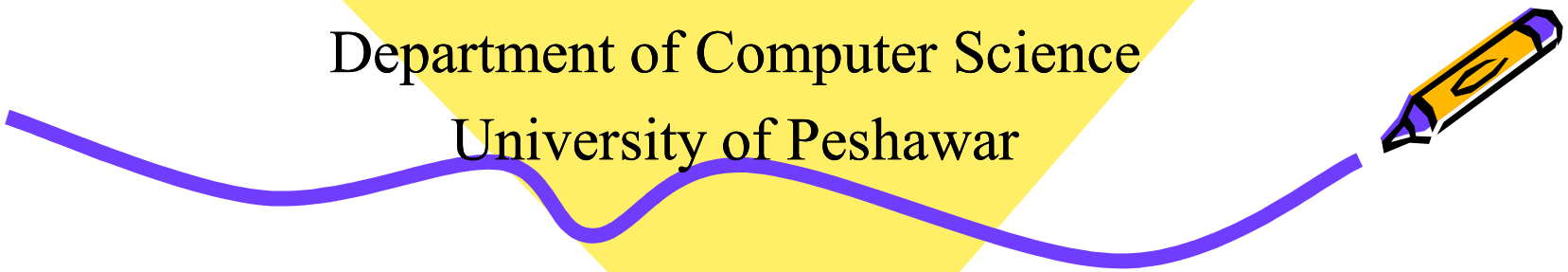
# Chapter # 7
# Pushdown Automata

Dr. Shaukat Ali
Department of Computer Science
University of Peshawar

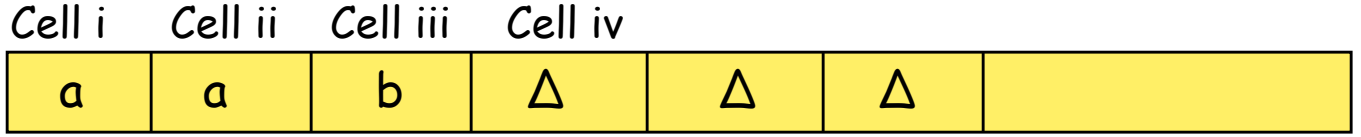


# Pushdown Automata.

- Pushdown automata is a pictorial representation of FAs.
- A pushdown automata contains several elements:

1. INPUT TAPE:
    1. Input tape is a data structure where the input string stays.
    2. The input tape must be long enough for any possible input and because any word in a* is a possible input, the TAPE must be infinitely long.
    3. The TAPE has a first location for the first letter of the input and then a second location.
    4. The locations into which we put the input letters are called cells. We name the cells with lowercase Roman numerals.
    5. The character Δ is used to indicate a blank in a TAPE cell.

# Pushdown Automata.

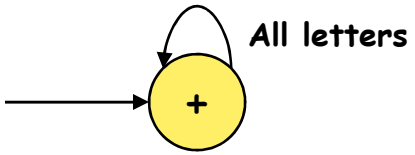



- As we process this TAPE on the machine, we read one letter at a time and eliminate each as it is used.
- When we reach the first blank cell, we stop.
  - We always presume that once the first blank is encountered, the rest of the TAPE is also blank.
- We read from right to left and never go back to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

# Pushdown Automata.

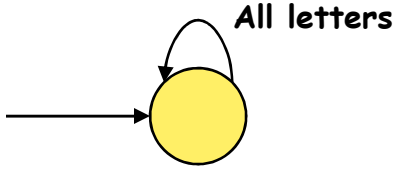
2. Symbols:
   1. Different symbols are used in the design of the PDA.


   2. The arrows into or out of these states can be drawn at any angle.
   3. The start state like – state connected to another state in the machine by a Λ edge (optional).
   4. We begin the process there by we read no input letter.
   5. A start state has one out edge and no arrows coming into it.
   6. An ACCEPT state is for a dead-end final state – once encountered, it cannot be left.
      1. Have in-edges and not out-edges

