Theory of Computation
Chapter 2

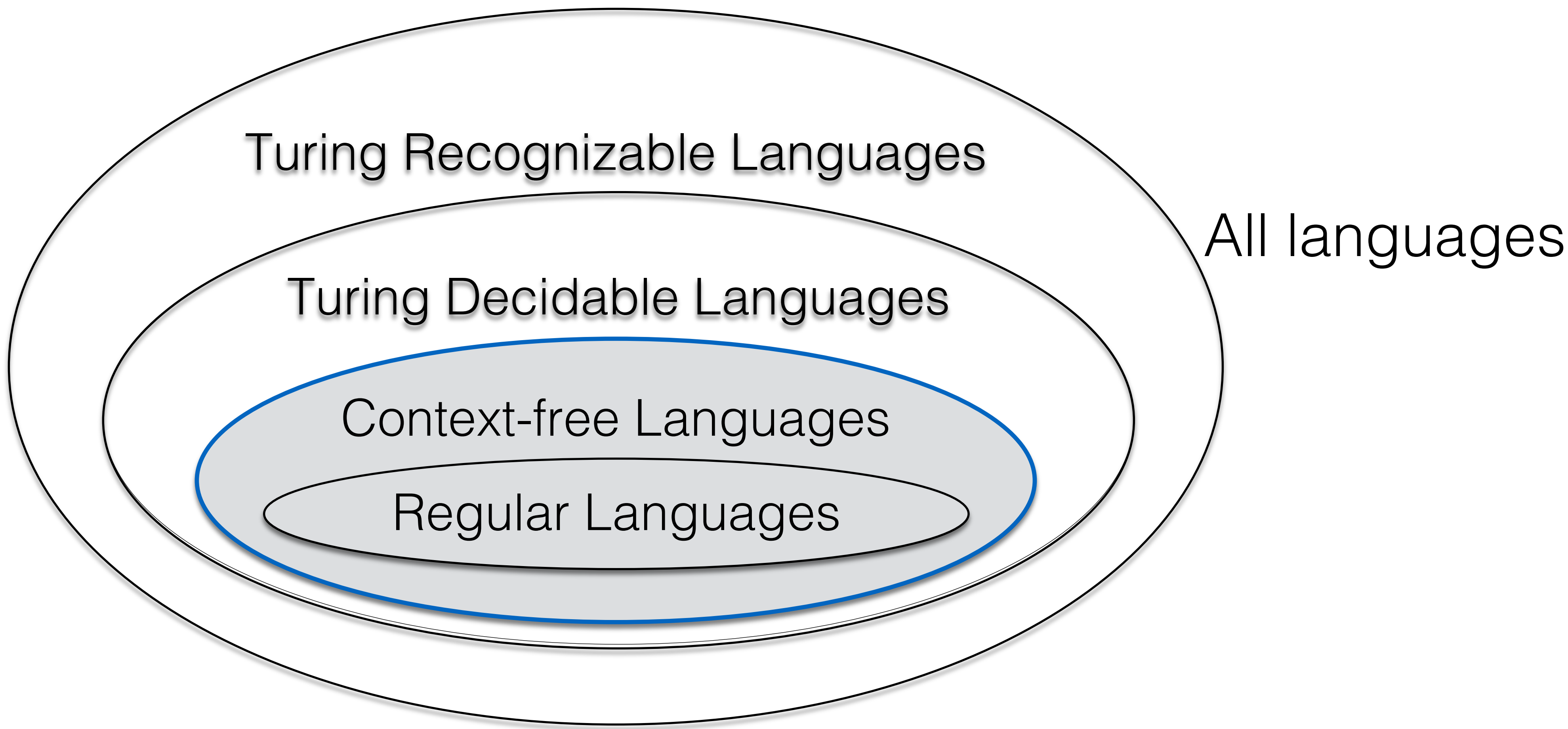
Pushdown Automata (PDA)

School of Engineering | Computer Science

J. Presper Eckert 1919-1995

- Co-designer of ENIAC, the first general purpose, electronic, digital computer
- Also, co-designer of UNIVAC I, the first commercial computer
- Electrical engineer

Context-Free Languages



Context-Free Languages

PDA = CFG = CNF

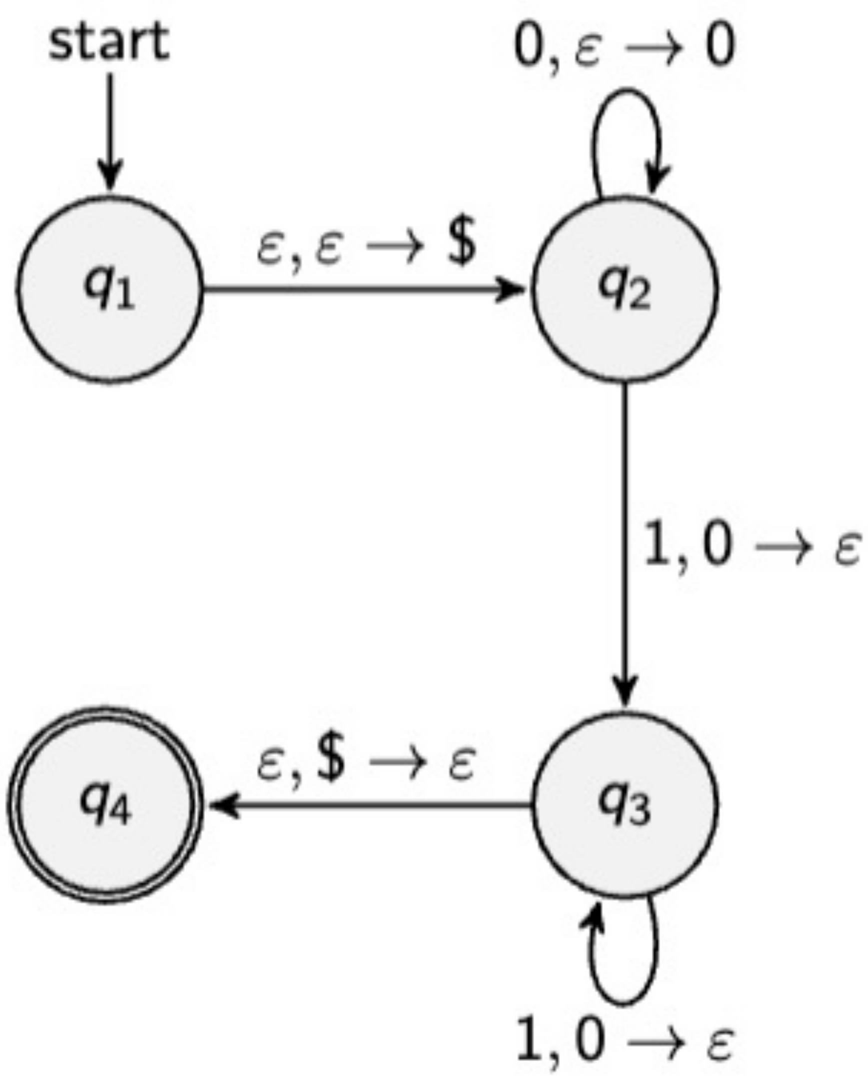
Closed under union, $\cup$, concatenation, $\circ$, and star, $*$.

