

# CS 6041

# Theory of Computation

## Pushdown Automata

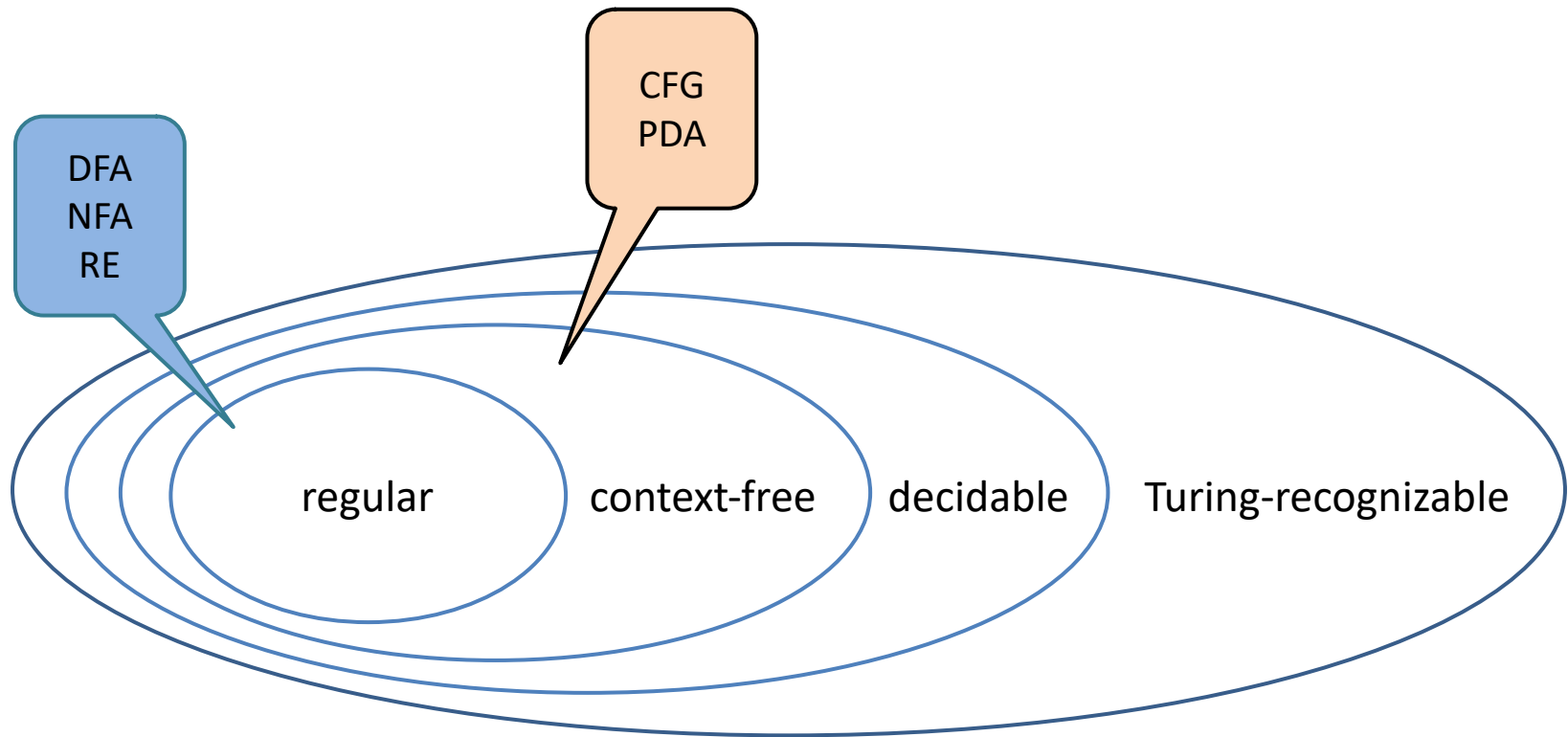
**Kun Suo**

Computer Science, Kennesaw State University

<https://kevinsuo.github.io/>

# Pushdown Automata (PDA)

---



# Pushdown Automata (PDA)

---

- Pushdown automatas are equivalent in power to context-free grammars (PDA=CFG)
- PDA can recognize some nonregular languages

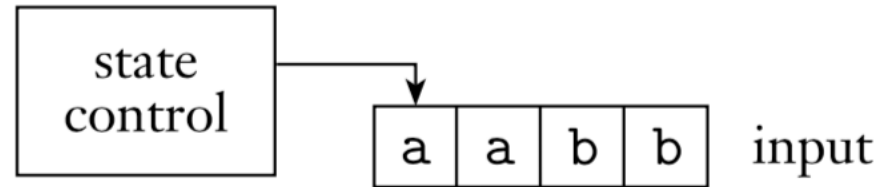


# What does PDA look like?

---

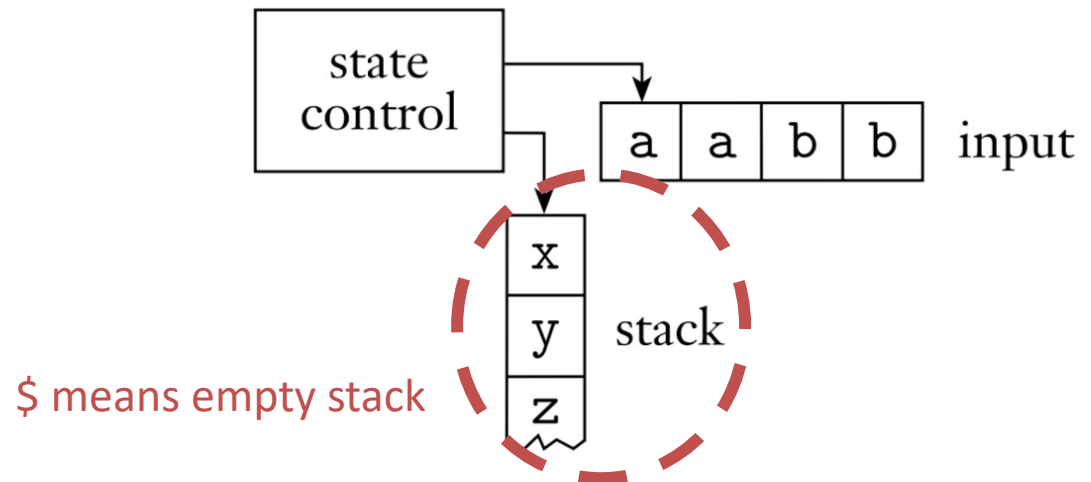
finite automaton

Memory = 1



pushdown automaton

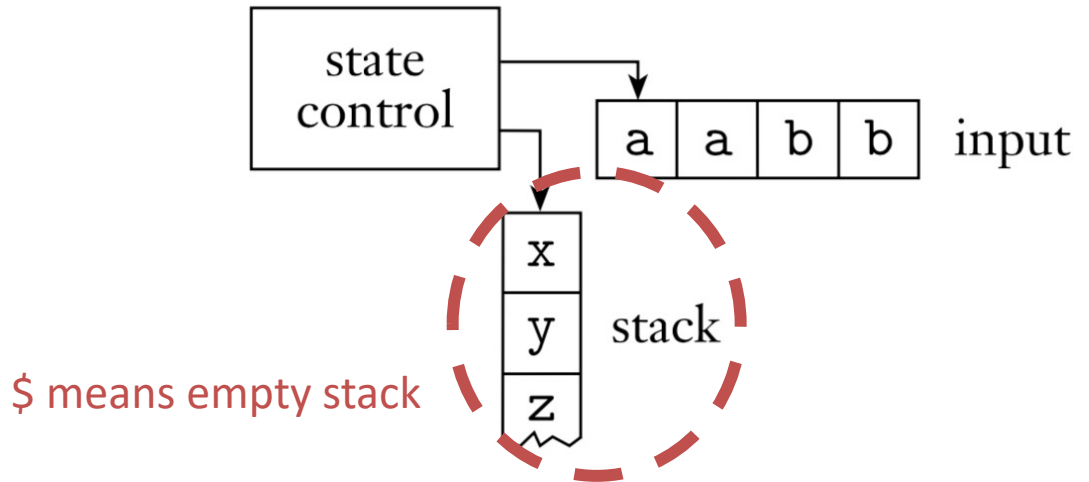
Memory = N



# What does PDA look like?

---

pushdown automaton

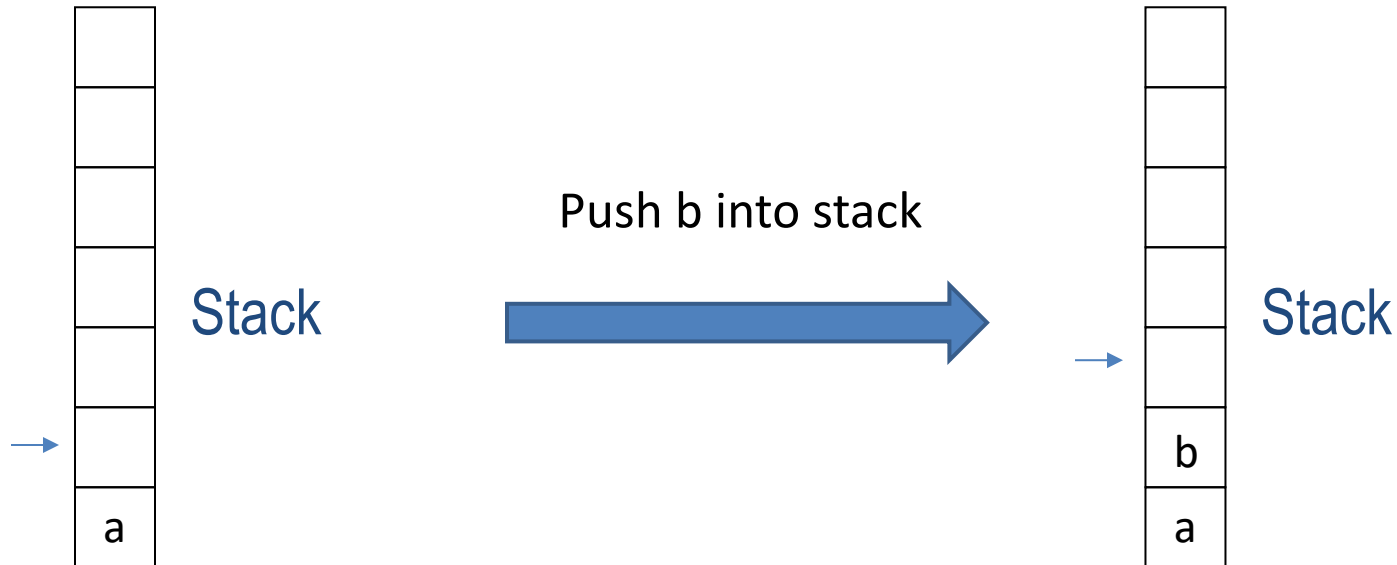


- Pushdown automata has more memories than finite automata
- PDA = finite automata + **A stack (unlimited size)**