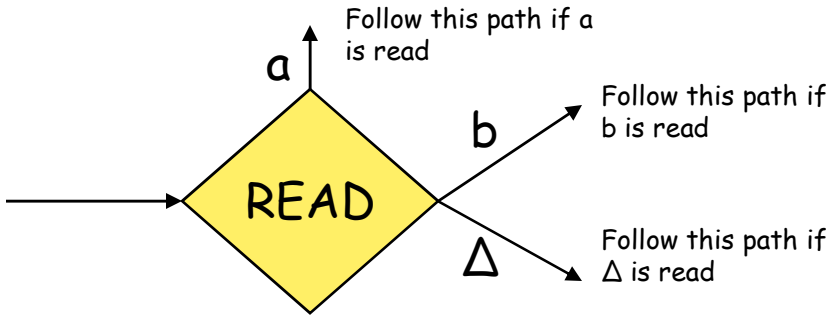


# Pushdown Automata.

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



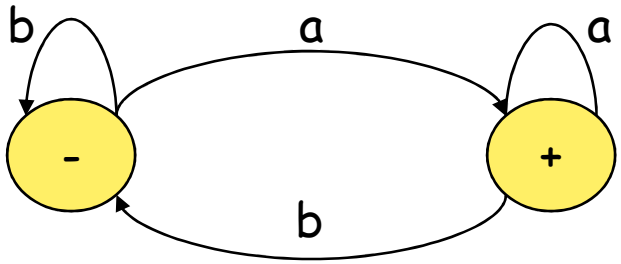

3. Function:
   1. Every function that a state performs in a FA is represented by a separate diamond symbol in the picture.
   2. The most important job performed by a state in an FA is to read an input letter from the tape and branch to other states depending on what letter has been read.
   3. For this purpose READ state has been used which is represented by a diamond symbol.



# Pushdown Automata.

4. The directions of the edges show only one of the many possibilities.
5. When Δ is read, it means that we are finished processing the input string. The Δ edge will lead to ACCEPT or REJECT depending on the input.
6. No restriction on duplication and instances of each letter

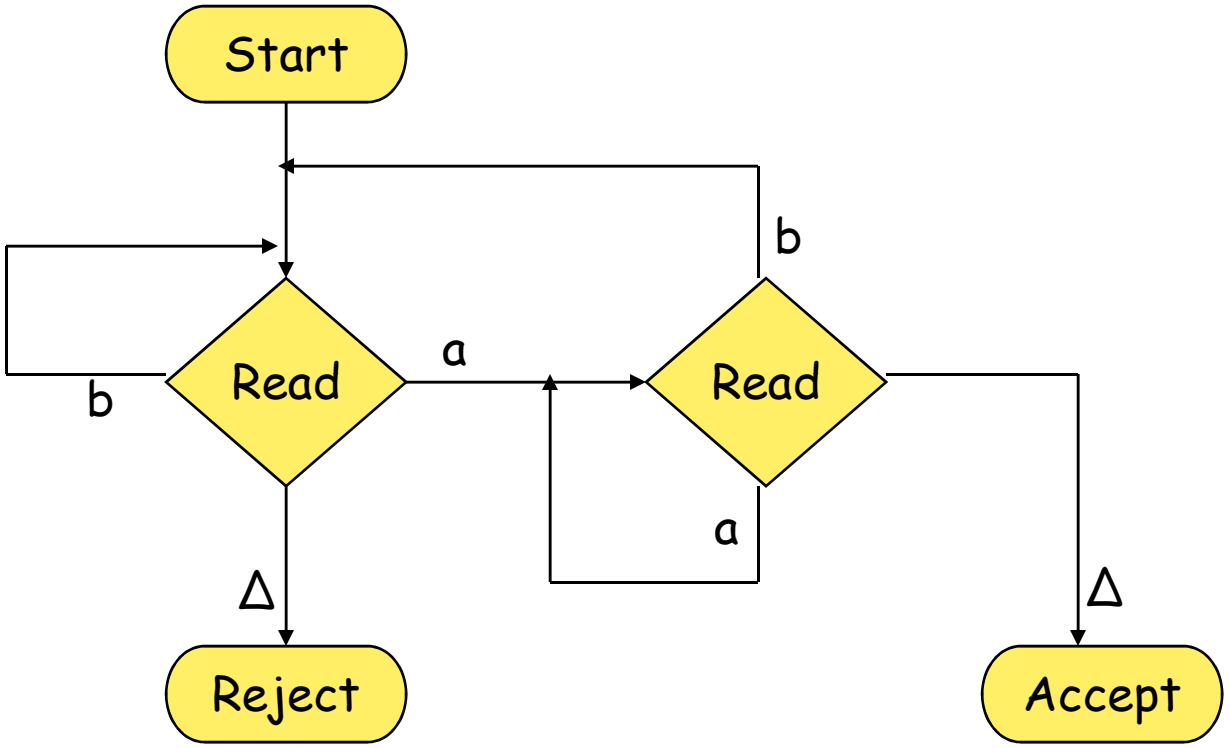
- Consider the following FA machine:




This FA accepts all words ending in the letter a.

