MIT-WPU

।। विश्वशान्तिर्ध्रुवं ध्रुवा ।।

Dr. Vishwanath Karad

MIT WORLD PEACE UNIVERSITY | PUNE

TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

Theory of Computation
SY BTech CSE-AIDS

Push Down Automata(PDA)

Course Objectives & Course Outcomes

<u>Course Objectives</u>:

•To understand the basics of automata theory and its operations.

•To understand problem classification and problem solving by machines.

•To study computing machines by describing, classifying and comparing different types of computational models.

•To understand the fundamentals of decidability and computational complexity.

<u>Course Outcomes</u>:

• After completion of this course students will be able:

•To construct finite state machines to solve problems in computing.

•To write mathematical expressions and syntax verification for the formal languages.

•To construct and analyze Push Down Automata and Turing Machine for formal languages.

•To express the understanding of decidability and complexity.

Text Books & Reference Books

- **Text Books**

•John C. Martin, Introduction to Language and Theory of Computation, TMH, 3rdEdition, ISBN: 978-0-07-066048-9.

•Vivek Kulkarni, Theory of Computation, Oxford University Press, ISBN-13: 978-0-19-808458-7.

- **Reference Books**

•K.L.P Mishra,N. Chandrasekaran,Theory of Computer Science (Automata, Languages and Computation), Prentice Hall India, 2nd Edition.

•Michael Sipser, Introduction to the Theory of Computation,CENGAGE Learning, 3rd Edition, ISBN:13:978-81-315-2529-6.

•Daniel Cohen, Introduction to Computer Theory, Wiley India, 2nd Edition, ISBN: 9788126513345.

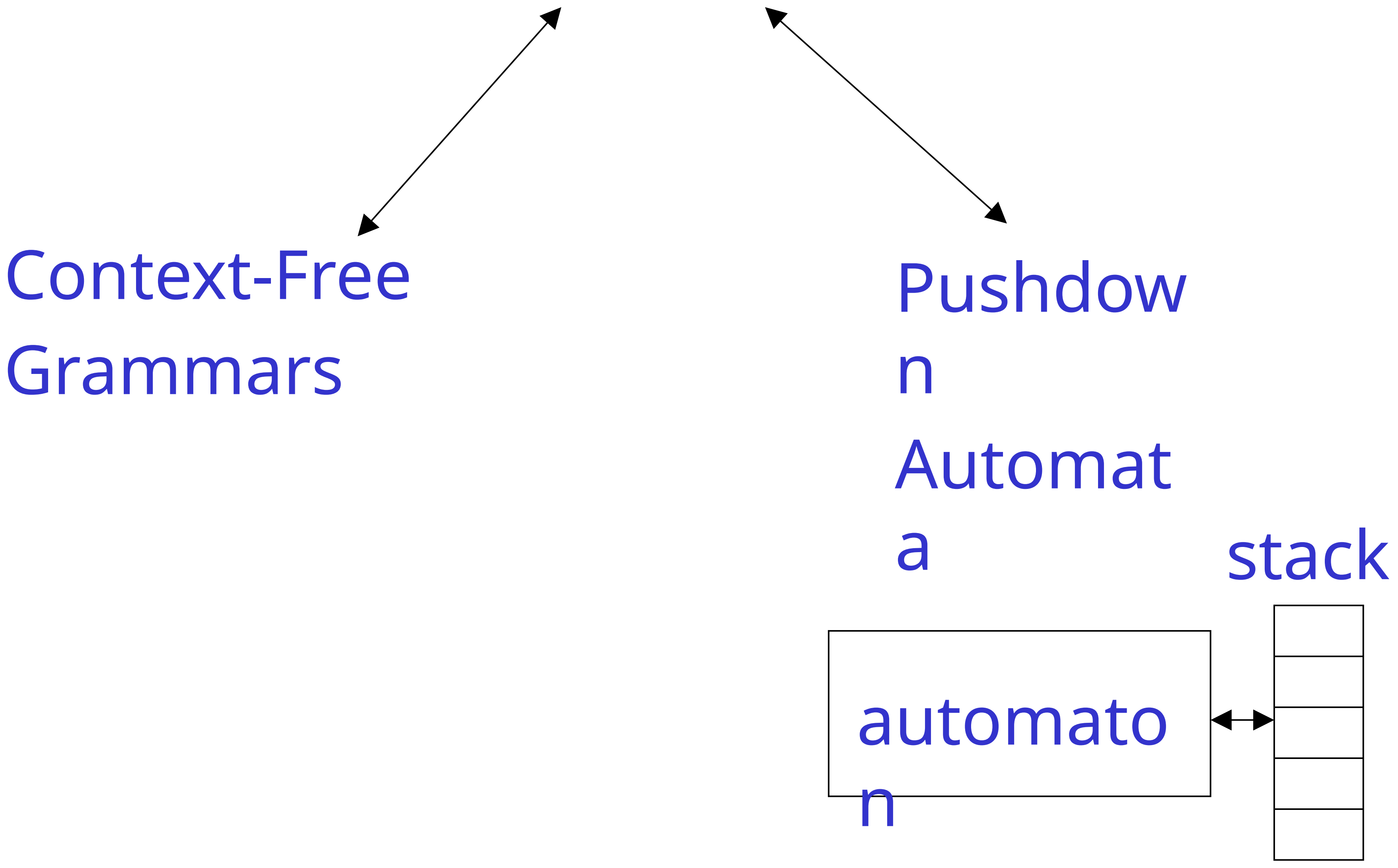
•Kavi Mahesh, Theory of Computation: A Problem Solving Approach, 1st Edition, Wiley-India, ISBN: 978-81-265-3311-4.

