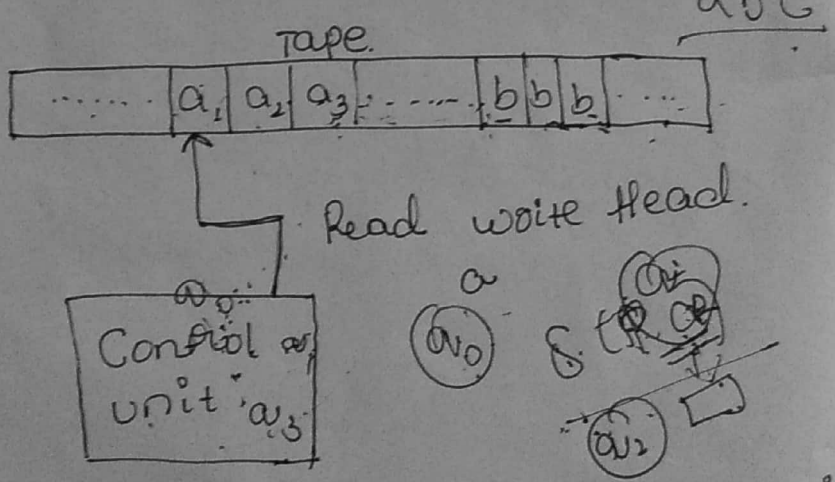
module id: 5

Topic: Turing machines & undecidability

Introduction to Turing machine:

The Turing machine is a generalized machine which can recognize all types of languages like, Regular Languages (Type 3)
Context free languages (Type 2)
Context sensitive languages (Type 1).
unrestricted grammar (Type 0). PDA

Turing machine model.



* Basically Turing machine is also a finite automata.
* which contains 3 basic parts.
  - Tape.
  - Read write head.
  - Control unit.

* Tape is used to store the information and is divided into cells

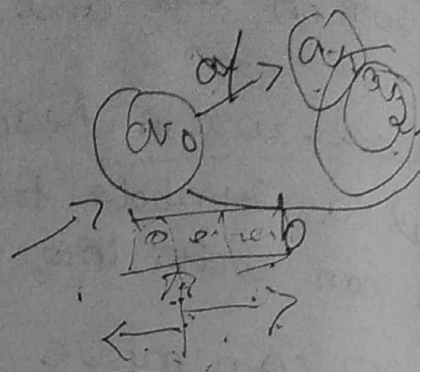
- each cell can store only one symbol.
- The string to be scanned will be stored from the leftmost position on the tape. (The ending of string will be identified by continuous blank symbols)
- basically This tape is assumed to be infinite both on left and right side of the string.

* Read-write head: The read-write head can read a symbols from where it is pointing to and it can write into the tape to where it points to.
- It can move either of sides, left or right.
- It can perform both side movements.
- it is also responsible for performing various actions
1. change of state from one state to another state
2. The symbol pointing to by the read-write head can be replaced by another symbol.
3. The read-write head may move either towards left or towards right.

* Control unit: The reading from the tape or writing into the tape is determined by the control unit. The different moves performed by the machine depends on the current scanned symbol and the current state. The control unit consults action table i.e., transition table and carryout the tasks.

The Turing machines can be represented using various notations such as.

- Transition tables.
- Instantaneous descriptions
- transition diagram.



- Transition Table

δ	Tape symbols (Γ)				
States.	a	b	X	Y	B
q_0	(q_1, X, R)	-	-	(q_1, Y, R)	-
q_1	(q_1, a, R)	(q_2, Y, L)	-	(q_1, Y, R)	-
q_2	(q_2, a, L)	-	(q_0, X, R)	(q_2, Y, L)	-
q_3	-	-	-	(q_3, Y, R)	(q_4, B, R)
q_4	-	-	-	-	-

The transitions shown in the table can also be written as.

$\delta(q_0, a) = (q_1, X, R)$

$\delta(q_0, Y) = (q_3, Y, R)$

$\delta(q_1, a) = (q_1, a, R)$

$\delta(q_1, b) = (q_2, Y, L)$

$\delta(q_1, Y) = (q_1, Y, R)$

$\delta(q_2, a) = (q_2, a, L)$

$\delta(q_2, X) = (q_0, X, R)$

$\delta(q_2, Y) = (q_2, Y, L)$

$\delta(q_3, Y) = (q_3, Y, R)$

$\delta(q_3, B) = (q_4, B, R)$

$\delta : Q \times \Sigma \to Q \times \Sigma^* \times \{L, R\}$

$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$

Turing machine $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ where

Q is set of finite state.