# Pushdown Automata.

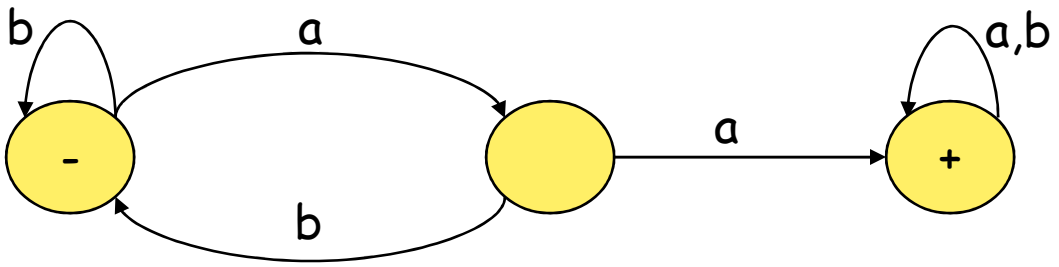
- Using these new symbolisms the machine becomes.

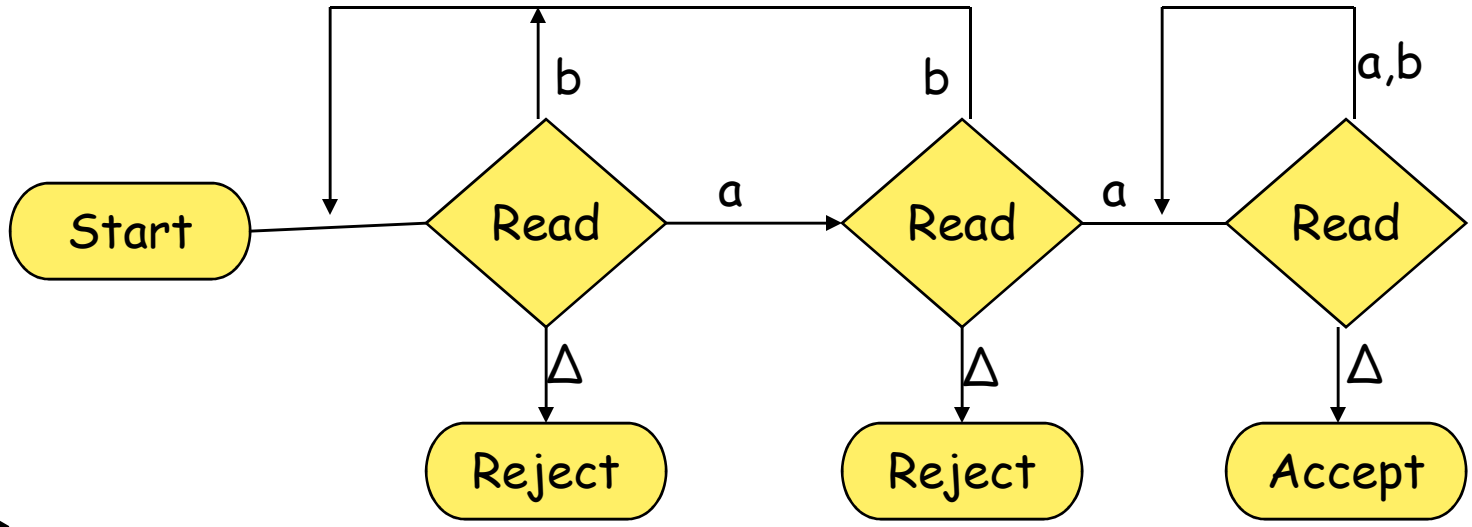



- The edge from STRAT need no label because START requires no letter.
- All the other edges do require labels.
- The edges are drawn straight line segments, not curves and loops as in FA.

# Example.

- Consider the following machine.



# PUSHDOWN STACK.

- A PUSHDOWN STACK is a place where input letters can be stored until we want to refer to them again.
- The operation PUSH adds a new letter to the top of the column.
- The new letter is placed on top of the STACK, and all 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.