Unit IV Push Down Automata

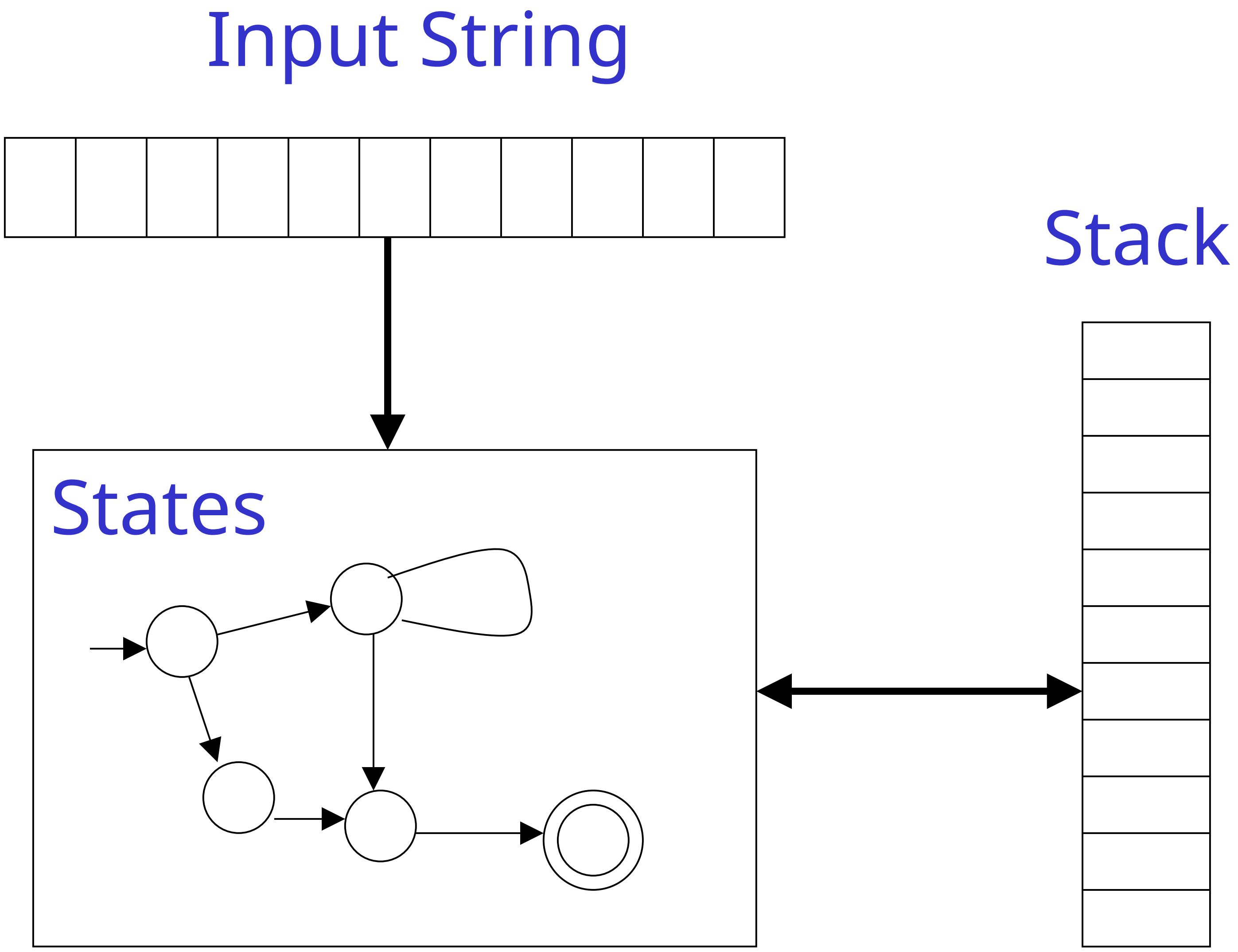
Pushdown automata : Definition, Acceptance of PDA by final State and Empty Stack, Designing PDA, Equivalence of Pushdown automata and CFG, Deterministic Pushdown Automata, Nondeterministic Pushdown Automata.