# Stack operation

---

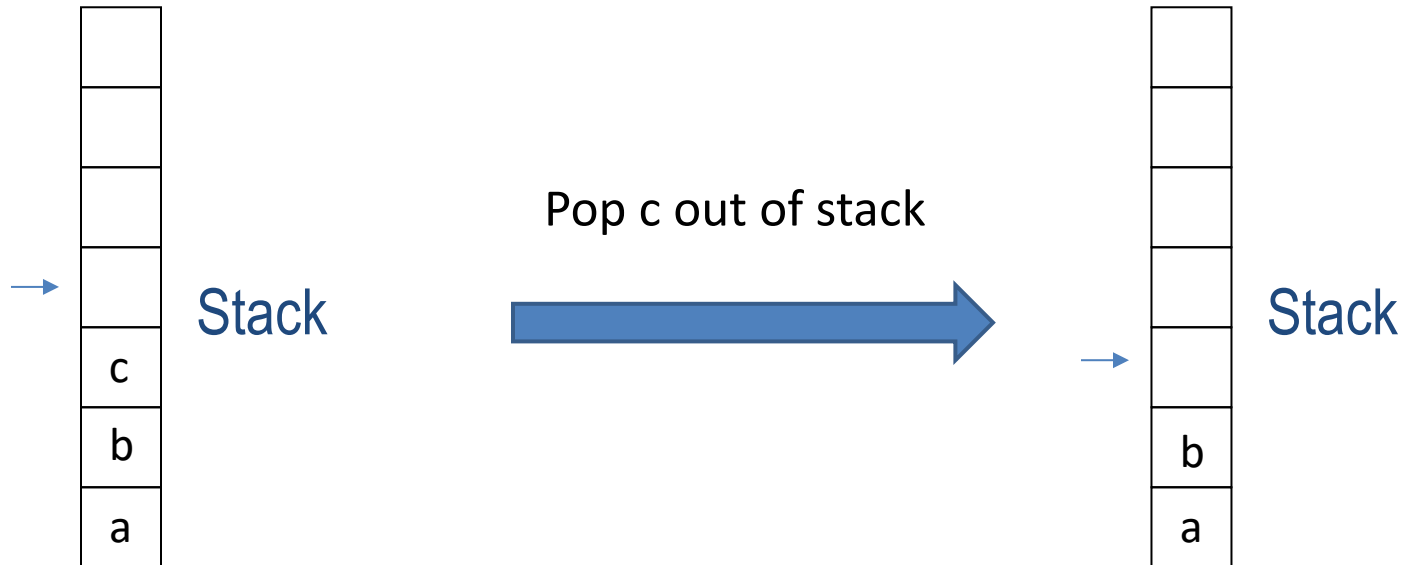
- Push: add to the top of stack



# Stack operation

---

- Pop: remove from the top of stack

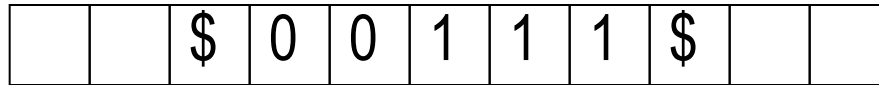


# To test whether 00111 is in $0^n1^n$

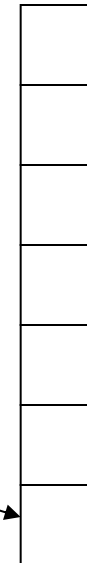
---

Single direction ( $\rightarrow$ ),  
read only input tape

$\{ 0^n1^n \mid n \geq 0 \}$



Single direction  
read only head



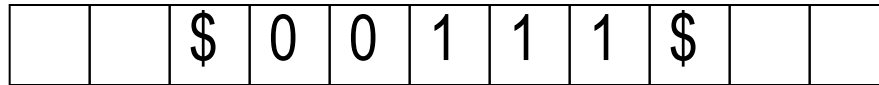
Stack





# To test whether 00111 is in $0^n1^n$

Single direction,  
read only input tape



Single direction  
read only head



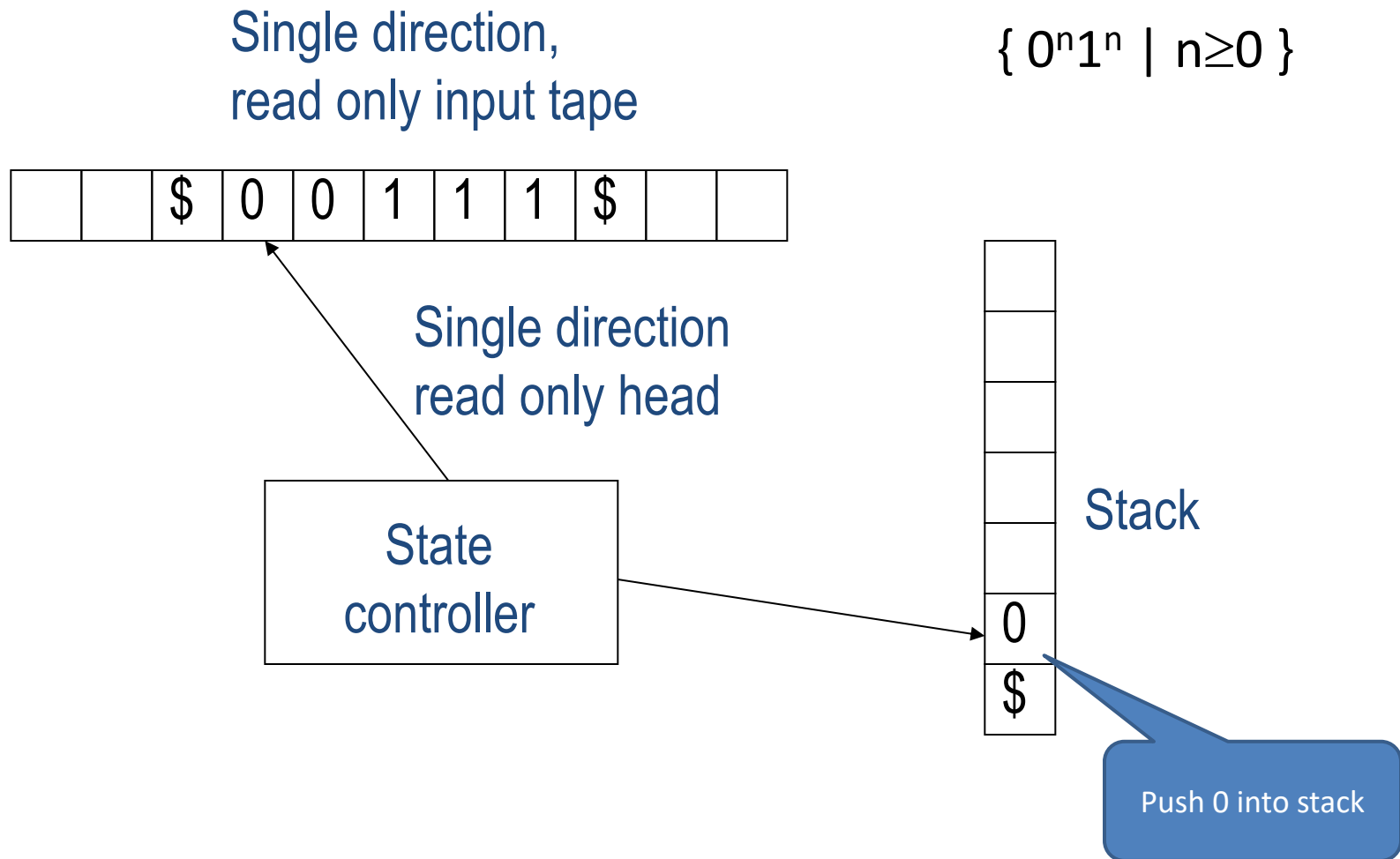
$\{ 0^n1^n \mid n \geq 0 \}$

Stack



Push \$ into stack: means start reading,  
and the stack is empty

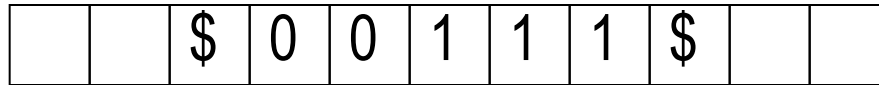
# To test whether 00111 is in $0^n1^n$



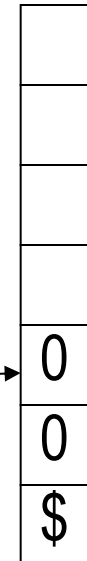
# To test whether 00111 is in $0^n1^n$

Single direction,  
read only input tape

$\{ 0^n1^n \mid n \geq 0 \}$



Single direction  
read only head



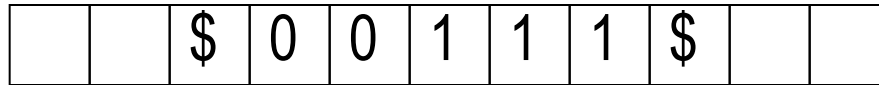
Stack



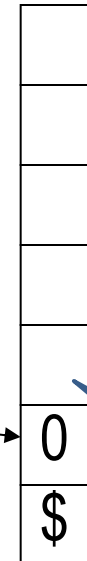
# To test whether 00111 is in $0^n1^n$

Single direction,  
read only input tape

$\{ 0^n1^n \mid n \geq 0 \}$



Single direction  
read only head



Stack

Pop 0 out of stack

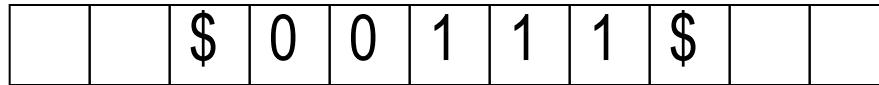


# To test whether 00111 is in $0^n1^n$

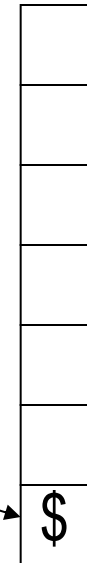
---

Single direction,  
read only input tape

$\{ 0^n1^n \mid n \geq 0 \}$



Single direction  
read only head



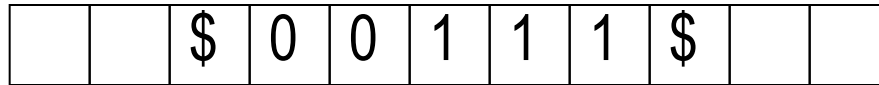
Stack



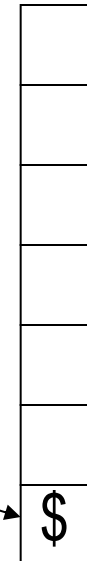
# To test whether 00111 is in $0^n1^n$

Single direction,  
read only input tape

$\{ 0^n1^n \mid n \geq 0 \}$



Single direction  
read only head



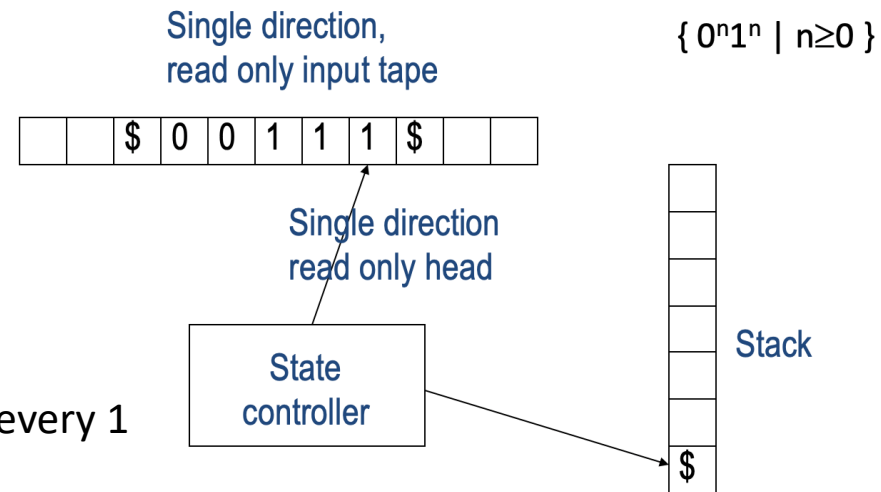
Stack

No 0 to pop  
out of stack



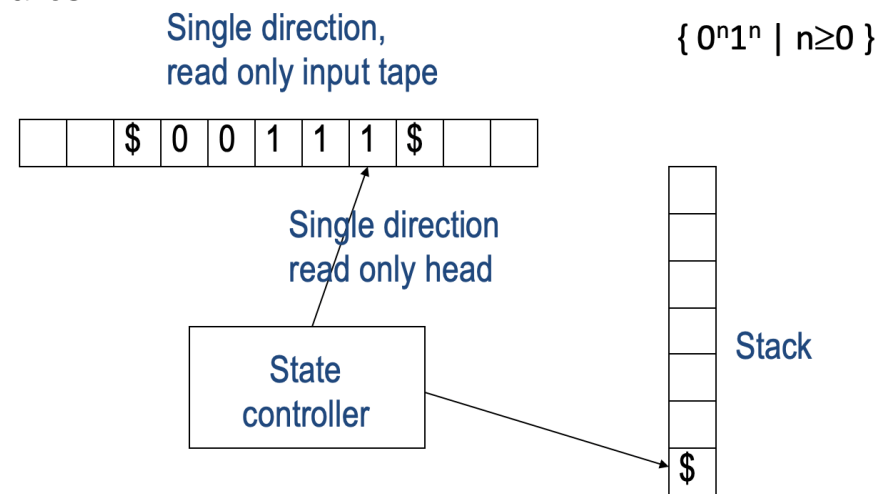
# Informal description for PDA to recognize some languages

- $A = \{0^n 1^n \mid n \geq 0\}$
- Read symbols from input
  - Operation
    - ▶ For every 0s, push 0 into stack
    - ▶ When read 1s, pop one 0 from stack for every 1
  - Determine accept/reject:
    - ▶ When finish reading string and there is no 0s in stack, **accept**;
    - ▶ When there exist 0s after 1s (not in  $0^n 1^n$ ), **reject**.
    - ▶ When tape is not finished while the stack is empty ( $1 > 0$ ), **reject**;
    - ▶ When tape finished while the stack is non-empty ( $1 < 0$ ) **reject**;



# Informal description for PDA to recognize some languages

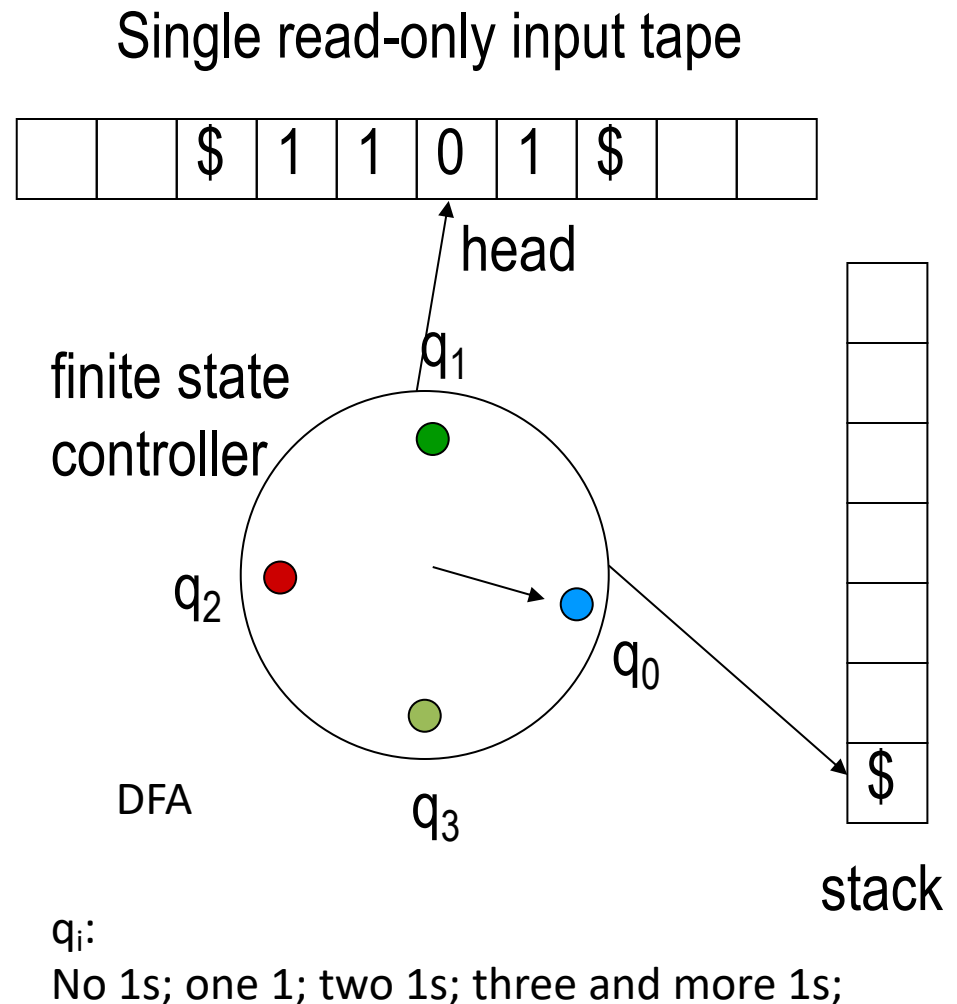
- $L = \{w \mid w \text{ has some features}\}$
- Read symbols from input
  - **STEP1: regular?**
    - ▶ If the language is regular, do not need to use stack; if not regular, define operations on stack
  - **STEP2: define operations:**
    - ▶ When to push
    - ▶ When to pop
  - **STEP 3: determine accept/reject:**
    - ▶ Under which cases, accept
    - ▶ Under which cases, reject





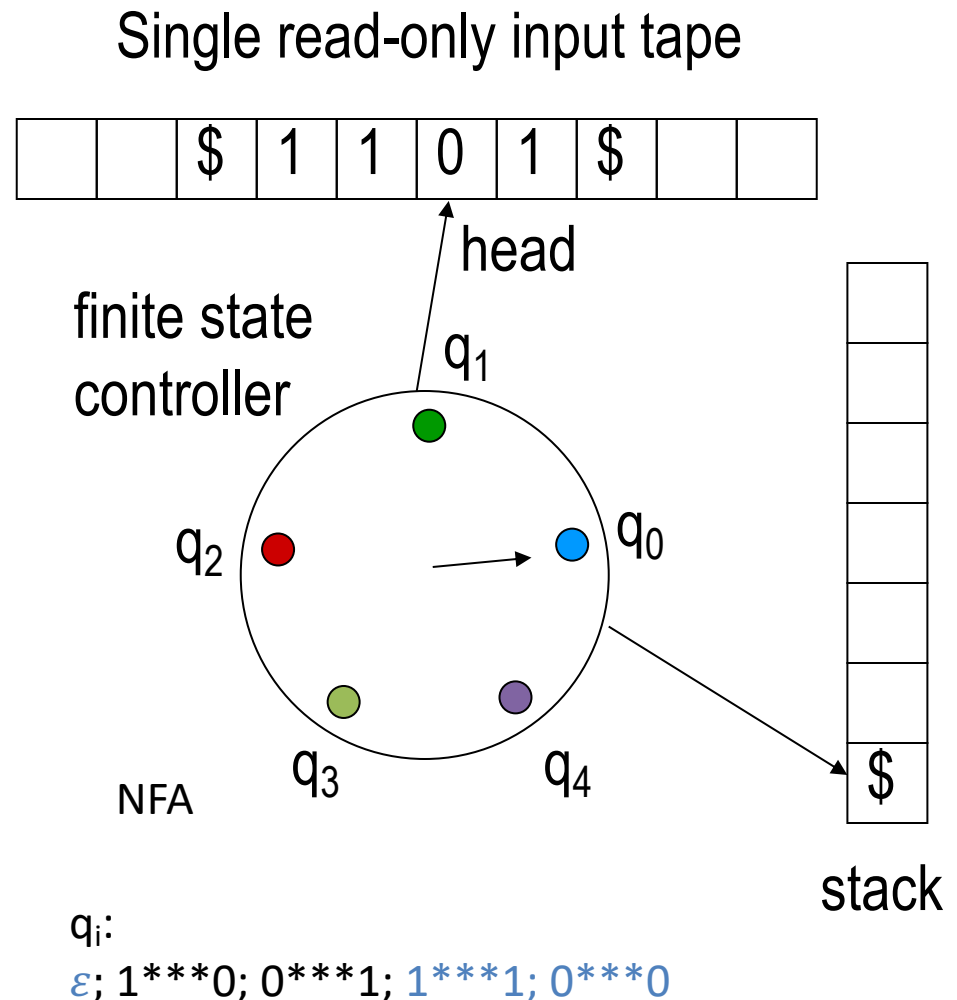
# Question: Informal description

- $L1 = \{w \mid w \text{ has at least three } 1\text{s}\}$ 
  - This set is regular ( $\Sigma^*1\Sigma^*1\Sigma^*1\Sigma^*$ ), so the PDA doesn't even need to use its stack.
  - The PDA scans the string and uses its finite control to maintain a counter which counts up to 3. The PDA accepts the moment it sees three ones.



