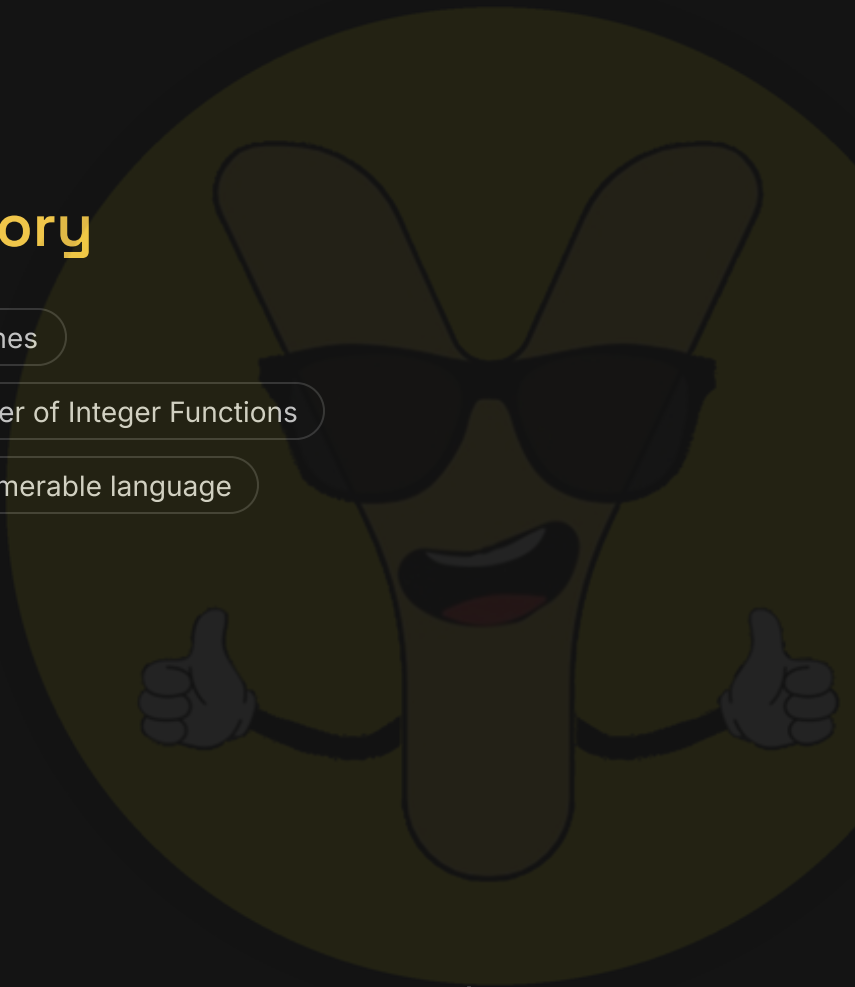
AKTU · 2ND YEAR · SEMESTER 4 · BCS402

Theory of Automata and Formal Languages

Unit 5 — Turing Machines and Recursive Function Theory

- Basic Turing Machine Model
- Representation of Turing Machines
- Language Acceptability of Turing Machines
- Techniques for Turing Machine Construction
- Modifications of Turing Machine
- Turing Machine as Computer of Integer Functions
- Universal Turing machine
- Linear Bounded Automata
- Church's Thesis
- Recursive and Recursively Enumerable language
- Halting Problem
- Post's Correspondance Problem
- Introduction to Recursive Function Theory



