CSE 2001: Introduction to Theory of Computation

Winter 2013

Suprakash Datta

datta@cse.yorku.ca

Office: CSEB 3043

Phone: 416-736-2100 ext 77875

Course page: http://www.cse.yorku.ca/course/2001

Next

- Chapter 2:
  - Pushdown Automata

More examples of CFLs

- $L(G) = \{0^n 1^{2n} \mid n = 1,2,\dots\}$
- $L(G) = \{xx^R \mid x \text{ is a string over } \{a,b\}\}$
- $L(G) = \{x \mid x$ is a string over $\{1,0\}$ with an equal number of 1's and 0's$\}$

Next: Pushdown automata (PDA)

Add a stack to a Finite Automaton

- Can serve as type of memory or counter
- More powerful than Finite Automata
- Accepts Context-Free Languages (CFLs)
- Unlike FAs, nondeterminism **makes a difference** for PDAs. We will only study non-deterministic PDAs and omit Sec 2.4 (3rd Ed) on DPDAs.

Pushdown Automata

Pushdown automata are for context-free languages what finite automata are for regular languages.

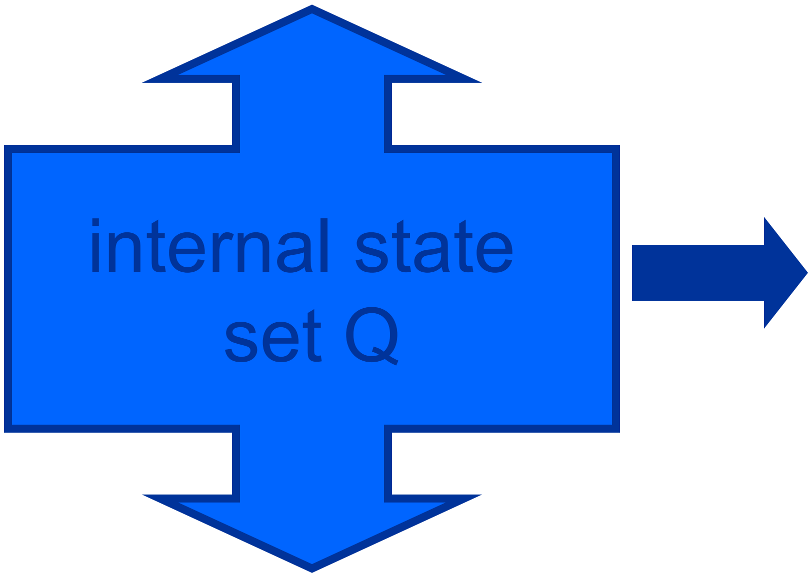
PDAs are *recognizing automata* that have a single stack (= memory):

Last-In First-Out *pushing* and *popping*

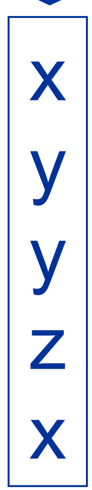
Non-deterministic PDAs can make non-deterministic choices (like NFA) to find accepting paths of computation.

Informal Description PDA (1)

input w = 00100100111100101



stack



The PDA M reads w and stack element.

Depending on

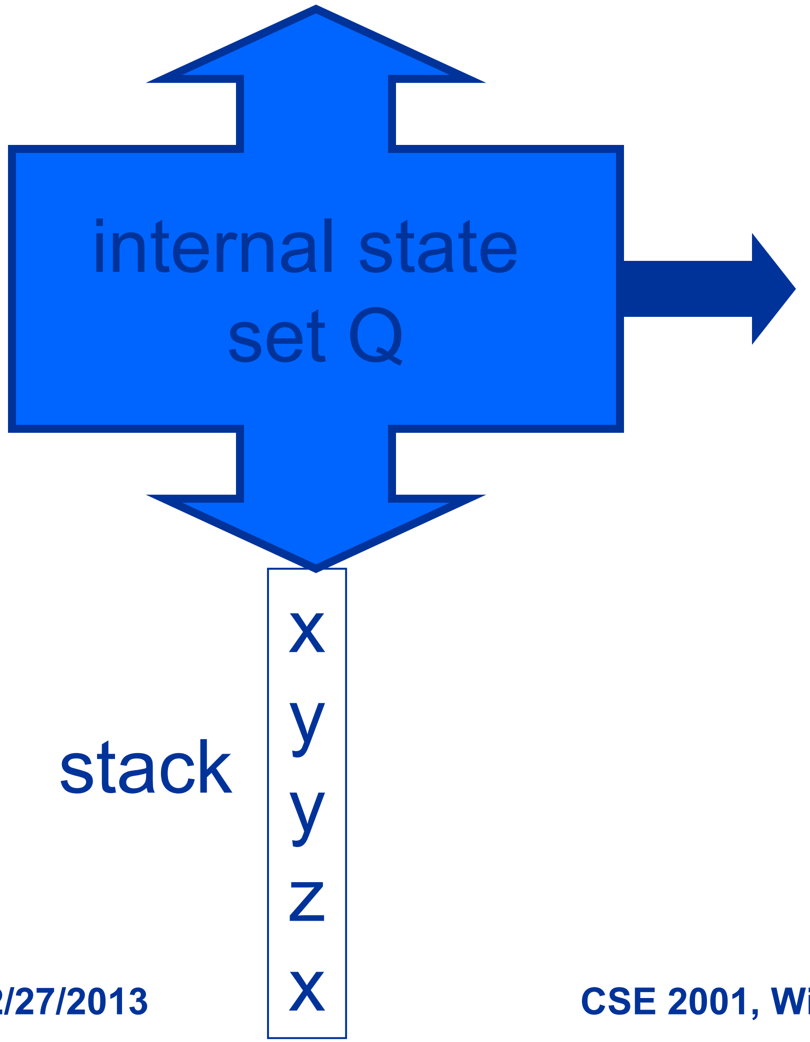
- input $w_i \in \Sigma_\varepsilon$,
- stack $s_j \in \Gamma_\varepsilon$, and
- state $q_k \in Q$

