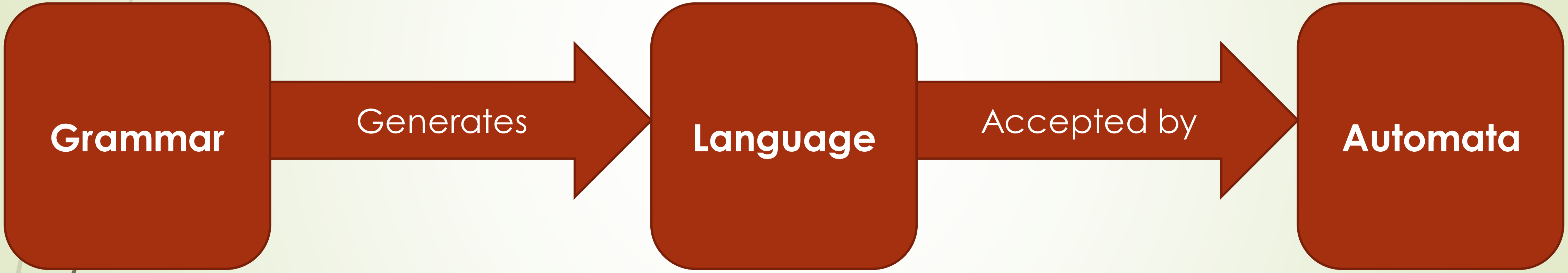
Theory of Computation

Presented by - JPK

Course Overview
One of the most fundamental course in Computer Science

- Mathematical preliminaries
- Formal Languages
- Finite State Automata
- Regular Language and Regular Grammar
- Context free language and Context Free Grammar
- Pushdown Automata
- Turing Machine

Languages, Grammars and Automata



Languages, Grammar and Automata

- Grammar is a language generating device
- Mathematical model of Grammar
- Automata is a language accepting device
- Mathematical model of Automata/Computation

Languages, Grammars and Automata

Grammar	Language	Automata
Type 0 Grammar Unrestriced Grammar	Recursively Enumerable Language	Turing Machine
Type 1 Grammar Context Sensitive Grammar	Context Sensitive Language	Linear Bounded Automata
Type 2 Grammar Context Free Grammar	Context Free Language	Pushdown Automata
Type 3 Grammar Regular Grammar	Regular Language	Finite State Automata

Theory of Computation
What is computation?

- **Problem Examples:**
- Given a number n, is it even?
- Given a number n, is it prime?
- Addition of two numbers
- Searching an element in a list
- Sorting a list of elements
- Definition – A computation is any type of calculation that includes both arithmetical and non arithmetical steps and follows a well defined model to solve a particular **problem**.
- A computer is a device that performs computation.

Basic Goal of Theory of Computation

- This course is **not about writing algorithms for a problem.**
- In this we study **“Can we have an algorithm / program for a problem or not”.**
- Unfortunately very tiny fraction of problems admits an algorithm. Most problems do not admit any algorithm.
- **For example given a program in c++/java will it halt or go in infinite loop is undecidable.**
- There many mathematical models of computation such as lamda calculus, finite state automata, pushdown automata, turing machines etc
- We will study FSA, PDA and TM as model of computation **to solve problem**

Mathematical preliminaries

- Set theory concepts
- Finite set and infinite set
- Subset, Null set, universal set
- Set operations such as union, intersection, complement, set difference
- Cartesian Product, Relation and Function
- Mathematical induction
- Graph and Tree
- Logic

Formal Languages

- Symbol
- Alphabet
- String
- Language and operations on languages
- Chomsky classification of languages
- Languages and automata
- Languages and grammar
- Derivation

Finite State Automata

- Deterministic Finite Automata DFA
- Non Deterministic Finite Automata NFa
- Epsilon NFA
- Conversion of NFA to DFA

Regular Language and Regular Grammar

- Regular Language
- Closure properties of regular set / language
- Regular expression
- Equivalence of Regular expression and FSA
- Regular Grammar
- Equivalence of Regular grammar and FSA
- Pumping lemma for regular language

Context Free Language and Context Free Grammar

- Definition and examples of context free language CFL
- Definition and examples of context free grammar CFG
- Derivation and derivation trees
- Ambiguity in CFG and CFL
- Normal forms of CFG

Pushdown Automata

- Deterministic and non deterministic Pushdown Automata PDA
- Acceptance by PDA
- Design of PDA
- Equivalence of CFG and PDA

Turing Machine

- Basic definition of Turing Machine TM
- Acceptance by TM
- Church Turing thesis
- TM as transducer (input output device)
- Design of TMs