Pushdown Automata

- The DFA’s and NFA’s that represent regular languages <u>lack memory</u>
- In contrast, Context-Free Languages are represented by Pushdown Automata (PDA), which contain a stack.
- They are essentially NFA’s with a stack (LIFO data structure)

Pushdown Automata

- Ex: The PDA for $L = \{0^n 1^n \mid n \geq 0\}$ is:



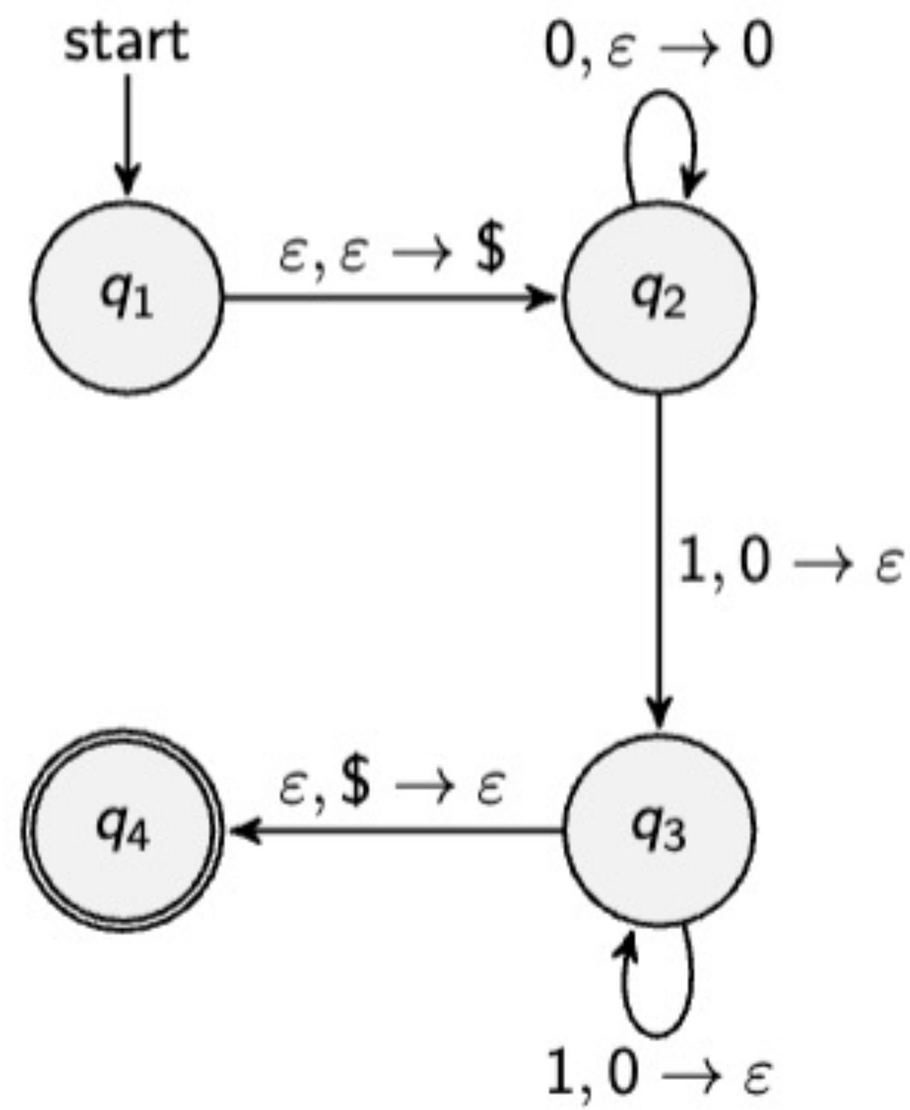
- The transition arrow can be read as: input symbol, pop from the stack → push onto the stack.
- So ε, ε → \$ means that the input symbol is the empty string, you are popping the empty string from the stack and pushing the \$, the start symbol onto the stack.

VCU School of Engineering | Computer Science

Pushdown Automata

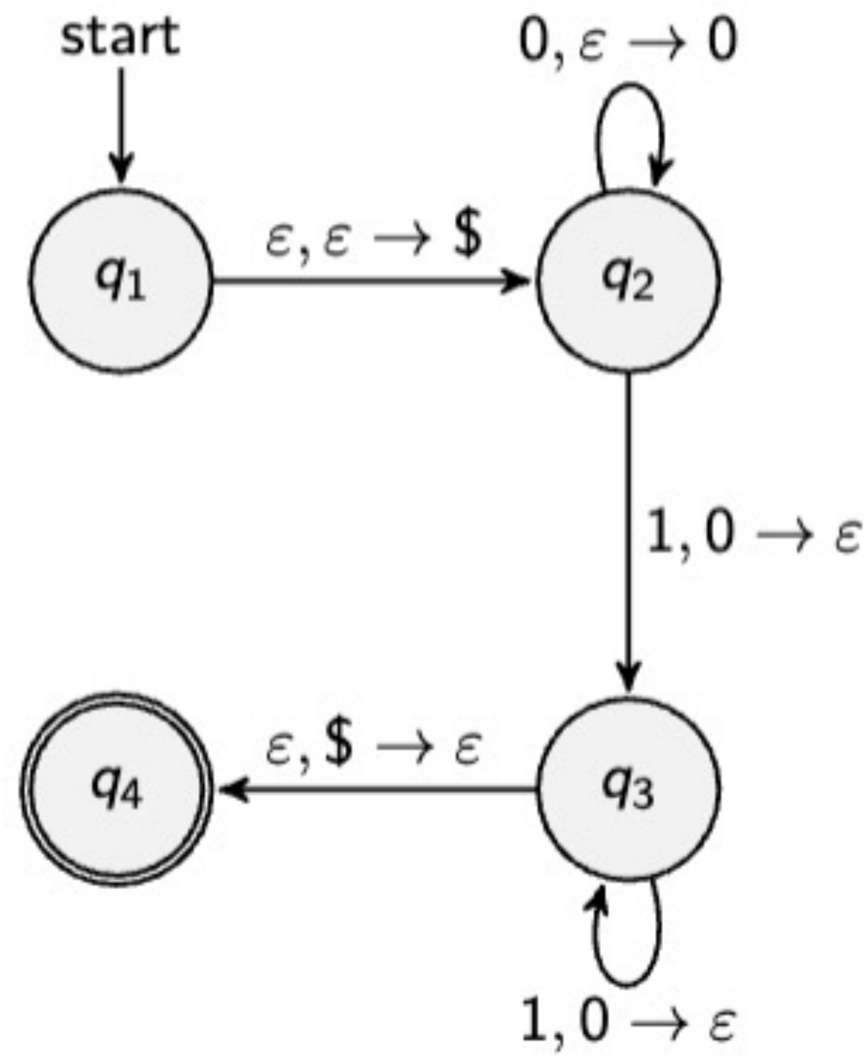
- Ex: The PDA for $L = \{0^n 1^n \mid n \geq 0\}$ is:
  - Idea:

1. Push \$, start symbol, on stack
2. If read a 1 first reject
3. Each time you read a 0, push a 0 on the stack
4. Each time you read a 1, pop a 0 off the stack
5. Accept if all symbols have been read and the \$ is the last symbol popped from the stack



Pushdown Automata

- Ex: The PDA for $L = \{0^n 1^n \mid n \geq 0\}$ is:
  - Will this PDA accept 00011?
    - Push \$ on stack
    - Read 0, push 0 on stack
    - Read 0, push 0 on stack
    - Read 0, push 0 on stack
    - Read 1, pop 0 off stack
    - Read 1, pop 0 off stack
    - Read empty string, cannot pop \$ off stack since still have a 0 on the stack, reject.