the PDA M:

- jumps to a new state,
- pushes an element Γ_ε (nondeterministically)

Informal Description PDA (2)

input w = 00100100111100101



After the PDA has read complete input, M will be in state $\in$ Q

If possible to end in accepting state $\in F \subseteq Q$, then M accepts w

Formal Description of a PDA

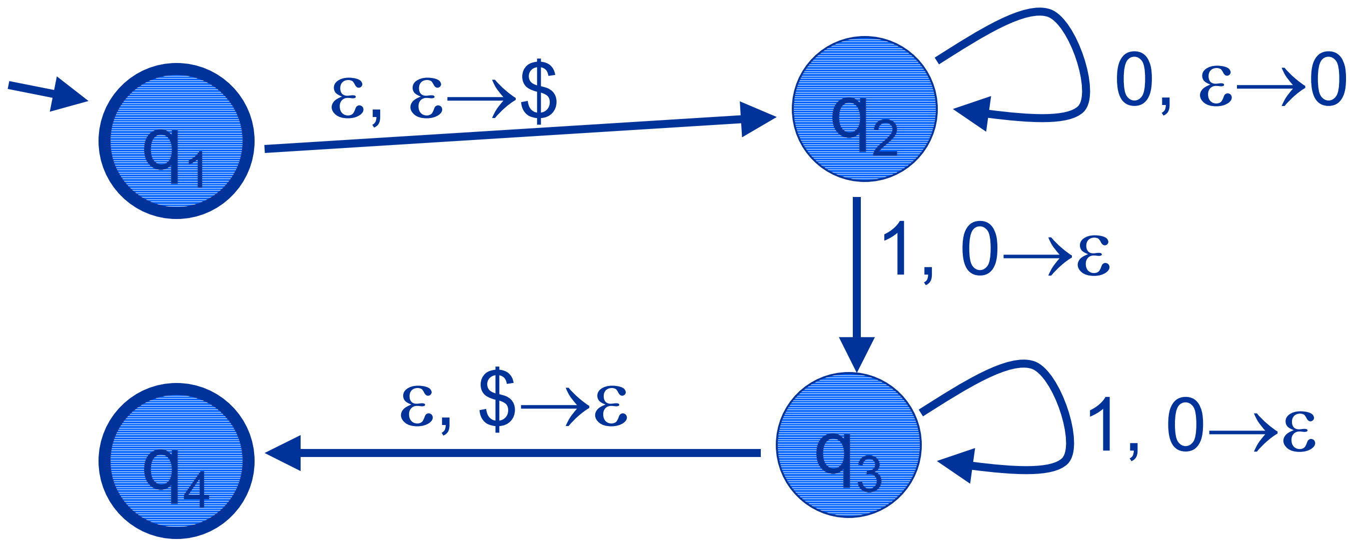
A Pushdown Automata M is defined by a six tuple $(Q,\Sigma,\Gamma,\delta,q_0,F)$, with

- Q finite set of states
- Σ finite input alphabet
- Γ finite stack alphabet
- q_0 start state $\in Q$
- F set of accepting states $\subseteq Q$
- δ transition function

