Lec 1:

- TOC helps us formulate mathematical models of a computer
- These mathematical models in turns help us understand what a computer can/can't do, efficiency of it and so on.

- Automata / machine / mathematical model. ⟶ all mean the same.

- But to make a computer understand all the shinanigans we need a pattern maker which is where GRAMMARS come at play.

- Next we have PUSH DOWN AUTOMATA which will be used to verify syntax.
- And at last TURING MACHINES where we learn about whether a given program will halt or not (THE Halting Problem).

Lec 2: Mathematical Recap:

- **Set :** Collection of distinct objects ; no repetations.
- **Cardinality :** Size of a set / no. of elements in the set.

- **Cartesian Product :**

If $A = \{1, 2\}$
$B = \{a, b, c\}$

then $A \times B = \{(1,a), (2,a), (1,b), (2,b), (1,c), (2,c)\}$

↳ These are called ordered pairs/tuples.

- **Relation :** $R \subseteq A \times B$; A relation is a subset of cartesian product.

$R = \{(1,a), (2,a), (2,b)\}$
implying $R \subseteq A \times B$

- **Function :** Special type of relations.

→ It says that an element of set A can be mapped to

only one other element in B.

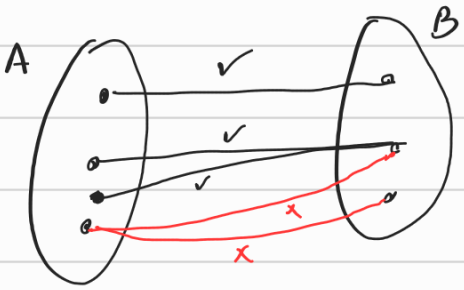

Fig: The one shown in red cannot happen.

- **Graphs :** It is a collection of vertices and edges. (V, E)

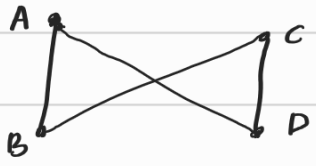

Fig: undirected graph.

- **Sequences :** It is a collection of elements (SET) where the order matters.

Ex: (A, B, C, D, ... Z) : seq of alphabets

(1, 1, 2, 3, 5, 8, ----) : seq of fibonacci numbers

One more fundamental difference of a set vs sequence is that sequences allow repetations.

meaning : {a, b, c, d, a, b, c, d, a, b, c, d} is not a set but a valid sequence.

One other way to put is that a sequence is a function from natural numbers to objects (where order matters).

Lec 3 : Alphabets, Strings, Languages, Grammar

Alphabet : It is nothing but a finite set of symbols.

Is represented using Σ

The empty alphabet is represented using $\Sigma_{empty} = \phi$

Ex: $\Sigma_{eng} = \{a, b, c, d \ldots, z\}$ $\Sigma_{bin} = \{0, 1\}$.

$\Sigma_{hindi} = \{अ, आ, इ, ई \ldots\}$

Strings/Words : finite sequence of alphabets ; cannot be infinite.

Length of a string = # no. of alphabets.

Empty string $\Rightarrow$ len = 0 ; Is represented by $\in$ (epsilon)

$|\in| = 0$

$\in . S = S . \in = S$ {concatenation}

Σ^* = set of all strings that can be formed using Σ including empty strings.

Language : It is a subset of Σ^*

A symbol 'L' is used for the denote a language.

Grammar : Set of rules to produce valid strings in a formal language.

Substring : Ex : abcdede — 'bcd' is a substring.

Formally, u is a substring of v if $\exists\ x u y = v$ where $u, x, y, v \in \Sigma^*$

Prefix : Ex : 01 0101 (u = 01, x = 0101) $\Sigma = \{0, 1\}$

u is a prefix of v if $\exists\ x$ s.t $ux = v$

Similarly, we can define what a suffix is.

Concatenation of strings : $u . v$ such that $u, v \in \Sigma^*$