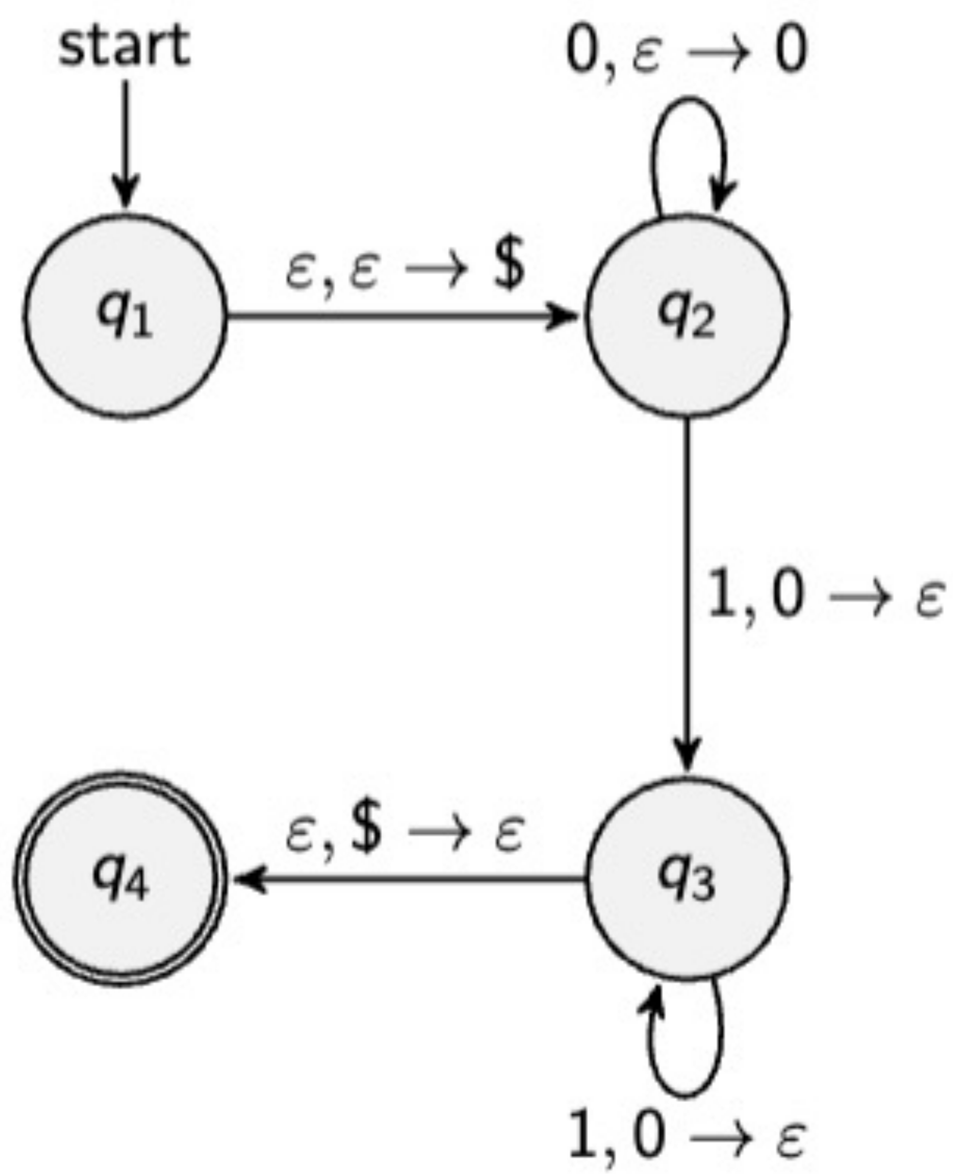
VCU School of Engineering | Computer Science

PDA Formal Definition

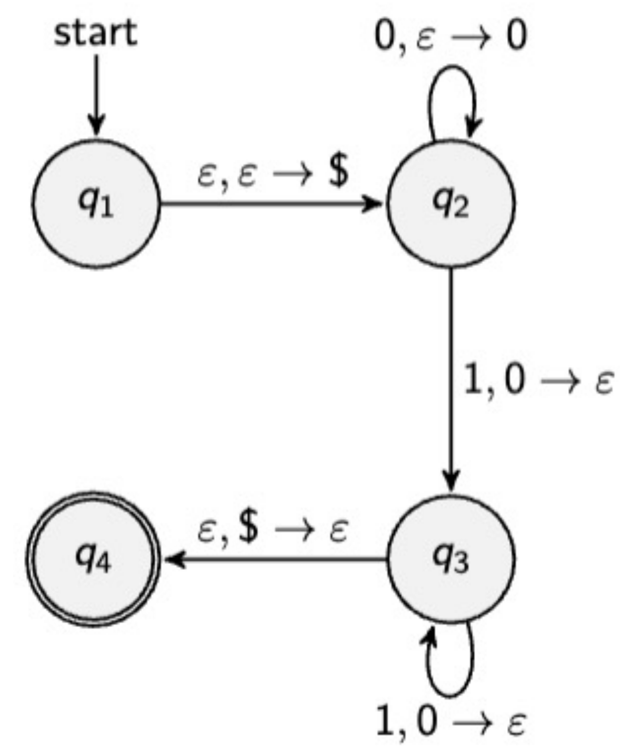
- Formal Definition of PDA: A PDA is a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_0, F)$ such that:
  1. Q is a finite set of states
  2. Σ is the input alphabet
  3. Γ is the stack alphabet
  4. $\delta: Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \rightarrow P(Q \times \Gamma_\varepsilon)$
  5. $q_0 \in Q$ is the start state
  6. $F \subseteq Q$ is the set of accept states

PDA Formal Definition

- Formal Definition of PDA Ex:
- Ex: PDA $L = \{0^n 1^n \mid n \geq 0\}$:
  - $Q =$
  - $\Sigma =$
  - $\Gamma =$
  - $q_0 =$
  - $F =$
  - $\delta =$