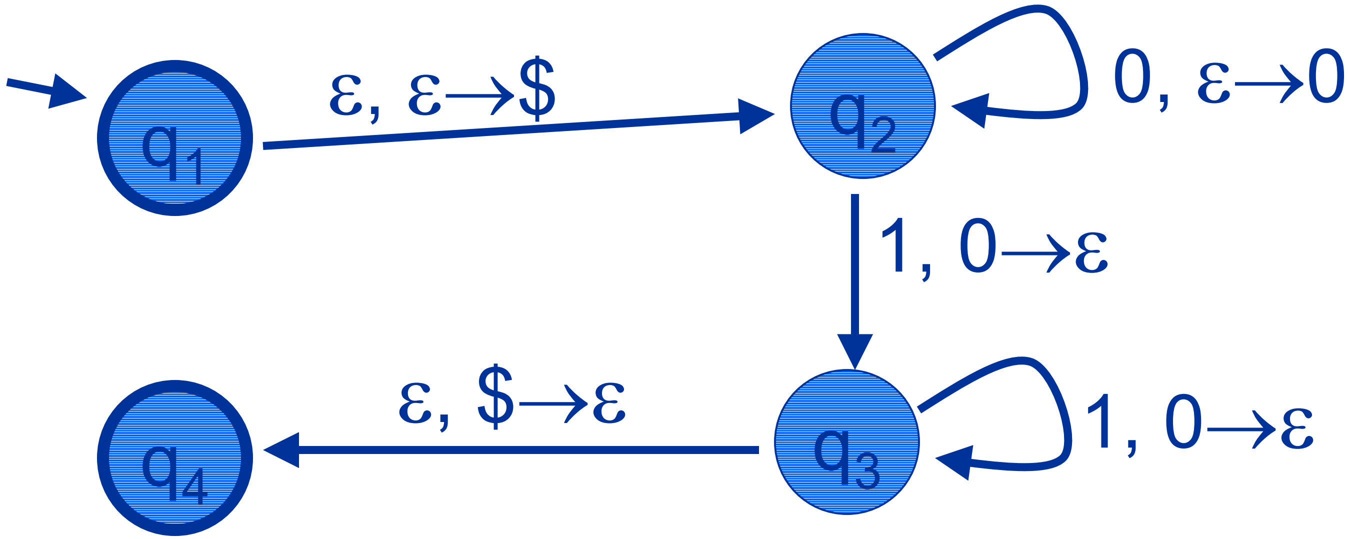
$$\delta: Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \rightarrow \mathcal{P}(Q \times \Gamma_\varepsilon)$$

PDA for L = { $0^n1^n \mid n \geq 0$ }

<u>Example 2.9</u>:
The PDA first pushes " \$ 0^n " on stack.
Then, while reading the 1^n string, the zeros are popped again.
If, in the end, \$ is left on stack, then "accept"



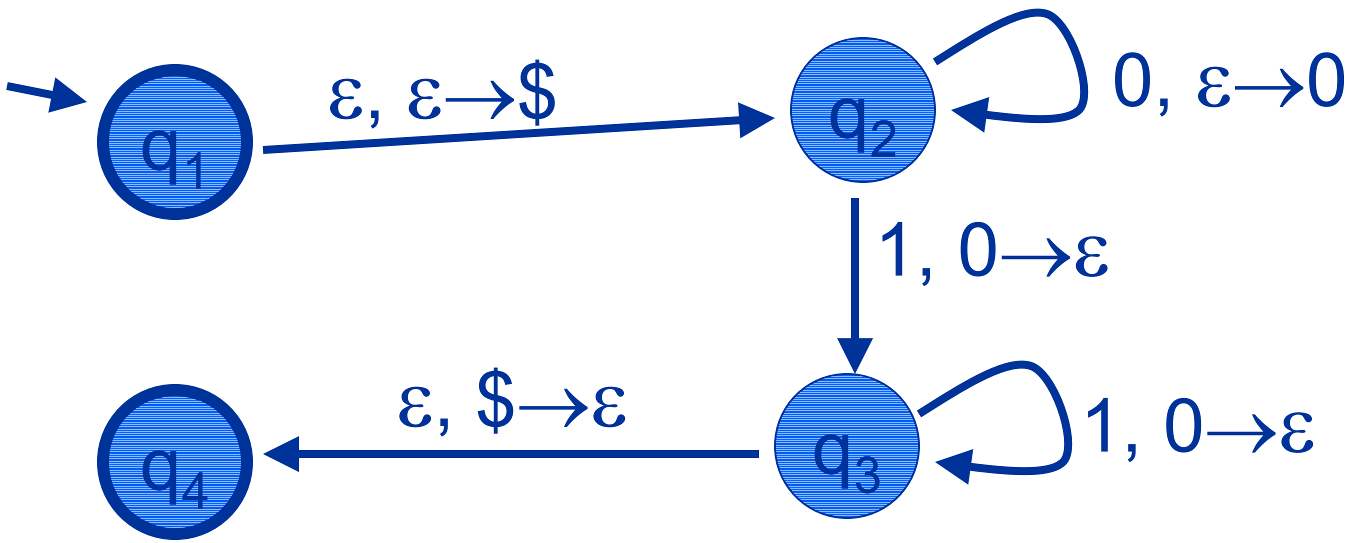
Machine Diagram for 0^n1^n



On w = 000111 (state; stack) evolution:

$(q_1; \varepsilon) \to (q_2; \$) \to (q_2; 0\$) \to (q_2; 00\$)$
$\to (q_2; 000\$) \to (q_3; 00\$) \to (q_3; 0\$) \to (q_3; \$)$
$\to (q_4; \varepsilon)$ This final q_4 is an accepting state

Machine Diagram for 0^n1^n



On w = 0101 (state; stack) evolution:

$(q_1; \varepsilon) \to (q_2; \$) \to (q_2; 0\$) \to (q_3; \$) \to (q_4; \varepsilon) \ldots$

But we still have part of input "01".

There is no accepting path.

An important example

- $L = \{a^i b^j a^k \mid i=j \text{ or } i=k \}$
- (Example 2.16, p 115. 3rd ed)

- Try $L = \{ww^R \mid w \text{ is any binary string} \}$