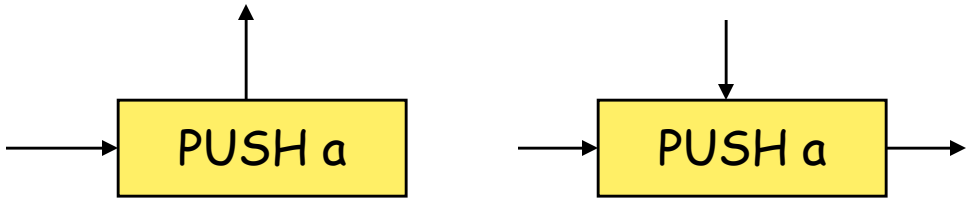
# PUSHDOWN STACK.

- If the SATCK is fed the letters a, b, c, d by the following sequence of instructions.
    1. PUSH a
    2. PUSH b
    3. PUSH c
    4. PUSH d

  then the 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

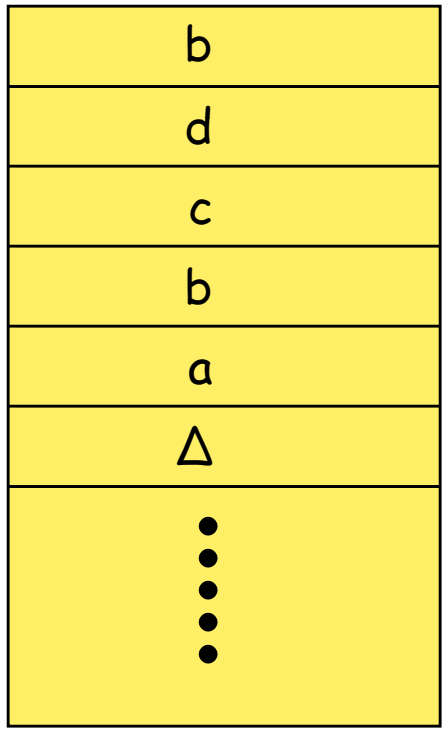
  PUSH b

  the letter b will be added to the STACK on the top. Each and every other letter will be pushed down to one position.
- A PUSH operation is represented by a rectangle labeled with PUSH and letter.



# PUSHDOWN STACK.

- A STACK has infinite location like INPUT TAPE.
- A pictorial representation of STACK with the above pushed letters can be.

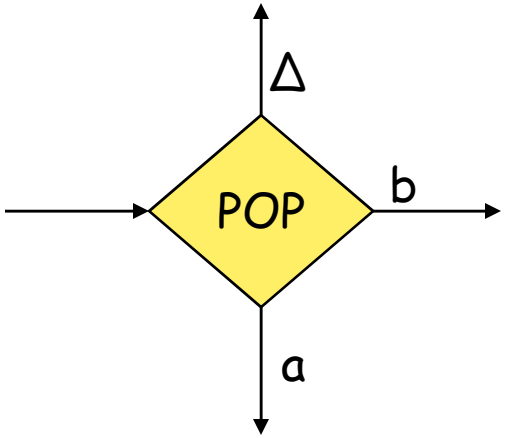


# PUSHDOWN STACK.

- The instruction to take a letter out of the STACK is called POP.
- This causes the letter on the top of the STACK to be brought out of the STACK.
- The rest of the letters are moved up one location each.
- A PSUHDOWN STACK is called a LIFO file, which stands for “the last in is the first out:.
- Popping an empty STACK, like reading an empty TAPE, give us the blank character Δ. Which means that the STACK is empty.

# PUSHDOWN STACK.

- A POP operation can be represented by a diamond symbol with three possible output edges.

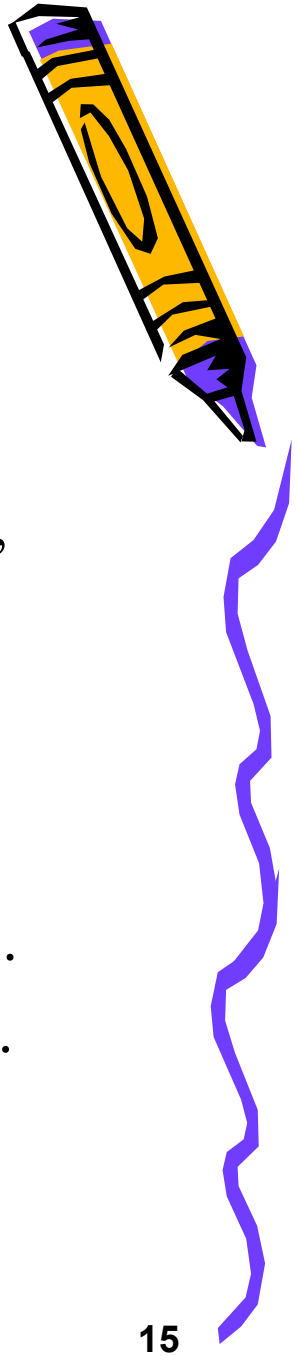



- Branching can occur at the POP state but not at PUSH states.
- We can leave PUSH states by the one indicated route, although we can enter a PSUH state from any direction.

