Dr. Muhammad Zeeshan Muzaffar **email id: Zeeshan@biit.edu.pkWhatsapp# 0331-5824043**
Dr. Naseer Ahmad Sajid **email id: Naseer@biit.edu.pkWhatsapp# 0346-5100010**

(Week15)Lecture29-30

Objectives: **Learningobjectivesofthislectureare**

- **A New Format for FA**
  - **New Terminologies**
  - **Input Tape parsing**
  - **New Symbols for FA**
  - **Examples**
- **Adding a Push Down Stack to make Push Down Automata**
  - **Examples**

TextBook&Resources:IntroductiontoComputerTheory–2nd Edition-(I.O.Cohen)

Video Recources link

Dr. Muhammad Zeeshan Muzaffar **email id: Zeeshan@biit.edu.pkWhatsapp# 0331-5824043**
Dr. Naseer Ahmad Sajid **email id: Naseer@biit.edu.pkWhatsapp# 0346-5100010**

A new Format for FAs

- We will start with our old FA's and throw in some new diagrams that will augment them and make them more powerful.
- In this chapter complete new designs will be created for modeling FA's.

New terminologies

- We call it INPUT TAPE to the part of the FA where the input string lives while it is being run.
- The INPUT TAPE must be long enough for any possible input, and since any word in a* is a possible input, the TAPE must be infinitely long.
- The TAPE has a first location for the first letter of the input, then a second location, and so on.
- Therefore, we say that the TAPE is infinite in one direction only.

A new Format

- The locations into which put the input letters are called cells. (See table below)

a	**a**	**b**	**Δ**	**Δ**	

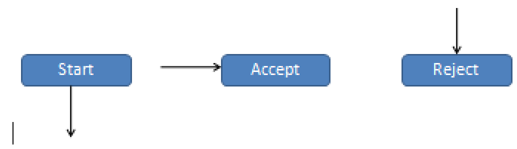
- Name the cells with lowercase Roman numerals.
- The Δ used to indicate the blank
- Input string is aab

Input tape parsing

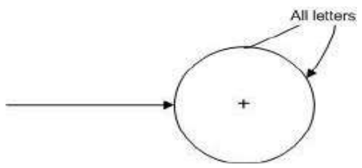
- As TAPE is processed, on the machine we read one letter at a time and eliminate each as it is used.
- When we reach the first blank cell we stop.
- We always presume that once the first blank is encountered the rest of the TAPE is also blank.
- We read from left to right and never go back to a cell that was read before.

New symbols for FA

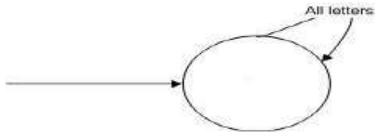
- To streamline the design of the machine, some symbols are used.
- The arrows (directed edges) into or out of these states can be drawn at any angle. The START state is like a state connected to another state in a TG by a Λ edge.
- We begin the process there, but we read no input letter. We just proceed immediately to the next state.
- A start state has no arrows coming into it.



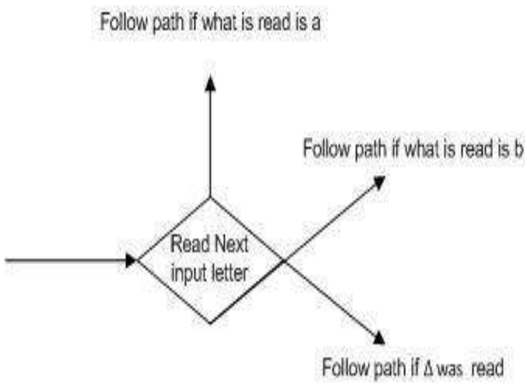
- An ACCEPT state is a shorthand notation for a dead-end final state-once entered, it cannot be left, shown:

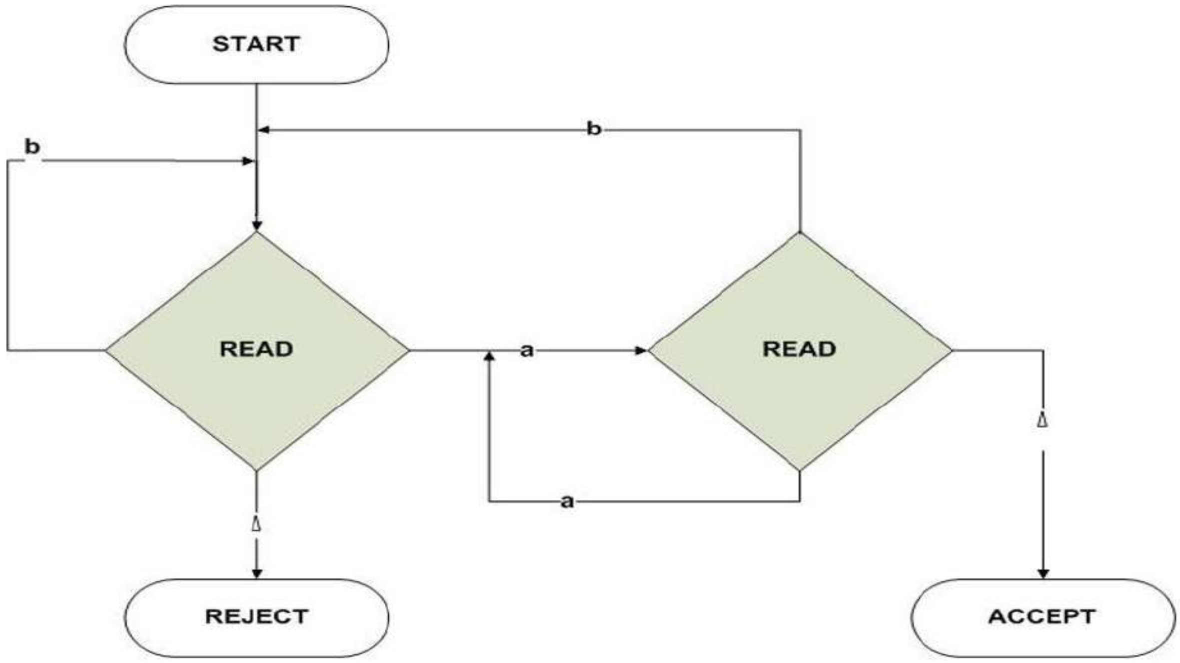


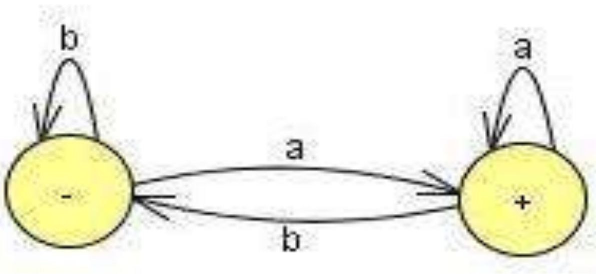
- A REJECT state is a dead-end state that is not final



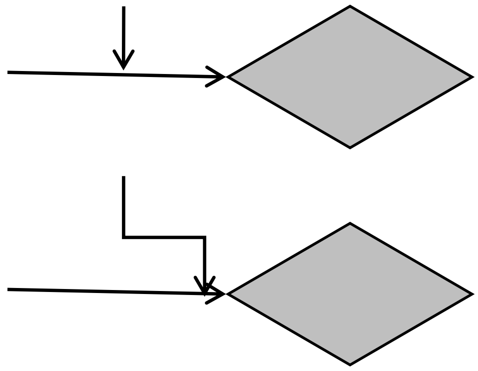
- READ states are introduced.
- These are depicted as diamond shaped boxes



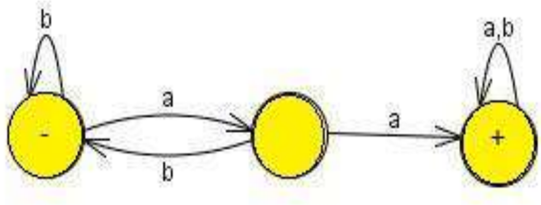




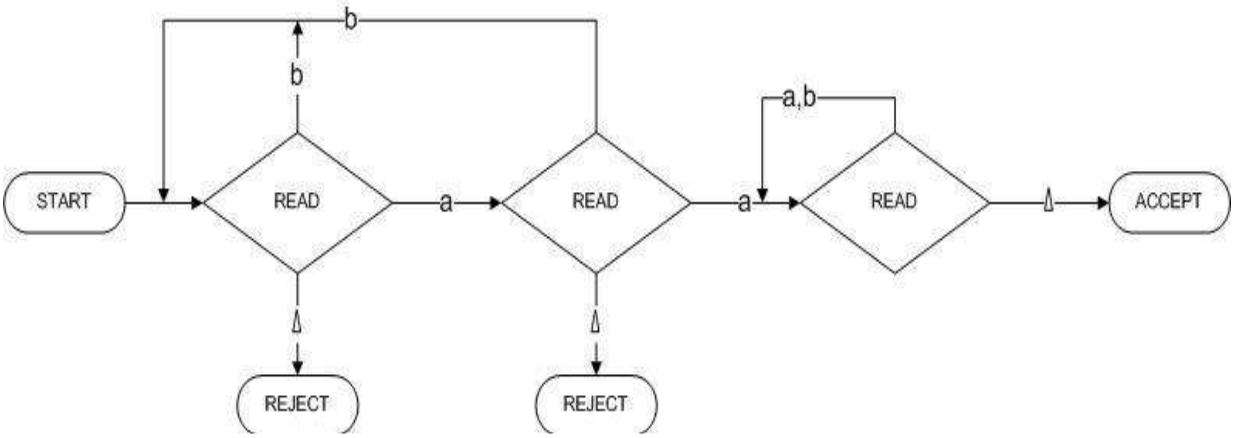
- We have also used the electronic diagram notation for wires flowing into each other.