PDA Formal Definition

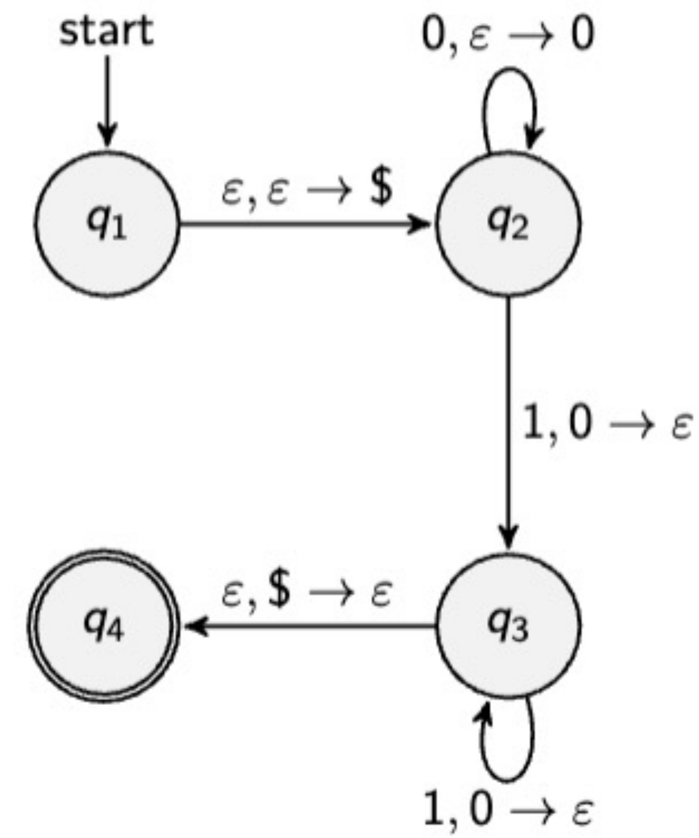


- Formal Definition of PDA Ex:
- Ex: PDA $L = \{0^n 1^n \mid n \geq 0\}$:
  - $\delta =$

Input:	0			1			ε		
Pop off stack:	0	$	ε	0	$	ε	0	$	ε
q_1									
q_2									
q_3									
q_4									

PDA Formal Definition

- <u>Formal Definition of PDA Ex</u>:
- Ex: PDA $L = \{0^n 1^n \mid n \geq 0\}$:
  - $Q = \{q_1, q_2, q_3, q_4\}$, $\Sigma = \{0, 1\}$,
  - $\Gamma = \{0, \$\}$, $q_0 = q_1$, $F = \{q_4\}$,
  - $\delta =$



Input:	0			1			ε		
Pop off 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									

$\{(q_3, \varepsilon)\}$ Means move to state, q_3, and push ε on the stack

Formal Definition of Acceptance

- A PDA M accepts a string w if w can be written as $w = w_1 w_2 \ldots wm$ for $m \geq 0$ where $w_i \in \Sigma_\varepsilon$, and there exists sequences of states $(r_0, r_1, \ldots, rm)$ and there exist strings $s_0, s_1, \ldots, sm \in \Gamma^*$ such that:

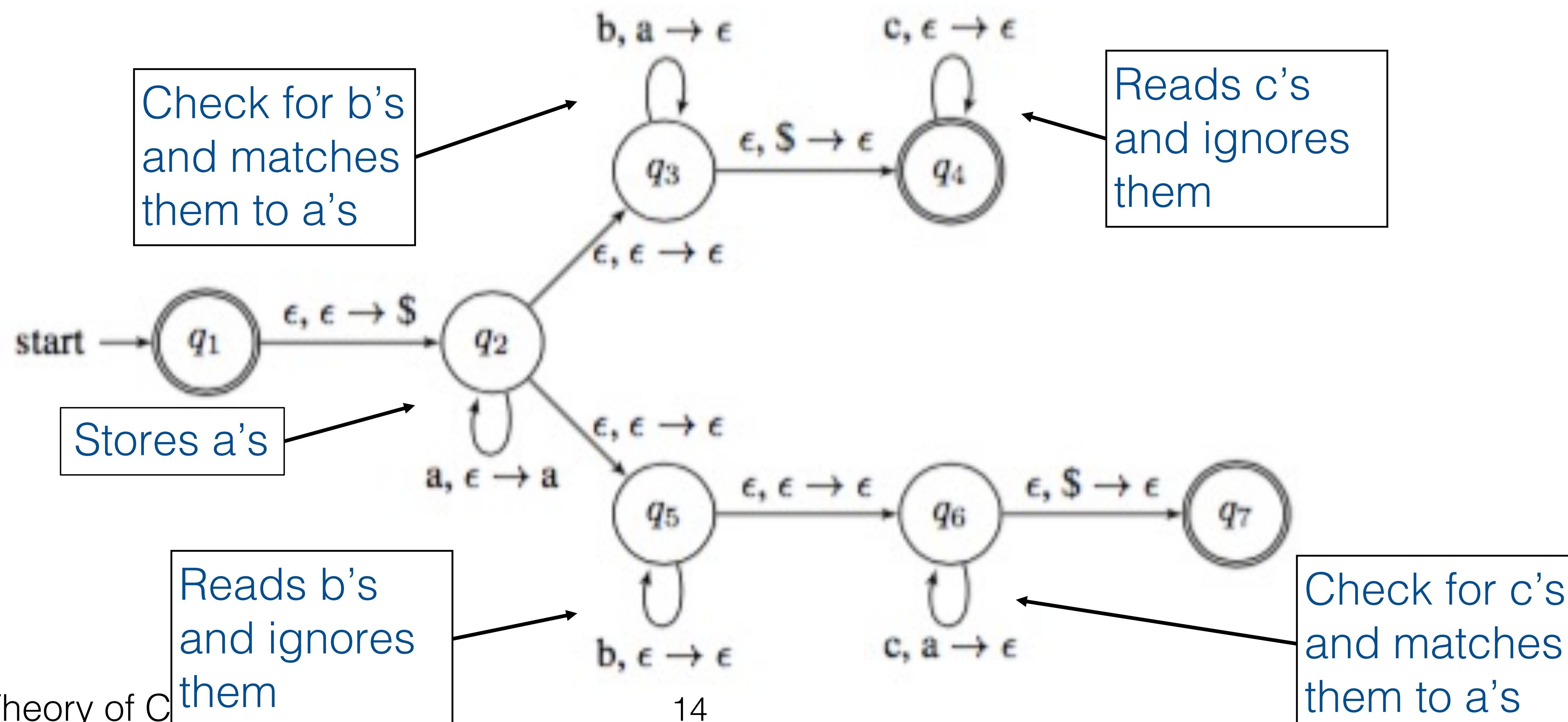
  1. $r_0 = q_0$ and $s_0 = \varepsilon$ (start on start state with empty stack)
  2. For $i = 0, \ldots, m-1$, have $(r_{i+1}, b) \in \delta(r_i, w_{i+1}, a)$, where $s_i = at$ and $s_{i+1} = bt$ for some $a, b \in \Gamma_\varepsilon$ and $t \in \Gamma^*$ (b is what is pushed on the stack, w_{i+1} is input, a is popped off the stack, s_i are the contents of the stack at state i)
  3. $r_m \in F$ (end in accept state)