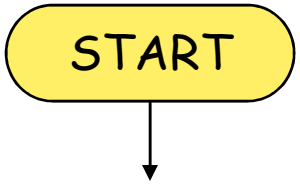
# PUSHDOWN STACK.

- When the pictorial representation of a FA has been mixed with a STACK and POP and PUSH states, we call them PUSHDOWN AUTOMATA abbreviated PDA.
- The benefit of PDA is that when it is incorporated into an FA, its language recognizing capabilities are increased considerably.

# PushDown Automata (PDA).



- A pushdown automata is a collection of eight things:
  1. An alphabet $\sum$ of input letters.
  2. An input TAPE (infinite in one direction). Initially, the string of input letters is placed on the TAPE starting in cell i. The rest of the TAPE is blank.
  3. An alphabet $\Gamma$ of STACK characters.
  4. A pushdown STACK (infinite in one direction). Initially, the STACK is empty (contains all blanks).
  5. One start state that has only out-edges, no in-edges.




# PushDown Automata (PDA).

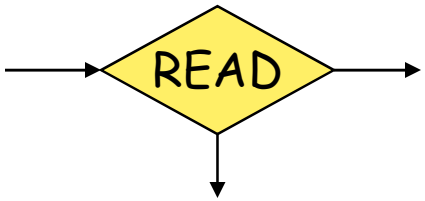
6. Halt states of two kinds: some ACCEPT and some REJECT. They have in-edges and no out-edges.


7. Finitely many nonbranching PUSH states that introduce characters onto the top of the STACK. They are of the form:


   where X is any letter in Γ.

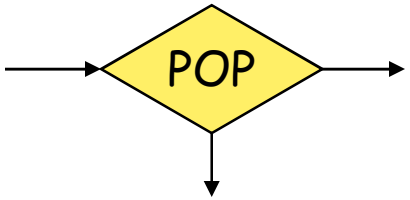
8. Finitely many branching states of two kinds:
   1. States that read the next unused letter from the TAPE.

# PushDown Automata (PDA).




- Which may have out-edges labeled within letters from $\sum$ and the blank character $\Delta$ and does not have insistance that there be a label for each letter of $\sum$ or $\Delta$.

2. State that read the top character of the STACK.




- Which may have out-edges labeled with the letters of $\Gamma$ and the blank character $\Delta$ again with no restriction.

- These states are connected so as to become a connected directed graph.

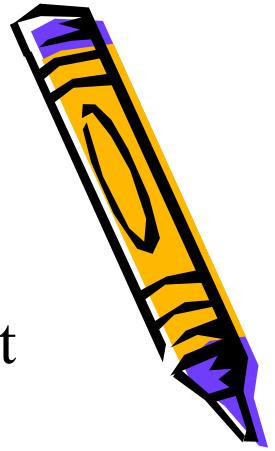
# PDA Working.

- To run a string of input letters on a PDA means to begin from the START state and follow the unlabeled edges and labeled edges that apply to produce a path through the graph.
- This path will end either at a halt state or will crash in a branching state when there is no edge corresponding to the letter/character read/popped.
- When letters are read from the TAPE or characters are popped from the STACK, they are used up and vanished.
- An input string with a path that ends in ACCEPT is said to be accepted.
- The set of all input strings accepted by a PDA is called the language accepted by the PDA or the language recognized by the PDA.

# Types of PDA.

- There are two types of PDA:
  - Deterministic PDA.
  - Non-Deterministic PDA.
- A Deterministic PDA is one for which every input string has a unique path through the machine.
- A Non-Deterministic PDA is one for which at certain times we may have to choose among possible paths through the machine.
- PDA's that are equivalent to CFG's are the Non-Deterministic ones.

# Theorem 1.



- The theorem is:
  - For every regular language L there is some PDA that accepts it.
- As we have discussed that PDA is more powerful than FA.
  - Each FA corresponds to a regular language. A FA cannot represent a non-regular language.
  - A PDA can be used to corresponds to both a regular language and non-regular language.
- As a regular langue L is accepted by some FA. The constructive algorithm can be used for converting FA into an equivalent PDA.

- End of Chapter # 7