PUSHDOWN STACK or PUSHDOWN STORE or PUSHDOWN AUTOMATA

A new format for FAs

- A class of machines (FAs) has been discussed accepting the regular language *i.e.* corresponding to a regular language there is a machine in this class, accepting that language and corresponding to a machine of this class there is a regular language accepted by this machine.
- It has also been discussed that there is a CFG corresponding to regular language and CFGs *also define some nonregular languages, as well*

A new format for FAs contd. ...

- There is a question whether there is a class of machines accepting the CFLs?

  The answer is yes.

- The new machines which are to be defined are more powerful and can be constructed with the help of FAs with new format.
- To define the new format of an FA, the following are to be defined

A new format for FAs contd. ...

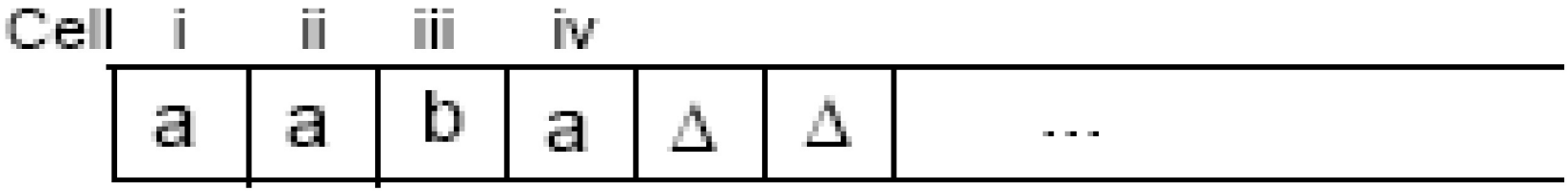
Input TAPE

- The part of an FA, where the input string is placed before it is run, is called the input TAPE.

- The input TAPE is supposed to accommodate all possible strings. The input TAPE is partitioned with cells, so that each letter of the input string can be placed in each cell.

- The input string abbaa is shown in the

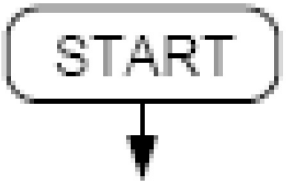
Input TAPE contd...



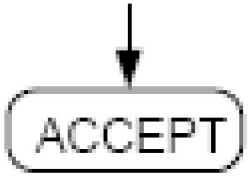
- The character Δ indicates a blank in the TAPE. The input string is read from the TAPE starting from the cell (i).
- It is assumed that when first Δ is read, the rest of the TAPE is supposed to be blank.

The START state

- **START**: The START state is like a - state in an FA. We begin the process here, but we read no input letter and go immediate[illegible]e next state.

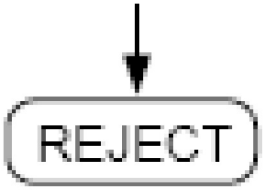


An Accept state



- **ACCEPT:** This state is like a dead-end final state + once entered, it cannot be left, such as

A REJECT state

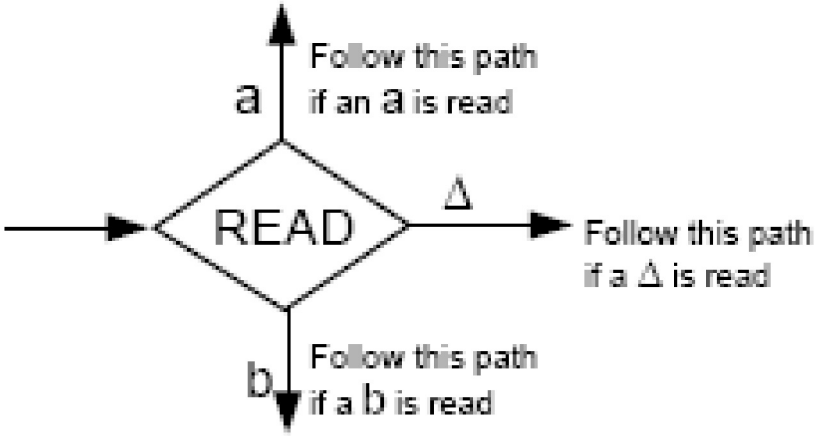


- **REJECT**: This state is like dead-end non final state such as

- **NOTE:** It may be noted that the ACCEPT and REJECT states are called the halt states. Halt states cannot be traversed (i.e., cannot be passed through like a final state in an FA).

READ states

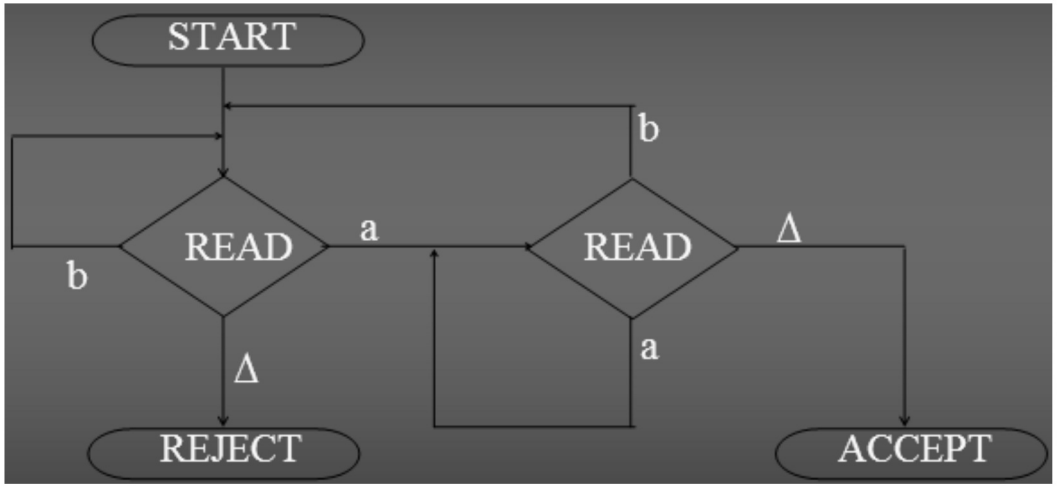
- READ states are depicted as diamond-shaped boxes as shown below:



- When Δ is read from the TAPE, it means that we are out of input letters. We are then finished processing the input string. Hence, the Δ-edge will lead to ACCEPT or REJECT.

