Computing Theory

COMP 147

Chapter 2: Context-Free Languages
Section 2.2: Pushdown Automata

Pushdown Automata

Definition
Moves of the PDA
Languages of the PDA

Pushdown Automata

- The PDA is an automaton equivalent to the CFG in language-defining power.

- Only the nondeterministic PDA defines all the CFL's.

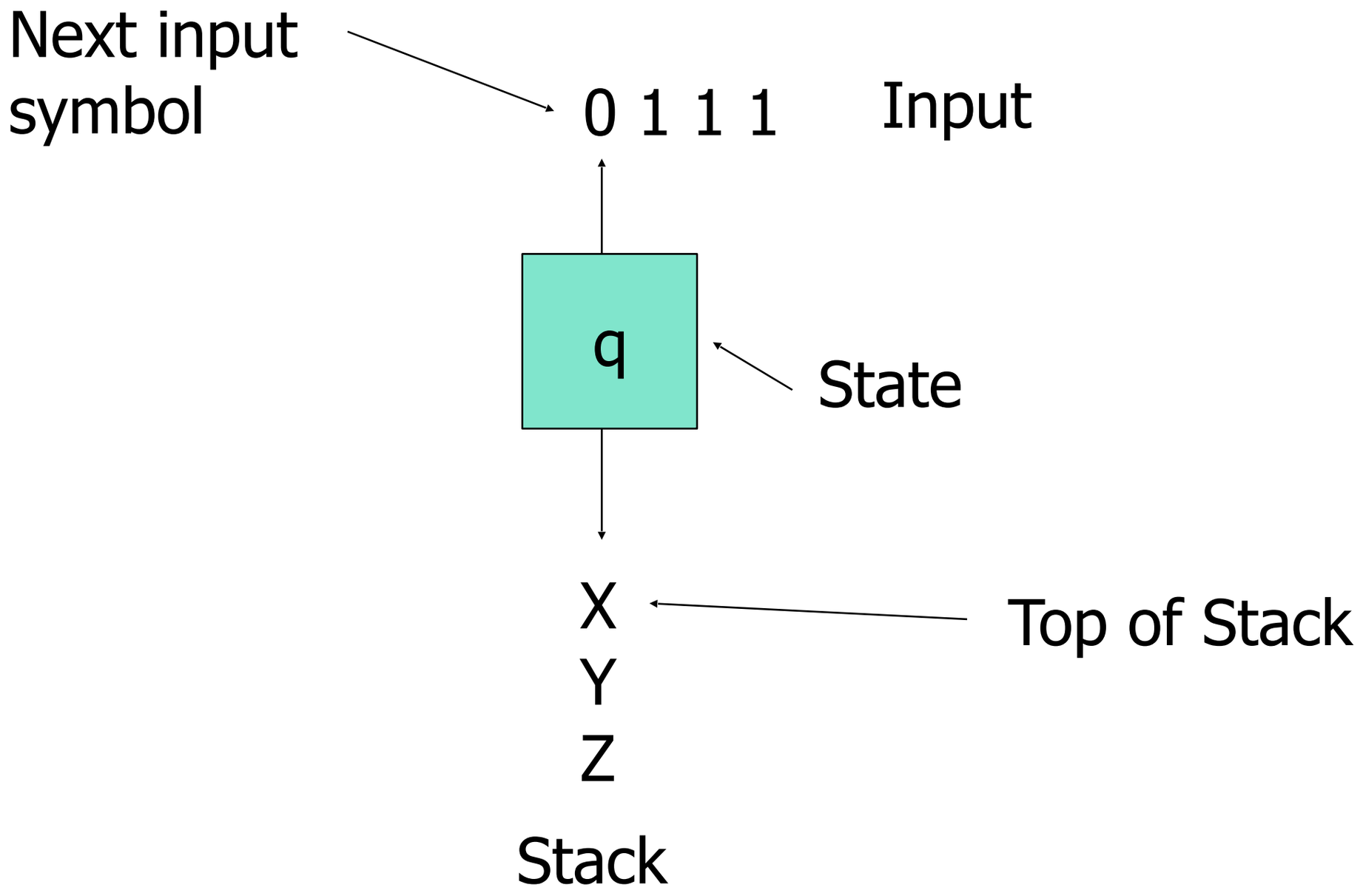
- But the deterministic version models parsers.
  - Most programming languages have deterministic PDA's.