Creating a PDA for a Language

- Ex: What would the PDA for language $L = \{a^i b^j c^k \mid i, j, k \geq 0 \; and \; i = j \; or \; i = k\}$ be:
    - Ex: aabbccc, aabbbcc, aabb, …

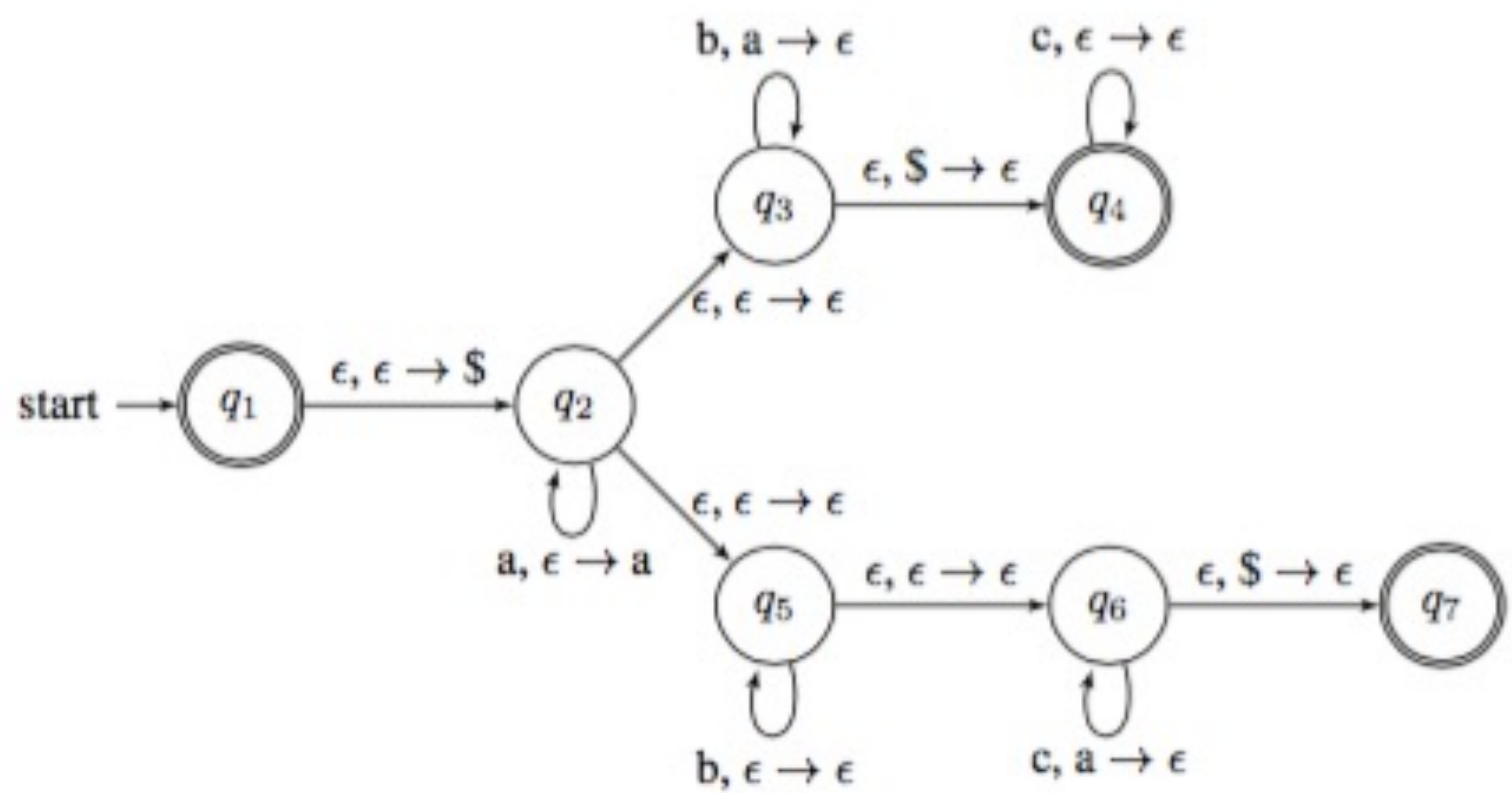
Creating a PDA for a Language

- Ex: What would the PDA for language $L = \{a^i b^j c^k \mid i,j,k \geq 0 \; and \; i = j \; or \; i = k\}$ be:
  - Ex: aabbccc, aabbbcc, aabb, …

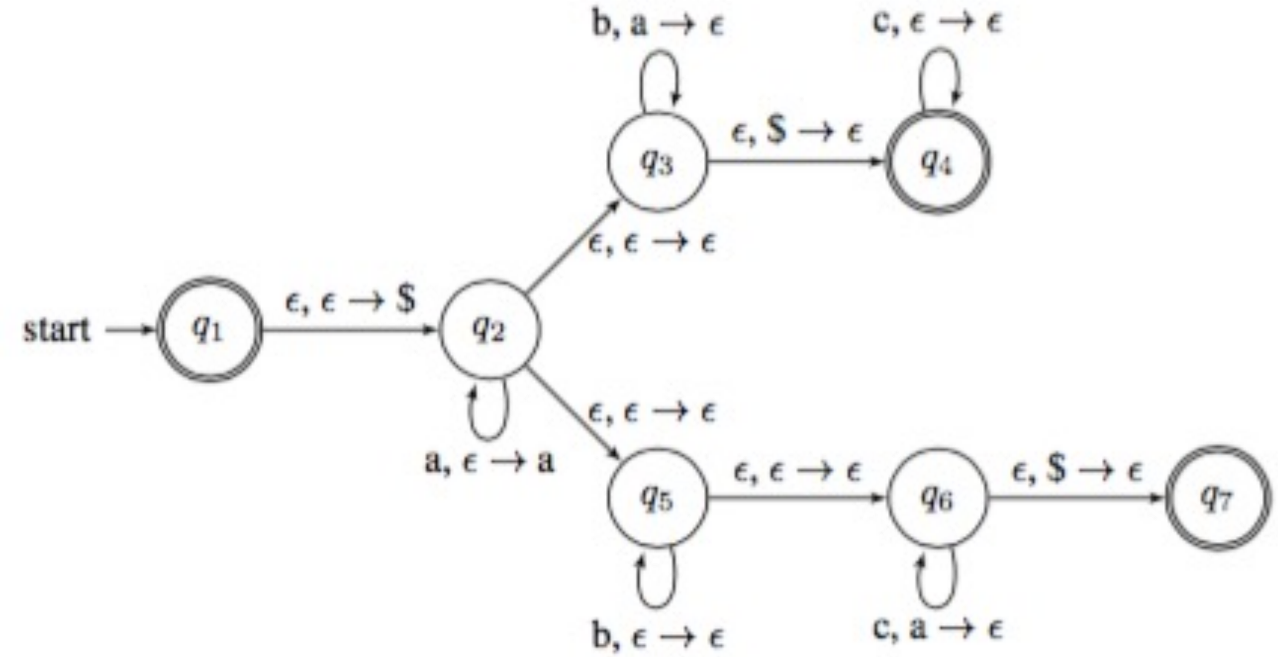


Formal Definition

- Ex: PDA $L = \{a^i b^j c^k \mid i, j, k \ \geq 0 \ and \ i = j \ or \ i = k\}$:
    - Q =
    - Σ =
    - Γ =
    - q_0 =
    - F =



