Pushdown Automata-PDA

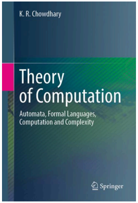
Prof. (Dr.) K.R. Chowdhary
Email: kr.chowdhary@iitj.ac.in

Formerly, Prof. & Head, Department of CS
MBM Engineering College, Jodhpur

Saturday 19th July, 2025

K.R. Chowdhary

Theory of Computation



Automata, Formal Languages, Computation and Complexity

Focuses on pedagogy in its writing, that represents a refreshing approach

Ensures comprehensive and enjoyable learning

Undergone a rigorous classroom testing

Introduction to Pushdown Automata (PDA)

Definition

A PDA consists: a infinite tape, a read head, set of states, and a start state. The additional components from FA are: Pushdown stack, initial symbol on stack, and stack alphabets (Γ). PDA $M = (Q, \Sigma, \delta, s, \Gamma, Z_0, F)$, where,

Q is finite set of states,

Σ is finite set of terminal symbols (language alphabets),

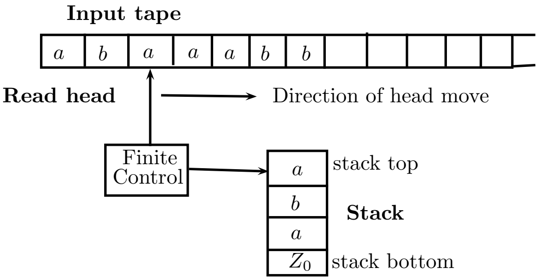
s start state (q_0), F is final state.

δ is transition function: $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow$ *finite subset of* $Q \times \Gamma$.

The transition function of a PDA is so defined, because a PDA may have transitions without any input read.

Introduction to PDA

The PDA has two types of storage; 1) infinite tape, just like the FA, 2) pushdown stack, is read-write memory of arbitrary size, with the restriction that it can be read or written at one end only.



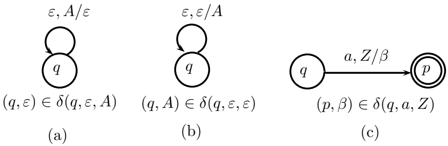
Definition

ID (Instantaneous Description) of a PDA is: $ID : Q \times \Sigma^* \times \Gamma^*$, start-id $\in \{q_0\} \times \Sigma^* \times \{Z_0\}$, e.g., start ID may be (q_0, aaa, Z_0).

PDA Transitions

$\delta(q, a, Z) =$ finite subset of $\{(p_1, \beta_1), (p_2, \beta_2), \ldots, (p_m, \beta_m)\}$. Therefore, $(p_i, \beta_i) \in \delta(q, a, z))$, for $1 \leq i \leq m$.
By default, a *PDA* is non-derministic machine. Due to this fact, a PDA can manipulate the stack without any input from tape. Following are some of the transitions in PDA:

- Case - (a): A PDA currently in state q, stack symbol A, with input ε, moves to state q and write ε on the stack: $\delta(q, \varepsilon, A) = (q, \varepsilon)$.
- Case - (b): A PDA currently in state q, with ε input, and stack symbol ε, moves to state q, and writes A on stack: $\delta(q, \varepsilon, \varepsilon) = (q, A)$.
- Case - (c): A PDA in state q, reads input a, with stack symbol Z, moves to state p and write β on stack: $\delta(q, a, Z) = (p, \beta)$.



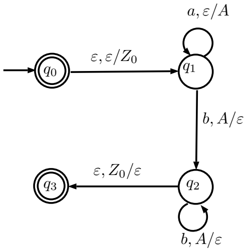
A move of a PDA is defined as $(q, ax, Z\alpha) \vdash_M (q', x, \beta\alpha)$, if $(q', \beta) \in (q, a, Z)$. (In $Z\alpha$, Z is top symbol on stack)

Example

Construct a PDA to recognize $L = \{a^n b^n | n \geq 0\}$.

$M = (Q, \Sigma, \delta, s, F, \Gamma, Z_0)$,
$\Sigma = \{a, b\}$, $\Gamma = \{A\}$
$Q = \{q_0, q_1, q_2, q_3\}$, $F = \{q_0, q_3\}$
$\delta(q_0, \varepsilon, \varepsilon) = (q_1, Z_0)$
$\delta(q_1, a, \varepsilon) = (q_1, A)$
$\delta(q_1, b, A) = (q_2, \varepsilon)$
$\delta(q_2, b, A) = (q_2, \varepsilon)$
$\delta(q_2, \varepsilon, Z_0) = (q_3, \varepsilon)$

$$(q_0, aabb, \varepsilon) \vdash (q_1, aabb, Z_0)$$
$$\vdash (q_1, abb, AZ_0)$$
$$\vdash (q_1, bb, AAZ_0)$$
$$\vdash (q_2, b, AZ_0),$$
$$\vdash (q_2, \varepsilon, Z_0),$$
$\vdash (q_3, \varepsilon, \varepsilon)$, the PDA halts & accepts.



Language Recognition: wcw^R

Example

Construct a PDA to recognize $L = \{wcw^R | w \in \{a,b\}^*\}$.

Solution: Transition function, moves, and PDA:

$$M = (Q, \Sigma, \delta, s, F, \Gamma, Z_0)$$

$$\Sigma = \{a, b, c\},\ d \in \{a, b\},$$

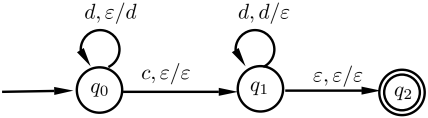
$$Q = \{q_0, q_1, q_2\},\ F = \{q_2\},$$
$$\Gamma = \{a, b, Z_0\}$$

$$\delta(q_0, d, \varepsilon) = (q_0, d)$$

$$\delta(q_0, c, \varepsilon) = (q_1, \varepsilon)$$

$$\delta(q_1, d, d) = (q_1, \varepsilon)$$

$$\delta(q_1, \varepsilon, \varepsilon) = (q_2, \varepsilon)$$

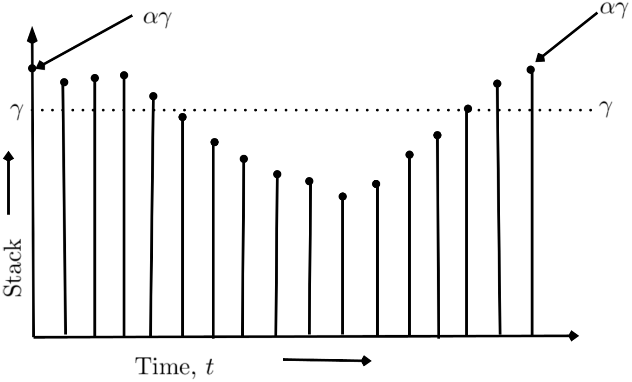


Note that we have not included the transitions corresponding to first writing Z_0 on stack and finally retrieving it back. This is acceptable as PDA is non-deterministic.

- **PDA moves**

1. $(q, x, \alpha) \vdash^* (q', \varepsilon, \beta) \Rightarrow (q, xy, \alpha) \vdash^* (q', y, \beta)$
2. $(q, xy, \alpha) \vdash^* (q', y, \beta) \Rightarrow (q, xy, \alpha\gamma) \vdash^* (q', y, \beta\gamma)$

   The case 1., above is obvious, however, the case 2., is not guaranteed due to the trace of computation shown below.



Bibliography

Chowdhary, K.R. (2025). Pushdown Automata and Parsers. In: Theory of Computation. Springer, Singapore. https://doi.org/10.1007/978-981-97-6234-7_8