DECODED BY

Chirag Jain

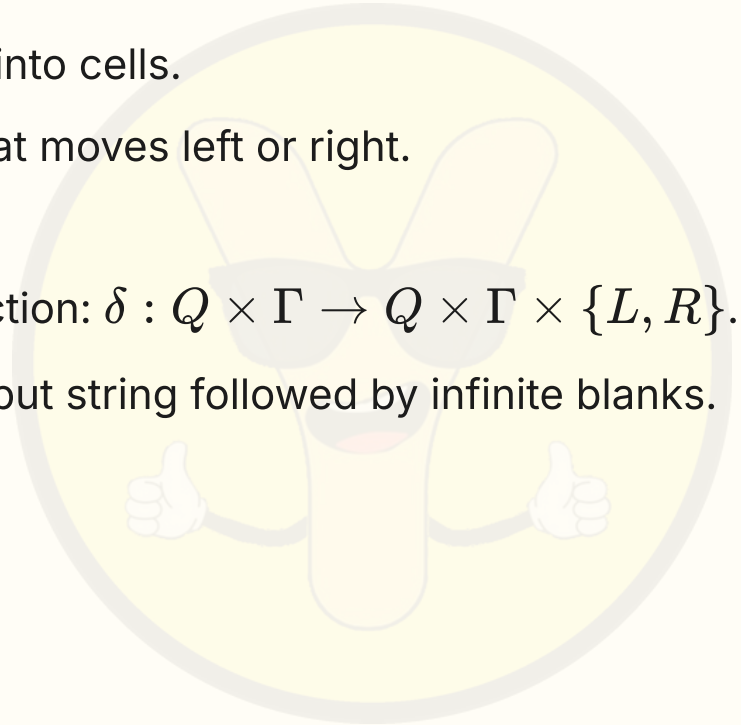
Founder, Ycotes · Gold Medalist in Artificial Intelligence, IIT Madras · taught 10,000+ students

theycotes.com

Basic Turing Machine Model

A Turing Machine (TM) is a mathematical model of computation that defines an abstract machine.

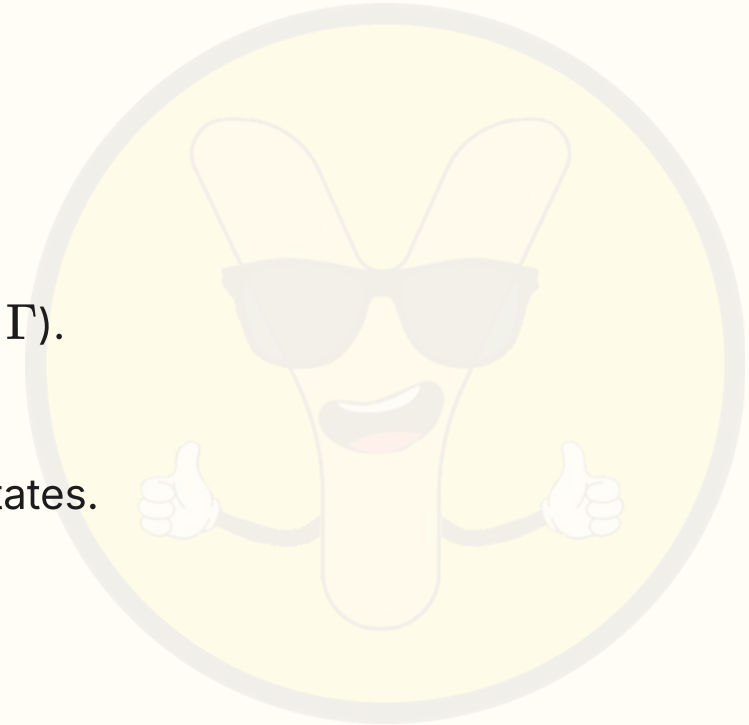
- Consists of an infinite tape divided into cells.
- Equipped with a read/write head that moves left or right.
- Has a finite state control unit.
- Operates based on a transition function: $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$.
- The tape is initially filled with the input string followed by infinite blanks.



Representation of Turing Machines

A TM can be formally defined as a 7-tuple.