- We have also used the electronic diagram notation for wires flowing into each other.



Becomes



Adding A Pushdown Stack

- A PUSHDOWN STACK is a place where input letters can be stored until we want to refer to them again.
- It holds the letters it has been fed in a long line. The operation PUSH adds a new letter to the line.
- The new letter is placed on top of the STACK, and all the other letters are pushed back (or down) accordingly.
- Before the machine begins to process an input string the STACK is presumed to be empty, which means that every storage location in it initially contains a blank.
- If the STACK is then fed the letters a, b, c, d by this sequence of instructions:

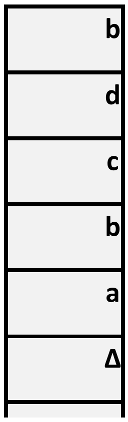
  PUSH a

  PUSH b

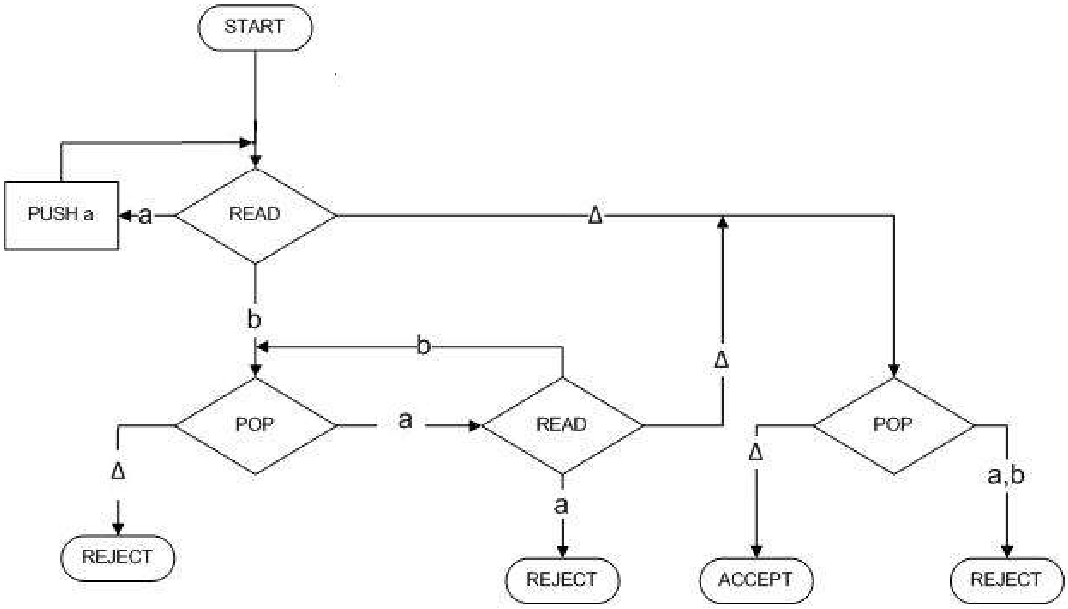
  PUSH c

  PUSH d
- Then top letter in the STACK is d, the second is c, the third is b, and the fourth is a.
- If we now execute the instruction:
- PUSH b the letter b will be added to the STACK on the top. The d will be pushed down to position 2, the c to position 3, the other b to position 4, and the bottom a to position 5.
- One pictorial representation of a STACK with these letters in it is shown below.

Beneath the bottom a we presume that the rest of the STACK, which, like the INPUT TAPE, has infinitely many storage locations, holds only blanks.



