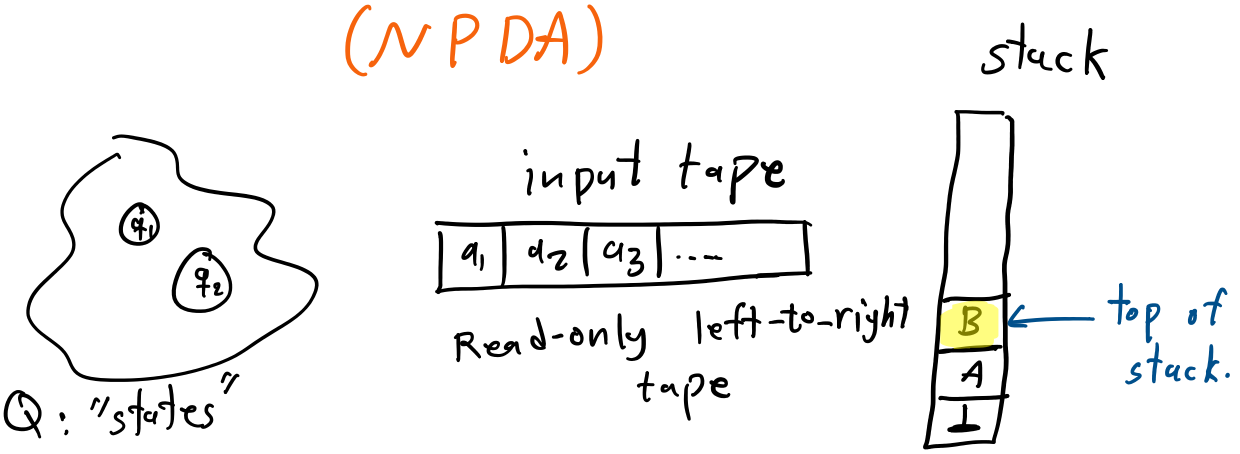
Lecture 27

Tuesday, March 26, 2024 11:40 AM

Quiz # 5

Non Deterministic Pushdown Automata (NPDA)



At each step, the machine <u>acts</u> based on:

(i) current state

(ii) current input symbol

(iii) the top stack symbol

"Act" means: update the state, goes to the next input symbol, and pushes some stack symbols to stack.

An NPDA is 7 tuple

$$M = (Q, \Sigma, \Gamma, \delta, s, \perp, F)$$

* Q: finite set of states
* Σ: " " " input symbols.
* Γ: " " " stack symbol
* $\delta \subseteq ((Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma) \times (Q \times \Gamma^*))$

* $\delta \subseteq (Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma) \times (Q \times \Gamma^*)$

  $|\delta| < \infty$ (finite set)

* s : start state
* $\perp$: initial stack symbol
* $F \subseteq Q$: set of accept states

For example if $((q_1, a, B), (q_2, \overset{\text{top}}{C}DE)) \in \delta$ and the current state is q_1, the current input letter is "a" and top of stack is "B", then the machine can pop B and push E, D, C and transition to q_2 and the next input letter.

Before we formally define acceptance by an NPDA, let's see an example.

$A = \{ x \in \{[,]\}^* : x \text{ is a valid parenthesization} \}$

$S \to SS \mid [S] \mid \varepsilon$

How do we design an NPDA for A?

$Q = \{s, q_2\}$, $F = \{q_2\}$

$Q = \{s, q_2\}$, $F = \{q_2\}$

$\Gamma = \{\perp, [\}$, $\Sigma = \{[,]\}$

δ includes:

① $((s, [, \perp), (s, [\perp))$

② $((s,], [), (s, \varepsilon))$

③ $((s, \varepsilon, \perp), (q_2, \perp))$

④ $((s, [, [), (s, [[))$

let's see how [[]] can be accepted:

$$\underbrace{(s, [[]], \perp)}_{\text{initial config}} \xrightarrow[M]{1} (s, []], [\perp)$$

$$\xrightarrow[M]{1} (s,]], [[\perp) \xrightarrow[M]{1} (s,], [\perp) \longrightarrow$$

$$\xrightarrow[M]{1} (s, \varepsilon, \perp) \xrightarrow[M]{1} (q_2, \varepsilon, \perp)$$

we have consumed the whole string and we are in an accept state