Intuition: PDA

- Think of an NFA with the additional power that it can manipulate a stack.

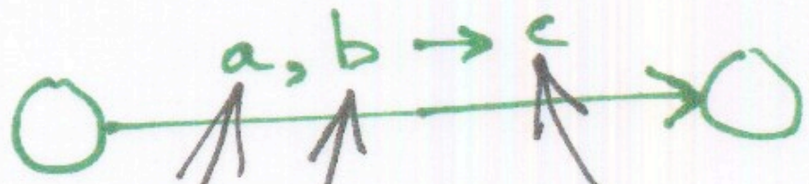
- Its moves are determined by:
  - The current state (of its “NFA”),
  - The current input symbol (or **ε**), and
  - The current symbol on top of its stack.

Picture of a PDA



Finite State Machine

Pushdown Automaton



Input symbol

Symbol on top of the stack.

This symbol is popped.

This symbol is pushed onto the stack.

May "ϵ" be

"ϵ" means the stack is neither read nor popped

"ϵ" means nothing is pushed.

PDA Formalism

- A PDA is described by:
  1. A finite set of states (Q).
  2. An input alphabet (Σ).
  3. A stack alphabet (Γ).
  4. A transition function (δ).
  5. A start state (q_0).
  6. A set of final states ($F \subseteq Q$).

The Transition Function

- Takes three arguments:
  - A state, in Q.
  - An input, which is either a symbol in **Σ** or **ε**.
  - A stack symbol in **Γ**.

- **δ**(q, a, Z) is a set of zero or more actions of the form (p, α).
  - p is a state; α is a string of stack symbols.

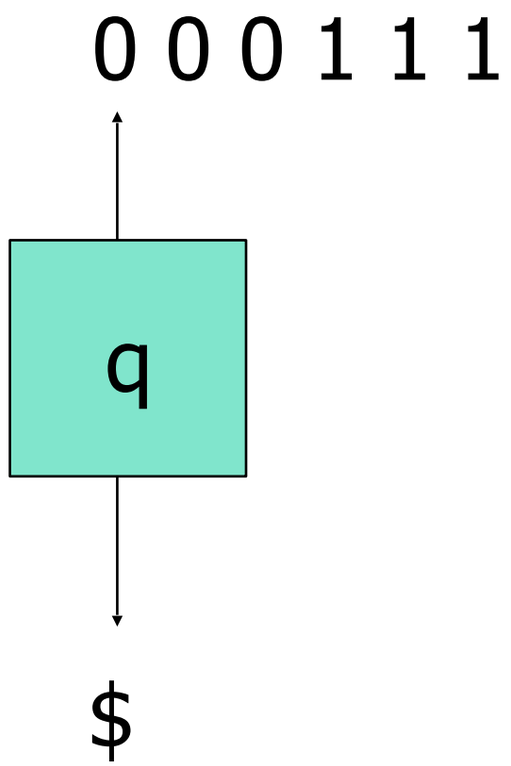
Example: PDA

- Design a PDA to accept $\{0^n1^n \mid n \geq 1\}$.
- The states:
  - q = start state. We are in state q if we have seen only 0's so far.
  - p = we've seen at least one 1 and may now proceed only if the inputs are 1's.
  - f = final state; accept.