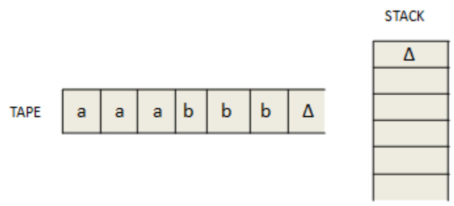
- How following PDA is working:

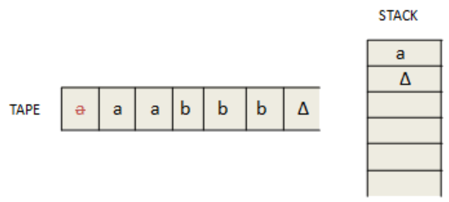


State	Stack	Tape
START	Δ ...	$aaabbb\ \Delta\Delta$
READ	Δ ...	$\cancel{a}aabbb\ \Delta\Delta$
PUSH a	$a\ \Delta$ ...	$\cancel{a}aabbb\ \Delta\Delta$
READ	$a\ \Delta$ ...	$\cancel{a}\cancel{a}abbb\ \Delta\Delta$

PUSH a	$aa\ \Delta\ \ldots$	$\cancel{aa}abbb\ \Delta\Delta$
READ	$aa\ \Delta\ \ldots$	$\cancel{aaa}bbb\ \Delta\Delta$
PUSH a	$aaa\ \Delta\ \ldots$	$\cancel{aaa}bbb\ \Delta\Delta$
READ	$aaa\ \Delta\ \ldots$	$\cancel{aaab}bb\ \Delta\Delta$
POP	$aa\ \Delta\ \ldots$	$\cancel{aaab}bb\ \Delta\Delta$
READ	$aa\ \Delta\ \ldots$	$\cancel{aaabb}b\ \Delta\Delta$
POP	$a\ \Delta\ \ldots$	$\cancel{aaabb}b\ \Delta\Delta$
READ	$a\ \Delta\ \ldots$	$\cancel{aaabbb}\ \Delta\Delta$
POP	$\Delta\Delta\ \ldots$	$\cancel{aaabbb}\ \Delta\Delta$
READ	$\Delta\Delta\ \ldots$	$\cancel{aaabbb\ \Delta}\Delta$
POP	$\Delta\ \ldots$	$\cancel{aaabbb\ \Delta}\Delta$

- Its operation on the input string aaabbb.
- We begin by assuming that this string has been put on the TAPE.

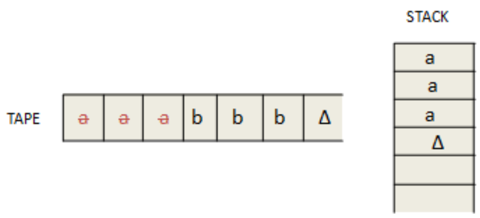




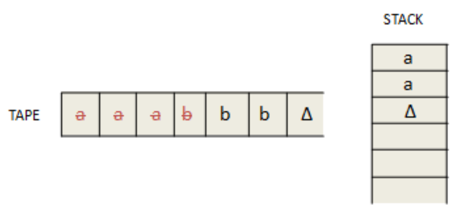
- We now read another a and proceed as before along the a edge to push it into the STACK.
- Again we are returned to the READ box.

Dr. Muhammad Zeeshan Muzaffar email id: Zeeshan@biit.edu.pk Whatsapp# 0331-5824043
Dr. Naseer Ahmad Sajid email id: Naseer@biit.edu.pk Whatsapp# 0346-5100010

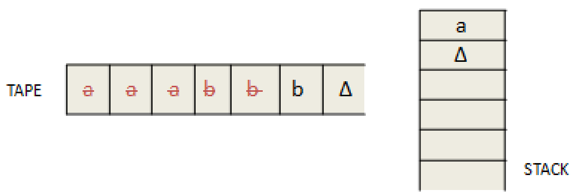
- Again we read an a (our third), and again this a is pushed onto the STACK.



- After the third PUSH a, we are routed back to the same READ state again.
- This time, we read the letter b.
- This means that we take the b edge out of this state down to the lower left POP.



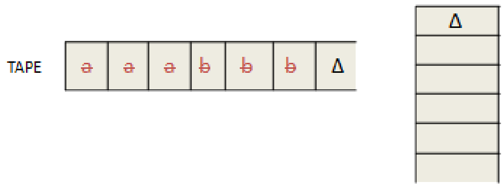
- The b road from the second READ state now takes us back to the edge feeding into the POP state.
- So we pop the STACK again and get another a.
- The STACK is now down to only one a.
- The a line from POP takes us again to this same READ.
- There is only one letter left on the input TAPE, a b



- We read it and leave the TAPE empty, hat is, all blanks. However, the machine does not yet know that the TAPE is empty.

