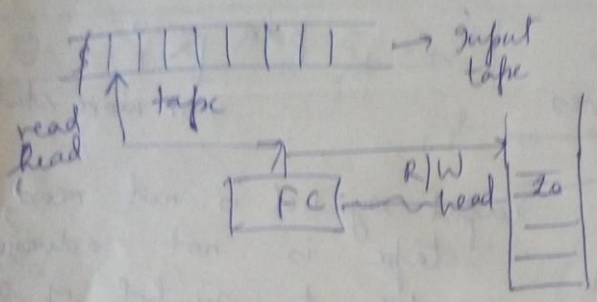
Pushdown Automata → used to represent CFLs & Gs

PDA = FSM + a stack.

Lecture - 21

PDA
↓
3 components

input tape | control unit | stack with infinite size.



pda – nondeterministic device
– recognizes CFL

dpda → recognizes most of the programming langs

CFGs are recursive
↓
useful for prog langs

(*) A PDA may or may not read an input symbol, but it has to read the top of the stack in every transition.

(*) A PDA can be described as :-

$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ → set of final state

q_0 → initial state ; Z_0 → initial stack symbol

Q → finite # of states

Σ → input alphabet

Γ → stack symbol

δ → transition function from $Q \times (\Sigma \cup \varepsilon) \times \Gamma$ to finite subset of $Q \times \Gamma^*$

(*) An automaton to recognize CFL may require additional amount of storage which will be used to store data.

- processing of pda is in terms of moves.

- move indicates ← present state, inputs (tape / stack), possible next state and strings to be written on the stack.

dpda < 0, 1

pda ← 0, 1 or more states.

Types of moves

- input symbol read from tape and tape advances
- input symbol is not read, ∴ tape is not advanced. topmost symbol of stack is used.
- stack top is modified without reading input symbol — ε_move

---

ID (Instantaneous Description)

Let M be a pda, then its config is at a given instant is defined by an ID ← state, remaining input string, remaining stack string.

- represented by ⊢ (turnstile notation)

parenthesis matching.

① ( ) ← pop the corresponding (.
push onto to stack

② $a^n b^n$ $\{ n \geq 0 \}$

read n as, find n # of bs.

Ex. $L = \{ wcw^R \mid$ where $w \in (a+b)^*$

w^R is reverse of w and
$c \in \Sigma$ indicates middle of string.

Sol

Assume $w = abb$ $\quad w^R = bba$

$\therefore$ abbcbba

↓ palindrome.

→ Devise a method to accept the string of palindrome.

- c is the middle of the string.
- start with q_0.
- keep pushing all input symbols on to the stack and stay in state q_0 till c is encountered.
- As c is encountered, change state to q_1. ⇒ middle of string reached.
- In state q_1, after scaning the input symbol, compare with one on top of stack. If they are same discard both and scan next symbol.

If there is a mismatch, (3)
- machine halts and string is rejected
- As the last symbol on the stack is matched, stack is empty.

⇓

string is a palindrome

∴ state changes to q_2 ⇒ accepting state.

Transitions

A ~~PDA~~ has ε-transitions ⇒ non-deterministic

⇓

there may be multiple transition from the ~~given~~ same state on a given input when a specified symbol is present on the stack

=

$\delta(q_0, a, Z_0)$	(q_0, aZ_0)
$\delta(q_0, b, Z_0)$	(q_0, bZ_0)
$\delta(q_0, a, a)$	(q_0, aa)
$\delta(q_0, b, a)$	(q_0, ba)
$\delta(q_0, a, b)$	(q_0, ab)
$\delta(q_0, b, b)$	(q_0, bb)
$\delta(q_0, c, Z_0)$	(q_1, Z_0)
$\delta(q_0, c, a)$	(q_1, a)
$\delta(q_0, c, b)$	(q_1, b)
$\delta(q_1, a, a)$	(q_1, ε)
$\delta(q_1, b, b)$	(q_1, ε)
$\delta(q_1, \varepsilon, Z_0)$	(q_2, Z_0)

$\delta: Q \times (\Sigma \cup \varepsilon) \times \Gamma$ to $Q \times \Gamma^*$

(4) Actions performed by the PDA depend on
1) current state
2) next input symbol
3) symbol on top of stack.

action performed by the machine consists of (4)

(1) Changing the states from one state to another

(2) replacing the symbol on the stack.