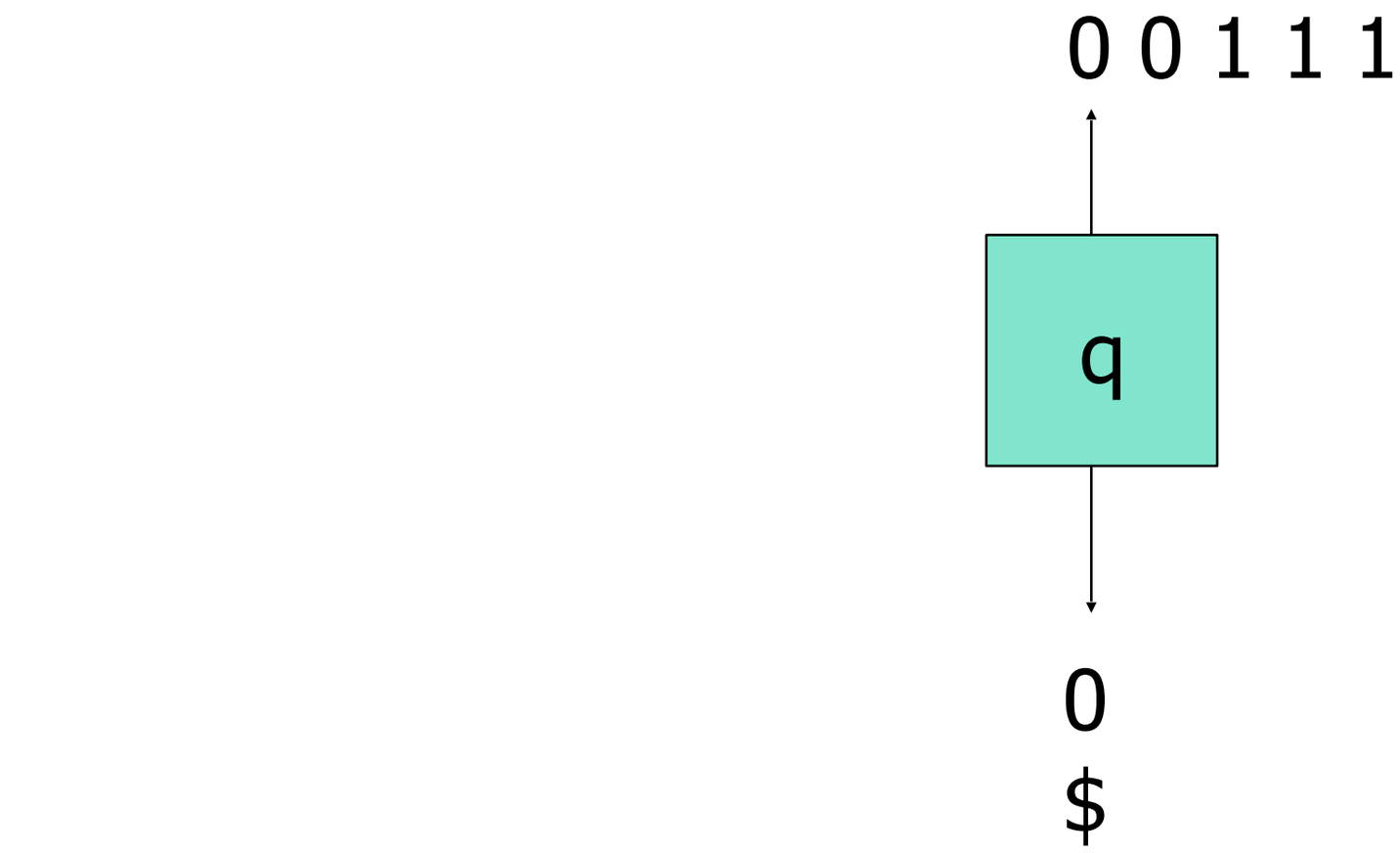
Example: PDA – (2)

- The stack symbols:
  - $ = start symbol. Also marks the bottom of the stack, so we know when we have counted the same number of 1's as 0's.
  - 0 = used to count the number of 0's seen on the input.

