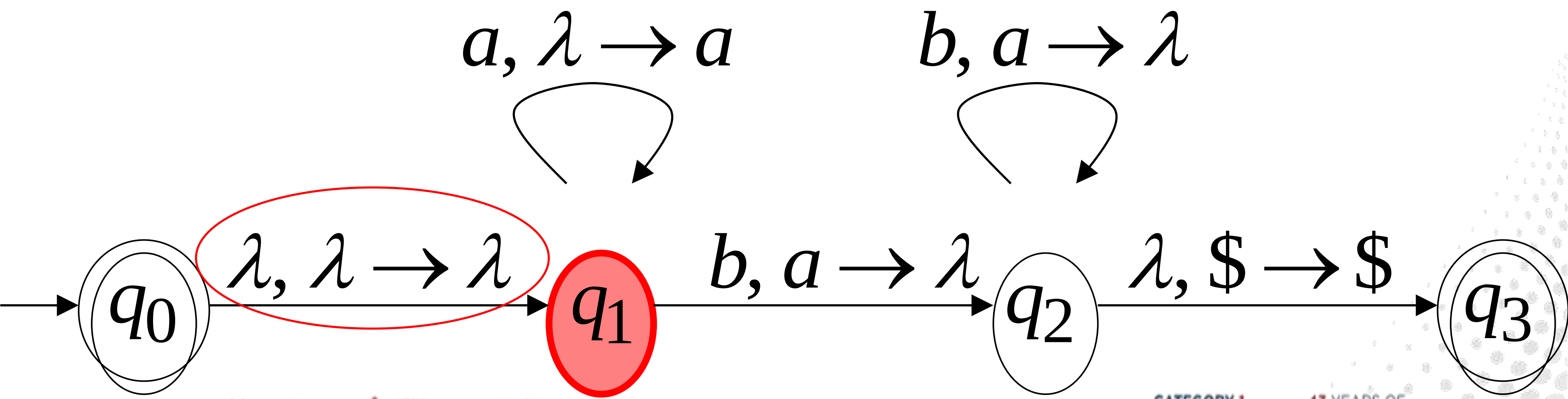
Time 1

Input

a	a	a	b	b	b

$\$$

Stack



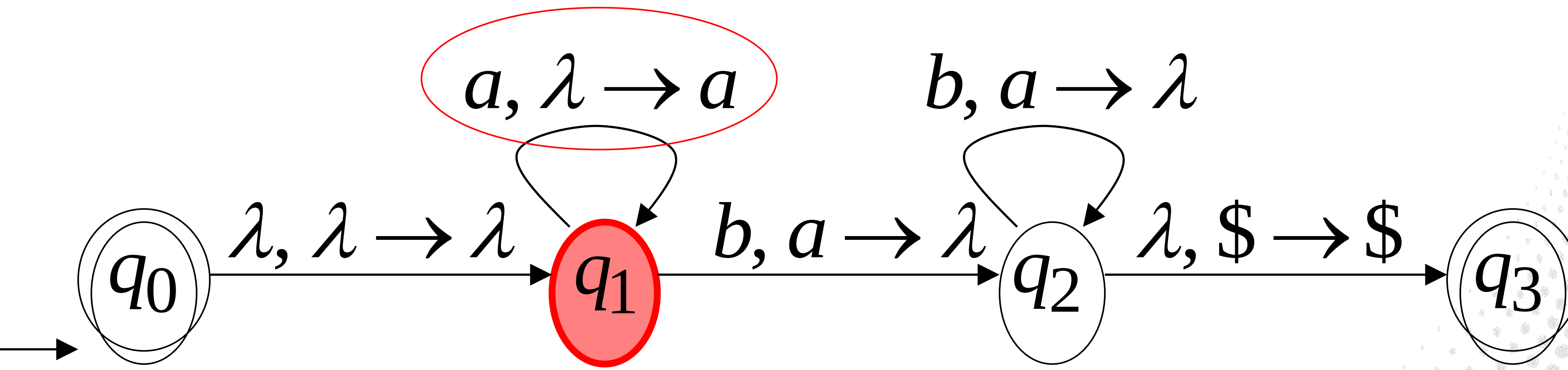
Time 2

Input

a	a	a	b	b	b

a
$\$$

Stack



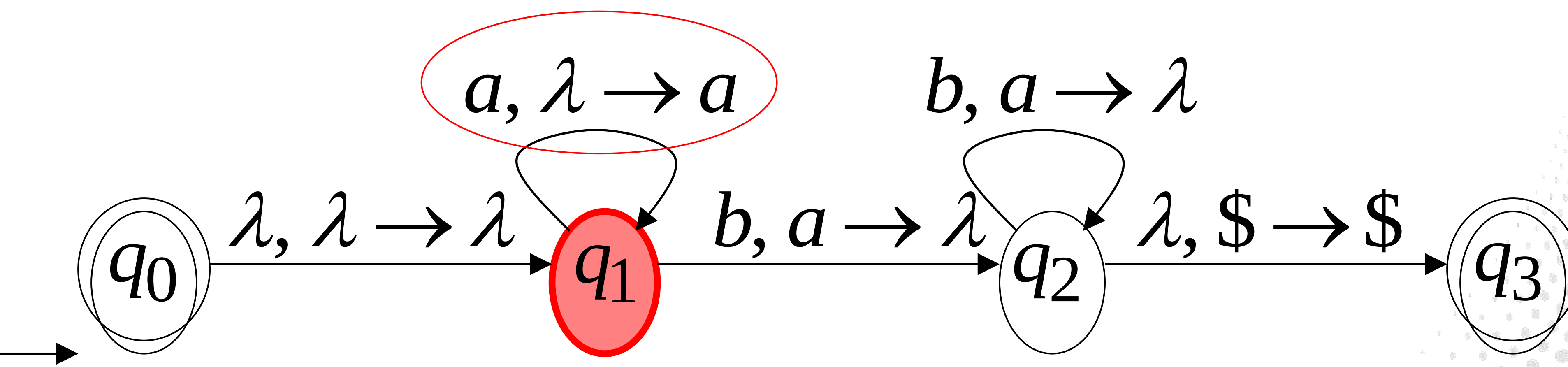
Time 3

Input

a	a	a	b	b	b

a
a
$\$$

Stack



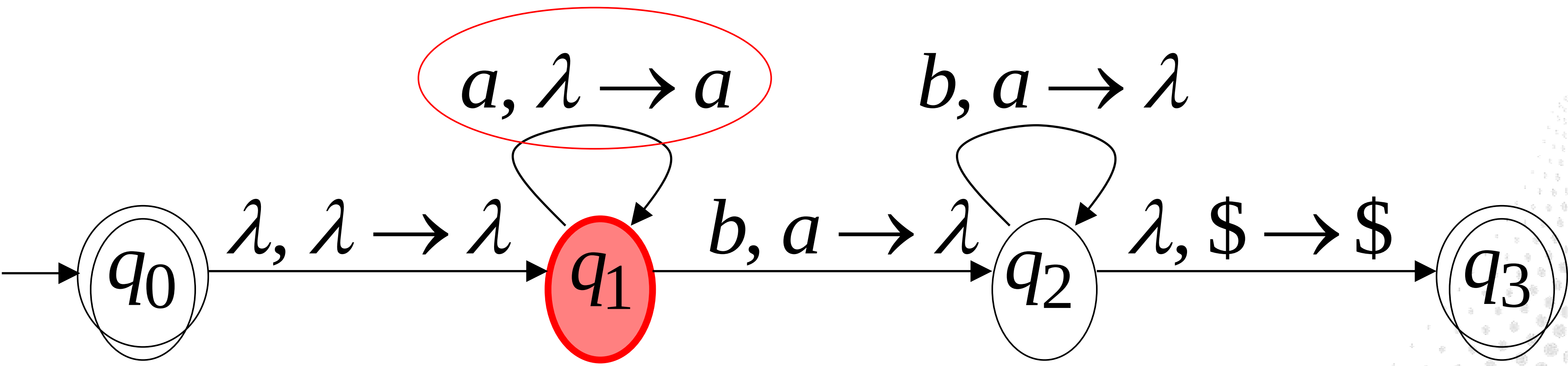
Time 4

Input

a	a	a	b	b	b

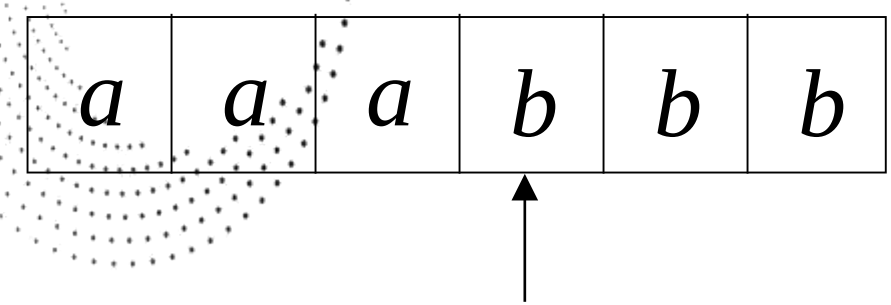
Stack
a
a
a
$\$$

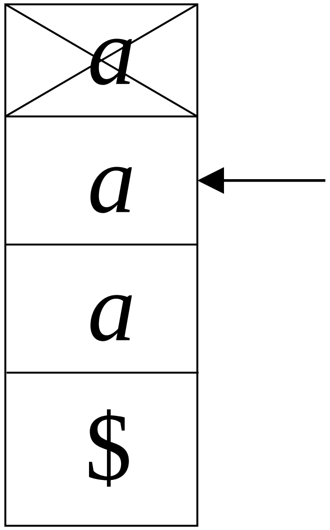
Stack



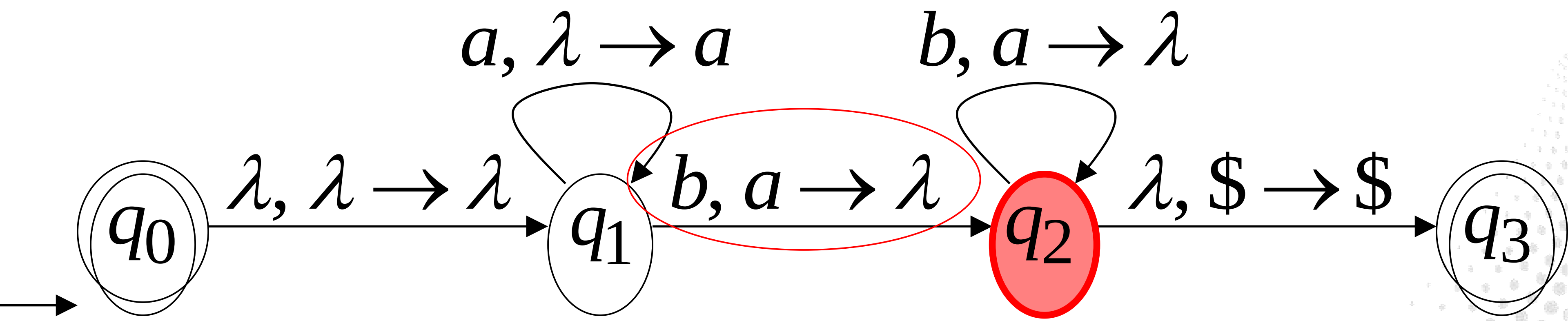
Time 5

Input



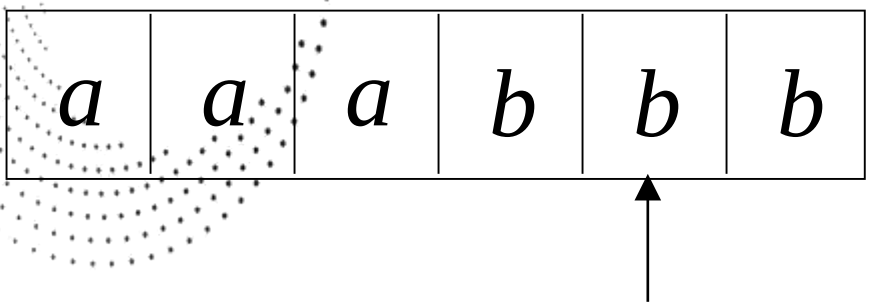


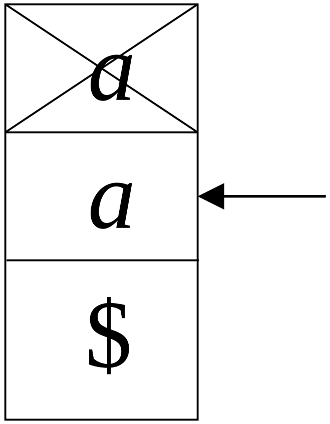
Stack



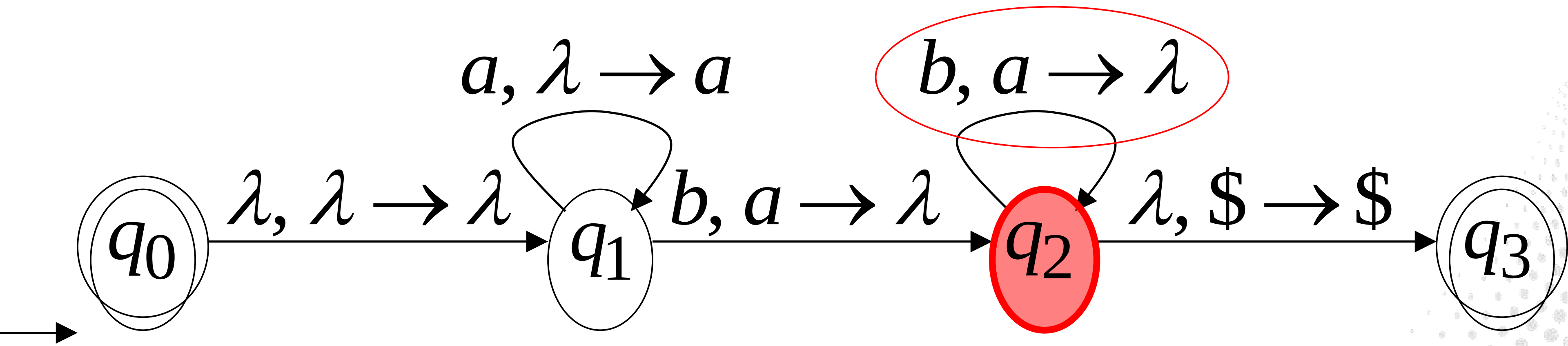
Time 6

Input

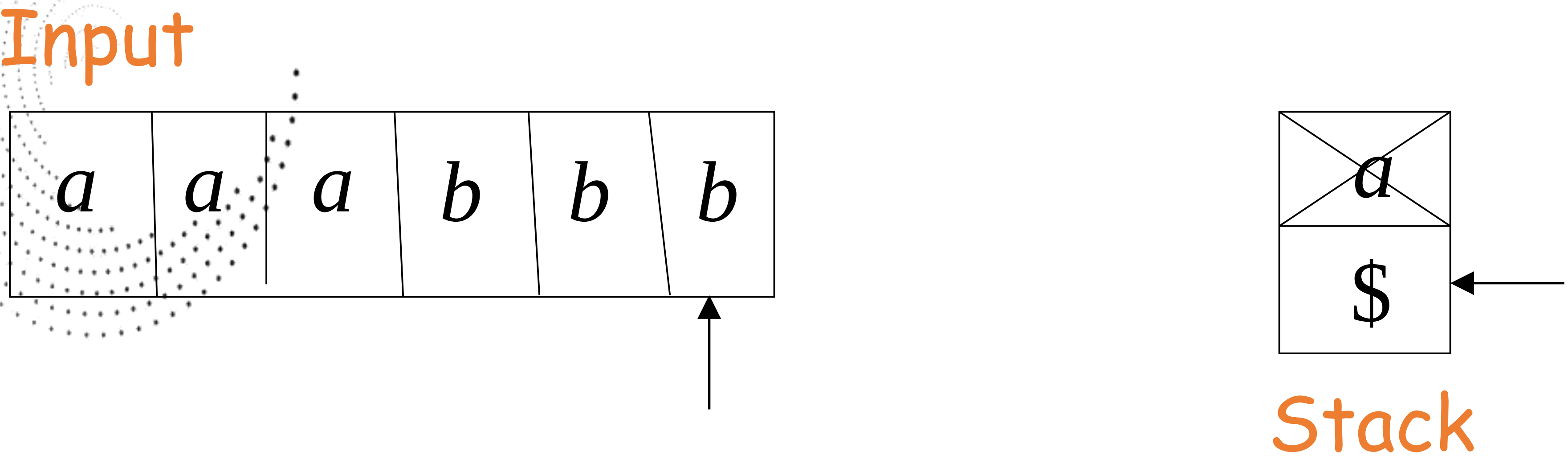




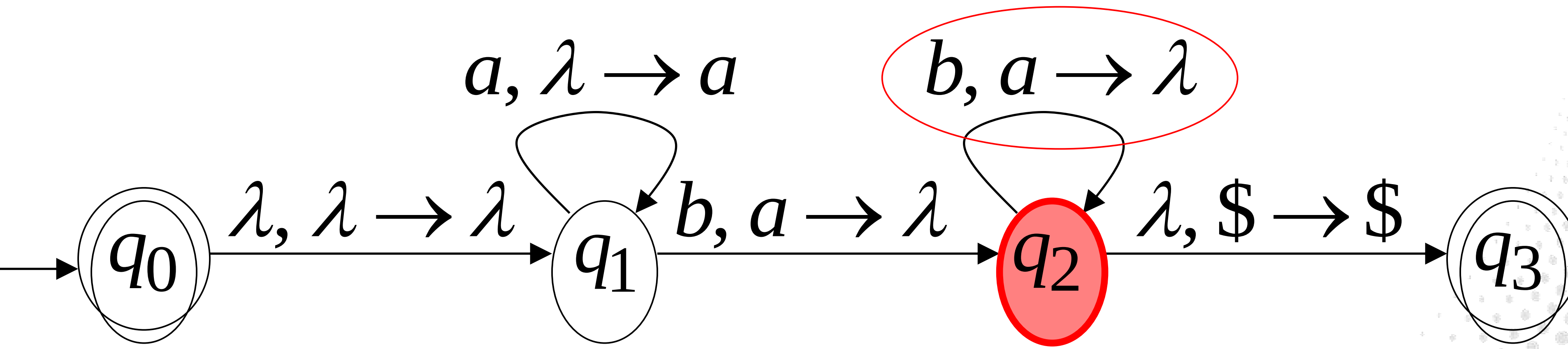
Stack



Time 7



Input
a
a
a
b
b
b
a
$
Stack



$a, \lambda \rightarrow a$
$b, a \rightarrow \lambda$
$\lambda, \lambda \rightarrow \lambda$
$b, a \rightarrow \lambda$
$\lambda, \$ \rightarrow \$$
q_0
q_1
q_2
q_3

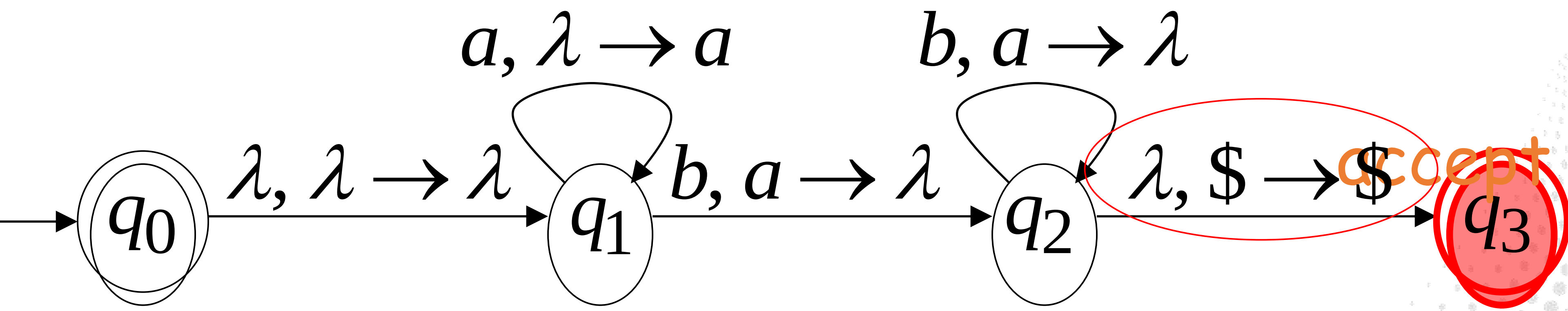
Time 8

Input

a	a	a	b	b	b

$\$$

Stack



A string is accepted if there is a computation such that:

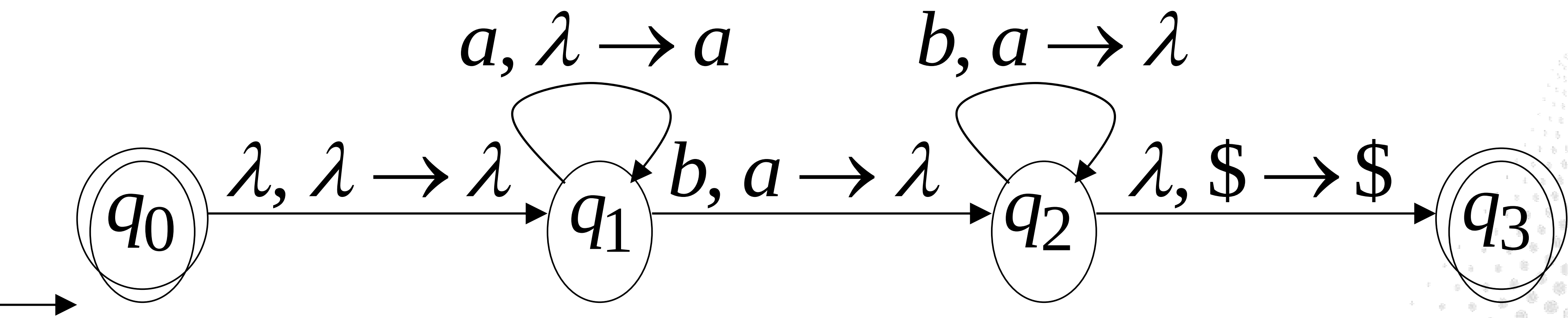
All the input is consumed

AND

The last state is a final state

At the end of the computation, we do not care about the stack contents

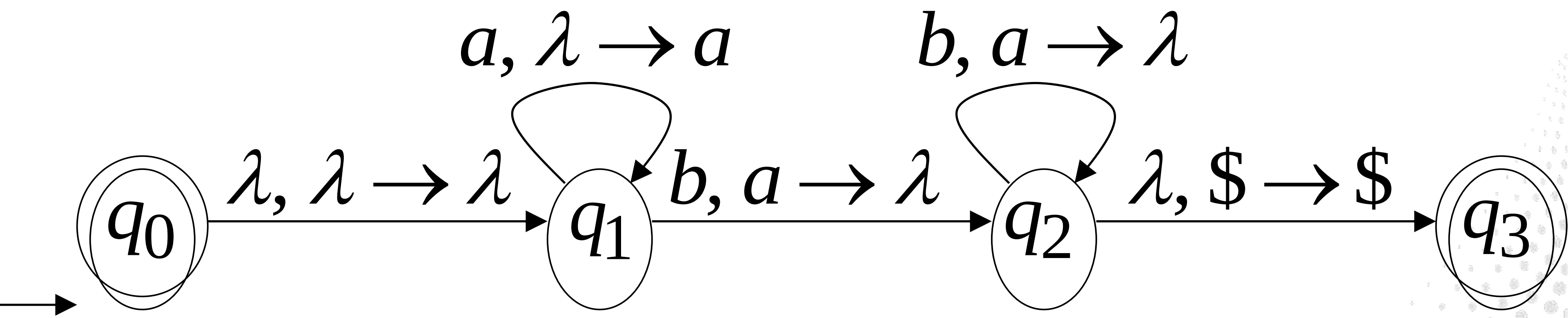
The input string $aaabbb$
is accepted by the NPDA:



In general,

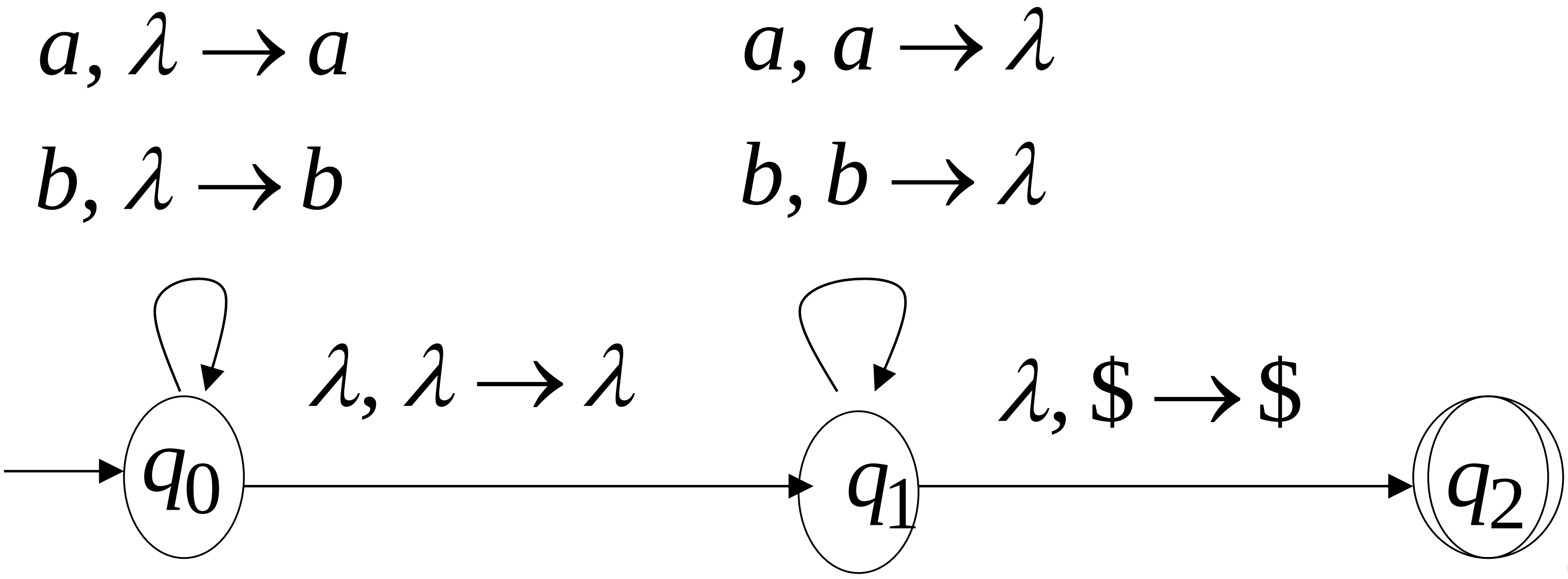
$$L = \{a^n b^n : n \geq 0\}$$

is the language accepted by the NPDA:



Another NPDA example

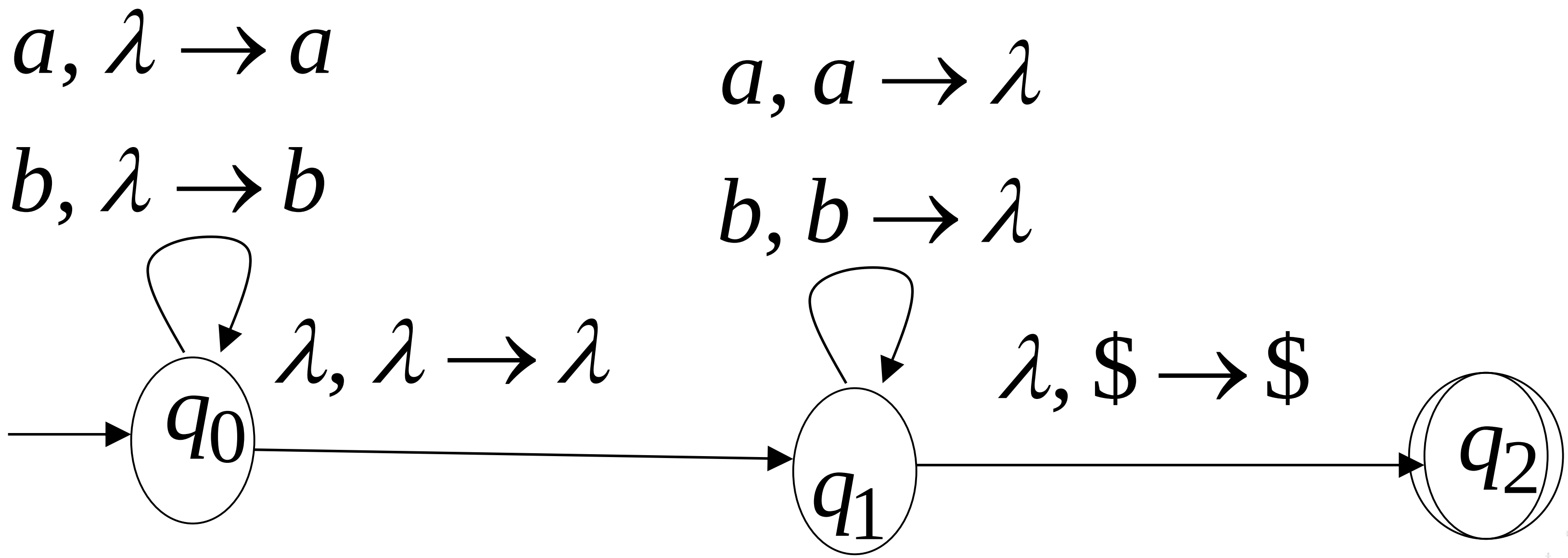
NPDA M



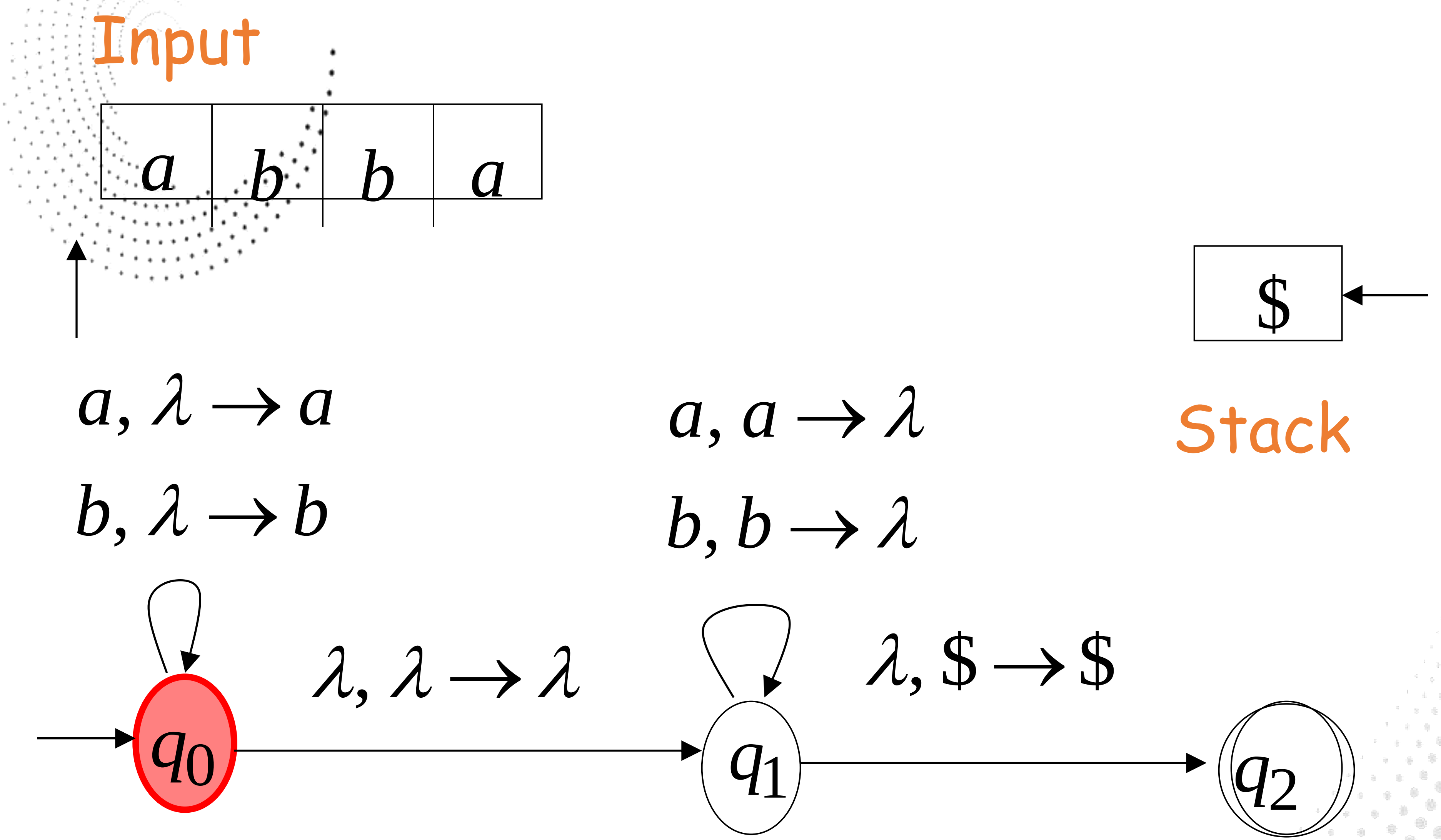
Another NPDA example

NPDA M

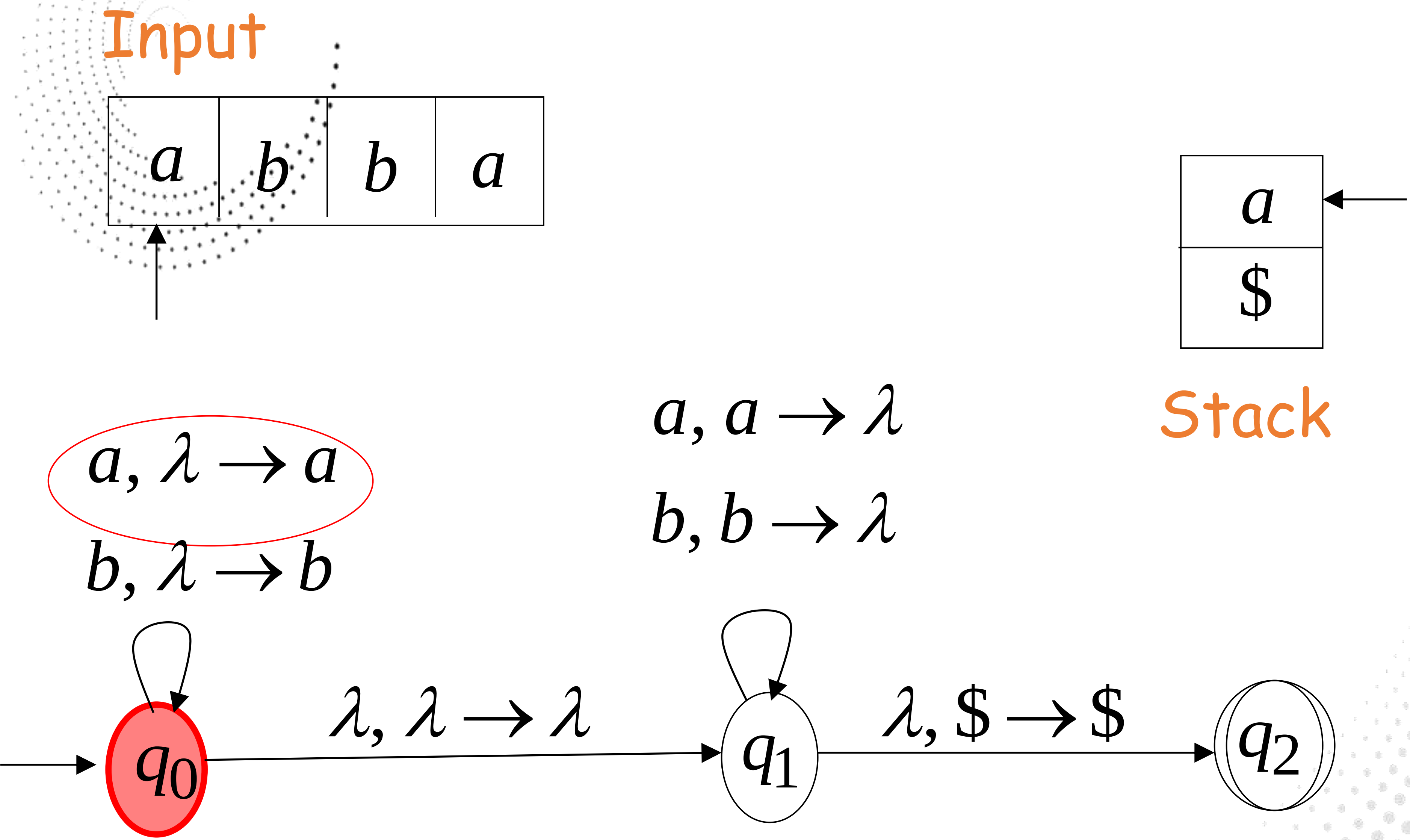
$$L(M) = \{ww^R\}$$



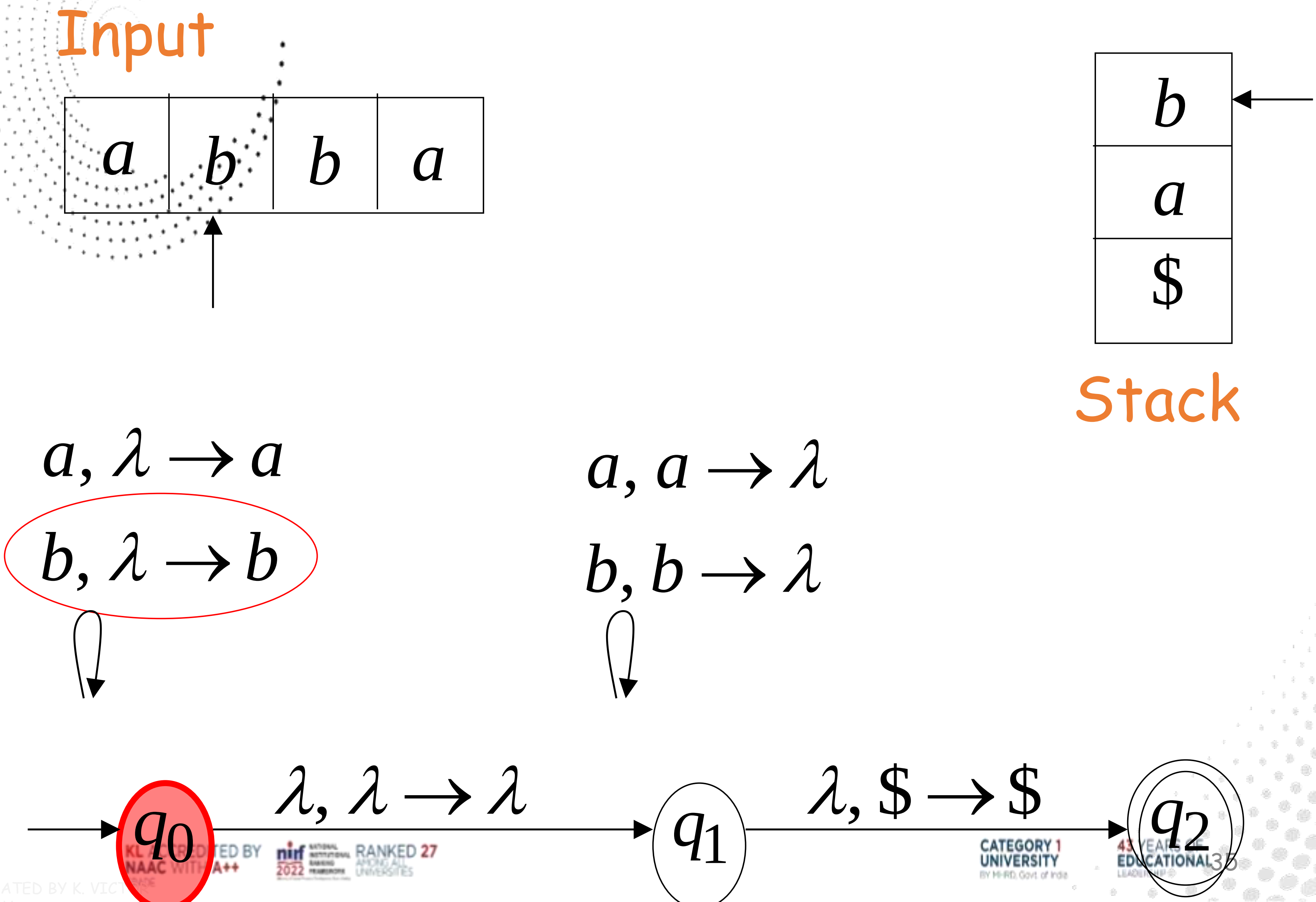
Execution Example: Time 0



Time 1


Input
a b b a
a
$
Stack
$a, \lambda \to a$
$b, \lambda \to b$
$a, a \to \lambda$
$b, b \to \lambda$
$\lambda, \lambda \to \lambda$
$\lambda, \$ \to \$$
q_0
q_1
q_2

Time 2


Input
a b b a
b
a
$
Stack
$a, \lambda \rightarrow a$
$b, \lambda \rightarrow b$
$a, a \rightarrow \lambda$
$b, b \rightarrow \lambda$
$\lambda, \lambda \rightarrow \lambda$
$\lambda, \$ \rightarrow \$$
q_0
q_1
q_2

Time 3

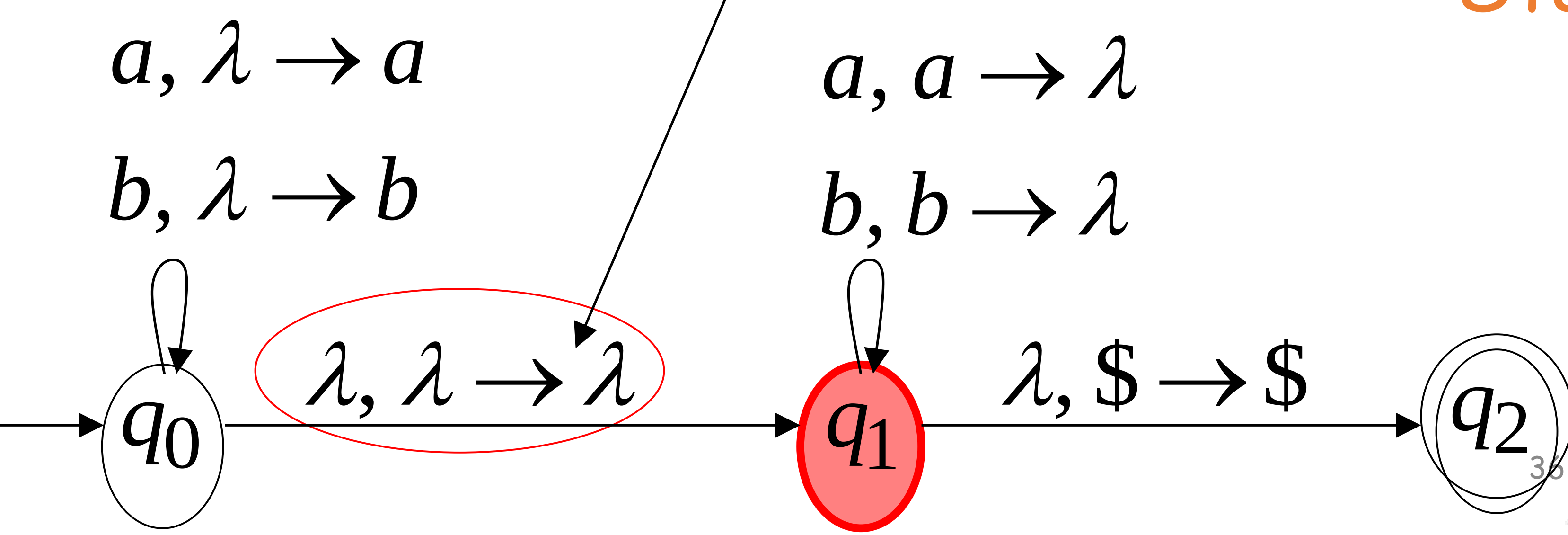
Input

a	b	b	a

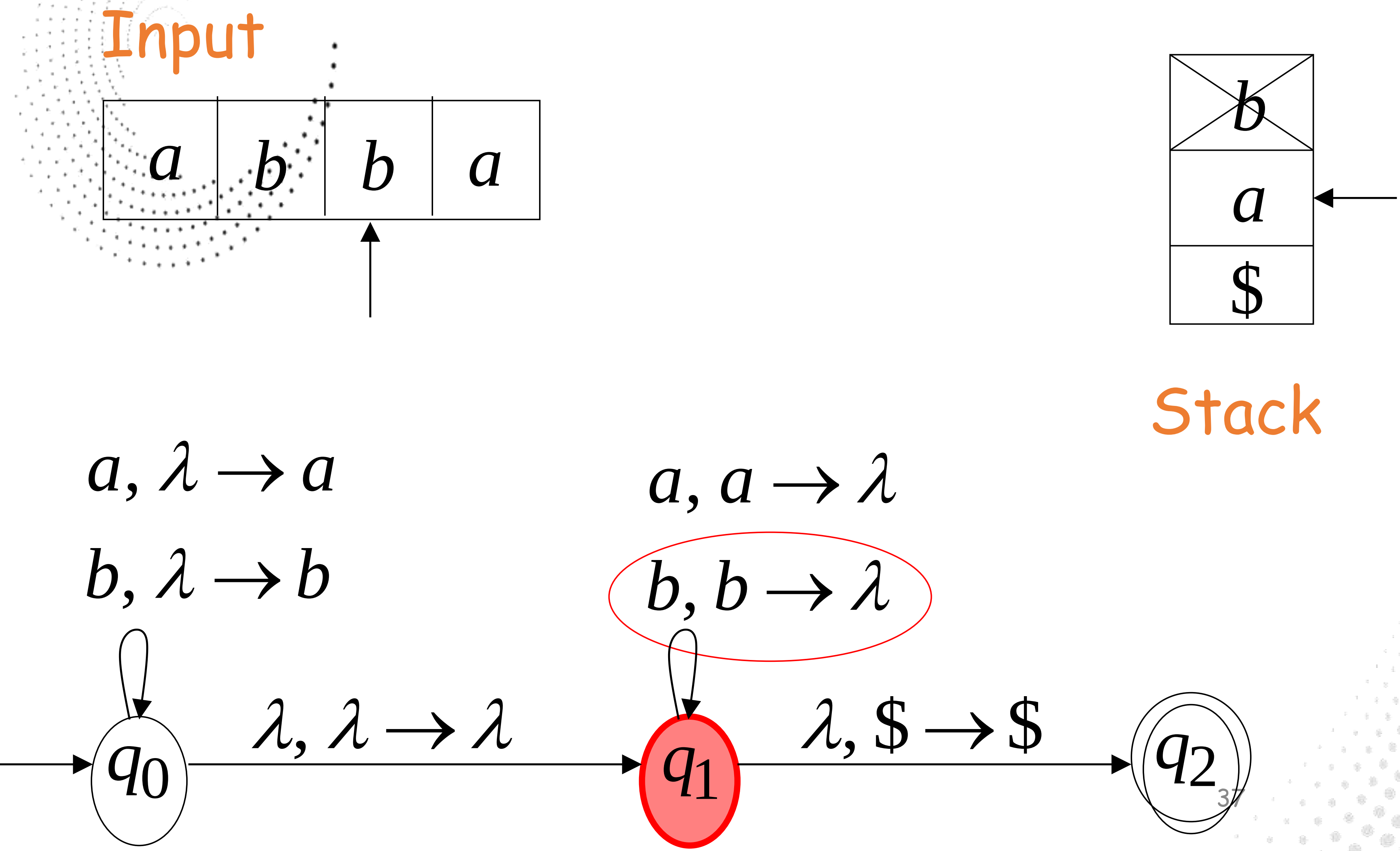
Guess the middle of string

b
a
$\$$

Stack

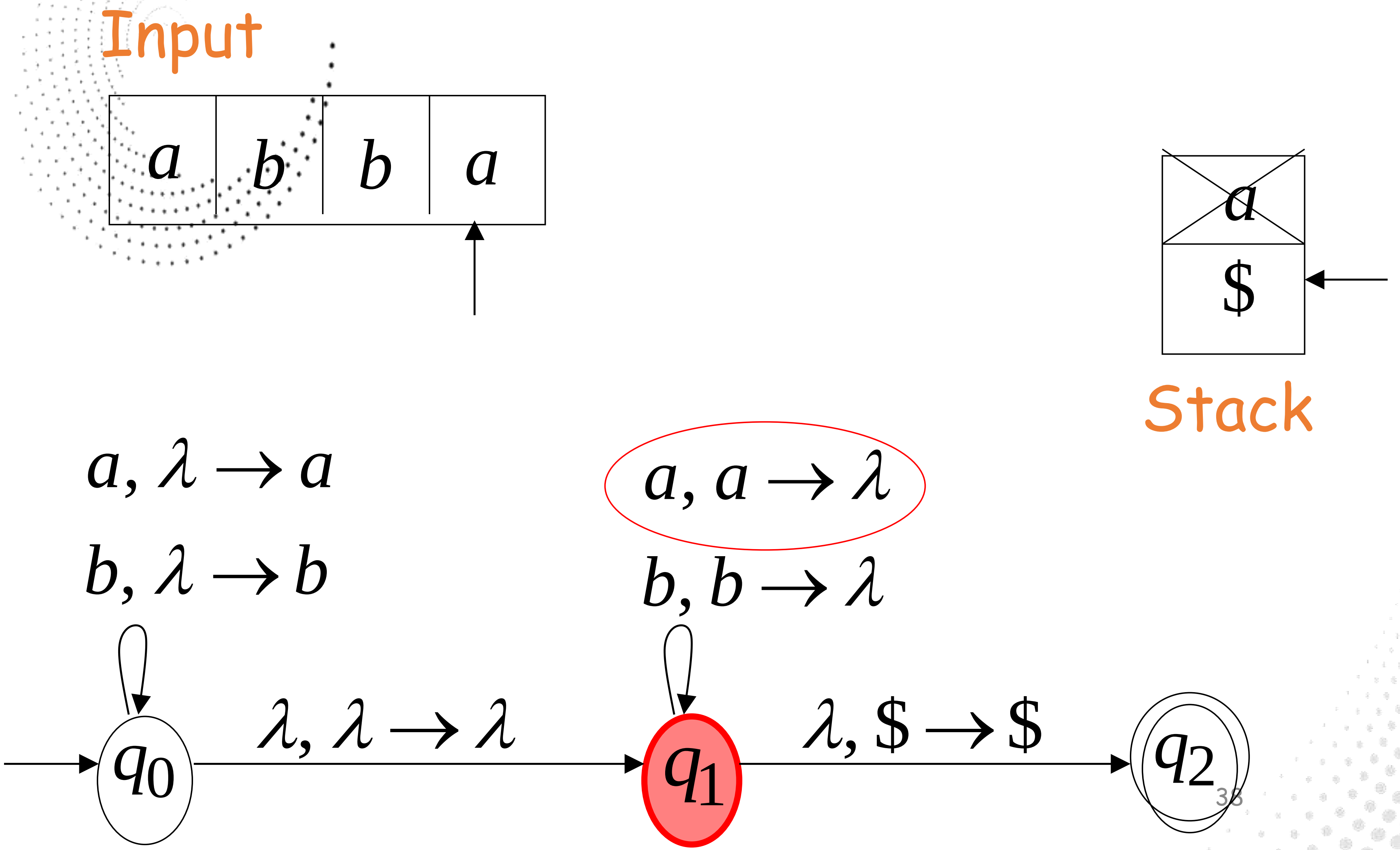


Time 4

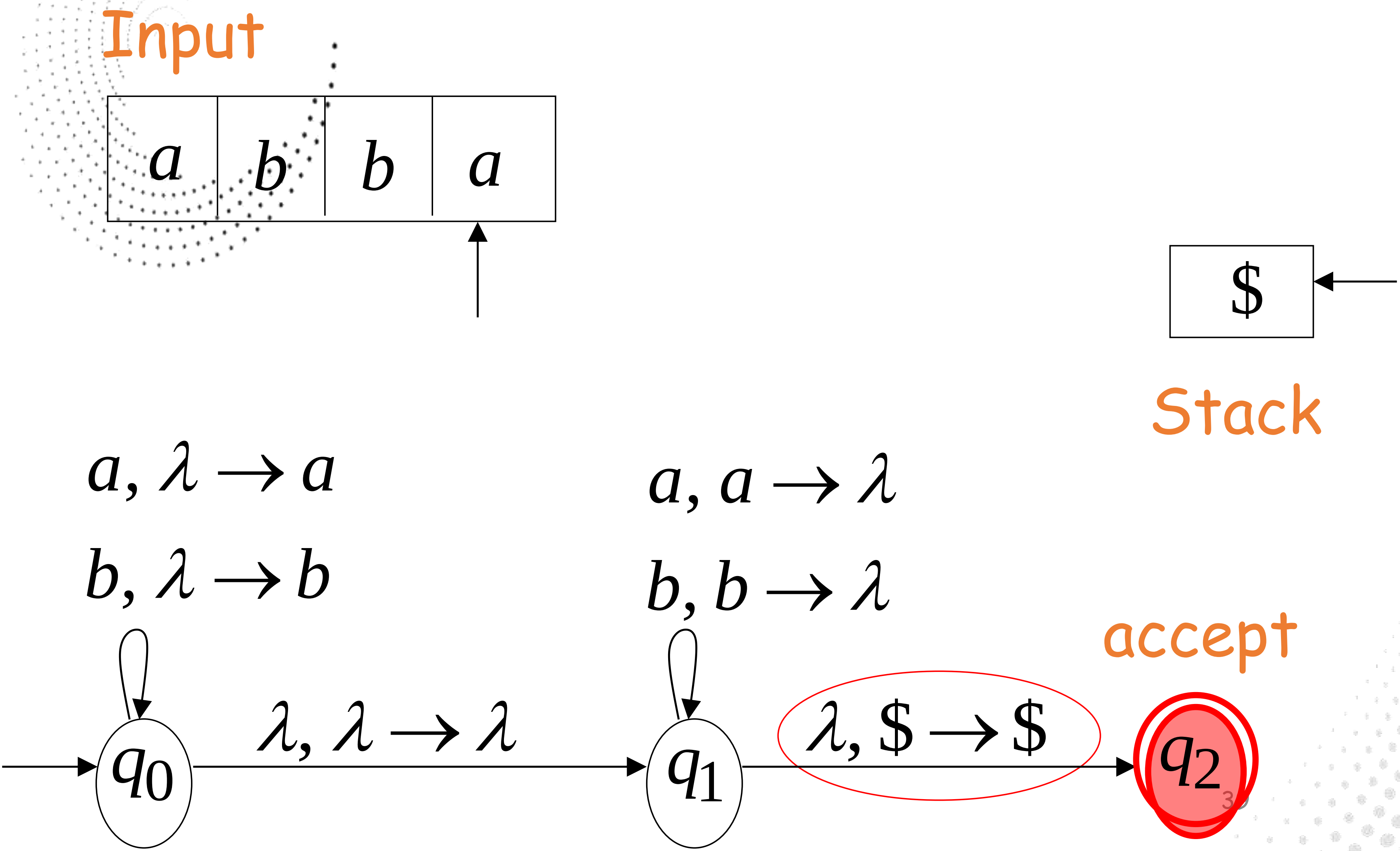


Input
a b b a
b
a
$
Stack
$a, \lambda \rightarrow a$
$b, \lambda \rightarrow b$
$a, a \rightarrow \lambda$
$b, b \rightarrow \lambda$
$\lambda, \lambda \rightarrow \lambda$
$\lambda, \$ \rightarrow \$$
q_0
q_1
q_2

Time 5


Input
a b b a
a
$
Stack
$a, \lambda \rightarrow a$
$b, \lambda \rightarrow b$
$a, a \rightarrow \lambda$
$b, b \rightarrow \lambda$
$\lambda, \lambda \rightarrow \lambda$
$\lambda, \$ \rightarrow \$$
q_0
q_1
q_2

Time 6


Input
a b b a
$\$$
Stack
$a, \lambda \rightarrow a$
$b, \lambda \rightarrow b$
$a, a \rightarrow \lambda$
$b, b \rightarrow \lambda$
accept
$\lambda, \lambda \rightarrow \lambda$
$\lambda, \$ \rightarrow \$$
q_0
q_1
q_2