Example

- Below is an FA that accepts all words ending with letter a, and its equivalent new picture

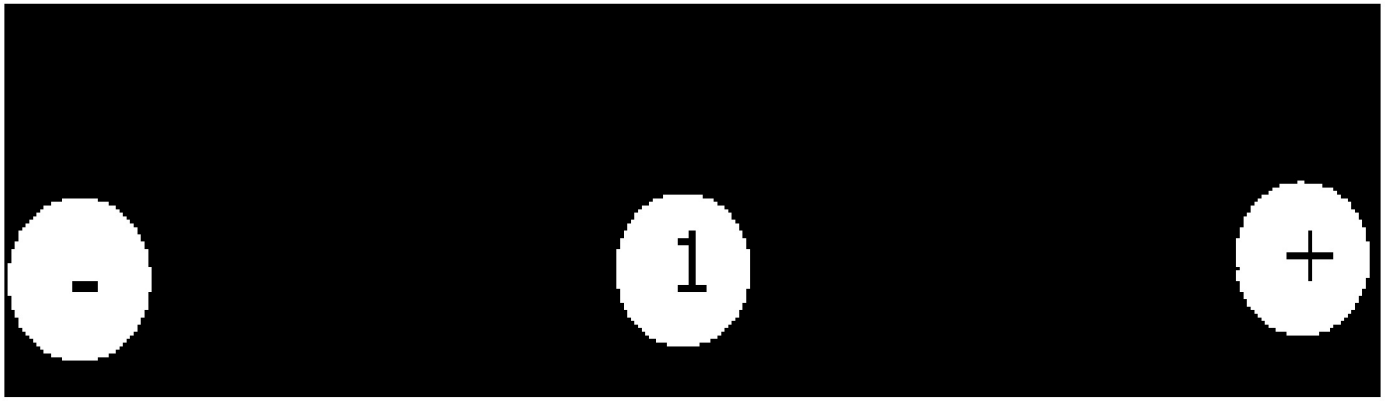


Note

- The Δ edge should not be confused with Λ -labeled edge. The Δ-edges start only from READ boxes and lead to halt states.