Dr. Muhammad Zeshan Muzaffar **email id: Zeeshan@biit.edu.pkWhatsapp# 0331-5824043**
Dr. Naseer Ahmad Sajid **email id: Naseer@biit.edu.pkWhatsapp# 0346-5100010**

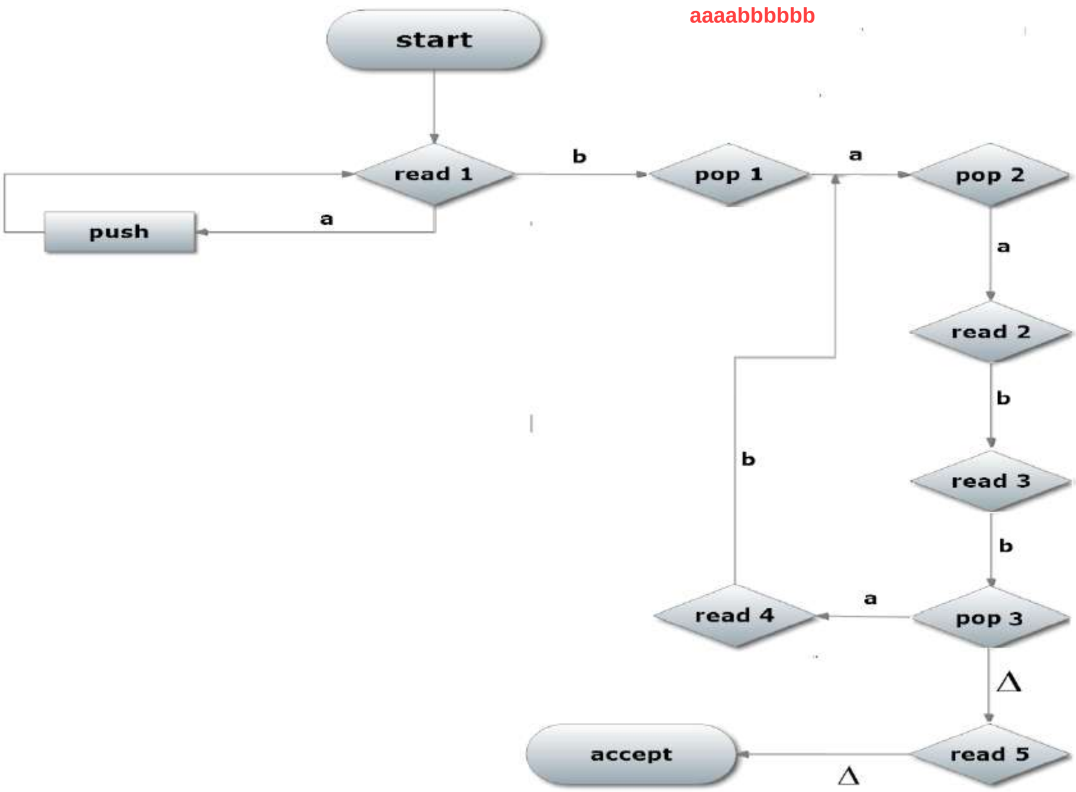
- It will discover this only when it next tries to read the TAPE and finds Δ.



Example

m=3,6,9,...

The language $a^n b^m$ where $n = 2,4,6 \ldots$ and $m = n + \frac{n}{2}$



For example if we pass the word *aabbb*, then

State	Stack	Tape
Start	$\Delta \ldots$	$aabbb\ \Delta\Delta \ldots$
read1	$\Delta \ldots$	$\cancel{a}abbb\ \Delta\Delta \ldots$

Dr. Muhammad Zeeshan Muzaffar **email id: Zeeshan@biit.edu.pkWhatsapp# 0331-5824043**
Dr. Naseer Ahmad Sajid **email id: Naseer@biit.edu.pkWhatsapp# 0346-5100010**

push a	*a* Δ ...	~~*a*~~*abbb* ΔΔ ...
read1	*a* Δ ...	~~*aa*~~*bbb* ΔΔ ...
push a	*aa* Δ ...	~~*aa*~~*bbb* ΔΔ ...
read1	*aa* Δ ...	~~*aab*~~*bb* ΔΔ ...
pop1	*a* Δ ...	~~*aab*~~*bb* ΔΔ ...
pop2	Δ ...	~~*aab*~~*bb* ΔΔ ...
read2	Δ ...	~~*aabb*~~*b* ΔΔ ...
read3	Δ ...	~~*aabbb*~~ ΔΔ ...
pop3	Δ ...	~~*aabbb*~~ ΔΔ ...
read5	Δ ...	~~*aabbb* Δ~~Δ ...
accept	Δ ...	~~*aabbb* Δ~~Δ ...