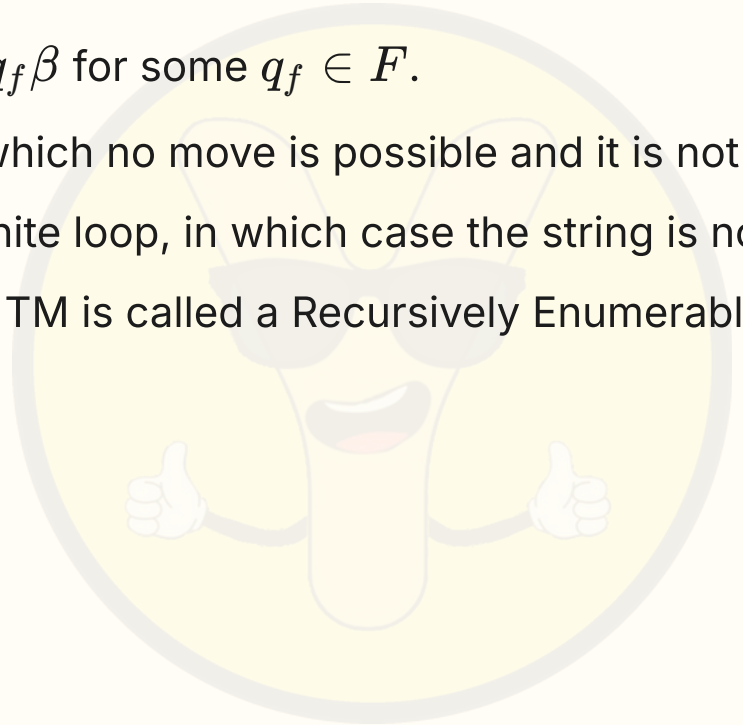
- $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$
- Q: Finite set of states.
- Σ: Finite set of input symbols.
- Γ: Finite set of tape symbols ($\Sigma \subset \Gamma$).
- $B \in \Gamma$: Blank symbol.
- $F \subseteq Q$: Set of final or accepting states.



Language Acceptability of Turing Machines

A TM accepts a string if it enters an accepting state.

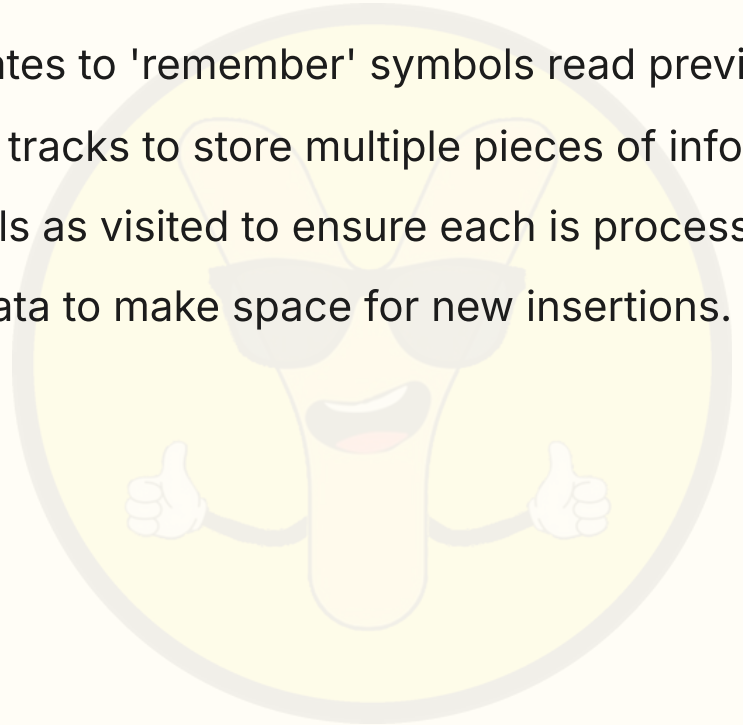
- A string w is accepted if $q_0 w \vdash^* \alpha q_f \beta$ for some $q_f \in F$.
- If the machine enters a state from which no move is possible and it is not a final state, the string is rejected.
- The machine may also enter an infinite loop, in which case the string is not accepted.
- The set of all strings accepted by a TM is called a Recursively Enumerable language.



Techniques for TM Construction

Building a TM involves designing transitions to manipulate tape data.