Σ is set of input alphabets.

Γ is set of tape symbols

δ is transition function from $Q \times \Gamma$ to $Q \times \Gamma \times \{L, R\}$

q_0 is the start state.

B is a special symbol indicating blank character

$F \subseteq Q$ is set of final states.

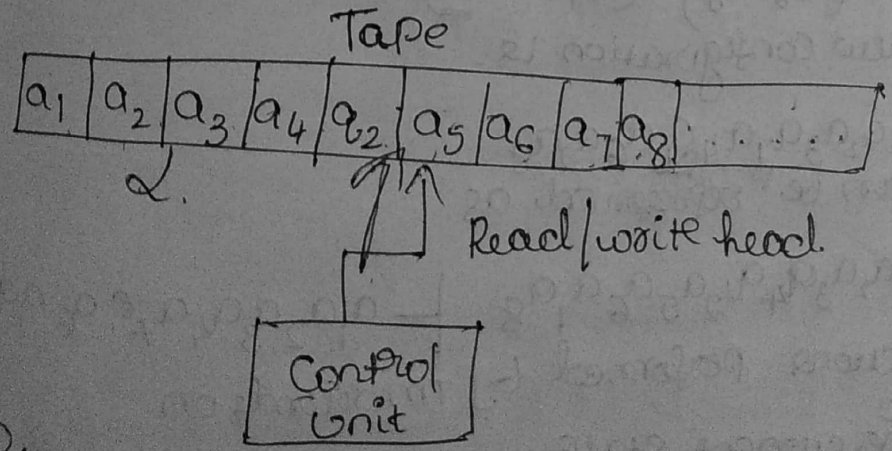
Instantaneous Description (ID)

ID of TM is a string in $\alpha q \beta$, where q is the current state.

where $\alpha\beta$ is the string made from tape symbols denoted by Γ.

where α - is set of characters of left side of current state q β - is set characters of right side of current state.

Example:



In this machine $a_i \in \Gamma$

As per the example

The read write head which is next symbol to be

The general form of ID is

$a_1a_2a_3a_4q_2a_5a_6a_7a_8$ $\quad \alpha q \beta.$

$\alpha \quad q \quad \beta.$

- where $a_1a_2a_3a_4$ is substring towards left of q_2
- $a_5a_6a_7a_8$ is substring towards right of q_2
- q_2 is current state of the machine.

As per the example, suppose there is a transition

$\delta(q_2, a_5) = (q_3, b_1, R)$

Now the tape string is.

$a_1a_2a_3a_4b_1q_3a_6a_7a_8 \vdash$

a | b

a b a

it can be represented as.

$a_1a_2a_3a_4q_2a_5a_6a_7a_8 \vdash a_1a_2a_3a_4b_1q_3a_6a_7a_8.$

Suppose

Consider same string again

$a_1a_2a_3a_4q_2a_5a_6a_7a_8.$

The transition is

$\delta(q_2, a_5) = (q_1, c_1, L)$

The new configuration is.

$a_1a_2a_3q_1a_4c_1a_6a_7a_8.$

This can be represented as.

$a_1a_2a_3q_4a_2a_5a_6a_7a_8. \vdash a_1a_2a_3q_1a_4c_1a_6a_7a_8$

These actions performed by TM depends on

- The current state.
- The whole string to be scanned.
- The current position of Read-write head.

- Construction of Turing machine (TM)

Example: Obtain a Turing machine to accept the language $L = \{0^n 1^n \mid n \geq 1\}$.

General Procedure.

Let q_0 be the start state and let the R/W head points to the first symbol of the string to be scanned. The general procedure to design TM for this case is shown below,

1. Replace the left most 0 by X and change the state to q_1. and move Read write head towards right. This is because after a zero is replaced, we have to replace the corresponding 1 so the no. of zero's matches with no. of 1's.

2. Search for the left most 1 and replace it by the symbol Y and move towards Left (so as to obtain the leftmost 0 again). steps 1 & 2 can be repeated.

Design: Consider the string.