# Question: Informal description

- $L_2 = \{w \mid w \text{ starts and ends with the same symbol}\}$ 
  - This set is regular ( $0(\Sigma)^*0 \cup 1(\Sigma)^*1$ ), so the PDA doesn't even need to use its stack.
  - The PDA scans the string and keep track of the first and last symbol in its finite control. If they are the same, accepts.



# Question: Informal description

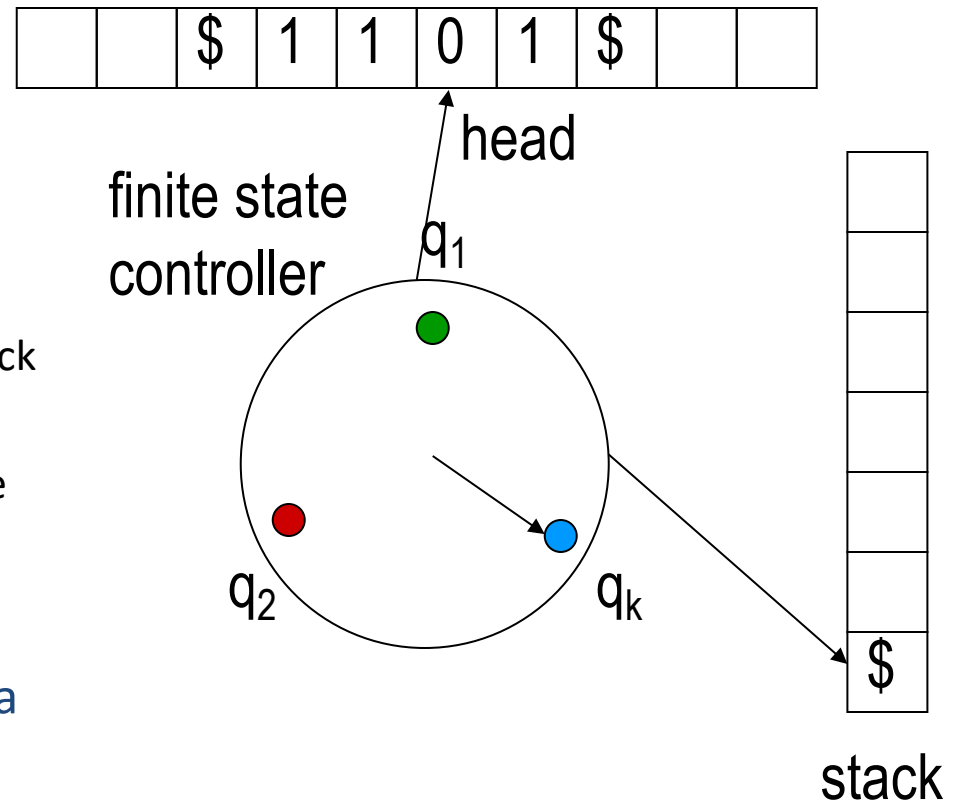
- $L_3 = \{w \mid w \text{ has more 1s than 0s}\}$       Single read-only input tape

- This set is not regular.

- The PDA scans across the input.

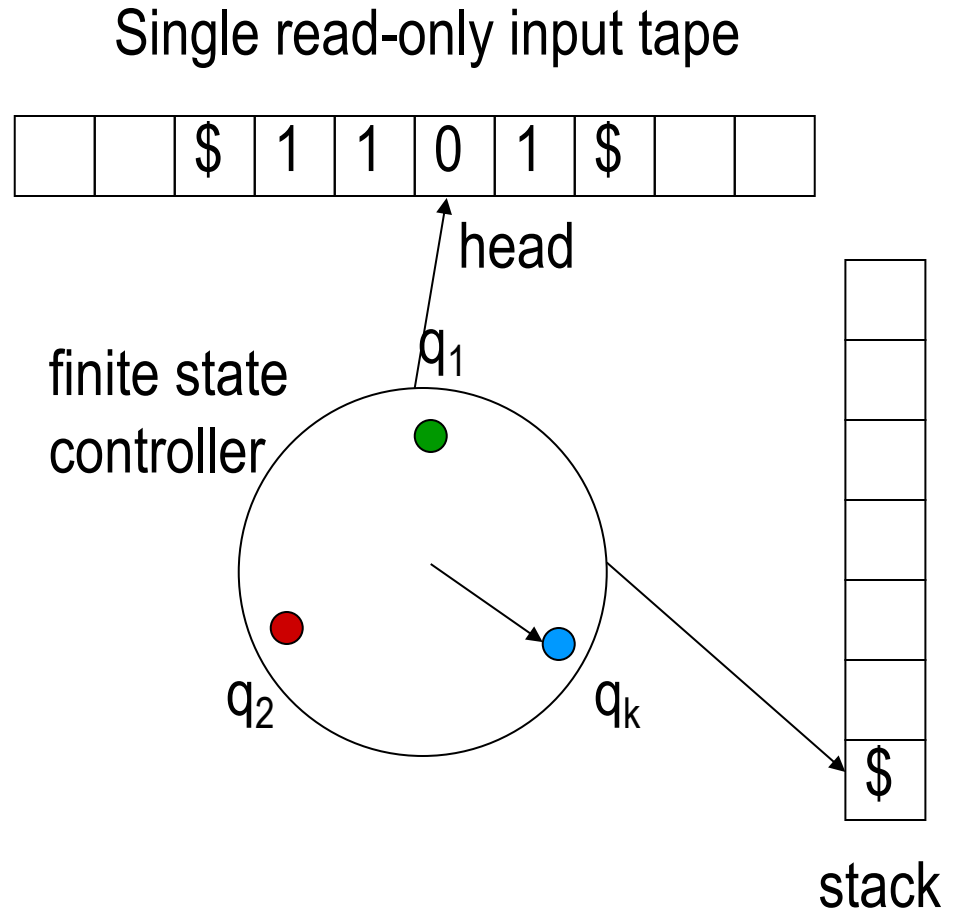
- ▶ POP: If it sees a 1 and its top stack symbol is a 0, it pops the stack. Similarly, if it scans a 0 and its top stack symbol is a 1, it pops the stack.
- ▶ PUSH: In all other cases, it pushes the input symbol onto the stack.

- After it scans the input, if there is a 1 on top of the stack, it accepts. Otherwise it rejects.



## Question: Informal description

- $L_4 = \emptyset$ 
  - Just reject.



# Definition of PDA (non-deterministic)

---

- PDA  $M=(Q,\Sigma,\Gamma,\delta,q_0,F)$ , where
  - 1)  $Q$ : set of states
  - 2)  $\Sigma$ : input alphabet,  $\Sigma_\epsilon=\Sigma\cup\{\epsilon\}$
  - 3)  $\Gamma$ : stack alphabet,  $\Gamma_\epsilon=\Gamma\cup\{\epsilon\}$
  - 4)  $\delta: Q\times\Sigma_\epsilon\times\Gamma_\epsilon\rightarrow P(Q\times\Gamma_\epsilon)$ ,  
transition function
  - 5)  $q_0\in Q$ : start state
  - 6)  $F\subseteq Q$ : accept state set



# PDA vs. NFA

---

A *pushdown automaton* is a 6-tuple  $(Q, \Sigma, \Gamma, \delta, q_0, F)$ , where  $Q$ ,  $\Sigma$ ,  $\Gamma$ , and  $F$  are all finite sets, and

1.  $Q$  is the set of states,
2.  $\Sigma$  is the input alphabet,
3.  $\Gamma$  is the stack alphabet,
4.  $\delta: Q \times \Sigma_\epsilon \times \Gamma_\epsilon \longrightarrow \mathcal{P}(Q \times \Gamma_\epsilon)$  is the transition function,
5.  $q_0 \in Q$  is the start state, and
6.  $F \subseteq Q$  is the set of accept states.

A *nondeterministic finite automaton* is a 5-tuple  $(Q, \Sigma, \delta, q_0, F)$ , where

1.  $Q$  is a finite set of states,
2.  $\Sigma$  is a finite alphabet,
3.  $\delta: Q \times \Sigma_\epsilon \longrightarrow \mathcal{P}(Q)$  is the transition function,
4.  $q_0 \in Q$  is the start state, and
5.  $F \subseteq Q$  is the set of accept states.

# Computation on PDA

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

input  $w = w_1 w_2 \dots w_m,$

$w_i \in \Sigma_\epsilon$

- Computation: (state, stack)

$(r_0, s_0), (r_1, s_1), \dots, (r_m, s_m),$

Where  $r_i \in Q, s_i \in \Gamma^*$ , satisfying

1)  $(r_0, s_0) = (q_0, \epsilon);$

At first, the first state is  $q_0$  and stack is empty

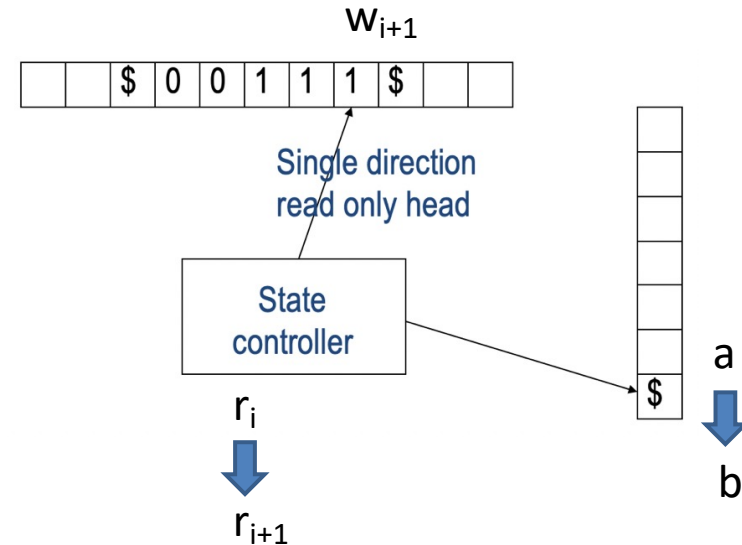
2)  $(r_{i+1}, b) \in \delta(r_i, w_{i+1}, a);$

where  $s_i = at; s_{i+1} = bt,$

$a, b \in \Sigma_\epsilon,$

$t \in \Gamma^*$  (t are other elements in stack)

After input  $w_{i+1}$ , state changes from  $r_i$  to  $r_{i+1}$  and the top element in stack changes from  $a$  to  $b$



# Computation on PDA

---

- Accept of computation:

3)  $r_m \in F$ ;

- M accepts w:

M is at accept states after input of w

- The language that M accepts:

$$L(M) = \{ x \mid M \text{ accepts } x \}$$

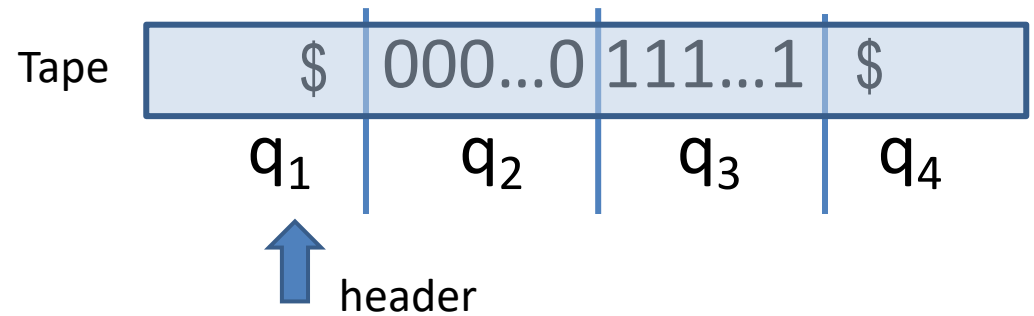




# PDA example

- $L = \{0^n 1^n \mid n \geq 0\}$
- $M_1 = (\{q_1, q_2, q_3, q_4\}, \{0, 1\}, \{0, \$\}, \delta, q_1, \{q_1, q_4\})$

Can you explain what this PDA means?