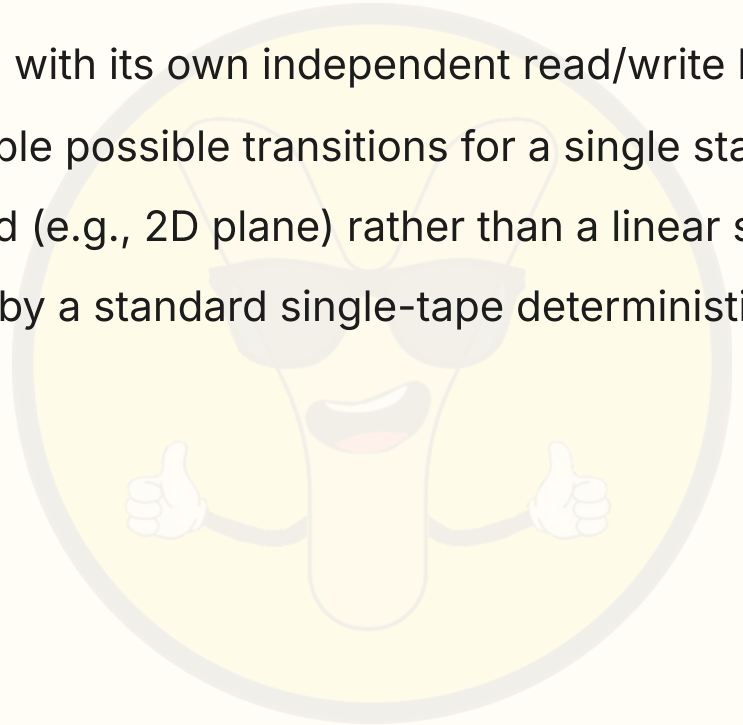
- Storage in the finite control: Use states to 'remember' symbols read previously.
- Multiple tracks: Divide the tape into tracks to store multiple pieces of information.
- Checking off symbols: Mark symbols as visited to ensure each is processed exactly once.
- Shifting symbols: Move blocks of data to make space for new insertions.



Modifications of Turing Machine

Several variants exist that are equivalent in computational power to the standard TM.

- Multi-tape TM: Multiple tapes, each with its own independent read/write head.
- Non-deterministic TM: Allows multiple possible transitions for a single state-symbol pair.
- Multi-dimensional TM: Tape is a grid (e.g., 2D plane) rather than a linear strip.
- All these variants can be simulated by a standard single-tape deterministic TM.



TM as Computer of Integer Functions

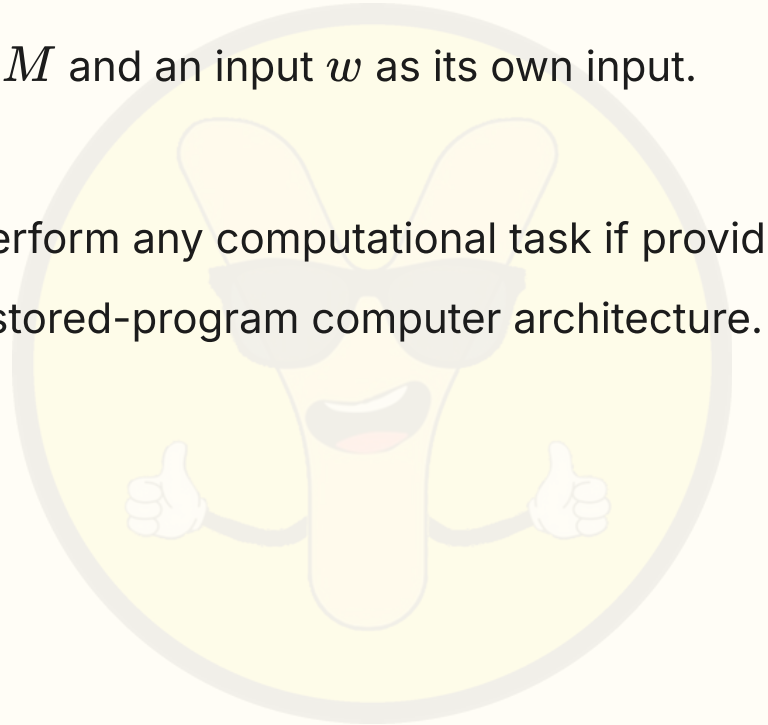
TMs can perform arithmetic by representing integers as strings.

- Unary representation: Integer n is represented as $n+1$ ones (1^{n+1}).
- Function computation: A TM computes $f(n_1, ..., n_k)$ if it halts with the result on the tape.
- The machine starts with input $1^{n_1+1}B1^{n_2+1}$ and ends with $1^{f(n_1,n_2)+1}$.
- TMs can compute addition, subtraction, and multiplication of integers.