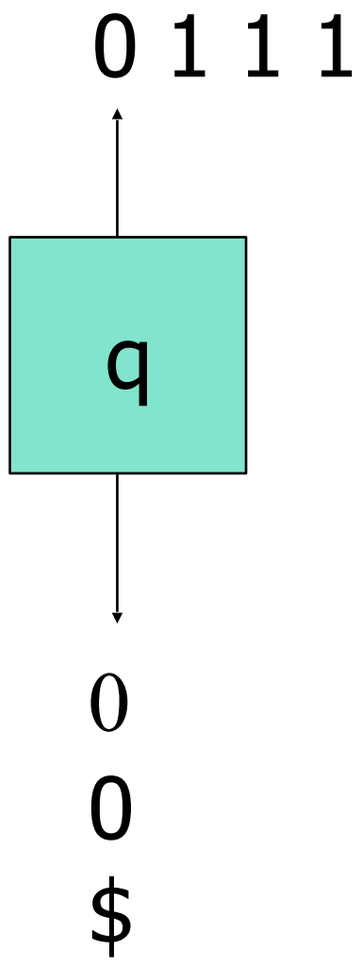
Actions of the Example PDA



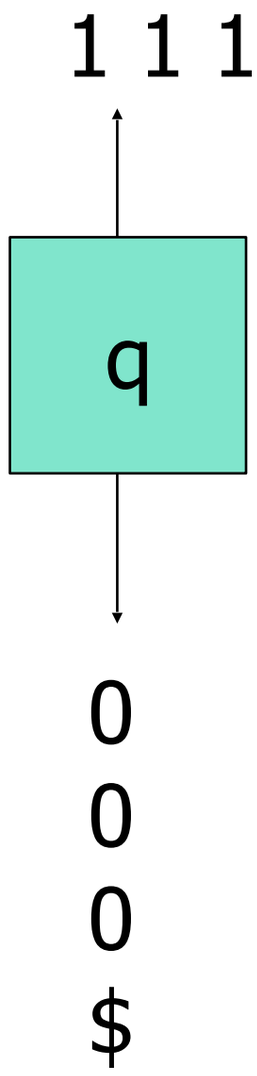
Actions of the Example PDA



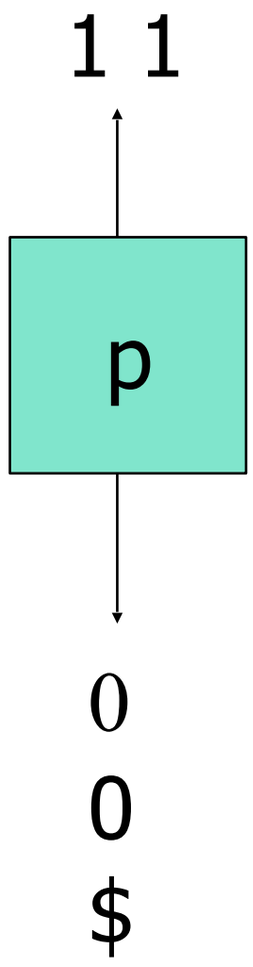
Actions of the Example PDA



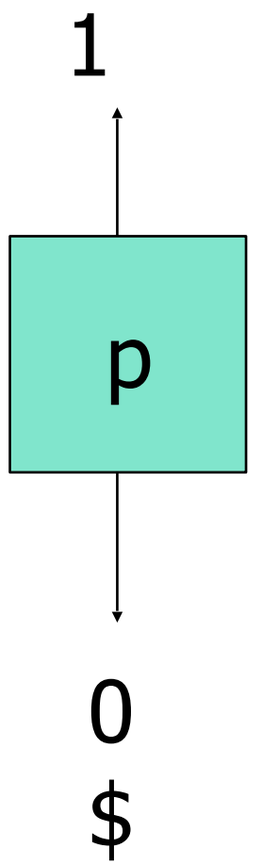
Actions of the Example PDA



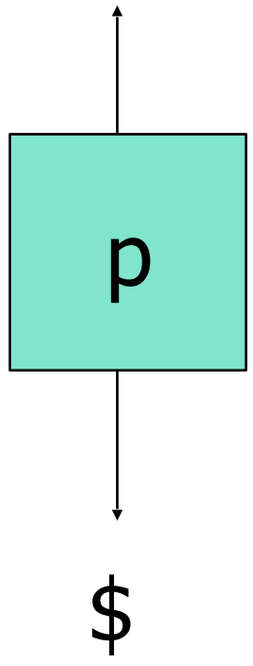
Actions of the Example PDA



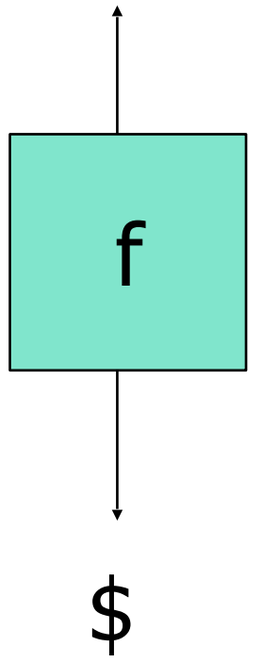
Actions of the Example PDA



Actions of the Example PDA



Actions of the Example PDA

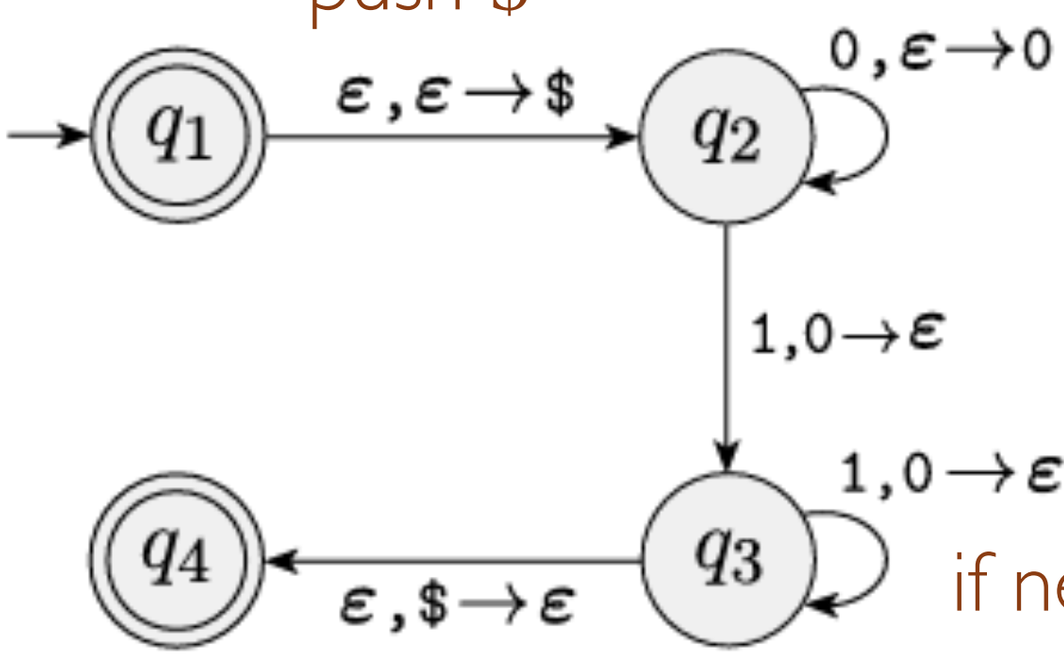


PDA Example

$$L = \{ 0^n 1^n \mid n \geq 0 \}$$

push $

if next input is 0, push 0



if top of stack is $,
pop $

if next input is 1,
and top of stack is 0,
pop 0

PDA: Transition Notation

- ε, ε → ε
  - transition immediately, do not consume input, do not modify stack
- x, ε → ε
  - transition if next input is x,
    consume x from input,
    do not modify stack
- ε, y → ε
  - transition if top of stack is y
    do not consume input
    pop y from stack
- x, y → ε
  - transition if next input is x and top of stack is y
    consume x from input
    pop y from stack