## Definition of PDA (non-deterministic)

- PDA  $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ , where

- 1)  $Q$ : set of states
- 2)  $\Sigma$ : input alphabet,  $\Sigma_\epsilon = \Sigma \cup \{\epsilon\}$
- 3)  $\Gamma$ : stack alphabet,  $\Gamma_\epsilon = \Gamma \cup \{\epsilon\}$
- 4)  $\delta: Q \times \Sigma_\epsilon \times \Gamma_\epsilon \rightarrow P(Q \times \Gamma_\epsilon)$ ,  
transition function
- 5)  $q_0 \in Q$ : start state
- 6)  $F \subseteq Q$ : accept state set

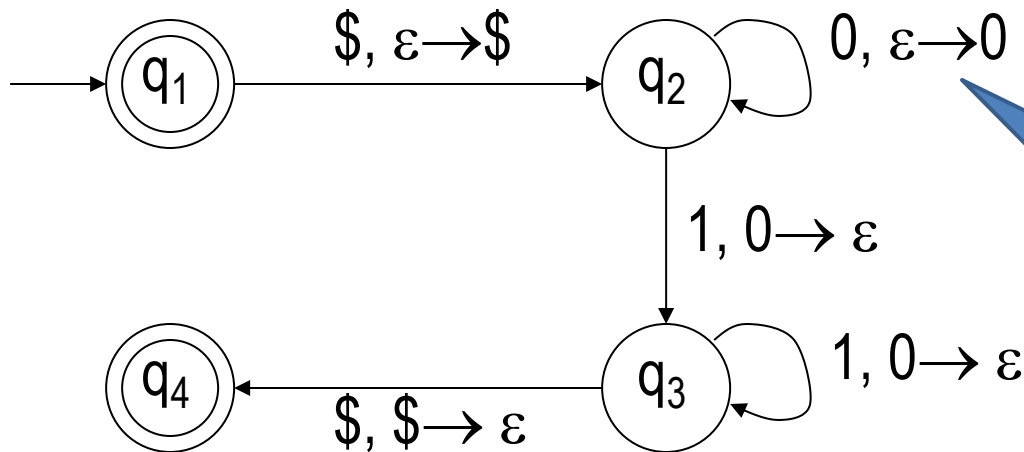


# PDA example

\$	000...0	111...1	\$
$q_1$	$q_2$	$q_3$	$q_4$

- $L = \{0^n 1^n \mid n \geq 0\}$
- $M_1 = (\{q_1, q_2, q_3, q_4\}, \{0, 1\}, \{0, \$\}, \delta, q_1, \{q_1, q_4\})$

We only put 0 or \$ into stack

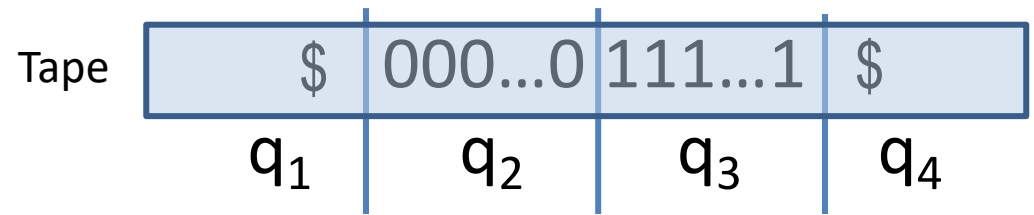


$a, b \rightarrow c$   
 $a$ : input  
 $b \rightarrow c$ : the top of stack changes

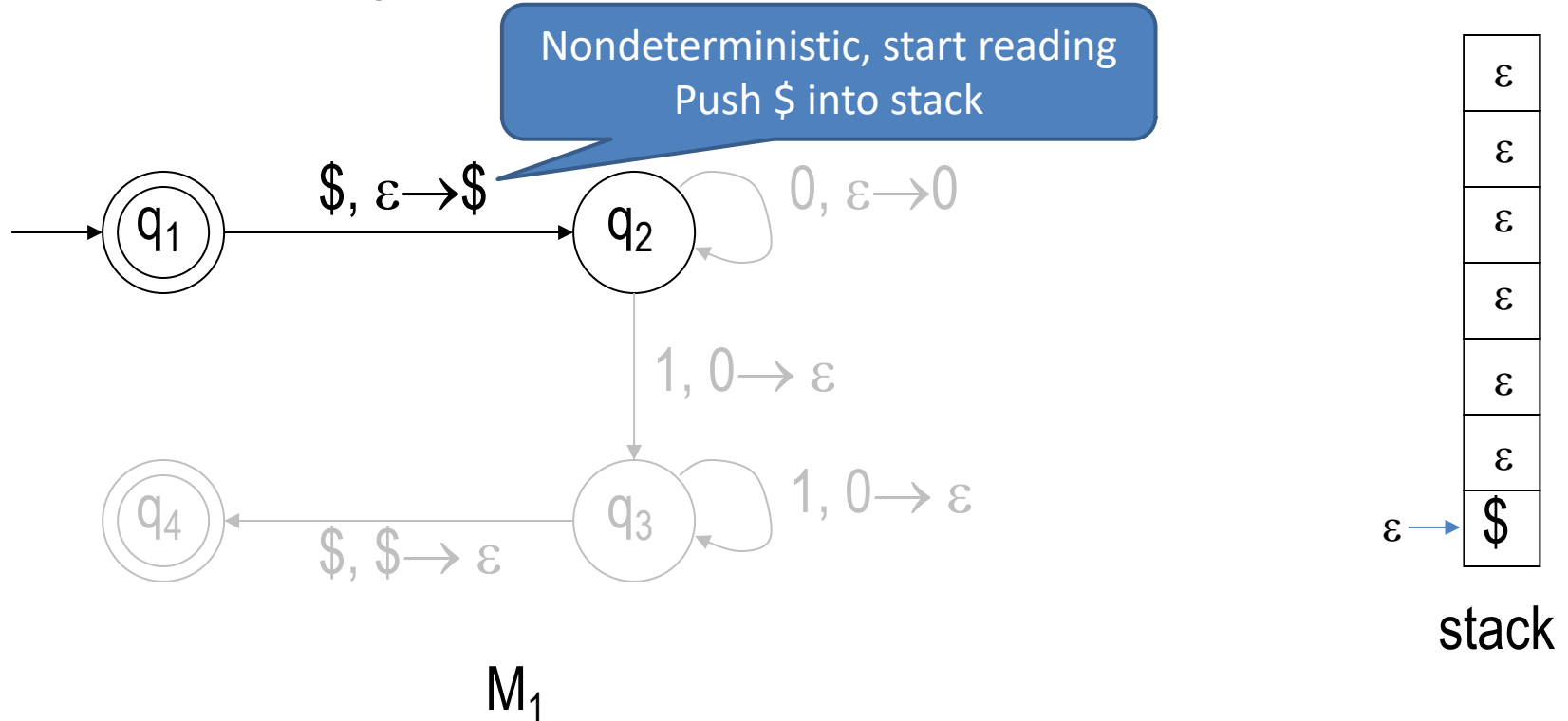
$M_1$



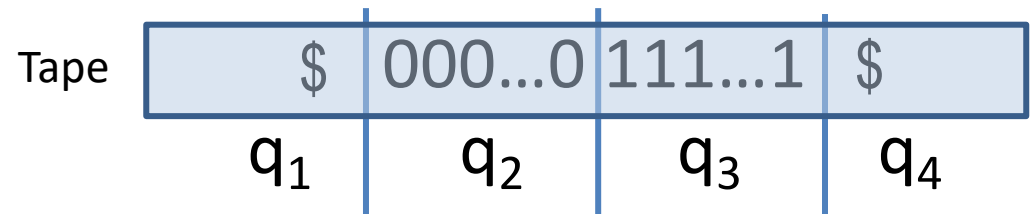
# PDA example



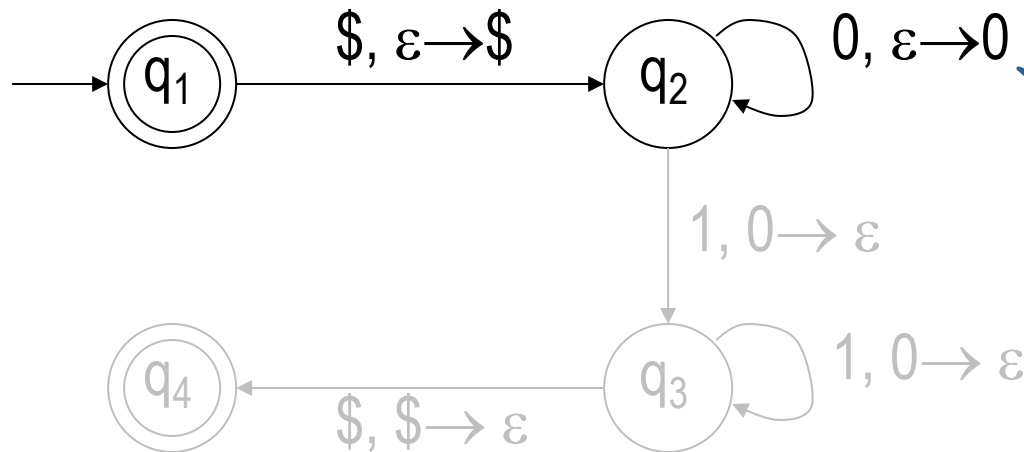
- $L = \{0^n 1^n \mid n \geq 0\}$
- $M_1 = (\{q_1, q_2, q_3, q_4\}, \{0, 1\}, \{0, \$\}, \delta, q_1, \{q_1, q_4\})$



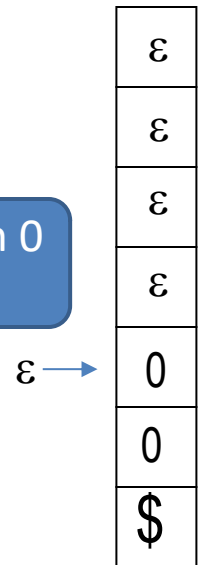
# PDA example



- $L = \{0^n 1^n \mid n \geq 0\}$
- $M_1 = (\{q_1, q_2, q_3, q_4\}, \{0, 1\}, \{0, \$\}, \delta, q_1, \{q_1, q_4\})$



For input 0, push 0 into the stack

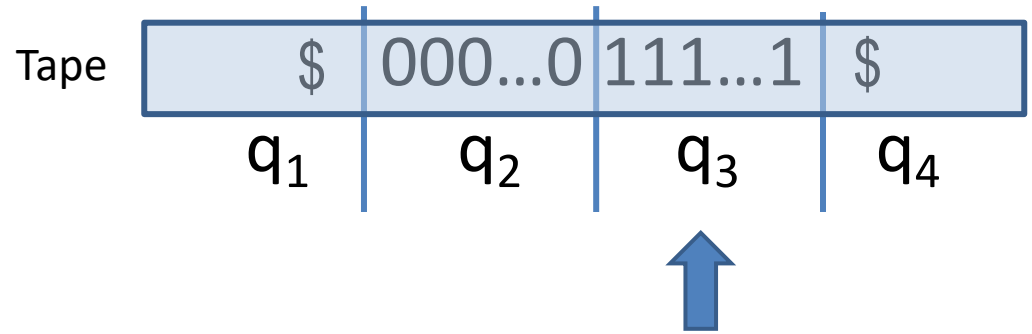


stack

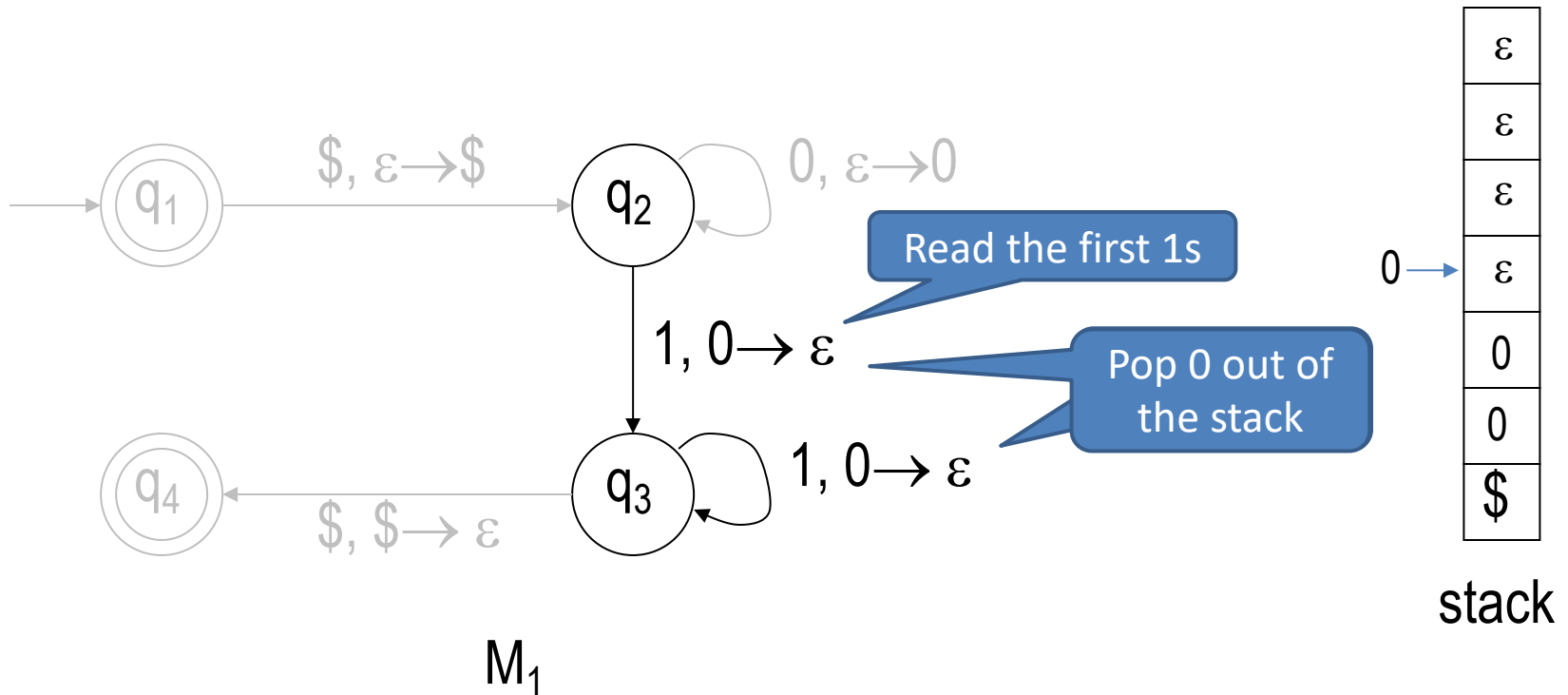
$M_1$



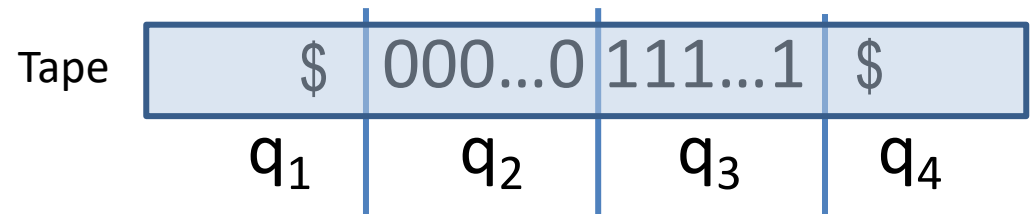
# PDA example



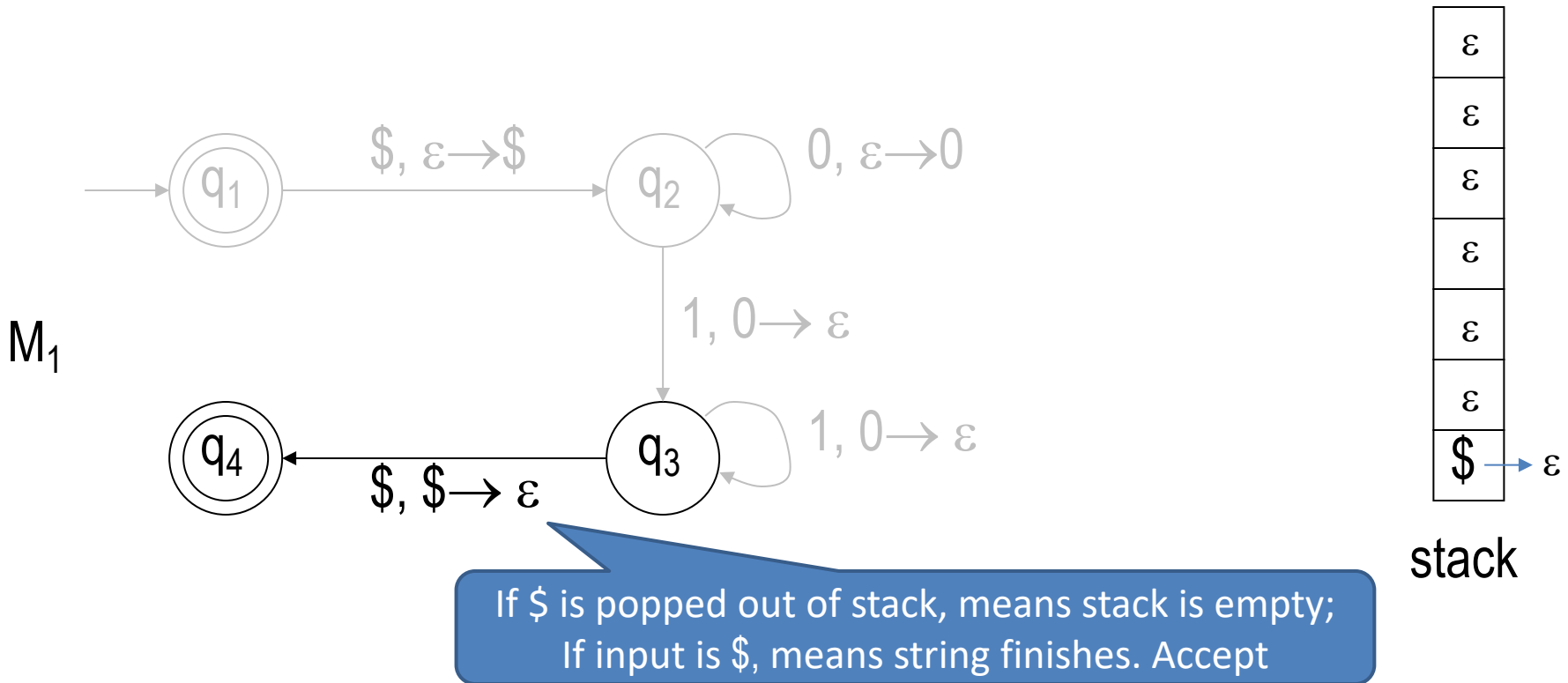
- $L = \{0^n 1^n \mid n \geq 0\}$
- $M_1 = (\{q_1, q_2, q_3, q_4\}, \{0, 1\}, \{0, \$\}, \delta, q_1, \{q_1, q_4\})$