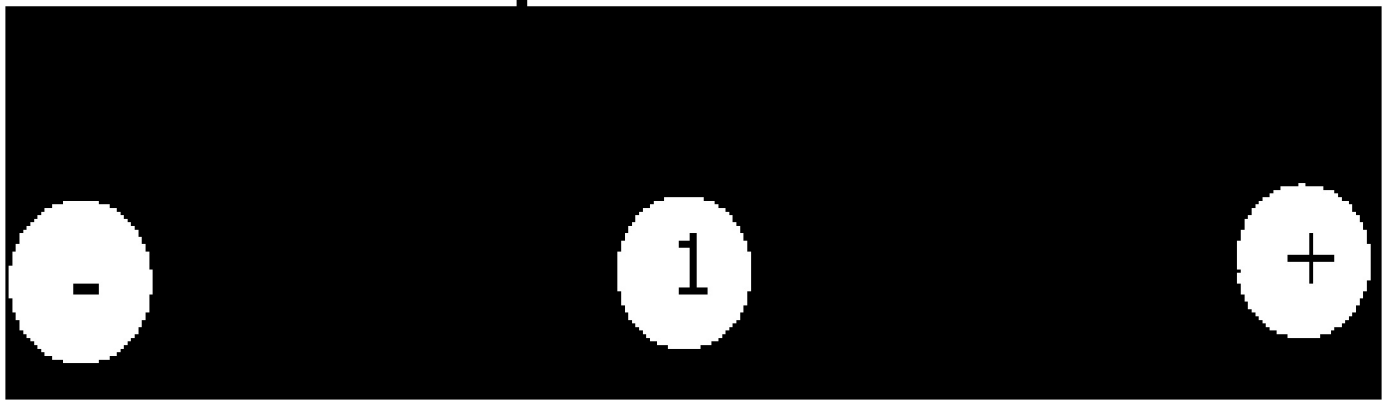
- Following is another example

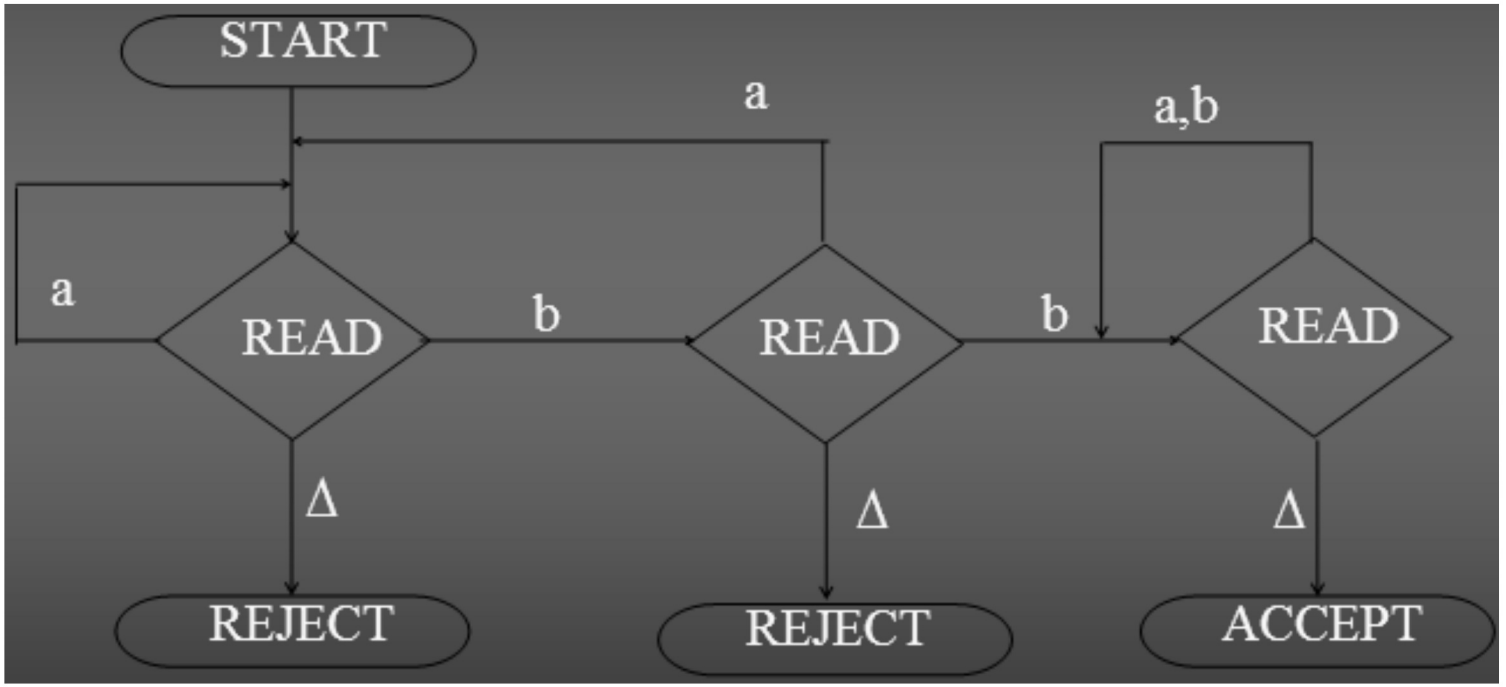
Example



- The above FA accepts the language expressed by (a+b)*bb(a+b)*

Example cont. ...

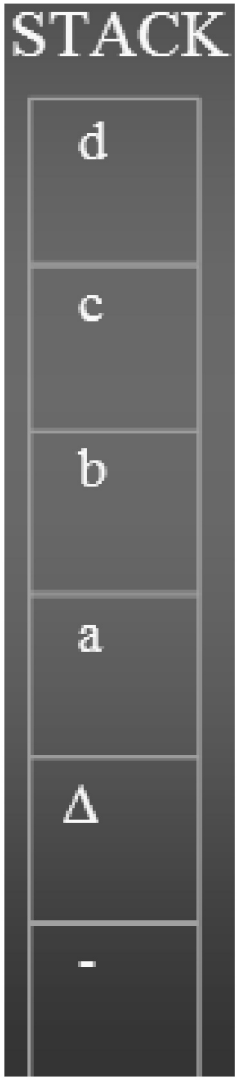




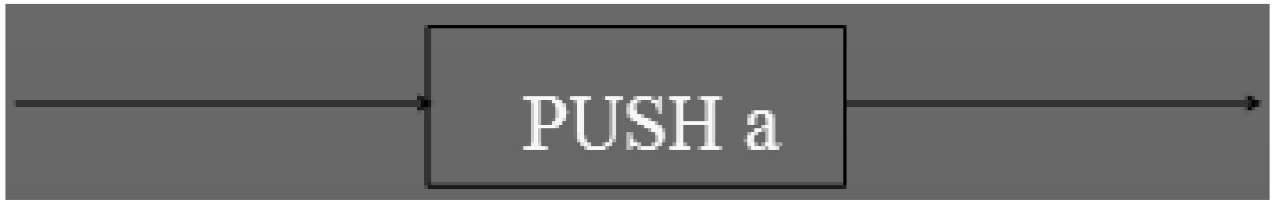
- **PUSHDOWN STACK:** PUSHDOWN STACK is a place where the input letters can be placed until these letters are referred again. It can store as many letters as one can in a long column. Initially the STACK is supposed to be empty *i.e.* each of its storage location contains a blank.

- **PUSH :** **A** PUSH operator adds a new letter at the top of STACK, for *e.g.* if the letters a, b, c and d are pushed to the STACK that was initially blank, the STACK

PUSH and STACK contd. ...



- The PUSH state is expressed by

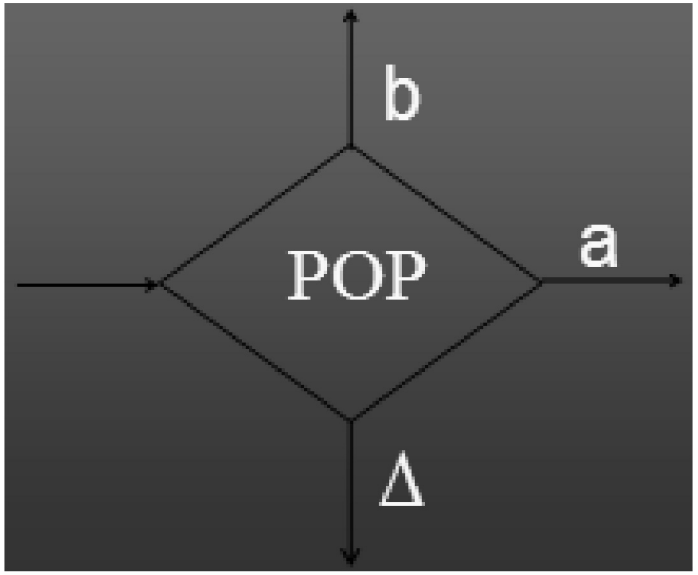


- When a letter is pushed, it replaces the existing letter and pushes it one position below.