PDA: Transition Notation

- ε, ε → z
  - transition immediately
    do not consume input
    push z on stack
- x, ε → z
  - transition if next input is x
    consume x from input
    push z on stack
- ε, y → z
  - transition if top of stack is y
    do not consume input
    pop y from stack and push z on stack
- x, y → z
  - transition if next input is x and top of stack is y
    consume x from input
    pop y from stack and push z on stack

PDA: Formal Definition

DEFINITION 2.13

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

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

$$\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\} \text{ and } \Gamma_\varepsilon = \Gamma \cup \{\varepsilon\}$$

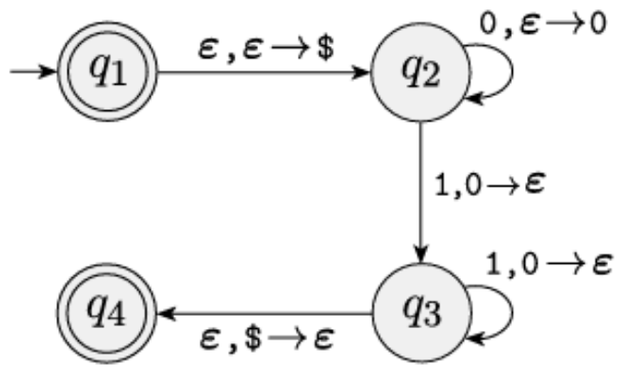
PDA Example: Formal Definition

$M = \{ Q, \Sigma, \Gamma, \delta, q_1, F \}$ recognizes $L = \{ 0^n 1^n \mid n \geq 0 \}$.

$Q = \{q_1, q_2, q_3, q_4\}$,

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

$\Gamma = \{0, \$\}$,



$F = \{q_1, q_4\}$, and

δ is given by the following table, wherein blank entries signify $\emptyset$.