XX0OYY11
↑
q_0

where first two 0's are replaced by X's and first two 1's are replaced by Y's.

step 1: In state q_0, replace 0 by X,

$$\delta(q_0, 0) = (q_1, X, R)$$

The resulting configuration is,

XXX0YY11
↑
q_1

Step 2:

$$\delta(q_1, 0) = (q_1, 0, R)$$
$$\delta(q_1, Y) = (q_1, Y, R)$$

XXX0YY11
↑

Step 3: $\delta(q_1, 1) = (q_2, Y, L)$

XXX0YYY1
↑ q_2

Step 4: $\delta(q_2, Y) = (q_2, Y, L)$

$\delta(q_2, 0) = (q_2, 0, L)$

Step 5:

$\delta(q_2, X) = (q_0, X, R)$

XXX0YYY1
↑ q_0

Now repeating the step's 1 through 5, we get configuration

XXXXYYYY
↑ q_0

Step 6: In state q_0, if the scanned symbol is Y, it means There are no more 0's. if there are no 0's, we should see that there are no 1's, for this we change the state to q_3.

$\delta(q_0, Y) = (q_3, Y, R)$

XXXXYYYY
↑ q_3

$\delta(q_3, Y) = (q_3, Y, R)$. [Repeadly applying this transitions]

XXXXYYYYB
↑ q_3

$\delta(q_3, B) = (q_4, B, R)$

here is q_4 is final state & obtained configuration is.

XXXXYYYYBB
↑
q_4

So Turing machine accept the language.

$L = \{a^n b^n \mid n \geq 1\}$

$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

where,

$Q = \{q_0, q_1, q_2, q_3\}$

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

$\Gamma = \{0, 1, X, Y, B\}$

$q_0 \in Q$ is the start state of machine

$B \in \Gamma$ is blanck symbol.

$F = \{q_4\}$ is final state.

δ is.