Push Down Automata

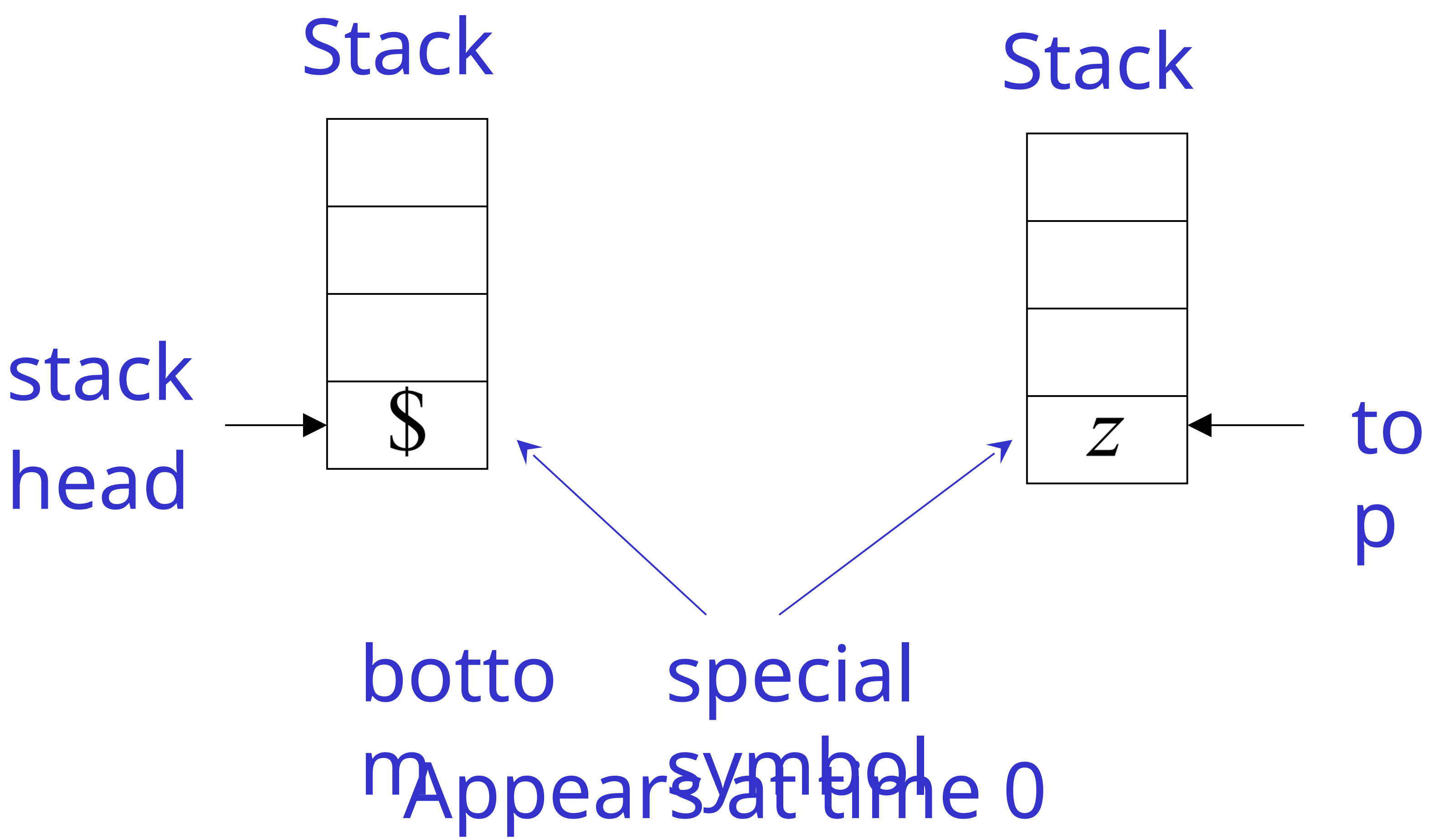
Context-Free Languages



Pushdown Automaton - PDA



Initial Stack Symbol



Appears at time 0

Formal definition of PDA

The PDA is as:

$A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

Where

Q : A finite set of states

Σ : A finite set of input symbols

Γ : A finite stack alphabet or pushdown symbols

δ : the transition function **$Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma$ to the set of finite subsets of $Q \times \Gamma^*$**

q_0: the start state

Z_0 : the start symbol(pushdown symbol)

F : the set of accepting state or final states

Transition Function

δ :The transition function is a triple **δ(q,a,x)**

where

1. q is a state in Q
2. a is either an input symbol in Σ or a =ϵ, the empty string,
3. x is a stack symbol, that is a member of Γ.

The output of δ is a finite set of pairs **(p,ɤ)**

Where

p is the new state and

ɤ is the string of stack symbols that replaces x at the top of the stack.