Input:	0			1			ε		
Stack:	0	$	ε	0	$	ε	0	$	ε
q_1									$\{(q_2, \$)\}$
q_2			$\{(q_2, 0)\}$	$\{(q_3, \varepsilon)\}$					
q_3				$\{(q_3, \varepsilon)\}$				$\{(q_4, \varepsilon)\}$	
q_4									

PDA: Nondeterminism

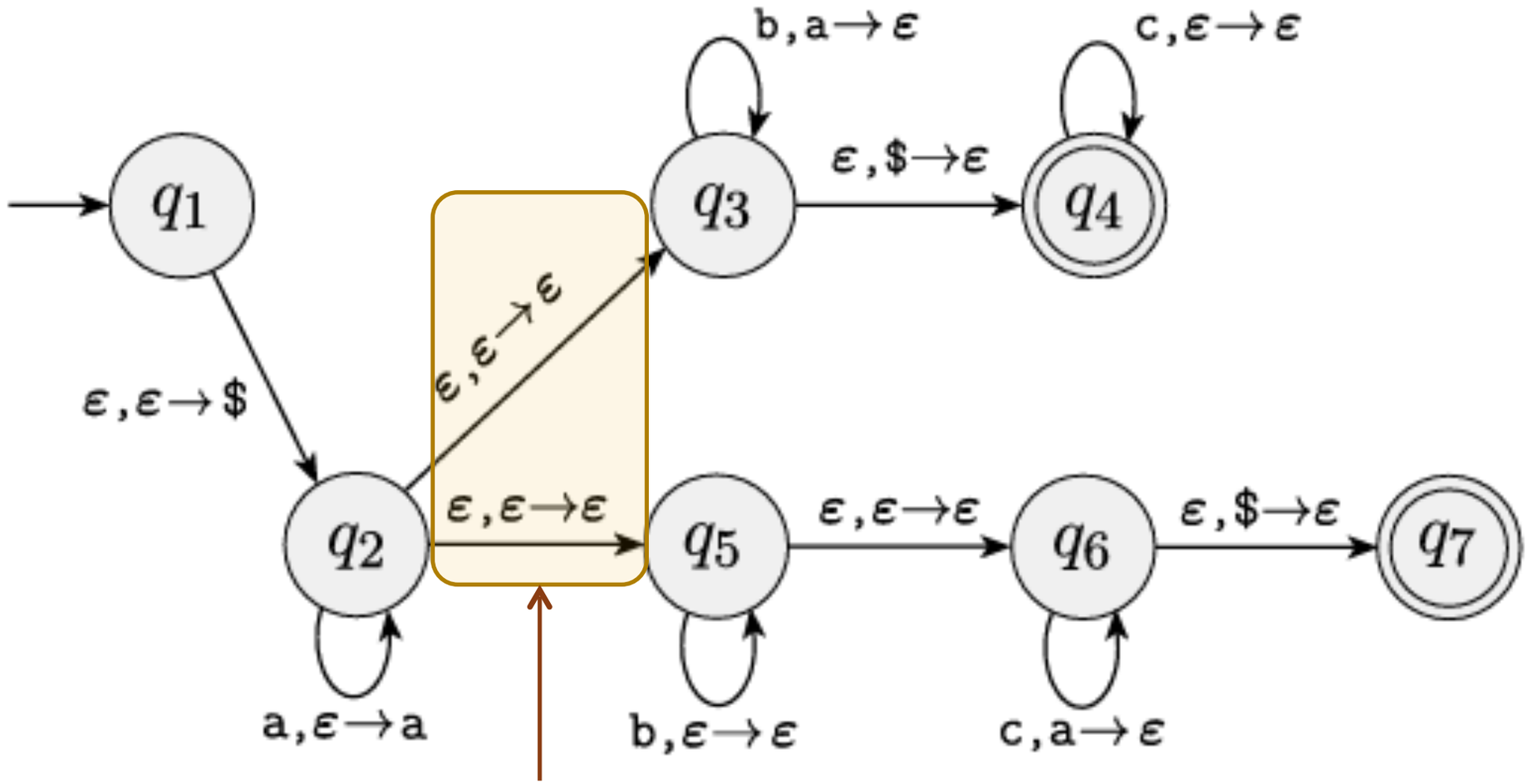
$L = \{ a^i b^j c^k \mid i,j,k \geq 0$ and $i = j$ or $i = k \}$

a's and # b's is the same, or
a's and # c's is the same

- Read a's and push on stack (as in prev. example)
- Pop the a's to count b's or the c's?

