Church Turing Hypothesis

(Lecture 25) (0)

Church's thesis

Various formal models of computation such as "Recursive functions" and "Post Systems" were established by 03 prominent persons: Church, Keleene and Post.

- A function is called primitive recursive iff 'it' can be constructed from the basic functions by sucessive composition and primitive recursion.

- A post system is similar to unrestricted grammar consisting of an alphabet and some production rules by which sucessive strings can be derived.

⇒ these computational model though looking different expressed the same thing.

⇒ This observation was formalized in church's thesis:

Church Turing Hypothesis

1

"Any effective computation" or "any algorithmic" procedure that can be carried out by a human being or a team of human beings or a computer, can be carried out by some turing machine".

$\Rightarrow$ there is an effective procedure to solve a decision problem P iff there is a TM that answers yes ~~to solve the decision problem P~~ on input $w \in P$ and no on input $w \notin P$.

? – This theory maintains that all the models of computations those are proposed or yet to be proposed are equivalent in their power to recognise languages or compute functions.

– This thesis predicts that it is unable to construct models of computation more powerful than the existing ones.

$\Rightarrow$ Turing thesis states that we cannot go beyond turing m/cs or their equivalents $\Downarrow$ Church – Turing Hypothesis

"Effective computation" $\leftarrow$ no precise definition

a "Algorithmic procedure" $\swarrow$

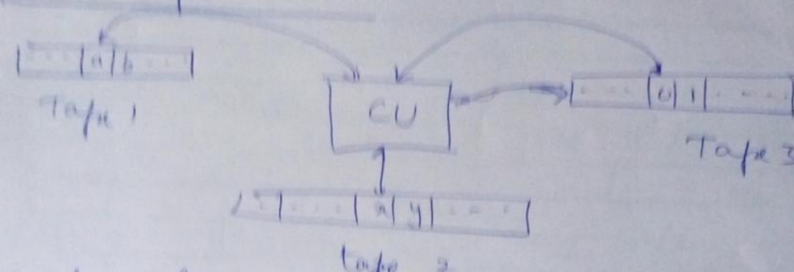
Church's thesis is not a mathematically precise statement. $\Rightarrow$ neither proved ~~Nor~~ disproved.

$\Downarrow$ accepted by scientists.

Variations of TMs

With some minor modification, we can have 2 types of TMs
(1) Multi-tape TM (2) Non-deterministic TM.

Multi-Tape TM



component of multi-tape TM

(a) Finite control.
(b) Each tape having its own symbols and a R/W head.

- Each tape is divided into cells which can hold any symbol from the given alphabet.
- to start with TM should be in start state q_0
- If the R/W pointing to tape 1 moves towards right, the R/W head pointing to tape 2 and 3 may move towards right or left depending on the transition.

Def

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

- The move of a multi-tape TM depends on the current state and the scanned symbol by each of the tape heads.

$\delta(q, a, b, c) = (p, x, y, z, L, R, S)$

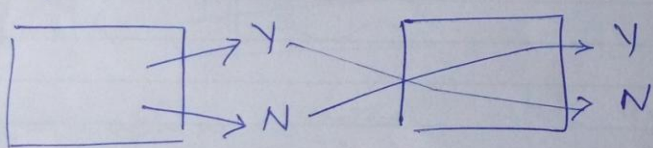
Equivalence of single-tape and multi-tape TMs

- Theorem: Every lang accepted by a multi-tape TM is recursively enumerable.

Recursive lang

- (*) If there exists a TM that accepts every string of L and rejects every string not in L, then lang is said to recursive.

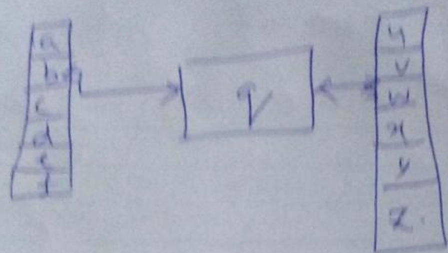
- (*) If a lang is recursive, then its complement is also recursive.



L $\qquad\qquad$ $\bar{L}$

- (*) If L is a language for some TM, then it is recursively enumerable.

A recursively enumerable lang is a formal lang for which there exists a TM (or other computable function) that will halt and accept when presented with any string in the lang as input BUT may either halt and reject or loop forever when presented with a string not in the language.

