## Computer Theory

Background

Automata Theory is a branch of theoretical computer science that deals with the study of abstract machines and their computational capabilities.

**Finite Automata (FA)** and **Pushdown Automata (PDA)** both types of automata, but they have key differences in terms of their computational power.

|  |  |
| --- | --- |
| **Finite Automata (FA)** | **Pushdown Automata (PDA)** |
| simplest form of automata | have an additional component called a stack.  A stack allows for recognition of patterns that involve nested structures |
| limited to recognizing regular languages | recognize context-free languages, which are a more powerful class of languages compared to regular languages. |
| finite set of states, an input alphabet, transition rules, an initial state, and a set of accepting (or final) states |  |
| limited in terms of memory; they can only recognize regular languages. | can accept languages that can be described by context-free grammars and are capable of recognizing nested structures, such as matching parentheses. |
| Regular languages can be described by regular expressions, and finite automata are equivalent in power to regular expressions. |  |

Pushdown Automata Theory

|  |  |
| --- | --- |
| Section 1 | Context-Free Grammars  Grammatical Format  Pushdown Automata  CFG = PDA |
| Section 2 | Non-Context-Free Languages  Context-Free Languages  Decidability |
| Section 3 | Turing Machines  Post Machines  Minsky's Theorem  Variations on t he TM |
| Section 4 | TM Languages  The Chomsky Hierarchy  Computers |

**Lesson 1**

Context-Free Grammars (CFG)

Context-Free Grammars (CFG) are a formal way of describing the syntax or structure of languages.

They play a crucial role in the analysis and design of programming languages, compilers, and various tools in computer science.

A context-free grammar, CFG, is a collection of three things:

**Terminals:** An alphabet of letters that appear in the final strings of the language (words of a language)

**Non-terminals:** Symbols that can be replaced by other symbols according to the rules.

*(designated by capital letters)*

Start Symbol: for “start here”. Specifies where the generation of strings begins.

#Expr (Expression): Represents arithmetic expressions.

#Term: Represents terms in arithmetic expressions.

#Factor: Represents factors in arithmetic expressions.

#Number: Represents numerical values in the expressions.

Expr -> Expr + Term | Expr - Term | Term

Term -> Term \* Factor | Term / Factor | Factor

Factor -> (Expr) | Number

Number -> 0 | 1 | 2 | ... | 9

**Production Rules:** Define how non-terminals can be replaced by sequences of terminals and/or other non-terminals.

A finite set of productions of the form

One Nonterminal finite string of terminals and/or Non-terminals

A language generated by a

CFG is called a context-free language, abbreviated **CFL.**