PDA: Nondeterminism

$L = \{ a^i b^j c^k \mid i,j,k \geq 0 \text{ and } i = j \text{ or } i = k \}$



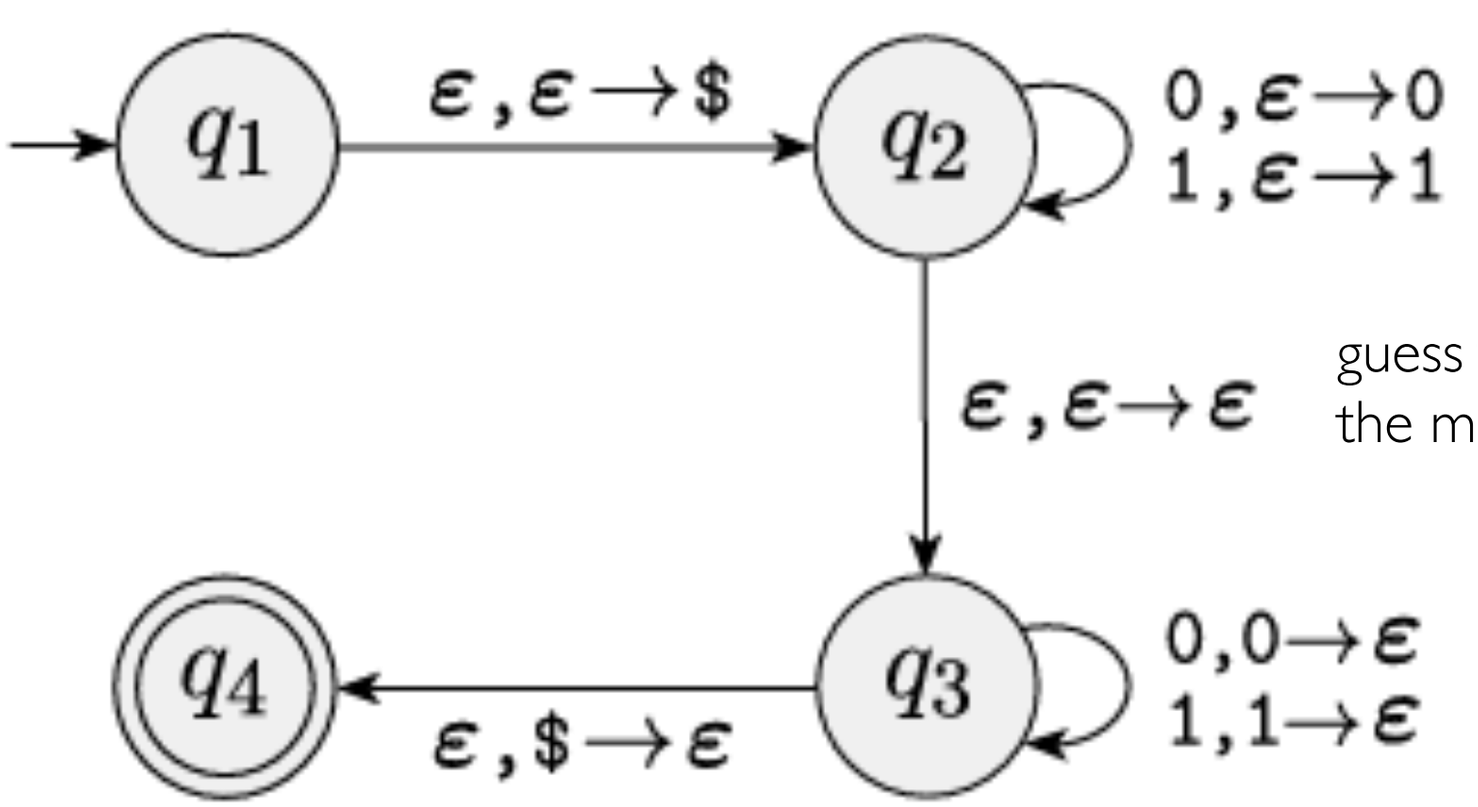
We can view this as a nondeterministic choice.

(a) Take both paths in parallel, or

(b) Guess (correctly) which one to take.

PDA Example: String Reversal

$L = \{ ww^R \mid w \in \{ 0, 1 \}^* \}$ (w^R means reverse 0f w)



push symbols from w

guess that we've reached the middle

pop symbols from w and compare to symbols from w^R