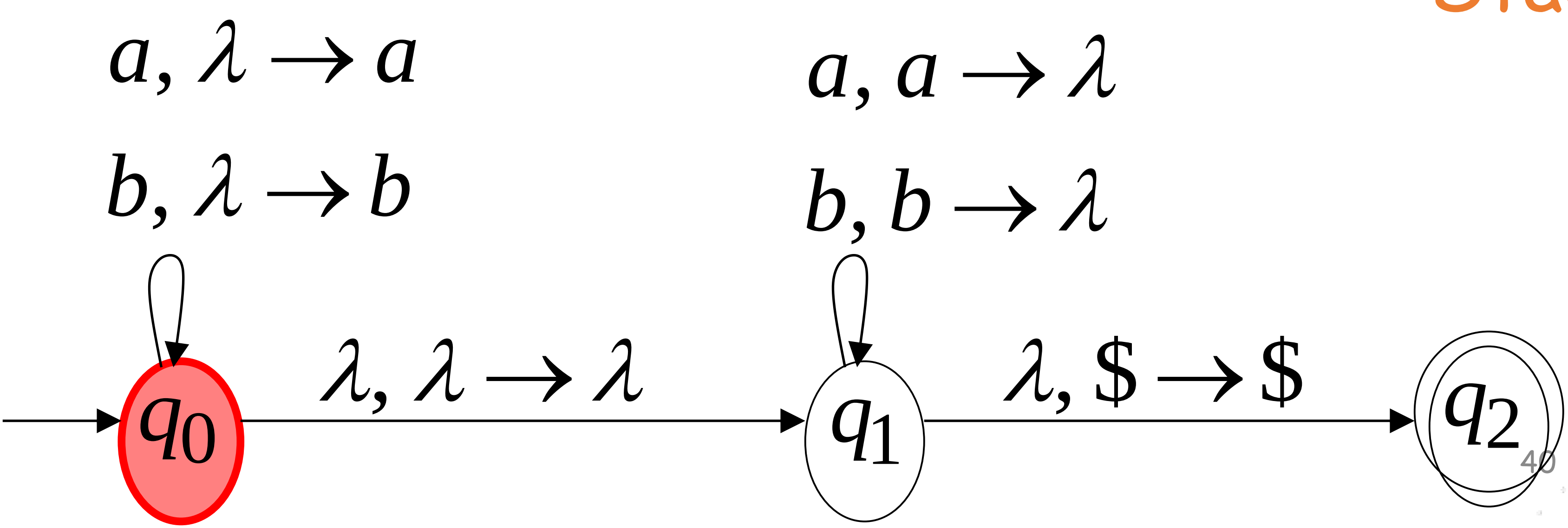
Rejection Example: Time 0

Input

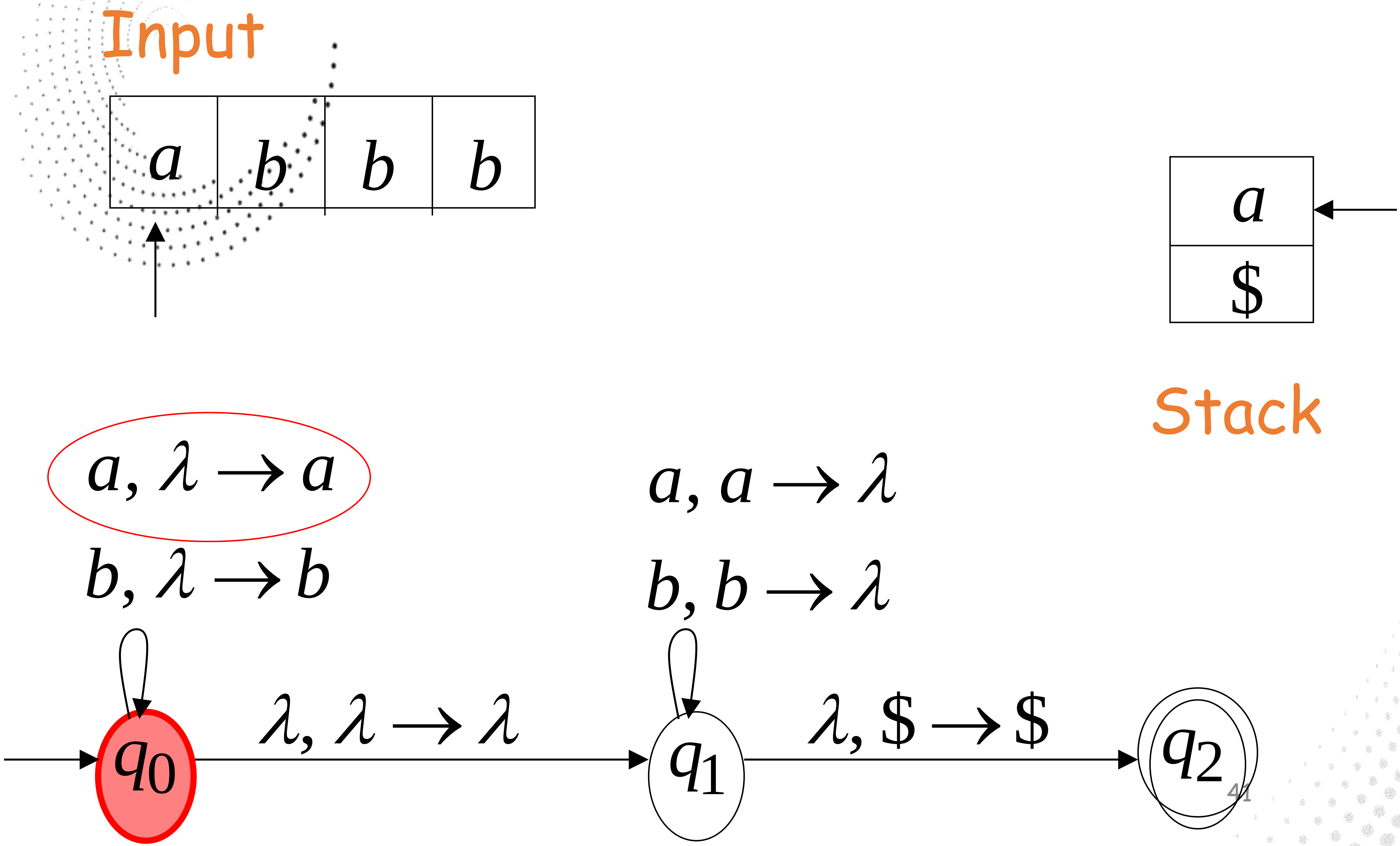
a	b	b	b

Stack

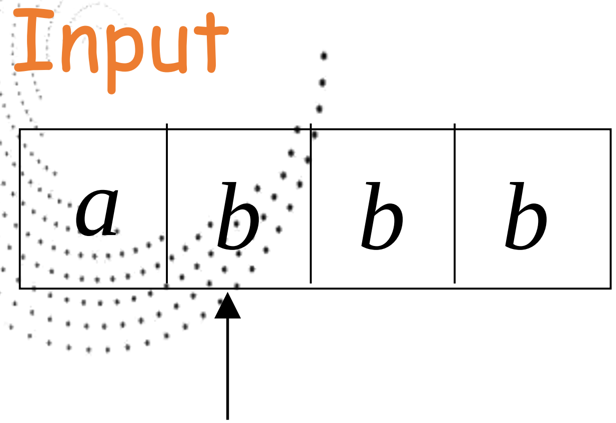
$\$$

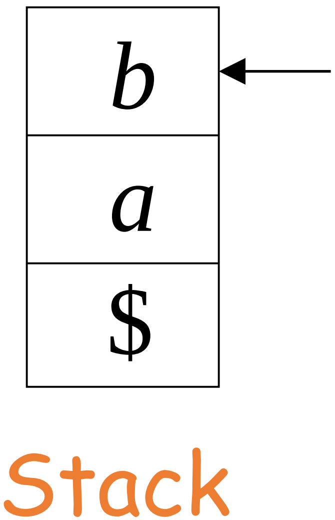


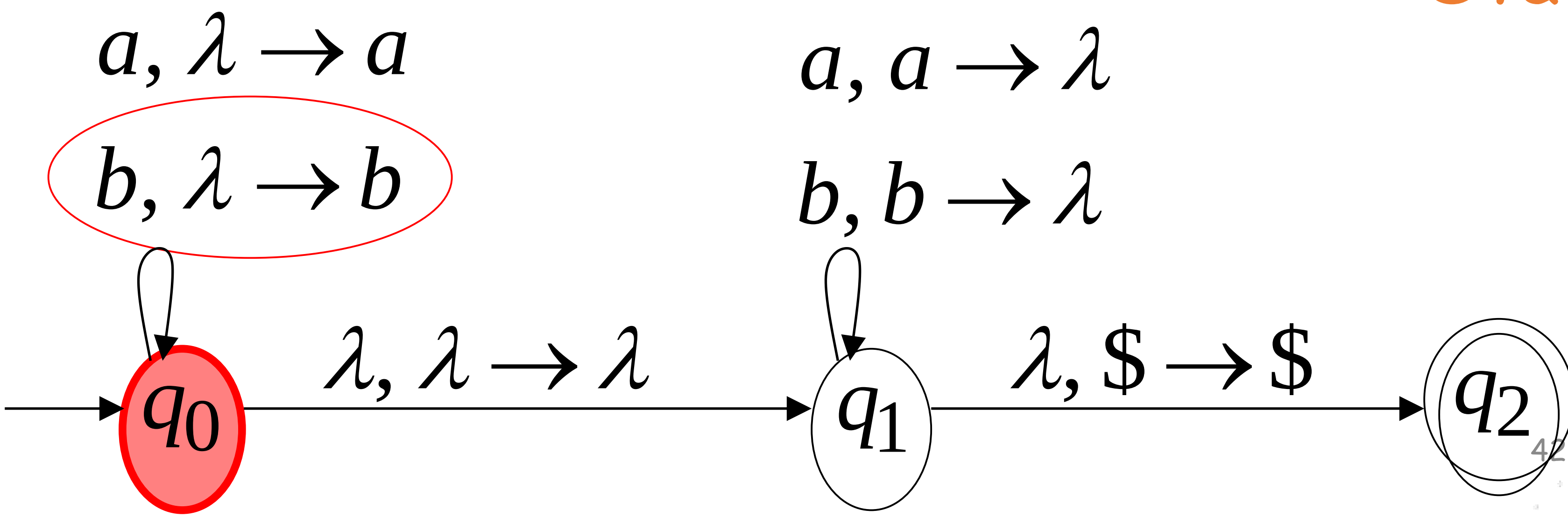
Time 1


Input
a b b b
a
$
Stack
$a, \lambda \rightarrow a$
$b, \lambda \rightarrow b$
$a, a \rightarrow \lambda$
$b, b \rightarrow \lambda$
$\lambda, \lambda \rightarrow \lambda$
$\lambda, \$ \rightarrow \$$
q_0
q_1
q_2

Time 2


Input
a
b
b
b


b
a
$
Stack


$a, \lambda \to a$
$b, \lambda \to b$
$a, a \to \lambda$
$b, b \to \lambda$
$\lambda, \lambda \to \lambda$
$\lambda, \$ \to \$$
q_0
q_1
q_2

Time 3

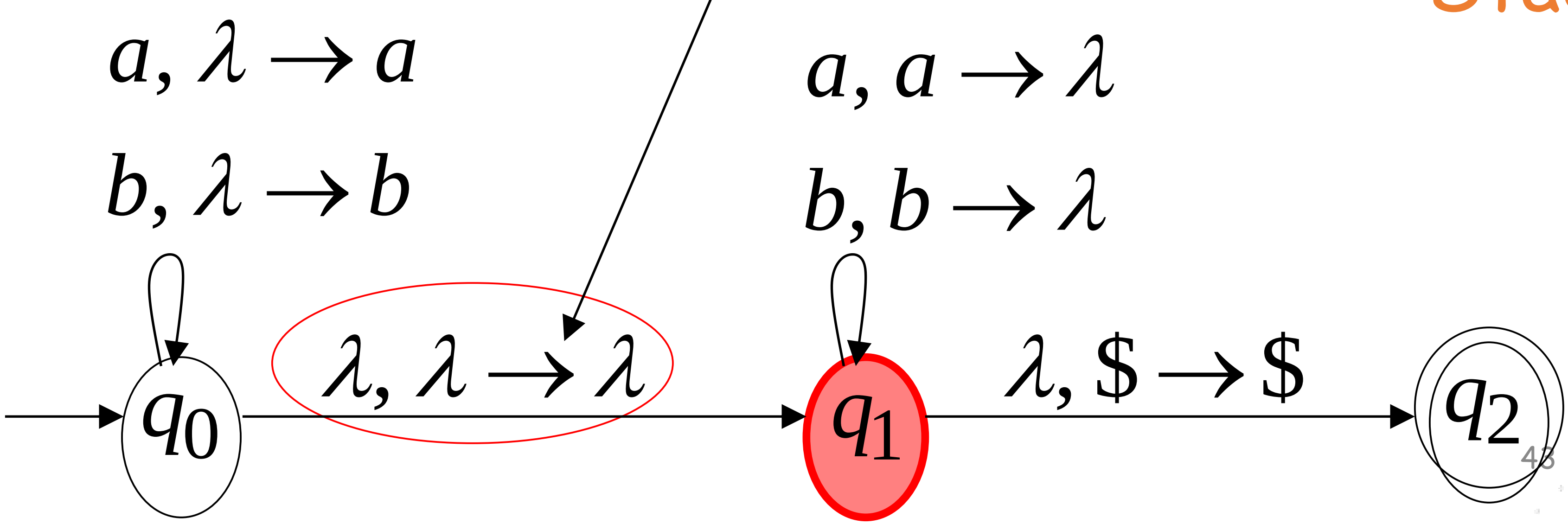
Input

a	b	b	b

Guess the middle of string

b
a
$\$$

Stack



Time 4

Input

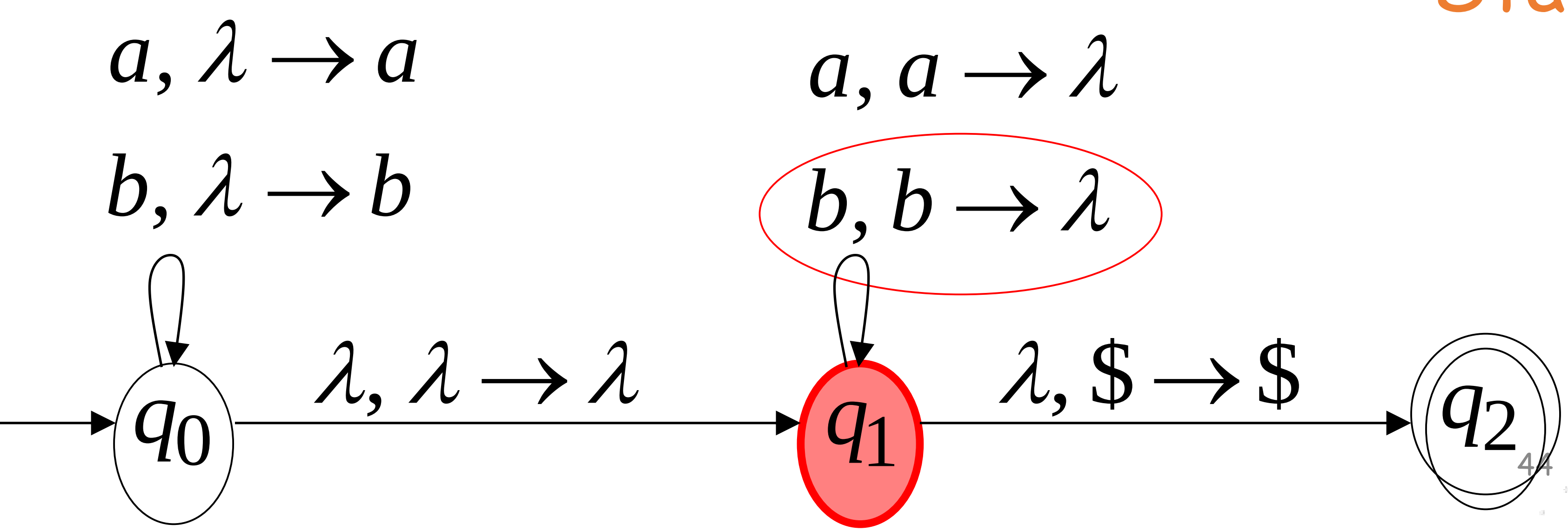
a	b	b	b

$\cancel{b}$

a

$\$$

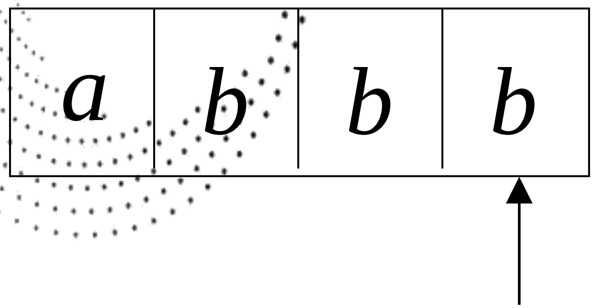
Stack



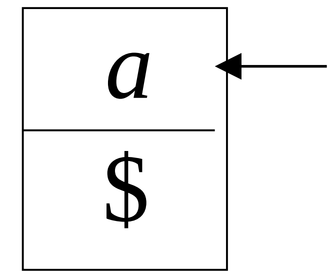
Time 5

There is no possible transition.

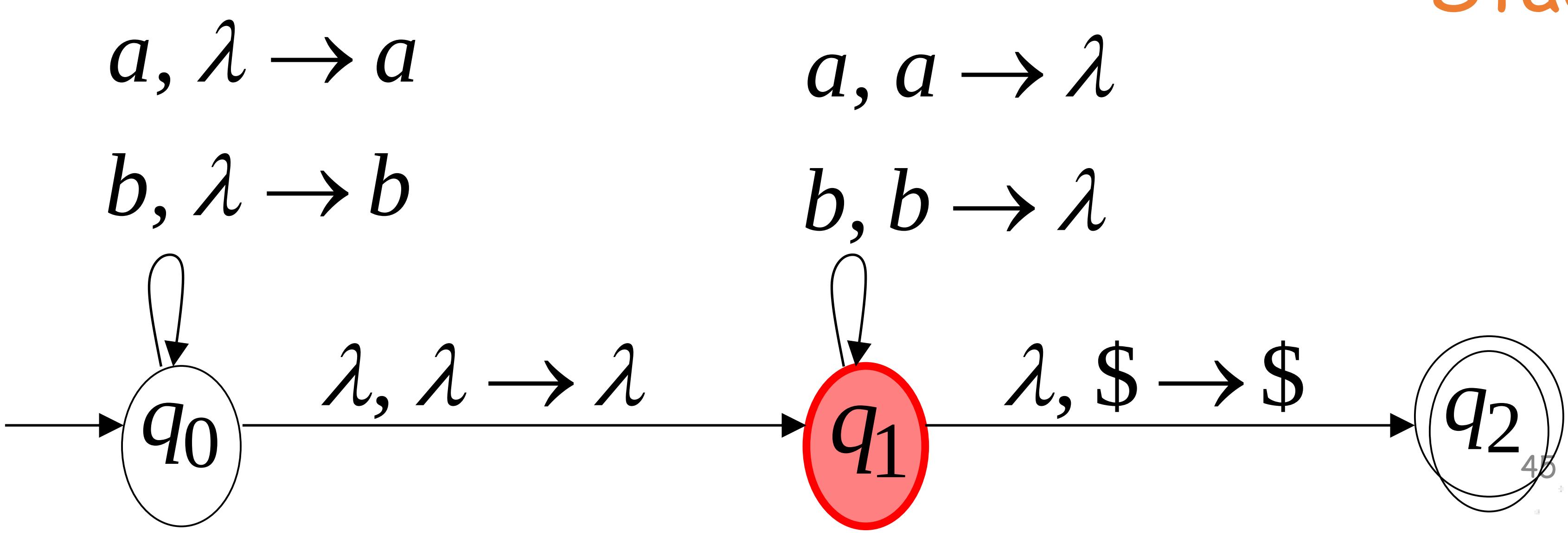
Input



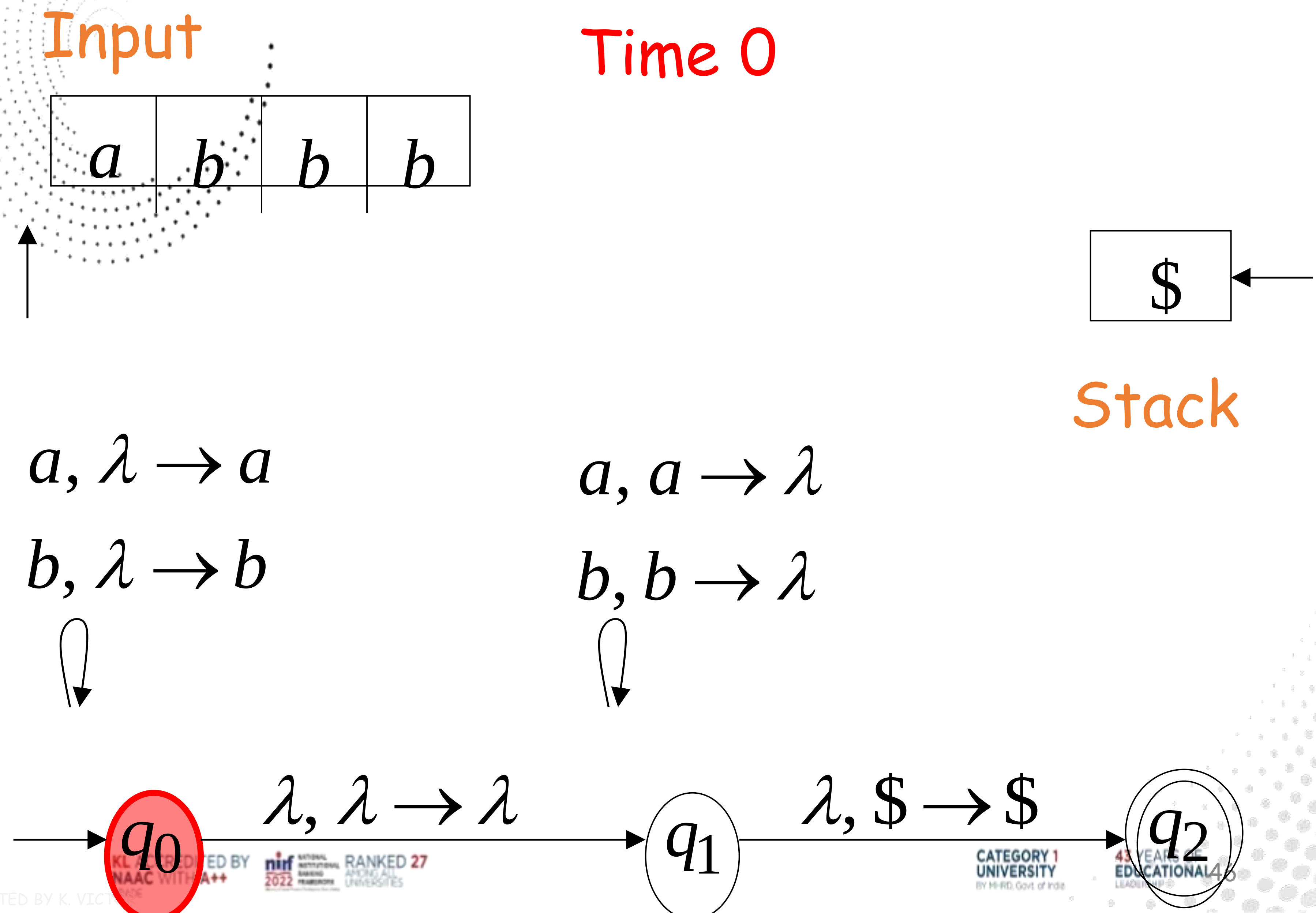
Input is not consumed



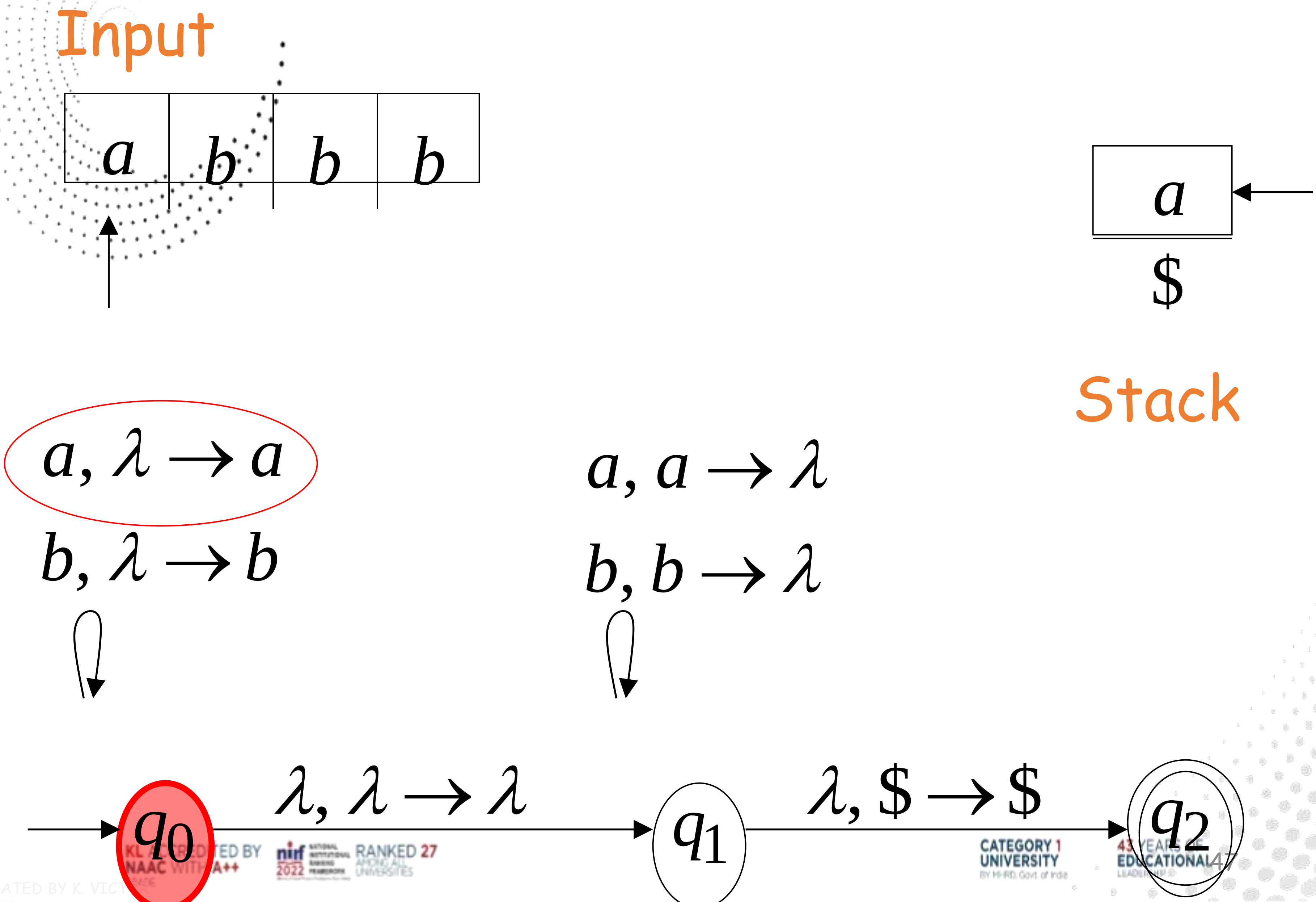
Stack



Another computation on same string:

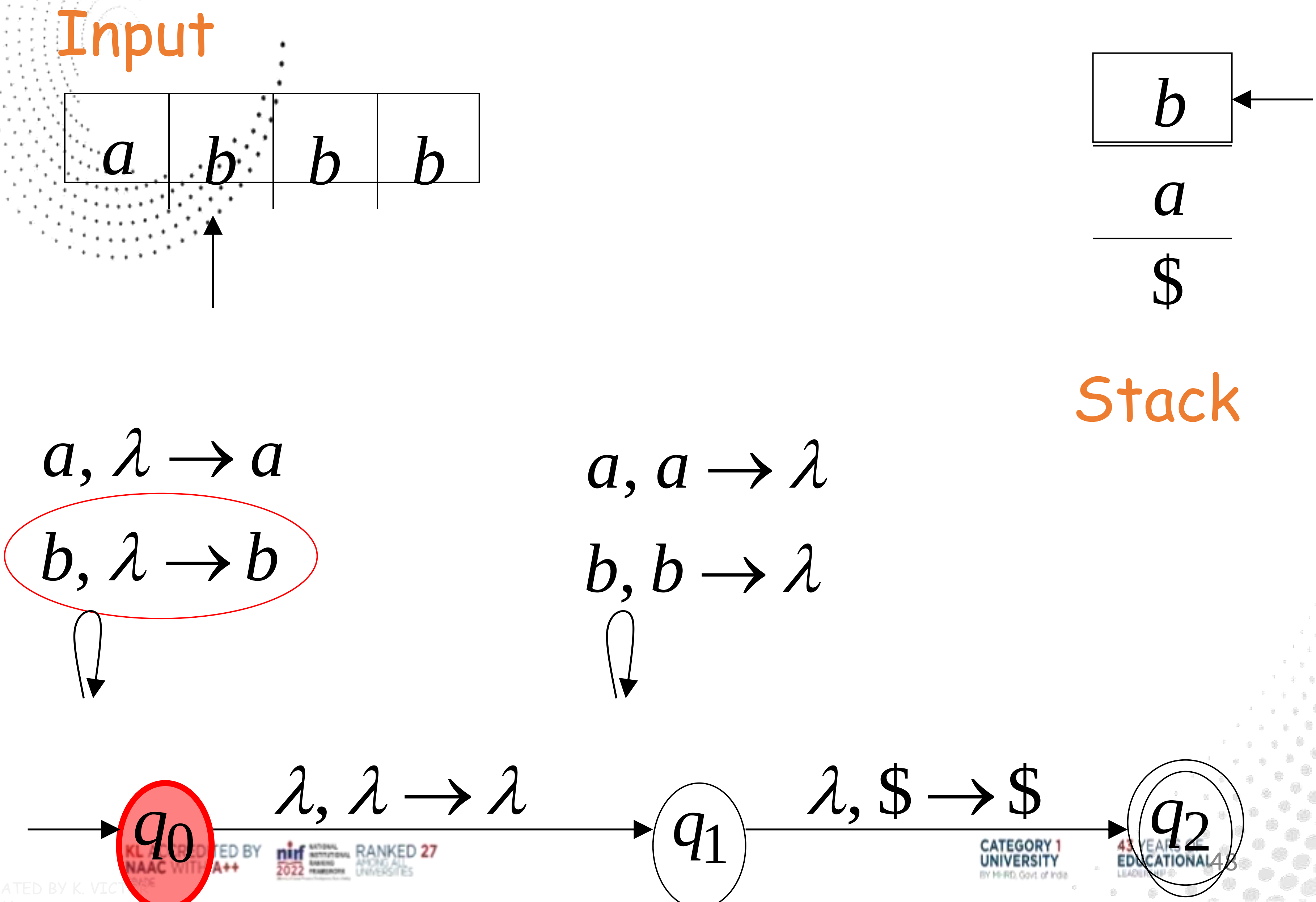


Time 1



Input
a
b
b
b
a
$
Stack
$a, \lambda \to a$
$b, \lambda \to b$
$a, a \to \lambda$
$b, b \to \lambda$
$\lambda, \lambda \to \lambda$
$\lambda, \$ \to \$$
q_0
q_1
q_2

Time 2



Input
a b b b
b
a
$
Stack
$a, \lambda \rightarrow a$
$b, \lambda \rightarrow b$
$a, a \rightarrow \lambda$
$b, b \rightarrow \lambda$
$\lambda, \lambda \rightarrow \lambda$
$\lambda, \$ \rightarrow \$$
q_0
q_1
q_2

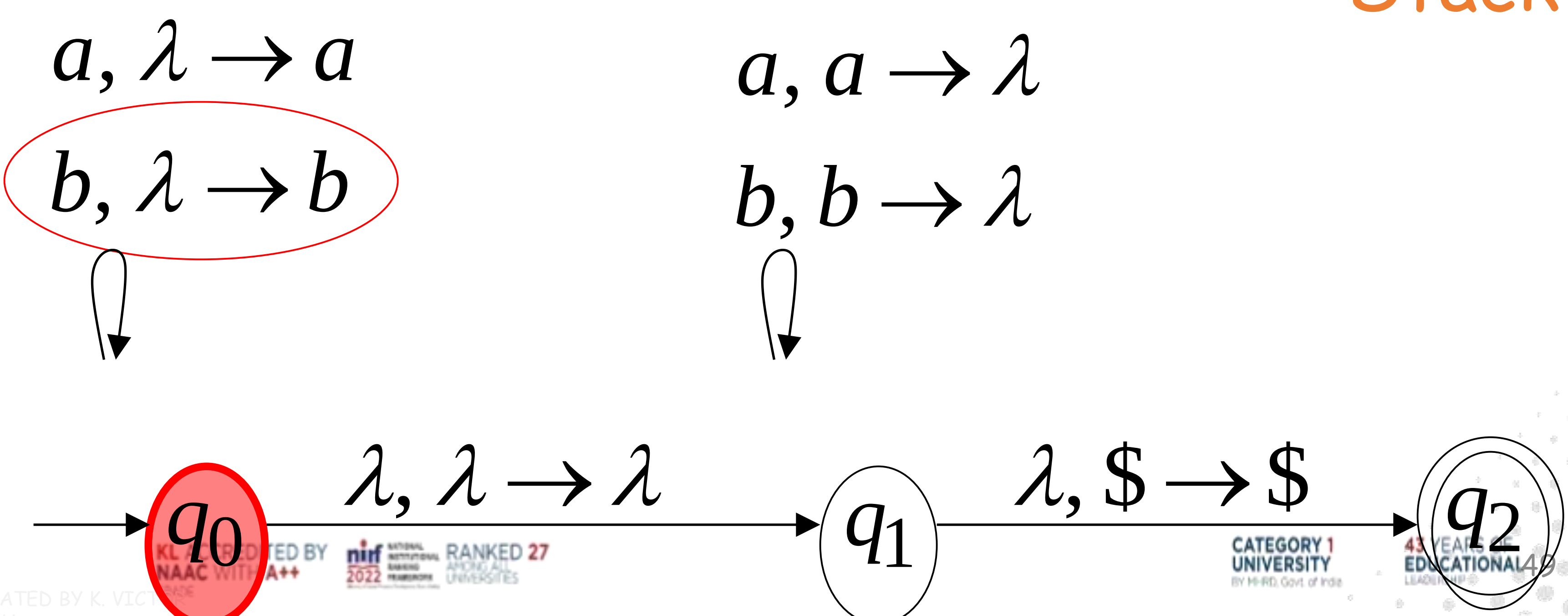
Time 3

Input

a	b	b	b

Stack

b
b
a
$\$$



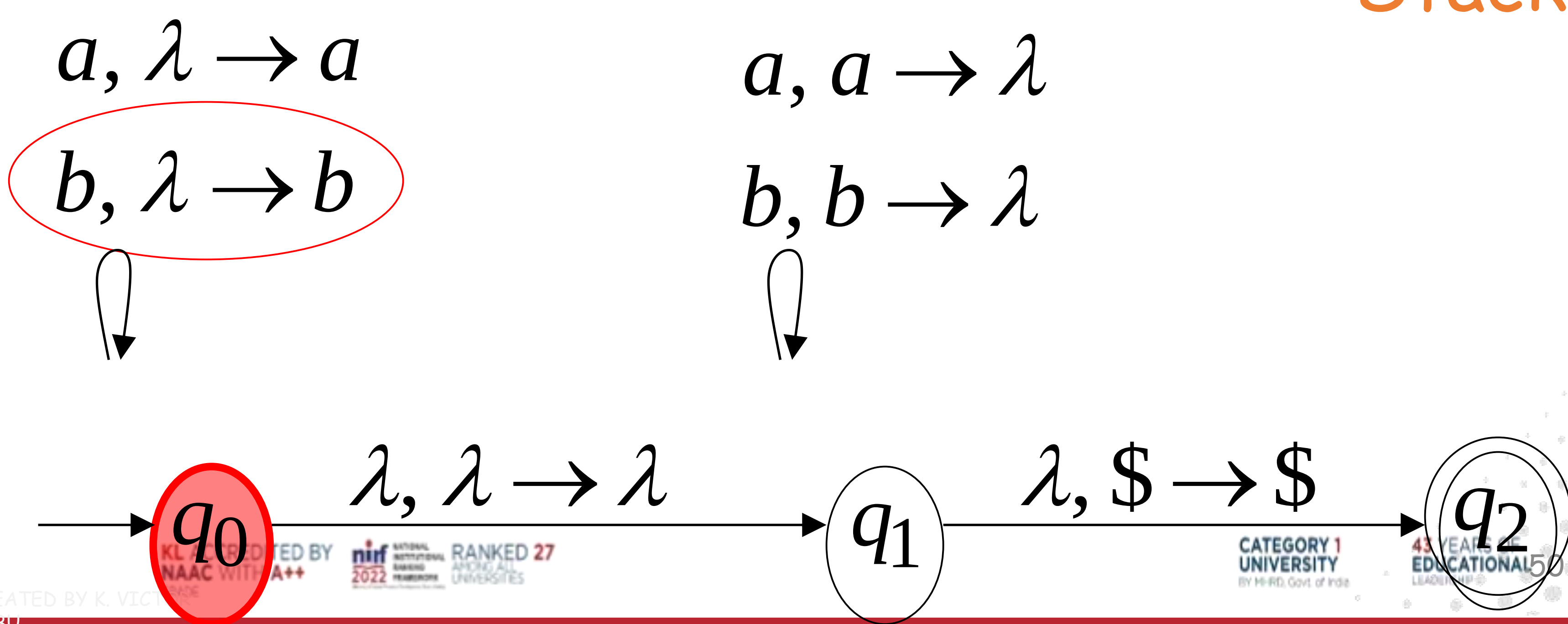
Time 4

Input

a	b	b	b

Stack

b
b
b
a
$\$$



Time 5

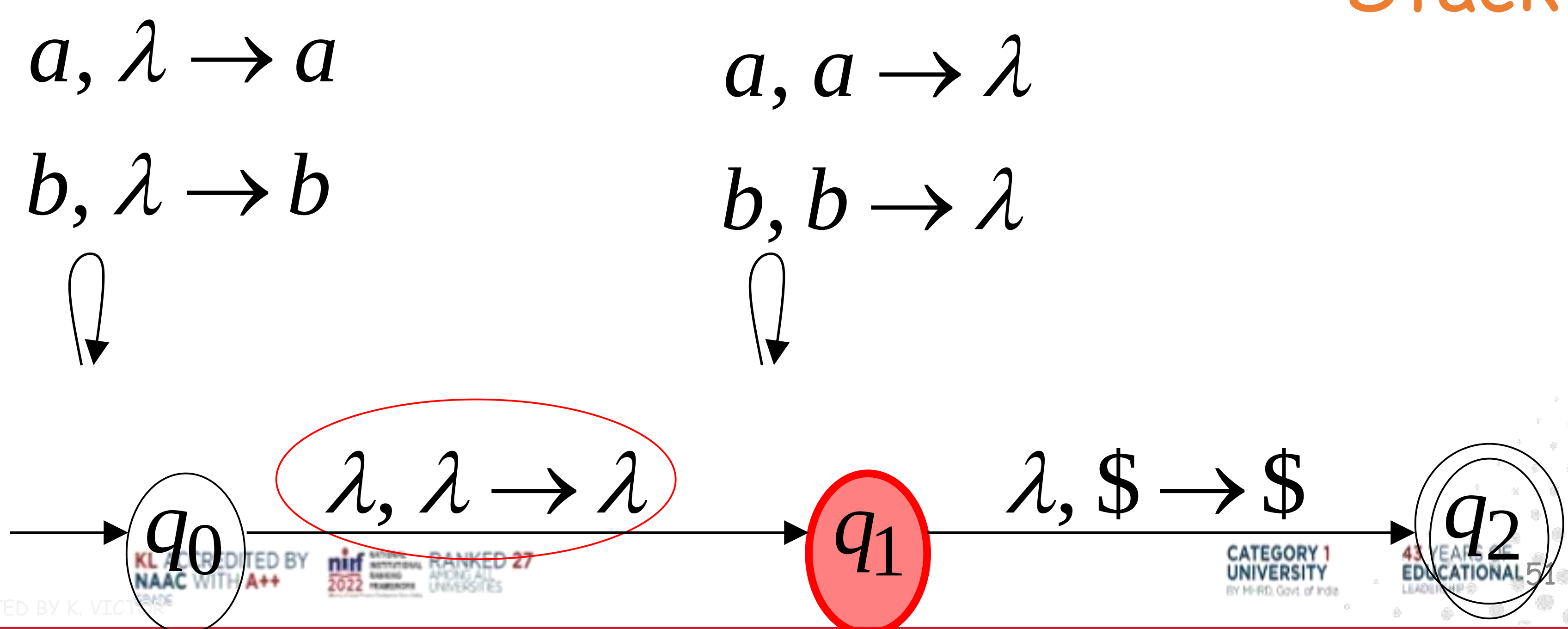
Input

a	b	b	b

No final state is reached

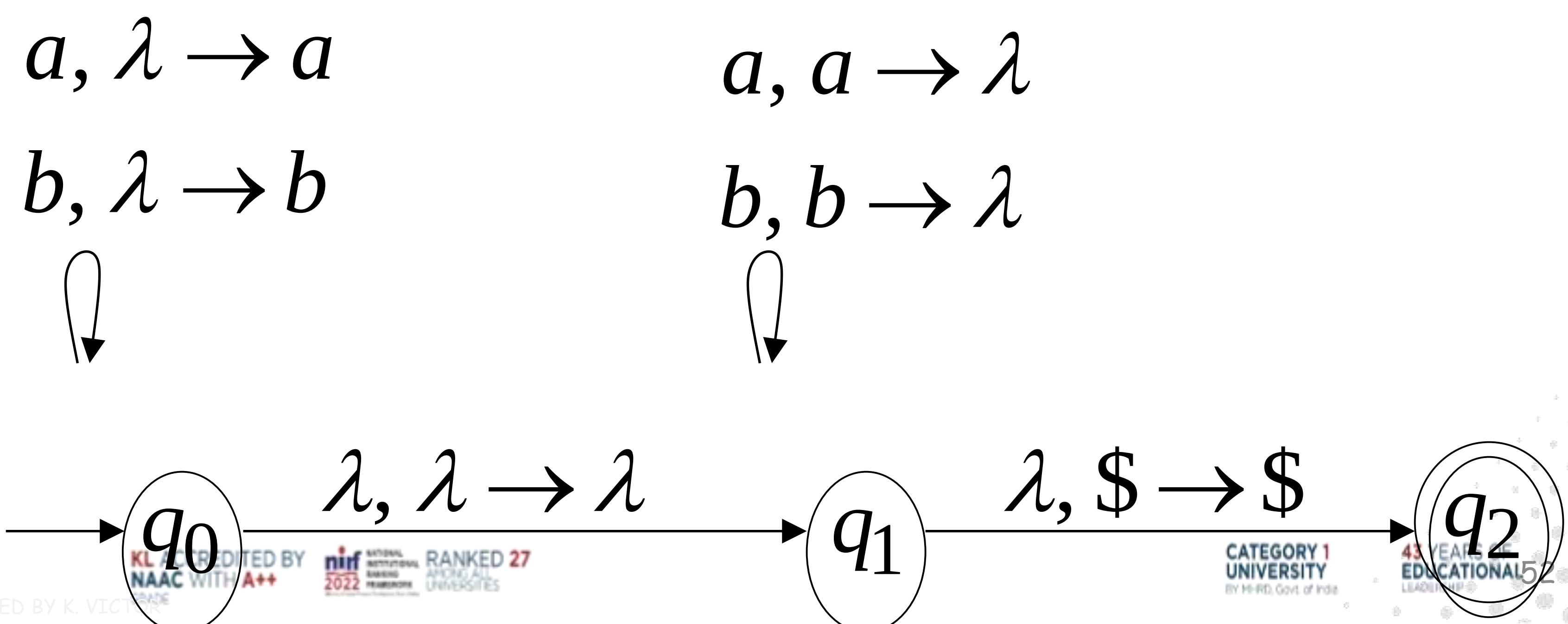
Stack

b
b
b
a
\$



There is no computation that accepts string $abbb$

$$abbb \notin L(M)$$



A string is rejected if there is no computation such that:

All the input is consumed

AND

The last state is a final state

At the end of the computation,
we do not care about the stack contents

In other words, a string is rejected if in every computation with this string:

The input cannot be consumed

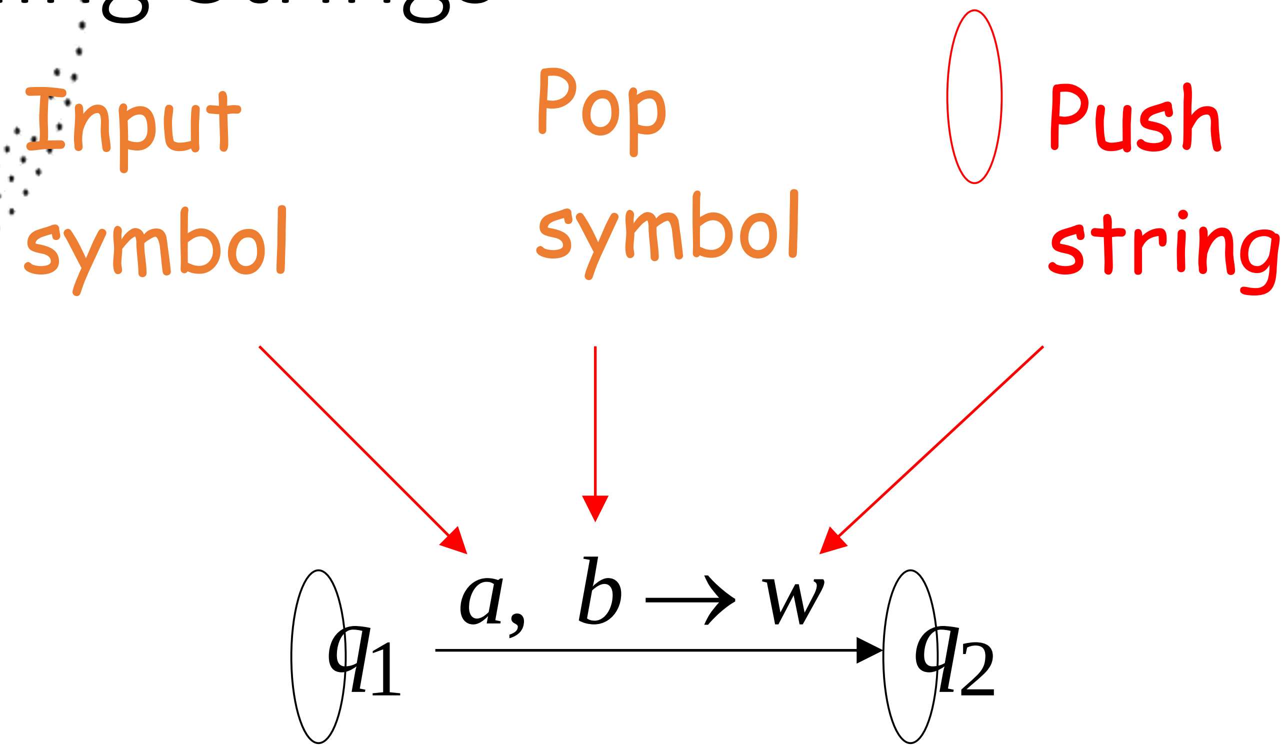
OR

The input is consumed and the last state is not a final state

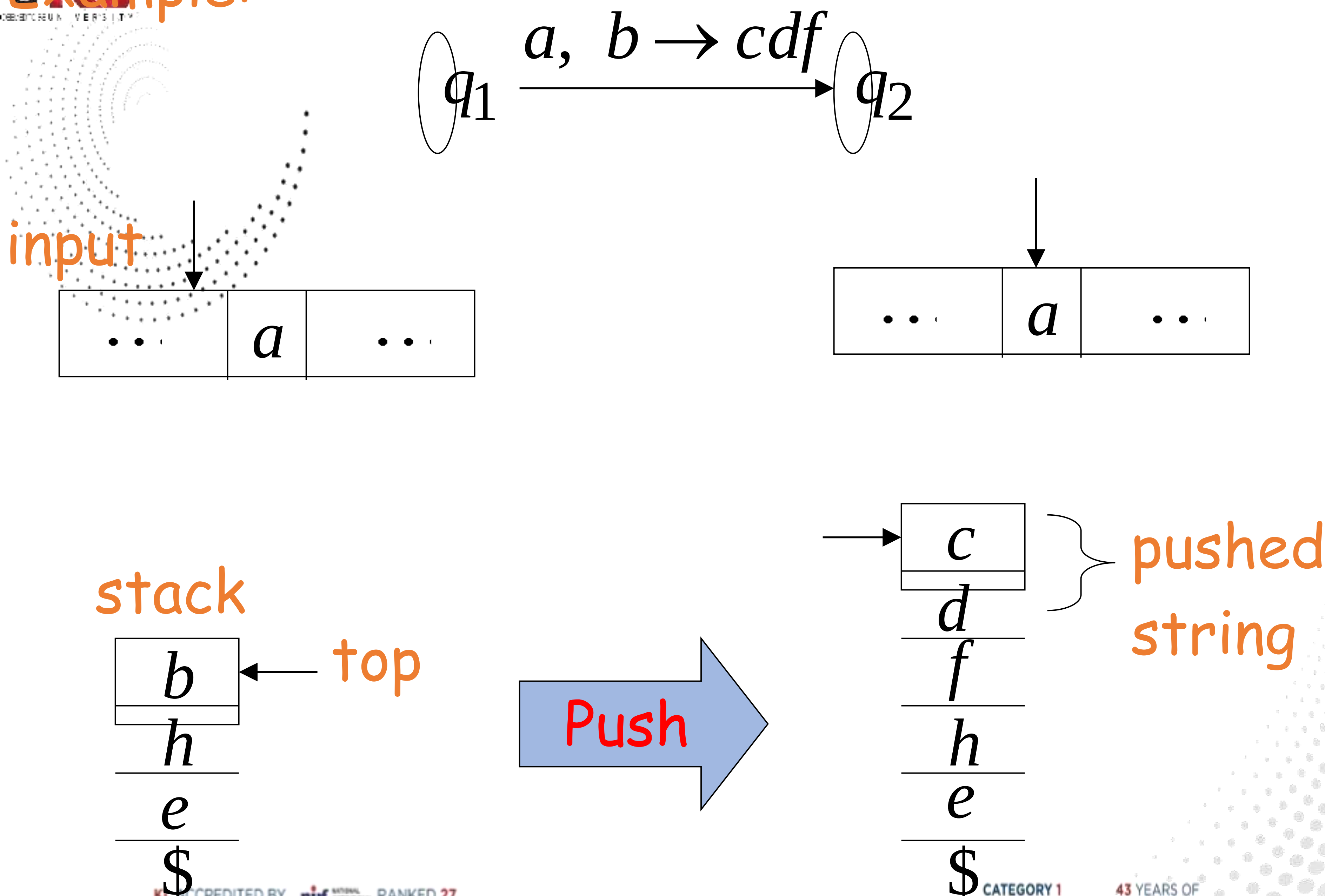
OR

The stack head moves below the bottom of the stack

Pushing Strings



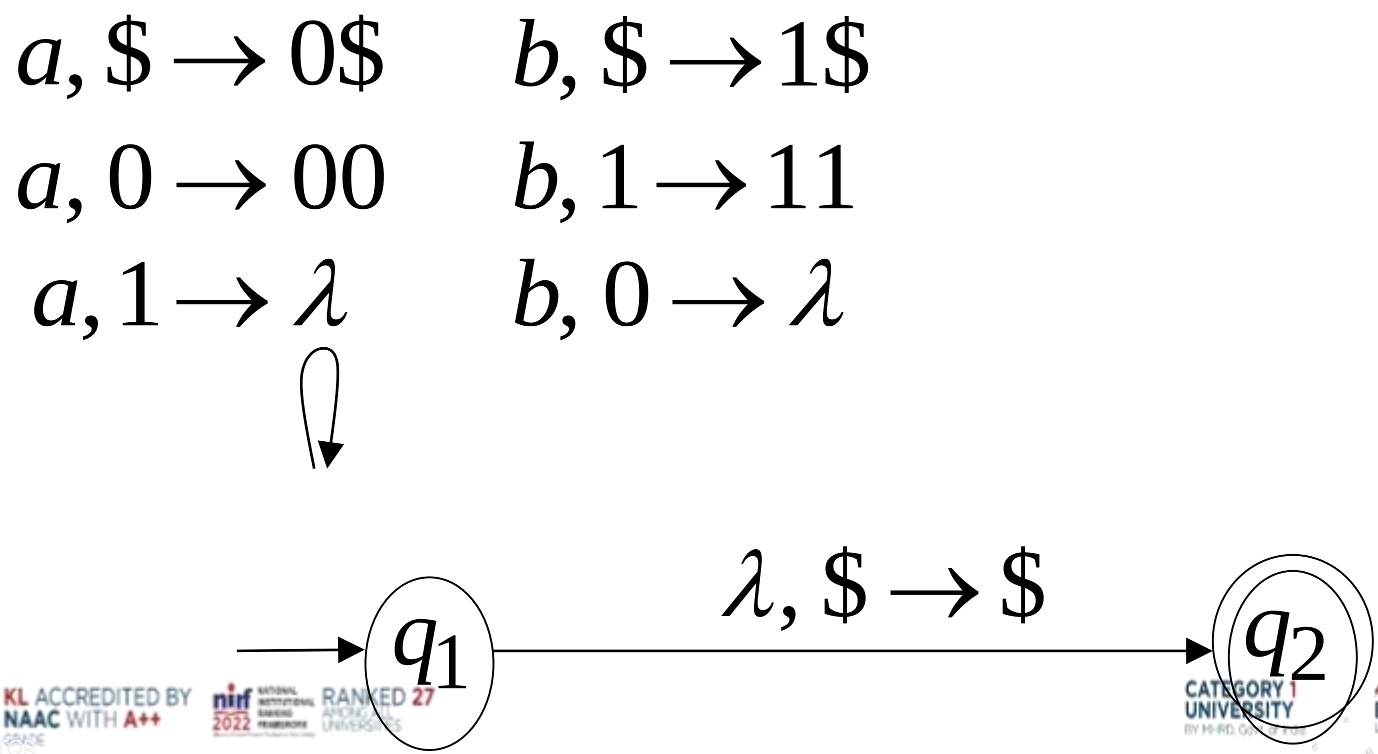
Example:


q_1
$a, \; b \to cdf$
q_2
input
a
a
stack
top
b
h
e
$\$$
Push
c
d
f
h
e
$\$$
pushed
string