δ(state, input_symbol, stack_symbol) = (next_state, stack symbol)

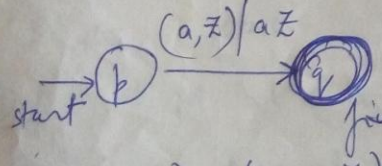
- has 3 arguments
- 2 same as NFA.
- 3rd is symbol at top of stack.

Graphical representation of a PDA

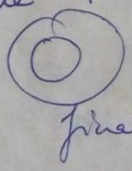
- The actions performed by a PDA in state q on an input symbol a and when the top of the stack is Z can be represented by 2 methods: →

(1) transitions — δ notation

(2) graphical notation.

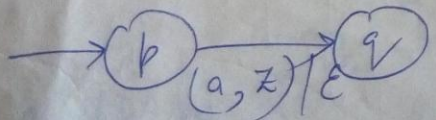


p, q are states of the PDA.

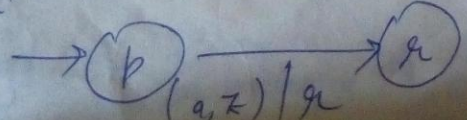


(1) $\delta(p, a, Z) = (q, aZ)$

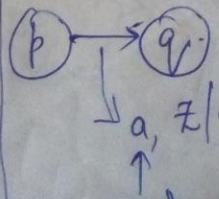
(2) $\delta(p, a, Z) = (q, \varepsilon)$



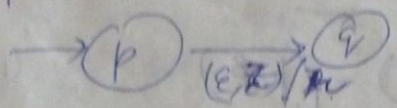
(3) $\delta(p, a, Z) = (q, \alpha)$



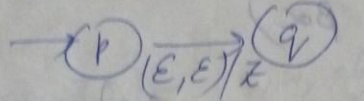
$\delta(p, a, Z) = (q, \alpha)$



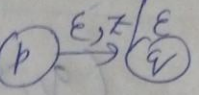
(4) $\delta(p, \varepsilon, z) = (q, a)$



(5) $\delta(p, \varepsilon, \varepsilon) = (q, z)$



(6) $\delta(p, \varepsilon, z) = (q, \varepsilon)$



Instantaneous description (ID)

- The current configuration of PDA at any given instant can be decided by an ID.
- ID gives the current state of the PDA, the remaining string to be processed and the entire contents of the stack.

Def $M = (Q, \Sigma, \Gamma, \delta, q^0, z_0, A/F)$ be a PDA.

An ID is defined as

ID $\Rightarrow (q, w, \alpha)$ → current contents of stack

- q: current state
- w: string to be processed
- α: leftmost symbol in string α is at top of stack.
- rightmost is at the bottom

Let a PDA be

$(q, aw, z\alpha)$

- q: current state
- aw: string to be processed
- $z\alpha$: current contents of stack, z is at top

Acceptance of a language by a PDA (6)

w is accepted by a PDA, when
(1) we get the final state from initial state.
(2) get an empty stack from the start state.

$(p, aw, T\beta) \vdash (q, w, ab)$.
input symbol 'a' is consumed and is put on top of stack.

⊢ turnstile notation
↓
used for connecting pairs of IDs that represent one or more moves of a PDA.
⊢ represents transition.

* Language accepted by a final state is defined as

$$L(M) = \{ w \mid (q_0, w, Z_0) \vdash^* (p, \varepsilon, \alpha) \}$$
$$\alpha \in \Gamma^*, \; p \in A/F, \; w \in \Sigma^*$$

⇒ PDA currently in state q_0, after scanning string w enters state p.

⇒ the machine is in state p, the input symbol should be ε and the contents of stack are irrelevant.

$$N(M) = \{ w \mid (q_0, w, Z_0) \vdash^* (p, \varepsilon, \varepsilon) \}$$
↑ null stack
$$w \in \Sigma^*$$
$$q_0, p \in Q$$