$\delta(q_0, 0) = (q_1, X, R)$

$\delta(q_1, 0) = (q_1, 0, R)$

$\delta(q_1, Y) = (q_1, Y, R)$

$\delta(q_1, B) = (q_2, Y, L)$

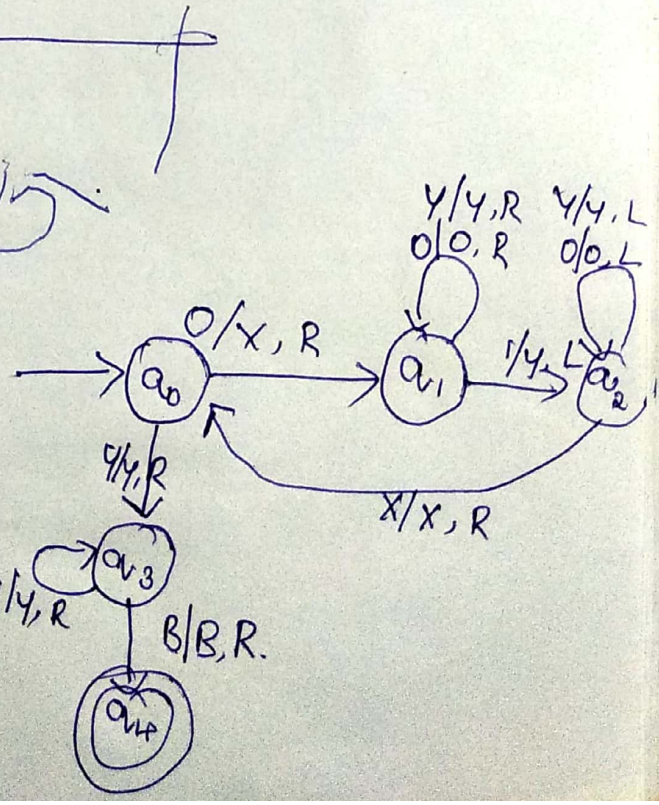
$\delta(q_2, Y) = (q_2, Y, L)$

$\delta(q_2, 0) = (q_2, 0, L)$

$\delta(q_2, X) = (q_0, X, R)$

$\delta(q_0, Y) = (q_3, Y, R)$

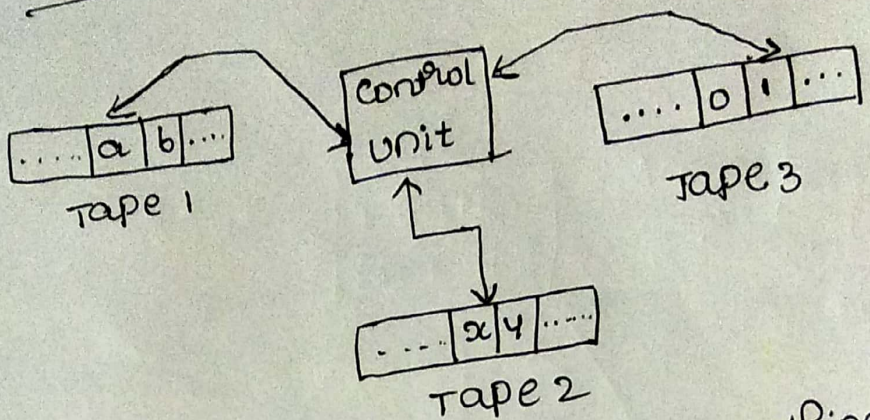
$\delta(q_3, Y) = (q_3, Y, R)$



17/4/2020

Extensions of Turing machines

*Multi-tape Turing machines.



A multi-tape Turing machine is nothing but a standard Turing machine with more no. of tapes.
- Each tape is controlled independently with indep-
-ndent Read-write head.
- It is Represented in above diagram.
- various components of multi-tape turingmachine are.

* Finite control.
* Each tape with its own Symbol's and R/W head

- Depending on the type of Input from start state. Tape is selected.
- The formal definition of multi-tape Turingmachine

where, $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

Q - is Set of finite States
Σ - is set of input alphabets.
Γ - is set of tape Symbol's
δ - is transition function from $Q \times \Gamma^n$ to $Q \times \Gamma^n \times \{L, R\}^n$
$\delta(q, a, b, c) = (p, x, y, z, L, \ldots)$
q_0 - is the start state.
B - is special Symbol indicating blanks character
$F \subseteq Q$ is set of final States.

* Turing machine with stay-option.

In the standard Turing machine, after scanning the symbol and after replacing the symbol on the tape with R/W head used to move either left or right based on the control mechanism. Apart from that there is one more option is there, where in the read/write head stays in same position after updating the symbol on the tape. Then the Turing machine is called TM with stay-option.

formally the T. machine with stay option can be written as. $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

Q - states,

Σ - Input Symbols.

Γ - Tape Symbol's

δ - $Q \times \Gamma$ to $Q \times \Gamma \times \{L, R, S\}$

where L - indicates left movement.

R - Right movement

S - Stay option.

q_0 - Start State.

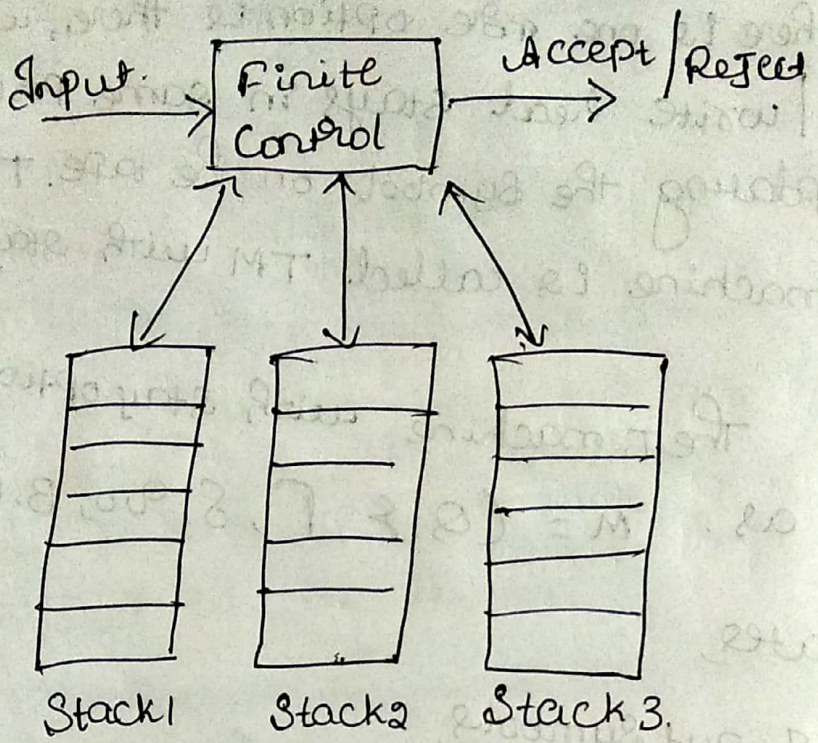
B - Special Symbol indicating blank character.