POP and STACK

stack follow the prinicple of lifo

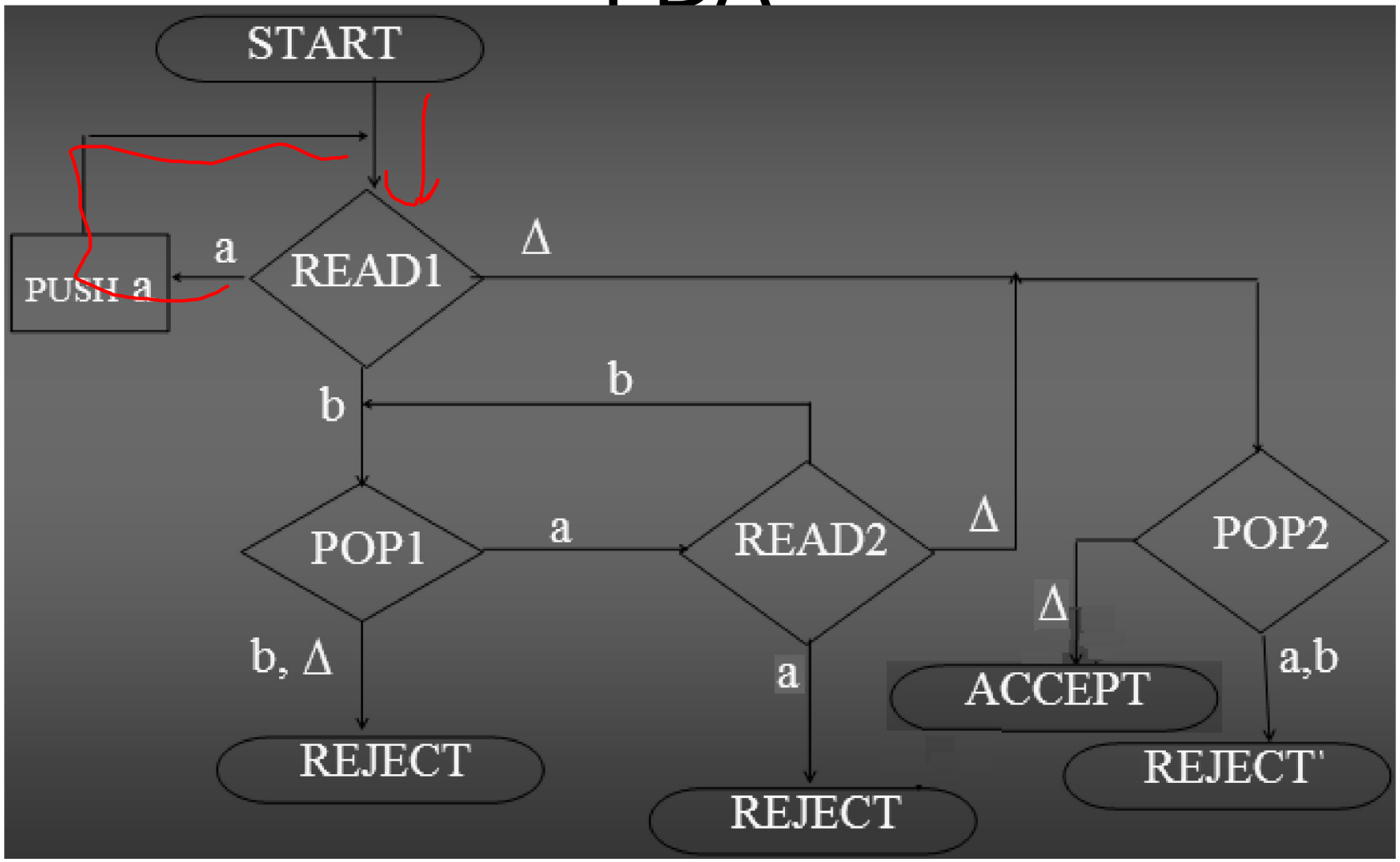
- **POP**: POP is an operation that takes out a letter from the top of the STACK. The rest of the letters are moved one location up. POP state is expressed as



Note

- It may be noted that popping an empty STACK is like reading an empty TAPE, *i.e.* popping a blank character Δ.
- It may also be noted that when the new format of an FA contains PUSH and POP states, it is called PUSHDOWN Automata or PDAs. It may be observed that if the PUSHDOWN STACK (the memory structure) is added to an FA then its language recognizing capabilities are increased considerably.
- Following is an example of PDA

Example: Consider the following PDA

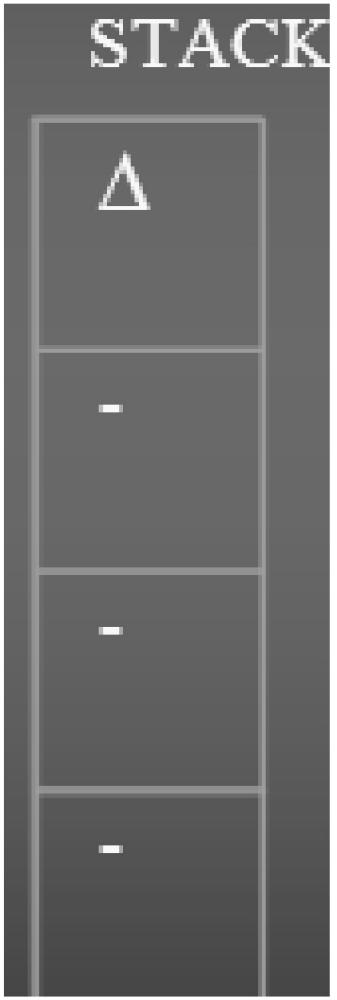


Example contd. ...