Instantaneous Description of a PDA

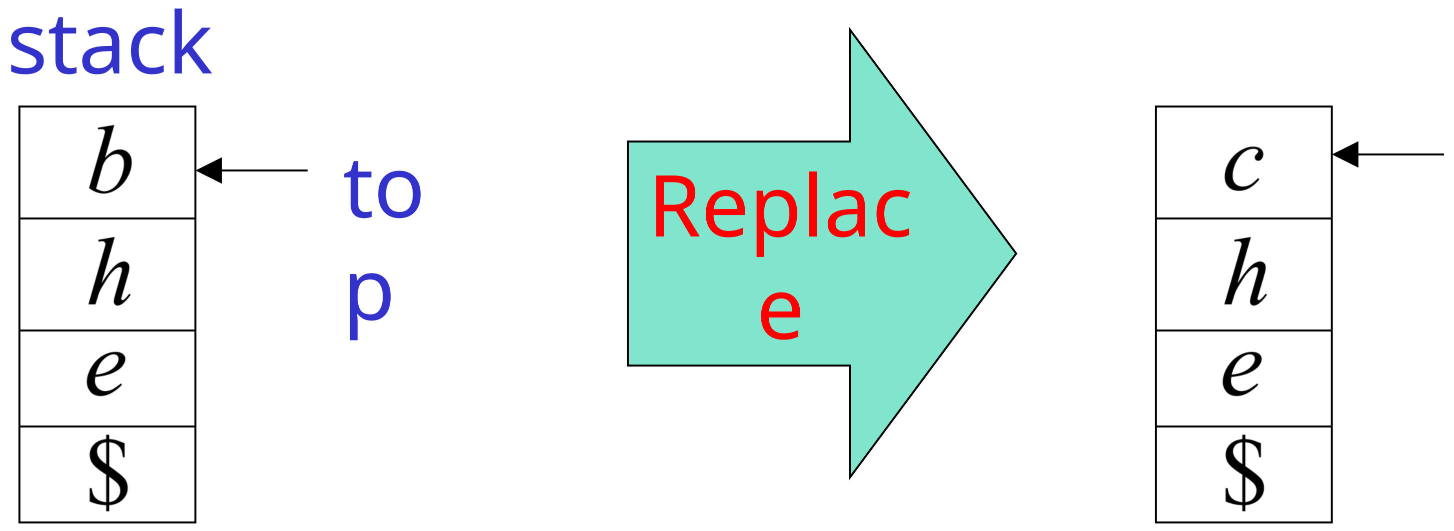
The configuration of a PDA by a **triple (q,w,ɤ)** where

1. **q** is the state
2. **w** is the remaining input
3. **ɤ** is the stack contents

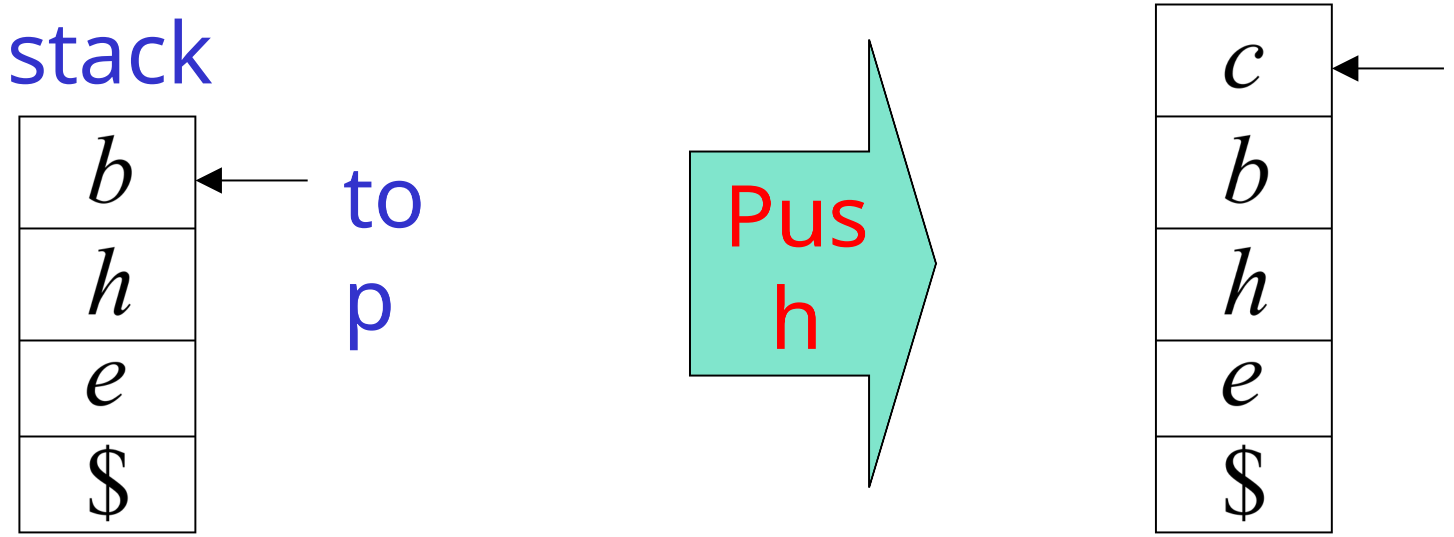
we show the top of the stack at the left end of ɤ and the bottom at the right end.