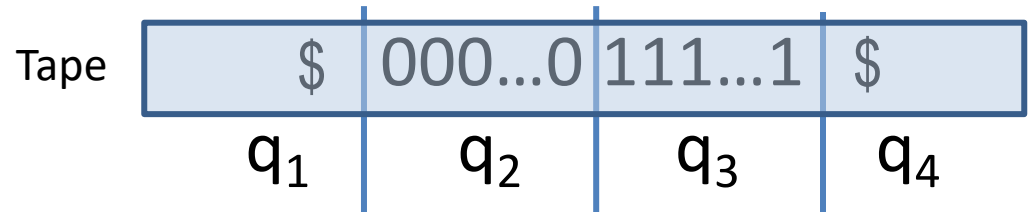
# PDA example



- $L = \{0^n 1^n \mid n \geq 0\}$
- $M_1 = (\{q_1, q_2, q_3, q_4\}, \{0, 1\}, \{0, \$\}, \delta, q_1, \{q_1, q_4\})$



# PDA example



- $L = \{0^n 1^n \mid n \geq 0\}$
- $M_1 = (\{q_1, q_2, q_3, q_4\}, \{0, 1\}, \{0, \$\}, \delta, q_1, \{q_1, q_4\})$

If input is 0,  
 $(q_2, \varepsilon)$  changes to  $(q_2, 0)$   
 $\varepsilon$  in stack change to 0 (PUSH 0)

$$\delta: Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \rightarrow P(Q \times \Gamma_\varepsilon)$$

$\delta$  table

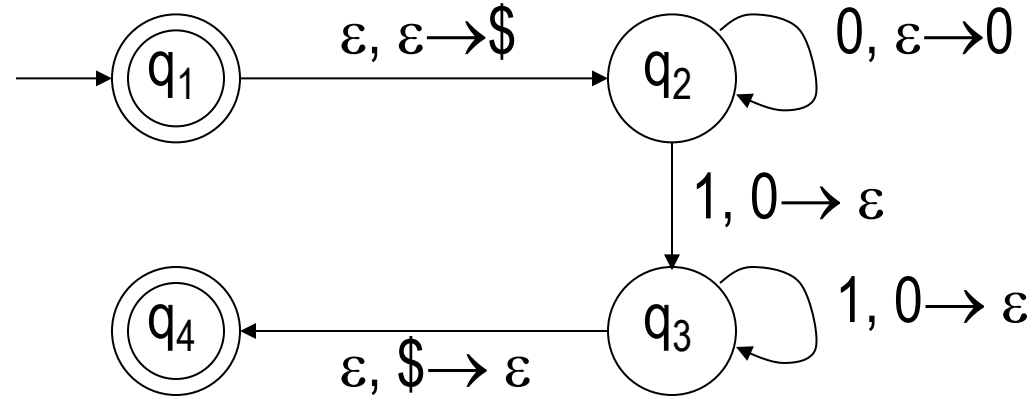
$\Sigma_\varepsilon$		Input			0			1			$\varepsilon$		
$\Gamma_\varepsilon$		stack			0	\$	$\varepsilon$	0	\$	$\varepsilon$	0	\$	$\varepsilon$
Q	state	$q_1$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\{(q_2, \$)\}$
		$q_2$	$\emptyset$	$\emptyset$	$\{(q_2, 0)\}$	$\{(q_3, \varepsilon)\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
		$q_3$	$\emptyset$	$\emptyset$	$\emptyset$	$\{(q_3, \varepsilon)\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{(q_4, \varepsilon)\}$	$\emptyset$	$\emptyset$	$\emptyset$
		$q_4$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$



# PDA example

$\epsilon, \epsilon \rightarrow \$$   
 $=$   
 $\$, \epsilon \rightarrow \$$

$\delta$  graph



II

$\delta$  table

Input		0			1			$\epsilon$		
stack		0	\$	$\epsilon$	0	\$	$\epsilon$	0	\$	$\epsilon$
state	q <sub>1</sub>	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\{(q_2, \$)\}$
	q <sub>2</sub>	$\emptyset$	$\emptyset$	$\{(q_2, 0)\}$	$\{(q_3, \epsilon)\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
	q <sub>3</sub>	$\emptyset$	$\emptyset$	$\emptyset$	$\{(q_3, \epsilon)\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{(q_4, \epsilon)\}$	$\emptyset$
	q <sub>4</sub>	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

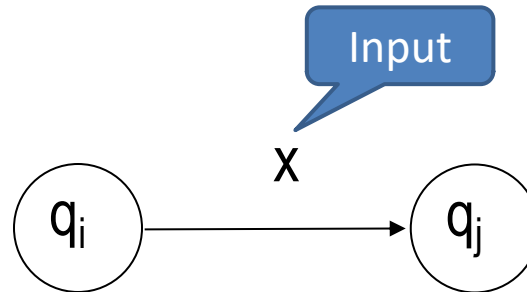




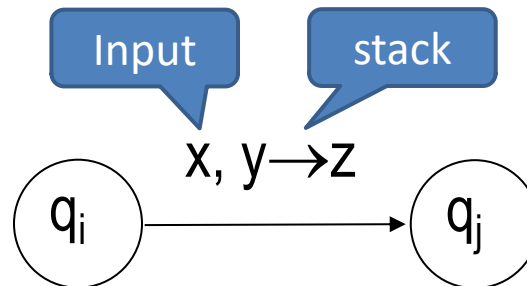
# Design PDA

---

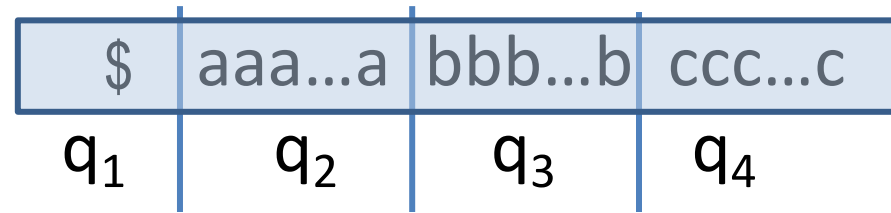
DFA/NFA:



PDA:



# Design PDA



- $L(M_2) = \{ a^n b^n c^m \mid m, n \geq 0 \}$

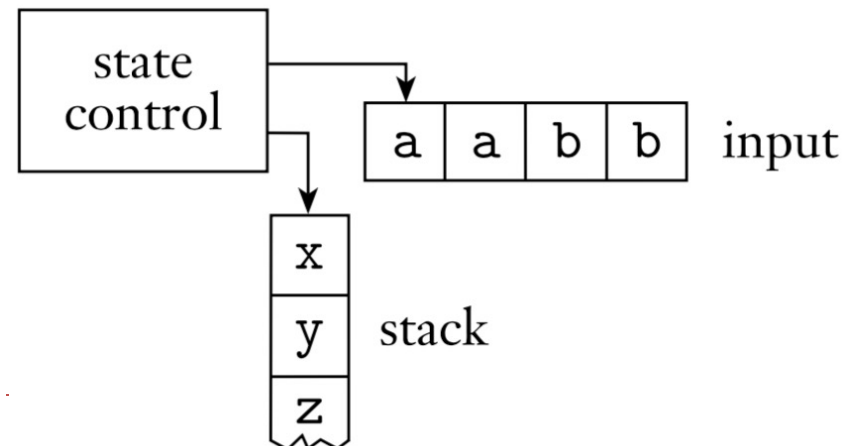
- ▶ Operation:

- For an input a, and push a into stack
- For an input b, pop one a from the stack

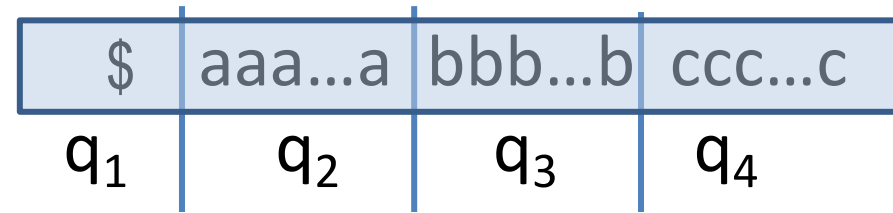
- ▶ Determine accept/reject

- If the stack is empty when finish reading b, then after reading all the cs, accept;
- Otherwise, reject;

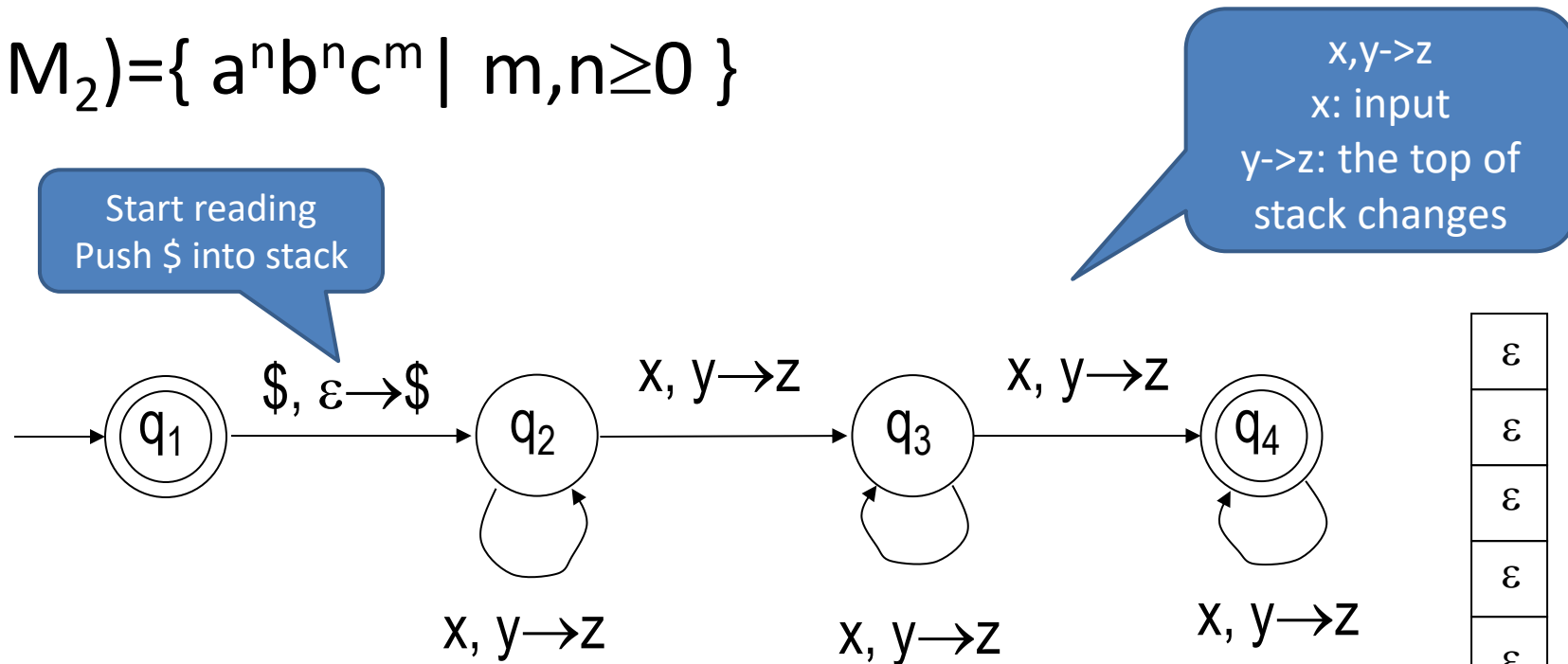
Can you explain the states?



# Design PDA



- $L(M_2) = \{ a^n b^n c^m \mid m, n \geq 0 \}$



- $L(M_2) = \{ a^n b^n c^m \mid m, n \geq 0 \}$

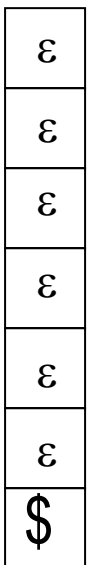
► Operation:

- For an input a, and push a into stack
- For an input b, pop one a from the stack

► Determine accept/reject

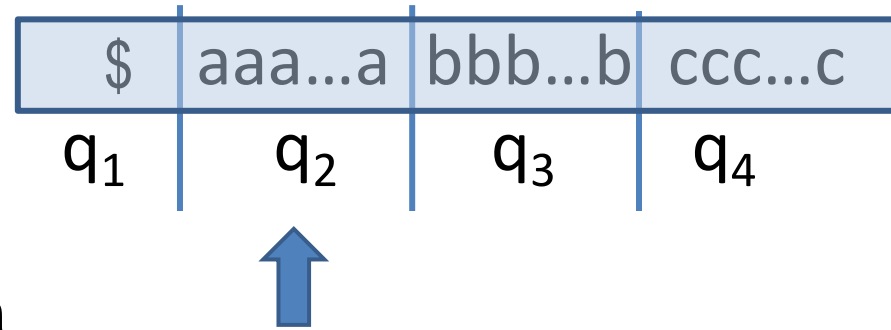
- If the stack is empty when finish reading b, then after reading all the cs, accept;
- Otherwise, reject;

Can you define the transitions  $x, y \rightarrow z$ ?