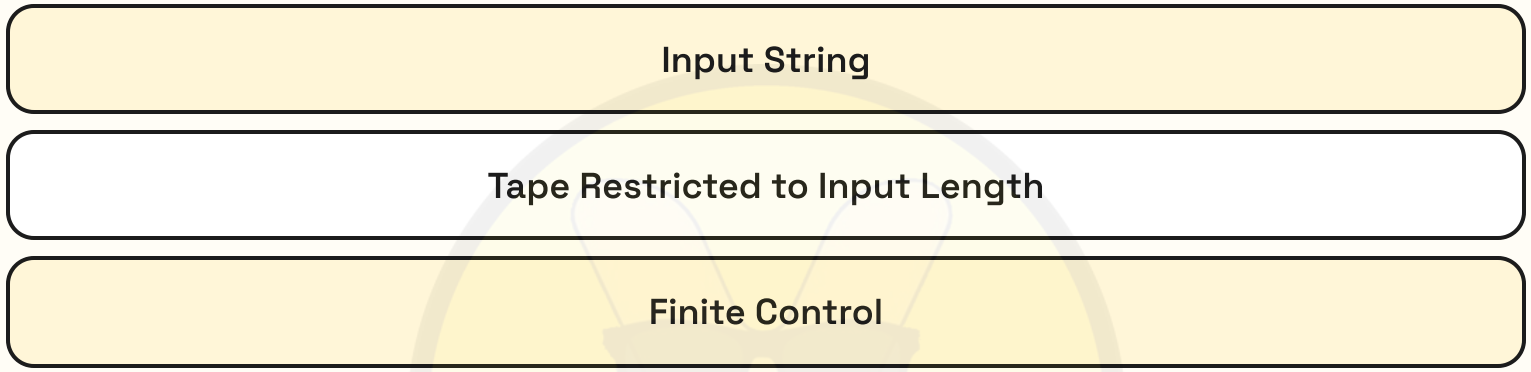
Universal Turing Machine (UTM)

A UTM is a programmable Turing Machine that can simulate any other TM.

- Takes the description of a machine M and an input w as its own input.
- Simulates the behavior of M on w.
- Proves that a single machine can perform any computational task if provided with the correct program.
- Forms the theoretical basis for the stored-program computer architecture.



Linear Bounded Automata (LBA)

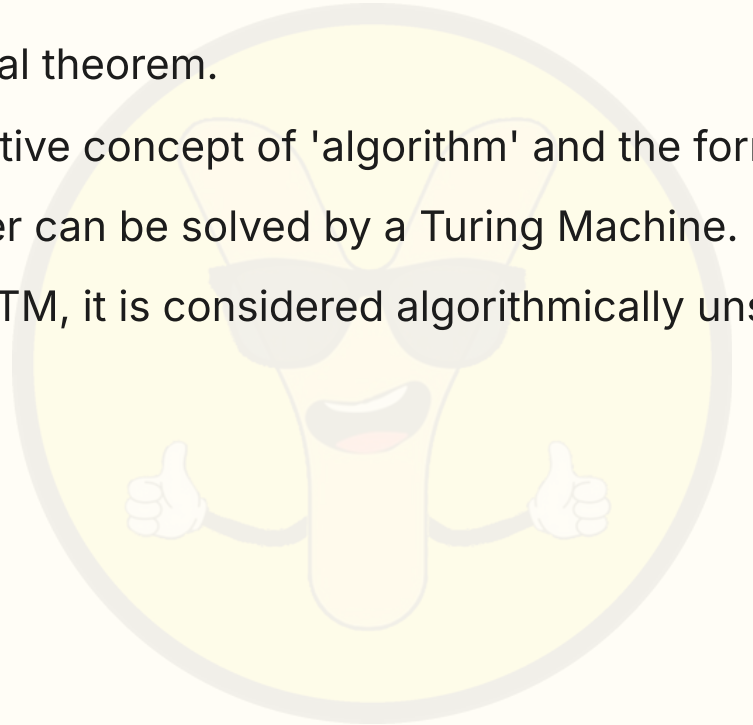


An LBA is a restricted TM where the tape head cannot move beyond the boundaries of the input string.