Such a triple is called an **Instantaneous Description** or **ID** of a PDA.

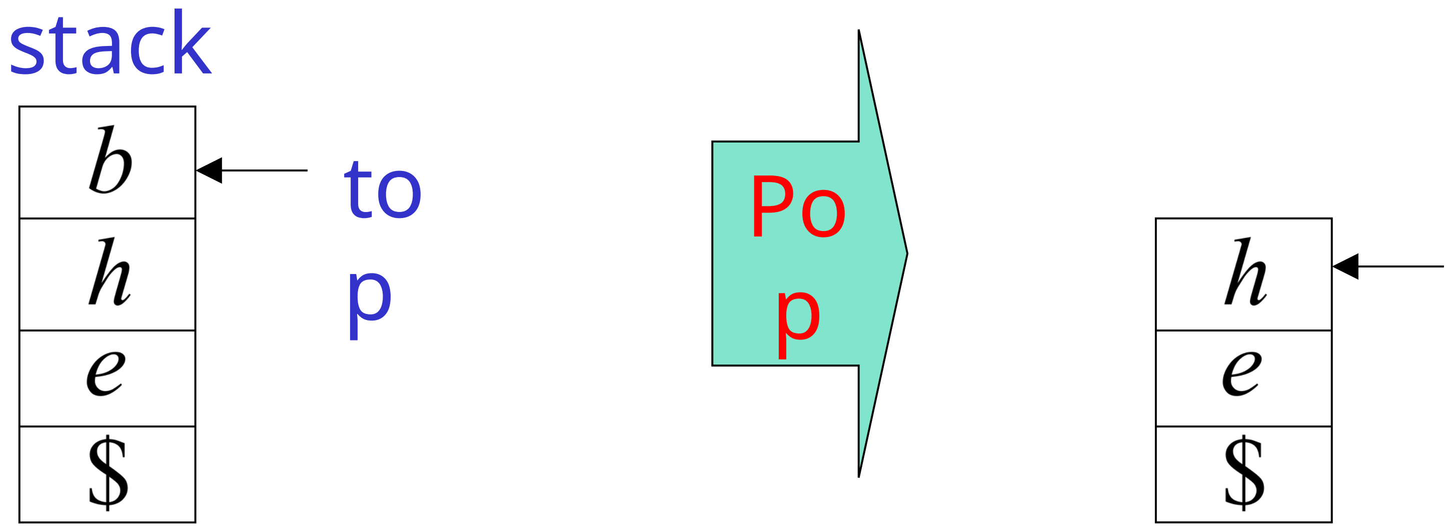
$$\delta(q, a, b) = (q, c)$$



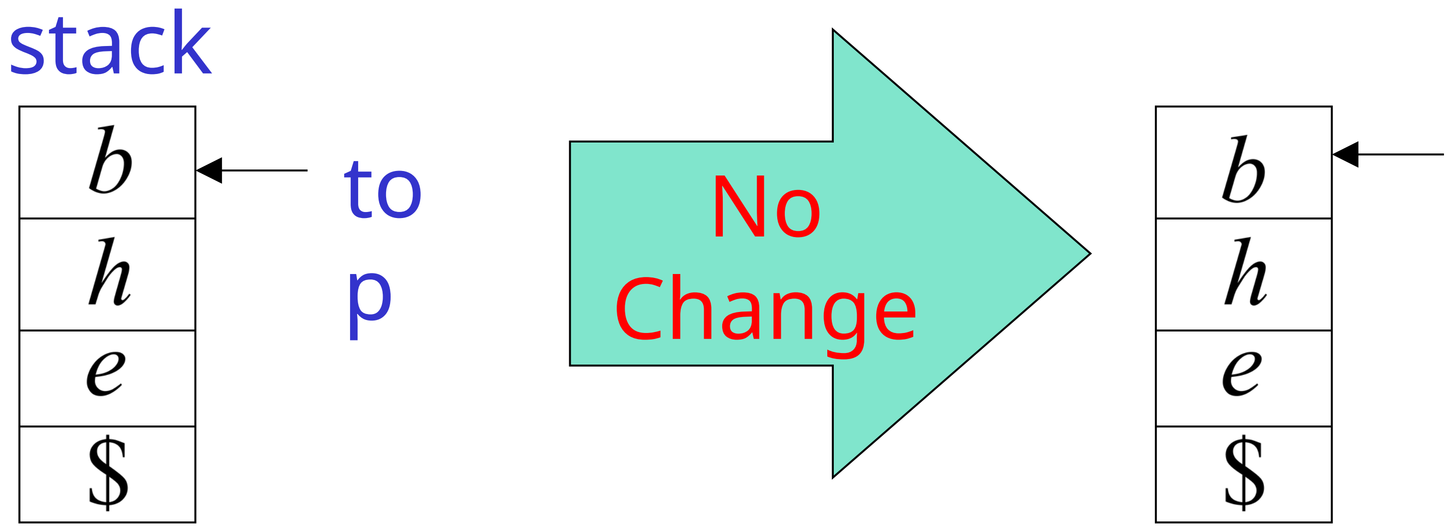
$$\boldsymbol{\delta(q, a, b) = (q, cb)}$$