$F \subseteq Q$ is set of final states.

* multistack Turing machines.

The languages which accepted by PDA are also accepted by TM. At the same time, any language which is not accepted by PDA is also accepted by TM.

- TM is a generalized machine which can accept all languages.
- The multistack Turing machine is generalized PDA.



- The above diagram represents the multistack TM
- which contains multiple stacks. depending on type of input and operation stacks will be chos
- the move of the multistack depends on.
  - current state of finite control.
  - the input symbol chosen from Σ
  - The symbol on top of each stack.
- once the transition is applied for current state, input symbol and current top of each stack The multistack machine may.
  - > Change the state.
  - > it may replace the top of the stack.

$$\delta(q, a, X_1, \cdots X_n) = (P, \alpha_1, \alpha_2, \ldots \alpha_n)$$

undecidability

* Introduction:

* language that is not recursively enumerable

The machine m is recursively enumerable language if and only if $L = L(m)$, Any instance of a Problem for which the Turing machine halts whether the input is accepted / rejected is called solvable or decidable Problem.

* undecidable Problems that are RE.

The Problems that run forever on a Turing machine are not solvable. In other words, there are some Problem input instances for which Turing machines will not halt on input that they do not accept. Those Problems are called unsolvable or undecidable Problems. In general, if there is no general algorithm capable of solving every instance of the problem, then the decision Problem is unsolvable. more Precisely, if there is no Turing machine recognizing the language of all strings for various instances of the Problems input for which the answer is yes or no. then the decision Problem is unsolvable

* Halting Problem

The Halting Problem can informally be stated as "Given a Turing machine M and an input string w with the initial configuration q_0, after some computations do the machine M halts?

It can be defined as (M, w) where M is Turing machine, Halts or does not halt when w is applied as the input. The domain of this problem is to be taken as the set of all Turing machines and all w i.e given the description of an arbitary Turing machine and the input string w.

- it is not possible to find the answer for Halting Problem by simulating the action of m on w by a universal Turing machine, because there is no limit on the length of the computation.
- if machine M enters into an infinite loop, then no matter how long we have to wait for, the machine may be in infinite loop

* Post's Correspondence Problem.

The post correspondence problem can be stated as follows. given two sequence of n strings on some alphabet Σ say,

$$A = w_1, w_2, w_3 \ldots w_n$$

and

$$B = v_1, v_2, v_3 \ldots v_n$$

we say that there exists a post correspondence solution for pair (A, B) if there is a non Empty sequence of integers i, j, ... k such that,

$$w_i w_j \ldots w_k = v_i v_j \ldots v_k$$

The Post correspondence Problem is to device an algorithm that will tell us, for any (A, B) whether or not there exists a PC-Solution.

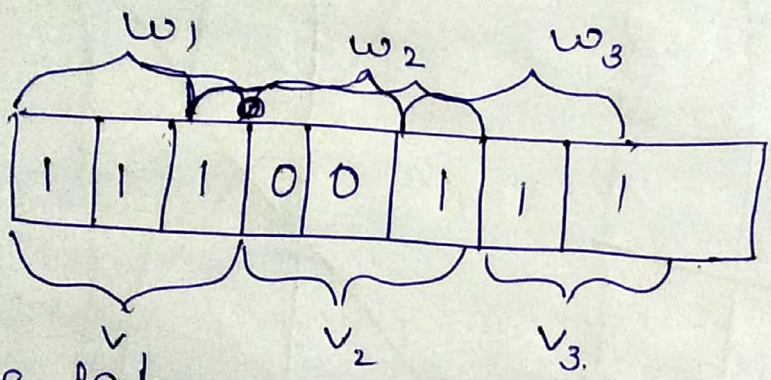
For Example, let $\Sigma = \{0, 1\}$. Let A is w_1, w_2, w_3 as shown below:

$w_1 = 11$, $w_2 = 100$, $w_3 = 111$

let B is v_1, v_2, v_3 as shown below;

$v_1 = 111$, $v_2 = 001$, $v_3 = 11$.

For this case, there exists a PC-Solution as shown below.



if we take,

$w_1 = 00$, $w_2 = 001$, $w_3 = 1000$

$v_1 = 0$, $v_2 = 11$, $v_3 = 011$

There cannot be any PC-solution simply because any string composed of elements of A will be longer than the corresponding string from B.