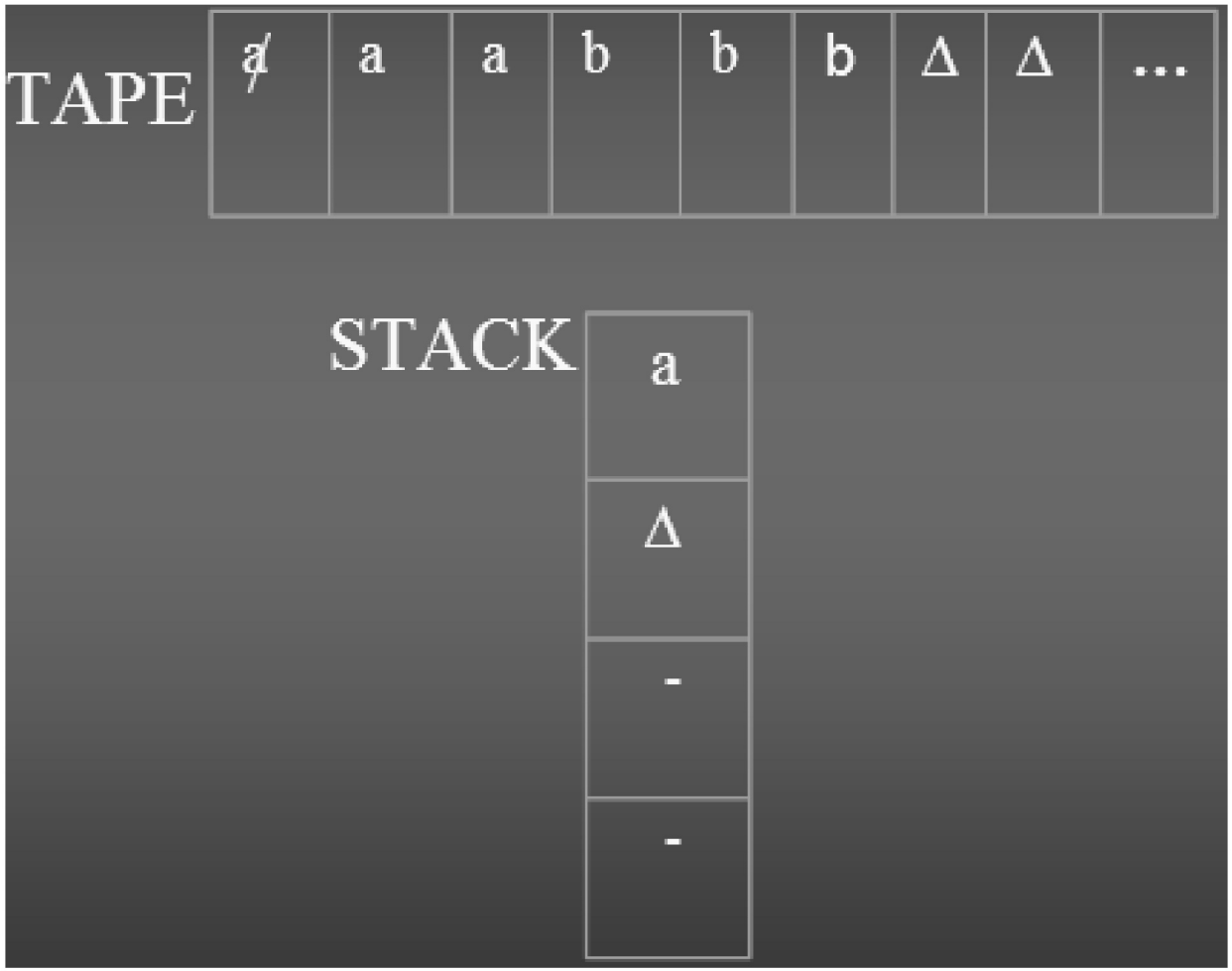
- The string **aaabbb** is to be run on this machine. Before the string is processed, the string is supposed to be placed on the TAPE and the STACK is supposed to be empty as shown below

Example contd. ...

- Reading first **a** from the TAPE we move from READ1 State to PUSH a state, it causes the letter a deleted from the TAPE and added to the top of the STACK, as shown below

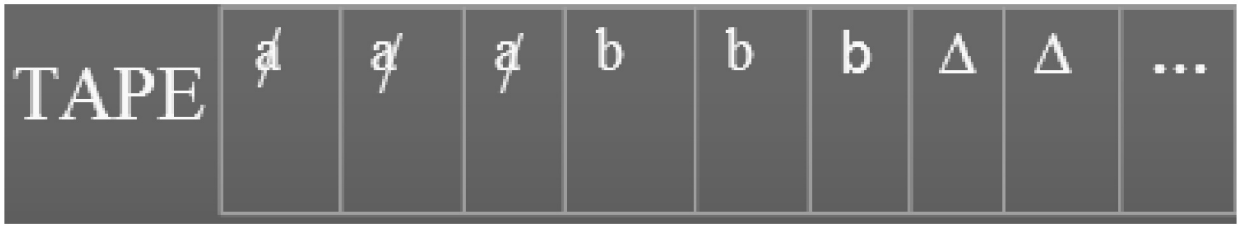


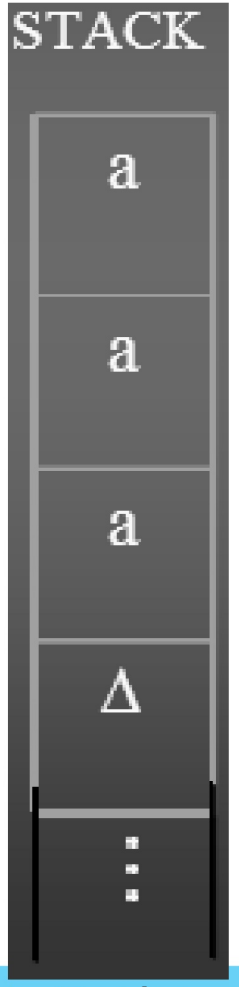
Example contd. ...



Example contd. …

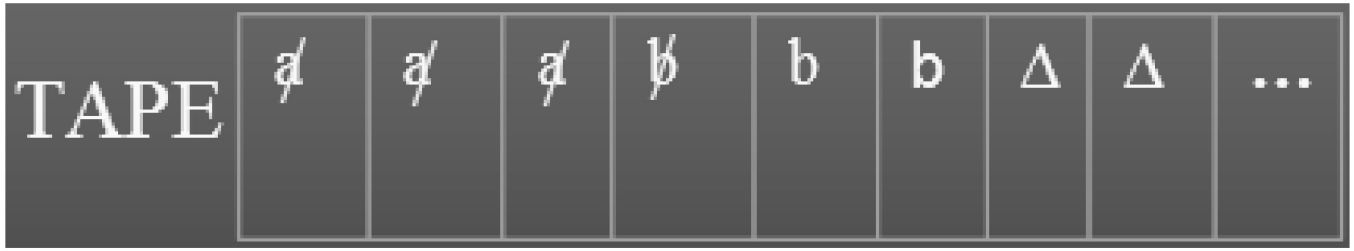
- Reading next two a's successively, will delete further two a's from the TAPE and add these letters to the top of the STACK, as shown below