$\delta(q, a, b) = (q, \epsilon)$



$$\delta(q, a, b) = (q, b)$$



A PDA Example

$A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

$Q=\{q_0, q_1, q_2\}$, $\Sigma=\{a,b\}$, $\Gamma=\{Z_0, a\}$, $F=\{q_2\}$, δ as below

Move no	**State**	**input**	**stack symbol**	**Move**
1	q_0	a	Z_0	(q_0, aZ_0)
2	q_0	a	a	(q_0, aa)
3	q_0	b	a	$(q_1, \in)$
4	q_1	b	a	$(q_1, \in)$
5	q_1	$\in$	Z_0	**(q_2, Z_0)**

Acceptance by PDA using final state

Move no	State	input	stack symbol	Move
1	q_0	a	Z_0	(q_0,aZ_0)
2	q_0	a	a	(q_0,aa)
3	q_0	b	a	$(q_1, \in)$
4	q_1	b	a	$(q_1, \in)$
5	q_1	$\in$	Z_0	**(q_2, Z_0)**

For the string aabb

$(q_0, aabb , Z_0)$
$\vdash (q_0, abb , aZ_0)$
$\vdash (q_0, bb , aaZ_0)$
$\vdash (q_1, b , aZ_0)$
$\vdash (q_1, \in , Z_0)$
$\vdash (q_2, \in , Z_0)$

String **aabb** is accepted, as final state q2 is reached on reading string **aabb** completely

Rejection by PDA

Move no	State	input	stack symbol	Move
1	q_0	a	Z_0	(q_0,aZ_0)
2	q_0	a	a	(q_0,aa)
3	q_0	b	a	$(q_1, \in)$
4	q_1	b	a	$(q_1, \in)$
5	q_1	$\in$	Z_0	(q_2, Z_0)

For the string aabbb

(q_0, aabbb, Z_0)
$\vdash (q_0, \text{abbb}, aZ_0)$
$\vdash (q_0, \text{bbb}, aaZ_0)$
$\vdash (q_1, \text{bb}, aZ_0)$
$\vdash (q_1, \text{b}, Z_0)$

String **aabbb** is rejected as q_1 is not final state and string **aabbb** is not read completely.

Acceptance by PDA using null store or empty store or empty stack

Move no	State	input	stack symbol	Move
1	q_0	a	Z_0	(q_0,aZ_0)
2	q_0	a	a	(q_0,aa)
3	q_0	b	a	$(q_1, \in)$
4	q_1	b	a	$(q_1, \in)$
5	q_1	$\in$	Z_0	$(q_1, \in)$

For the string aabb

$(q_0, aabb , \mathbf{Z_0})$
$\vdash (q_0, abb , a\mathbf{Z_0})$
$\vdash (q_0, bb , aa\mathbf{Z_0})$
$\vdash (q_1, b , a\mathbf{Z_0})$
$\vdash (q_1, \in , \mathbf{Z_0})$
$\vdash (q \quad \in \quad \in)$

String **aabb** is accepted, as stack is empty on reading string **aabb** completely

Rejection by PDA

Move no	**State**	**input**	**stack symbol**	**Move**
1	q_0	a	Z_0	(q_0, aZ_0)
2	q_0	a	a	(q_0, aa)
3	q_0	b	a	$(q_1, \in)$
4	q_1	b	a	$(q_1, \in)$
5	q_1	$\in$	Z_0	$(q_1, \in)$

For the string aabbb

$(q_0, aabbb, Z_0)$

$\vdash (q_0, abbb, aZ_0)$

$\vdash (q_0, bbb, aaZ_0)$

$\vdash (q_1, bb, aZ_0)$

$\vdash (q_1, b, Z_0)$

String **aabbb** is rejected as string **aabbb** is not read completely and stack is not empty.

Acceptance by PDA

Acceptance of input strings by PDA can be defined in terms of **final states** or in terms of **PDS(pushdown store).**

Let A= $(Q,\Sigma, \Gamma ,\delta,q_0,Z_0,F)$ be a PDA.

The set accepted by final state is defined by

$T(A) =\{w \in \Sigma^* \mid (q_0,w,Z_0) \vdash^* (q_f, \Lambda, \alpha)$ for some $q_f \in F$ and $\alpha \in \Gamma^*\}$

The set accepted by null store(or empty store)is defined by

$N(A) = \{w \in \Sigma^* \mid (q_0,w,Z_0) \vdash^* (q, \Lambda, \Lambda)$ for some $q \in Q\}$

PDA for $L=\{a^n b^n \mid n>0\}$

Logic:

W

Let

$A=(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

$Q=\{q_0,q_1\}$,

$\Sigma=\{a,b\}$,

$\Gamma=\{a,Z_0\}$,

$F=\{q_1\}$

is a PDA.