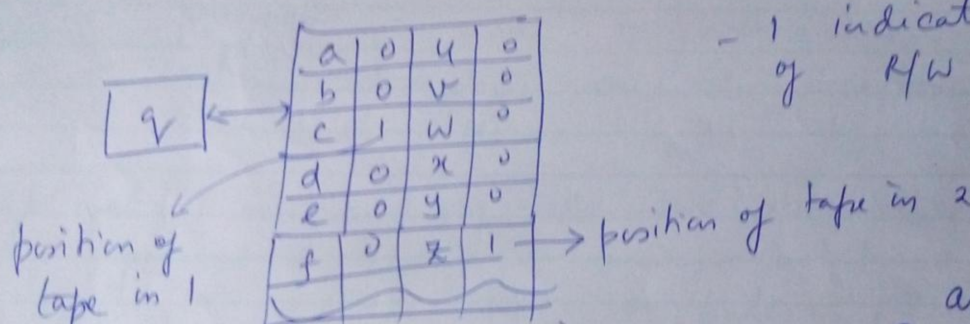


Multitape

2 tape.

- Can be simulated using a single tape TM which has 4-stack.

- 1 indicates position of R/W head.



- The machine ~~enters~~ is in state q as the first tape head is pointing to c and second to z.
- m/c enters state p iff the transition is defined for TM with multi-tapes

⇒ So, whatever transitions have been applied for multi-tape TM, if we apply the same for the new m/c we constructed, then the two m/c^s are equivalent.

Non-deterministic TM

difference lies only in the definition of δ

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

δ is a transition from $Q \times \Gamma$ to $2^{Q \times \Gamma \times (L/R)}$

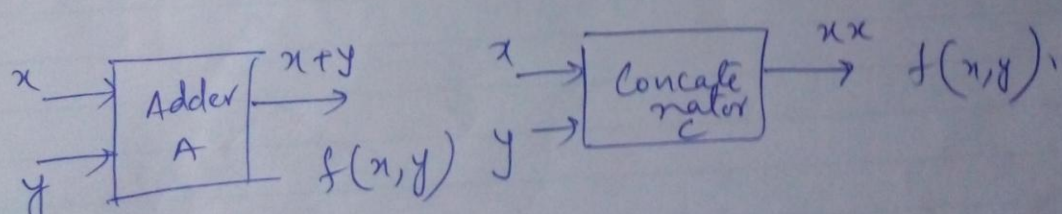
Programming techniques for TMs

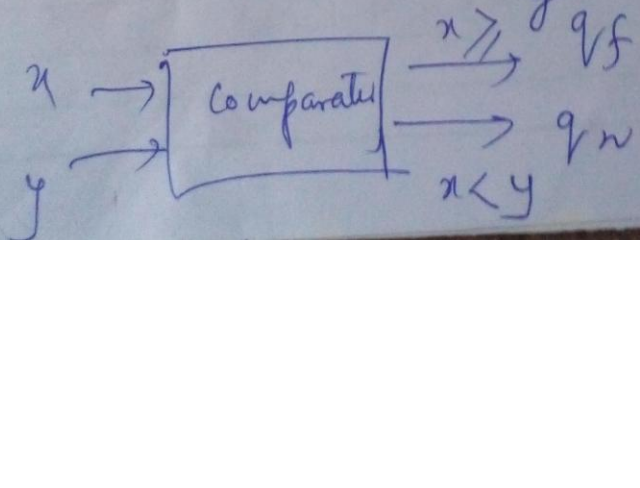
- Initially hard wired.
- stored program concept
- program ← set of subroutines.
- each subroutine can be represented by individual TMs'

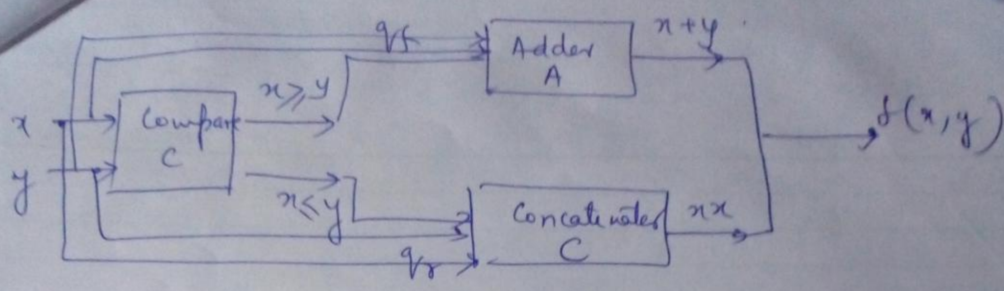
Q/ Let x and y be two positive integers represented using unary notation. Design a TM that computes the function

$$f(x,y) = x+y \quad \text{if } x \geq y$$
$$f(x,y) = xx \quad \text{if } x < y.$$