Church’s Thesis

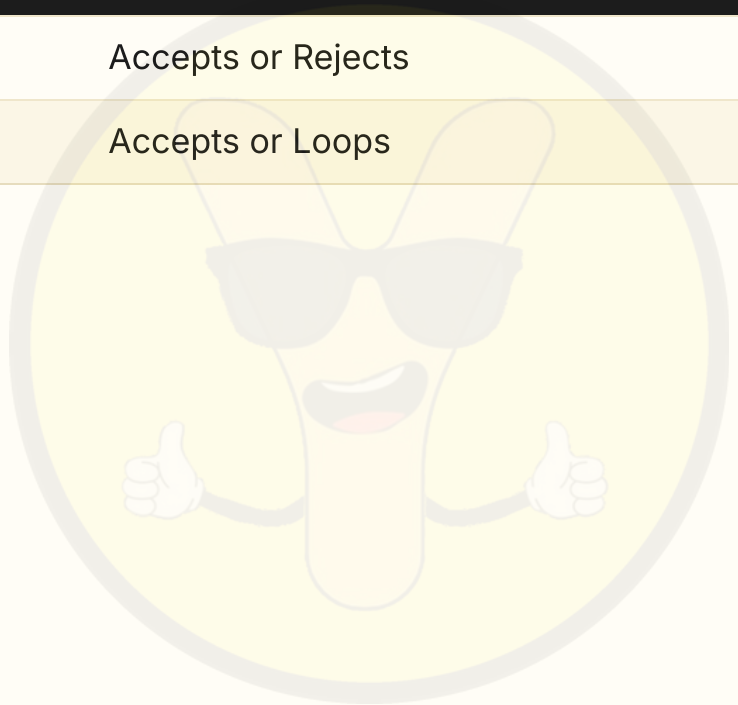
Church-Turing Thesis states that any effectively calculable function can be computed by a TM.

- It is a hypothesis, not a mathematical theorem.
- It bridges the gap between the intuitive concept of 'algorithm' and the formal definition of a TM.
- Any problem solvable by a computer can be solved by a Turing Machine.
- If a problem cannot be solved by a TM, it is considered algorithmically unsolvable.



Recursive vs Recursively Enumerable

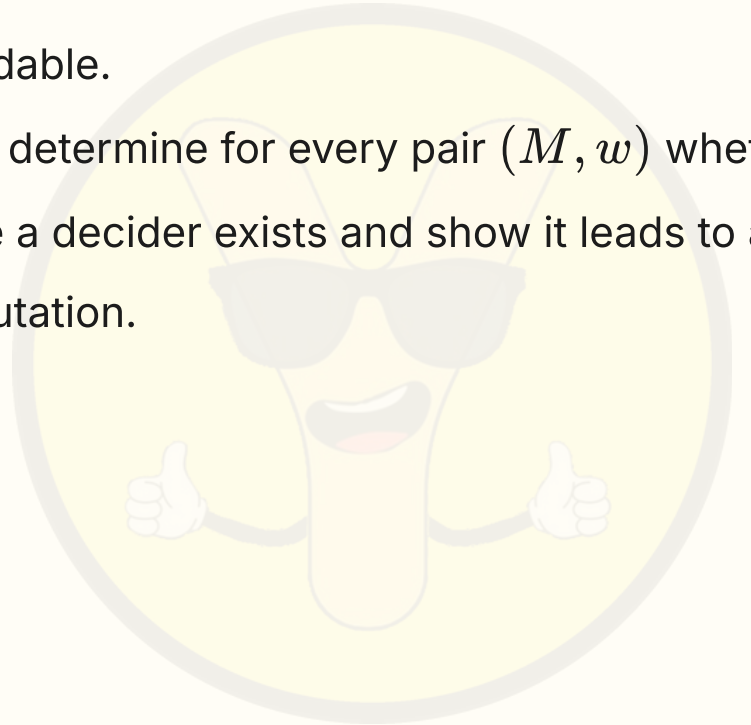
Language Type	TM Behavior	Halting
Recursive	Accepts or Rejects	Always Halts
Recursively Enumerable	Accepts or Loops	May not halt on reject



The Halting Problem

The Halting Problem asks if a given TM halts on a given input.