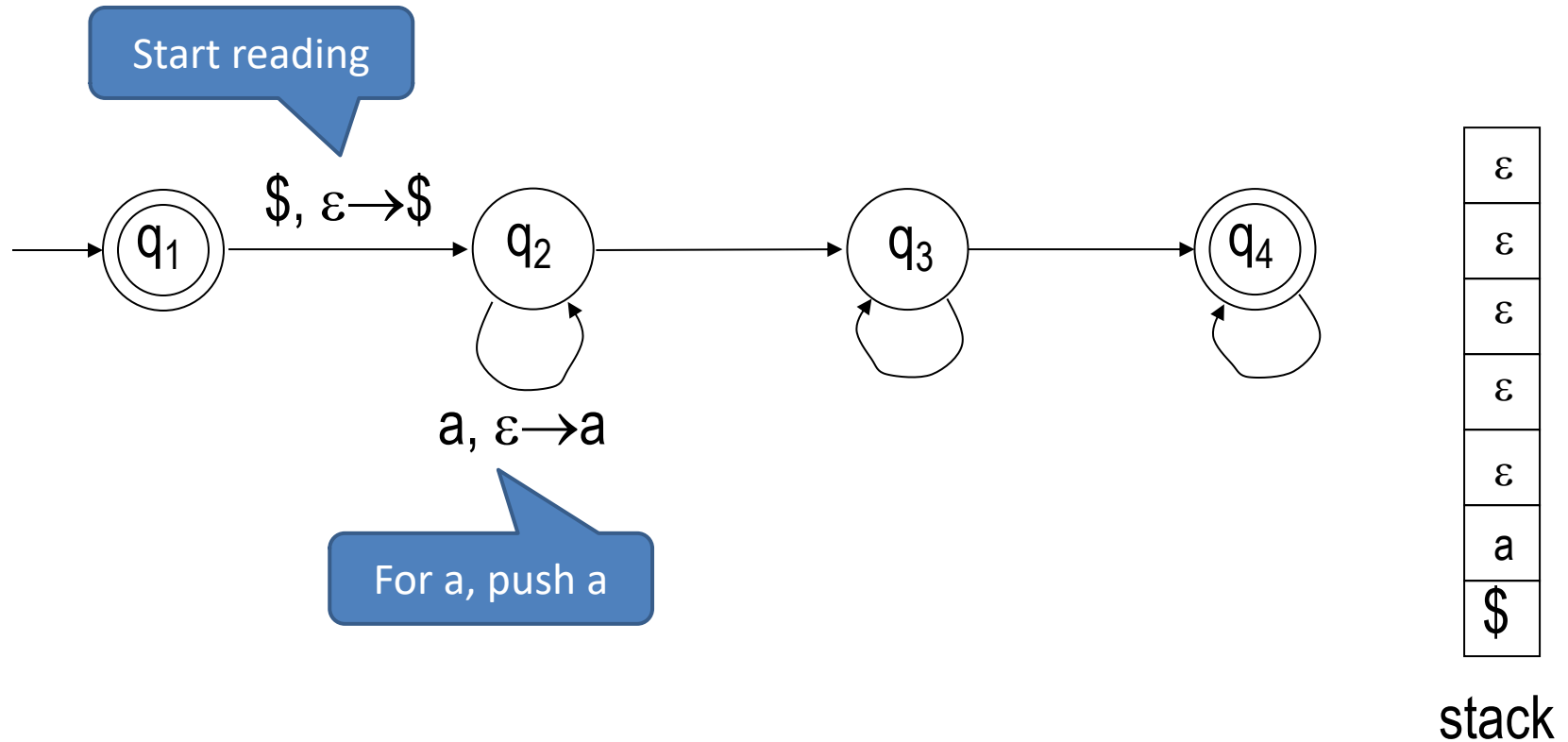


stack

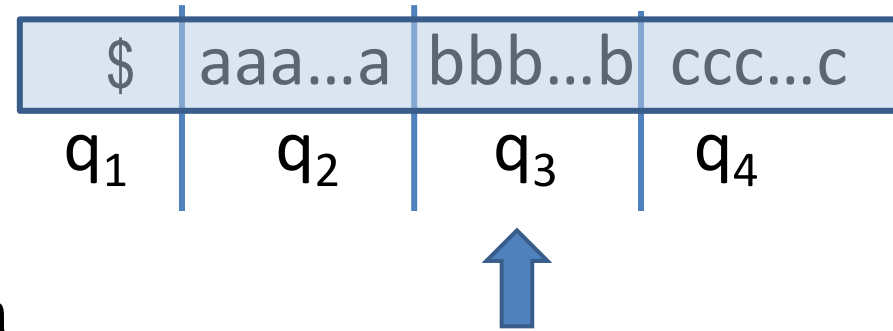
# Design PDA



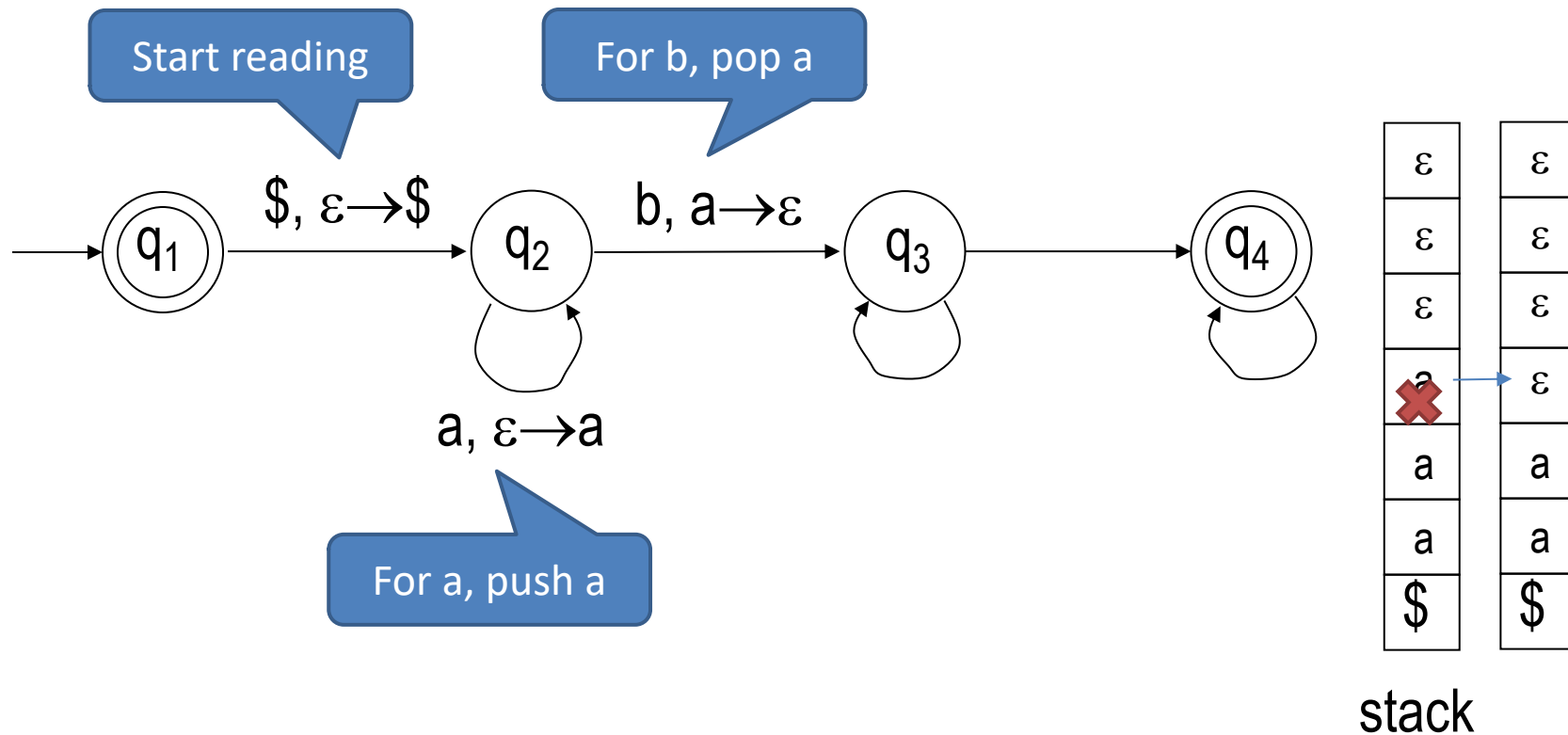
- $L(M_2) = \{ a^n b^n c^m \mid m, n \geq 0 \}$



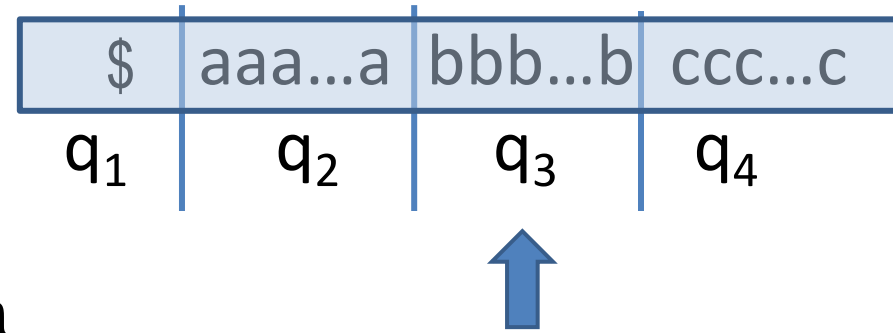
# Design PDA



- $L(M_2) = \{ a^n b^n c^m \mid m, n \geq 0 \}$



# Design PDA



- $L(M_2) = \{ a^n b^n c^m \mid m, n \geq 0 \}$

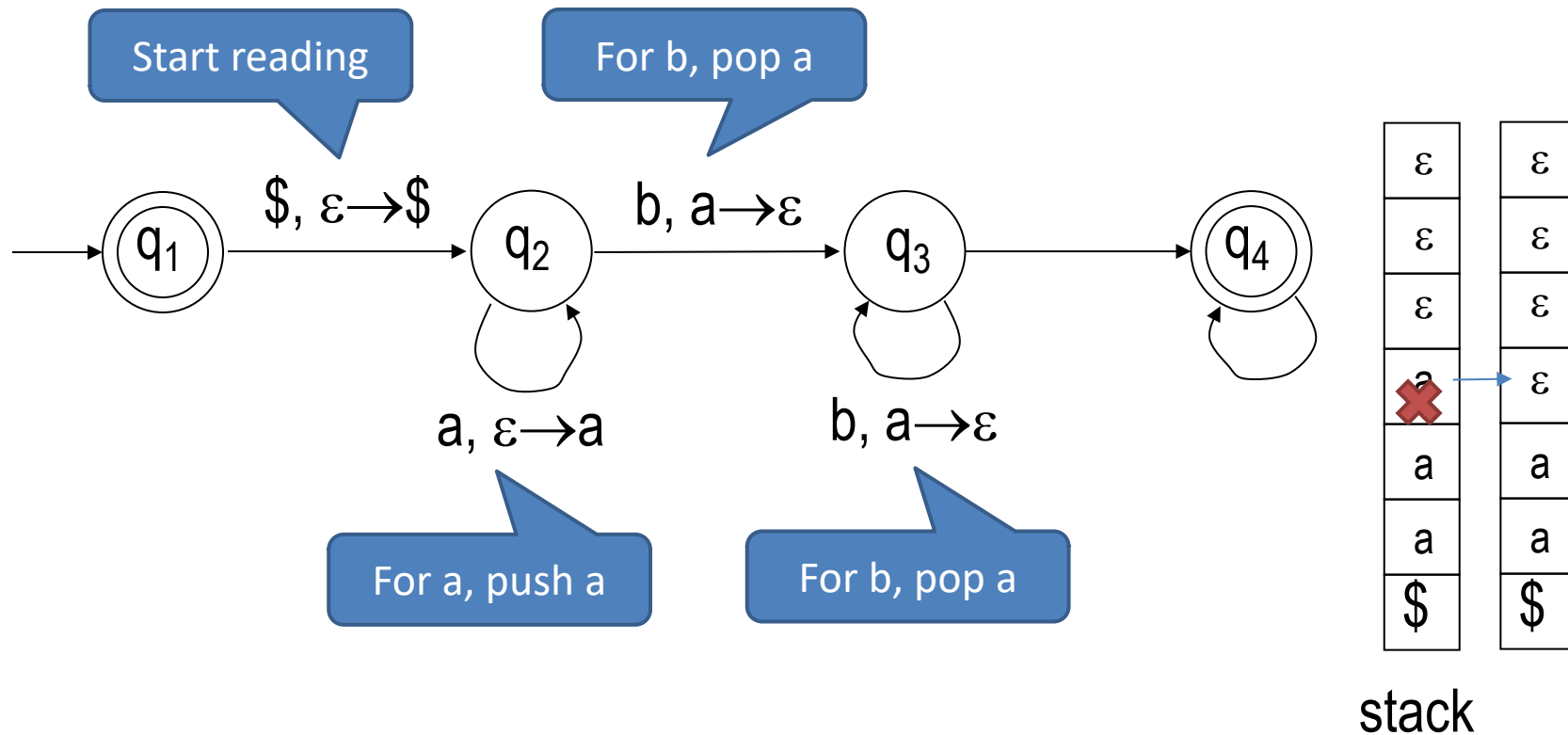


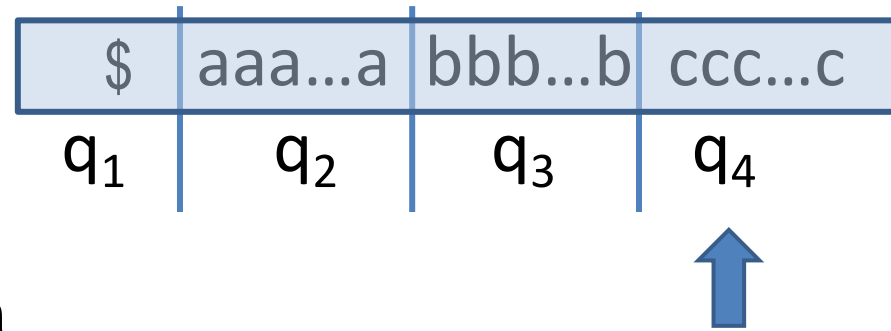
Diagram illustrating a tape configuration with four cells labeled  $q_1$ ,  $q_2$ ,  $q_3$ , and  $q_4$ . The contents of the cells are:

- $q_1$ :  $\$$
- $q_2$ :  $aaa...a$
- $q_3$ :  $bbb...b$
- $q_4$ :  $ccc...c$

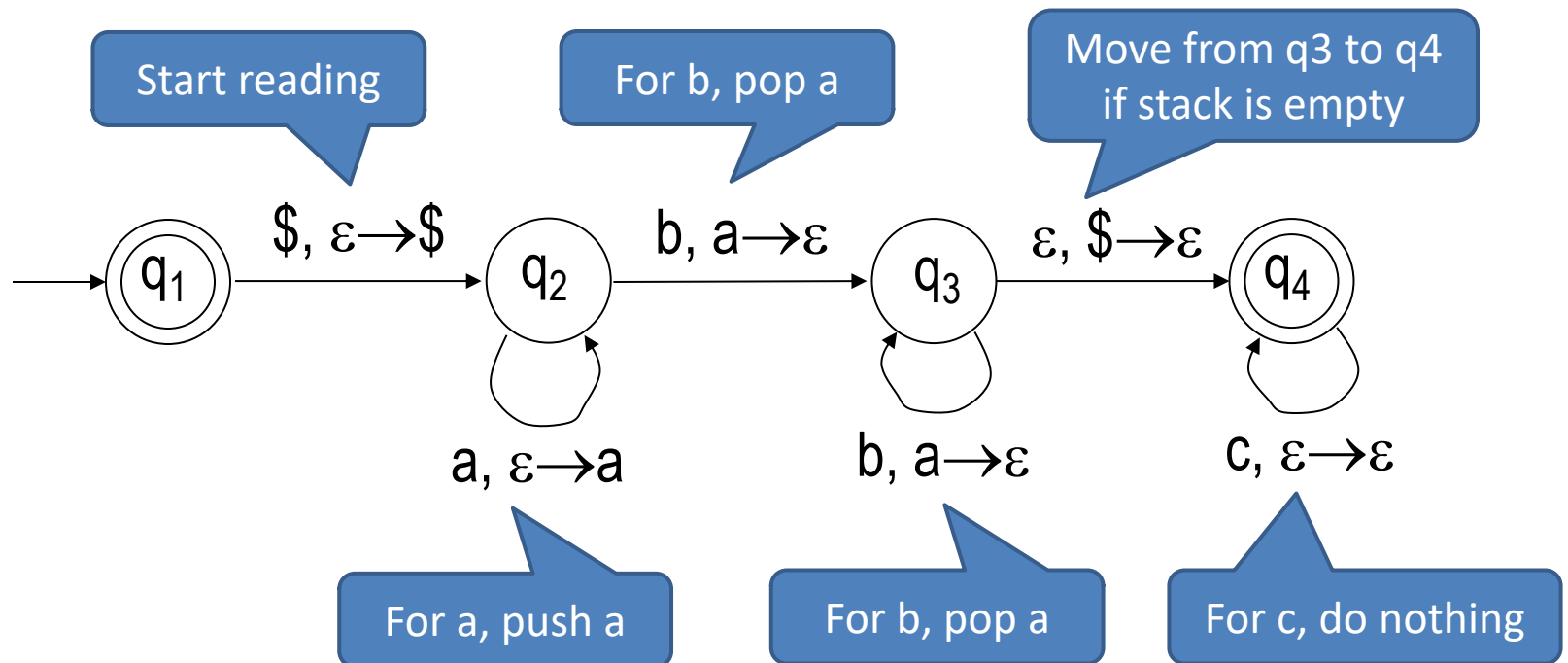
A blue arrow points to the cell  $q_4$ .