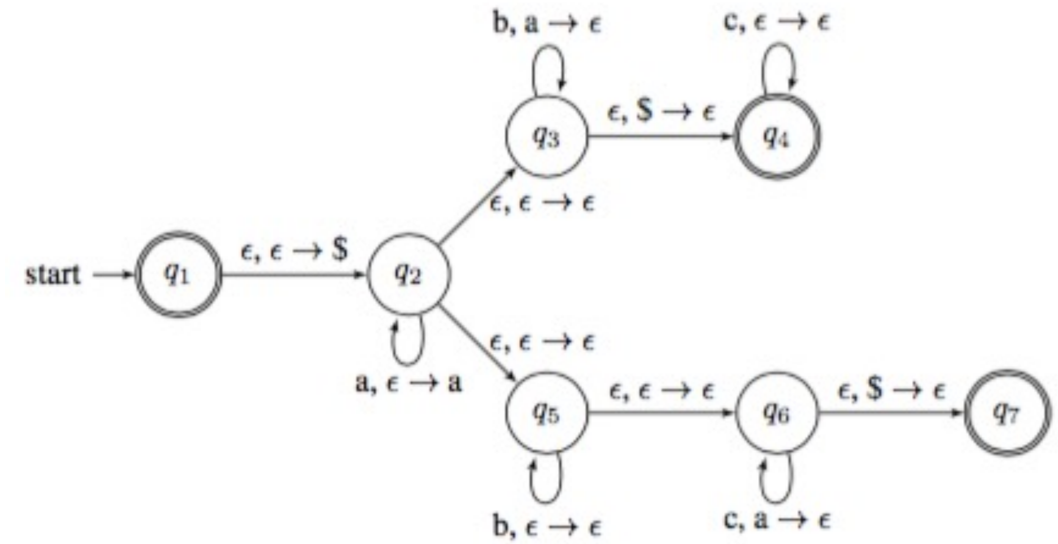
Formal Definition



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

Input:	a			b			c			ε		
Pop off stack:	a	\$	ε	a	\$	ε	a	\$	ε	0	\$	ε
q_1												
q_2												
q_3												
q_4												
q_5												
q_6												
q_7												

Formal Definition



- Ex: PDA $L = \{a^i b^j c^k \mid i, j, k \geq 0 \; and \; i = j \; or \; i = k\}$:
  - $Q = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$, $\Sigma = \{a, b, c\}$, $\Gamma = \{a, \$\}$, $q_0 = q_1$, $F = \{q_1, q_4, q_7\}, \delta =$

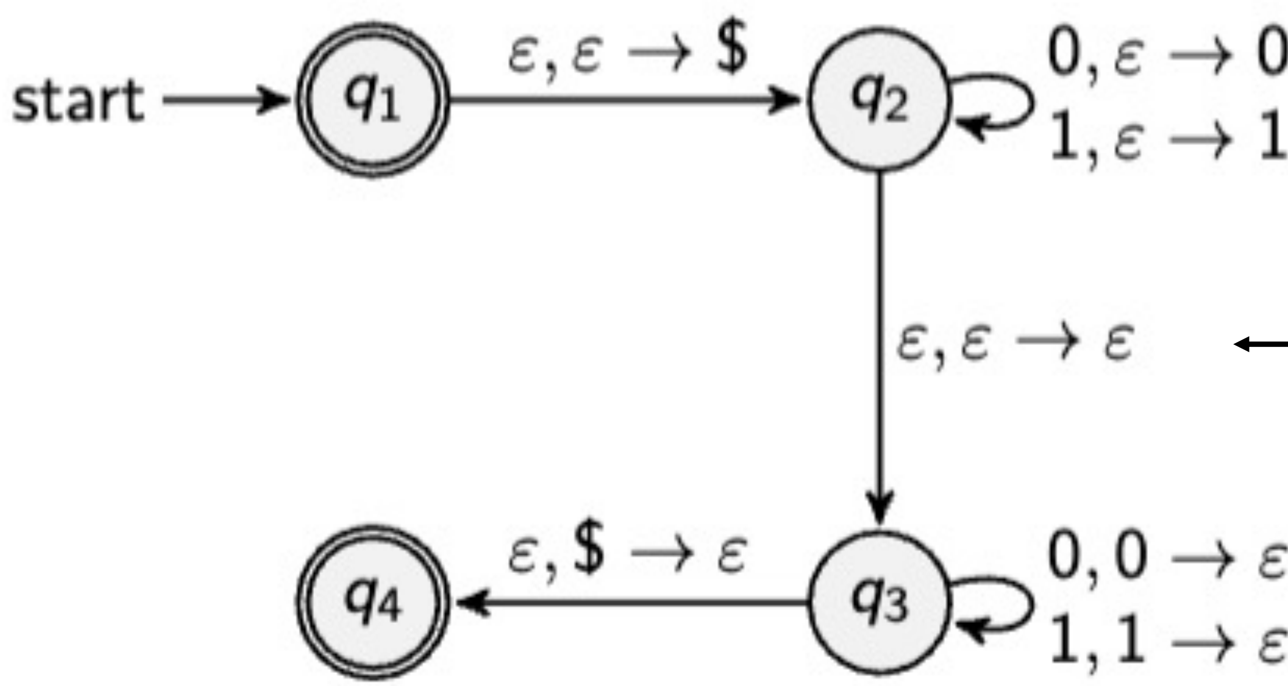
Input:	a			b			c			ε		
Pop off stack:	a	$	ε	a	$	ε	a	$	ε	0	$	ε
q_1												$\{(q_2, \$)\}$
q_2			$\{(q_2, a)\}$									$\{(q_3, \varepsilon)\}$ $\{(q_5, \varepsilon)\}$
q_3				$\{(q_3, \varepsilon)\}$							$\{(q_4, \varepsilon)\}$	
q_4									$\{(q_4, \varepsilon)\}$			
q_5						$\{(q_5, \varepsilon)\}$						$\{(q_6, \varepsilon)\}$
q_6							$\{(q_6, \varepsilon)\}$				$\{(q_7, \varepsilon)\}$	
q_7												

Creating a PDA for a Language

- Ex: What would the PDA for language $L = \{ww^R \mid w \in \{0, 1\}^*$, where w^R is the reverse of $w\}$ be:
  - Ex: w = car, w^R = rac

