**Chapter 12: REAL-LIFE APPLICATIONS**

**Topic – 1: Overview**

* **Automata** is study about **fundamental concepts** in computation which forms basis of **computer science**.
* Automata are basically study of **computable mathematical models**.
* It helps in **recognition** of **text** or **data patterns**.
* Each type of automata has its **unique set of features**.
* Tells how computer programs **behave**.
* Also directly used in **compiler & interpreter designing**.

**Topic – 2: Applications**

**Finite Automata**

* ***Recognizing pattern in text or other data –*** Text processing software & spam filter etc.
* ***Implementing regular expressions –*** Matching input pattern in programming languages.
* ***Digital circuits –*** Helps in making optimal design of a circuit.
* ***Designing compilers & interpreters***
* ***Identifying bugs in OS & networking***
* ***Machine learning & NLP –***Recognizing pattern & make predictions.

**Pushdown Automata**

* ***Parsing CFGs –*** Not only CFGs but also parsing programming language syntaxes.
* ***Stack based memory management***
* ***Parsing sentences in NLPs***
* ***Context-sensitive languages –*** Mostly used in NLPs and DBMS.
* ***Creating network protocols***

**Linear Bounded Automata (LBA)**

**Note!**

**🡪 LBAs are a type Turing machines which have limited amount of memory.**

**🡪 Amount of memory is directly proportional to the input size.**

* ***Model verification –*** Verifying workings of hardware & software systems.
* ***String matching –*** Checking patterns in large amount of data.
* ***NLPs –*** Analysis & generation of sentences.
* ***Image & speech recognition –*** LBA can classify various elements from images & speech.
* ***Genome assembly –*** Analyse complex DNA structures & reconstruct complete DNA sequences of an organism.

**Turing Machine**

* ***Artificial intelligence –*** Machine learning & NLPs for recognizing and learning from large set of data.
* ***Cryptography –*** Encryption, decryption & generating random numbers.
* ***Algorithmic complexity theory –*** Time & space complexity.
* ***Compiler design –*** Can be used in parsing & translating phases.

**Topic – 3: Expressive Powers Of Different Automata**

**Definition**

* **Expressive power:** It gives an idea about **how many** languages can be **recognized & generated** by each level of automata.

**Various Automata’s Expressive Powers**

* ***Finite automata –*** Recognize regular languages defined by regular expressions or finite state machine.
* ***Pushdown automata –*** Recognizes context-free languages defined by context-free grammar.
* ***Linear bounded automata –*** Recognizes context-sensitive languages defined by context-sensitive grammar.
* ***Turing machine –*** Recognizes recursively enumerable languages defined by recursive functions.

**Note!**

**🡪 Class of recursively enumerable languages includes all other types of languages.**

**🡪 Expressive power increases when memory & computational power increases.**

**🡪 Choice on type of automaton depends on the type of requirement in our problem.**