- 
- Start reading
- For a, push a
- For b, pop a
- Move from q3 to q4 if stack is empty
- stack

# Design PDA



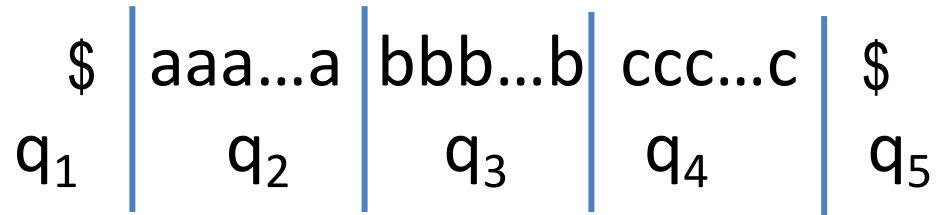
- $L(M_2) = \{ a^n b^n c^m \mid m, n \geq 0 \}$





# Design PDA

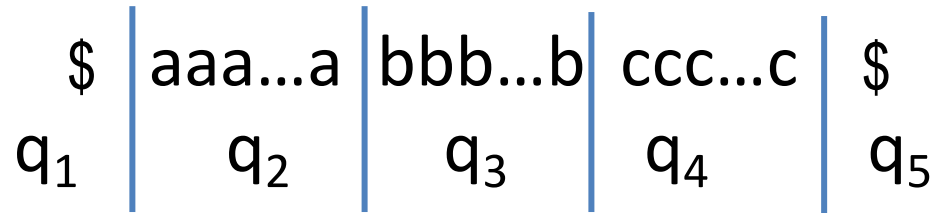
---



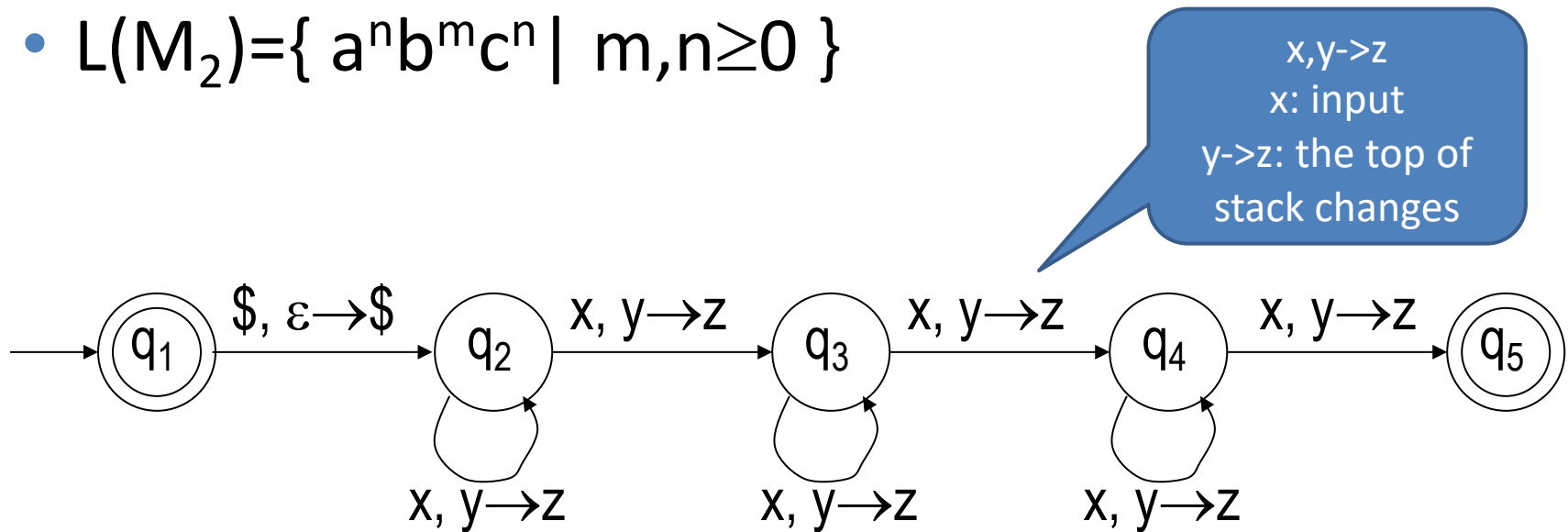
- $L(M_2) = \{ a^n b^m c^n \mid m, n \geq 0 \}$ 
  - ▶ Operation:
    - For an input a, and push a into stack
    - After reading some bs, every time, for an input c, pop one a from the stack
  - ▶ Determine accept/reject
    - If the stack is empty when input is done, accept;
    - Otherwise, reject.



# Design PDA



- $L(M_2) = \{ a^n b^m c^n \mid m, n \geq 0 \}$



- $L(M_2) = \{ a^n b^m c^n \mid m, n \geq 0 \}$

► Operation:

- For an input  $a$ , and push  $a$  into stack
- After reading some  $b$ s, every time, for an input  $c$ , pop one  $a$  from the stack

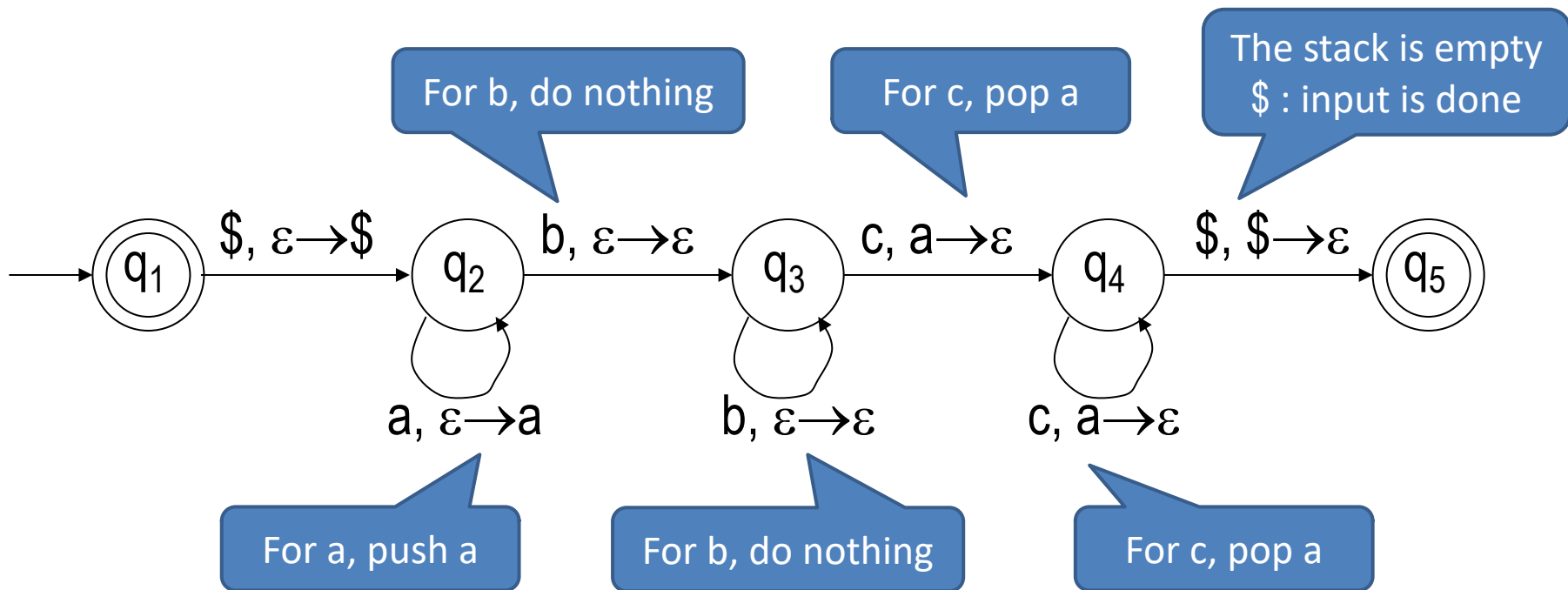
► Determine accept/reject

- If the stack is empty when input is done, accept;
- Otherwise, reject.

# Design PDA

\$	aaa...a	bbb...b	ccc...c	\$
$q_1$	$q_2$	$q_3$	$q_4$	$q_5$

- $L(M_2) = \{ a^n b^m c^n \mid m, n > 0 \}$



# Design PDA

---

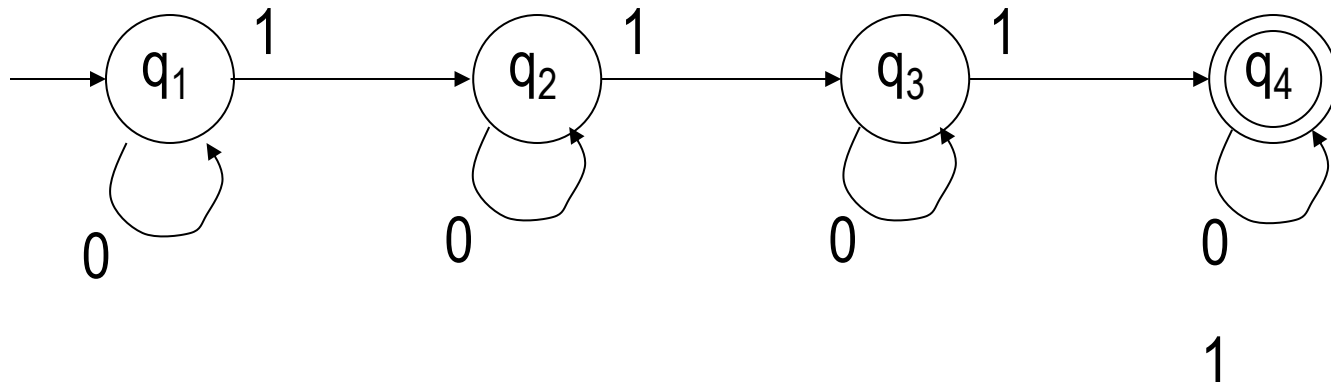
- $L(M_3) = \{ w \mid w \text{ contains at least three 1s} \}$ , input =  $\{0, 1\}$ 
  - ▶ Input : 001101
    - Output : Accepted
  - ▶ Input : 100010
    - Output : Not Accepted
  - ▶ Regular language
    - Does not need the stack,  $\varepsilon \rightarrow \varepsilon$



# Design PDA

- $L(M_3) = \{ w \mid w \text{ contains at least three 1s} \}$ , input =  $\{0, 1\}$

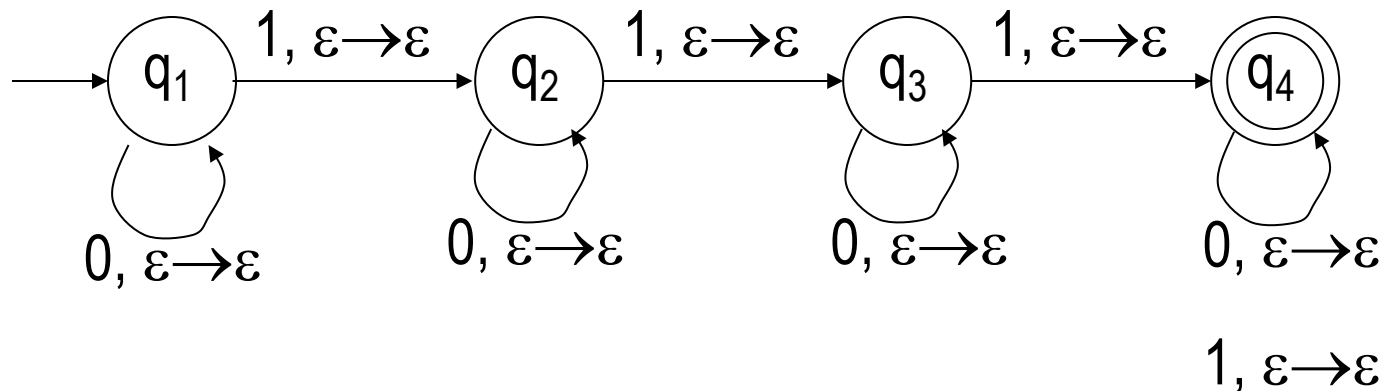
What are the states?