Example:

Consider the following CFG in CNF:

S → SB | AB

A → CC

B → b

C → a

We now propose the following nondeterministic PDA where the STACK alphabet is

Γ = {S, A, B, C}

and the TAPE alphabet is only

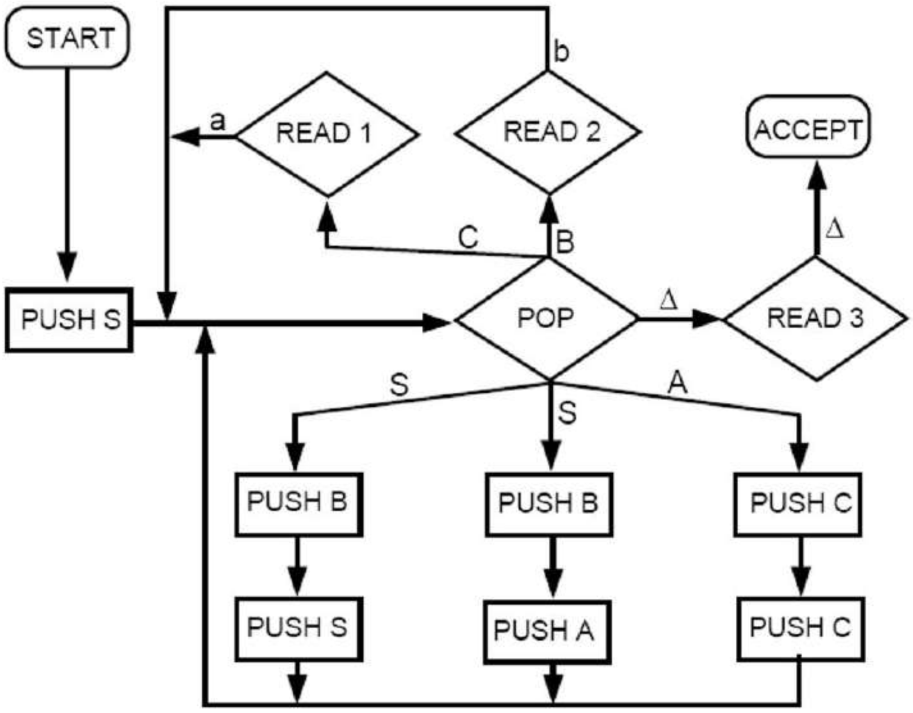
Dr. Muhammad Zeeshan Muzaffar email id: Zeeshan@biit.edu.pk Whatsapp# 0331-5824043
Dr. Naseer Ahmad Sajid email id: Naseer@biit.edu.pk Whatsapp# 0346-5100010

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

Solution

S->SB|AB
A->CC
B->b
C->a

aabbb



S->SB
->SBB
->ABBB
->CCBBB
->CBBB
->BBB
->BB
->B
->delta

aabbb(delta)

To verify the solution, let pass some word from this PDA, then we can see that

$$S \Rightarrow SB$$

$$S \Rightarrow SBB$$

$$S \Rightarrow ABBB$$

$$S \Rightarrow CCBBB$$

$$S \Rightarrow aabbb$$

Now according to given CFG, word *aabbb* belongs to the CFG

State	Stack	Tape
START	Δ ...	*aabbb* ΔΔ ...
PUSH S	S Δ ...	*aabbb* ΔΔ ...
POP	Δ ...	*aabbb* ΔΔ ...

Dr. Muhammad Zeeshan Muzaffar **email id: Zeeshan@biit.edu.pk Whatsapp# 0331-5824043**
Dr. Naseer Ahmad Sajid **email id: Naseer@biit.edu.pk Whatsapp# 0346-5100010**

PUSH B	B△ ...	*aabbb* △△ ...
PUSH S	SB△ ...	*aabbb* △△ ...
POP	B△ ...	*aabbb* △△ ...
PUSH B	BB△ ...	*aabbb* △△ ...
PUSH S	SBB△ ...	*aabbb* △△ ...
POP	BB△ ...	*aabbb* △△ ...
PUSH B	BBB△ ...	*aabbb* △△ ...
PUSH A	ABBB△ ...	*aabbb* △△ ...
POP	BBB△ ...	*aabbb* △△ ...
PUSH C	CBBB△ ...	*aabbb* △△ ...
PUSH C	CCBBB△ ...	*aabbb* △△ ...
POP	CBBB△ ...	*aabbb* △△ ...
READ 1	CBBB△ ...	*~~a~~abbb* △△ ...
POP	BBB△ ...	*~~a~~abbb* △△ ...
READ 1	BBB△ ...	*~~aa~~bbb* △△ ...
POP	BB△ ...	*~~aa~~bbb* △△ ...
READ 2	BB△ ...	*~~aab~~bb* △△ ...
POP	B△ ...	*~~aab~~bb* △△ ...
READ 2	B△ ...	*~~aabb~~b* △△ ...