δ (Transition Function) by **Final State is**

Move no	**State**	**input**	**stack symbol**	**Move**
1	q_0	a	Z_0	(q_0,aZ_0)
2	q_0	a	a	(q_0,aa)
3	q_0	b	a	$(q_1, \in)$
4	q_1	b	a	$(q_1, \in)$
5	q_1	$\in$	Z_0	$\mathbf{(q_1, Z_0)}$

Acceptance of a string aabb

$(q_0, aabb, Z_0)$

$\vdash (q_0, abb, aZ_0)$

$\vdash (q_0, bb, aaZ_0)$

$\vdash (q_1, b, aZ_0)$

$\vdash (q_1, \in, Z_0)$

δ (Transition Function) by **Empty stack or Empty store or Null stack** is

Move no	**State**	**input**	**stack symbol**	**Move**
1	q_0	a	Z_0	(q_0, aZ_0)
2	q_0	a	a	(q_0, aa)
3	q_0	b	a	$(q_1, \in)$
4	q_1	b	a	$(q_1 \in)$
5	q_1	$\in$	Z_0	$(q_1, \in)$

Acceptance of a string aabb

$(q_0, aabb, Z_0)$

$\vdash (q_0, abb, aZ_0)$

$\vdash (q_0, bb, aaZ_0)$

$\vdash (q_1, b, aZ_0)$

$\vdash (q_1, \in, Z_0)$

$\vdash (q_1, \in, \in)$

Ex: PDA to accept language of palindromes with the marker. i.e. $L=\{xcx^r \mid x \in \{a,b\}^*\}$

Let

$A= (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ is a PDA.

where

$Q=\{q_0,q_1,q_f\}$,

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

$\Gamma=\{a,b,Z_0\}$,

$F=\{q_f\}$

δ (Transition Function) by **Final State** is

Move no	**State**	**input**	**stack symbol**	**Move**
1	q_0	a	Z_0	(q_0,aZ_0)
2	q_0	b	Z_0	(q_0,bZ_0)
3	q_0	a	a	(q_0,aa)
4	q_0	b	b	(q_0,bb)
5	q_0	a	b	(q_0, ab)
6	q_0	b	a	(q_0, ba)
7	q_0	c	Z_0	(q_1,Z_0)
8	q_0	c	a	(q_1,a)
9	q_0	c	b	(q_1,b)
10	q_1	a	a	$(q_1, \in)$
11	q_1	b	b	$(q_1, \in)$
12	q_1	$\in$	Z_0	$\boldsymbol{(q_f, Z_0)}$

δ (Transition Function) by **Empty stack or Empty store or Null stack** is

Move no	**State**	**input**	**stack symbol**	**Move**
1	q_0	a	Z_0	(q_0,aZ_0)
2	q_0	b	Z_0	(q_0,bZ_0)
3	q_0	a	a	(q_0,aa)
4	q_0	b	b	(q_0,bb)
5	q_0	a	b	(q_0, ab)
6	q_0	b	a	(q_0, ba)
7	q_0	c	Z_0	(q_1,Z_0)
8	q_0	c	a	(q_1,a)
9	q_0	c	b	(q_1,b)
10	q_1	a	a	$(q_1, \in)$
11	q_1	b	b	$(q_1, \in)$
12	q_1	$\in$	Z_0	$\mathbf{(q_1, \in)}$

Examples for practice

1. PDA for $L=\{a^n b^{2n} \mid n>0\}$
2. PDA for $L=\{a^n b^n c^m d^m \mid n, m>0\}$
3. PDA for $L=\{a^m b^n \mid m>n>=1\}$

Deterministic and non-deterministic PDA

DPDA:

transition function is :

Q X Σ X Γ ✉ Q X Γ*

e.g. δ(q,a,Z) is either empty or a singleton.