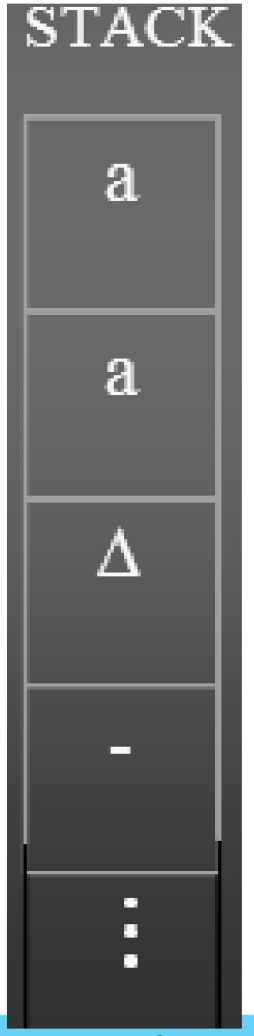




Example contd. ...

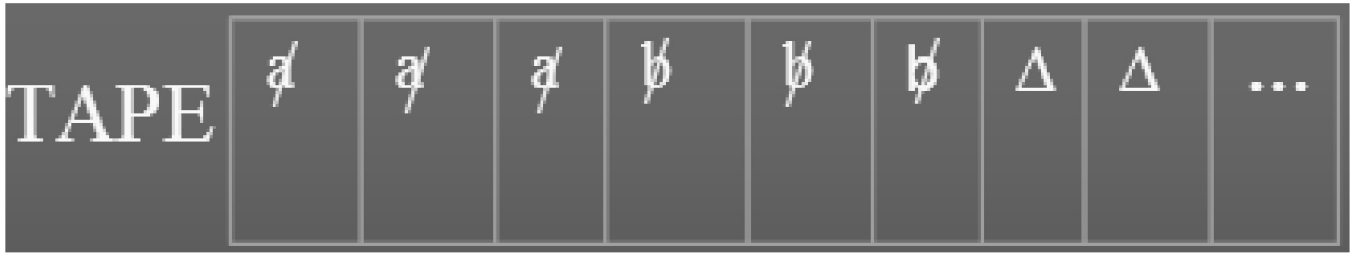
- Then reading the next letter which is b from the TAPE will lead to the POP1 state. The top letter at the STACK is a, which is popped out and READ2 state is entered. Situation of TAPE and STACK is shown below

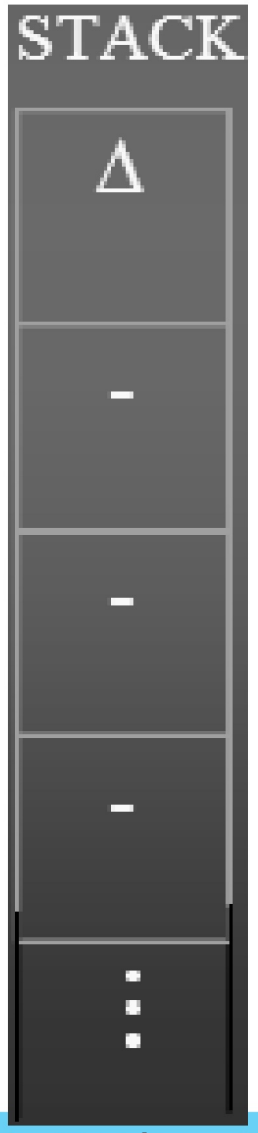




Example contd. …

- Reading the next two b's successively will delete two b's from the TAPE, will lead to the POP1 state and these b's will be removed from the STACK as shown below

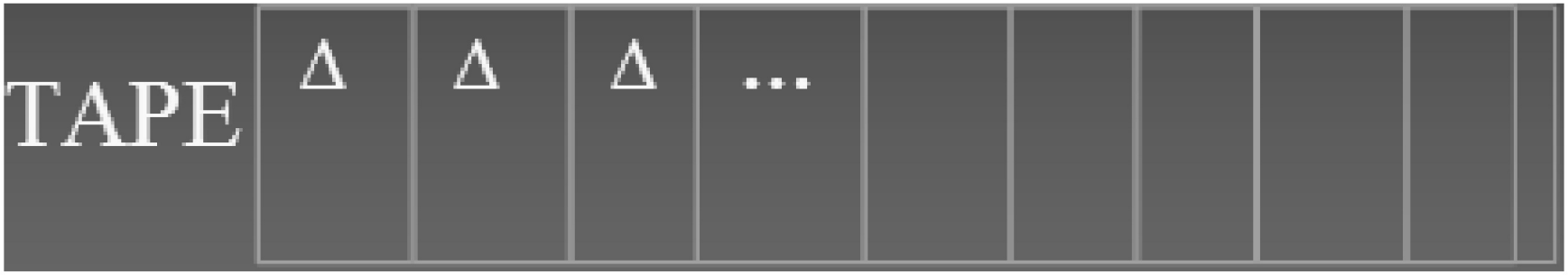




Example contd. ...

- Now there is only blank character Δ is left to be read from the TAPE, which leads to POP2 state. While the only blank characters is left in the STACK to be popped out and the ACCEPT state is entered, which shows that the string aaabbb is accepted by this PDA.
- It may be observed that the above PDA accepts the language {$\mathbf{a^n b^n : n=0,1,2, \ldots}$ }. Since the null string is like a blank character, so to determine how the null string is accepted, it can be placed in the TAPE as shown below

Example contd. ...



- Reading Δ at state READ1 leads to POP2 state and POP2 state contains only Δ, hence it leads to ACCEPT state and the null string is accepted.