Creating a PDA for a Language

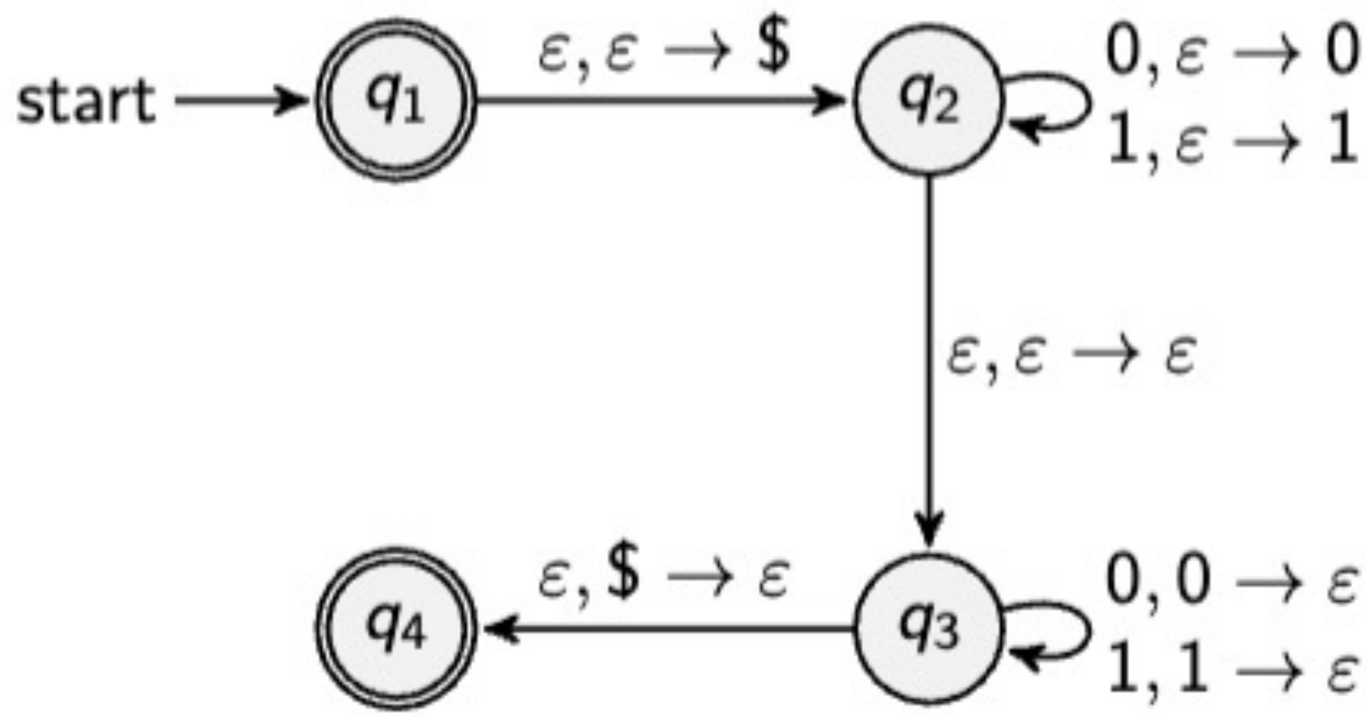
- Ex: What would the PDA for language $L = \{ww^R \mid w \in \{0, 1\}^*$, where w^R is the reverse of $w\}$ be:
  - Ex: w = car, w^R = rac



Checks for the middle of the string after each inputted symbol

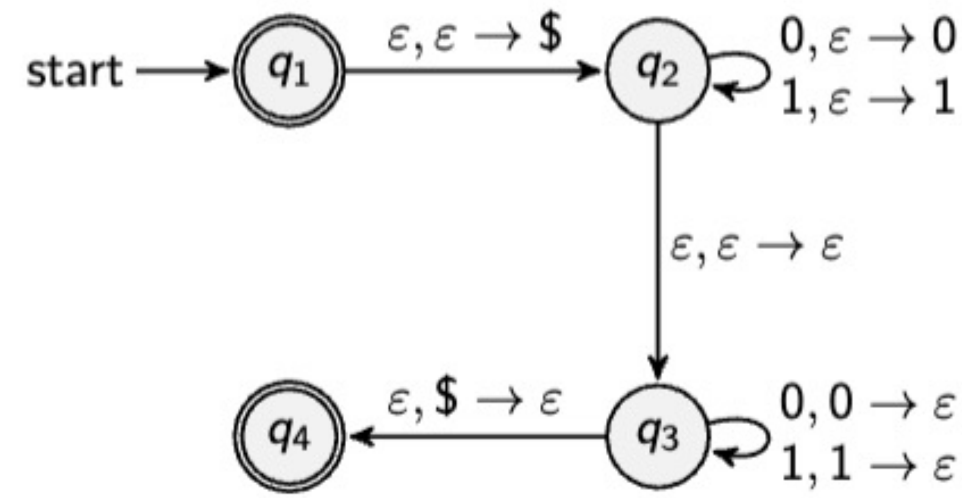
Formal Definition

- Ex: PDA L = L = $\{ww^R \mid w \in \{0, 1\}^*$, where w^R is the reverse of $w\}$:



  - Q =
  - Σ =
  - Γ =
  - q_0 =
  - F =

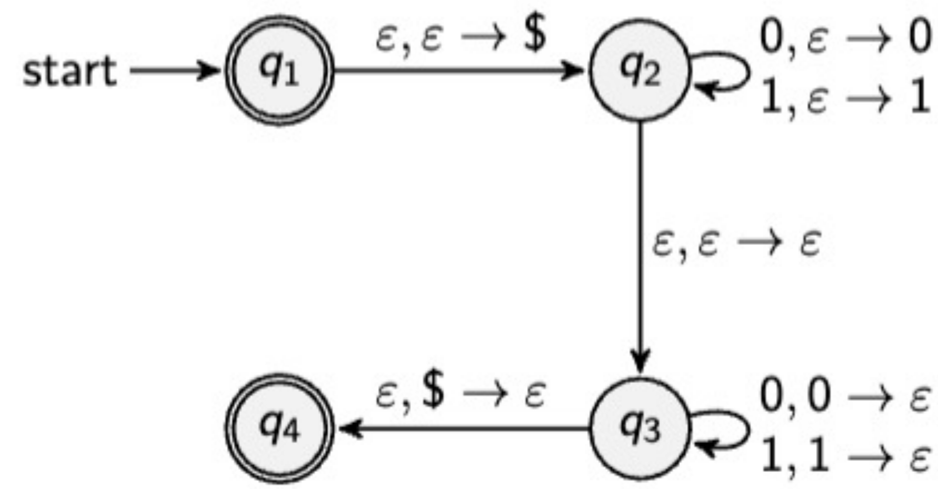
Formal Definition



- Ex: PDA L = L = $\{ww^R \mid w \in \{0, 1\}^*$, where w^R is the reverse of w$\}$:
  - $\delta =$

Input:	0				1				ε			
Pop off stack:	0	1	\$	ε	0	1	\$	ε	0	1	\$	ε
q_1												
q_2												
q_3												
q_4												

Formal Definition



- Ex: PDA L = L = $\{ww^R \mid w \in \{0, 1\}^*$, where w^R is the reverse of w$\}$:
  - $Q = \{q_1, q_2, q_3, q_4\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \$\}$, $q_0 = q_1$, $F = \{q_1, q_4\}$,

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

Try It

- Give state diagrams of PDAs that accepts $\{0^i 1^j \mid i \geq 2, j \geq 1, i > j\}$. (Modify the examples from this lecture.)