Dr. Muhammad Zeeshan Muzaffar **email id: Zeeshan@biit.edu.pkWhatsapp# 0331-5824043**
Dr. Naseer Ahmad Sajid **email id: Naseer@biit.edu.pkWhatsapp# 0346-5100010**

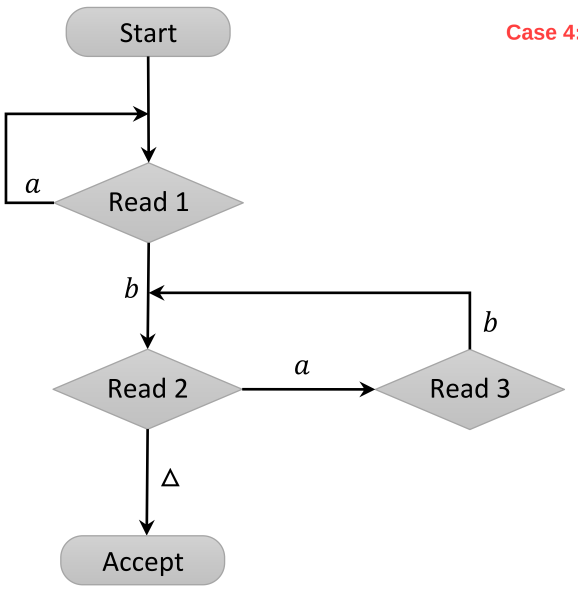
POP	$\Delta \ldots$	$\cancel{aabbb}\ \Delta\Delta \ldots$
READ 2	$\Delta \ldots$	$\cancel{aabbb}\ \Delta\Delta \ldots$
POP	$\Delta \ldots$	$\cancel{aabbb}\ \Delta\Delta \ldots$
READ 3	$\Delta \ldots$	$\cancel{aabbb\ \Delta}\Delta \ldots$
ACCEPT	$\Delta \ldots$	$\cancel{aabbb\ \Delta}\Delta \ldots$

Example

The language $a^n b(ab)^m$ where $n, m \geq 0$

Solution

Case 1: n=2, m=0 aab
Case 2: n=0, m=2 babab
Case 3: n=0, m=0 b
Case 4: n=2, m=3 aabababab



Now let suppose we want to pass the word *aabababab* to check our solution, then

State	**Stack**	**Tape** n=2, m=3
Start	$\Delta \ldots$	$aabababab\ \Delta\Delta \ldots$