- Proven by Alan Turing to be undecidable.
- There is no algorithm (TM) that can determine for every pair (M, w) whether M halts on w.
- Proof uses diagonalization: Assume a decider exists and show it leads to a logical contradiction.
- This is a fundamental limit of computation.



Post's Correspondence Problem (PCP)

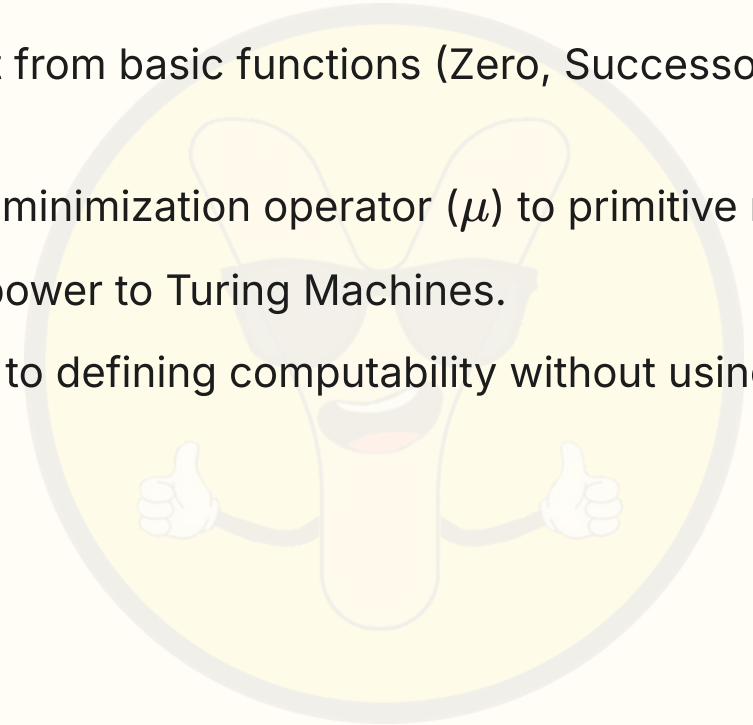
PCP is a classic undecidable decision problem involving string matching.

- Given two lists of strings $A = (w_1, w_2, ..., w_k)$ and $B = (x_1, x_2, ..., x_k)$.
- Task: Find a sequence of indices $i_1, i_2, ..., i_m$ such that $w_{i_1} w_{i_2} ... w_{i_m} = x_{i_1} x_{i_2} ... x_{i_m}$.
- It has been proven that no algorithm exists to solve PCP for arbitrary lists.
- Often used to prove other problems are undecidable via reduction.

Introduction to Recursive Function Theory

Recursive function theory defines functions using basic operations and recursion.

- Primitive Recursive Functions: Built from basic functions (Zero, Successor, Projection) using composition and primitive recursion.
- Mu-Recursive Functions: Adds the minimization operator (μ) to primitive recursive functions.
- These functions are equivalent in power to Turing Machines.
- Provides a mathematical approach to defining computability without using machines.



Exam corner

- **Q1** Define a Turing Machine and explain its 7-tuple representation.
- **Q2** Differentiate between Recursive and Recursively Enumerable languages.
- **Q3** Explain the Halting Problem and provide a brief sketch of its proof by contradiction.
- **Q4** What is a Universal Turing Machine? Why is it significant?
- **Q5** State and explain the Church-Turing Thesis.
- **Q6** Define Linear Bounded Automata (LBA) and its significance in the Chomsky hierarchy.
- **Q7** Describe the Post Correspondence Problem with an example.
- **Q8** Explain the concept of primitive recursive functions.

Unit 5 done. You are exam-ready.

Revise the Exam corner once more, then open Unit 1 on theycotes.com. Every AKTU subject, unit by unit, free. Notes by Chirag Jain, Ycotes.