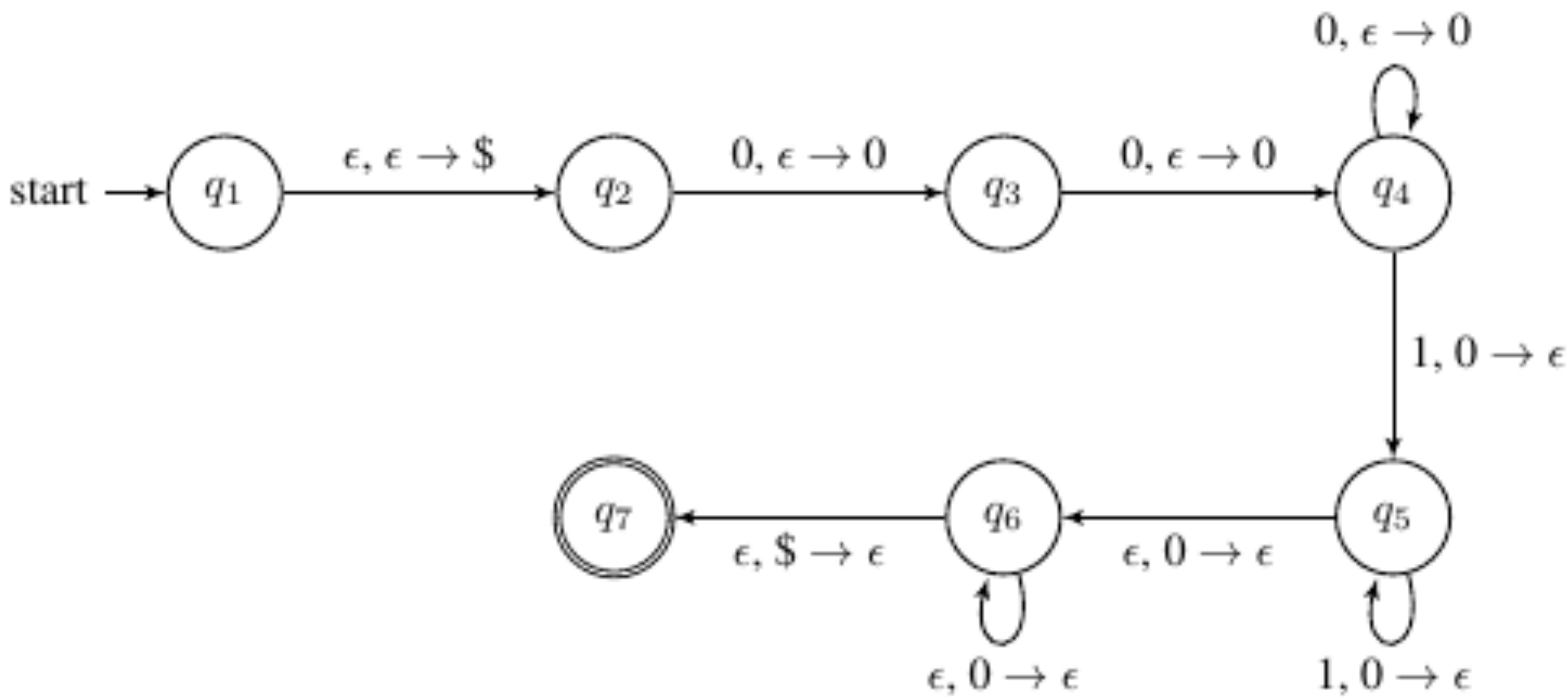
- Create a state diagram for a PDA recognizing the language as defined below:
  - $Q = \{q_0, q_1, q_2, q_3\}$
  - $\Sigma = \{a, b\}$
  - $\Gamma = \{a, \$\}$
  - $F = \{q_3\}$

$\delta =$

δ	a			b			ϵ		
pop	a	\$	ϵ	a	\$	ϵ	a	\$	ϵ
q_0									$(q_1, \$)$
q_1			(q_1, a)						(q_2, ϵ)
q_2				(q_2, ϵ)		(q_2, ϵ)		(q_3, ϵ)	
q_3									

Try It

- Give state diagrams of PDAs that accepts $\{0^i 1^j \mid i \geq 2, j \geq 1, i > j\}$. (Modify the examples from this lecture.)

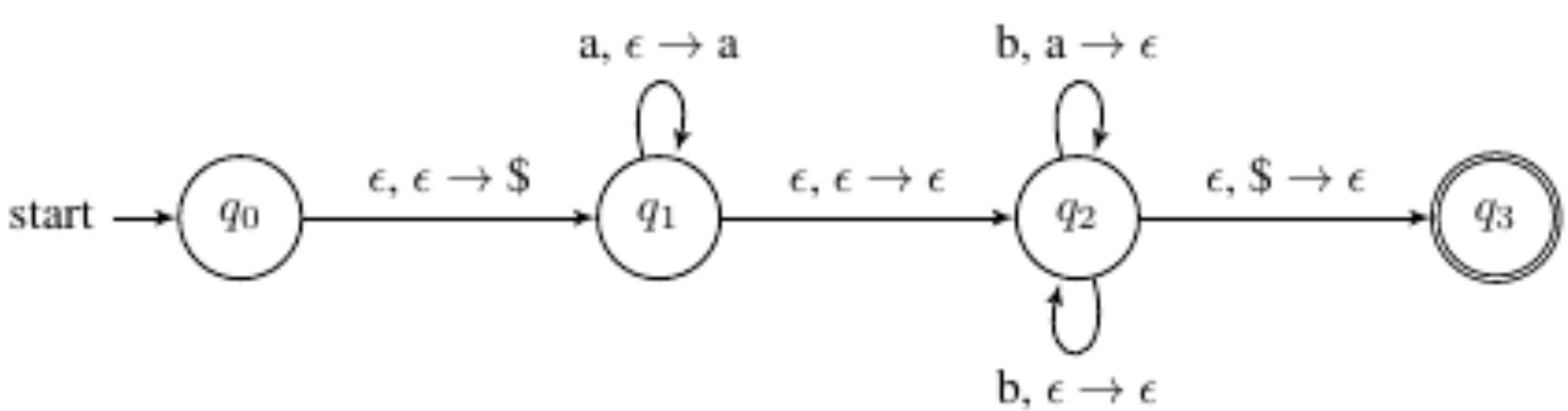


Try It

- Create a state diagram for a PDA recognizing the language as defined below:
  - $Q = \{q_0, q_1, q_2, q_3\}$
  - $\Sigma = \{a, b\}$
  - $\Gamma = \{a, \$\}$
  - $F = \{q_3\}$

$\delta =$

δ	a			b			ϵ		
pop	a	\$	ϵ	a	\$	ϵ	a	\$	ϵ
q_0									$(q_1, \$)$
q_1			(q_1, a)						(q_2, ϵ)
q_2				(q_2, ϵ)		(q_2, ϵ)		(q_3, ϵ)	
q_3									



Language:
$\{a^i b^j \mid i \geq 0, j \geq 0, i \leq j\}$