Nondeterministic PDA

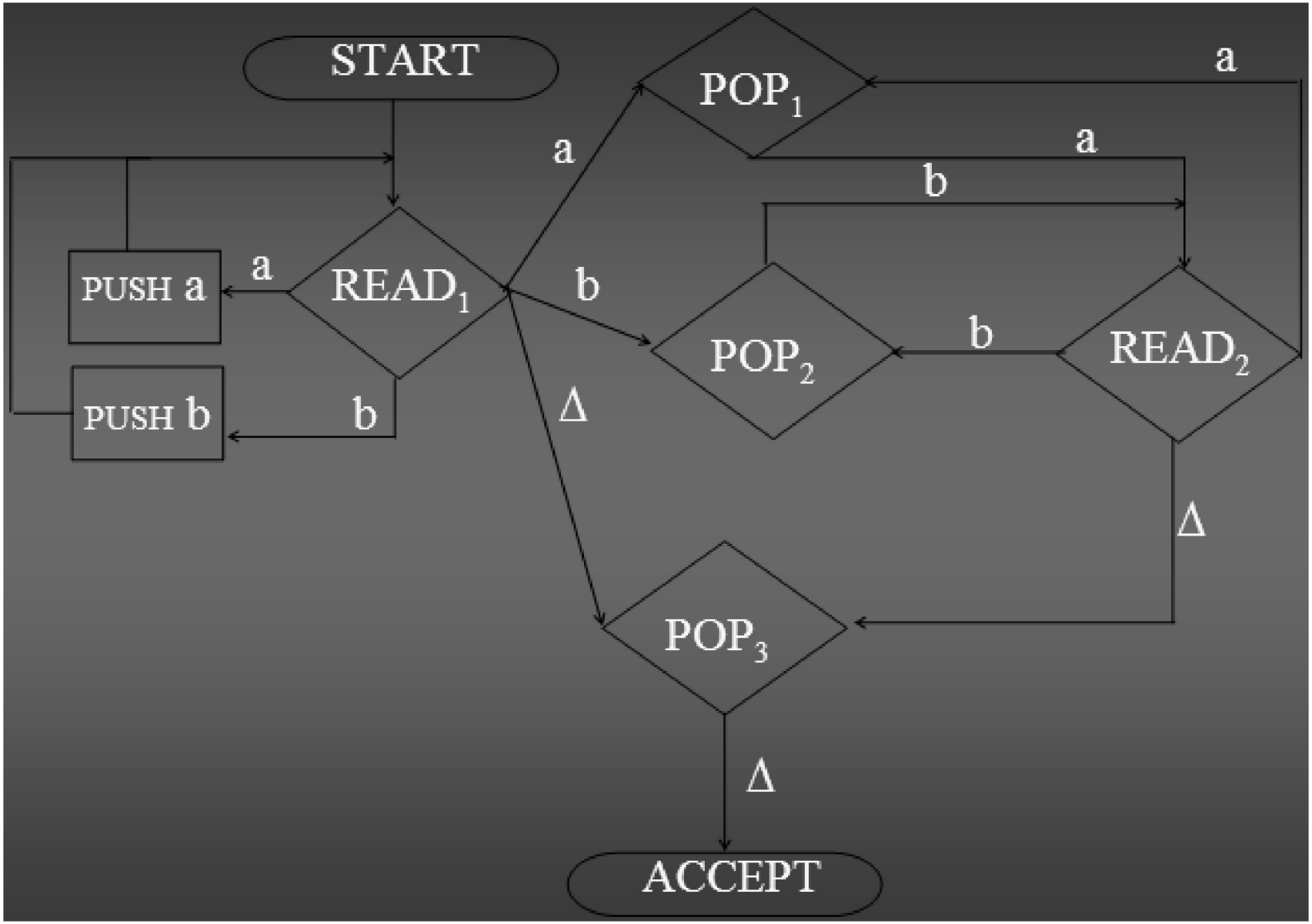
- Like TGs and NFAs, if in a PDA there are more than one outgoing edges at READ or POP states with one label, then it creates nondeterminism and the PDA is called nondeterministic PDA.

- In nondeterministic PDA no edge is labeled by string of terminals or nonterminals, like that can be observed in TGs. Also if there is no edge for any letter to be read from the TAPE, the machine crashes and the string is rejected.

Nondeterministic PDA cont ...

- In nondeterministic PDA a string may trace more than one paths. If there exists at least one path traced by a string leading to ACCEPT state, then the string is supposed to be accepted, otherwise rejected.

- Following is an example of nondeterministic PDA


START
POP1
a
a
a
b
PUSH a
a
READ1
b
POP2
b
READ2
PUSH b
b
Δ
Δ
POP3
Δ
ACCEPT

Nondeterministic PDA continued …

- Here the nondeterminism can be observed at state READ1. It can be observed that the above PDA accepts the language

EVENPALINDROME=

{Λ, aa, bb, aaaa, abba, baab, bbbb, …}

- Now the definition of PDA including the possibility of nondeterminism may be given as follows:

PUSHDOWN AUTOMATON (PDA)

Pushdown Automaton (PDA), consists of the following

1. An alphabet Σ of input letters.
2. An input TAPE with infinite many locations in one direction. Initially the input string is placed in it starting from first cell, the remaining part of the TAPE is empty.
3. An alphabet Γ of STACK characters.
4. A pushdown STACK which is initially empty, with infinite many locations in one direction. Initially the STACK contains blanks.

PUSHDOWN AUTOMATON (PDA)

Pushdown Automaton (PDA), consists of the following

1. An alphabet Σ of input letters.
2. An input TAPE with infinite many locations in one direction. Initially the input string is placed in it starting from first cell, the remaining part of the TAPE is empty.
3. An alphabet Γ of STACK characters.
4. A pushdown STACK which is initially empty, with infinite many locations in one direction. Initially the STACK contains blanks.

PDA Continued ...

5. One START state with only one out-edge and no in-edge.
6. Two halt states *i.e.* ACCEPT and REJECT states, with in-edges and no out-edges.
7. A PUSH state that introduces characters onto the top of the STACK.
8. A POP state that reads the top character of the STACK, (may contain more than one out-edges with same label).
9. A READ state that reads the next unused letter from the TAPE, (may contain more than one out-edges with same label).