Sol: Block diagrams







Block diagram for a ~~sub~~ subroutine to compu[te]

Q) obtain a TM to multiply two unary numbers separated by a delimiter 1

<u>Sol.</u> ① Let x and y be two unary numbers x has <u>m</u> 0's and y has <u>n</u> 0's. ~~to~~

$0^m 1 0^n$.

② product should be stored and original number should not be destroyed.

$0^m 1 0^n 1 0^{mn}$

$x = 00$
$y = 0000$

<u>Input</u>

$\underbrace{00}_{x}1\underbrace{0000}_{y}1BBBB$

<u>Output</u>

$\underbrace{00}_{x}1\underbrace{0000}_{y}1\underbrace{00000000}_{xy}BBBB$

...cedure

Assume tape has a string $0^m 1 0^n 1 BBB$ which is divided into 3 groups.

$$\underbrace{0^m}_{\text{1st group}} 1 \underbrace{0^n}_{\text{2nd group}} 1 \underbrace{BBBB}_{\text{3rd group}}$$

q_0 0 0 1 0 0 0 0 1 B B B B B B B - - -

q_1 B 0 1 0 0 0 0 1 | 0 0 0 0 B B B B

B B 1 0 0 0 0 1 | 0 0 0 0 0 0 0 0 B B B

q_2 X → q_3 q_4

$\delta(q_0, 0) \Rightarrow (q_1, B, R)$

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

$\delta(q_1, 1) = (q_2, 1, R)$

$\delta(q_2, 0) = (q_3, X, R)$

$\delta(q_3, 0) = (q_3, 0, R)$

$\delta(q_3, 1) = (q_3, 1, R)$

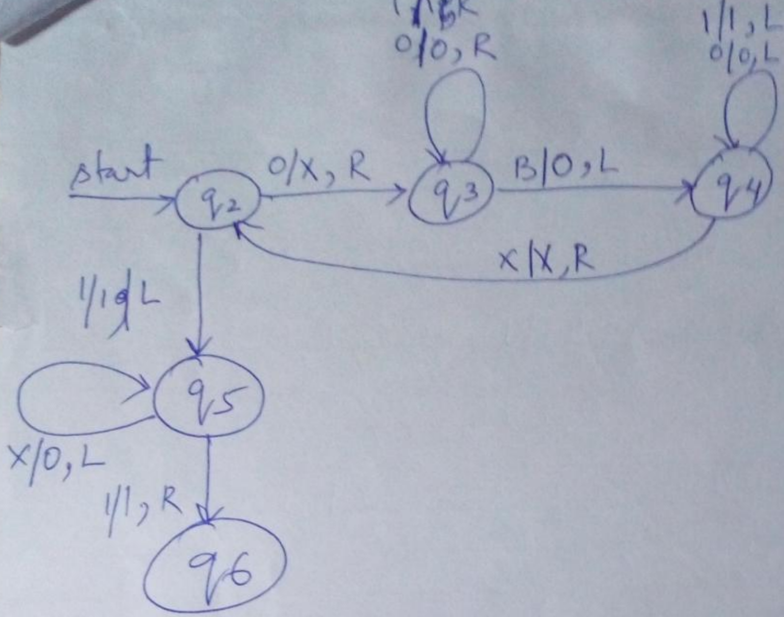
$\delta(q_3, B) = (q_4, 0, L)$

$\delta(q_4, 0) = (q_4, 0, L)$

$\delta(q_4, 1) = (q_4, 1, L)$

$\delta(q_4, X) = (q_2, X, R)$

$\delta(q_2, 1) = (q_5, 1, L)$



$$\delta(q_5, X) = (q_5, 0, L)$$
$$\delta(q_5, 1) = (q_6, 1, R)$$