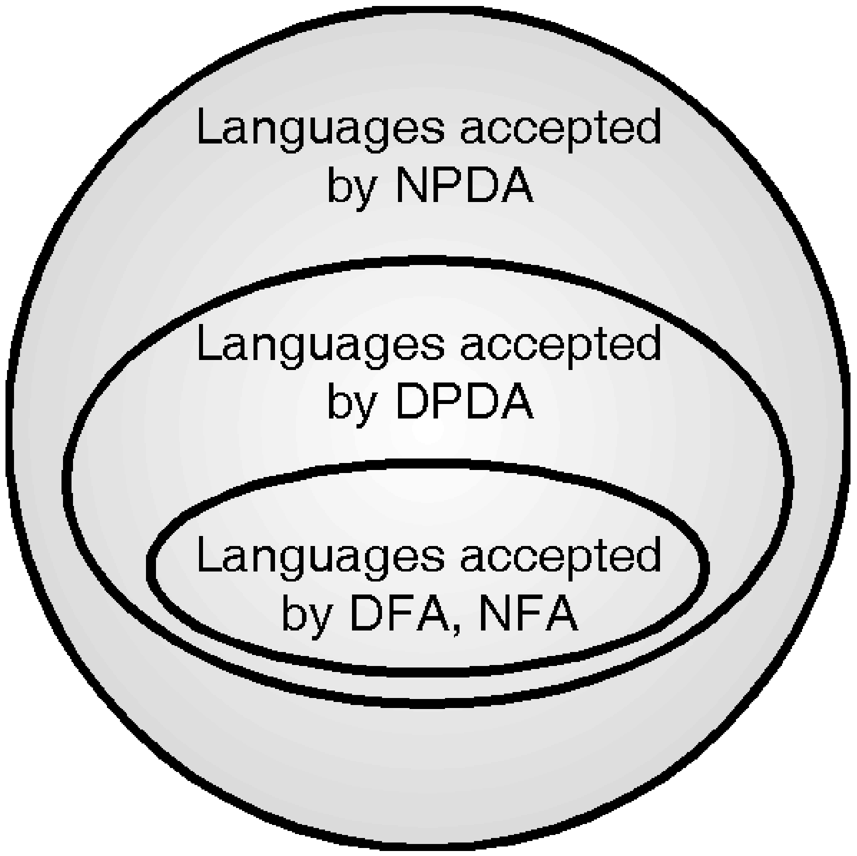
$\delta(q,a,Z) \neq \emptyset$

NPDA:

Q X Σ U {ϵ } X Γ ✉ finite subsets of Q X Γ*

e.g. δ(q,a,Z) = {(p1,ɤ1) ,(p2, ɤ2)…..(pm, ɤm)}

DPDA and NPDA


Languages accepted by NPDA
Languages accepted by DPDA
Languages accepted by DFA, NFA

NPDA and DPDA

- For every NPDA, there may not exist an equivalent DPDA.

- The NPDA can accept any CFL, while DPDA is a special case of NPDA that accepts only a subset of the CFLs accepted by the NPDA.

- Thus, DPDA is less powerful than NPDA.

NPDA to accept language of palindromes without the marker.

Let

A= (Q, Σ, Γ , δ, q_0, Z_0, F) is a PDA.

where

Q={q_0,q_1,q_f},

Σ={a,b},

Γ={a,b,Z_0},

F={q_f}

NPDA to accept language of all palindrome strings

Move no	**State**	**input**	**stack symbol**	**Move**
1	q_0	a	Z_0	$\{(q_0,aZ_0), (q_1, Z_0)\}$
2	q_0	b	Z_0	$\{(q_0,bZ_0), (q_1, Z_0)\}$
3	q_0	a	a	$\{(q_0,aa),(q_1,a)\}$
4	q_0	b	a	$\{(q_0,ba), (q_1,a)\}$
5	q_0	a	b	$\{(q_0, ab), (q_1,b)\}$
6	q_0	b	b	$\{(q_0, bb), (q_1,b)\}$
7	q_0	$\in$	Z_0	$\{(q_1,Z_0)\}$
8	q_0	$\in$	a	$\{(q_1,a)\}$
9	q_0	$\in$	b	$\{(q_1,b)\}$
10	q_1	a	a	$\{(q_1, \in)\}$
11	q_1	b	b	$\{(q_1, \in)\}$
12	q_1	$\in$	Z_0	$\{(q_f, Z_0)\}$

CFG to PDA

Theorem: **If L is a CFL then we can construct a PDA A accepting L by empty store ie. L=N(A).**

Proof: We construct A by making use of productions in G.

Let L=L(G) where G=(V, T, P, S) is a CFG.

We construct PDA A as

A= $(Q, \Sigma, \Gamma, \delta, q, Z_0, F)$

where $\Sigma = T$

Γ is (V U T)

$Z_0 = S$

$F = \Phi$

δ is defined as

R_1 : $\delta(q, \in, A) = \{(q, \alpha) \mid A$ ✉ α is in P$\}$

R_2 : $\delta(q, a, a) = \{(q, \in)\}$ for every a in Σ.

CFG to PDA

1 .Construct a PDA for the CFG

S✉ 0BB

B✉0S | 1S |0

Test whether 010^4 is in N(A).

We construct PDA A as

$A = (Q, \Sigma, \Gamma, \delta, q, Z_0, F)$

$Q = \{q\}$

$\Sigma = \{0,1\}$

$\Gamma = \{S, B, 0, 1\}$

$Z_0 = S$

$F = \Phi$

Move no	**State**	**input**	**stack symbol**	**Move**
1	q	$\in$	S	$\{(q,0BB)\}$
2	q	$\in$	B	$\{(q,0S), (q,1S), (q,0)\}$
3	q	0	0	$\{(q, \in)\}$
4	q	1	1	$\{(q, \in)\}$

CFG to PDA

1 .Construct a PDA for the CFG

S✉ 0BB

B✉0S | 1S |0

Test whether 010^4 is in N(A).

2. Convert the grammar

S✉aSb | A

A✉bSa |S| ∈

To a PDA that accepts the same language by empty stack.

Thank You…!!!