Example

- Consider the CFG

  S → S+S | S*S | 4

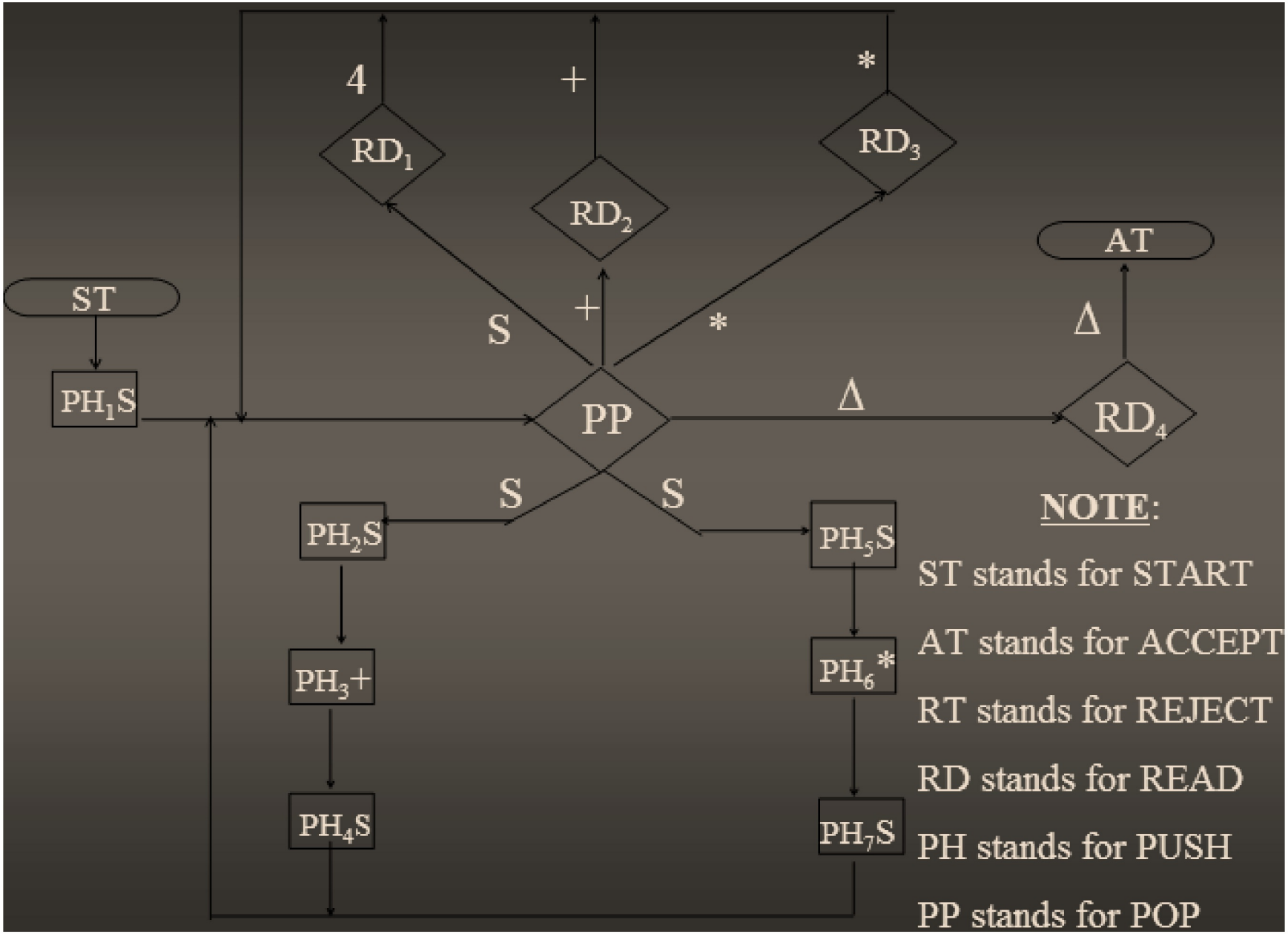
- Following is the PDA accepting the corresponding CFL

Example

- Consider the CFG

  S → S+S | S*S | 4

- Following is the PDA accepting the corresponding CFL


4
+
*
RD1
RD2
RD3
AT
ST
S
+
*
Δ
PH1S
PP
Δ
RD4
S
S
NOTE:
PH2S
PH5S
ST stands for START
AT stands for ACCEPT
PH3+
PH6*
RT stands for REJECT
RD stands for READ
PH4S
PH7S
PH stands for PUSH
PP stands for POP

- The string 4 + 4 * 4 traces the path shown in the following table

·Check Assinment As well as

STATE	STACK	TAPE
START	Δ	4+4*4
$PUSH_1$ S	S	4+4*4
POP	Δ	4+4*4
$PUSH_2$ S	S	4+4*4
$PUSH_3$ +	+S	4+4*4
$PUSH_4$ S	S+S	4+4*4
POP	+S	4+4*4
$READ_1$	+S	+4*4
POP	S	+4*4

Example continued ...

STATE	STACK	TAPE
$READ_2$	S	4*4
POP	Δ	4*4
$PUSH_5$ S	S	4*4
$PUSH_6$ *	*S	4*4
$PUSH_7$ S	S*S	4*4
POP	*S	4*4
$READ_1$	*S	*4
POP	S	*4

Example continued ...

STATE	STACK	TAPE
$READ_3$	S	4
POP	Δ	4
$READ_1$	Δ	Δ
POP	Δ	Δ
$READ_4$	Δ	Δ
ACCEPT	Δ	Δ

Example

converion CFG into PDA

1:convert into the Chmosky normal form

2:then start making PDA

- Consider the following CFG

  S □ XY

  X □ aX | bX | a

  Y □ Ya | Yb | a

- First of all, converting the CFG to be in CNF, introduce the non-terminals A and B as

  A □ a

  B □ b

Example Cont....

- The following CFG is in CNF

  S → XY

  X → AX | BX | a

  Y → YA | YB | a

  A → a

  B → b

Example Cont....

- The PDA corresponding to the above CFG w[illegible]