# Design PDA

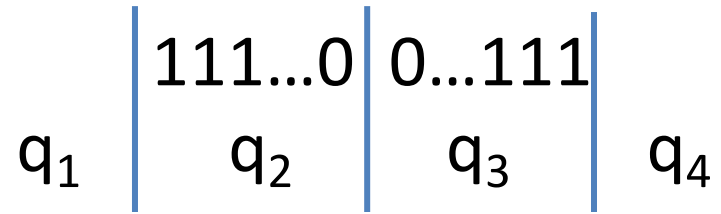
---

- $L(M_3) = \{ w \mid w \text{ contains at least three 1s} \}$ , input =  $\{0, 1\}$



# Design PDA: $\{ ww^R \}$

---

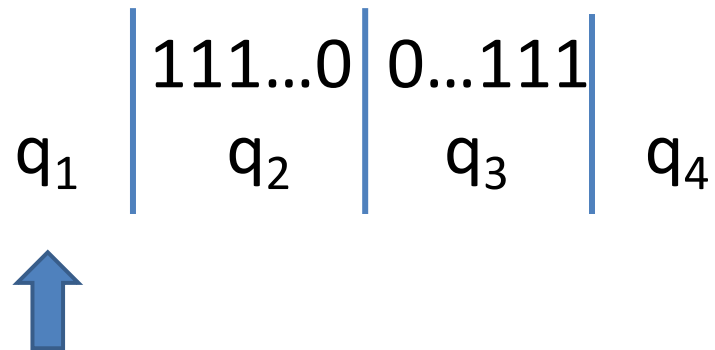


- Palindromes:
- Examples:
  - NOON
  - 123321
  - abba

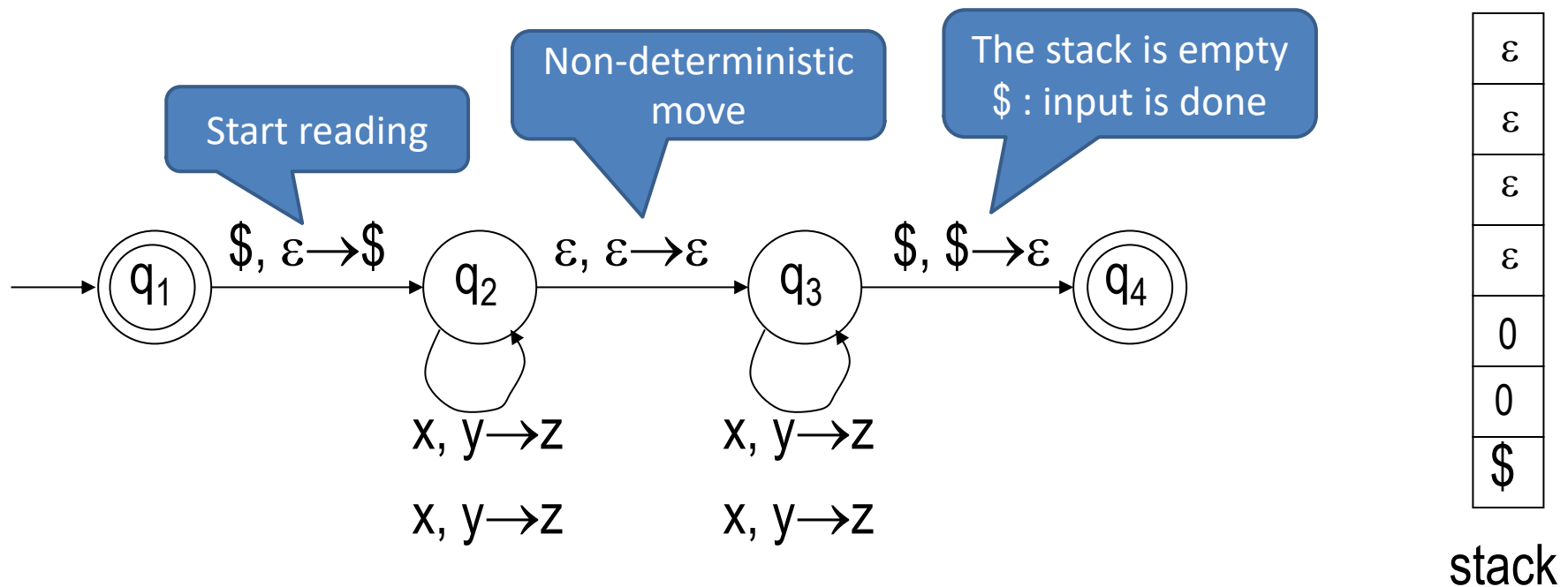
What are the states?



# Design PDA: $\{ ww^R \}$



- Palindromes:

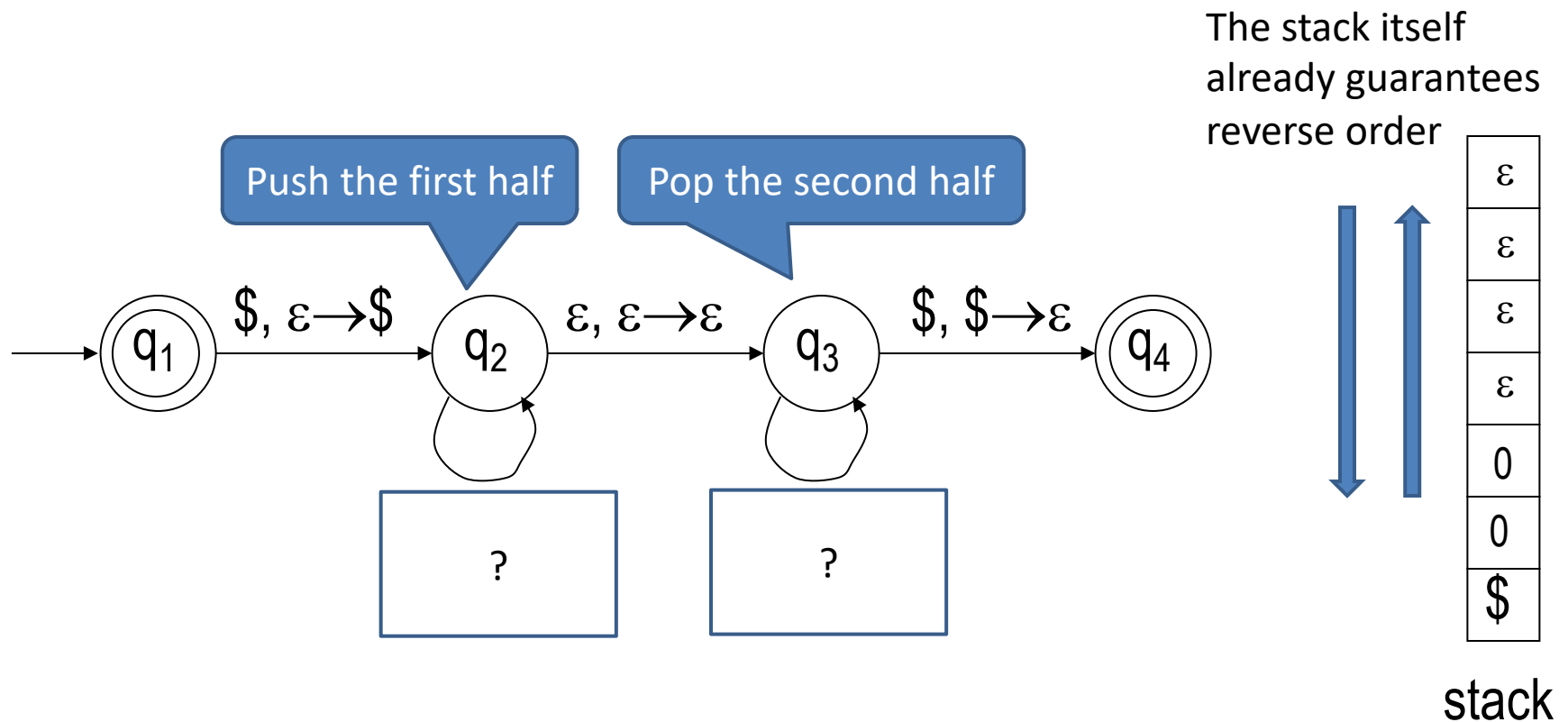




# Design PDA: $\{ ww^R \}$

$q_1$  |  $abc\dots z$  |  $z\dots cba$  |  $q_4$   
 $q_2$  |  $q_3$

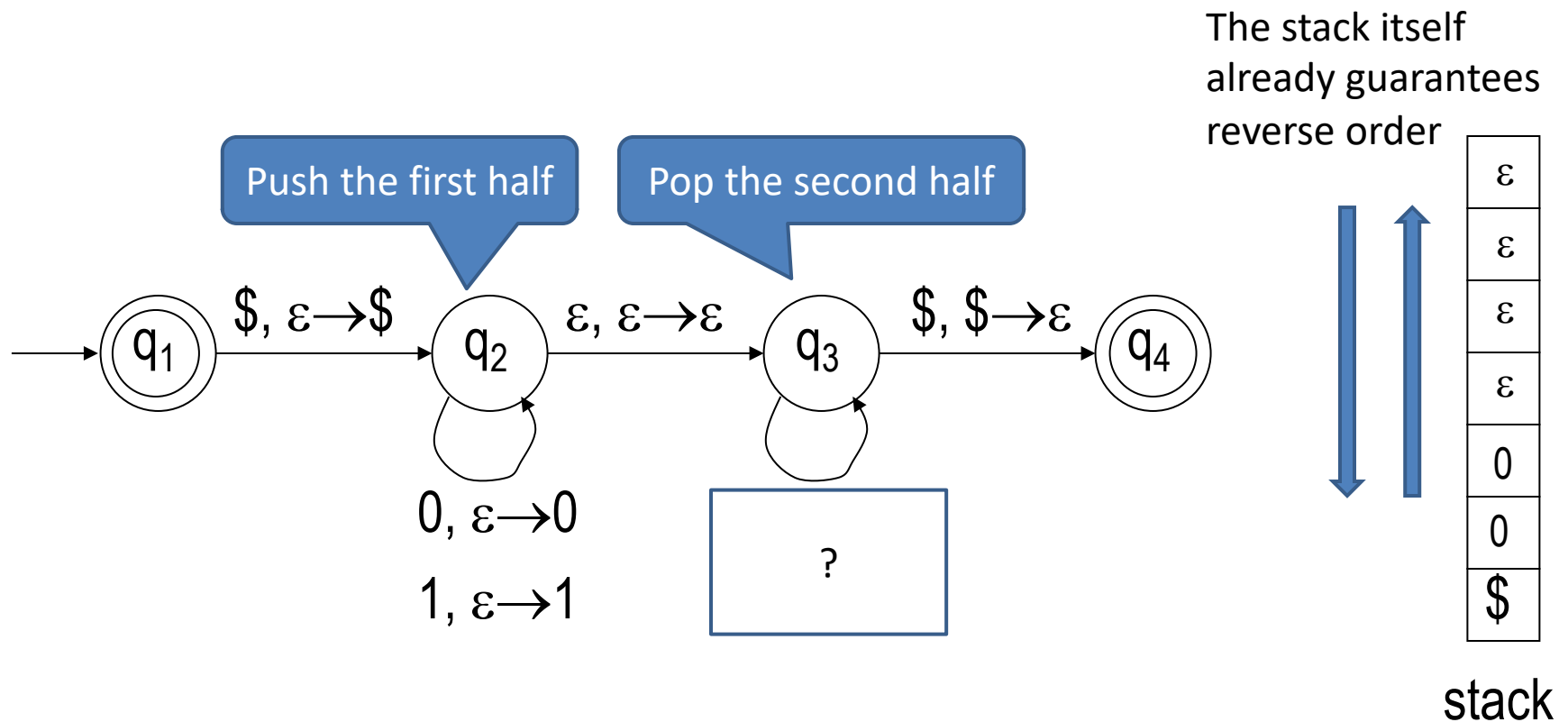
- Palindromes:



# Design PDA: $\{ ww^R \}$

$q_1$  |  $abc...z$  |  $z...cba$  |  $q_4$   
 $q_2$        $q_3$

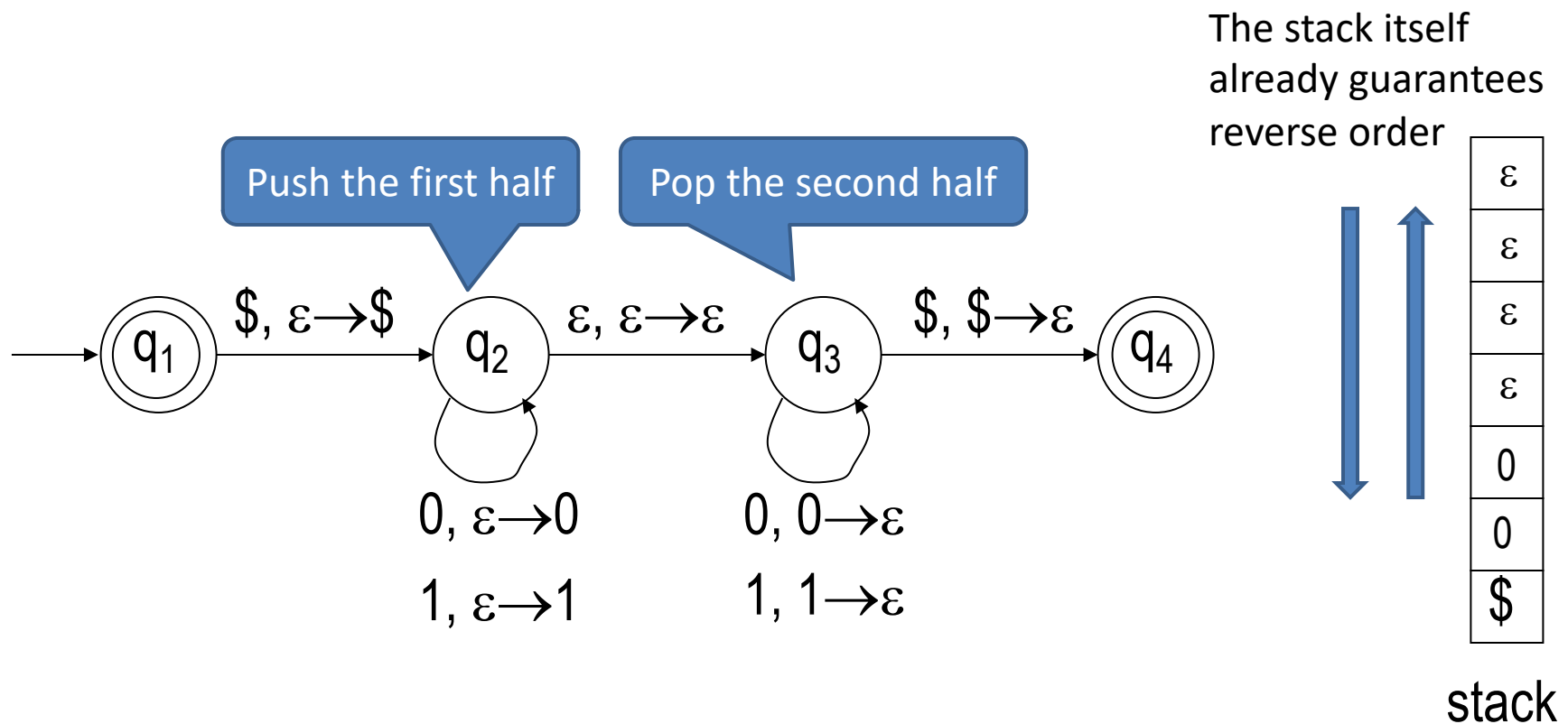
- Palindromes:



# Design PDA: $\{ ww^R \}$

$q_1$  |  $abc\dots z$  |  $z\dots cba$  |  $q_4$   
 $q_2$  |  $q_3$

- Palindromes:



# Conclusion

---

- What is pushdown automata (PDA)?
- How to use PDA to recognize some CFL? Informal description
- Definition of PDA  $M=(Q,\Sigma,\Gamma,\delta,q_0,F)$
- PDA examples,  $\delta: x, y \rightarrow z$

PUSH  $z: x, \varepsilon \rightarrow z$

POP  $z. : x, z \rightarrow \varepsilon$