Another NPDA example

NPDA M

$$L(M) = \{w: \; n_a = n_b\}$$



Execution Example: Time 0

Input

a	b	b	a	a	b

$

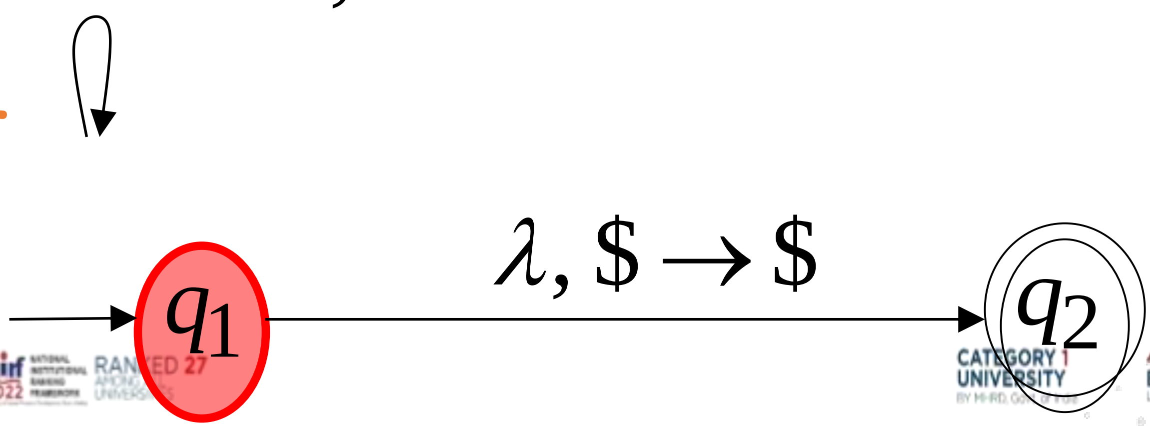
Stack

$$a, \$ \rightarrow 0\$ \qquad b, \$ \rightarrow 1\$$$

$$a, 0 \rightarrow 00 \qquad b, 1 \rightarrow 11$$

$$a, 1 \rightarrow \lambda \qquad b, 0 \rightarrow \lambda$$

current state



Time 1

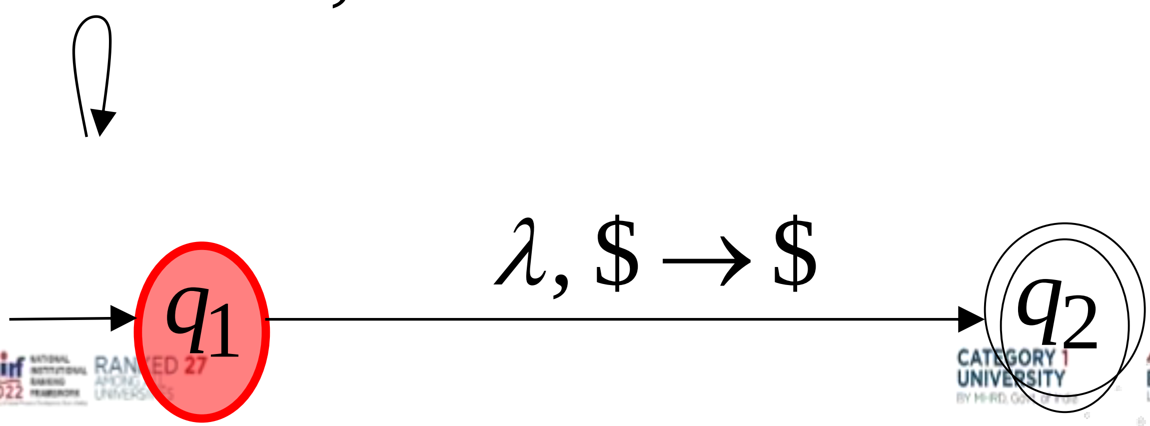
Input

a	b	b	a	a	b

$$a, \$ \rightarrow 0\$ \qquad b, \$ \rightarrow 1\$$$

$$a, 0 \rightarrow 00 \qquad b, 1 \rightarrow 11$$

$$a, 1 \rightarrow \lambda \qquad b, 0 \rightarrow \lambda$$



0
\$

Stack

Time 3

Input

a	b	b	b	a	a

Stack

$\cancel{0}$
$\$$

$$a, \$ \rightarrow 0\$ \qquad b, \$ \rightarrow 1\$$$
$$a, 0 \rightarrow 00 \qquad b, 1 \rightarrow 11$$
$$a, 1 \rightarrow \lambda \qquad b, 0 \rightarrow \lambda$$

$$\rightarrow q_1 \xrightarrow{\lambda, \$ \rightarrow \$} q_2$$

Time 4

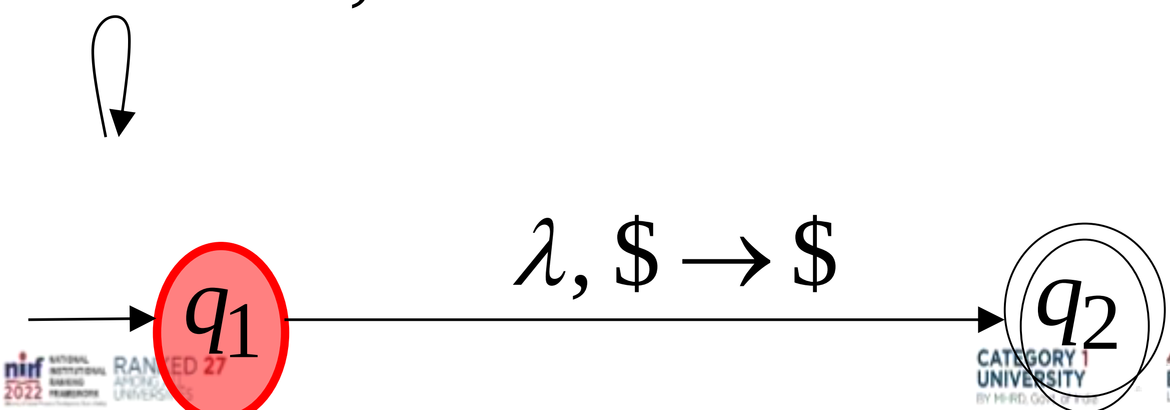
Input

a	b	b	b	a	a

$$a, \$ \rightarrow 0\$ \qquad b, \$ \rightarrow 1\$$$

$$a, 0 \rightarrow 00 \qquad b, 1 \rightarrow 11$$

$$a, 1 \rightarrow \lambda \qquad b, 0 \rightarrow \lambda$$



Stack

1
$

Time 5

Input

a	b	b	b	a	a

1

1

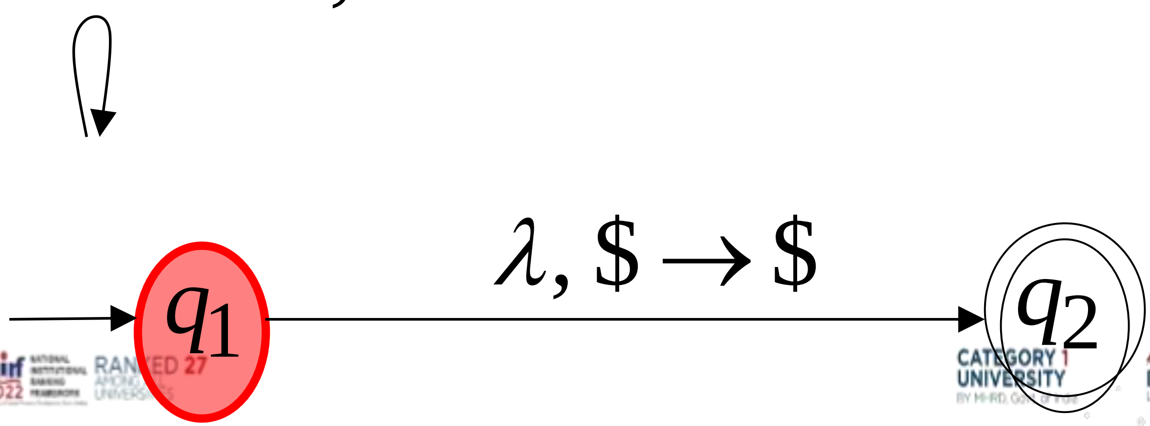
$\$$

Stack

$$a, \$ \rightarrow 0\$ \qquad b, \$ \rightarrow 1\$$$

$$a, 0 \rightarrow 00 \qquad b, 1 \rightarrow 11$$

$$a, 1 \rightarrow \lambda \qquad b, 0 \rightarrow \lambda$$



Time 6

Input

a	b	b	b	a	a

$\cancel{1}$

1

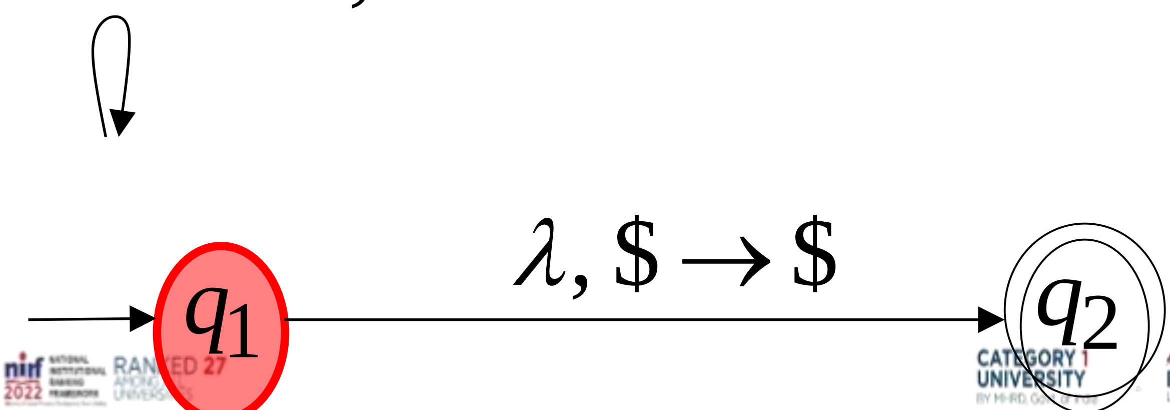
$\$$

Stack

$$a, \$ \rightarrow 0\$ \qquad b, \$ \rightarrow 1\$$$

$$a, 0 \rightarrow 00 \qquad b, 1 \rightarrow 11$$

$$a, 1 \rightarrow \lambda \qquad b, 0 \rightarrow \lambda$$



Time 7

Input

a	b	b	b	a	a

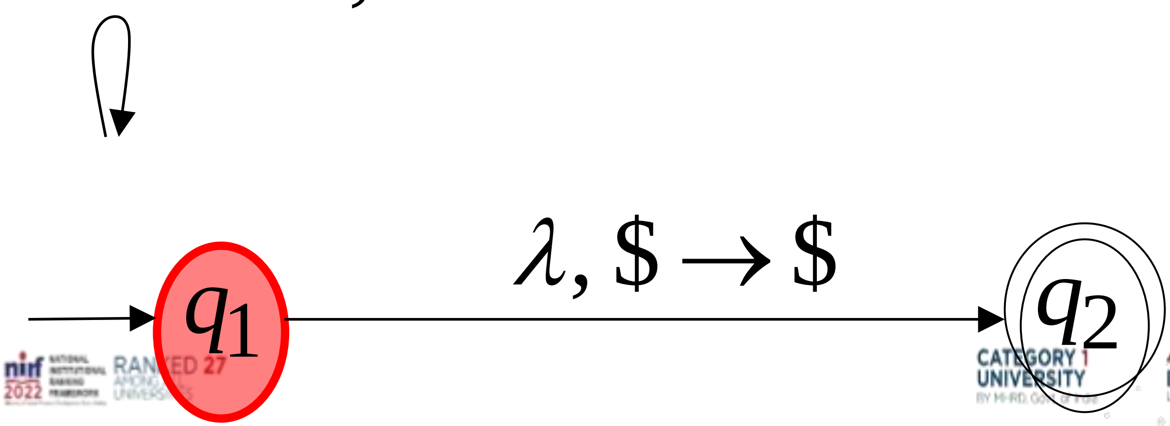
$$a, \$ \rightarrow 0\$ \qquad b, \$ \rightarrow 1\$$$

$$a, 0 \rightarrow 00 \qquad b, 1 \rightarrow 11$$

$$a, 1 \rightarrow \lambda \qquad b, 0 \rightarrow \lambda$$

~~1~~
$\$$

Stack



Time 8

Input

a	b	b	b	a	a

\$

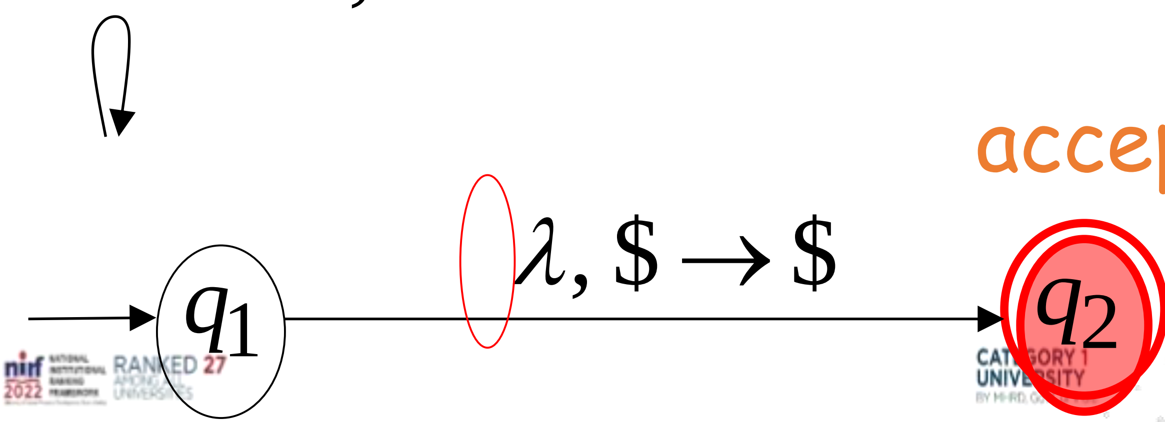
Stack

$a, \$ \rightarrow 0\$$ $\qquad b, \$ \rightarrow 1\$$

$a, 0 \rightarrow 00$ $\qquad b, 1 \rightarrow 11$

$a, 1 \rightarrow \lambda$ $\qquad b, 0 \rightarrow \lambda$

accept

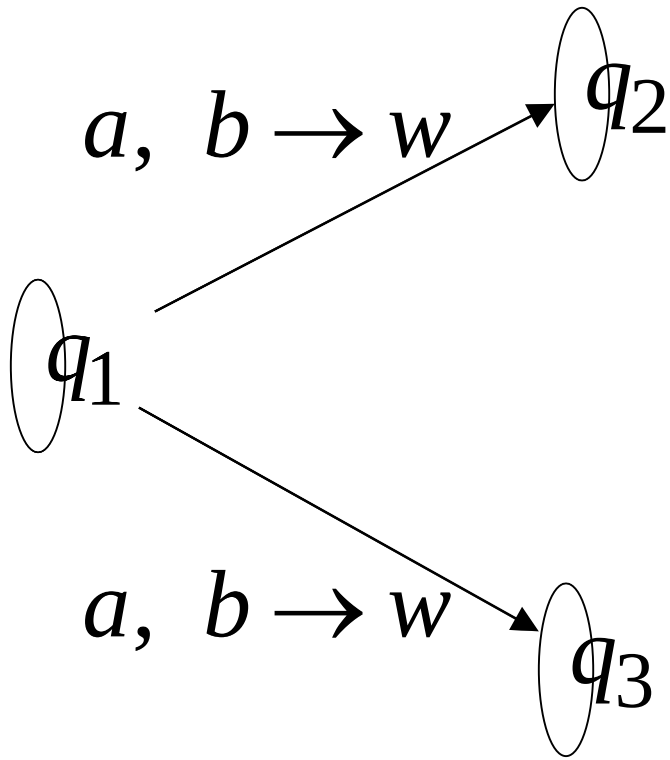


Formalities for NPDAs

$$q_1 \xrightarrow{a,\ b \to w} q_2$$

Transition function:

$$\delta(q_1, a, b) = \{(q_2, w)\}$$



Transition function:

$$\delta(q_1, a, b) = \{(q_2, w), \ (q_3, w)\}$$

Formal Definition

Non-Deterministic Pushdown Automaton NPDA

$$M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$$

States

Input alphabet

Stack alphabet

Transition function

Initial state

Stack start symbol

Final states

Instantaneous Description

$$(q,\ u,\ s)$$

Current state

Remaining input

Current stack contents

Example: Instantaneous Description

$$(q_1, bbb, aaa\$)$$

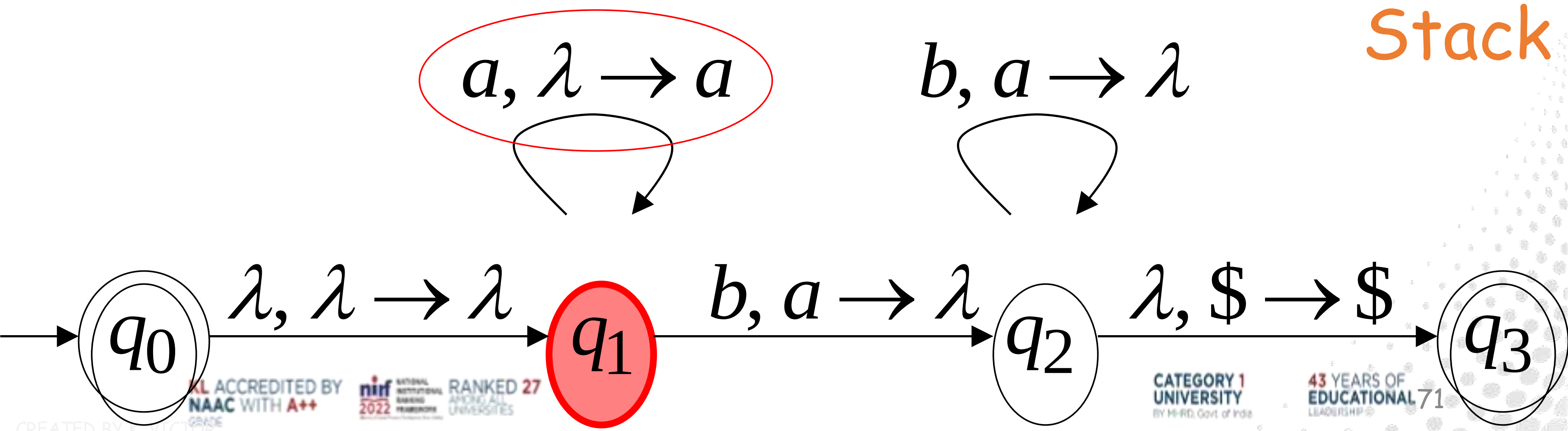
Time 4:

Input

a	a	a	b	b	b

Stack

a
a
a
$\$$



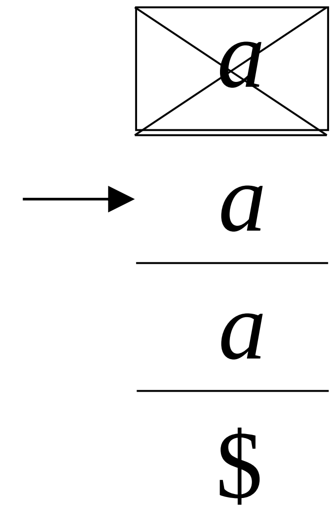
Example: Instantaneous Description

$$(q_2, bb, aa\$)$$

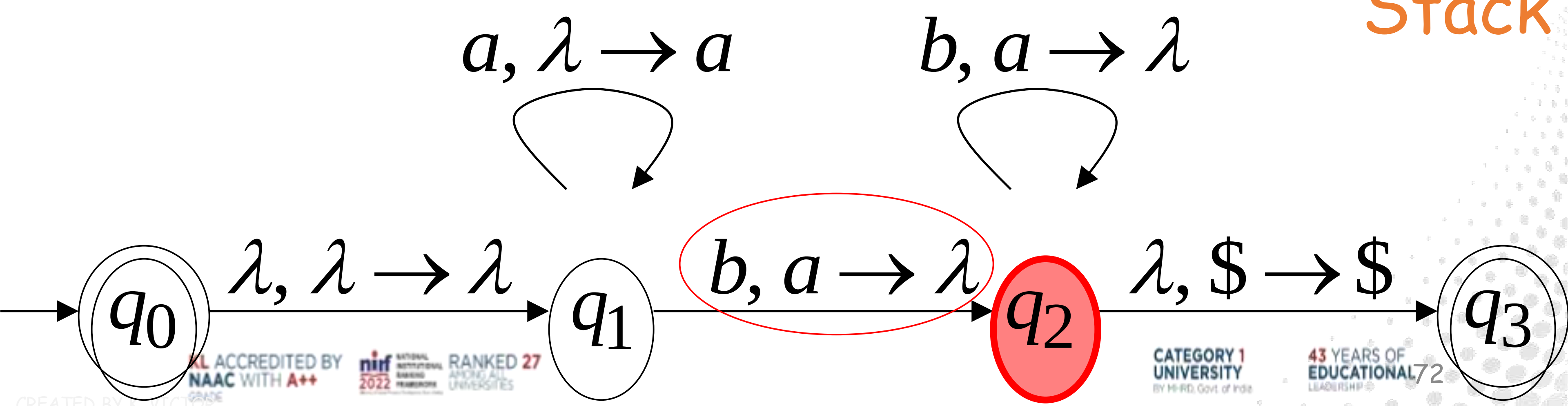
Time 5:

Input

a	a	a	b	b	b



Stack



We write:

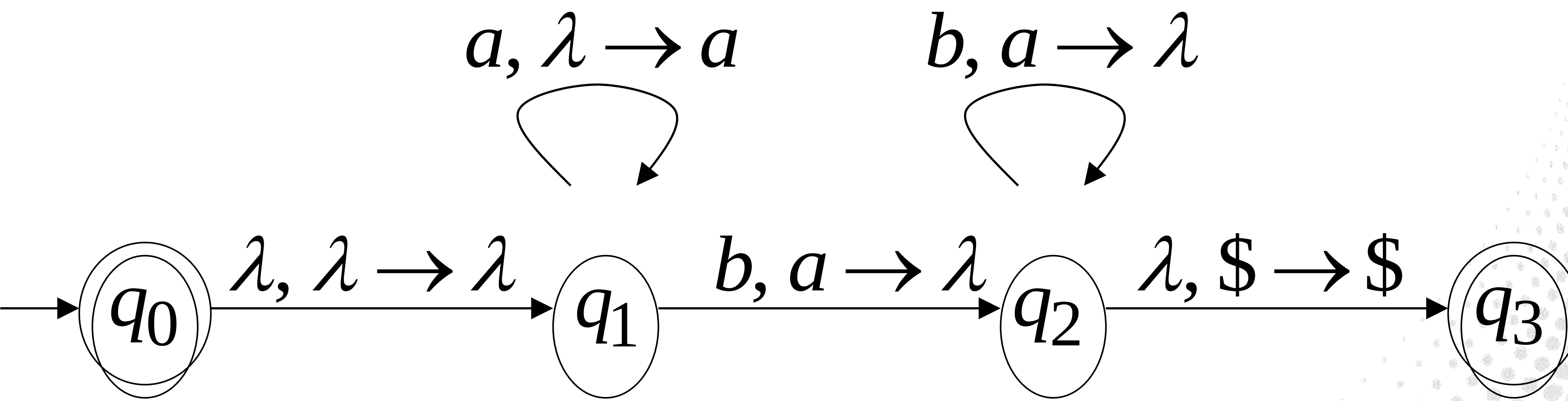
$$(q_1, bbb, aaa\$) \succ (q_2, bb, aa\$)$$

Time 4

Time 5

A computation:

$$(q_0, aaabbb, \$) \succ (q_1, aaabbb, \$) \succ$$
$$(q_1, aabbb, a\$) \succ (q_1, abbb, aa\$) \succ (q_1, bbb, aaa\$) \succ$$
$$(q_2, bb, aa\$) \succ (q_2, b, a\$) \succ (q_2, \lambda, \$) \succ (q_3, \lambda, \$)$$



$$(q_0, aaabbb, \$) \succ (q_1, aaabbb, \$) \succ$$
$$(q_1, aabbb, a\$) \succ (q_1, abbb, aa\$) \succ (q_1, bbb, aaa\$) \succ$$
$$(q_2, bb, aa\$) \succ (q_2, b, a\$) \succ (q_2, \lambda, \$) \succ (q_3, \lambda, \$)$$

For convenience we write:

$$(q_0, aaabbb, \$) \overset{*}{\succ} (q_3, \lambda, \$)$$

Formal Definition

Language $L(M)$ of NPDA M :

$$L(M) = \{w : (q_0, w, s) \overset{*}{\succ} (q_f, \lambda, s')\}$$

Initial state

Final state

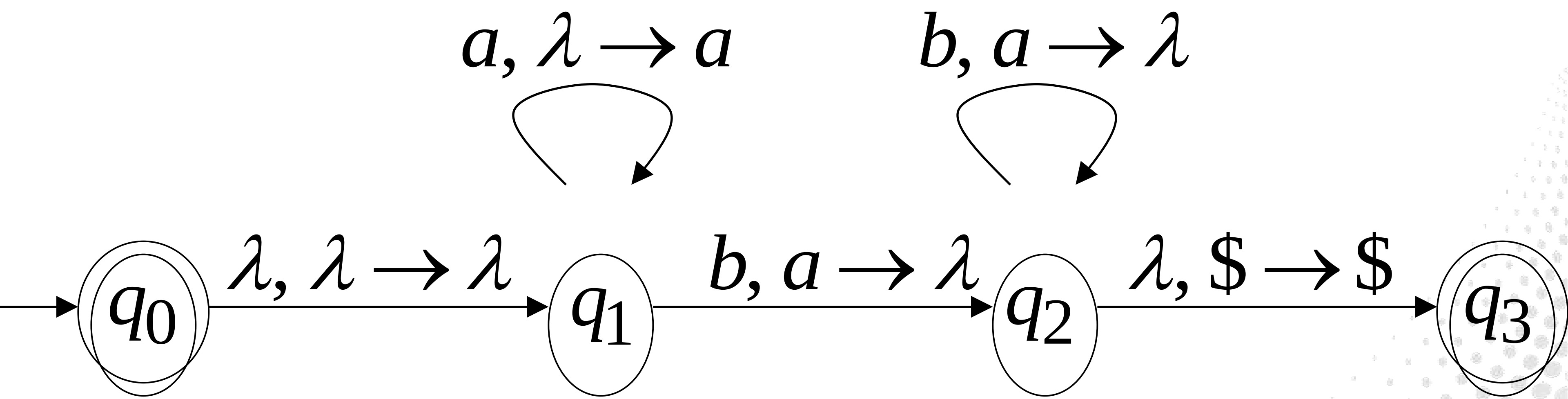
Example:

$$(q_0, aaabbb, \$) \overset{*}{\succ} (q_3, \lambda, \$)$$

$$\Downarrow$$

$$aaabbb \in L(M)$$

NPDA M :

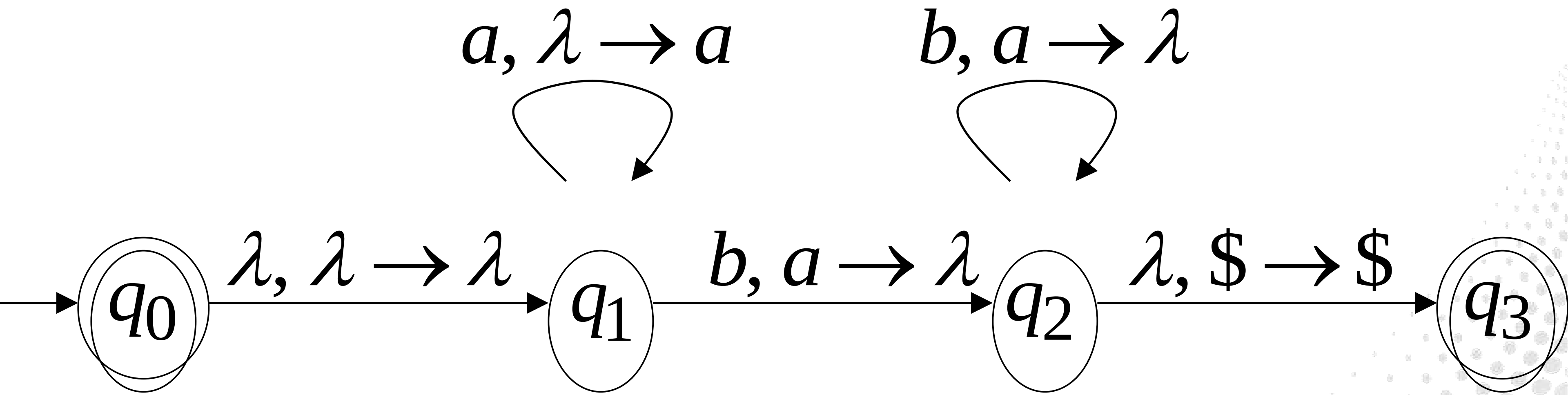


$$(q_0, a^n b^n, \$) \overset{*}{\succ} (q_3, \lambda, \$)$$

$$\Downarrow$$

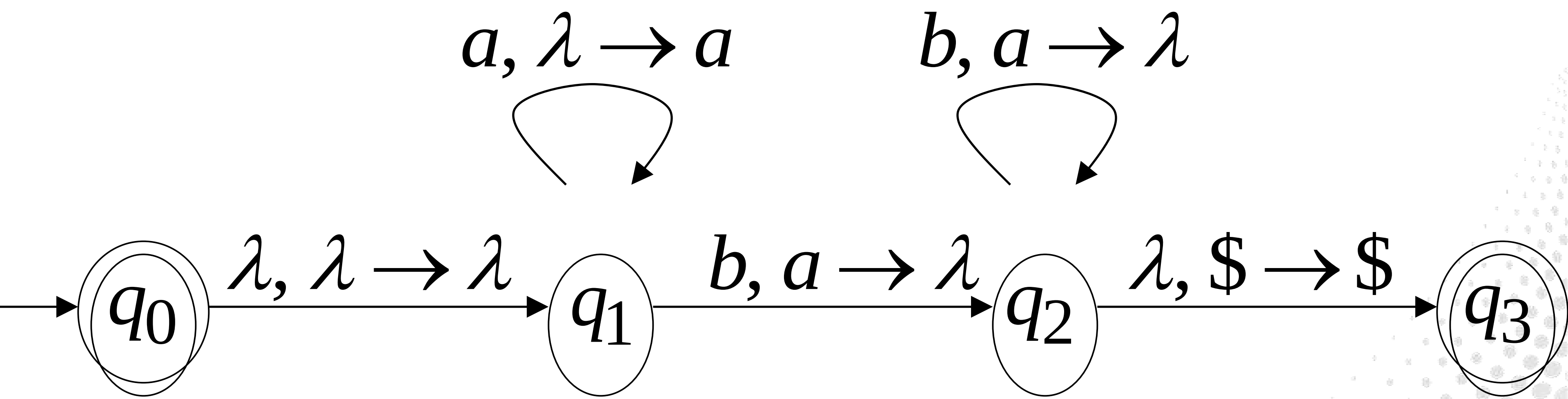
$$a^n b^n \in L(M)$$

NPDA M :