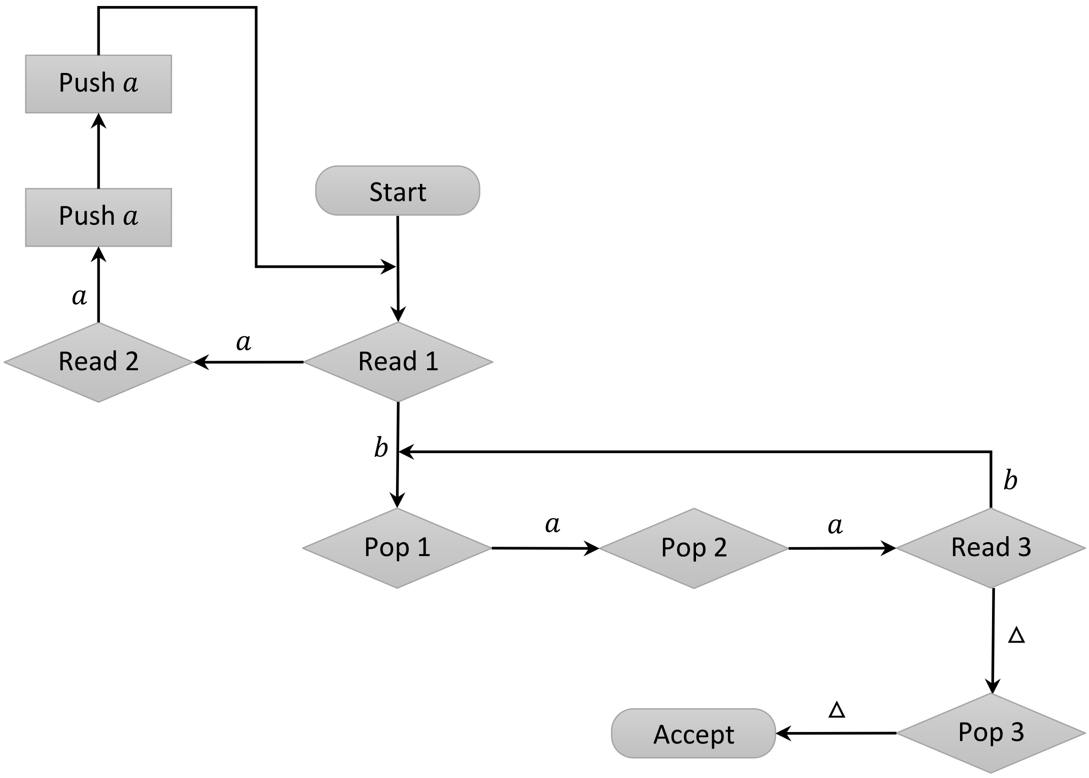
Read1	Δ ...	~~*a*~~*abababab* ΔΔ ...
Read 1	Δ ...	~~*aa*~~*bababab* ΔΔ ...
Read 1	Δ ...	~~*aab*~~*ababab* ΔΔ ...
Read 2	Δ ...	~~*aaba*~~*babab* ΔΔ ...
Read 3	Δ ...	~~*aabab*~~*abab* ΔΔ ...
Read 2	Δ ...	~~*aababa*~~*bab* ΔΔ ...
Read 3	Δ ...	~~*aababab*~~*ab* ΔΔ ...
Read 2	Δ ...	~~*aabababa*~~*b* ΔΔ ...
Read 3	Δ ...	~~*aabababab*~~ ΔΔ ...
Read 2	Δ ...	~~*aabababab* Δ~~Δ ...
Accept	Δ ...	~~*aabababab* Δ~~Δ ...

Dr. Muhammad Zeeshan Muzaffar **email id: Zeeshan@biit.edu.pk Whatsapp# 0331-5824043**
Dr. Naseer Ahmad Sajid **email id: Naseer@biit.edu.pk Whatsapp# 0346-5100010**

Example

The language $a^{2n}b^n$ where $n > 0$

Solution



Now to check our solution, let the word of the given language taken is $aaaabb$, then this word will pass through the following criteria

State	**Stack**	**Tape**
Start	Δ ...	$aaaabb\ \Delta\Delta$ ...
Read1	Δ ...	$\cancel{a}aaabb\ \Delta\Delta$ ...
Read 2	Δ ...	$\cancel{aa}aabb\ \Delta\Delta$ ...
Push a	$a\ \Delta$ ...	$\cancel{aa}aabb\ \Delta\Delta$ ...
Push a	$aa\ \Delta$ ...	$\cancel{aa}aabb\ \Delta\Delta$ ...

Read 1	$aa\ \Delta$ ...	$\cancel{aa}aabb\ \Delta\Delta$ ...
Read 2	$aa\ \Delta$ ...	$\cancel{aaa}abb\ \Delta\Delta$ ...
Push a	$aaa\ \Delta$ ...	$\cancel{aaa}abb\ \Delta\Delta$ ...
Push a	$aaaa\ \Delta$ ...	$\cancel{aaa}abb\ \Delta\Delta$ ...
Read 1	$aaaa\ \Delta$ ...	$\cancel{aaaab}b\ \Delta\Delta$ ...
Pop 1	$aaa\ \Delta$ ...	$\cancel{aaaab}b\ \Delta\Delta$ ...
Pop 2	$aa\ \Delta$ ...	$\cancel{aaaab}b\ \Delta\Delta$ ...
Read 3	$aa\ \Delta$ ...	$\cancel{aaaabb}\ \Delta\Delta$ ...
Pop 1	$a\ \Delta$ ...	$\cancel{aaaabb}\ \Delta\Delta$ ...
Pop 2	Δ ...	$\cancel{aaaabb}\ \Delta\Delta$ ...
Read 3	Δ ...	$\cancel{aaaabb\ \Delta}\Delta$ ...
Pop 3	Δ ...	$\cancel{aaaabb\ \Delta}\Delta$ ...
Accept	Δ ...	$\cancel{aaaabb\ \Delta}\Delta$ ...

-:ASSIGNMENT:-

Question No. 1

Draw the PDA for the language $a^n b(ab)^m$ where $n, m \geq 1$

Question No. 2

Draw the PDA for the CFG

$$S \rightarrow aS \mid abX$$

$$X \rightarrow abX \mid ab$$