Therefore: $L(M) = \{a^n b^n : n \geq 0\}$

NPDA M :



Problems

4:

Design a NPDA to accept the following Language:

$$L(M) = \{ a^n b^{2n} , n \geq 0 \}$$

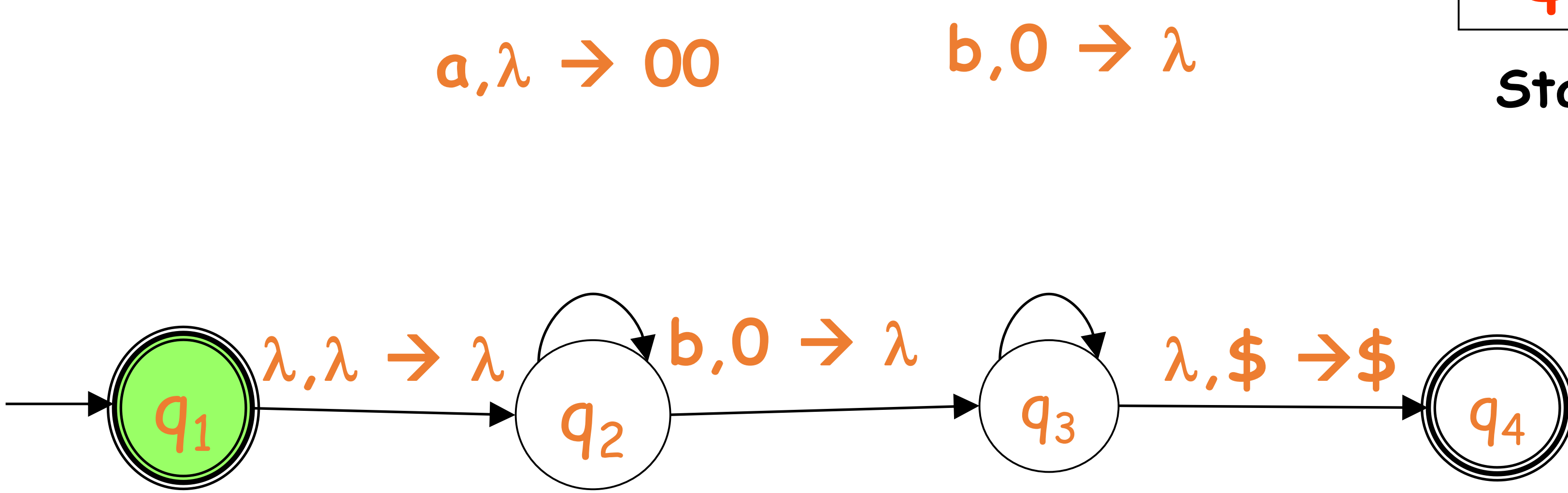
Time - 0

Input String

a	a	b	b	b	b

$

Stack



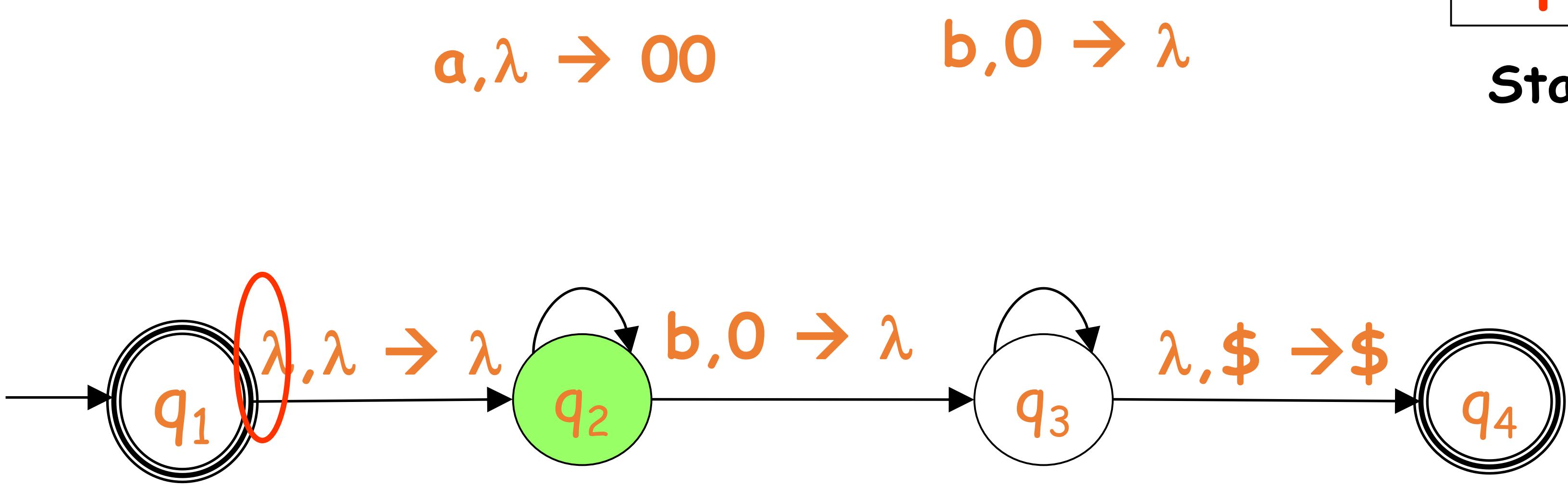
Time - 1

Input String

a	a	b	b	b	b

$

Stack



Time - 2

Input String

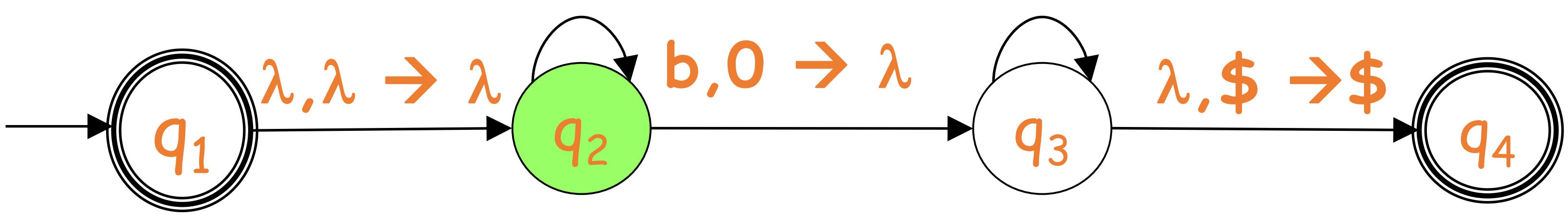
a	a	b	b	b	b

Stack
0
0
$

Stack

$a, \lambda \rightarrow 00$

$b, 0 \rightarrow \lambda$



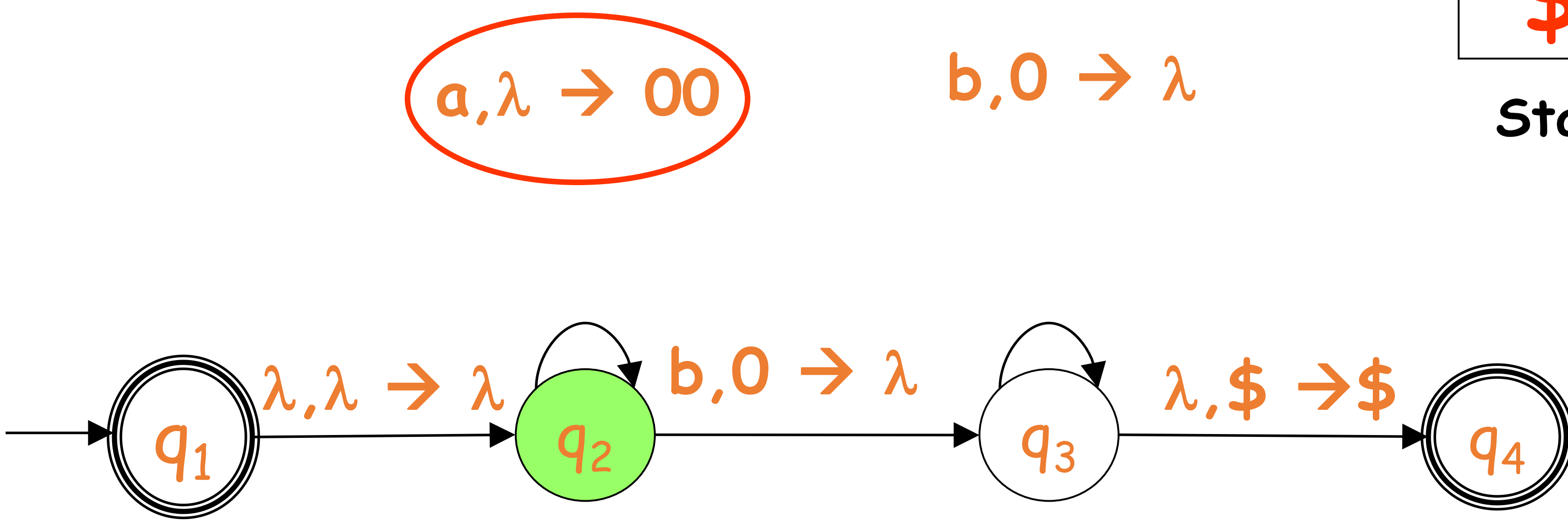
Time - 3

Input String

a	a	b	b	b	b

Stack

0
0
0
0
$



Time - 4

Input String

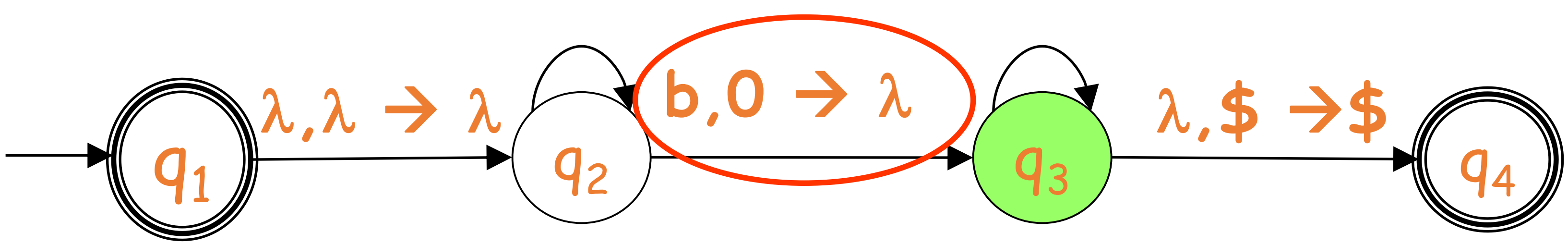
a	a	b	b	b	b

Stack

~~0~~
0
0
0
$

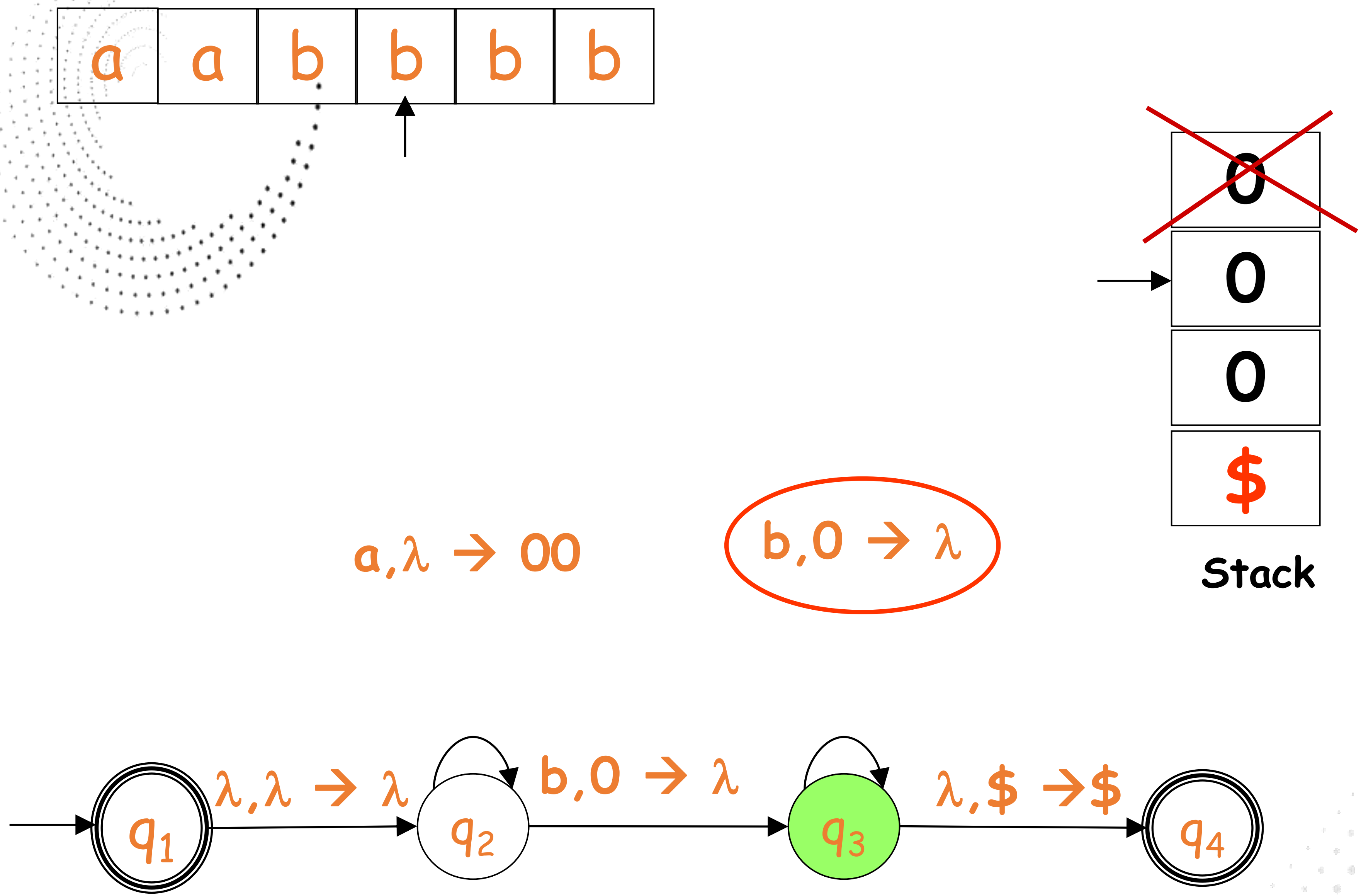
a,λ → 00

b,0 → λ



Time - 5

Input String



Time - 6

Input String

a	a	b	b	b	b

~~0~~

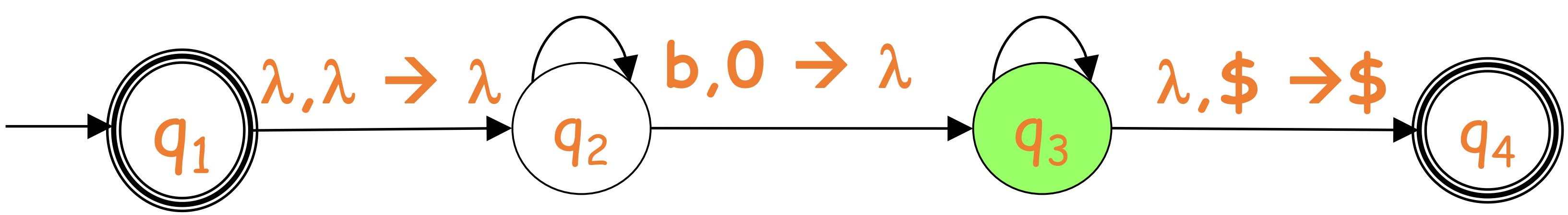
0

$

Stack

a,λ → 00

b,0 → λ



Time - 7

Input String

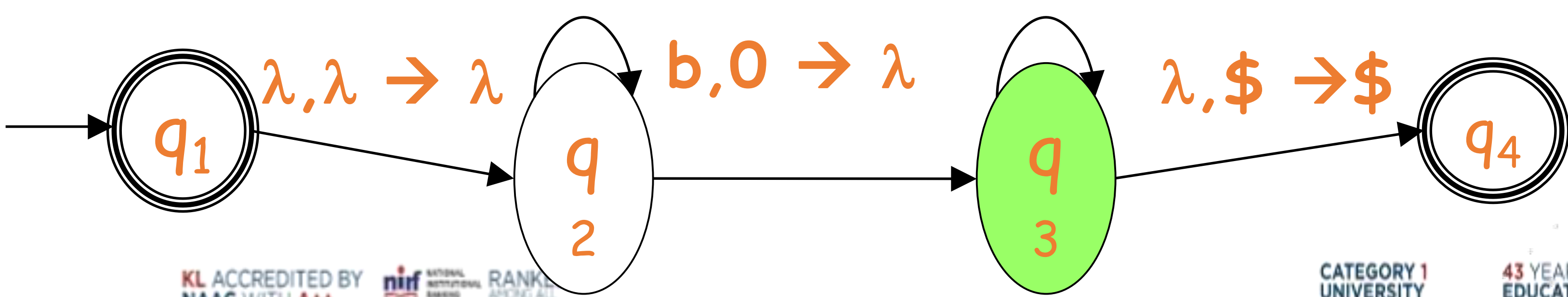
a	a	b	b	b	b

$a,\lambda \rightarrow 00$

$b,0 \rightarrow \lambda$

$

Stack



Time - 8

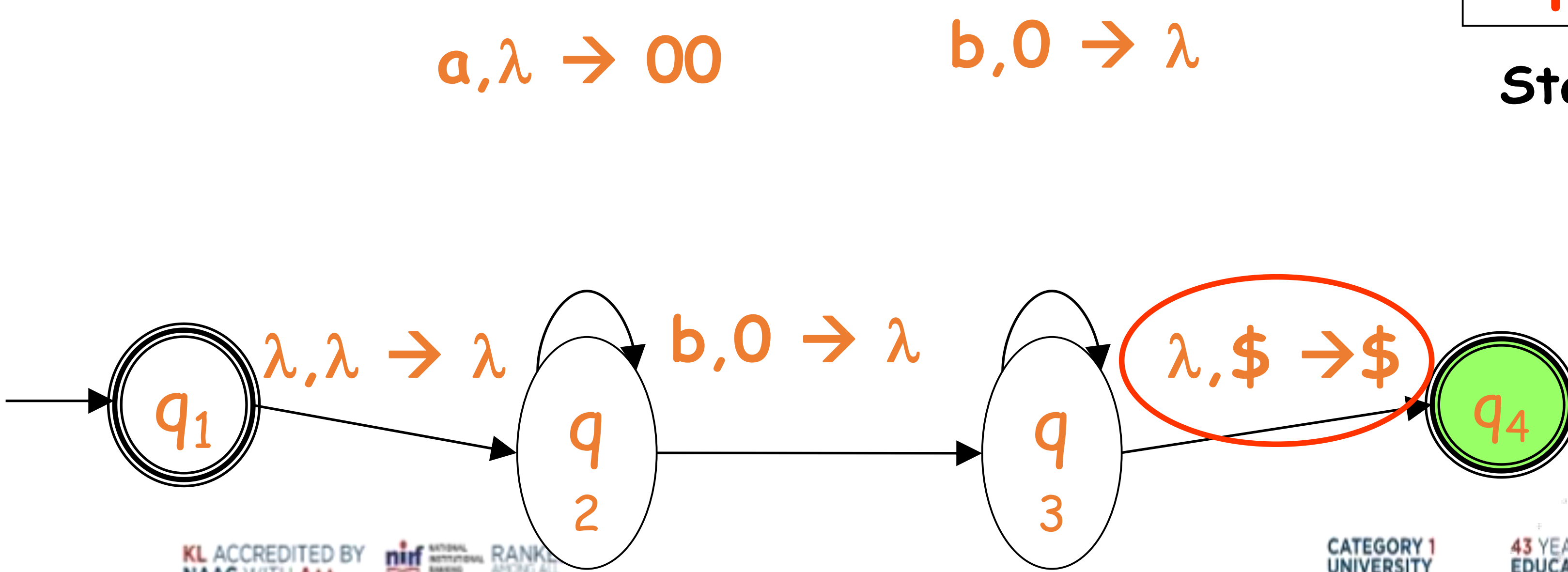
Input String

a	a	b	b	b	b

"Accept"

$

Stack



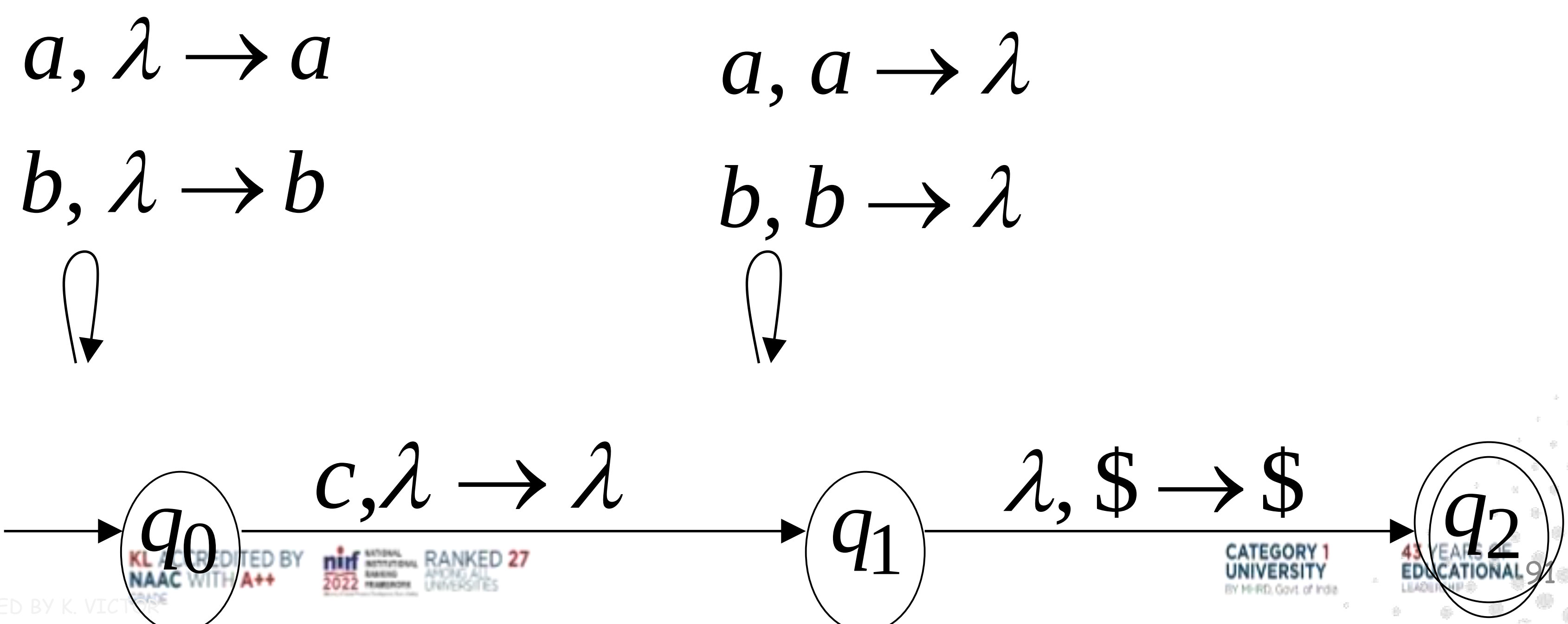
Problems

5:

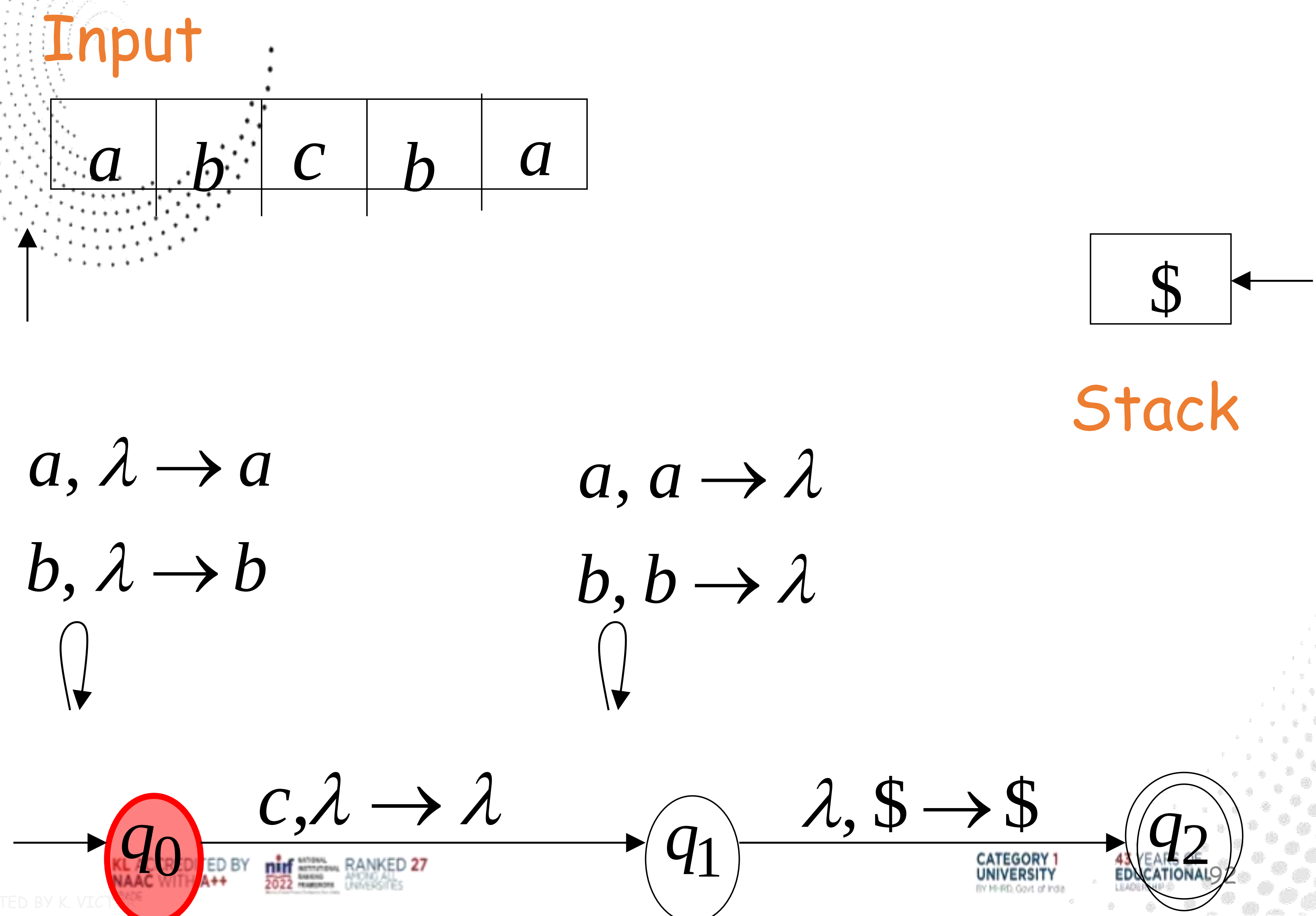
Design a NPDA to accept the following Language:

$$L(M) = \{ wcw^R, w \in \{a,b\}^* \}$$

$L(M) = \{ wcw^R, w \in \{a,b\}^* \}$



Execution Example: Time 0



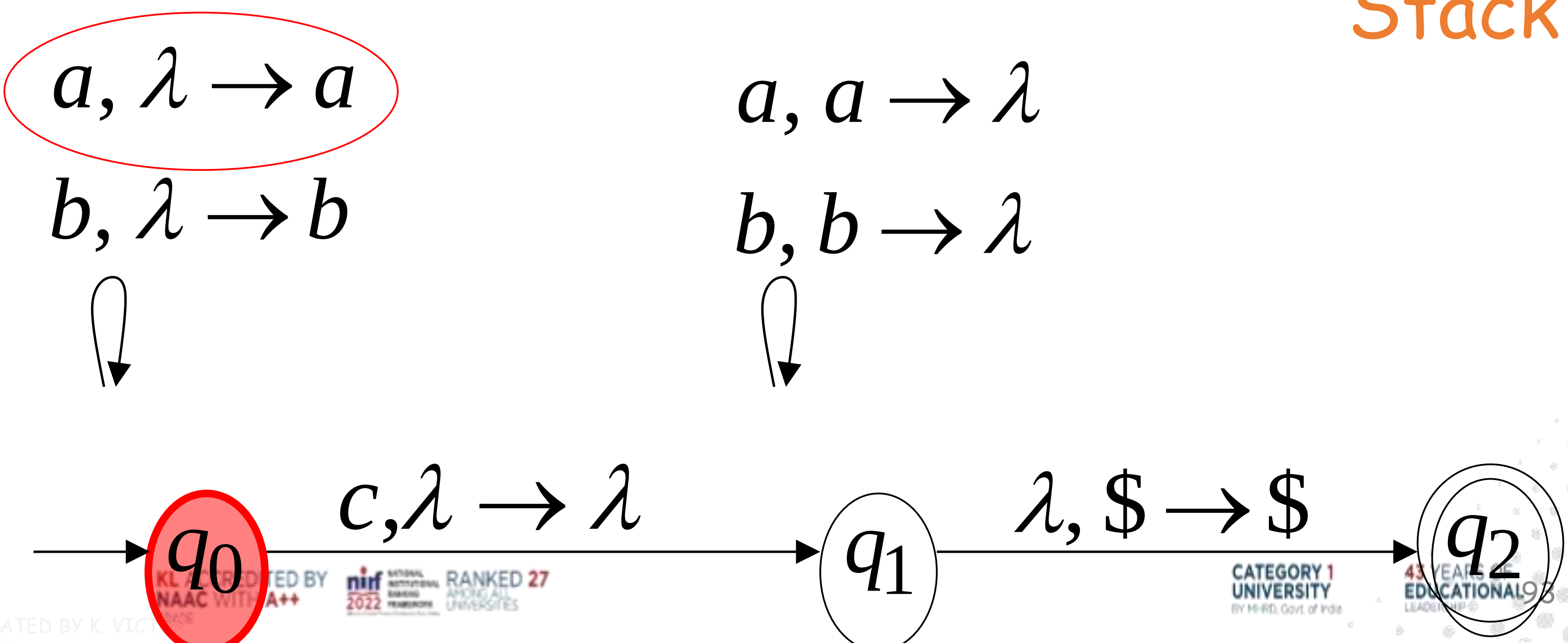
Time 1

Input

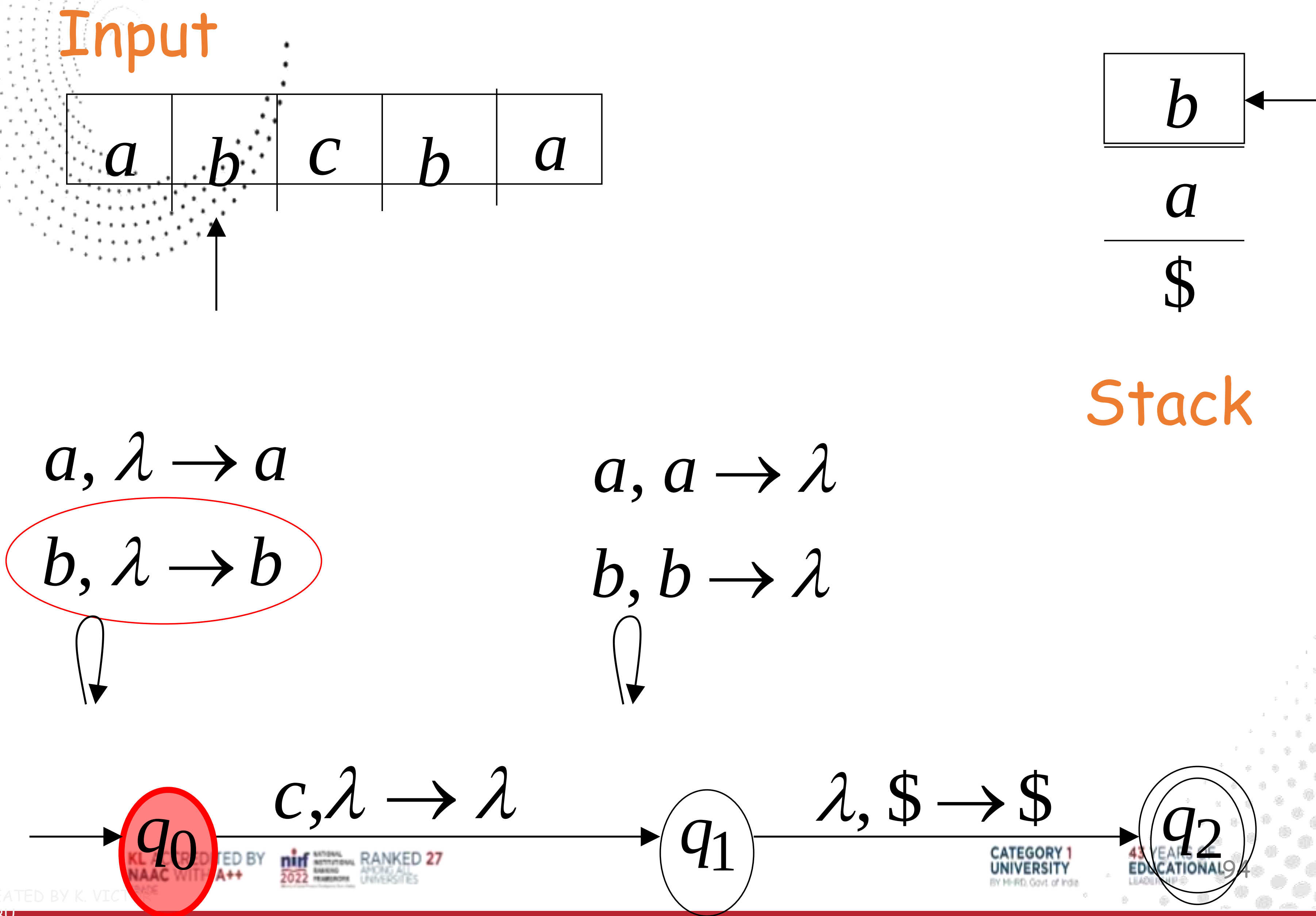
a	b	c	b	a

a
$\$$

Stack

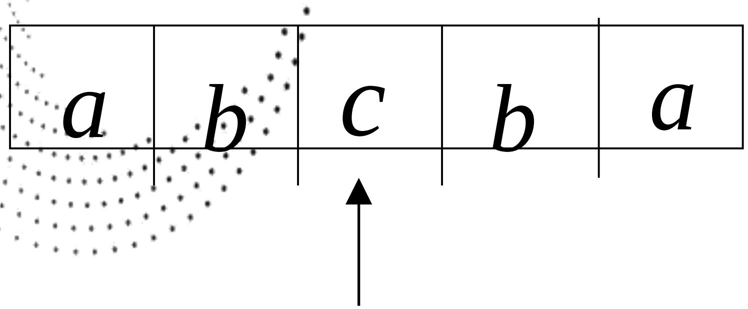


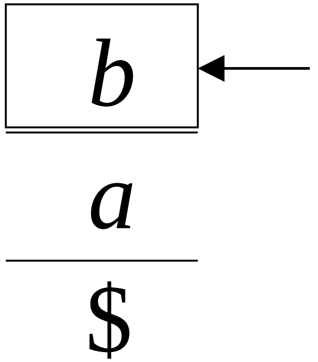
Time 2



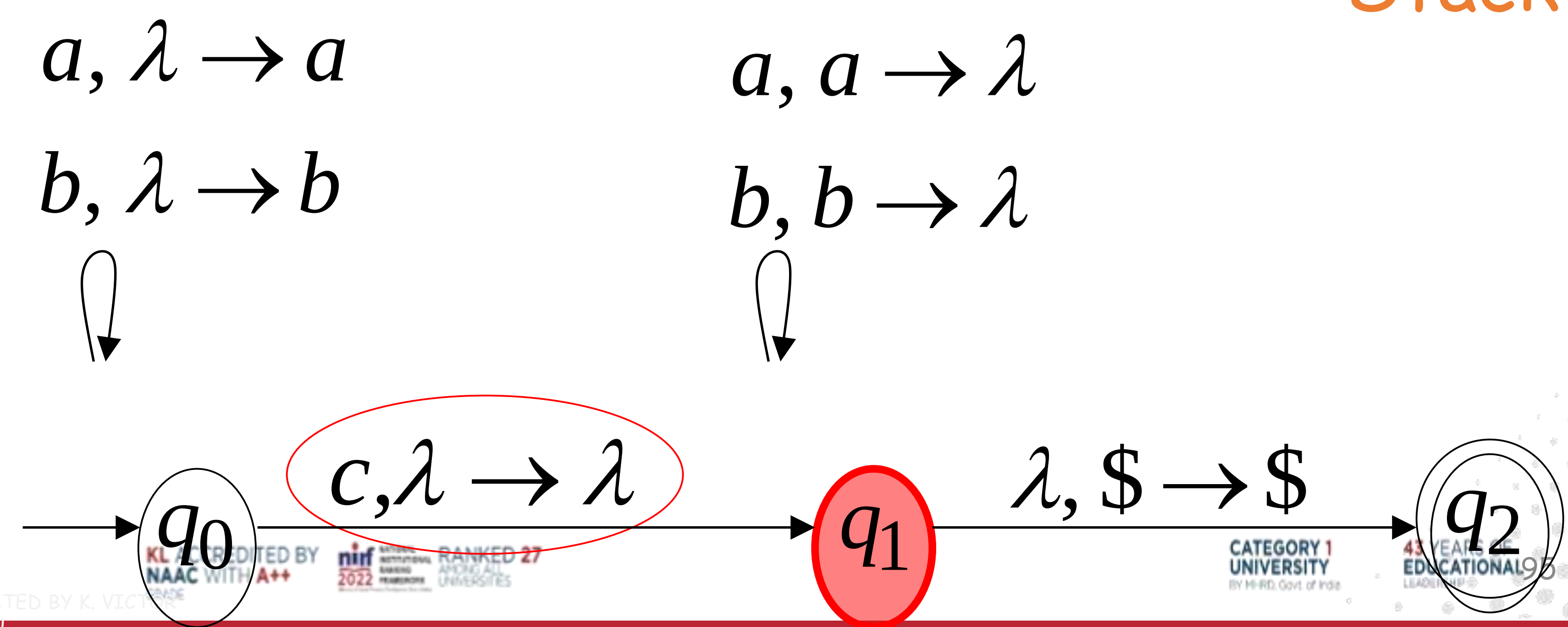
Time 3

Input

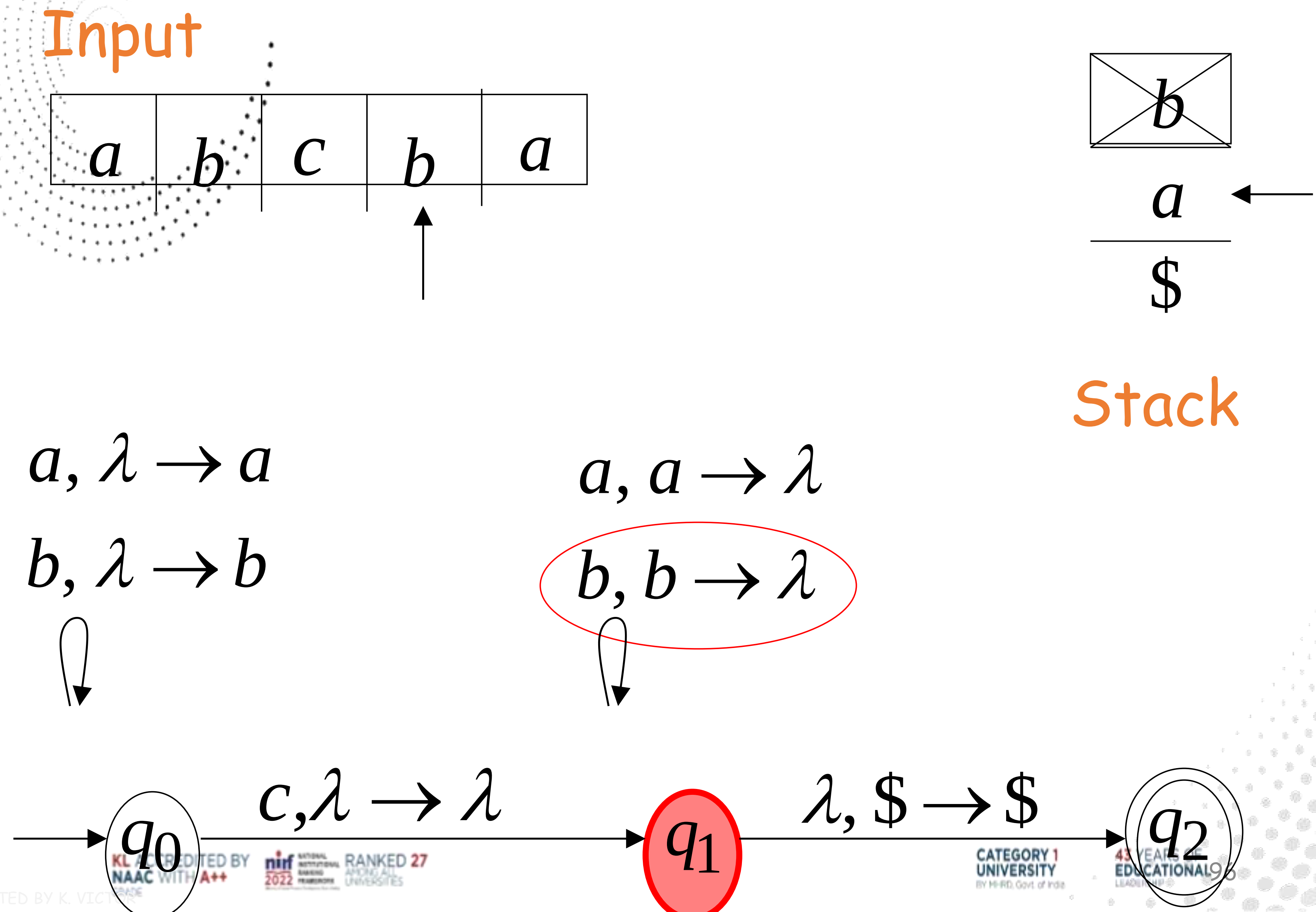




Stack

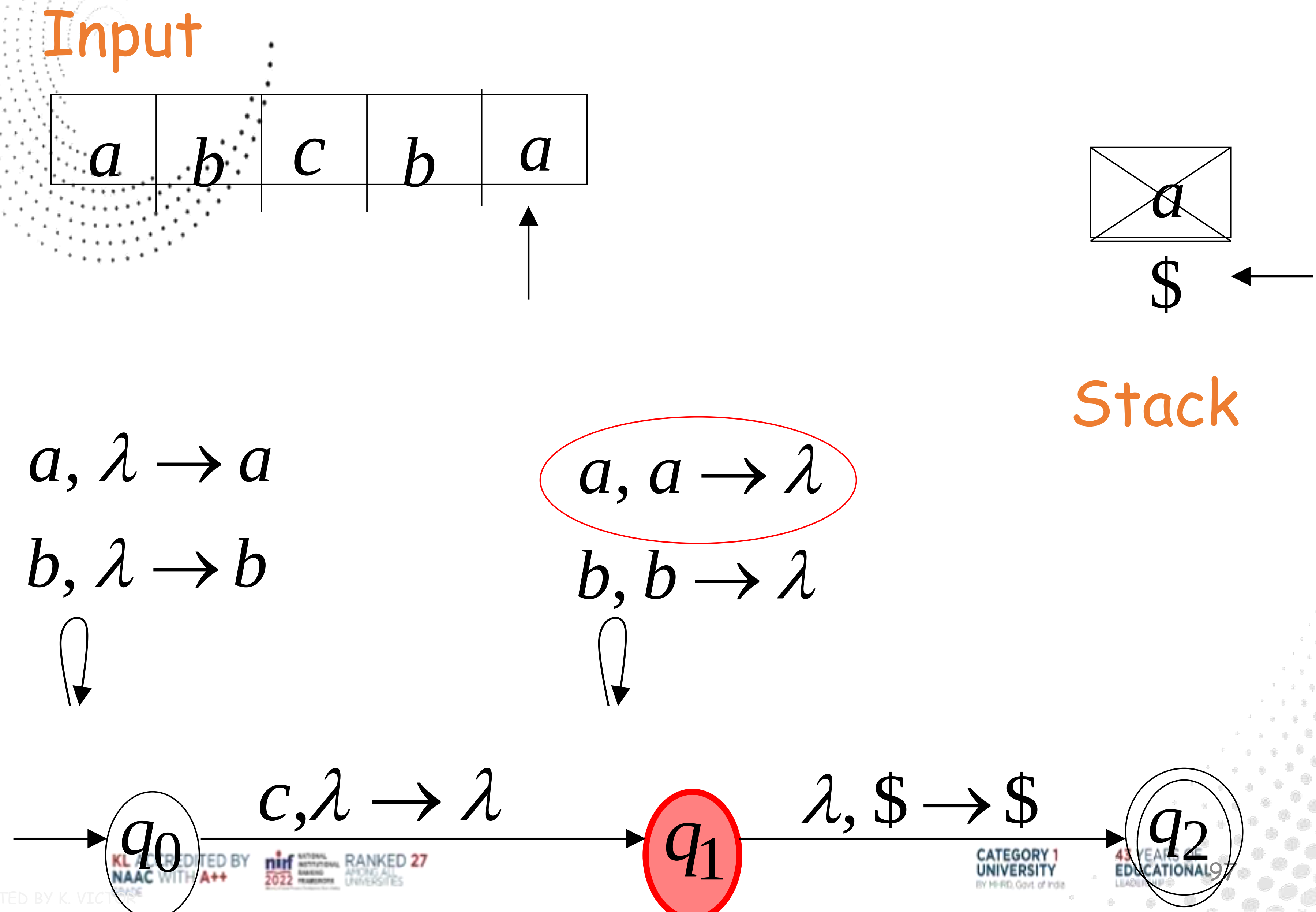


Time 4



Input
a
b
c
b
a
b
a
$
Stack
$a, \lambda \to a$
$b, \lambda \to b$
$a, a \to \lambda$
$b, b \to \lambda$
$c, \lambda \to \lambda$
$\lambda, \$ \to \$$
q_0
q_1
q_2

Time 5



Input
a b c b a
a
$
Stack
$a, \lambda \rightarrow a$
$b, \lambda \rightarrow b$
$a, a \rightarrow \lambda$
$b, b \rightarrow \lambda$
$c, \lambda \rightarrow \lambda$
$\lambda, \$ \rightarrow \$$
q_0
q_1
q_2

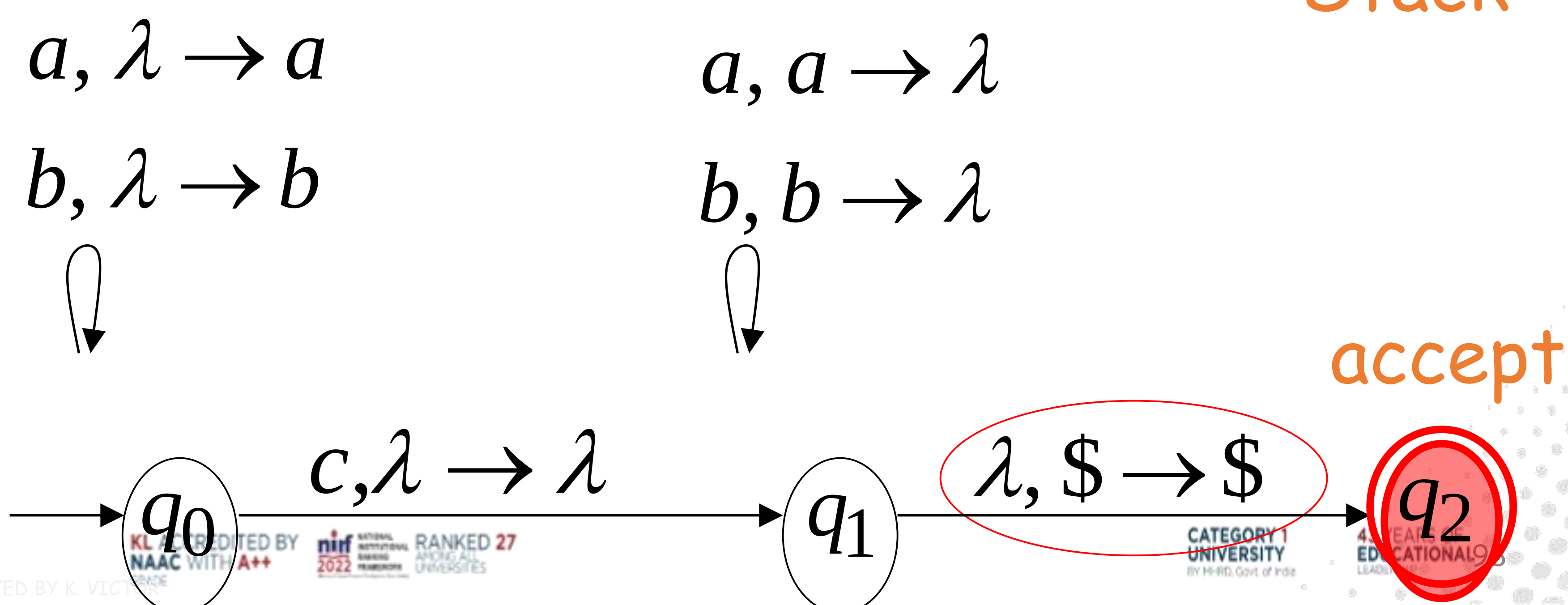
Time 6

Input

a	b	c	b	a

$$\$$$

Stack



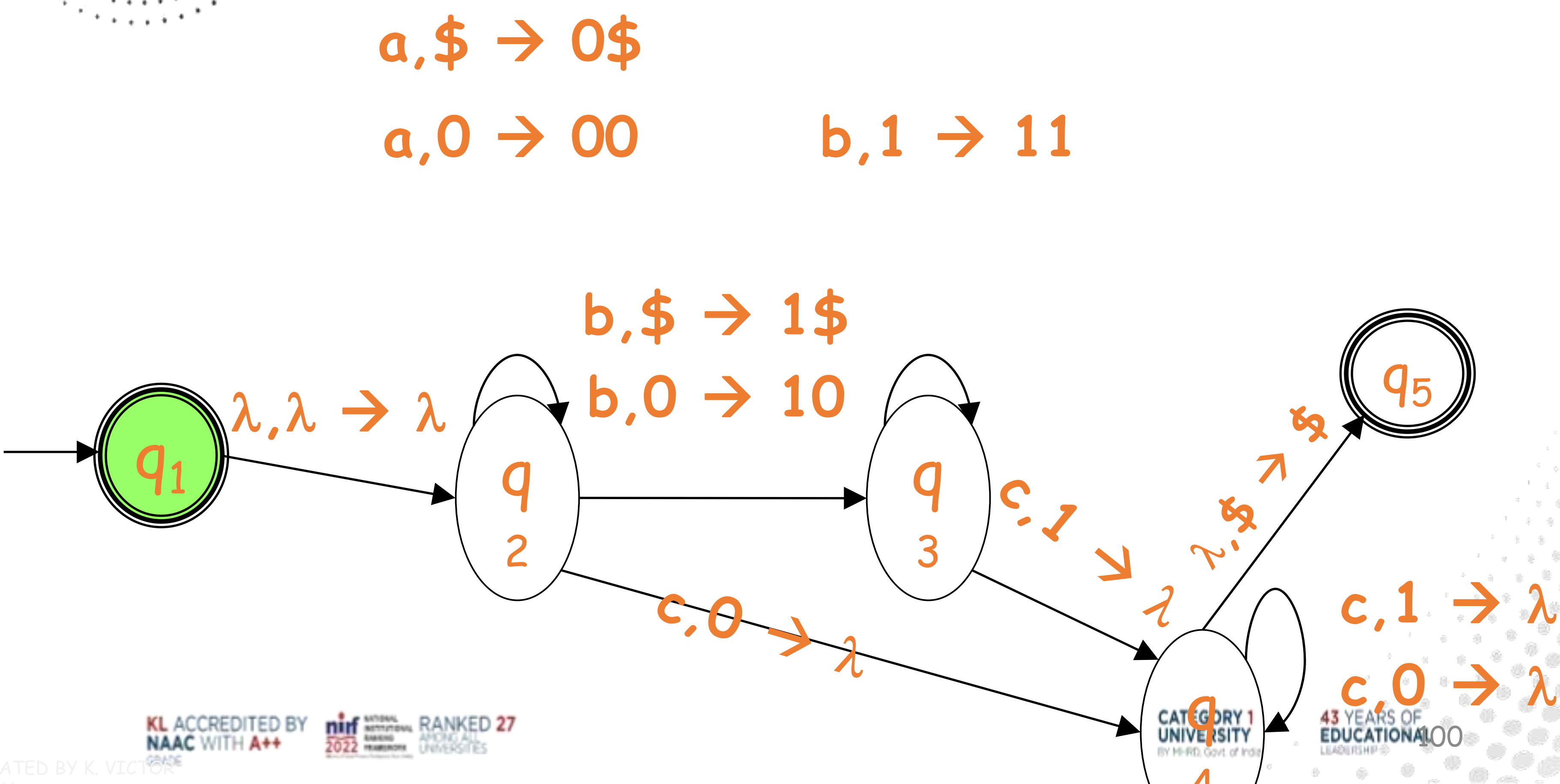
Problems

6:

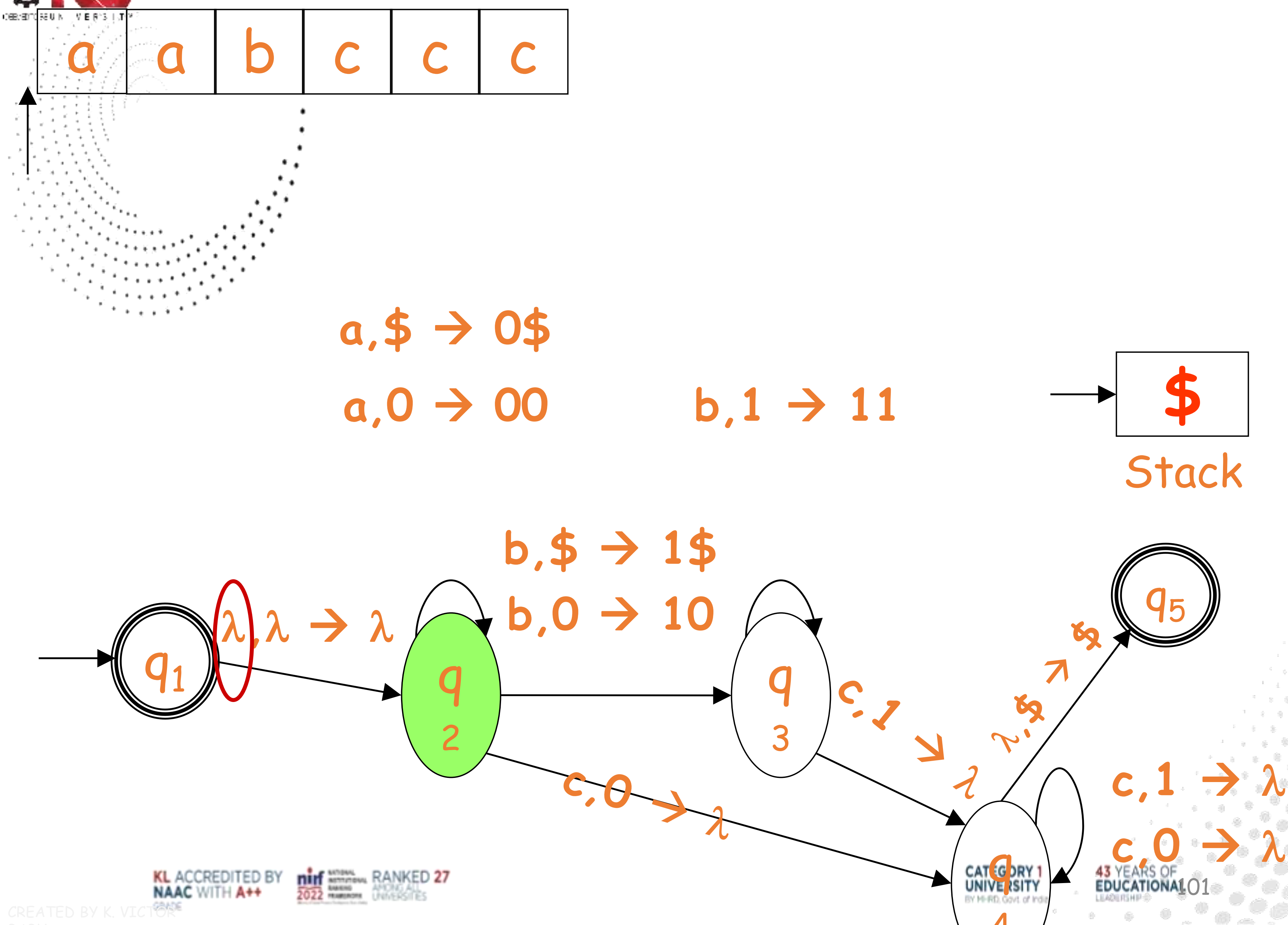
Design a NPDA to accept the following Language:

$$L(M) = \{ a^n b^m c^{n+m} , n \geq 0, m \geq 0 \}$$

$L(M) = \{ a^n b^m c^{n+m}, n \geq 0, m \geq 0 \}$

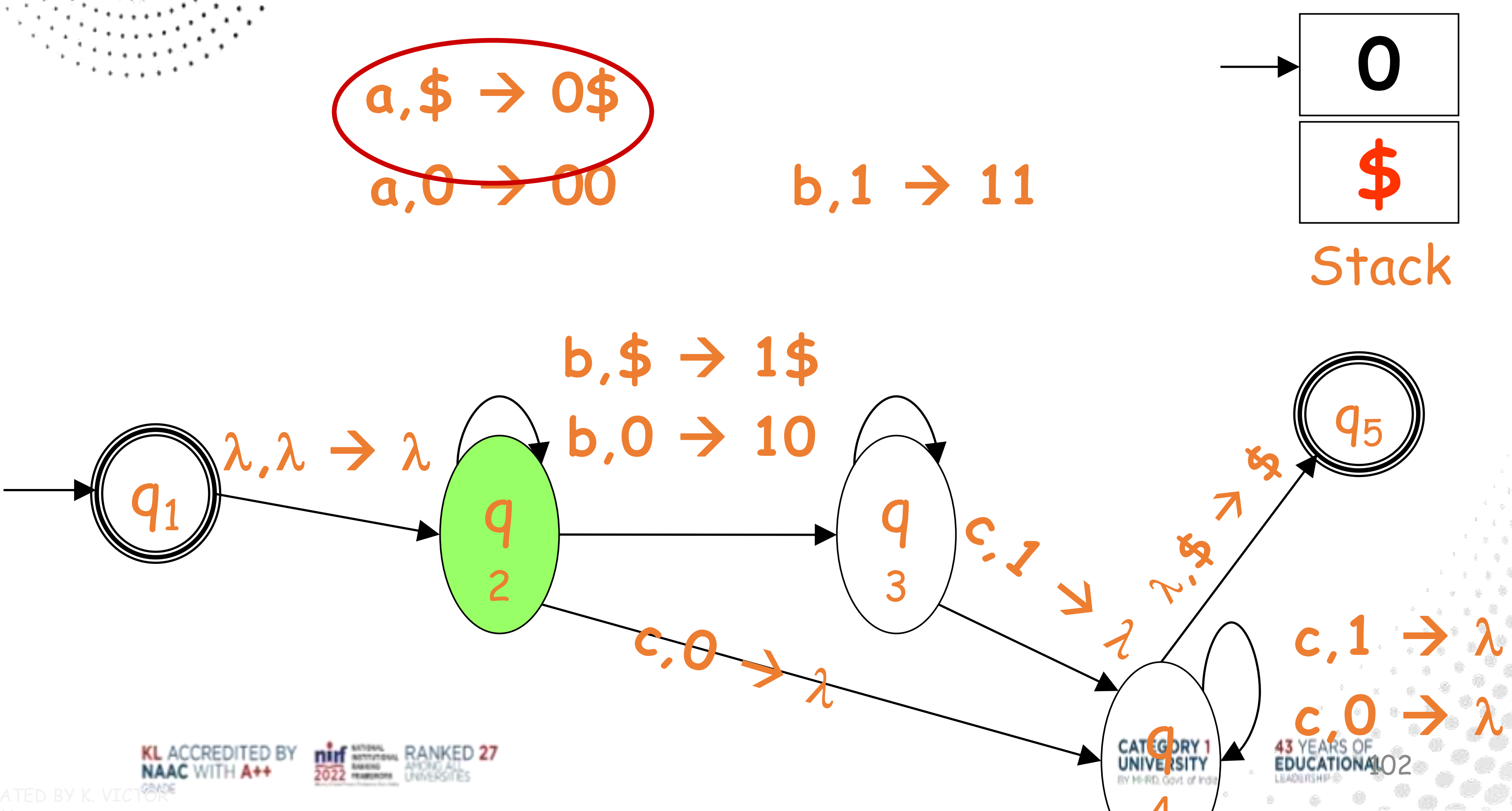


Input String

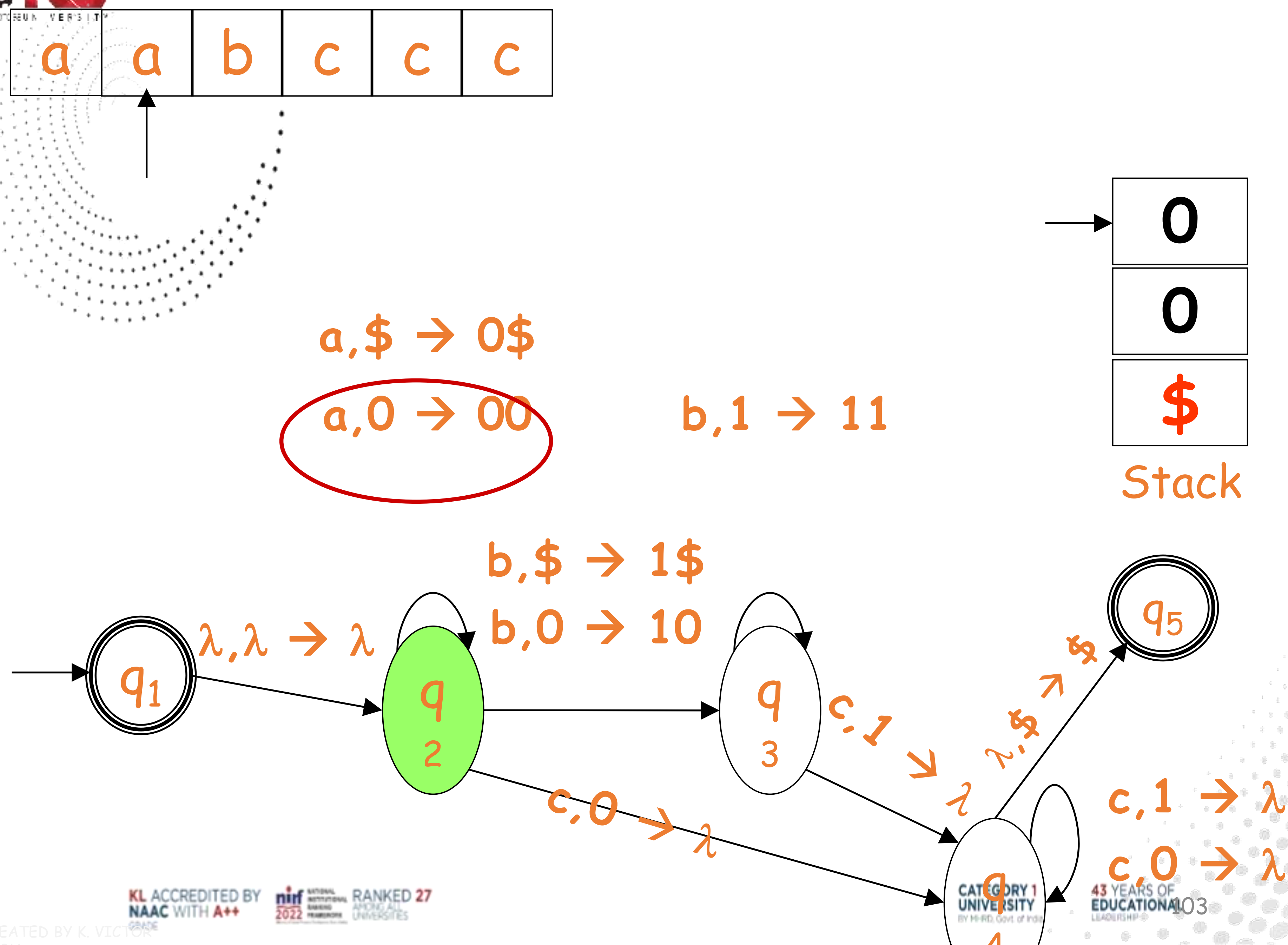


Input String

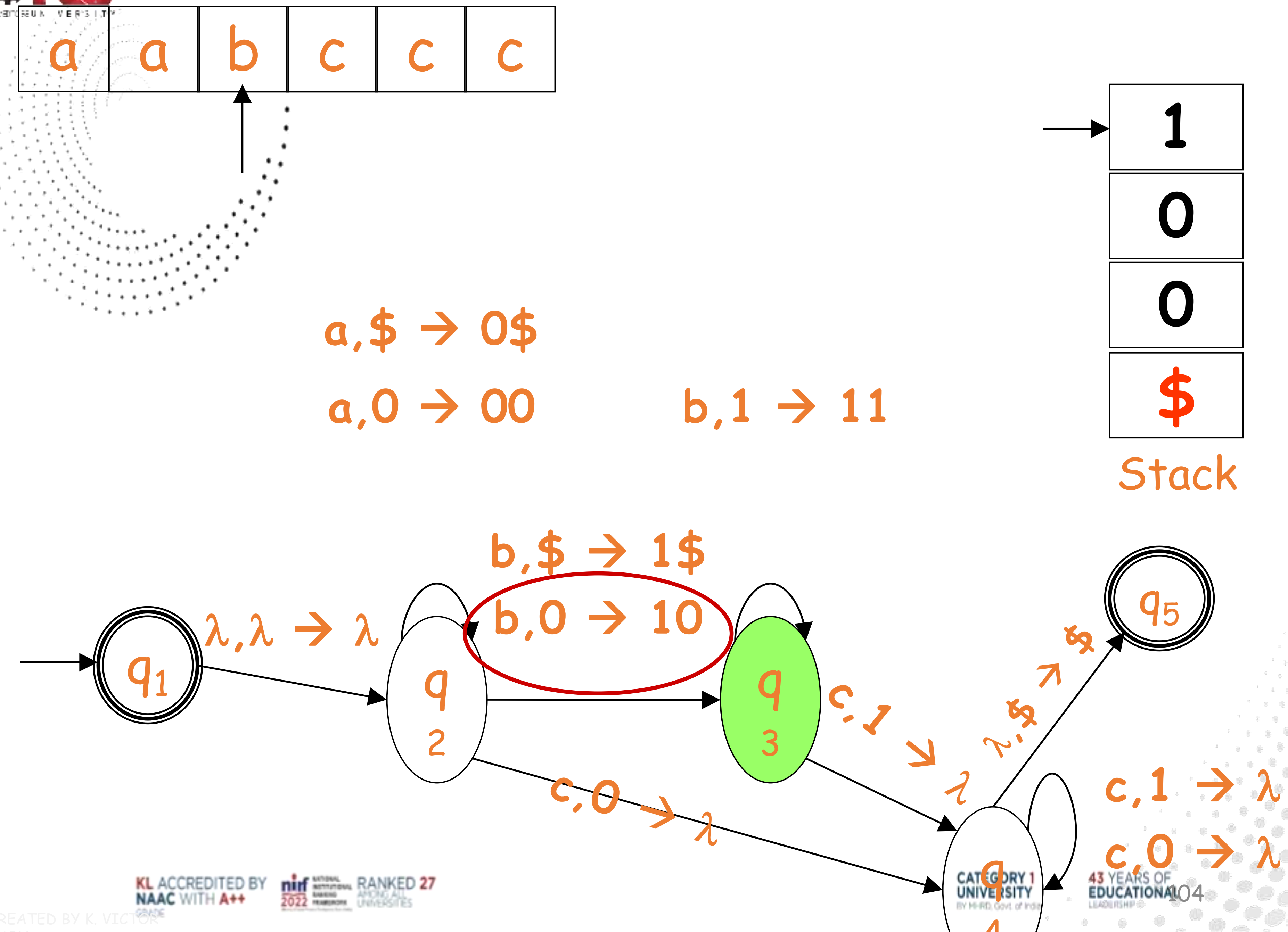
a	a	b	c	c	c



Input String



Input String



Input String

a	a	b	c	c	c

a,$ → 0$

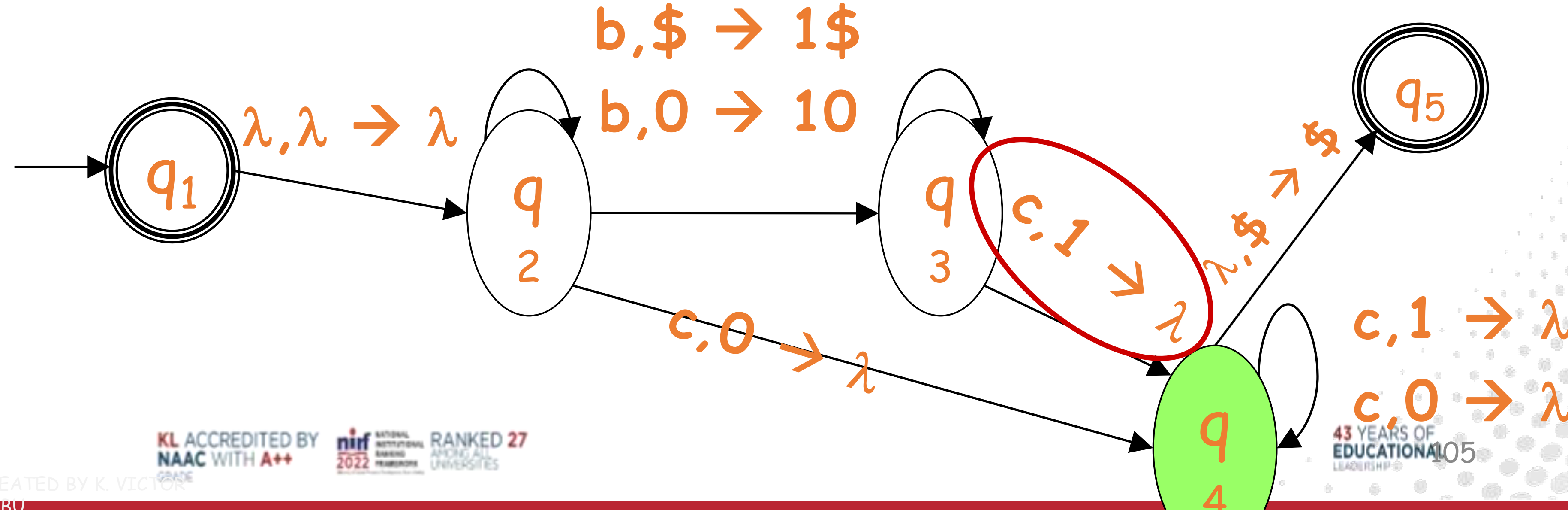
a,0 → 00

b,1 → 11

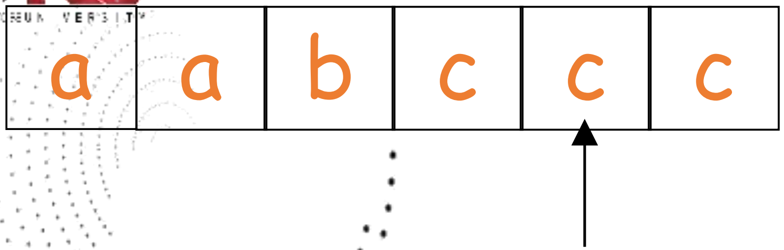
Stack:

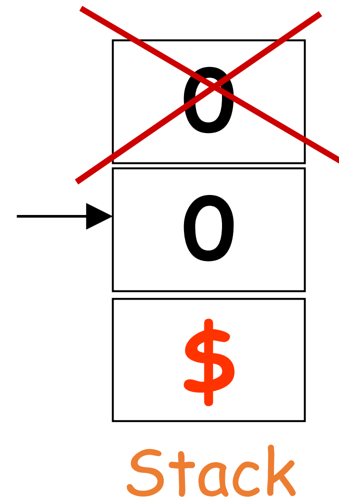
~~1~~
0
0
$

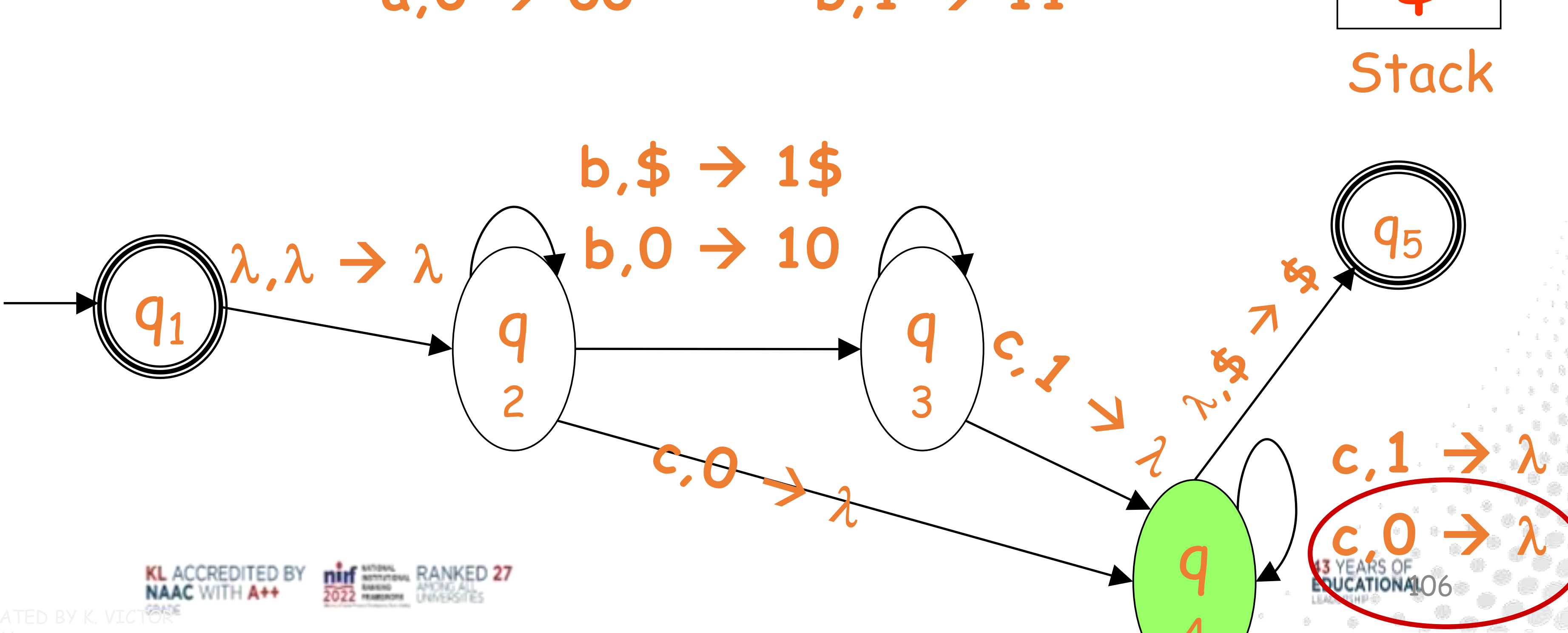
Stack



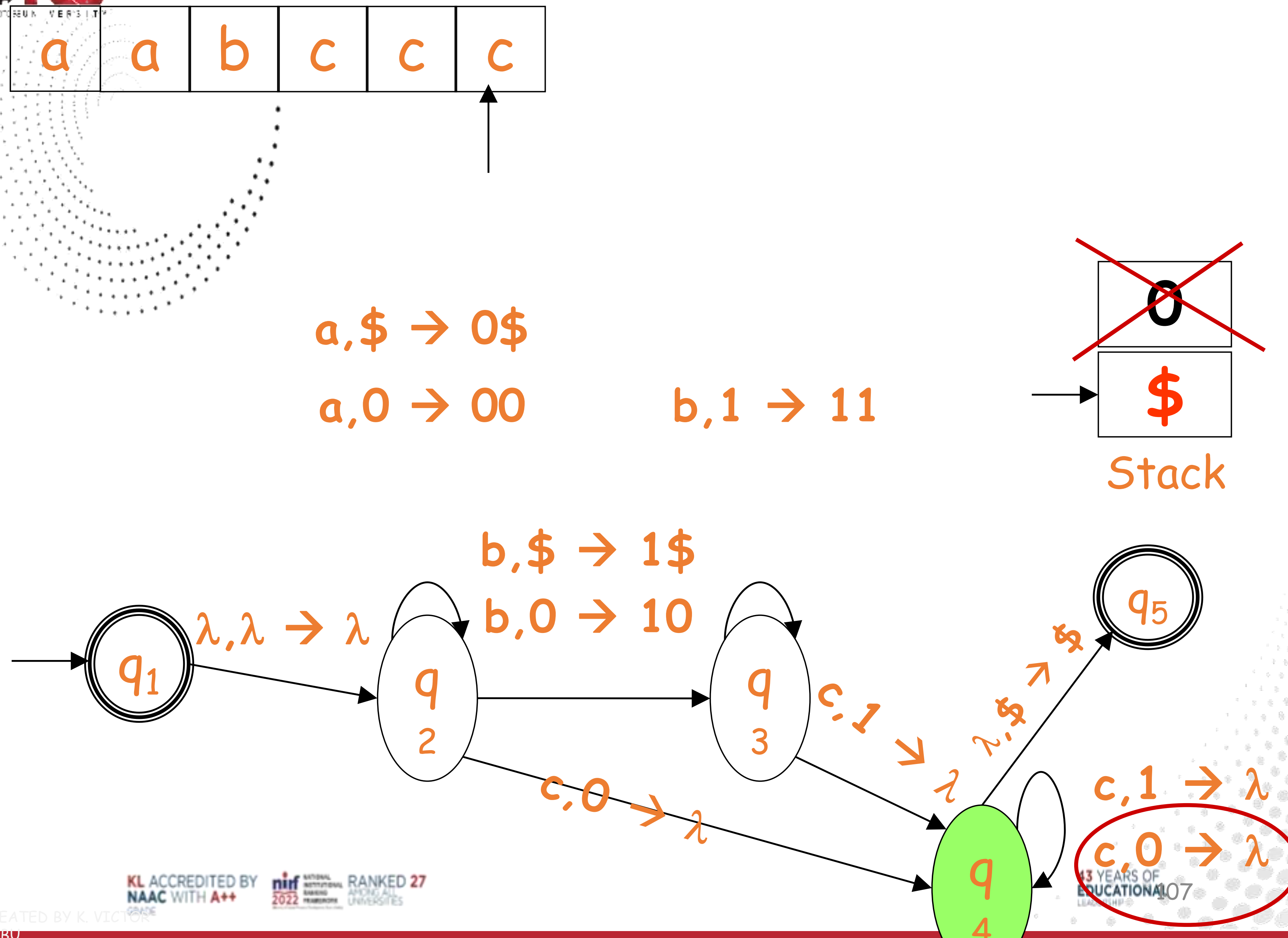
Input String



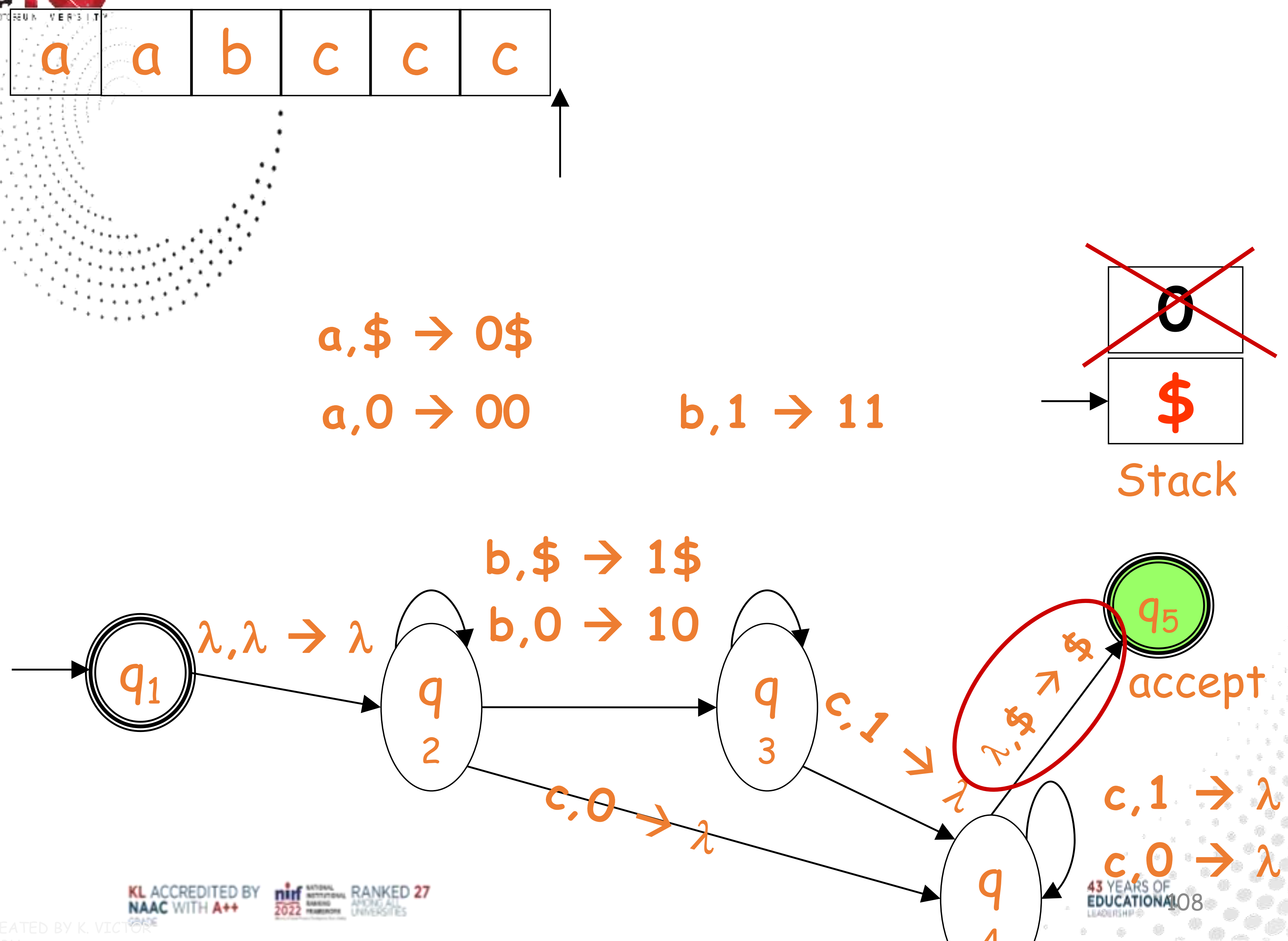




Input String



Input String



Self Assessment Questions

- Q.1. The power of a Pushdown Automaton (PDA) lies in its ability to recognize:
- a) Context-free languages.
- b) Regular languages.
- c) Recursively enumerable languages.
- d) Context-sensitive languages.

Answer: a) Context-free languages.

Self Assessment Questions

Q.2. The stack in a Pushdown Automaton (PDA) is used to:

- a) Store the input symbols.
- b) Perform arithmetic calculations.
- c) Track the position in the input tape.
- d) Track the context of the computation.

Answer: d) Track the context of the computation.

Self Assessment Questions

Q.3) Which of the following is not a component of a Pushdown Automaton (PDA)?

a) Input alphabet.

b) b) Stack alphabet.

c) c) Transition table.

d) d) Output tape.

Answer: d)Output tape.

Terminal Questions

- Q.1. What is a Pushdown Automaton (PDA), and what are its main components?
- Q.2. Describe the role of the stack in a Pushdown Automaton (PDA) and its significance in language recognition.
- Q.3. Describe the role of the stack in a Pushdown Automaton (PDA) and its significance in language recognition.

THANK YOU

Team – TOC